بسم الله الرحمن الرحيم

دو
١٤٢٥٨٩

دانشگاه شهید بهشتی

دانشکده مهندسی برق و کامپیوتر

بهبود کد توربو با استفاده از کنترل الگوریتم کدگشایی

پایان نامه کارشناسی ارشد مهندسی برق

گرایش مخابرات سیستم

نام دانشجو:

مریم سادات حسینی توشمانلوئی

استاد راهنما:

دکتر حسین شریفی

۱۳۸۹/۷/۲۴

بهمن ۱۳۸۸

سه

۱۴۲۵۸۹

دانشگاه شهید بهشتی

دانشکده مهندسی برق و کامپیوتر

پایان نامه کارشناسی ارشد مهندسی برق- گرایش مخابرات سیستم

تحت عنوان:

بهبود کد توربو با استفاده از کنترل الگوریتم کدگشایی

در تاریخ ۱۳۸۸/۱۱/۱۲ پایان نامه دانشجو، مریم سادات حسینی توشمانلوئی، توسط کمیته تخصصی داوران مورد بررسی و تصویب نهائی قرار گرفت.

۱- استاد راهنما اول: دکتر حسین شریفی امضاء

۴-استاد داور (داخلی): دکتر محمد سجاد صدوق امضاء

۵- استاد داور (خارجی): دکتر محمود احمدیان امضاء

۶- نماینده تحصیلات تکمیلی: دکتر اسلام ناظمی امضاء

کلیه حقوق مادی مترتب بر نتایج مطالعات،

ابتکارات و نوآوریهای ناشی از تحقیق موضوع

این پایان نامه متعلق به دانشگاه شهید بهشتی

می باشد.

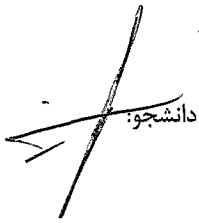
به نام خدا

نام ونام خانوادگی: مریم سادات حسینی توشمانلوئی

عنوان پایان نامه: بهبود کد توربو با استفاده از کنترل الگوریتم کدگشایی

استاد راهنما: دکتر حسین شریفی

اینجانب مریم سادات حسینی توشمانلوئی تهیه کننده پایان نامه کارشناسی ارشد حاضر خود را ملزم به حفظ امانت داری و قدردانی از زحمات سایر محققین و نویسندگان بنا بر قانون Copyright می دانم. بدین وسیله اعلام می نمایم که مسئولیت کلیه مطالب درج شده با اینجانب می باشد و در صورت استفاده از اشکال؛ جداول، و مطالب سایر منابع، بلافاصله مرجع آن ذکر شده و سایر مطالب از کار تحقیقاتی اینجانب استخراج گشته است و امانتداری را به صورت کامل رعایت نموده ام. در صورتی که خلاف این مطلب ثابت شود، مسئولیت کلیه عواقب قانونی با شخص اینجانب می باشد.



مریم سادات حسینی توشمانلوئی: نام ونام خانوادگی دانشجو:

امضاء وتاریخ

فهرست

فصل اول: مقدمه ۱

فصل دوم: توربو کد ۸

۲-۱مقدمه ۹

۲-۲انکدر توربو ۹

۲-۲-۱کدهای سازنده ۹

۲-۲-۲تکنیکهای الحاق ۱۱

۲-۲-۳درهمریزنده ۱۳

۲-۳دیکدر توربو ۱۷

۲-۳-۱نرخ شباهت لگاریتمی ۱۸

۲-۳-۲الگوریتم دیکدینگ MAP ۲۱

۲-۳-۳الگوریتم دیکدینگ Log-MAP ۲۲

۲-۳-۴الگوریتم Max-Log-MAP ۲۵

۲-۳-۵شروط توقف ۲۶

۲-۴جمع بندی ۳۳

فصل سوم: روشهای بررسی همگرایی توربو کد ۳۴

۳-۱مقدمه ۳۵

۳-۲روشهای تصادفی ۳۵

۳-۲-۱تقریب گاوسی ۳۶

۳-۲-۲نمو چگالی ۳۸

۳-۲-۳روش اطلاعات مشترک ۳۹

۳-۲-۴آنالیز Fu ۴۲

۳-۳روشهای قطعی ۴۴

۳-۳-۱مدل دینامیکی ریچاردسون ۴۵

۳-۳-۲سیستمهای دینامیکی ۵۲

۳-۳-۳روشهای مبتنی بر مدل دینامیکی ۵۲
۳-۴شروط توقف مبتنی بر مدل دینامیکی ۵۳
۳-۴-۱روش آگراوال ۵۳
۳-۴-۲روش ZED ۵۵
۳-۵جمع بندی ۵۵
فصل چهارم: بررسی نقاط ثابت ۵۷
۴-۱مقدمه ۵۸
۴-۲بررسی نقاط ثابت با روشهای تصادفی ۵۸
۴-۲-۱روش تقریب گاوسی ۵۸
۴-۲-۲نمایش EXIT chart ۶۰
۴-۳بررسی نقاط ثابت با روشهای قطعی ۶۲
۴-۳-۱بررسی نقاط ثابت الگوریتم Log-MAP ۶۲
۴-۴بررسی نقاط ثابت الگوریتم Max-Log-MAP ۶۶
۴-۵بررسی روش کنترل آشوب ۷۰
۴-۶جمع بندی ۷۳
فصل پنجم: بهبود دیکدینگ توربو ۷۵
۵-۱مقدمه ۷۶
۵-۲ضریب وزندهی وفقی ۷۶
۵-۳روش مبتنی بر هیستوگرام ۷۸
۵-۴جمع بندی ۸۳
فصل ششم: جمع بندی و نتیجه گیری ۸۵
۶-۱نتیجه گیری ۸۶
۶-۲ ادامه کار ۸۶

