تشخیص کور الگو های همزمانی تکرارشونده در رشته بیت خطادار دریافتی

حمید محمّدصادقی[1]، حسین خالقی بیزکی[2]، حمیدرضا کاکایی مطلق[3]

۱کارشناسی ارشد مخابرات سیستم، مجتمع دانشگاهی برق و الکترونیک، دانشگاه صنعتی مالک اشتر تهران

۲دانشیار، مجتمع دانشگاهی برق و الکترونیک، دانشگاه صنعتی مالک اشتر تهرانhbizaki@gmail.com

۳ دانشجوی دکترای مخابرات، دانشگاه جامع امام حسین(ع)

تاریخ دریافت: ۹۳/۱۰/۲۹ تاریخ پذیرش: ۹۵/۵/۱۵

چکیده

یکی از مهم ترین بخش های یک سیستم مخابراتی، ایجاد همزمانی بین فرستنده و گیرنده است. بنابراین در شرایطی که الگوهای همزمانی برای گیرنده نامشخص باشد، پیدا کردن ساختار فریم و الگوهای موجود در آن ضروری است، درغیر این صورت گیرنده نمی تواند به یک متن واضح یا یک متن رمزنگاری شده دست یابد. به طور کلی فریم ها می توانند در لایه فیزیکی دارای طول بزرگ یا کوچک باشند و هرکدام‌طبق کاربرد، یک یا چند الگوی همزمانی داشته باشند. در این مقاله روش جدیدی برای تشخیص کور الگوهای همزمانی موجود در ساختارهای متفاوت فریم و در خطاهای مختلف کانال BSC ارایه شده است. در این روش ابتدا بر مبنای مفهوم آنتروپی، طول الگوی همزمانی پیدا می شود و در ادامه با استخراج یک ویژگی روی پنجره ای که در حال پیشروی در رشته بیت است، مکان دقیق الگو تعیین می گردد. به طور کلی روش ارایه شده برای تشخیص یک یا چند الگوی همزمانی با طول کوچک یا بزرگ که درساختارهای متفاوت فریم به کار گرفته شده اند، قابل استفاده است. نتایج به دست آمده با استفاده از شبیه سازی نشان می دهد روش مطرح شده در شرایطی که حجم دادهٔ در دسترس بسیار کم باشد و رشته بیت خطای زیادی داشته باشد، کارآیی مطلوبی دارد.

کلیدواژه

تشخیص کور، الگوی همزمانی، آنتروپی رنیی، استخراج ویژگی، پنجره لغزشی، تئوری اطلاعات، تحلیل رشته بیت

مقدمه

یکی از مراحلی که در برقراری یک ارتباط مخابراتی بی‌سیم ضروری است، ایجاد همزمانی بین فرستنده و گیرنده است. برای این منظور، فرستنده و گیرنده یک «الگوی همزمانی[1]» را بین خود توافق می‌نمایند. طول و محتوای این الگو باید به نحوی انتخاب گردد که احتمال وقوع آن در دنباله داده بسیار کم باشد. فرستنده در ابتدای هر بلوک داده، این الگو را ارسال می‌نماید، گیرنده نیز با مشاهده این الگو در دنباله دریافتی متوجه ابتدای هر بلوک خواهد شد و به این ترتیب فرآیند همزمانی انجام می‌گیرد. چنانچه به هر دلیلی این الگوی همزمانی در گیرنده موجود نباشد باید به روش کور به شناسایی کور الگوی همزمانی و سپس استخراج اطلاعات پرداخت.

۱–شناسایی کور الگوی همزمانی

مساله شناسایی الگوی همزمانی را می توان به دو طریق انجام داد: روش نیمه-کور و روش کور. در روش نیمه کور فرض بر این است که اطلاعات اولیه ای از الگوی همزمانی و/یا فواصل تکرار الگوی همزمانی در دسترس است و هدف تعیین مکان این الگوی همزمانی در دنباله دریافتی است. اما در روش کور، هیچگونه اطلاعات اولیه ای از الگوی همزمانی در دسترس نیست.

در مقالات [۱] و [۲] به جای استفاده از اطلاعات الگوی تکرار شونده، بر اساس اطلاعات ناشی از ساختار درهم نه[2] به همزمانی ابتدای فریم دست یافتند. اگر چه استفاده از درهم نه سبب کاهش سربار ارسالی می شود اما این روش زمانی قابل استفاده است که گیرنده تمامی اطلاعات درهم نه را دراختیار داشته باشد.

در [۳] با فرض اینکه الگوی همزمانی مشخص باشد و در فواصل ثابتی تکرار می‌گردد و نیز با فرض اینکه فواصل تکرار نیز مشخص باشد، نویسندگان تلاش نموده‌اند تا مکان الگوی مورد نظر را در داده دریافتی را با محاسبه همبستگی سیگنال دریافتی بدست آورند. عیب روش همبستگی کارآیی پایین آن در شرایط نویزی می باشد به همین دلیل نویستندگاه برای غلبه بر خطا نیز روش خود را تا اندازه ای بهبود داده اند. عیب مقاله [۳] در این است که به دنبال الگوی مورد نظر در فواصل مشخص شده از داده می‌گردد. اینکار مانند عملیاتی است که گیرنده‌ی مجاز نیز برای برقراری

1 Synchronization pattern

2 Interleaver

همزمانی آنرا انجام می‌دهد. در حالیکه در این مقاله به دنبال یافتن روشی هستیم که اولا الگویهای تکرار شونده ممکن اس در فوصل مشخیص تکرار نشوند و ثانیا بیش از یک الگوی تکرار شوند در داده دریافتی وجود داشته باشد که آن هم می تواند طولی متفاوت با دنباله اولی داشته باشد.

در [۴] نویسندگان با شرایطی نظیر مقاله [۳]، با تشکیل ماتریسی از داده ها و استفاده از نگاشت بیتی به یافتن مکان الگوی همزمانی پیشنهاد داده اند. از جمله ای ایرادات دو روش قبلی این است که تنها برای یک مساله خاصی قابل استفاده بوده و علاوه براین، مساله را با فرض مشخص بودن الگوی همزمانی و نیز فرض مشخص و ثابت بودن فواصل تکرار الگوی همزمانی حل نموده اند. در حالی که در مساله شناسایی کاملاً کور، که در این مقاله پیشنهاد می شود، بدون دانستن الگوی همزمانی و حتی فواصل تکرار آن، بصورت کور مقدار، قادر به تعیین مکان و فواصل تکرار الگوی همزمانی برای طولهای تکرار شونده کوچک و بزرگ می باشد. بطوری که برای طولهای بزرگ روشی کاملا ابتکاری پیشنهاد شده است که طی دو مرحله تنظیم اولیه و تنظیم دقیق طول الگوی تکراری را پیدا می کند.

یکی دیگر از نکات برجسته مقاله حاضر استخراج دنباله های تکراری در شرایطی است که الگوهای تکراری با تناوب نامنظم تکرار شوند. بدیهی است که سایر مقالات بررسی شده‌ی قبلی قادر تحت این شرایط توانایی استخراج دنباله های همزمانی را ندارند. از جمله مشکلات الگوریتم های ارایه شده در مقالات قبلی این است که قادر به به شناسایی تنها یک الگوهای تکرار شونده هستند در حالیکه در این مقاله، دو یا هر تعداد دلخواه الگوی تکرار شونده، حتی با طولها و پریود تکرار متفاوت، قابل استخراج می باشند.

در یک تلاش دیگر، بسیاری از مقالات که به مساله یافتن همزمانی بصورت کور پرداخته‌اند، با فرض اینکه آخرین بلوک استفاده شده در فرستنده یک کد اصلاح خطا باشد، این موضوع را بررسی نموده‌اند [۱۲-۵]. ایده اصلی در انجام این کار به این صورت است که با فرض وجود یک کد اصلاح خطای مشخص در آخرین لایه از فرستنده و با فرض معلوم بودن ماتریس بررسی توازن آن کد، می-توان عمود بودن بردارهای ماتریس بررسی توازن را بر دنباله دریافتی به ازای آفست‌های مختلف آن محاسبه نمود. قطعاً زمانی که دنباله دریافتی به ابتدای کلمه کد برسد مقدار تست عمود بودن، حداکثر خواهد گردید. در بین این مقالات عده‌ای نیز تلاش نموده‌اند با کمک گرفتن از روش‌های کدگشایی نرم تصمیمی، به بالا بردن بازده این روش کمک نمایند.

