|  |  |
| --- | --- |
|  |  |

**دانشگاه تهـران**

**پردیس دانشکده های فنی**

**دانشکده مهندسی برق و کامپیوتر**

**پروژه درس حفاظت دیجیتال**

**تخمین مسیر حرکت امپدانس در رله دیستانس**

**نگارش:  
آراز باقرزاده   
(810197435 )**

**استاد درس:**

**دکتر صنایع­پسند**

**دی 1397**

فهرست مطالب

* [مقدمه](#_Toc503225486)

[**معرفی منحنی های Spiral** 3](#_Toc503225487)

[**برازش دو بعدی با الگوریتم حداقل مربعات**  7](#_Toc503225488)

* [**اعمال الگوریتم به رله دیستانس** 9](#_Toc503225501)

1. [**بهبود عملکرد در تشخیص خطا** 9](#_Toc503225502)
2. [**تشخیص Evolving fault** 12](#_Toc503225503)
3. [**adaptive کردن با fault detector** 14](#_Toc503225504)

* [**نتیجه گیری** 15](#_Toc503225505)

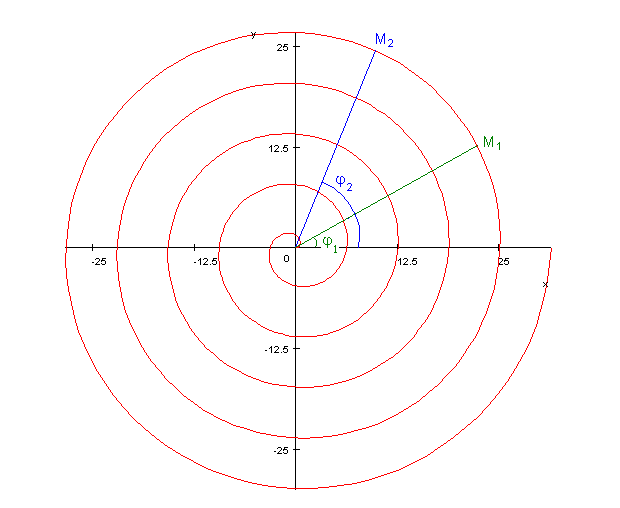
|  |
| --- |
| **معرفی منحنی های Spiral** |

مارپیچ در ریاضیات، یک خم است که از یک نقطه مرکزی سرچشمه می‌گیرد و به تدریج از نقطه مرکزی دورتر می‌شود. مارپیچ ها انواع دو بعدی و سه بعدی دارند که به علت ماهیت پروژه به معرفی مارپیچ های دو بعدی می پردازیم. مهم ترین مارپیچ های دو بعدی به شرح زیر می باشند:

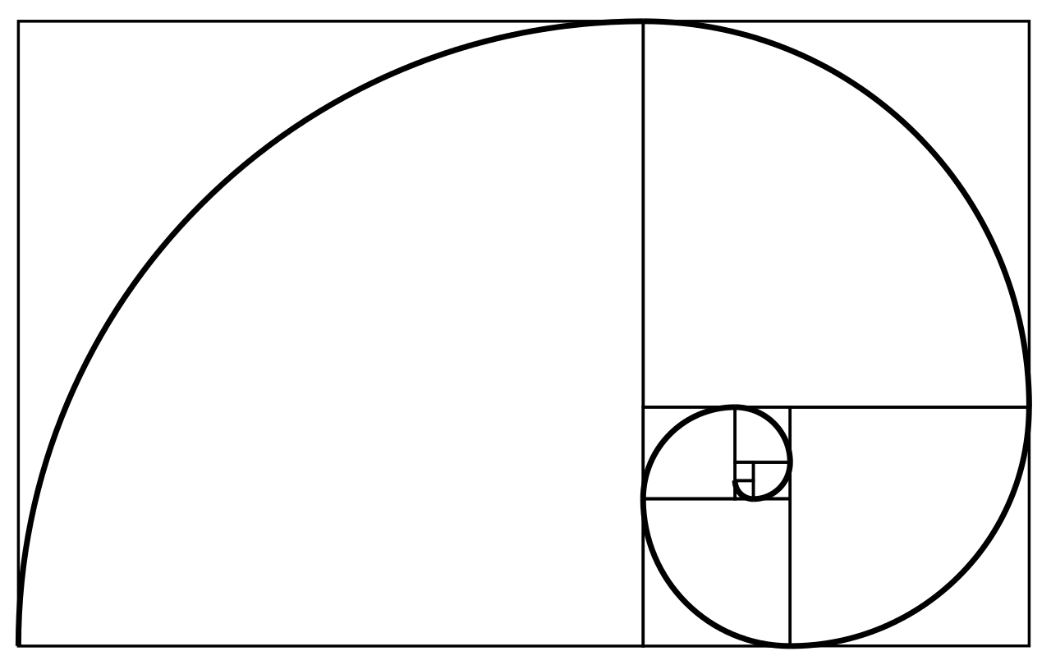
* مارپیچ ارشمیدس
* مارپیچ طلایی
* مارپیچ لگاریتمی
* مارپیچ فیبوناچی
* مارپیچ سینوسی
* مارپیچ هذلولوی
* مارپیچ اویلر
* مارپیچ فرما