فهرست اشکال

شکل ۱-۱: مدل شانون برای کانال مخابراتی ۳
شکل ۲-۱:الف) کد کانولوشنال سیستماتیک بازگشتی ب) کد کانولوشنال غیر سیستماتیک غیر بازگشتی ۱۰
شکل۲-۲: خروجی کد کانولوشنال غیر بازگشتی، کلمه کدی با وزن زیاد تولید میکند. ۱۱
شکل۲-۳: خروجی کد کانولوشنال بازگشتی، کلمه کدی با وزن کم تولید میکند. ۱۱
شکل ۲-۴: ساختار کلی توربو کد ۱۲
شکل ۲-۵: توربو کد با ساختار سری ۱۲
شکل ۲-۶: درهمریزنده وزن کد خروجی انکدر دوم را در مقایسه با انکدر اول افزایش میدهد. ۱۳
شکل ۲-۷: مثالی از نقش درهمریزنده ۱۴
شکل ۲-۸: درهمریزنده بلوکی $\mathbf{M \times N}$ ۱۴
شکل ۲-۹: درهمریزنده تصادفی به طول ۸ ..... ۱۶
شکل ۲-۱۰: درهمریزنده شبه تصادفی به طول ۱۶ ..... ۱۷
شکل ۲-۱۱: دیکدر توربو کد ۱۸
شکل ۲-۱۲: نمودار تغییرات LLR بر حسب احتمال بیت ۲۰
شکل ۲-۱۳: توابع تقریب تصحیح ارائه شده در [23] ۲۴
شکل ۲-۱۴: الف)و ب) توابع تصحیح شده در [24] ۲۵
شکل ۳-۱: رشد تابع توزیع اطلاعات فرعی در روش نمو چگالی ۳۹
شکل ۴-۱: بررسی همگرایی با تقریب گاوسی در EbN0 متفاوت ۵۹
شکل ۴-۲: خطای تقریب گاوسی ۶۰
شکل ۴-۳: روش اطلاعات مشترک ۶۱
شکل ۴-۴: خطای اطلاعات مشترک ۶۱
شکل ۴-۵: الگوریتم Log-MAP در EbN0 پایین به نقطه ثابت غیر قطعی همگرا میشود. ۶۲
شکل ۴-۶: الگوریتم Log-MAP همچنان در EbN0 نسبتاً پایین پایدار است. ۶۳
شکل ۴-۷: با افزایش EbN0 نقطه ثابت غیر قطعی پایداری خود را از دست میدهد. ۶۴
شکل ۴-۸: هرچه EbN0 بیشتر شود، نقطه ثابت غیر قطعی ناپایدارتر میشود. ۶۴
شکل ۴-۹: در ناحیه آبشاری دیکدینگ توربو رفتاری شبه پریودیک از خود نشان میدهد. ۶۵
شکل ۴-۱۰: در ناحیه آبشاری دیکدینگ توربو به یک مجموعه جذب کننده آشوبناک همگرا میشود. ۶۶
شکل ۴-۱۱: با افزایش بیشتر EbN0 به نقطه ثابت قطعی همگرا میشود. ۶۶
شکل ۴-۱۲: در الگوریتمMax-Log-MAP دیکدینگ حتی در EbN0 بسیار پایین همگرا نمیشود. ۶۷
شکل ۴-۱۳: در الگوریتمMax-Log-MAP بهبود یافته دیکدینگ توربو حتی در EbN0 بسیار پایین همگرا نمیشود. ۶۸
شکل ۴-۱۴: الگوریتم Max-Log-MAP در EbN0 نسبتاً پایین نیز همگرا نمیشود. ۶۸
شکل ۴-۱۵: در الگوریتم Max-Log-Map در هیچ مقداری از EbN0 رفتار شبه پریودیک مشاهده نمیشود. ۶۸
شکل ۴-۱۶: در الگوریتم Max-Log-Map بهبود یافته مشابه الگوریتم Max-Log-Map معمولی دارد. ۶۹

شکل ۴-۱۷: الگوریتم Max-Log-Map مشابه با الگوریتم Log-Map با افزایش EbN0 همگرا میشود. ۶۹
شکل ۴-۱۸: تأثیرتابع کنترل در همگرایی دیکدینگ توربو در N=400, EbN0=0.2dB ۷۰
شکل ۴-۱۹: تأثیرتابع کنترل در همگرایی دیکدینگ توربو در N=400, EbN0=-1 dB ۷۱
شکل ۴-۲۰: کنترل آشوب عملکرد الگوریتم Log-MAP را بدتر میکند. ۷۲
شکل ۴-۲۱: نتیجه تأثیر کنترل آشوب بر الگوریتم Max-Log-MAP معمولی و Max-Log-MAP بهبود یافته ۷۳
شکل ۵-۱: عملکرد روش ضریب وزندهی در مقایسه با روشهای رایج ۷۸
شکل ۵-۲: هیستوگرام متغیر Q1 به ازای تکرارهای مختلف EbN0=0.7dB ۷۹
شکل ۵-۳ هیستوگرام متغیر Q1 به ازای تکرارهای مختلف در EbN0=2dB ۸۰
شکل ۵-۴: هیستوگرام متغیر Q1 به ازای تکرارهای مختلف EbN0=1.5dB ۸۰
شکل ۵-۵: هیستوگرام متغیر Q1 به ازای تکرارهای مختلف EbN0=-0.5dB ۸۲
شکل ۵-۶: مقایسه متوسط تکرار نسبت به SNR برای دو روش ZED و HB ۸۳
شکل ۵-۷: مقایسه نرخ خطای بیت نسبت به SNR برای دو روش ZED و HB ۸۳