در [۱۳] فرضیات مسالهٔ یافتن همزمانی بصورت پیچیده تری مطرح گردیده است. در این مقاله تلاش شده است تا میزان آفست را برای درهم‌سازی که روی یک کد کانولوشنال اعمال گردیده است، بدست آورد.

در مقاله حاضر فرض شده است که طول الگوی همزمانی و مکان آن مشخص نیست، هم چنین فواصل بین تکرار الگو (ها)ی همزمانی نیز تغییر می کند. بنابراین شرایط و فرض های محدودکننده ای که در مقالات قبلی برای استخراج الگو ذکر شده است، در این مقاله کنار گذاشته می شود. علاوه براین هیچ فرضی در مورد وجود یا عدم وجود کدهای کنترل خطا نیز نشده است و تلاش شده است تا بصورت کاملا کور اطلاعات الگوهای تکراری بدست آید.

ساختار مقاله به این صورت است: در بخش دوم به مفهوم آنتروپی رنی و آنتروپی هم پوشان، به منظور تشخیص سریع الگوهای همزمانی، پرداخته می شود. در رابطه آنتروپی رنیی پارامتر مهمی وجود دارد که در بخش سوم نشان داده می شود این پارامتر نقش مهمی در تشخیص طول الگو ایفا می کند. بطوریکه ثابت می کنیم که آنتروپی رنیی با استفاده از همین پارامتر، حجم رشته بیت مورد نیاز برای پیدا کردن الگو را به طور قابل ملاحظه ای کاهش می دهد. در بخش چهارم با استفاده از آنتروپی رنیی و استخراج یک ویژگی[3] روی رشته بیت دریافتی ، الگوریتمی برای پیدا کردن طول و مکان الگو به صورت کور ارایه می شود. الگوریتم به گونه ای است که یک پنجره لغزشی با گام یک بیت روی رشته بیت حرکت می کند و در هر گام ، یک ویژگی روی بیت های داخل پنجره اِعمال می شود. تغییرات ویژگی مطرح شده می تواند به خوبی عبور پنجره از روی الگو را نشان دهد و کاربر را به مکان الگو در رشته بیت برساند. کارآیی استخراج ویژگی مطرح شده در خطاهای ۰/۰۱ تا ۰/۲ کانال BSC در بخش پنجم بررسی می شود. نتایج نشان می دهند ویژگی مطرح شده می تواند حتی در خطاهای زیاد کانال، با دقت یک بیت مکان الگو را تشخیص دهد.

۲- تعاریف و مفاهیم

۲-۱-مفهوم آنتروپی هم پوشان

آنتروپی[4] به معنای متوسط میزان ابهام رخداد یک پدیده است. از آنجا که الگوهای همزمانی در یک رشته بیت چندین بار تکرار می شوند، انتظار می رود این الگوها میزان آنتروپی را کاهش دهند. بنابراین برای تشخیص الگو در فریم، از آنتروپی استفاده می شود.

آنتروپی هم پوشان برای یک رشته بیت به ازای طولهای ۲ بیتی ، ۳ بیتی ، ... ، n بیتی های موجود در آن رشته بیت تعریف می شود. مثلا برای محاسبه آنتروپی ۲ بیتی، پنجره ای با طول ۲ روی دو بیت سمت چپ رشته بیت قرار می گیرد. این پنجره با گام یک بیت به صورت هم پوشان پیشروی می کند و هر بار بیت های داخل پنجره به سمبول تبدیل می شود. به این ترتیب رشته بیت، تبدیل به سمبول های ۰، ۱، ۲ و ۳ خواهد شد (بیت های انتهای

3 Feature extraction
4 entropy

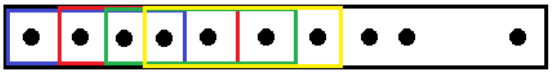
رشته بیت که داخل یک پنجره قرار نمی گیرند حذف می شوند). سپس، احتمال هر یک از سمبول های ۰، ۱، ۲ و ۳ در کلّ بیت های دریافتی محاسبه شده و با جاگذاری در رابطه زیر آنتروپی (شانون) دوتایی محاسبه می گردد:

(۱)

$$H(p) = -\sum_{i=1}^{4} p_i \log p_i$$
$$= -(p_{00}\log p_{00} + p_{10}\log p_{10} + p_{01}\log p_{01} + p_{11}\log p_{11})$$

در رابطه (۱)، برای آنتروپی دو بیتی، p_{00} احتمال رخداد دنباله "۰۰" در رشته بیت است. برای محاسبه این احتمال، تعداد رخدادهای ۰۰ بر تعداد کل دوبیتی ها در رشته بیت ، وقتی که پنجره به صورت هم پوشان پیشروی می کند، تقسیم می شود. به همین ترتیب، p_{01} احتمال رخداد دنباله "۰۱"، p_{10} احتمال رخداد دنباله "۱۰" و p_{11} احتمال رخداد دنباله "۱۱" در رشته بیت است.

هر گاه پنجره لغزشی برای یافتن الگو با گام یک بیت پیشروی کند، طول فریم چه ثابت باشد و چه متغیر، برای تحلیل گر هیچ تفاوتی نمی کند. چون الگو هر جای فریم قرار داشته باشد قطعا داخل یکی از پنجره ها قرار گرفته و در محاسبات آنتروپی وارد می شود (شکل ۱).



شکل ۱. پنجره لغزشی هم پوشان در تحلیل آنتروپی

۲-۲-آنتروپی رنیی هم پوشان

آنتروپی رنیی[5] به صورت رابطه زیر تعریف می شود :

$$H_{RENYI}(p) = \log_2\left(\frac{\sum_{i=1}^{n} p_i^{\,q}}{1-q}\right) \qquad (۲)$$

در رابطه (۲) ، p_i ها احتمال سمبول ها در رشته بیت و q پارامتر آنتروپی است. آنتروپی شانون برای تشخیص رویدادهایی با اهمیت خاص یعنی همان الگو های تکرارشونده مناسب است ولی اگر طول الگو بزرگ باشد، حجم محاسبات پردازشی بسیار زیاد می شود و عملا کارآیی لازم را ندارد، در صورتی که آنتروپی رنیی با استفاده از پارامتر q این مشکل را به خوبی برطرف می کند. این پارامتر حساسیت آنتروپی را به ازای حتی چند بار تکرار شدن الگو بسیار بالا می برد و در نمودار آنتروپی کاهش شدید و مشخصی ایجاد می کند که سبب تشخیص مکان الگوی تکرار شوند می شود.

۳– آنتروپی رنیی در تشخیص الگوهای با طول کوچک

به طور کلی برای محاسبه آنتروپی، از هیستوگرام و یا تعداد تکرارهای هر دنباله چندتایی، به منظور محاسبه احتمال وقوع سمبل ها، استفاده می شود. مثلا برای یافتن یک الگو با طول ۲۰ باید هیستوگرام ۲۰ تایی محاسبه شود و البته محاسبه هیستوگرام ۲۰ تایی و مقادیر بیشتر از آن با نرم افزارهای شبیه ساز موجود، از جمله MATLAB، به پردازش زیادی نیاز دارد. به همین دلیل در این مقاله، الگو های با طول کمتر از ۲۰ بیت "الگوهای با طول کوچک"و الگو های با طول بیشتر از ۲۰ بیت "الگوهای با طول بزرگ" تعریف گردیده است.

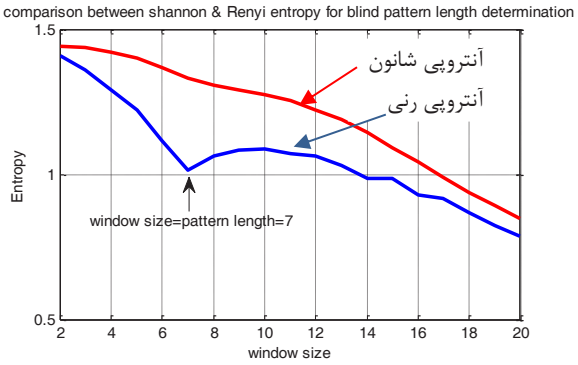
فرض کنید رشته بیتی از پشت سرهم قراردادن ۱۰۰ فریمِ ۳۵ بیتی به صورت شکل ۲ تولید شده است. هر فریم دارای یک الگوی ۷ بیتی ثابت است و این الگو می تواند برای هر فریم، متفاوت با بقیه فریم ها در جاهای مختلفی قرار داده شود (یعنی می تواند دارای تناوب تکرار متقاوتی باشد). بخش داده (همه بیت های غیر از الگو) در فریم شکل ۲، و در تمام فریم هایی که در ادامهٔ مقاله آورده می شود، تصادفی فرض شده است.

