

به نام خدا

# علوم اعصاب یادگیری، حافظه، شناخت

دکتر کربلایی آقا جان



دانشگاه صنعتی شریف

علیرضا جهانی فر - رضا نائب حبیب  
۴۰۱۱۰۲۶۹۴-۴۰۱۱۰۱۵۶۸

گزارش تمرین دوم کامپیوتری  
محاسبه توابع ارتباطات مغزی و مقایسه معیارها

## فهرست مطالب

۳	.....	مقدمه	۱
۵	.....	(dPLI) Index Lag Phase Directed	۲
۸	.....	Correlation Power-to-Power	۳
۱۲	.....	(PAC) Coupling Phase-Amplitude	۴
۱۴	.....	MLV	۱۰۴
۲۳	.....	مقایسه معیار های ارتباطی	۵
۲۳	.....	تأثیر تاخیر در نوسانات ناشی از فرستنده	۱۰۵
۲۳	.....	dPLI	۱۰۱۰۵
۲۳	.....	PPC	۲۰۱۰۵
۲۳	.....	PAC	۳۰۱۰۵
۲۳	.....	اثر منبع مشترک	۲۰۵
۲۴	.....	dPLI	۱۰۲۰۵
۲۴	.....	PPC	۲۰۲۰۵
۲۴	.....	PAC	۳۰۲۰۵
۲۴	.....	اثر نویز	۳۰۵
۲۴	.....	dPLI	۱۰۳۰۵
۲۴	.....	PPC	۲۰۳۰۵
۲۴	.....	PAC	۳۰۳۰۵



# علوم اعصاب یادگیری، حافظه، شناخت

علیرضا جهانی فر - رضا نائب حبیب

گزارش تمرین دوم کامپیوتری

محاسبه توابع ارتباطات مغزی و مقایسه معیارها

## ۱ مقدمه

ارتباطات مغزی به عنوان یکی از مفاهیم اساسی در مطالعات علوم اعصاب، نمایانگر تعاملات بین بخش‌های مختلف مغز است. این ارتباطات به سه دسته اصلی تقسیم می‌شوند: ارتباطات ساختاری، ارتباطات کارکردی و ارتباطات مؤثر.

ارتباطات ساختاری مبتنی بر وجود اتصال آناتومیک مستقیم بین نواحی مغزی است و نشان‌دهنده وجود مسیرهای نورونی فیزیکی بین بخش‌های مغز است. از سوی دیگر، ارتباطات کارکردی، یک بیان ریاضی از هماهنگی عملکرد میان نواحی مختلف مغز است. این نوع ارتباطها صرفاً به هماهنگی در فعالیت مغزی اشاره دارند و لزوماً نشان‌دهنده ارتباط علی یا مستقیم نیستند. ارتباطات مؤثر، با افزودن جهت به ارتباطات کارکردی، امکان بررسی روابط شبه علی بین نواحی مختلف مغز را فراهم می‌کنند و بیانگر این هستند که چگونه رفتار الکتروفیزیولوژیکی یک ناحیه تابعی از گذشته ناحیه دیگری است.

در این تمرین، از سیگنال‌های EEG برای بررسی این ارتباطات استفاده می‌شود. این سیگنال‌ها به دلیل دقت زمانی بالا ابزار مناسبی برای رهگیری ارتباطات کارکردی و مؤثر محسوب می‌شوند، هرچند دقت پایین مکانی آن‌ها چالش‌هایی را در تفسیر نتایج ایجاد می‌کند.

## انواع ارتباطات مغزی و معیارهای اندازه‌گیری

### ارتباطات ساختاری

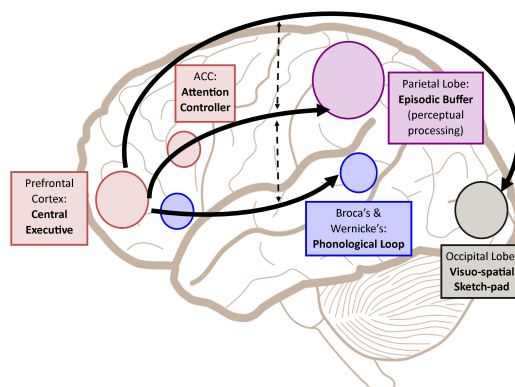
ارتباطات ساختاری مبتنی بر اتصالات نورونی مستقیم است. برای مثال، ارتباط مستقیم بین قسمت‌های مختلفی از مغز که در حافظه بلندمدت نقش دارد، یک ارتباط ساختاری است. در این تمرین، ارتباطات ساختاری مستقیماً اندازه‌گیری نمی‌شوند، زیرا معیارهای انتخاب‌شده تمرکز بیشتری بر ارتباطات کارکردی و مؤثر دارند.

### ارتباطات کارکردی

ارتباطات کارکردی نشان‌دهنده هماهنگی فعالیت‌های مغزی بدون در نظر گرفتن وجود اتصالات مستقیم هستند. به عنوان مثال، فعالیت هماهنگ بین قسمت‌های مختلفی از قسمت پس سری که وظیفه پردازش بینایی را دارند هنگام پردازش بصری و تصمیم‌گیری، یک ارتباط کارکردی محسوب می‌شود. در این تمرین، معیار زیر برای بررسی ارتباطات کارکردی استفاده می‌شود:

#### • Power-to-Power Correlation:

این معیار ارتباط بین قدرت سیگنال (Amplitude) در دو ناحیه مغزی را بررسی می‌کند و به نوعی بیانگر ارتباط دامنه به دامنه است.



### ارتباطات مؤثر

ارتباطات مؤثر نشان‌دهنده جهت اطلاعات بین نواحی مختلف مغزی هستند و به بررسی روابط شبه علی می‌پردازند. به عنوان مثال، اثر ناحیه بروکا بر ناحیه ورنیکه در پردازش زبان، نشان‌دهنده یک ارتباط مؤثر است. معیار زیر برای بررسی ارتباطات مؤثر در این تمرین استفاده می‌شود:

### • Directed Phase Lag Index (dPLI):

این معیار با تمرکز بر اختلاف فاز بین سیگنال‌های دو ناحیه، جهت جریان اطلاعات را تعیین می‌کند و ارتباط فاز-به-فاز را توصیف می‌کند.

### ارتباطات فاز دامنه

نوع خاصی از ارتباطات مغزی به ارتباط بین فاز یک سیگنال در یک ناحیه و دامنه سیگنال در ناحیه دیگر اشاره دارد. این ارتباطات برای بررسی تعاملات پیچیده‌تر مغزی، مانند ارتباط بین امواج آهسته و سریع در خواب، کاربرد دارند. در این تمرین از معیار زیر استفاده می‌شود:

### • Phase-Amplitude Coupling (PAC):

این معیار نشان‌دهنده ارتباط بین فاز سیگنال در یک ناحیه و دامنه سیگنال در ناحیه دیگر است.

هر یک از این معیارها را پیاده سازی، سپس اثر نویز و منبع مشترک که در اندازه‌گیری سیگنال‌های عصبی در نظر نگرفتن آن‌ها تقریباً اجتناب ناپذیر را بررسی می‌کنیم و تاثیر آن را بر هر یک از معیارها ارائه می‌دهیم.