چکیده

پس از این که شانون، تئوری ظرفیت کانال را ارائه کرد، سال‌ها بین سیگنال به نویز مورد نیاز برای انتقال بدون خطا در تئوری و در عمل، 3dB و حتی بیشتر فاصله بود. اختراع توربو کد، این شکاف را تا حد زیادی از بین برد. با توجه به این که دیکدینگ توربو یک الگوریتم تکراری است، روش‌های مختلفی برای بررسی همگرایی توربو کد ارائه شده است. برخی از روش‌ها از خصوصیات آماری ورودی و خروجی دیکدر توربو استفاده می‌کنند. برخی دیگر بر مبنای مدل دینامیکی ریچاردسون بنا هستند که در آن دیکدینگ توربو به صورت یک سیستم دینامیکی مدل شده است که سعی در حل یک دستگاه 2n معادله 2n مجهول داشت و در آن n طول بلوک داده است. جواب این دستگاه معادلات، نقاط ثابت الگوریتم دیکدینگ توربو خواهد بود. هدف این پایان نامه بهبود الگوریتم توربو کد از طریق کنترل الگوریتم کدگشایی آن، بر اساس بررسی همگرایی آن است.

کارهای انجام شده در این پایان‌نامه در دو بخش تنظیم شده است. در بخش اول ابتدا با استفاده از روش‌هایی که رویکرد تصادفی دارند، همگرایی توربو کد را بررسی کرده و شرایط صحت این روش‌ها نشان داده می‌شود. در ادامه با استفاده از مدل دینامیکی ریچاردسون و مفاهیم سیستم‌های غیر خطی، رفتار دو الگوریتم Log-MAP و Max-Log-MAP بررسی می‌شود و همگرایی به نقطه ثابت غیر قطعی، رفتار نوسانی و آشوبناک و همگرایی به نقطه ثابت قطعی نشان داده می‌شود. در ادامه روش‌های کنترل آشوب بررسی شده و نشان داده می‌شود که این روش تأثیری از لحاظ کنترل آشوب ندارد و تنها در الگوریتم Max-Log-MAP به دلیل جبران تقریب‌های اعمال شده در آن مؤثر است.

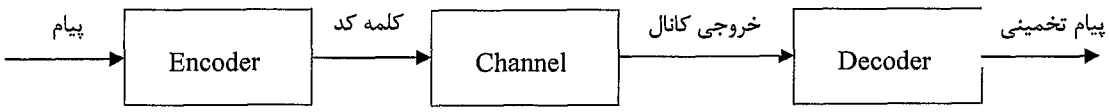
بخش دوم به عنوان کاربردی از نتایج آنالیز انجام شده، شامل دو قسمت خواهد بود. در قسمت اول روش جدیدی برای بهبود الگوریتم دیکدینگ Max-Log-MAP ارائه می‌شود. در این روش از یک تابع برای وزن‌دهی اطلاعات فرعی در خروجی دیکدر استفاده می‌شود که مقدار آن با توجه به مقادیر اطلاعات فرعی تغییر می‌کند. در قسمت دوم، روش توقف جدیدی بر مبنای هیستوگرام متغیرهای مدل دینامیکی الگوریتم رمزگشایی توربو، ارائه می‌شود که قادر به تشخیص همگرایی و عدم همگرایی توربو کد خواهد بود، در حالی که سایر شروط توقف در صورت برآورده نشدن یک آستانه مشخص -که این آستانه عموماً در نزدیکی همگرایی برآورده می‌شود- تا بیشترین میزان تکرار ممکن، ادامه می‌یابند.این روش با بررسی میزان تغییرات هیستوگرام متغیرها، تکرارهای غیر لازم را تشخیص داده و متوسط تعداد تکرار پایین می‌آورد.

کلمات کلیدی : دیکدینگ توربو ، همگرایی، نقاط ثابت، ضریب وزن‌دهی، شروط توقف.

فصل اول

۱ مقدمه

هدف اصلی سیستم مخابراتی، ارسال صحیح داده از یک نقطه به نقطه دیگر است. پارامترهای مؤثر در این سیستم، پهنای باند، نرخ خطای بیت، توان سیگنال، توان نویز، تأخیر، پیچیدگی و هزینه است. مسأله اساسی تعیین کارایی یک سیستم مخابراتی، توسط شانون [1]، در تئوری کدینگ کانال نویزی، بیان شده است. شانون برای آنالیز یک سیستم مخابراتی، آن را به سه بخش اصلی تقسیم کرد: منبع، کانال، گیرنده(شکل ۱-۱). پیام توسط منبع، تولید شده و از طریق مسیر ارتباطی که کانال خوانده می‌شود؛ به گیرنده ارسال می شود. گیرنده با استفاده از خروجی کانال سعی در ساخت پیغام ارسالی دارد. هدف این سیستم ارسال صح یام از منبع به مقصد است. برای کاهش خطا در پیغام ارسالی، قبل از ارسال پیام "کد" می شود. بدین منظور داده‌ی ورودی به بلوک‌هایی تقسیم می‌شود، هر بلوک داده به بلوک دیگری به نام کد نگاشت می‌شود. مجموعه کلمه کدها و نحوه نگاشت بلوک‌های داده به کلمه کد، *"کدگذاری"* نامیده می شود. شانون اثبات کرد که می‌توان کدی با طول بلند ساخت که از طریق آن نرخ خطای بیت به صفر برسد. اما هیچ روشی برای ساخت چنین کدی ارائه نداد. در واقع، هدف اصلی کدکردن کانال ایجاد افزونگی در پیام اولیه است، تا بتوان تأثیر مخرب کانال بر روی پیام را از بین برد. کدینگ کانال می بایست نیازمندیهای زیر را برآورده کند: ۱- دستیابی به نرخ داده بالا R_b امکان‌پذیر باشد؛ یعنی بتوان با حداکثر نرخ ممکن، داده ارسال کرد. ۲- نرخ خطای بیت کم در سیگنال به نویز پایین قابل حصول باشد؛ تا کمترین توان ممکن برای ارسال داده مصرف شود. ۳- با توجه به محدویت اندازه و هزینه، پیچیدگی انکدر و دیکدر تا حد ممکن کم شود. ۴- با توجه به این که در سیستم های ارتباطی دو طرفه، مطلوب، ارسال داده به صورت بلادرنگ است، تأخیر در سیستم مینیمم باشد. ۵- پهنای باند مورد استفاده تا حد ممکن کم شود. تأمین همزمان این نیازمندیها مشکل است و اصولاً برآورده کردن یکی از نیازها به صورت مطلوب، نیازمندی دیگری را به شدت تضعیف می‌کند. پس از ارائه‌ی تئوری شانون، تلاش‌های زیادی برای یافتن کدهایی صورت گرفت که بتوانند به نرخ تعیین شده توسط شانون دست یابند و در عین حال نیازمندیهای فوق را تأمین کنند. کدهای بلوکی و کدهای کانولوشنال از مهم‌ترین کدهای ابداع شده پس از تئوری شانون بودند؛ اما هیچ یک از این کدها به حد شانون دست نیافتند. یعنی، بین سیگنال به نویز محاسبه شده توسط شانون برای انتقال بدون خطا و سیگنال به نویز مورد نیاز به صورت عملی، همواره فاصله بوده است. در سال ۱۹۹۳، Berrou و همکاران، دسته‌ای از کدهای ترکیبی موازی را، تحت نام توربو کد، ارائه دادند [2] که عملکرد آن نزدیک به حد بیان شده توسط شانون بود. آنها ادعا کردند که یک توربو کد باینری با نرخ $\frac{1}{3}$ به نرخ خطای بیت 10^{-5} در سیگنال به نویز ۰/۷ dB دست می‌یابد. این عملکرد که سیگنال به نویز موردنیاز در آن تنها ۰/۷ dB از حد بیان شده توسط شانون بیشتر بود، بسیار بهتر از کدهای موجود در آن زمان بود؛ تا حدی که جامعه کدینگ در ابتدای معرفی آن، نسبت به عملکرد آن شک داشتند تا هنگامی که گروه دیگری از محققان این نتایج را تأیید کردند[3].