۲۵ بیت داده تصادفی	۷ بیت الگو	۳بیت داده تصادفی

شکل ۲. ساختار فریم ۳۵ بیتی فرضی با یک الگوی ۷ بیتی تکرارشونده

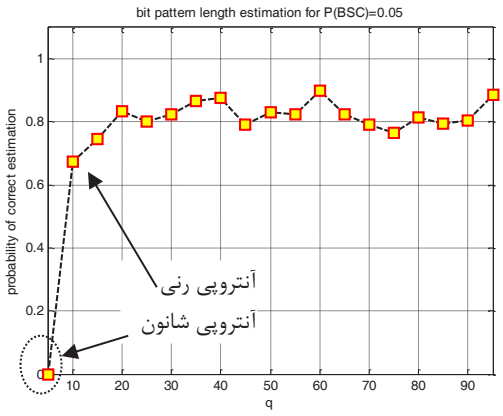
در رابطه (۲) در صورتی که $q \to 1$ میل کند، آنتروپی رنیی معادل با آنتروپی شانون می شود. اگر q عدد مثبت بزرگی باشد، آنتروپی نسبت به رویدادهایی که بیشتر اتفاق می افتند، بسیار حساس شده و اگر q عدد منفی بزرگی باشد، نسبت به رویدادهایی که به ندرت اتفاق می افتند، حساس می شود [۱۴]. به عبارت دیگر، توان q در آنتروپی رنیی، حساسیت بسیار زیادی نسبت به تکرار شدن الگو در رشته بیت ایجاد می کند. از این ویژگی می توان استفاده کرد و با در دسترس داشتن حجم بسیار کمی از رشته بیت، با توجه به حساسیت آنتروپی رنیی به رخدادهای مکرر، الگو های همزمانی را پیدا کرد. در شکل ۳ مقایسه ای بین آنتروپی رنیی و شانون برای تشخیص الگوی موجود در شکل ۲ انجام شده است. در این شکل پارامتر q برابر ۸۰ قرار داده شده است (در ادامه در مورد این پارامتر بحث خواهد شد). طبق شکل ۳، طول الگوی تکرار شونده برابر ۷ تعیین می شود. این شکل نشان می دهد آنتروپی رنیی برخلاف آنتروپی شانون می تواند با دقت مناسبی طول الگو را پیدا کند.

5 RENYI entropy



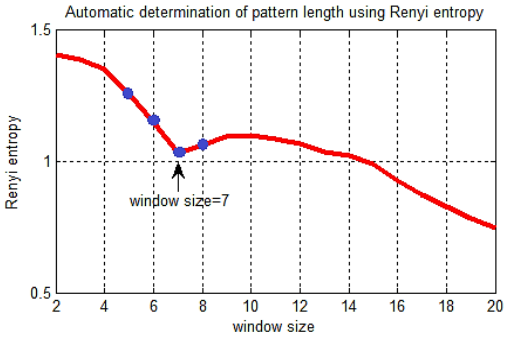
شکل ۳. مقایسه بین آنتروپی شانون و رنیی در تشخیص طول الگو

در شکل ۴ احتمال تشخیص صحیح طول الگو با آنتروپی رنیی و شانون مقایسه شده است. این شبیه سازی در شرایط یکسانِ کانال و حجم داده‌ی مساوی، انجام شده است. برای q های کوچک و نزدیک به ۱ (آنتروپی شانون) احتمال تشخیص صحیح طول الگو بسیار پایین است ولی برای q های بزرگ (q های بین ۲۰ تا ۹۰) (آنتروپی رنیی) احتمال تشخیص صحیح طول الگو زیاد است. مطابق این شکل مقدار مناسب برای q بین ۲۰ تا ۹۰ می باشد. در ادامه‌ی مقاله در مورد محدودهٔ مجاز q، یا همان کران بالا و پایین q ، بحث خواهد شد.



شکل ۴. مقایسه احتمال تشخیص صحیح طول الگو با آنتروپی رنیی و شانون (q کوچکتر از یک) در شرایط یکسانِ کانال و حجم داده

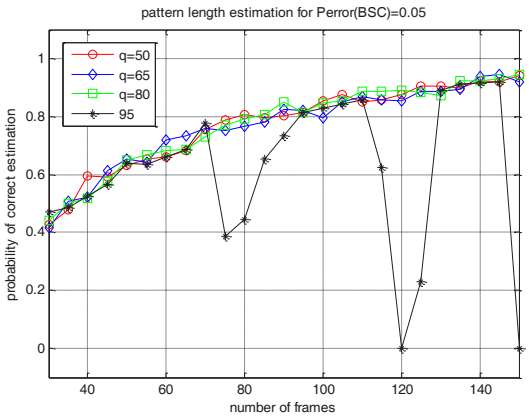
در صورتی که نیاز باشد درصد تشخیص صحیح طول الگو را به وسیله آنتروپی رنیی تعیین شود، باید طول الگو به صورت خودکار از روی نمودار آنتروپی رنیی تشخیص داده شود. برای این منظور از شیب نمودار آنتروپی استفاده می شود. هر نقطه ای روی نمودار آنتروپی که شیبِ دو نقطهٔ قبلش منفی و شیب نقطهٔ بعدش مثبت باشد و نیز از بین نقاط مشابه با این ویژگی، کمینه بیشتری داشته باشد به عنوان طول الگو انتخاب می شود (شکل ۵). از این روش برای تعیین درصد تشخیص الگوی تکراری بصورت خودکار استفاده خواهیم کرد.



شکل ۵. تشخیص خودکار طول الگو از روی نمودار آنتروپی

۳-۱-تعیین محدودهٔ مجاز پارامتر آنتروپی رنیی

پارامتر q در آنتروپی رنیی نمی تواند از حدی بزرگتر باشد، چرا که رابطه (۲) را بی نهایت می کند. در شکل ۶ با شبیه سازی برای ۶۰۰ بار تکرار نشان داده شده است که اگر پارامتر q مقادیری کمتر از ۹۵ داشته باشد، آنتروپی رنیی به خوبی کار می کند ولی در مقدار ۹۵، احتمال تشخیص الگوی ۷ تایی به ازای تعداد فریم های خاصی ، کاهش شدیدی به خود می گیرد. بنابراین در ادامهٔ کار برای اطمینان، میزان این پارامتر را همیشه کمتر از ۹۰ منظور خواهیم کرد و بر همین اساس، محدوده مجاز q درآنتروپی رنیی بین ۲۰ تا ۹۰ منظور می شود. البته در همه شبیه سازی های این مقاله، مقدار پارامتر q برابر ۸۰ قرار داده شده است تا احتمال تشخیص حداکثری بدست آید.



شکل ۶. تعیین محدوده مجاز پارامتر آنتروپی رنیی در محاسبات آنتروپی

شکل ۶ ، علاوه بر تعیین محدودهٔ q ، نشان می دهد با زیاد شدن تعداد فریم ها، طول الگو بهتر تخمین زده شده است.

۳-۲- حجم رشته بیت مورد نیاز برای تشخیص الگو

احتمال اینکه الگویی مشخص، در یک رشته n بیتی رخ دهد برابر است با :

$$p_{\text{occurrence}} = \frac{1}{2^n} \quad (۳)$$

پس برای اینکه الگویی با طول n بیت بصورت کور در یک رشته بیت پیدا شود، حداقل به تعداد $n \times 2^n$ بیت نیاز داریم تا تعداد قابل قبولی از الگو (جامعهٔ آماری کافی[6]) در رشته بیت وجود داشته باشد. جدول ۱، حداقل حجم دادهٔ مورد نیاز برای تشخیص طول الگو با آنتروپی شانون را نشان می دهد. این جدول نشان می دهد پیدا کردن الگو هایی با طول بزرگتر از ۵۰ بیت با آنتروپی شانون عملا ممکن نیست ولی با الگوریتم ارایه شده در این مقاله، پیدا کردن الگوهایی با طول های خیلی بزرگ با در اختیار داشتن فقط چند صد کیلو بیت امکان پذیر می شود.