## ۲ Directed Phase Lag Index (dPLI)

یکی از معیارهای مهم در مطالعات ارتباطات مغزی که مبتنی بر هماهنگی عملکرد فاز است، خانواده Phase Lag Index (PLI) می‌باشد. این معیار، با استفاده از توزیع اختلاف فاز سیگنال‌ها حول صفر، میزان هماهنگی فاز را ارزیابی کرده و اثر منابع مشترک را کاهش می‌دهد. رابطه اصلی PLI به صورت زیر تعریف می‌شود:

$$PLI(x, y) = |E[\text{sign}(\Delta\phi_{x,y})]|$$

در این رابطه،  $\Delta\phi_{x,y}$  اختلاف فاز بین سیگنال‌های  $x$  و  $y$  است. این معیار زمانی که توزیع اختلاف فازها حول صفر متقارن باشد، مقدار کوچکی می‌گیرد و زمانی که ارتباط قوی بین سیگنال‌ها وجود داشته باشد، مقدار بالاتری خواهد داشت.

در ادامه، برای بهبود تحلیل جهت‌داری ارتباطات، معیار Directed Phase Lag Index (dPLI) معرفی شد. این معیار علاوه بر ارزیابی هماهنگی فاز، به بررسی جهت‌داری ارتباط بین سیگنال‌ها می‌پردازد. رابطه این معیار به صورت زیر است:

$$dPLI = E[\text{Heaviside}(\Delta\phi_{x,y})]$$

که در آن، تابع Heaviside به صورت زیر تعریف می‌شود:

$$\text{Heaviside}(x) = \begin{cases} 1 & x > 0 \\ 0 & x < 0 \\ 0.5 & x = 0 \end{cases}$$

در dPLI، اگر مقدار برابر با 0.5 باشد، نشان‌دهنده عدم ارتباط یا نوسان هم‌فاز است. مقادیر بیشتر یا کمتر از 0.5 نشان‌دهنده ارتباط جهت‌دار قوی‌تر است. به عنوان مثال، مقدار 0.7 بدین معناست که سیگنال  $y$  در ۷۰ درصد مواقع سیگنال  $x$  را دنبال می‌کند.

ارتباط بین این دو معیار به گونه‌ای است که از dPLI می‌توان PLI را استخراج کرد، اما عکس این امکان‌پذیر نیست. به این ترتیب، dPLI ابزار قدرتمندتری برای تحلیل جهت‌داری و ارتباطات موثر در داده‌های EEG محسوب می‌شود. حال با توجه به تعریف، تابعی برای بدست آوردن میزان هماهنگی فازی مینویسیم.

```

1 import numpy as np
2 from scipy.signal import hilbert
3
4 def heaviside(x):
5     return np.heaviside(x, 0.5)
6 def calculateDPLI(signal1, signal2):
7     analytic_signal1 = hilbert(signal1)
8     analytic_signal2 = hilbert(signal2)
9     phase_1 = np.angle(analytic_signal1)
10    phase_2 = np.angle(analytic_signal2)
11    phase_diff = np.unwrap(phase_1 - phase_2)
12    expected_value = np.abs(np.mean(heaviside(phase_diff)))
13    return expected_value

```

در این کد، ابتدا سیگنال‌های ورودی با استفاده از تبدیل هیلبرت به حوزه تحلیلی منتقل شده و فازهای آن‌ها استخراج می‌شوند. سپس اختلاف فاز محاسبه و از طریق تابع Heaviside، میانگین مقادیر اختلاف فازها برای ارزیابی dPLI محاسبه می‌شود.

حال برای بررسی صحت کارکرد همچنین مشاهده اثر منبع مشترک و نویز سه سیگنال را تعریف می‌کنیم و اثر این دو را بررسی می‌کنیم:

```

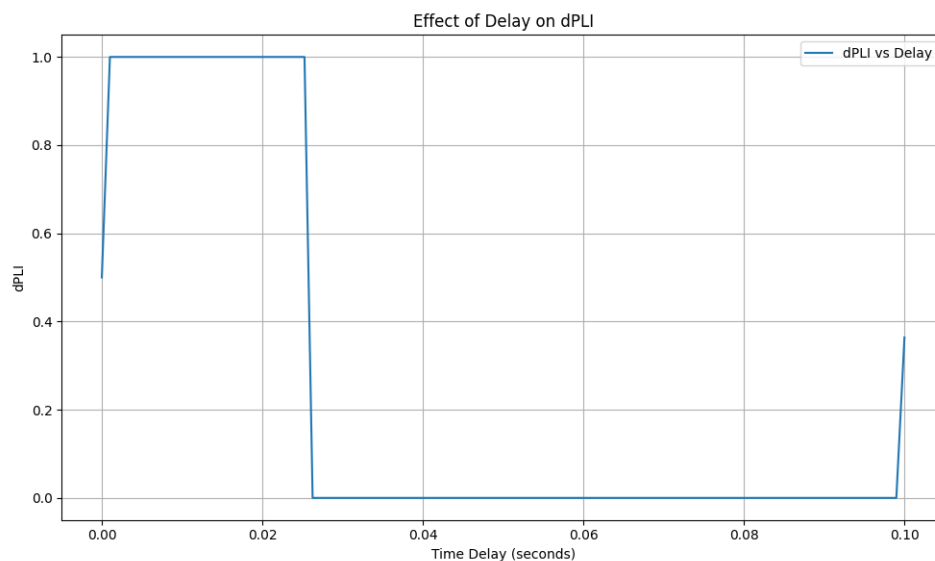
1 fs = 10
2 t = np.linspace(0, 1, 1000)
3 t0 = 1 / 500
4 x_signal = np.sin(2 * np.pi * fs * t)
5 y_signal = np.sin(2 * np.pi * fs * (t - t0))
6 cm_signal = np.sin(2 * np.pi * fs * t)

```

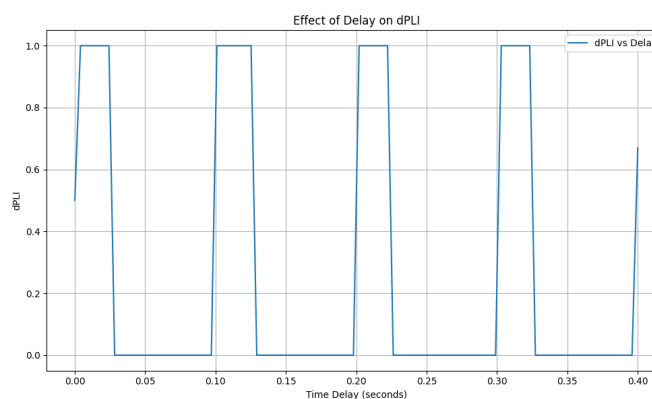
```

1 def effectdelay_DPLI(x_signal, t, fs):
2     dpli_values = []
3     for t0 in np.linspace(0, 0.4, 100):
4         y_delayed = np.sin(2 * np.pi * fs * (t - t0))
5         dpli_values.append(calculateDPLI(x_signal, y_delayed))
6     return dpli_values

```



حال خروجی را توجیه میکنیم، در هنگامی که تاخیر مقدار کوچکی دارد مشاهده میشود که سیگنال  $y$  مطابق انتظار سیگنال  $x$  را دنبال می کند، با افزایش این مقدار تا عدد 0.025 ترم  $\frac{\pi}{2}$  ظاهر میشود که باعث میشود، ترتیب تاخیر جابجا شود تا وقتی که به 0.1 برسیم که دوباره به سیگنال اولیه میرسیم.