شکل ۱-۱: مدل شانون برای کانال مخابراتی

ابداع توربو کد بر مبنای تئوری ریاضی خاصی نبود. توربو کد نتیجه کنار هم قرار دادن چند سیستم بود که قبل از اختراع توربو کد نیز موجود بودند. درواقع، توربو کد، به اذعان مخترعین آن، بیش از مفاهیم ریاضی بر مفاهیم فیزیکی استوار بود[3]. با توجه به این که اساس مخابرات دیجیتال بر مبنای ملاحظات عمیق ریاضی بود، به نظر می‌رسید که کدهای کنترل خطا تنها به دنیای ریاضیات تعلق دارند. اما توربو کد با مدنظرقرار دادن مفاهیم موجود در فیزیک ، مانند استفاده از مفهوم فیدبک در دیکدینگ و نیز تقارن در ساختار موازی کد، اختراع شد. مزیت توربو کد، علاوه بر دستیابی به نرخ خطای بیت نزدیک به حد شانون، قابل قبول بودن پیچیدگی دیکدر آن بود.

انکدر توربو کد، ترکیب موازی دو کد کانولوشنال است که ورودی آنها از طریق یک درهم‌ریزنده به هم متصل است. در واقع اطلاعات دو بار کد می‌شود؛ یکبار با همان ترتیب ارسالی به انکدر و بار دیگر با درهم‌ریختن بیت‌های موجود در یک بلوک. استفاده از درهم‌ریزنده تا حد ممکن تصادفی بودن را در سیستم کدینگ ایجاد می‌کند.

دیکدینگ توربو کد الگوریتمی تکراری بوده و شامل ماژول‌های دیکدینگ است که هر یک از ماژول‌ها از الگوریتم ارائه شده توسط BCJR استفاده می‌کند[4] تا تخمینی از بیت ارسالی به دست آوردند. خروجی یک دیکدر به ورودی دیکدر دیگر ارسال می‌شود تا در هر تکرار تخمین بهتری به دست آید. نام توربو کد به دلیل فیدبک موجود در دیکدر آن است؛ درست همان گونه که در موتورهای توربو قسمتی از توان خروجی، برای بهبود عملکرد سیستم، به ورودی فیدبک می‌شود. در واقع اصطلاح توربو به سیستم دیکدینگ آن اشاره دارد. بحث پایداری که در سیستم‌های فیدبک‌دار اهمیت دارد، با استفاده از اطلاعات فرعی به دست آمد که مانع از ایجاد فیدبک مثبت در سیستم می شود. الگوریتم دیکدینگ BCJR، تحت نام دیکدینگ MAP نیز شناخته شده است. به دلیل پیچیدگی‌های سخت‌افزاری روش MAP، نسخه‌های دیگری از این الگوریتم، روش Log-MAP و Max-Log-MAP در حوزه لگاریتمی ارائه شد[5]. روش Log-MAP همان الگوریتم MAP می‌باشد که در حوزه لگاریتمی پیاده شده است. روش Max-Log-MAP تقریب‌هایی را در روش Log-MAP اعمال کرده که پیاده‌سازی سخت‌افزاری آن را آسان‌تر می‌کند، اما عملکرد آن از لحاظ نرخ خطای بیت بدتر است. از طرفی دیگر الگوریتم‌های MAP و Log-MAP نیاز به تعیین دقیق سیگنال به نویز دارند و در صورتی که سیگنال به نویز دقیق تخمین زده نشود، عملکرد آن در عمل به دقت تئوری نخواهد بود. برای جبران کاهش عملکرد در الگوریتم Max-Log-MAP روش‌های مختلفی ارائه شده است.