در ادامه به معرفی پارپیچ های فوق می پردازیم تا مناسب ترین آن را برای فیت کردن به منحنی امپدانس بست آوریم:

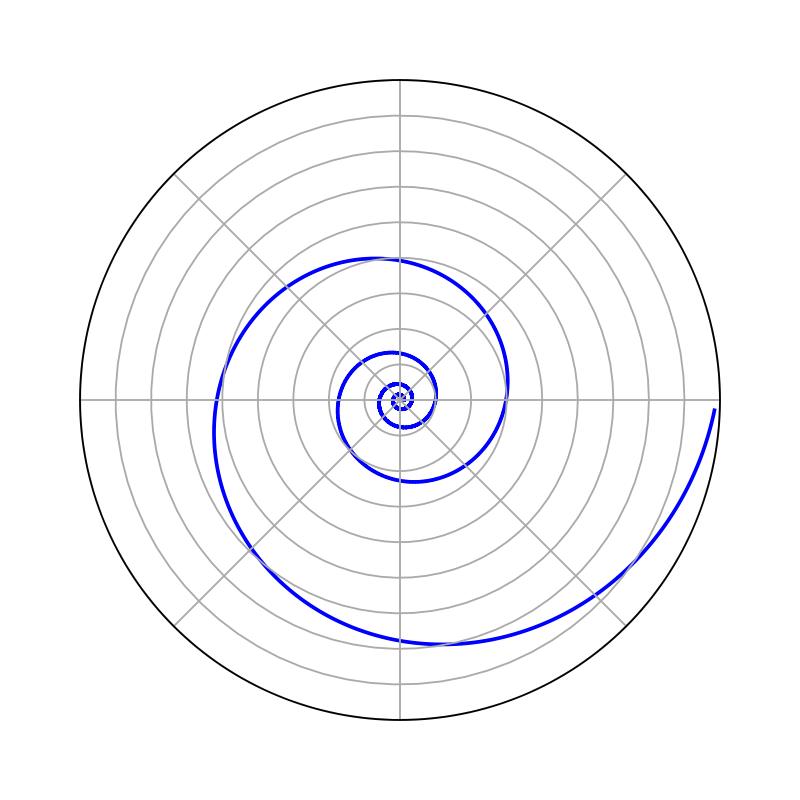
**مارپیچ ارشمیدس**



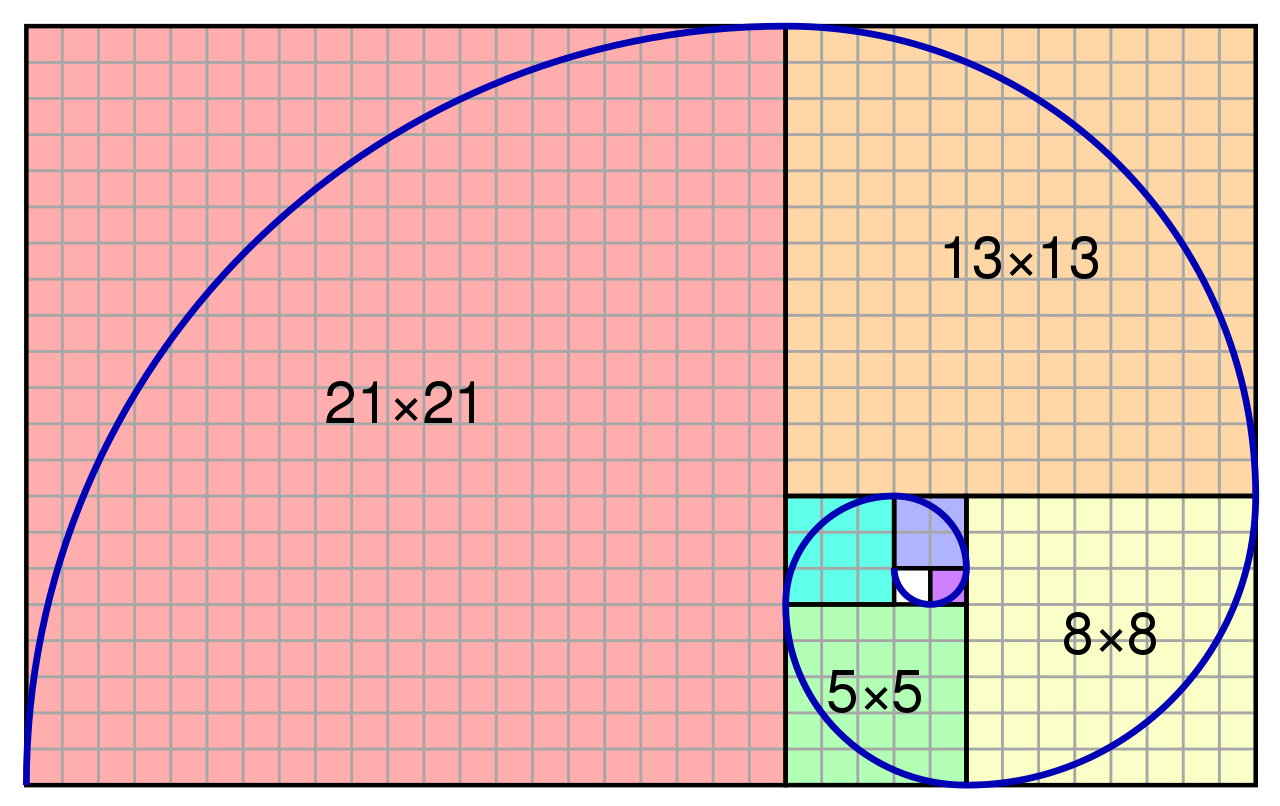
**مارپیچ طلایی**



**مارپیچ لگاریتمی**



**مارپیچ فیبوناتچی**



با مشاهده منحنی های فوق و مقایسه شهودی با مسیر امپدانس در هنگام خطا، مارپیچ لگاریتمی برای فیت کردن مناسب تر به نظر می رسد. پس روابط آن را استخراج می کنیم.

روابط قطبی آن به صورت زیر است:

حال اگر روابط را به صورت پارامتریک بنویسیم، داریم:

حال می خواهیم مارپیچ را به نقطه (c,d) در صفحه انتقال دهیم، داریم:

|  |
| --- |
| **برازش دو بعدی با الگوریتم حداقل مربعات** |



تخمین سمپل ها با منحنی اسپیرال

مشاهده می شود که نقاط تخمین زده شده بسیار نزدیک به محل همگرایی منحنی اصلی قرار می گیرند و به این ترتیب می توان گفت که با استفاده از داده های که هنوز همگرا نشده اند، ولی با یک منحنی مشخص به نقطه همگرایی خود در حرکت اند، محلی که قرار است منحنی در آن همگرا شود را پیشبینی کنیم.

|  |
| --- |
| **الگوریتم مورد استفاده** |

برای اینکه بتوانیم منحنی trajectory امپدانس را بطور تخمینی مدل کنیم از رابطه پارامتری زیر استفاده می کنیم.

این دو رابطه ی پارامتری یک منحنی Spiral را در صفحه x و y به ما نشان می دهد. با دادن این ضابطه ها به الگوریم حداقل مربعات معرفی شده می توان به ازای هر سمپل، مقادیر پارامتر های a،b،c،d،e و f را بدست آورد. اما در واقع پارامترهایی که برای عملکرد رله دیستانس بدرد می خورد، پارامتر های a و b است. به این دلیل که نشان می دهند اگر الگوریتم با همان فرم در سمپل های بعدی حرکت کند، نقطه ای که در بی نهایت به آن همگرا می شود، نقطه (a,b) می باشد.