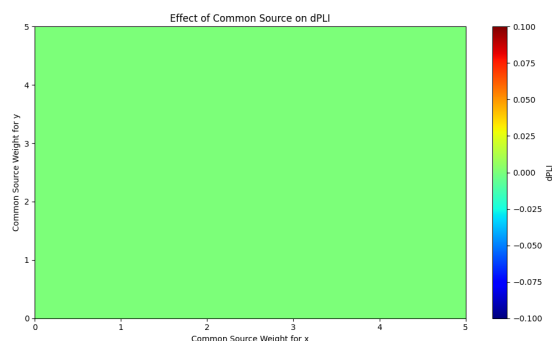
حال اثر منبع مشترک را بررسی میکنیم:

```

1 def effectCM_DPLI(x_signal, y_signal, cm_signal):
2     dpli_matrix = []
3     for a in np.linspace(0, 5, 50):
4         row = []
5         for b in np.linspace(0, 5, 50):
6             xc = x_signal + a * cm_signal
7             yc = y_signal + b * cm_signal
8             row.append(calculatedDPLI(yc, xc))
9         dpli_matrix.append(row)
10    return dpli_matrix

```

و خروجی را نمایش می دهیم:



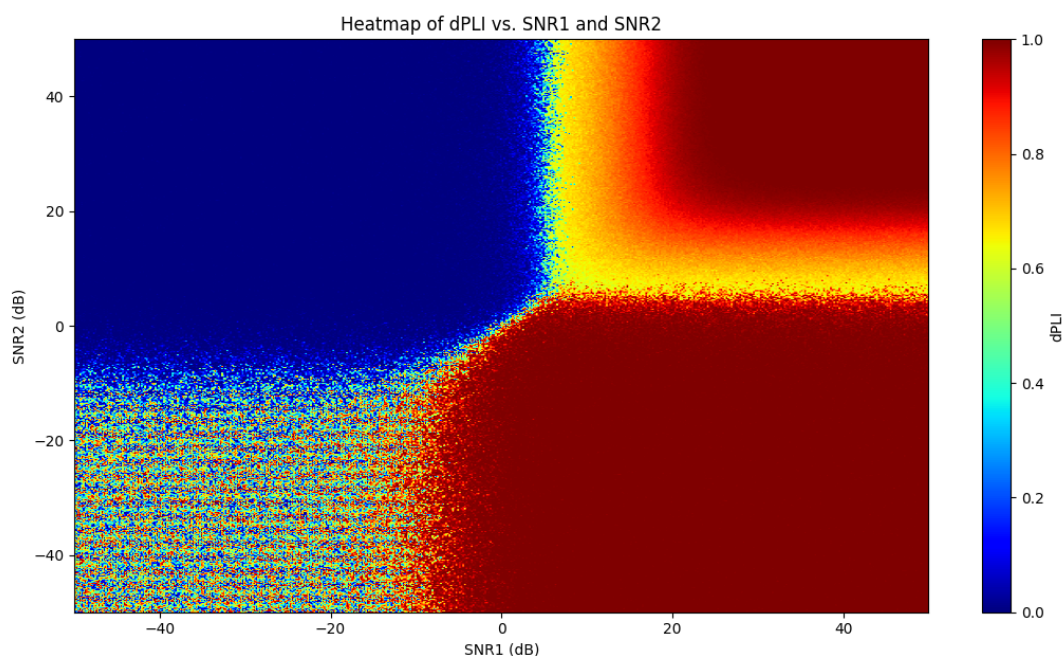
یکدست بودن هیت مپ نشان دهنده عدم تاثیر منبع مشترک بر روی این معیار است ، چرا که تاثیر مشترک دامنه باعث اثر گذاری در فاز نخواهد شد. حال این بار اثر نویز را بررسی میکنیم ، در سیگنال های مغزی ، در نظر نگرفتن نویز باعث اشتباه در آنالیز و تحلیل این سیگنال ها می شود.

```

1 def add_noise(signal, snr_db):
2     signal_power = np.mean(signal ** 2)
3     snr_linear = 10 ** (snr_db / 10)
4     noise_power = signal_power / snr_linear
5     noise = np.sqrt(noise_power) * np.random.randn(len(signal))
6     return signal + noise

1 snr_values = np.linspace(-50, 50, 100)
2 dpli_matrix = np.zeros((len(snr_values), len(snr_values)))
3 for i, snr1 in enumerate(snr_values):
4     for j, snr2 in enumerate(snr_values):
5         noisy_signal1 = add_noise(signal1, snr1)
6         noisy_signal2 = add_noise(signal2, snr2)
7         dpli_matrix[i, j] = calculate_dpli(noisy_signal1, noisy_signal2)

```



دو قسمت بالا سمت راست و پایین چپ به ترتیب نواحی ای هستند که اثر نویز بر سیگنال اصلی غالب نیست و نویز بر سیگنال اصلی غالب است ، مشاهده میشود در ناحیه ای که قدرت سیگنال بیشتر است طبق انتظار سیگنال دنبال کننده و دنبال شونده به خوبی قابل تمیز هستند ، اما هر چقدر که به نواحی پایینی و چپ میرویم مشاهده میکنیم که معیار رفتار تصادفی خواهد داشت. در سایر نواحی بستگی به قدرت سیگنال دنبال کننده/دنبال شونده به مقدار نویز سیگنال متقابل مقدار نزدیک °



یا ۱ خواهد گرفت.

### ۳ Correlation Power-to-Power

یکی از معیارهای مورد استفاده برای تحلیل ارتباطات کارکردی در سیگنال‌های عصبی، معیار Power-to-Power Correlation است. این معیار به بررسی رابطه بین دامنه سیگنال‌های ثبت شده در دو ناحیه مختلف مغزی می‌پردازد. دامنه یک سیگنال، معمولاً به عنوان توان یا قدرت سیگنال تعریف می‌شود که نشان‌دهنده شدت فعالیت الکتروفیزیولوژیکی در ناحیه مورد نظر است. به طور خاص میزان شباهت و هماهنگی قدرت نوسانات سیگنال‌های ثبت شده از دو ناحیه مغزی را در یک بازه زمانی مشخص تحلیل می‌کند.

برای محاسبه این معیار، ابتدا دامنه سیگنال‌ها از طریق روش‌های مختلفی استخراج می‌شود. سپس ضرایب همبستگی بین این دامنه‌ها به صورت زیر محاسبه می‌گردد:

$$\rho = \frac{\text{Cov}(X, Y)}{\sigma_X \sigma_Y}$$

که در آن:

- $X$  و  $Y$  نشان‌دهنده دامنه سیگنال‌ها در دو ناحیه مغزی هستند.
- $\text{Cov}(X, Y)$  کوواریانس بین دامنه دو سیگنال است.
- $\sigma_X$  و  $\sigma_Y$  انحراف معیار دامنه سیگنال‌ها هستند.

#### کاربردها

این معیار به طور گسترده در مطالعات علوم اعصاب کاربرد دارد. برخی از کاربردهای مهم آن عبارتند از:

۱. تحلیل هماهنگی کارکردی: بررسی همزمانی تغییرات دامنه فعالیت‌های مغزی بین دو ناحیه برای ارزیابی ارتباطات کارکردی.
۲. تشخیص اختلالات مغزی: شناسایی تغییرات غیرطبیعی در ارتباطات مغزی مرتبط با اختلالات عصبی مانند صرع یا اسکیزوفرنی.
۳. تحلیل حالت‌های هوشیاری: بررسی تغییرات همبستگی قدرت سیگنال‌ها در حالت‌های مختلف هوشیاری مانند خواب، بیداری یا بیهوشی.