همان طور که گفته شد، الگوریتم رمزگشایی توربو روشی تکراری است که در هر تکرار آن، نرخ خطای بیت مربوط به یک سیگنال به نویز معین بهبود می‌یابد. اما بهبود در نرخ خطای بیت با هر تکرار کاهش می‌یابد؛ زیرا با هر تکرار دنباله اطلاعات فرعی بیشتر همبسته می‌شود و این همبستگی بهبود در کارایی را کاهش می‌دهد. اما چه هنگام تکرار بیشتر تاثیر کمی بر بهبود دارد و یا بی‌تاثیر است؟ تکنیک‌های تشخیص و متوقف کردن الگوریتم در این حالت، تحت نام "شروط توقف" شناخته می‌شوند و تلاش زیادی برای به دست آوردن شروط توقف زودهنگام برای اتمام پروسه کدگشایی، بدون تأثیر زیادی بر کارایی کدگشا، انجام شده است.

نتایج اولیه‌ی شبیه‌سازی بر روی توربو کد، رفتاری جالب توجه را در این کد نشان می‌داد. این رفتار که به رفتار پله‌ای توربو کد معروف است، نشان می‌دهد که نرخ خطای بیت توربو کد، در سیگنال به نویزهای کم، با تغییر سیگنال به نویز تغییر چندانی نمی‌کند، با رسیدن به یک آستانه خاص تغییر کمی در سیگنال به نویز بهبود زیادی را در نرخ خطای بیت منجر می‌شود و پس از عبور از آستانه دیگری، نرخ خطای بیت با تغییر سیگنال به نویز، مجدداً ثابت می‌ماند. ناحیه بین دوآستانه مذکور را ناحیه آبشاری می‌گویند. دامنه این ناحیه برای کدهای سازنده متفاوت و به ازای طول بلوک متفاوت، متغیر است، یعنی آستانه شروع این ناحیه و آستانه پایان آن، در شرایط مختلف متفاوت است.

در ابتدای ظهور توربو کد، تئوری ریاضی برای توجیه رفتار و عملکرد آن وجود نداشت و سؤال‌های زیادی درباره چرایی رفتار آن مطرح بود. چه عاملی باعث عملکرد بسیار خوب آن است؟ چه رابطه‌ای بین دیکدینگ توربو و دیکدینگ ML وجود دارد؟ چه عاملی بهبود شدید رفتار توربو کد، به ازای تغییر کوچکی در سیگنال به نویز در ناحیه آبشاری را باعث می‌شود؟ چرا قبل و بعد از ناحیه آبشاری تغییر سیگنال به نویز تأثیر چندانی بر رفتار آن ندارد؟ چرا توربو کد در برخی نواحی حتی پس از هزاران تکرار به پاسخ درستی منجر نمی‌شود؟ آیا همواره پاسخ درستی برای همگرا شدن دیکدر موجود است؟ چگونه عرض ناحیه آبشاری برای کدهای سازنده‌ی متفاوت تغییر می‌کند؟ افزایش طول بلوک چگونه رفتار توربو کد را تحت تأثیر قرار می‌دهد؟

محققان به دنبال یافتن پاسخ این سؤال‌ها روش‌های مختلفی را برای آنالیز رفتار توربو کد ارائه دادند. الگوریتم توربو به عنوان یکی از زیرشاخه‌های الگوریتم کلی‌تری برای انتشار اطلاعات در گراف‌ها به نام " انتشار گمان[1]" مطرح شد[6]. اما الگوریتم "انتشار گمان" تنها هنگامی صحیح بود که گراف به دست آمده یک درخت باشد، اما گراف توربو کد دارای حلقه بود و این تئوری پاسخ دقیقی برای آنالیز توربو کد به خصوص برای طول بلوک پایین به دست نمی‌داد. در[9],[8],[7] برای بررسی همگرایی توربو کد و یافتن آستانه‌ای که در آن الگوریتم به نقطه‌ی ثابتی همگرا شود، دنباله اطلاعات ورودی و خروجی تنها با یک پارامتر مدل شد. در واقع در این مقالات با فرض گاؤسی بودن دنباله اطلاعات ورودی و خروجی، بدون

[1] Belief Propagation

اثباتی برای صحت این فرض، ورودی و خروجی را با سیگنال به نویز آن، اطلاعات مشترک این دنباله‌ها با مشاهدات کانال و میانگین و واریانس مدل کردند. این آنالیز نتایج جالبی برای بررسی رفتار توربو کد و به دست آوردن آستانه همگرایی برای کدهای سازنده مختلف ارائه داد؛ اما این نتایج تنها در طول بلوک بالا معتبر بودند. Fu شرایط معتبر بودن فرض گاؤسی و اثبات صحت آن در این شرایط را به دست آورد و نشان داد که مدل کردن ورودی و خروجی تنها با یک پارامتر کافی نیست و برای به دست آوردن آنالیز دقیق‌تر، می بایست از یک مدل دو ورودی-دو خروجی استفاده کرد[10]. در این پایان‌نامه این روش‌ها تحت عنوان روش‌های تصادفی بحث شده‌اند.