جدول ۱. حداقل حجم مورد نیاز برای تشخیص طول الگو با آنتروپی شانون

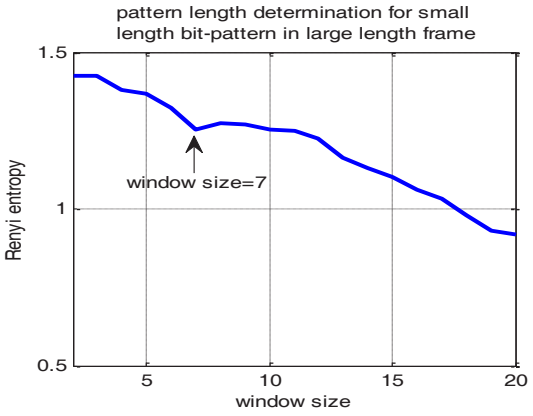
طول الگو	حداقل حجم مورد نیاز برای تشخیص طول الگو توسط آنتروپی شانون
۱۰	حداقل ۱/۲۵ کیلوبایت
۲۰	حداقل ۲/۵ مگابایت
۳۰	حداقل ۴ گیگا بایت
۵۰	بی نهایت

علاوه بر ویژگی های مطرح شده برای آنتروپی رنیی، این آنتروپی باید بتواند الگو هایی با طول کوچک که در فریم هایی با طول بزرگ قرار گرفته اند را تشخیص دهد. فرض کنید ۱۲ کیلوبایت از ساختار فریم شکل (۷) در دسترس گیرنده باشد:

۱۲۰ بیت داده تصادفی	۷ بیت الگو	۱۰ بیت داده تصادفی

شکل ۷. ساختاری با طول فریم بزرگ و الگوی ۷ بیتی موجود در فریم

شکل ۸ نشان می دهد در حالتی که طول فریم شکل ۷، تقریبا ۲۰ برابر طول الگو است، باز هم آنتروپی رنیی قادر به تشخیص طول الگو در حجم بسیار کم داده (۱۲ کیلوبایت) است.



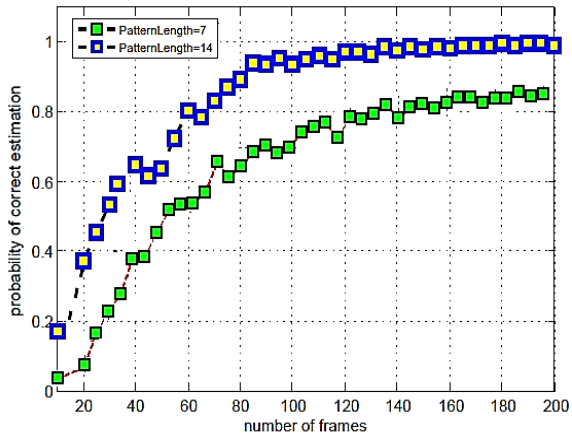
شکل ۸. آنتروپی رنیی برای طول فریم بزرگ و طول الگوی کوچک

۳-۳-توانایی آنتروپی رنیی در تشخیص دو الگو

ممکن است در برخی لینک های مخابراتی و یا ارتباطات مبتنی بر IP ، بیشتر از یک الگوی ثابت به کار رفته باشد. اگر دو الگو با طول های متفاوت در رشته بیت وجود داشته باشد (برای بیشتر از دو الگو نیز روش حل مشابهی داریم و کلیت مساله همچنان برقرار است)، آنتروپی رنیی در مرحله اول فقط قادر به تشخیص الگو با طول بزرگ تر است، چون فرضا اگر الگویی ۱۴ بیتی در رشته بیت وجود داشته باشد، قطعا الگو هایی ۱۳ بیتی ،۱۲ بیتی ، ... و ۲ بیتی نیز در رشته بیت وجود دارند، به همین دلیل آنتروپی رنیی فقط در مقدار ۱۴ کمینه می شود. برای حل این مشکل بعد از اینکه آنتروپی، طول الگوی ۱۴ تایی را تخمین می زند، روی رشته بیت هیستوگرام ۱۴ تایی گرفته می شود. مقدار ماکزیمم در هیستوگرام، همان رشته بیتِ الگوی ۱۴ تایی است. در ادامه، مکان این رشته بیت از طریق هم بستگی با کل رشته بیت مشخص می شود. پس از تعیین مکان، الگوی ۱۴ تایی از رشته بیت حذف می شود. در نهایت دوباره از رشته بیت آنتروپی رنیی گرفته تا این بار الگو با طول کوچکتر پیدا شود. احتمال تشخیص صحیح طول این دو الگو الگو برای ساختار فریم شکل ۹، در شکل ۱۰ نشان داده شده است. بخش داده در شکل ۹ می تواند تا بیش از ۲۰ برابرِطول الگو باشد.

داده تصادفی	۱۴ بیت الگو	داده تصادفی	۷ بیت الگو	داده تصادفی

شکل ۹. ساختار فریمی با دو الگوی متفاوت ۱۴ و ۷ بیتی



شکل ۱۰. احتمال تخمین صحیح دو الگو با طول متفاوت توسط آنتروپی رنیی در خطای ۰/۰۵ کانال، ۶۰۰ بار تکرار و پارامتر آنتروپی رنیی برابر ۸۰

از آنجا که الگوی ۷ تایی بعد از الگوی ۱۴ تایی تشخیص داده می شود، خطای تشخیص الگوی ۷ تایی به خطای تشخیص الگوی ۱۴ تایی وابسته می شود و نمودار تشخیص صحیح الگوی ۷ تایی طبق شکل ۱۰، زیرِ نمودار الگوی ۱۴ تایی قرار می گیرد.

6 Statistical population

۴- الگوریتم تشخیص طول و مکان الگوهایی با طول بزرگ بصورت کور

تشخیص الگو هایی با طول بزرگ با آنتروپی رنیی به تنهایی کافی نیست زیرا محاسبه هیستوگرام دنباله های طولانی، در عمل، بسیار مشکل است. برای تشخیص الگوهای بزرگ، در مرحله اول نرخ رشته بیت کاهش[7] داده شده، سپس آنتروپی رنیی به رشته بیت حاصل اِعمال می شود تا تقریبی[8] از طول الگو بدست آید. در ادامه با استخراج یک ویژگی مناسب تقریب به دست آمده، دقیق[9] می شود. مراحل کلی الگوریتم پیشنهادی را، با توجه به شکل (۱۳)، می توان بصورت زیر بیان نمود:

۱- خود همبستگی[10]: برای پیدا کردن الگو، با استفاده از استخراج ویژگی، به میزانی از رشته بیت برای تحلیل نیاز است که چندین مرتبه الگو در آن تکرار شده باشد. ماکزیمم های تابع خودهمبستگی رشته بیت می توانند بیان کننده تکرار الگوها و تعداد آنها در رشته بیت باشند. به عبارت دیگر این تابع، محدودهٔ جستجو برای یافتن الگو را مشخص می کند (البته با فرض معلوم بودن طول فریم می توان از این تحلیل صرف نظر کرد).

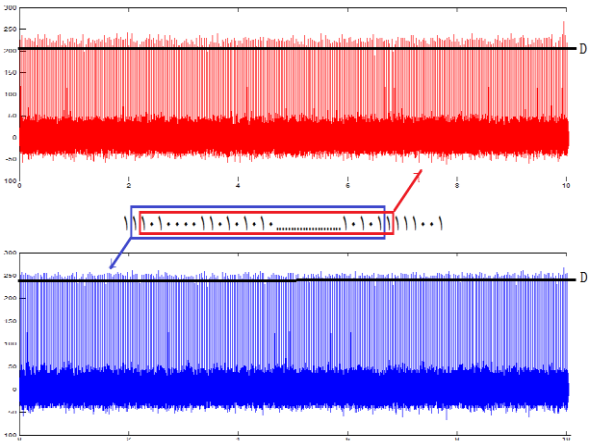
۲- کاهش نرخ[11]: از رشته بیت نمونه برداشته می شود. مثلا برای نمونه برداری (کاهش نرخ) معادل ۷، بیت های ۱ و ۸ و ۱۵ و ۲۲ و ... را برداشته و از رشته بیتِ حاصل آنتروپی رنیی هم پوشان گرفته می شود. در این صورت اگر فرضا الگویی ۱۲۱ بیتی در رشته بیت وجود داشته باشد و رشته بیت با مقدار ۷ نمونه برداری شود، الگویی با طول ۱۷ (بستگی به جای الگو در رشته بیت، ممکن است الگویی با طول ۱۸ نیز باشد)، در رشته بیت حاصل وجود دارد. با این روش به تقریبی از طول الگو خواهیم رسید.