#### مزایا و محدودیت‌ها

مزایا:

- سادگی محاسبات.
- امکان تحلیل داده‌های طولانی مدت با دقت زمانی بالا.

محدودیت‌ها:

- حساسیت به نویز و منابع مشترک.
- عدم توانایی در ارائه جهت‌گیری ارتباطات.

این معیار یک ابزار کارآمد برای مطالعه ارتباطات کارکردی در سیگنال‌های این‌EEG و MEG محسوب می‌شود، اما در ترکیب با سایر روش‌ها مانند Phase-based Measures می‌تواند اطلاعات کامل‌تری از فعالیت‌های مغزی ارائه دهد. به مانند معیار قبل، پس از ایجاد سیگنال‌ها به بررسی اثر تأخیر زمانی می‌پردازیم.

#### Listing : \ correlation

```

1 def calculate_correlation(signal1, signal2):
2     mean1 = np.mean(signal1)
3     mean2 = np.mean(signal2)
4
5     Cov = np.sum((signal1 - mean1) * (signal2 - mean2))
6     sigma1sigma2 = np.sqrt(np.sum((signal1 - mean1)**2) * np.sum((signal2 - mean2)**2))
7     return Cov / sigma1sigma2

```

```

1 def delay_signal(signal, delay, sampling_rate):
2     num_samples = int(delay * sampling_rate)
3     delayed_signal = np.concatenate((np.zeros(num_samples), signal))[len(signal)]
4     return delayed_signal

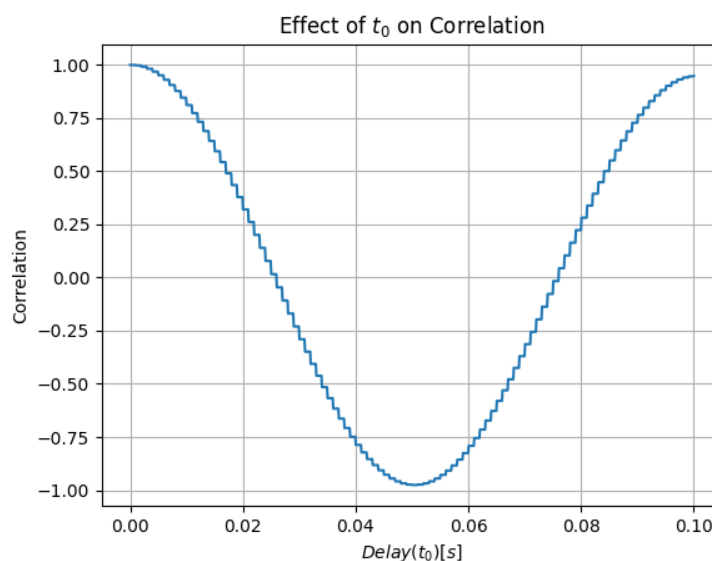
```



```

1 def effectt0_correlation(x_signal, y_signal, fs):
2     corre = []
3     delays = np.linspace(0, 0.1, 1000)
4     for t0 in delays:
5         y_signal_delay = delay_signal(y_signal, t0, fs)
6         corre.append(calculate_correlation(x_signal, y_signal_delay))
7     return delays, corre

```



برای بررسی اثر منبع مشترک :

```

1 fs = 1000
2 T = 1
3 N = T * fs
4 mean = 0
5 std_dev = 1
6 noise_signal1 = np.random.normal(mean, std_dev, N)
7 noise_signal2 = np.random.normal(mean, std_dev, N)

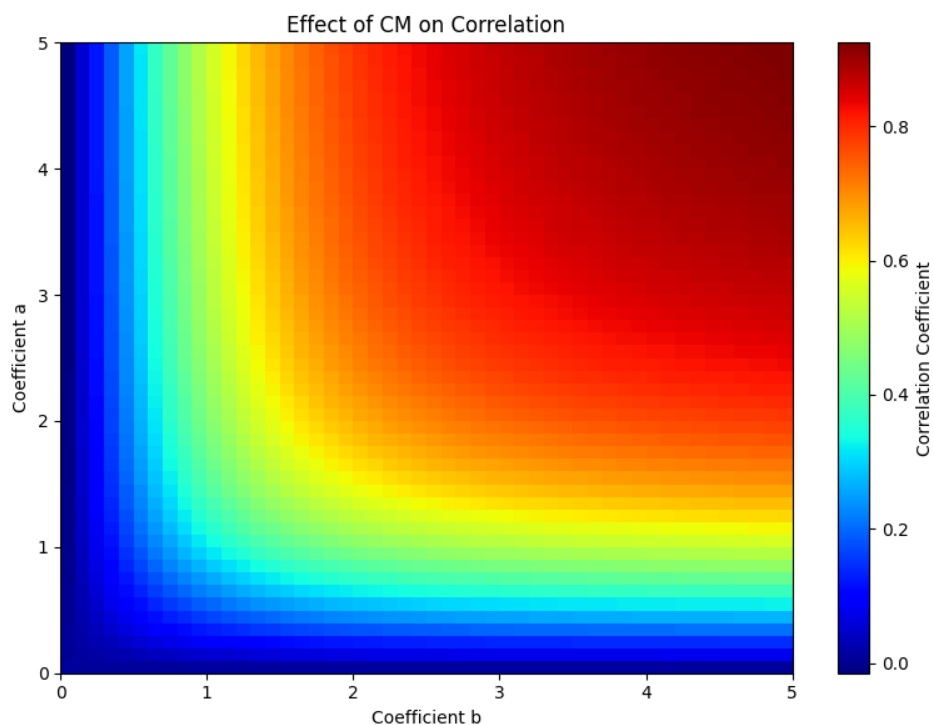
```

```

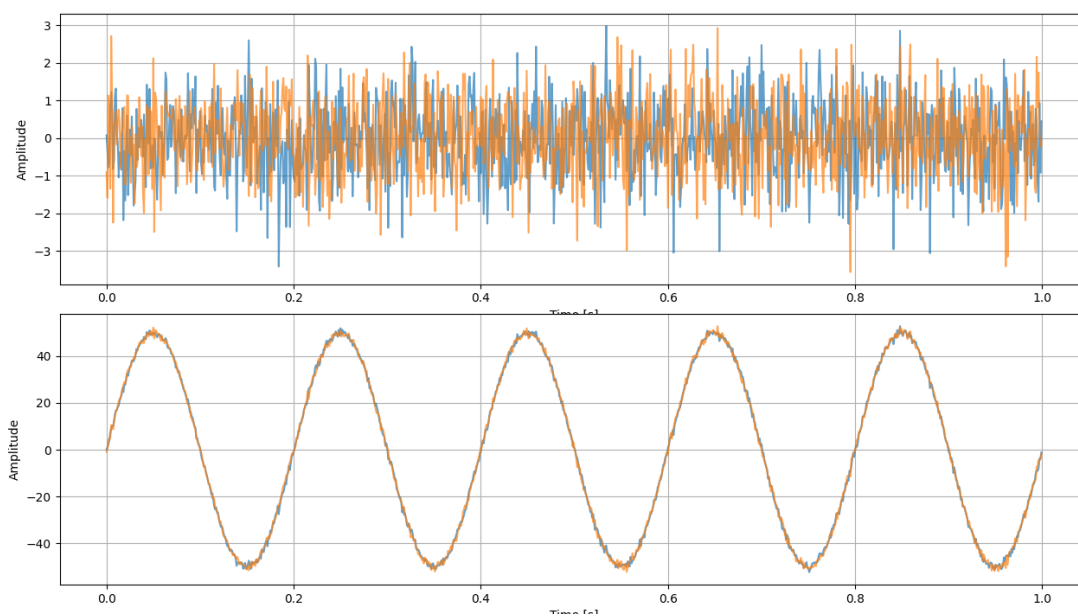
1 def effectCM_correlation(x_signal, y_signal, CM):
2     corre = []
3     for a in np.linspace(0, 5, 50):
4         row = []
5         for b in np.linspace(0, 5, 50):
6             xc_signal = x_signal + a * CM
7             yc_signal = y_signal + b * CM
8             row.append(calculate_correlation(yc_signal, xc_signal))
9         corre.append(row)
10    return corre