اما در این روش‌ها، تنها درباره‌ی وجود یا عدم همگرایی به پاسخ درست بحث شده است و صحبتی درباره‌ی چرایی همگرا شدن به میان نیامده است. در واقع این آنالیزها توجیهی برای رفتار غیر خطی الگوریتم دیکدینگ نیافته‌اند و همگرایی به پاسخ نادرست را نیز توجیه نکرده‌اند. ریچاردسون در مقاله‌ی معروف خود، دیکدینگ توربو را به صورت یک سیستم دینامیکی مدل کرد[11]. این روش توربو دیکدینگ را به صورت یک الگوریتم تکراری بیان کرد که سعی در حل یک دستگاه 2n معادله 2n مجهول دارد، که در آن n طول داده است. اگر الگوریتم همگرا شود، این نقطه همگرایی همان جواب معادل دستگاه 2n معادله 2n مجهول خواهد بود و برعکس جواب این دستگاه معادلات، همان نقطه ثابت توربو کد خواهد بود. نقاط ثابت، نقاطی هستند که در آن خروجی توربو دیکدر و ورودی آن یکسان خواهد بود، در واقع هنگامی که الگوریتم تکراری به یک نقطه ثابت برسد، می‌گوییم همگرا شده است. ریچاردسون با استفاده از مدل خود، وجود نقطه ثابت را اثبات کرده و شرایط لازم برای پایداری آن را بیان کرد. همچنین نزدیکی دیکدینگ توربو به روش دیکدینگ ML را نشان داد. مدل ریچاردسون زمینه تحقیقات جدیدی را در بررسی رفتار توربو کد باز کرد. آگراوال و وردی با استفاده از این مدل خط سیر الگوریتم توربو را به دست آوردند[12]. با استفاده از این خط سیر، سیگنال به نویز به سه قسمت تقسیم می‌شود و دیکدینگ توربو در هریک از این سه ناحیه رفتار متفاوتی از خود نشان می‌دهد. آنها نشان دادند که دیکدر تکراری توربو دو نوع نقطه ثابت دارد؛ یکی در سیگنال به نویزهای پایین که به آن نقطه‌ی ثابت غیر قطعی[1] گفته می‌شود و دیگری در سیگنال به نویزهای بالا که به آن نقطه‌ی ثابت قطعی[2] می‌گویند. دیکدینگ حاصل از نقطه ثابت غیر قطعی خطای زیادی به همراه دارد ولی دیکدینگ نتیجه شده از نقطه‌ی ثابت قطعی به پاسخ درستی منجر می‌شود. نقطه ثابت غیر قطعی پیش از ناحیه آبشاری، در سیگنال به نویزهای بسیار کم پایدار است و نقطه‌ی ثابت قطعی پس از ناحیه آبشاری پایدار می‌شود. در واقع نقطه ثابت قطعی در تمامی نواحی سیگنال به نویز موجود است، اما الگوریتم این نقطه را تا رسیدن به یک آستانه مشخص سیگنال به نویز نمی‌یابد. پس از رسیدن به این آستانه، نقطه ثابت غیر قطعی ناپدید می‌شود و الگوریتم به نقطه‌ی ثابت قطعی همگرا می‌شود. در ادامه این بررسی کوکارف و دیگران مفهومی به نام آنتروپی متوسط ارائه دادند و از

[1] Indecisive Fixed Point
[2] Unequivocal Fixed Point

طریق آن رفتار توربو کد را بررسی کردند[14],[13]. آنها نشان دادند که بسیاری از پدیده‌هایی که در حوزه سیستم‌های دینامیکی غیر خطی شناخته شده‌اند؛ مانند رفتار نوسانی، آشوب و آشوب گذرا در دیکدینگ تکراری توربو رخ می‌دهند. با استفاده از این آنالیز روشی را برای کنترل آشوب در الگوریتم توربو ارائه دادند و ادعا کردند که این روش به طور مؤثری نرخ خطای بیت را بهبود داده و همگرایی را سرعت می‌بخشد.

کارهای انجام شده در این پایان‌نامه در دو بخش تنظیم شده است. در بخش اول ابتدا با استفاده از روش‌هایی که رویکرد تصادفی دارند، همگرایی توربو کد را بررسی کرده و شرایط صحت این روش‌ها نشان داده می‌شود. در ادامه با استفاده از مدل دینامیکی ریچاردسون و مفاهیم سیستم‌های غیر خطی، رفتار دو الگوریتم Log-MAP و Max-Log-MAP بررسی می‌شود و همگرایی به نقطه ثابت غیر قطعی، رفتار نوسانی و آشوبناک و همگرایی به نقطه ثابت قطعی نشان داده می‌شود. در ادامه روش‌های کنترل آشوب بررسی شده و اثبات می‌شود که این روش تأثیری از لحاظ کنترل آشوب ندارد و تنها در الگوریتم Max-Log-MAP به دلیل جبران تقریب‌های اعمال شده در آن مؤثر است.

بخش دوم به عنوان کاربردی از نتایج آنالیز انجام شده، شامل دو قسمت خواهد بود. در قسمت اول روش جدیدی برای بهبود الگوریتم دیکدینگ Max-Log-MAP ارائه می‌شود. در این روش از یک تابع برای وزن‌دهی اطلاعات فرعی در خروجی دیکدر استفاده می‌شود که مقدار آن با توجه به مقادیر اطلاعات فرعی تغییر می‌کند. در قسمت دوم، روش توقف جدیدی بر مبنای هیستوگرام متغیرهای مدل دینامیکی الگوریتم رمزگشایی توربو، ارائه می‌شود که قادر به تشخیص همگرایی و عدم همگرایی توربو کد خواهد بود، در حالی که سایر شروط توقف در صورت برآورده نشدن یک آستانه مشخص -که این آستانه عموماً در نزدیکی همگرایی برآورده می‌شود- تا بیشترین میزان تکرار ممکن، ادامه می‌یابند.این روش با بررسی میزان تغییرات هیستوگرام متغیرها، تکرارهای غیر لازم را تشخیص داده و متوسط تعداد تکرار را بدون کاهش کارایی رمزگشا، به طور قابل توجهی پایین می‌آورد.

ساختار این پایان‌نامه در ادامه به صورت زیر خواهد بود: فصل دوم به ساختار توربو کد اختصاص دارد. در ابتدا انکدر توربو معرفی شده و بخش‌های مختلف آن بیان می‌شود. سپس دیکدر توربو توضیح داده شده و روش‌های مختلف دیکدینگ ذکر خواهد شد. در انتها شروط توقف رایج برای متوقف کردن الگوریتم توربو آورده شده است.