۳- تعیین دقیق طول و مکان الگوی همزمانی: طول پنجره لغزشی برابر با طول تقریبی الگوی بدست آمده در مرحله قبل قرار داده می شود. حال باید ویژگی ای استخراج شود که بتواند گذرِ پنجره‌ی لغزشی از روی الگو را نشان داده و در نتیجه کاربر را به طول و مکان دقیق الگو در رشته بیت برساند. شکل ۱۱، نشان می دهد که ویژگی طراحی شده باید چگونه باشد تا مکان الگو تشخیص داده شده و طول دقیق آن بدست آید. این ویژگی باید بتواند به کاربر بفهماند که پنجره در کجای رشته بیت در حال عبور از روی الگو است.

شکل ۱۱. طرح مفهومی ویژگی استخراج شده جهت تعیین الگو

برای بدست آوردن این ویژگی، بازه‌ای به اندازهٔ کافی بزرگ از رشته بیت اصلی انتخاب می شود، طوری که الگوی مجهول حداقل چند مرتبه در این بازه تکرار شده باشد. بازه مطلوب با توجه به تابع خودهمبستگی و طول فریم انتخاب می شود.

در ادامه پنجره ای لغزشی با گام یک بیت روی رشته بیت اصلی پیشروی می کند و در هر گام تابع همبستگی متقابل[12] بین بیت های داخل پنجره و کل رشته بیت تعیین می شود. به این ترتیب به ازای هر گام پنجره یک نمودار (بردار) هم بستگی بدست می آید (شکل ۱۲). حال، روی هر نمودار هم بستگی پارامتر D بصورت حاصلضرب یک حد آستانهٔ T درطول تقریبی الگو (طول تقریبی پنجرهٔ لغزشی) تعریف می شود. در نهایت ویژگی مطلوب، میانگین مقادیر بیشتر از D تعریف می گردد. بنابراین محور عمودی شکل ۱۱ ، میانگین مقادیر بیشتر از D را نشان می دهد. مقادیر حد آستانهٔ T نیز بصورت تجربی، بر حسب میزان خطای کانال، در جدول ۲ به دست آمده است. نحوه محاسبه هم بستگی متقابل در شکل ۱۲ نشان داده شده است. در بخش نتایج شبیه سازی، روش به دست آوردن طول و مکان دقیق الگو بر اساس شکل ۱۱، بحث خواهد شد.



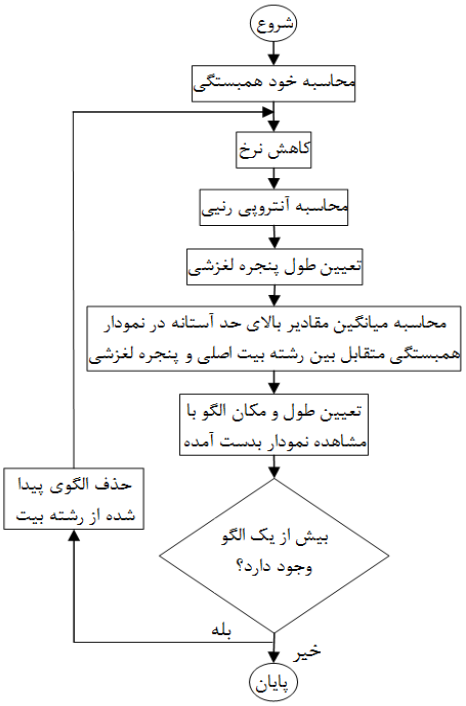
شکل ۱۲. هم بستگی متقابل با لغزش پنجره با گام یک بیت

شکل ۱۳ الگوریتم کلی تشخیص طول و مکان الگوی همزمانی در رشته بیت را نشان می دهد. باید ذکر شود در صورتی که برخی

7 Down sample
8 tune
9 Fine tune
10 Auto correlation
11 Down sample

12 Cross correlation

پارامترهای ارسالی فرستنده معلوم باشد، نیازی به اجرای همه مراحل الگوریتم نمی باشد.



شکل ۱۳. الگوریتم پیشنهادی برای تشخیص کور طول و مکان چندین الگوی با پریود غیر ثابت در رشته بیت با طول کوتاه و بلند

۵- شبیه سازی و تحلیل نتایج

میزان کارآیی الگوریتم پیشنهادی در شکل (۱۳) به ازای خطاهای مختلف کانال از طریق انجام شبیه سازی های متعدد در این بخش ارایه می گردد. بدین منظور، رشته بیتی با طول ۱۲/۵ کیلوبایت، طبق شکل ۱۴، تولید می شود. الگویی با طول ۱۲۰ بیت ، به طور تصادفی در رشته بیت قرار داده می شود تا فریمی با طول متغیر تولید گردد (یعنی پریود تکرار الگو ثابت نیست و این یک از تفاوت های عمده ای است که در هیچ یک از مقالات قبلی به آن پرداخته نشده بود).

۱۲۰ بیت الگو	داده	۱۲۰ بیت الگو	داده	۱۲۰ بیت الگو	داده

شکل ۱۴. ساختار فریم دلخواه برای بحث روی الگوریتم ارایه شده

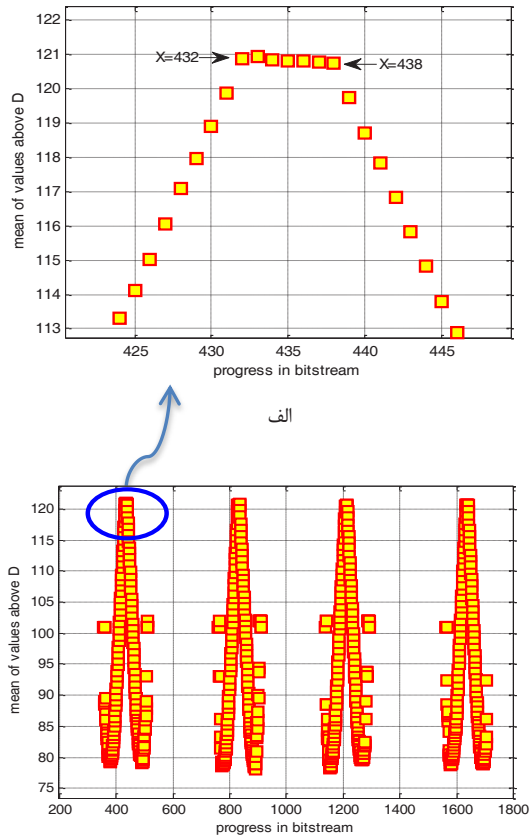
مقدار کاهش نرخ برابر ۷ قرار داده می شود. الگوریتم تشخیص الگو، طبق شکل ۵ به صورت خودکار طول الگو را برابر ۱۸ تعیین می کند. بنابراین طول تقریبی الگوی موجود با توجه به میزان کاهش نرخ برابر با ۱۲۶= ۷*۱۸خواهد بود. به عبارتی ۶ واحد خطا در بدست آوردن طول دقیق الگو وجود دارد (به دلیل کاهش نرخ ۷)، چرا که طول دقیق الگو ۱۲۰ بیت است. برای دقیق شدن طول الگو و اصلاح ۶ واحد خطا، الگوریتم ارایه شده اجرا می شود.

خروجی الگوریتم برای کانال BSC با خطای ۰/۰۰۱ در شکل ۱۵ نشان داده شده است.

بحث: به طور کلی ، طولِ پنجرهٔ لغزشی ای که در حال گذر از روی الگو است (که برابر با طول تقریبی الگو است)، یا بزرگتر از طول الگو و یا کوچکتر از آن، تخمین زده شده است. طبق شکل ۱۶- الف، اگر طول پنجرهٔ لغزشی بزرگتر از طول الگو باشد، در زمان لغزش پنجره از روی الگو، پنجرهٔ ۱ اولین لحظه و پنجرهٔ ۲ آخرین لحظه ای است که هم پوشانی کاملی بین پنجرهٔ لغزشی و الگو روی می دهد. برای شکل ۱۶- ب نیز همین مساله برقرار است، با این تفاوت که در این شکل، طول پنجرهٔ لغزشی کوچکتر از طول الگو است. طبق شکل ۱۶- الف و ۱۶- ب طول پنجرهٔ لغزشی، به ترتیب ۴ واحد بزرگتر و ۴ واحدکوچکتر از طول الگو است و در هر دو شکل، تعداد دفعاتی که پنجرهٔ لغزشی هم پوشانی کاملی با الگو دارد، ۵ مرتبه است، به عنوان یک قاعدهٔ کلی می توان از قاعده زیر استفاده نمود:

$$\text{تعداد دفعات هم پوشانی کامل} = \text{طول پنجره} - \text{اختلاف بین طول الگو} + 1 \quad (۴)$$