```

نویز های سفید ایجاد شده هیچ همبستگی ای با یکدیگر ندارند این موضوع را از روی شکل در نواحی نزدیک به صفر به وضوح می توان مشاهده کرد اما هنگامی که دامنه منبع مشترک افزایش می یابد و به نوعی میزان آن در مقابل نویز ها محسوس تر میشود یک همبستگی ایجاد میشود میدانیم این همبستگی نامعتبر و ناشی از منبع مشترک بود با زیاد شدن ضرایب این تاثیر بیشتر میشود به نوعی که معیار این دو نویز به همراه منبع را یک همبستگی کامل قلمداد می کند ، لذا تاثیر مخرب منبع مشترک بر روی این معیار مشاهده میشود.



شکل زیر تاثیر غالب بودن منبع مشترک بر سیگنال را نشان می دهد.

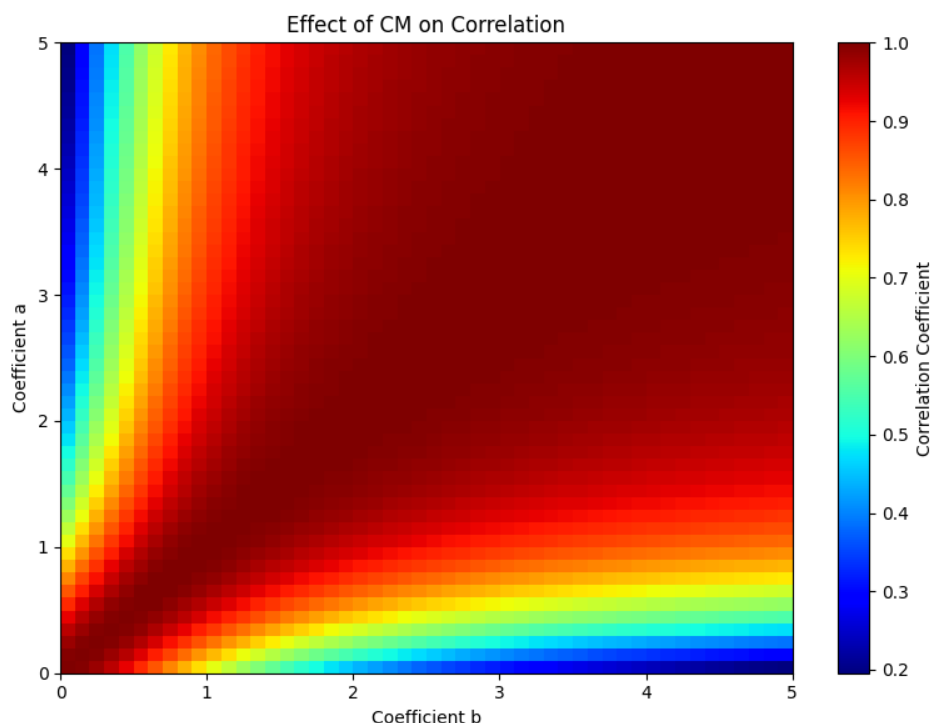


حال با دو سیگنال سینوسی به جای نویز به روش معیار قبلی هیت مپ را می آوریم:

```

۱ fs = 1000
۲ T = 1
۳ N = T * fs
۴ t = np.linspace(0, 1, 1000)
۵ signal1 = np.sin(2 * np.pi * fs * t)
۶ signal2 = np.sin(2 * np.pi * fs * (t))
۷ f_cm = 5
۸ t = np.linspace(0, T, N, endpoint=False)
۹ C_signal = np.sin(2 * np.pi * f_cm * t)
۱۰ correlation_matrix = effectCM_correlation(signal1, signal2, C_signal)

```

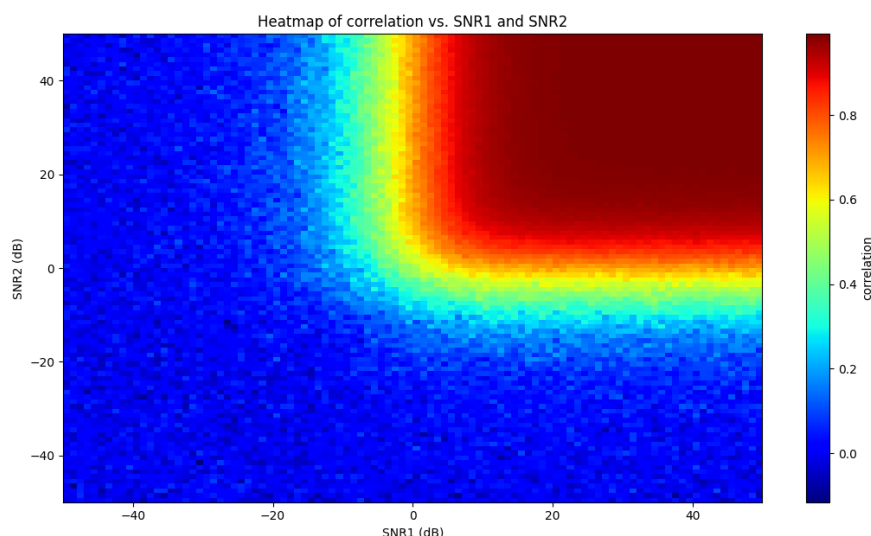


در اغلب موارد میزان همبستگی درست تشخیص داده شده است ، میتوانیم نتیجه بگیریم که منبع مشترک بر روی ایجاد همبستگی غیر واقعی اثر گذار است اما اگر همبستگی وجود داشته باشد وجود منبع چندان تاثیر نخواهد گذاشت.  
حال تاثیر نویز را بررسی میکنیم:

```

1 fs = 1000
2 T = 1
3 N = T * fs
4 t = np.linspace(0, T, N, endpoint=False)
5 f1, f2 = 10, 10
6 signal1 = np.sin(2 * np.pi * f1 * t)
7 signal2 = np.sin(2 * np.pi * f2 * (t-1/500))
8 def add_noise(signal, snr_db):
9     signal_power = np.mean(signal ** 2)
10    snr_linear = 10 ** (snr_db / 10)
11    noise_power = signal_power / snr_linear
12    noise = np.sqrt(noise_power) * np.random.randn(len(signal))
13    return signal + noise
14 def calculate_correlation(signal1, signal2):
15    mean1 = np.mean(signal1)
16    mean2 = np.mean(signal2)
17    Cov = np.sum((signal1 - mean1) * (signal2 - mean2))
18    sigma1sigma2 = np.sqrt(np.sum((signal1 - mean1)**2) * np.sum((signal2 - mean2)**2))
19    if sigma1sigma2 == 0:
20        return 0
21    return Cov / sigma1sigma2
22
23 snr_values = np.linspace(-50, 50, 100)
24 dpli_matrix = np.zeros((len(snr_values), len(snr_values)))
25 for i, snr1 in enumerate(snr_values):
26     for j, snr2 in enumerate(snr_values):
27         noisy_signal1 = add_noise(signal1, snr1)
28         noisy_signal2 = add_noise(signal2, snr2)
29         dpli_matrix[i, j] = calculate_correlation(noisy_signal1, noisy_signal2)

```



دو سیگنال با هم همبستگی دارند اما هنگامی که نسبت  $SNR$  ها از مقداری کمتر میشود این معیار به خوبی نتیجه را اعلام نمی کند.