در فصل سوم روش‌های بررسی همگرایی با تقسیم‌بندی تصادفی و قطعی معرفی خواهند شد. بخش اول شامل شرح روش‌های تصادفی [10],[9],[8],[7] خواهد بود. بخش دوم به بیان روش‌های قطعی می‌پردازد. در روش‌های قطعی در ابتدا مدل ریچاردسون با بیان ساده‌ای ارائه می‌شود و از طریق آن نزدیکی توربو دیکدینگ به ماکسیمم ML نشان داده می‌شود. پس از بررسی کوتاهی بر آنالیز آگراول [12]، مباحث بیان شده توسط کوکارف [14],[13] مطرح خواهد شد.

سپس روش کنترلی کوکارف برای کنترل آشوب بیان می‌شود. با توجه به این که شروط توقف مبتنی بر مدل ریچاردسون نیاز به بیان این مدل داشتند، این شروط به جای فصل دوم در انتهای این فصل بیان می‌شود.

در فصل چهارم به بررسی نقاط ثابت الگوریتم توربو پرداخته می‌شود. نخست با استفاده از روش‌های تصادفی همگرایی توربو کد بررسی شده و عدم دقت این روش‌ها در طول بلوک محدود نشان داده می‌شود. در ادامه نقاط ثابت قطعی و غیر قطعی و نیز رفتار آشوبناک دیکدینگ توربو تحقیق می‌شود. نشان داده می‌شود که در EbN0 پایین رفتار الگوریتم Log-MAPو Max-Log-MAP متفاوت است و الگوریتم Max-Log-MAP بر خلاف الگوریتم Log-MAP همگرا نخواهد شد. در ادامه روش کنترلی کوکارف برای کنترل آشوب بررسی شده و نشان داده می‌شود که این روش تنها در الگوریتم Max-Log-MAP مؤثر است. ضمناً اثبات می‌شود که این روش تأثیری از لحاظ کنترل آشوب ندارد و تنها به جبران تقریب‌های اعمال شده در روش Max-Log-MAP کمک می‌کند.

فصل پنجم، بهبود دیکدینگ توربو، به عنوان کاربردی از نتایج آنالیز فصل چهارم، شامل دو بخش خواهد بود. در بخش اول روشی ارائه می‌شود که در ابتدا با هدف کنترل آشوب دیکدینگ توربو انجام شده بود؛ اما با توجه به آنالیزهای صورت گرفته در فصل قبل و اثبات یکسان بودن کنترل آشوب و ضریب وزن‌دهی وفقی الگوریتم دیکدینگ Max-Log-MAP این روش به عنوان روش جدیدی برای بهبود الگوریتم دیکدینگ Max-Log-MAP ارائه شد. در این روش از یک تابع برای ضریب وزن‌دهی وفقی در خروجی دیکدر استفاده می‌شود که مقدار آن با توجه به مقادیر اطلاعات فرعی تغییر می‌کند. با توجه به این که توربو کد در نواحی مختلف سیگنال به نویز خصوصیات متفاوتی دارد، سیگنال به نویز به چند ناحیه تقسیم شده و پارامترهای تابع در هر ناحیه تعیین می‌شود. در ادامه این روش و روش‌های رایج، مقایسه شده و بهبود این روش نشان داده خواهد شد. در بخش دوم از این فصل، روش توقف جدیدی بر مبنای هیستوگرام متغیرهای مدل دینامیکی الگوریتم رمزگشایی توربو، یعنی اطلاعات فرعی خروجی دو دیکدر، ارائه می‌شود که قادر به تشخیص همگرایی و عدم همگرایی توربو کد خواهد بود.(در حالی که سایر شروط توقف در صورت برآورده نشدن یک آستانه مشخص -که این آستانه عموماً در نزدیکی همگرایی برآورده می‌شود- تا بیشترین میزان تکرار ممکن، ادامه می‌یابند.) این روش با مرزبندی هیستوگرام این متغیرها، شرط توقف زودهنگامی بدون کاهش کارایی در عملکرد رمزگشا، اعمال می‌کند. همچنین در مقادیر سیگنال به نویز کم با بررسی میزان تغییرات هیستوگرام متغیرها، تکرارهای غیر لازم را تشخیص داده و متوسط تعداد تکرار را بدون کاهش کارایی رمزگشا، به طور قابل توجهی پایین می‌آورد.

فصل دوم

۲ توربو کد

۲-۱ مقدمه

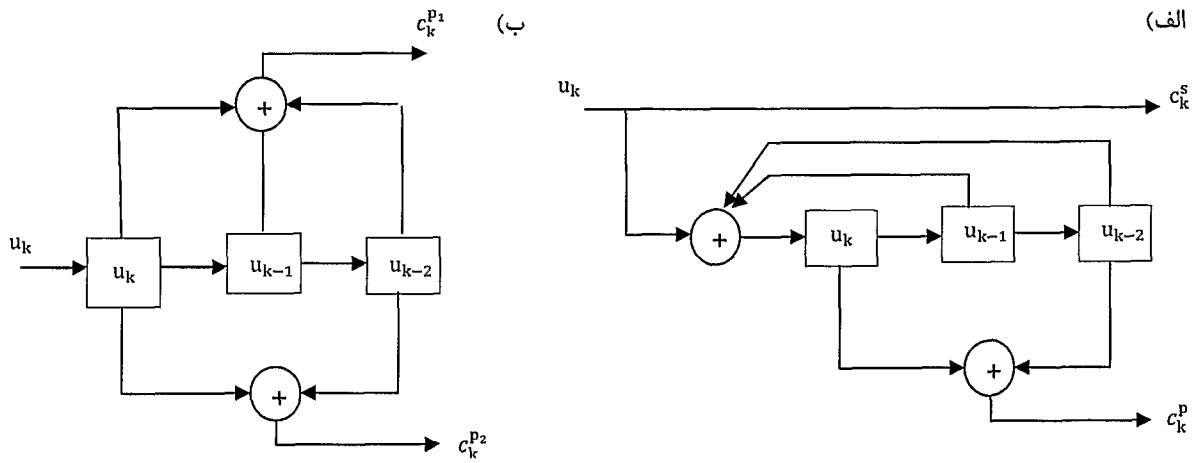
اختراع توربو کد، بی‌شک، یکی از بزرگ‌ترین ابداعات در زمینه تئوری کدینگ است. پیش از آن، طراحی کدی برای دستیابی به حد ارائه شده توسط شانون، برای سالها مورد تحقیق بود. توربو کد نه تنها به حد بیان شده توسط شانون دست یافت؛ بلکه پیچیدگی انکدر و دیکدر آن قابل قبول بود و به سرعت توانست در سیستم‌های عملی راه یابد. در این فصل توربو کد در دو بخش معرفی خواهد شد. در بخش اول انکدر توربو کد توضیح داده شده و اجزای تشکیل دهنده‌ی آن معرفی خواهد شد. بخش دوم مربوط به دیکدر توربو خواهد بود و در آن روش‌های مختلف دیکدینگ و شروط توقف بیان می‌شود.