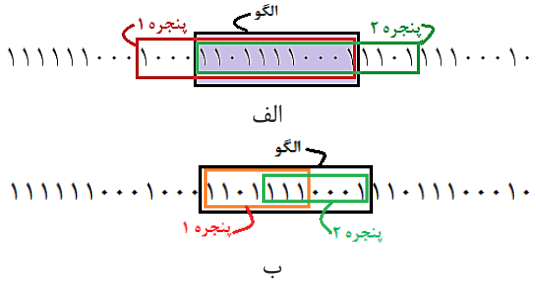
پنجرهٔ لغزشی با الگو



شکل۱۵. تعیین دقیق طول الگو از روی تابع ویژگی استخراج شده

تا زمانیکه پنجرهٔ لغزشی و الگو، هم پوشانی کاملی با هم داشته باشند، ویژگی مطرح شده در الگوریتم، یعنی میانگین مقادیر

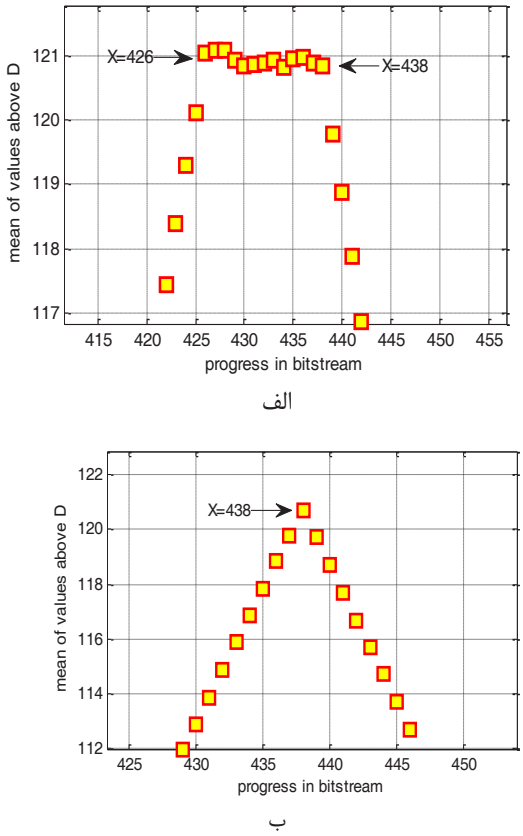
بیشتر از D، تقریبا ثابت می ماند و از بقیه زمان هایی که بین پنجره و الگو هم پوشانی کامل وجود نداشته باشد، مقدار بزرگتری دارد. به عبارت دیگر، تا زمانی که پنجرهٔ لغزشی در حال ورود به محدوده الگو است، شیب نمودارِ ویژگی بیان شده رو به افزایش است و تا زمانی که هم پوشانی کامل برقرار است، شیب ثابت می ماند و بعد از اتمامِ هم پوشانی یا همان خروج پنجره از الگو، شیب نمودار کاهش پیدا می کند (طبق طرح شکل ۱۱). شکل ۱۵، تحققِ طرح بیان شده در شکل ۱۱ را نشان می دهد که از روی آن می توان طول دقیق الگو را پیدا کرد.



شکل ۱۶. پنجرهٔ لغزشی عبوری از روی الگو الف) طول پنجرهٔ لغزشی بزرگ تر از طول الگو ب) طول پنجرهٔ لغزشی کوچکتر از طول الگو

در شکل ۱۵- ب ، ۷ نقطه روی نمودار (از بیت ۴۳۲ تا بیت ۴۳۸)، مقادیر تقریبا ثابت دارند و از بین این ۷ نقطه، نقاط ابتدایی و انتهایی، مربوط به اولین و آخرین لحظه ای هستند که پنجرهٔ لغزشی، هم پوشانی کاملی با الگو داشته است و در بقیه نقاط (۵ نقطه) نیز هم پوشانی کامل وجود دارد، به عبارتی تعداد دفعات هم پوشانیِ کامل برابر ۷ است. بنابراین طبق رابطهٔ ۴، اختلاف بین طول الگو و طول پنجره برابر ۶ خواهد بود، ولی هنوز مشخص نیست که طول الگو ۶ واحد بزرگتر از طول پنجرهٔ لغزشی است یا ۶ واحد کوچکتر از آن است. برای مشخص شدن این موضوع، یک بار طول پنجره لغزشی به میزان ۶ واحد بیشتر شده و الگوریتم پس از این اصلاح، اجرا می شود. یک بار هم طول پنجره لغزشی به میزان ۶ واحد کمتر شده و الگوریتم دوباره اجرا می شود. خروجی الگوریتم در هر یک از این دو حالت به ترتیب در شکل ۱۷- الف و ۱۷- ب نشان داده شده است.

باید دقت شود که اگر طول پنجرهٔ لغزشی با طول الگو دقیقا برابر شود، تعداد نقاط با مقدار ثابت روی شکل ۱۵- ب ، دقیقا یک نقطه می شود، چون در این حالت پنجره فقط یک بار هم پوشانی کامل با الگو خواهد داشت. بنابراین، شکل ۱۷- الف، نشان می دهد که اگر طول پنجرهٔ لغزشی ۶ واحد زیاد شود (یعنی معادل ۱۳۲=۶+۱۲۶ شود)، طول پنجرهٔ لغزشی برابر طول الگو نیست و طول الگوی ۱۳۲ اشتباه بدست آمده است. در حالیکه شکل ۱۷- ب، نشان می دهد که اگر طول پنجرهٔ لغزشی ۶ واحد کم شود (یعنی معادل ۱۲۰=۶-۱۲۶ شود)، طول پنجرهٔ لغزشی دقیقا برابر طول الگو شده است، چرا که تعداد نقاط با مقدار ثابت فقط یک نقطه است. به این ترتیب، با توجه به توضیحات مطرح شده، طول الگوی همزمانی برابر ۱۲۰ است که از بیت ۴۳۸ شروع شده و تا بیت ۵۵۷ ادامه دارد. به این ترتیب اجرای الگوریتم برای تعیین طول و مکان دقیق الگو در رشته بیت خاتمه می یابد.



شکل ۱۷. پیدا کردن محل دقیق الگو با الگوریتم ارایه شده

۵-۱-تعیین حد آستانه مناسب به صورت تجربی

از طریق تست های متعدد به صورت تجربی، می توان حد آستانهٔ T مناسب در الگوریتم تشخیص طول و مکان الگو را تخمین زد. طبیعی است که هر چه خطای رشته بیت بیشتر شود حد آستانهٔ مناسب پایین تر خواهد بود. جدول ۲، حد آستانهٔ مناسب برای خطاهای مختلف کانال باینری متقارن را نشان می دهد.

جدول ۲. حد آستانه الگوریتم تشخیص طول و مکان الگو

حد آستانه T	احتمال خطای کانال BSC
۰/۵۸۵	۰
۰/۵۷۵	۰/۰۰۱
۰/۵۵	۰/۰۱
۰/۵۴	۰/۰۲
۰/۵۲۵	۰/۰۴
۰/۴۹	۰/۰۶
۰/۴۵	۰/۰۸
۰/۴۲۵	۰/۱
۰/۳۷۵	۰/۱۵
۰/۳۵۴	۰/۱۸
۰/۳۲۵	۰/۲

۵-۲-نتایج اجرای الگوریتم در خطاهای متفاوتِ کانال

حد آستانهٔ موجود در الگوریتم برای خطاهای مختلف کانال، از جدول۲ گرفته شده است و همه شبیه سازی ها بر اساس مقادیر موجود در این جدول انجام شده است. نشان داده می شود الگوریتمِ ارایه شده در خطاهای متفاوت کانال به خوبی قادر به تشخیص مکان و طول دقیق الگو است. برای تعیین طول و مکان دقیق الگو از روی نمودارهای شبیه سازی شده، دو مرحله زیر باید انجام شود:

۱- الگوریتم بتواند به ازای خطاهای مختلف کانال، کاربر را به نموداری شبیه به نمودار شکل ۱۵ – ب برساند.