## ۴ Phase-Amplitude Coupling (PAC)

یکی از معیارهای مهم در تحلیل ارتباطات کارکردی در سیگنال‌های عصبی، کوپلینگ فازدامنه (Phase-Amplitude Coupling - PAC) است. این معیار به بررسی ارتباط بین فاز سیگنال در یک باند فرکانسی پایین و دامنه سیگنال در باند فرکانسی بالا می‌پردازد. به عبارت دیگر، PAC نشان‌دهنده این است که چگونه فاز یک سیگنال با قدرت (دامنه) نوسانات سیگنال دیگری مرتبط است. فرایند محاسبه PAC شامل مراحل زیر است:

۱. فیلتر کردن سیگنال به دو باند فرکانسی مشخص (یک باند فرکانس پایین برای استخراج فاز و یک باند فرکانس بالا برای استخراج دامنه).
۲. استخراج فاز از سیگنال فرکانس پایین.
۳. استخراج دامنه از سیگنال فرکانس بالا.
۴. محاسبه میزان همبستگی بین فاز و دامنه. این ارتباط معمولاً با استفاده از شاخص‌هایی مانند MI اندازه‌گیری می‌شود.

### مزایا

- توانایی شناسایی ارتباطات محلی: برخلاف معیارهای مبتنی بر همبستگی، PAC می‌تواند به صورت محلی روی یک کانال نیز محاسبه شود.
- ارتباطات بین فرکانسی: این معیار برای شناسایی تعاملات پیچیده بین نواحی مغزی با فرکانس‌های متفاوت کاربردی است.
- کاربرد گسترده: PAC در زمینه‌های مختلف مانند تحلیل خواب، صرع، و حافظه کاربرد دارد.

### معایب

- حساسیت به پارامترهای باند فرکانسی: نتایج PAC به شدت به انتخاب باند فرکانسی وابسته است.
- پیچیدگی محاسباتی: محاسبه این معیار به روش‌های زمان فرکانس پیچیده نیاز دارد.
- حساسیت به نویز: وجود نویز می‌تواند به تداخل در محاسبات ارتباطات واقعی شود.

## مقاومت در برابر نویز و منبع مشترک

معیار PAC به طور کلی حساسیت بالایی به نویز و اثرات منبع مشترک دارد، اما راهکارهایی برای بهبود مقاومت آن ارائه شده است:

- استفاده از روش‌های پیش‌پردازش: حذف نویز و سیگنال‌های ناخواسته با استفاده از تکنیک‌هایی مانند ICA یا فیلترهای تطبیقی می‌تواند دقت PAC را افزایش دهد.
- کنترل منابع مشترک: در مطالعات تجربی، انتخاب طراحی‌های کنترل‌شده و استفاده از داده‌های شبیه‌سازی‌شده به کاهش اثرات منبع مشترک کمک می‌کند.
- محاسبه روی کانال‌های محلی: محاسبه PAC روی کانال‌های واحد یا نزدیک به هم می‌تواند اثر منابع مشترک را کاهش دهد.

## کاربردها

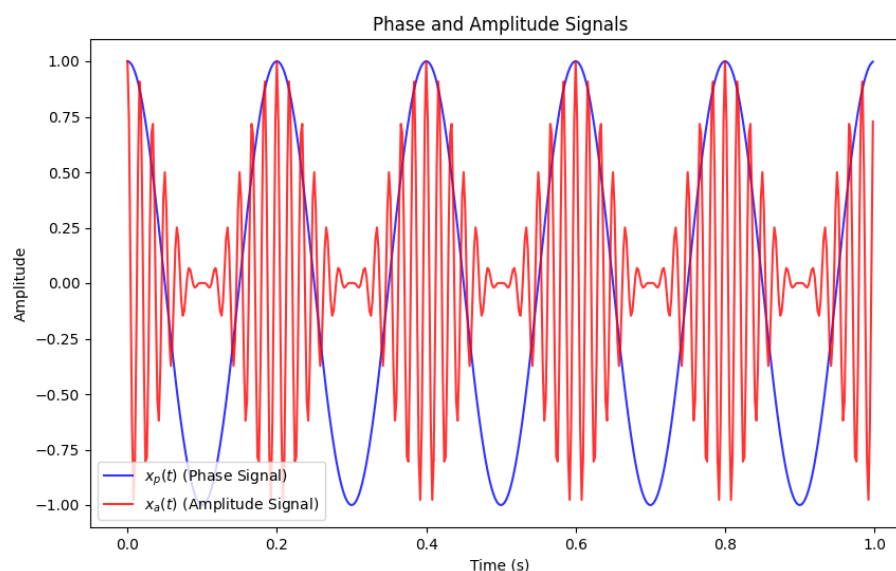
PAC در زمینه‌های زیر کاربرد دارد:

- تحلیل خواب: بررسی ارتباط بین امواج آهسته و نوسانات گاما در مراحل مختلف خواب.
- سرعت: شناسایی تغییرات غیرطبیعی در ارتباطات فرکانسی مرتبط با حملات صرع.
- حافظه و یادگیری: مطالعه نقش کوپلینگ فاز-دامنه در تثبیت حافظه و فرآیندهای یادگیری.

تابعی برای ایجاد این دوسیگنال مینویسیم:

```
1 def create_signals(kp, ka, f_phase, f_amp, chi, sigma_n, fs=500, duration=1):
2     t = np.linspace(0, duration, int(fs * duration), endpoint=False)
3     n1 = np.random.normal(0, 1, len(t))
4     n2 = np.random.normal(0, 1, len(t))
5     x_p = kp * np.cos(2 * np.pi * f_phase * t) + sigma_n * n1
6     am = ((1 - chi) * np.cos(2 * np.pi * f_phase * t) + (chi + 1)) / 2
7     x_a = ka * am * np.cos(2 * np.pi * f_amp * t) + sigma_n * n2
8     return t, x_p, x_a
```

```
1 kp = 1
2 ka = 1
3 f_phase = 5
4 f_amp = 60
5 chi = 0
6 sigma_n = 0
7 fs = 500
8 t, x_p, x_a = create_signals(kp, ka, f_phase, f_amp, chi, sigma_n, fs)
```



سیگنال  $x_p(t)$  نشان‌دهنده بخش فاز است که فرکانس پایینی دارد و چرخه کلی را تنظیم می‌کند. سیگنال  $x_a(t)$  شامل یک مدولاسیون دامنه است که به فاز وابسته است، با توجه به پارامترهای  $\chi$  و نویز، ارتباط بین دو سیگنال قابل مشاهده است.