با توجه به اینکه ضابطه های بیان شده برحسب پارامتر ها غیر خطی بوده و استخراج آن ها از طریق بسط تیلور مشکل خواهد بود، از الگوریتم حداقل مربعات خطای غیر خطی استفاده می کنیم که با استفاده از روش نیوتون و چند بار تکرار مقادیر پارامتر ها را بدست بیاوریم. برای انجام اینکار نیاز به optimization toolbox متلب داریم که تکرار های نیوتون رافسون را برای ما انجام دهد و در مینیموم های محلی گیر نکند. بنا بر این علاوه بر ضابطه های داده شده، یک سری داده اولیه برای شروع محاسبات به optimization toolbox بدهیم تا بتواند به دقت عمل بهینه سازی را برای یافتن بهترین fit پیدا کند.

|  |
| --- |
| **اعمال الگوریتم به رله دیستانس** |

با اعمال الگوریتم می خواهیم زمان عملکرد رله را مقایسه کنیم. برای بررسی یک case مرزی را انتخاب می کنیم و جدول زیر را بدست می آوریم:



|  |  |  |
| --- | --- | --- |
| زمان عملکرد | سمپل خارج زون |  |
| بیشتر | 10 | فوریه عادی |
| کمتر | 6 | الگوریتم معرفی شده |
| بیشترین | 21 | معادلات دیفرانسیل |

همچنین منطق قطع رله را برای زون اول قرار گرفتن 20 سمپل داخل آن در نظر گرفتیم.






|  |
| --- |
| **بهبود عملکرد در تشخیص خطا** |

برای بررسی عملکرد الگوریتم در رله دیستانس، ابتدا با شبیه سازی در نرم افزار PSCAD منحنی های مربوط به trajectory یک سری خطای نمونه را استخراج می کنیم تا روی آن عملکرد الگوریتم معرفی شده را بررسی نماییم. به عنوان مثال در شکل زیر با استفاده از الگوریتم 30 پنجره ای trajectory مربوط به بلوک CA یک رله دیستانس که ناشی از خطای سه فاز به زمین است را بررسی می کنیم:



مقایسه همگرایی الگوریتم معرفی شده و trajectory واقعی

با توجه به شکل بالا مشاهده می شود که از نمونه که منحنی به شکل spiral در می آید به بعد، تخمین دقیق تر و بهتری از نقطه نهایی عملکرد رله دیستانس خواهیم داشت، که می تواند در خطا هایی که در مرز زون اتفاق می افتند به تصمیم گیری عملکرد رله دیستانس کمک کند.



مقایسه عملکرد برای مقاومت



مقایسه عملکرد برای راکتانس

|  |
| --- |
| **تشخیص Evolving Fault** |

حال که متوجه شدیم الگوریتم می تواند در خطاهای که ساختار ثابت دارند بخوبی عمل کند و همگرایی مطلوبی داشته باشد، شایان توجه است که بدانیم برعکس این امر هم می تواند اطلاعات مفیدی را در اختیار بگذارد. به بیانی دیگر وقتی خروجی x و r الگوریتم از حالت همگرا خارج شده و دچار نوسان قابل توجه شود، یعنی تغییری در ساختار شبکه ایجاد شده است.

|  |
| --- |
| **Adaptive کردن با fault detector** |

به علت حجم محاسبات زیاد پیدا کردن نقطه همگرایی، با استفاده از فانکشن fault detector کاری می کنیم که محاسبات بیان شده از زمانی شروع شود که fault detector تشخیص خطا بدهد تا بی خودی بار محاسبتی در هنگامی خطا وجود ندارد، اشغال نشود.

|  |
| --- |
| **نتیجه گیری** |

در این پروژه سعی شد با در نظر گرفتن یک شکل spiral شبیه به trajectory امپدانس حین وقوع خطا و استفاده از الگوریتم حداقل مربعات خطا یک trajectory بهبود یافته بر اساس نقاط همگرایی منحنی اسپیرال بدست آوریم. در کل می توان مزایا و معایب روش فوق را در سرعت دادن به عملکرد رله دیستانس، در خطاهای که نزدیک مرز زون قرار می گیرند، و امکان تشخیص evolving بودن یا نبودن خطا دانست. اما اگر ایده معرفی شده در این پروژه را به خوبی ببینیم، می توان دید که می توان یک ضابطه های دیگری برای پدیده های دیگری که در سیستم قدرت رخ می دهد مانند power swing هم تعریف کرد که می تواند روش خوبی برای بدست آوردن شدت آن و بهبود عملکرد فانکشن power swing blocking باشد.

عیب این روش محاسبات نسبتا سنگین به ازای هر سمپل است که در ادامه پروژه سعی خواهد شد تا با خطی کردن ضابطه برحسب پارامتر ها محاسبات حداقل مربعات خطا را کاهش دهیم تا این عیب رفع شود.