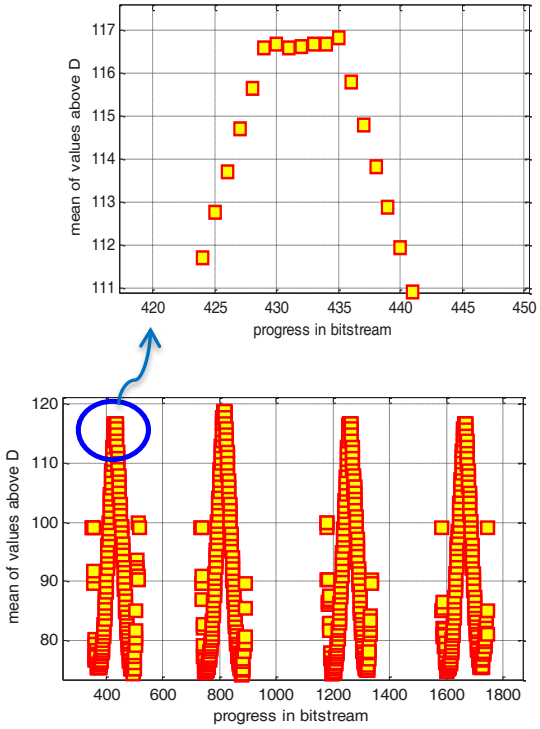
۲- برای تعیین طول و مکان دقیق الگو، تحلیل بر اساس رابطهٔ ۴ و توضیحات مربوط به آن ادامه یافته و خروجی آن به نموداری شبیه به شکل ۱۷- ب برسد و در نهایت طول و مکان دقیق الگو تعیین شود.

بنابراین برای سادگی، در ادامه مقاله برای هر مقدار خطای متفاوت (طبق جدول ۲)، فقط مرحله ۱ انجام می شود، چراکه اگر به ازای خطاهای مختلف کانال در مرحله ۱، نمودارهایی شبیه به شکل ۱۵ به دست آید، ادامه تحلیل در مرحله ۲ به راحتی انجام می شود و مکان و طول دقیق الگو به دست خواهد آمد. اگر الگوریتم پیشنهادی به ازای مقدار خطای خاصی، نتواند مرحله ۱ را به درستی انجام دهد، به این معنی است که برای آن میزان خطا، کارآمد نیست و ادامه کار در مرحلهٔ ۲ نیز عملا ممکن نخواهد بود.

در همهٔ شبیه سازی هایی که در ادامه آورده می شود، حداکثر حجم بیت مورد استفاده ۱۲۵کیلوبایت برای خطای کانال ۰/۲ بوده است. طول الگوی موجود در رشته بیت نیز برای همهٔ شبیه سازی ها ۱۲۰ بیت فرض شده است. مقدار کاهش نرخ در همهٔ شبیه سازی ها برابر ۷ منظور شده است. جهت کم شدن حجم بحث نیز، فقط خطاهای ۰/۰۱ ، ۰/۰۸ ، ۰/۱، ۰/۱۵ ، ۰/۱۸ و ۰/۲ بررسی شده و تنها مرحلهٔ ۱ اجرا شده است. به علاوه بدون از دست دادن کلیت مساله، برای هر مقدار خطا، از فریم متفاوتی نسبت به بقیهٔ خطاها استفاده شده ولی طول الگو برای همه فریم ها ۱۲۰ است.

الف- تعیین طول و مکان الگو برای خطای ۰/۰۱

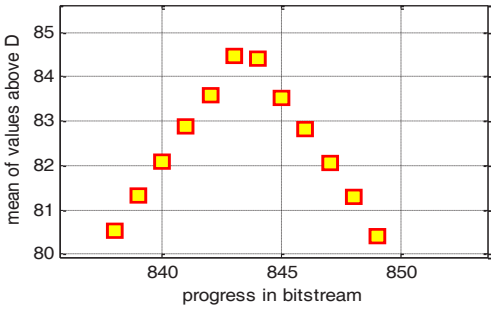
برای خطای ۰/۰۱، الگوریتم پیشنهادی تشخیص الگو به صورت خودکار، طول الگو را برابر ۱۸ بدست می آورد، بنابراین طول تقریبی الگو ۱۲۶= ۷*۱۸خواهد بود. شکل ۱۸ نشان می دهد پس از اجرای مرحلهٔ اول، طول و مکان دقیق الگو طبق مرحلهٔ ۲ قابل شناسایی است. در شکل مذکور، ماکزیمم مقدار موجود در بیضی نشان داده شده، ۱۱۷ است.

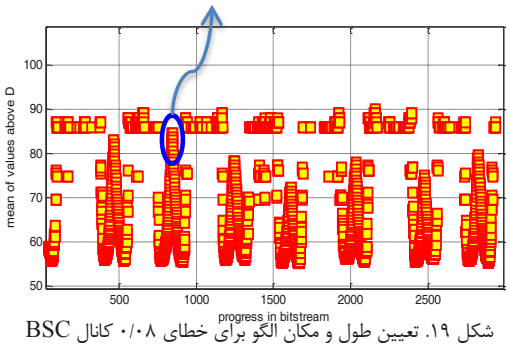


شکل ۱۸. تعیین طول و مکان الگو برای خطای ۰/۰۱ کانال BSC

ب- تعیین طول و مکان الگو برای خطای ۰/۰۸

برای خطای ۰/۰۸، الگوریتم پیشنهادی تشخیص الگو به صورت خودکار، طول تقریبی الگو را ۱۲۶= ۷*۱۸به دست آورده است (۶ واحد خطا). طبق شکل ۱۹، طول و مکان دقیق الگو قابل شناسایی است. ماکزیمم مقدار موجود در بیضی نشان داده شده در این شکل نیز، ۹۳ است و این کاهشِ مقدار نسبت به حالت قبل، به دلیل وجود خطای بیشتری روی رشته بیت است.

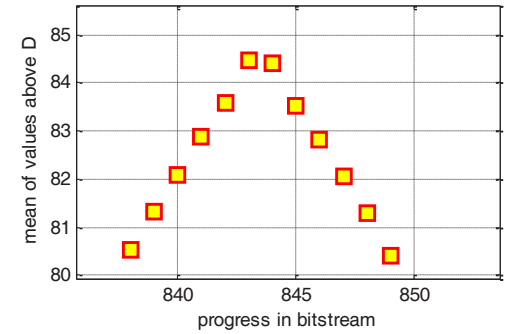


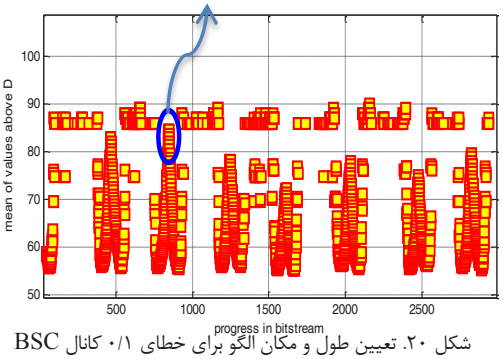


شکل ۱۹. تعیین طول و مکان الگو برای خطای ۰/۰۸ کانال BSC

ج- تعیین طول و مکان الگو برای خطای ۰/۱

برای خطای ۰/۱، الگوریتم پیشنهادی تشخیص الگو به صورت خودکار، طول الگو را برابر ۱۷ بدست آورده، بنابراین طول تقریبی الگو ۱۱۹= ۷*۱۷خواهد بود (۱ واحد خطا). شکل ۲۰ نشان می دهد پس از اجرای مرحلهٔ اول، طول و مکان دقیق الگو بر اساس مرحلهٔ دوم قابل شناسایی است. در شکل مذکور، ماکزیمم مقدار موجود در بیضی نشان داده شده، ۸۴ است. باید دقت کرد که با افزایش میزان خطا، این مقدار ماکزیمم در حال کم شدن است ولی هم چنان بر اساس ویژگی تعریف شده، طول و مکان دقیق الگو قابل تعیین است.

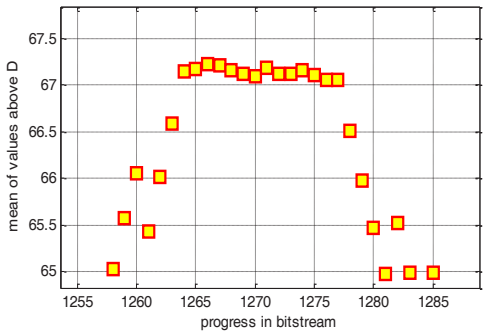


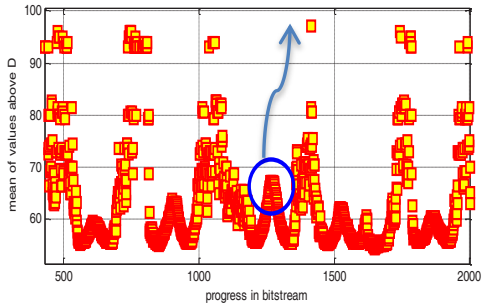


شکل ۲۰. تعیین طول و مکان الگو برای خطای ۰/۱ کانال BSC

د- تعیین طول و مکان الگو برای خطای ۰/۱۵

برای خطای ۰/۱۵، برنامهٔ پیشنهادی تشخیص الگو به صورت خودکار، طول تقریبی الگو را ۱۳۳= ۷*۱۹ بدست می آورد (۱۳ واحد خطا). شکل ۲۱ دارای ۱۴ نقطه با مقدار یکسان و تقریبا ثابت بوده و نشان می دهد پس از اجرای مرحلهٔ اول، طول و مکان دقیق الگو قابل شناسایی است. در شکل مذکور، ماکزیمم مقدارموجود در بیضی نشان داده شده، ۶۷ است.