حال برای بدست آوردن ماتریس های دامنه و فاز تابع زیر را داریم:

```
1 import pywt
2 def wavelet_transform(signal, fs, freq_range, wavelet='cmor1.5-0.5'):
3     frequencies = np.linspace(freq_range[0], freq_range[1], 100)
4     scales = pywt.scale2frequency(wavelet, 1) / frequencies * fs
5     coefficients, freqs = pywt.cwt(signal, scales, wavelet, sampling_period=1/fs)
6     return coefficients, freqs
```

```
1 def compute_phase_amplitude(signal_p, signal_a, fs):
2     phase_coeffs, phase_freqs = wavelet_transform(signal_p, fs, [2, 12])
3     phase_matrix = np.angle(phase_coeffs)
4
5     amp_coeffs, amp_freqs = wavelet_transform(signal_a, fs, [20, 80])
6     amplitude_matrix = np.abs(amp_coeffs)
7
8     return phase_matrix, amplitude_matrix, phase_freqs, amp_freqs
```

## ۱۰.۴ MLV

برای پیاده سازی این روش، در هر فرکانس  $f_a$  (فرکانس پایین مرتبط با فاز) و  $f_p$  (فرکانس بالا مرتبط با دامنه)، به صورت زیر عمل می شود:

۱. در هر لحظه زمانی  $t$ ، یک بردار مختلط تشکیل می شود که فاز لحظه ای  $\phi(t, f_p)$  و دامنه لحظه ای  $A(t, f_a)$  را ترکیب می کند.

۲. بردارهای محاسبه شده میانگین گیری شده و اندازه این میانگین محاسبه می شود.

۳. مقدار PAC از رابطه زیر بدست می آید:

$$PAC(n, m) = \left| \frac{1}{N} \sum_{t=0}^{N-1} A(t, f_a(n)) e^{j\phi(t, f_p(m))} \right|$$

که در این رابطه:

- $N$ : تعداد نمونه ها در بازه زمانی تحلیل.
- $A(t, f_a(n))$ : دامنه سیگنال در فرکانس  $f_a(n)$  در لحظه زمانی  $t$ .
- $\phi(t, f_p(m))$ : فاز سیگنال در فرکانس  $f_p(m)$  در لحظه زمانی  $t$ .

## ویژگی های PAC

- این روش امکان بررسی تعاملات بین فرکانس های مختلف را فراهم می کند.
- مقادیر بالای PAC نشان دهنده کوپلینگ قوی بین دامنه فرکانس بالا و فاز فرکانس پایین است.
- مقیاسی مناسب برای تحلیل ساختارهای زمانی پیچیده در داده های EEG و MEG است.

## مزایا و محدودیت ها

- مزایا:
  - توانایی اندازه گیری همبستگی بین فرکانس های مختلف.
  - مناسب برای تحلیل داده های غیرایستا.
- محدودیت ها:
  - حساسیت به نویز.
  - نیاز به پیش پردازش دقیق برای استخراج دقیق فاز و دامنه.

حال با توجه به رابطه تابع MLV را می نویسیم:

```

1 def calculate_MVL(phase_matrix, amplitude_matrix):
2     pac_matrix = np.zeros((amplitude_matrix.shape[0], phase_matrix.shape[0]))
3
4     for n in range(amplitude_matrix.shape[0]):
5         for m in range(phase_matrix.shape[0]):
6             complex_vector = amplitude_matrix[n, :] * np.exp(1j * phase_matrix[m, :])
7             pac_matrix[n, m] = np.abs(np.mean(complex_vector))
8     return pac_matrix

```

## (MI) Index Modulation

Modulation Index (MI) یکی از شاخص‌های پرکاربرد برای اندازه‌گیری همبستگی بین فاز فرکانس پایین و دامنه فرکانس بالا در سیگنال‌های عصبی است. این روش مبتنی بر بررسی توزیع نقاط فاز و دامنه در صفحه Phase-Amplitude و تحلیل عدم تقارن در این توزیع است. ایده کلی MI این است که اگر همبستگی فاز و دامنه وجود داشته باشد، دامنه‌های بزرگ‌تر به یک فاز خاص متمایل می‌شوند و دامنه‌های کوچک‌تر به فازهای دیگر.

### روش هیستوگرام

در این روش، مقادیر فاز و دامنه لحظه‌ای به صورت جفت‌های دو بعدی  $[A(t), \phi(t)]$  در هر نمونه زمانی تشکیل می‌شوند. سپس فازها در بازه  $[-\pi, \pi]$  به چندین بازه (bins) تقسیم شده و دامنه‌های متناظر با فازهای هر بازه محاسبه می‌شوند. این فرایند در قالب تابعی به نام ProxyHistogram به صورت زیر انجام می‌گیرد:

۱. بازه  $[-\pi, \pi]$  به تعداد مشخصی bins تقسیم می‌شود.

۲. دامنه‌های متناظر با فازهای موجود در هر بازه جمع شده و سپس نرمال‌سازی می‌شوند.

۳. خروجی این فرایند یک توزیع احتمالاتی از دامنه‌ها در بازه‌های مختلف فاز است.

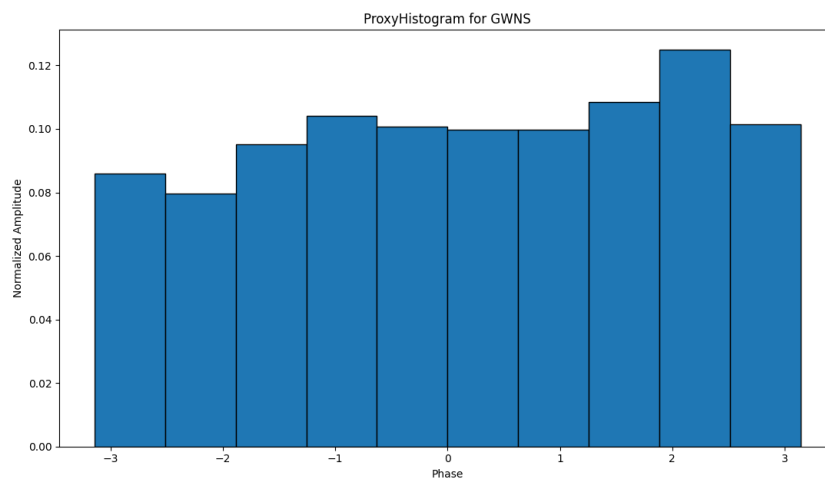
Phase	$\pi/9$	$4\pi/7$	$\pi$	$-\pi/8$	$7\pi/11$	0	$-2\pi/5$	$-3\pi/4$	$-5\pi/14$	$\pi/12$
Amplitude	0.8	0.25	0.1	0.7	0.6	0.95	0.4	0.14	0.35	0.9
Bin Number	2	3	3	2	3	2	1	1	1	2

```

1 def ProxyHistogram(points, num_bins):
2     x = np.array([p[0] for p in points])
3     y = np.array([p[1] for p in points])
4
5     bin_edges = np.linspace(-np.pi, np.pi, num_bins + 1)
6     bin_indices = np.digitize(x, bin_edges, right=True)
7
8     bin_sums = np.zeros(num_bins)
9     for i in range(1, num_bins + 1):
10         bin_sums[i - 1] = np.sum(y[bin_indices == i])
11
12     total_sum = np.sum(bin_sums)
13     normalized_amplitudes = bin_sums / total_sum if total_sum > 0 else bin_sums
14
15     return bin_edges, normalized_amplitudes
16 phases = np.array([np.pi/9, 4*np.pi/7, np.pi, -np.pi/8, 7*np.pi/11, 0, -2*np.pi/5, -3*np.pi/4, -5*np.pi/14, np.pi/12])
17 amplitudes = np.array([0.8, 0.25, 0.1, 0.7, 0.6, 0.95, 0.4, 0.14, 0.35, 0.9])
18 points = list(zip(phases, amplitudes))
19 num_bins = 3

```





خروجی تابع به ازای دو نویز

چون ارتباطی بین این دو سیگنال وجود ندارد، مطابق انتظار تقریباً توزیع یکنواخت است.