۲-۲ انکدر توربو

توربو کد در حالت کلی از الحاق دو یا چند کد کانال که از طریق درهم‌ریزنده به هم ارتباط دارند، به دست می‌آید. انکدر توربو شامل دوبخش است: کدهای سازنده و درهم‌ریزنده. کدهای سازنده به روش‌های مختلفی می‌توانند به هم متصل شوند. در ادامه هر یک از این اجزای انکدر توربو توضیح داده شده و تکنیک‌های الحاق مختلف بیان می‌شود.

۲-۲-۱ کدهای سازنده

کدهای سازنده می‌توانند هر نوعی از کدینگ کانال اعم از بلوکی، کانولوشنال و .. باشند، اما کدهای کانولوشنال بیشتر مورد توجه قرارگرفته‌اند، زیرا الگوریتم دیکدینگ آن‌ها در سیستم توربو در دسترس است. در این نوشته تمامی انکدرهای توربو از کدهای کانولوشنال استفاده می‌کنند. کدهای کانولوشنال دسته‌ای از کدهای کانال هستند که در آن‌ها افزونگی تولید شده، علاوه بر داده ورودی به حالت سیستم وابسته است. در این کدها، داده ورودی با ارسال به یک شیفت رجیستر خطی کد می‌شود. یک کد کانولوشنال با m شیفت رجیستر، یک کد کانولوشنال m حافظه‌ای گفته می‌شود. به طور معادل به آن کد 2^m-حالته نیز گفته می‌شود؛ زیرا داده‌ی باینری به 2^m حالت متفاوت می‌تواند در آن ذخیره شود. کدهای کانولوشنال می‌توانند سیستماتیک یا غیر سیستماتیک و بازگشتی یا غیر بازگشتی باشند. در کدهای سیستماتیک یک نسخه از ورودی به خروجی ارسال می‌شود، در حالی که کدهای غیر سیستماتیک چنین خصوصیتی ندارند. در کدهای بازگشتی خروجی به ورودی فیدبک شده است، اما کدهای غیر بازگشتی این چنین نیستند. همچنین در این کدها، حالت بعدی سیستم تنها به داده ورودی بستگی ندارد، بلکه با فیدبک کردن خروجی، حالت فعلی سیستم نیز در آن مؤثر است. شکل ۳ یک کدکانولوشنال غیر سیستماتیک غیر بازگشتی ۸-حالته و کد سیستماتیک بازگشتی معادل آن را نشان می‌دهد.



شکل ۲-۱:الف) کد کانولوشنال سیستماتیک بازگشتی ب) کد کانولوشنال غیر سیستماتیک غیر بازگشتی

همان طور که در شکل مشاهده می‌شود؛ داده خروجی تنها بر مبنای داده ورودی و حالت سیستم، یعنی حالت شیفت رجیسترها، به دست می‌آید. هنگامی که ورودی جدید به این سیستم اعمال می‌شود؛ حالت شیفت رجیسترها نیز به‌روز می‌شود. انکدر با حالت اولیه صفر شروع می‌کند؛ اما با ارسال داده‌ی ورودی حالت سیستم تغییر می‌کند. در انتهای بلوک داده M بیت "انتهایی" به داده اضافه می‌شود تا انکدر رابه حالت صفر بازگرداند. برای یک کد کانولوشنال با m حافظه نیاز به m+1 بیت پایانی می‌باشد.

ساختمان سخت‌افزاری کدهای کانولوشنال را می‌توان از طریق چند جمله‌ای مولد نشان داد. این چند جمله‌ای وجود یا عدم وجود ارتباط سخت افزاری یک شیفت رجیستر را با جمع کننده نشان می‌دهد. در بردار چند جمله‌ای مولد، "۱" وجود و "۰" عدم وجود ارتباط را نشان می‌دهد. برای مثال بردار مولد در شکل ۳ عبارت است از $g_1 = [111]$ و $g_2 = [101]$ که در آن اندیس ۱ و ۲ نشان دهنده خروجی مربوط است.

کدهای تشکیل دهنده توربو کد عمدتاً کدهای بازگشتی سیستماتیک (RSC)[1] می‌باشند. گرچه دلیل استفاده از انکدرهای RSC، خاصیت سیستماتیک آن‌ها بوده است، اما خاصیت بازگشتی بودن عملکرد توربو کد را تحت تأثیر قرار داده است. طوری که تمام انکدرهای توربو کد با استفاده از کدهای بازگشتی ساخته شده است. شکل زیر خروجی حاصل از یک انکدر کانولوشنال سیستماتیک بازگشتی را در مقابل انکدر سیستماتیک غیر بازگشتی به ازای داده یکسان را نشان می‌دهد:

[1] Recursive Systematic Convolutional Code