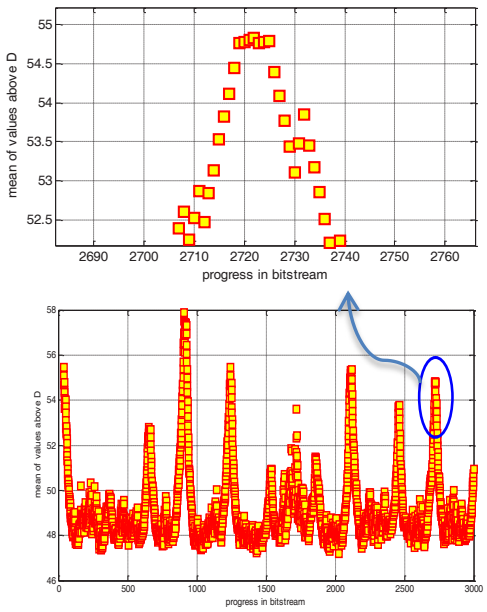




شکل ۲۱. تعیین طول و مکان الگو برای خطای ۰/۱۵ کانال BSC

ه-تعیین طول و مکان الگو برای خطای ۰/۱۸

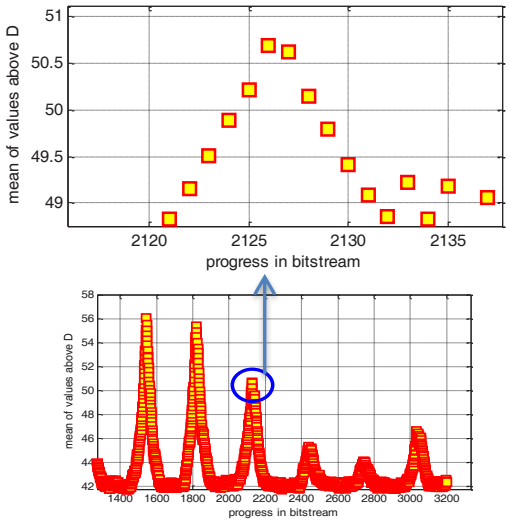
برای خطای ۰/۱۸، برنامهٔ پیشنهادی تشخیص الگو به صورت خودکار، طول الگو را برابر ۱۸ بدست آورده، بنابراین طول تقریبی الگو ۱۲۶= ۷*۱۸خواهد بود (۶ واحد خطا). شکل ۲۲ نشان می دهد پس از اجرای مرحلهٔ اول، طول و مکان دقیق الگو قابل شناسایی است. در شکل مذکور، ماکزیمم مقدار موجود در بیضی نشان داده شده، ۵۵ است.



شکل ۲۲. تعیین طول و مکان الگو برای خطای ۰/۱۸ کانال BSC

و- تعیین طول و مکان الگو برای خطای ۰/۲

برای خطای ۰/۲، الگوریتم پیشنهادی تشخیص الگو به صورت خودکار، طول الگو را برابر ۱۷ بدست آورده، بنابراین طول تقریبی الگو ۱۱۹= ۷*۱۷خواهد بود (۱ واحد خطا). شکل ۲۳ نشان می دهد پس از اجرای مرحلهٔ اول، طول و مکان دقیق الگو قابل شناسایی است. در شکل مذکور، ماکزیمم مقدار موجود در بیضی نشان داده شده، ۵۱ است، در صورتی که ماکزیمم مقدار موجود در بیضی شکل ۱۸ (خطای ۰/۰۱) برابر ۱۱۷ است و این مساله نشان می دهد الگوریتم در خطای شدید کانال نیز به خوبی کارآمد است.



شکل ۲۳. تعیین طول و مکان الگو برای خطای ۰/۲ کانال BSC

۶-نتیجه گیری

در این مقاله به تشخیص‌کور طول و مکان الگو با استفاده از ترکیبی از تحلیل های خودهمبستگی، نمونه برداری از رشته بیت، آنتروپی رنیی و ویژگی"میانگین‌گیری روی‌مقادیربالای‌حد آستانهٔ نمودار هم بستگی متقابل بین رشته بیت اصلی و پنجره لغزشی" پرداخته شد. نتایج بدست آمده نشان داد که الگوریتم ارایه شده برای تحلیل کور الگو های موجود در رشته بیت که ممکن است در مواردی بیش از یک الگو نیز باشند، علاوه بر اینکه در شرایط خطای شدید کانال با تنظیم صحیح حد آستانه به خوبی کار می کند، مدت زمان تحلیل و نیز حجم پردازش را نیز به میزان بسیار زیادی کاهش می دهد. هم چنین نشان داده شد الگوریتم مطرح شده در این مقاله، برخلاف مقالاتی که در بخش مقدمه به صورت مختصر و نیز در شرایط خاص و محدودی به تحلیل فریم پرداخته اند، برای تشخیص الگو های با طول بزرگ و کوچک در فریم های طولانی یا کوتاه کارآمد است.

مرجع‌ها

[1] D. P. Sharma, "Data Scrambler for Ultra-Wideband Communication Systems," 2013.
[2] H. Wu, Z. Sha, Z. Huang, and Y. Zhou, "Frame synchronization for DVB-S2 based on scrambling sequence," in Wireless Personal Multimedia Communications (WPMC), 2014 International Symposium on, 2014, pp. 103-105.
[3] A. Naseri, J. Salem, and M. Hajimohammadi, "Improving the Blind Recovering Algorithm and Synchronization Word in Satellite Communication Systems," Indian J. Sci. Res, vol. 1, pp. 15-19, 2014.
[4] B. Oh, J. Jeong, Y. Jang, and D. Yoon, "Blind Estimation of Frame Synchronization Patterns in Telemetry Signal," in The Twelfth International Conference on Networks, Seville, Spain, 2013, pp. 25-28.
[5] R. Imad and S. Houcke, "Theoretical analysis of a MAP based blind frame synchronizer," Wireless Communications, IEEE Transactions on, vol. 8, pp. 5472-5476, 2009.
[6] R. Imad, S. Houcke, and C. Douillard, "Blind frame synchronization on gaussian channel," sign (x (k), vol. 1, p. 2, 2007.
[7] R. Imad, S. Houcke, and C. Jego, "Blind frame synchronization of product codes based on the adaptation of the parity check matrix," in Communications, 2009. ICC09. IEEE International Conference on, 2009, pp. 1-5.
[8] R. Imad, C. Poulliat, and S. Houcke, "Frame synchronization techniques for non-binary LDPC codes over GF (q)," in Global Telecommunications Conference (GLOBECOM 2010), 2010 IEEE, 2010, pp. 1-6.
[9] R. Imad, G. Sicot, and S. Houcke, "Blind frame synchronization for error correcting codes having a sparse parity check matrix," Communications, IEEE Transactions on, vol. 57, pp. 1574-1577, 2009.
[10] Z. Jing, H. Zhiping, S. Shaojing, and Z. Yimeng, "Blind frame synchronization technique of sparse parity check matrix codes on secure communications," 2014.
[11] T. Xia and H. Wu, "Joint Blind Frame Synchronization and Encoder Identification for Low-Density Parity-Check Codes," 2014.
[12] T. Xia, H.-C. Wu, and S. Y. Chang, "Joint blind frame synchronization and encoder identification for LDPC codes," in Communications (ICC), 2014 IEEE International Conference on, 2014, pp. 5221-5226.
[13] R. Gautier, G. Burel, J. Letessier, and O. Berder, "Blind estimation of scrambler offset using encoder redundancy," in Signals, Systems and Computers, 2002. Conference Record of the Thirty-Sixth Asilomar Conference on, 2002, pp. 626-630.
[14]T. Maszczyk and W. Duch, " Comparison of Shannon, Renyi and Tsallis Entropy Used in Decision Trees", in Proceedings of the 9th international conference on Artificial Intelligence and Soft Computing (ICAISC), 2006 springer ,pp. 643-651.