شاخص MI از طریق محاسبه فاصله عدم تقارن بین توزیع هیستوگرام حاصل و یک توزیع یکنواخت به دست می‌آید. یکی از روش‌های استاندارد برای این کار استفاده از Kullback-Leibler Divergence (KLD) است که به صورت زیر تعریف می‌شود:

$$D_{KL}(P||Q) = \sum_x P(x) \log \left( \frac{P(x)}{Q(x)} \right)$$

که در این رابطه:

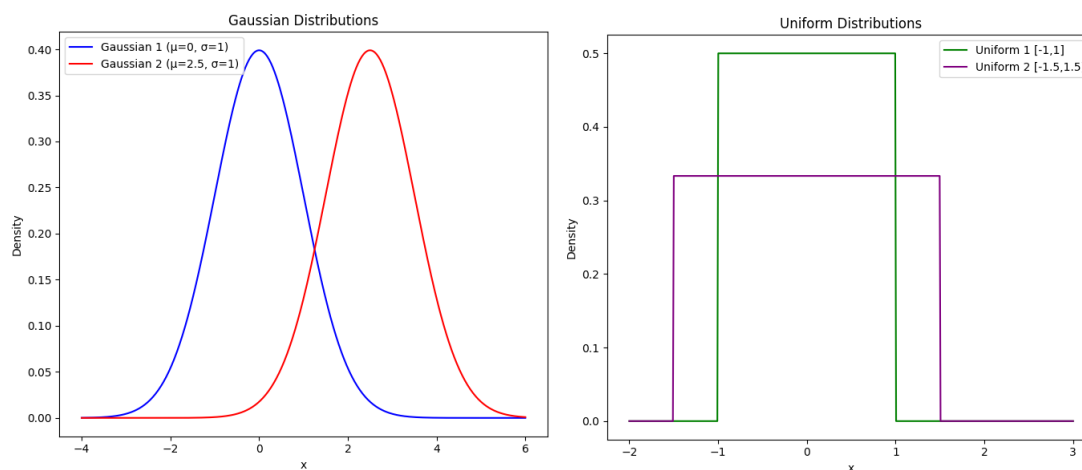
- $P(x)$ : توزیع حاصل از هیستوگرام فاز-دامنه.
- $Q(x)$ : توزیع یکنواخت (مرجع مقایسه).

```

1 def DKL(X, P, Q):
2     dkl = 0
3     for x in X:
4         if P[x] > 0 and Q[x] > 0:
5             dkl += P[x] * np.log2(P[x] / Q[x])
6     return dkl

```

به وسیله این شاخص می‌توان میزان شباهت دو توزیع را با یک عدد کیفی بررسی کرد به طور مثال برای توزیع‌های گوسی و یکنواخت، به ازای شباهت بیشتر میانگین و واریانس دو توزیع عدد بالاتری را مشاهده می‌کنیم.



```

۱ DKL(Gaussian_1 || Gaussian_2) = 2.9129
۲ DKL(Uniform_1 || Uniform_2) = 0.5367
۳ DKL(Gaussian_1 || Uniform_2) = -0.1159
۴ DKL(Uniform_1 || Gaussian_1) = 0.5167
۵ DKL(Uniform_2 || Gaussian_1) = 0.2639
۶ DKL(Uniform_1 || Gaussian_2) = 4.4893
۷ DKL(Uniform_2 || Gaussian_2) = 3.4413

```

## ویژگی‌ها و مزایا

- معیار تفکیک‌پذیری بالا: مقادیر MI بالا نشان‌دهنده همبستگی قوی‌تر بین فاز و دامنه است.
- مقاومت در برابر نویز: با پیش‌پردازش مناسب، این روش در برابر نویز نسبتاً مقاوم است.

## محدودیت‌ها

- حساسیت به تعداد bins: تعداد بازه‌های فاز (bins) می‌تواند بر دقت شاخص تأثیر بگذارد.
- نیاز به پردازش دقیق: استخراج دقیق فاز و دامنه سیگنال‌ها برای دقت بیشتر در محاسبه ضروری است.

مقادیر MI بزرگتر به معنای ارتباط قوی‌تر بین فاز و دامنه است. اگر توزیع دامنه‌های مربوط به فاز یکنواخت باشد، مقدار MI به صفر نزدیک خواهد بود، اما اگر دامنه‌ها در فاز خاصی متمرکز باشند، MI افزایش می‌یابد. این شاخص به‌طور گسترده برای مطالعه تعاملات فاز و دامنه در سیگنال‌های EEG و MEG استفاده می‌شود. حال به دنبال توابع پیاده شده شکل نهایی تابع را پیاده سازی میکنیم:

```

۱ def PAC_MI(phase_matrix, amplitude_matrix, num_bins):
۲     points = []
۳     for t in range(phase_matrix.shape[1]):
۴         for f in range(phase_matrix.shape[0]):
۵             points.append((phase_matrix[f, t], amplitude_matrix[f, t]))
۶
۷     _, normalized_amplitudes = ProxyHistogram(points, num_bins)
۸
۹     uniform_distribution = np.ones(num_bins) / num_bins
۱۰
۱۱     PAC = DKL(range(num_bins), normalized_amplitudes, uniform_distribution) / np.log2(num_bins)
۱۲     return PAC

```

برای بررسی صحت عملکرد و تأثیر پارامترها، مقادیر PAC را با دو معیار ارزیابی می‌کنیم و نمودار کومدولوگرام را بررسی می‌کنیم. رابطه کلی سیگنال به صورت زیر تعریف می‌شود:

$$x_a(t) = k_a \left( \frac{(1 - \chi) \cos(2\pi f_{\text{phase}} t) + \chi + 1}{2} \right) \cos(2\pi f_{\text{amp}} t) + \sigma_n n_2(t) \quad (۱)$$

## توضیح پارامترها:

- $k_a$ : ضریب مقیاس‌دهنده که دامنه کلی سیگنال را تنظیم می‌کند.
- $\cos(2\pi f_{\text{phase}} t)$ : مولفه‌ای با فرکانس پایین که برای کنترل فاز استفاده می‌شود.
- $\cos(2\pi f_{\text{amp}} t)$ : مولفه‌ای با فرکانس بالا که دامنه اصلی سیگنال را مشخص می‌کند.
- $\chi$ : پارامتری کلیدی که میزان وابستگی دامنه  $x_a(t)$  به فاز فرکانس پایین را تنظیم می‌کند:
  - مقدار  $\chi$  در بازه  $[0, 1]$  قرار دارد.
  - افزایش یا کاهش مقدار  $\chi$ ، میزان همبستگی دامنه و فاز را تنظیم می‌کند.
- $\sigma_n n_2(t)$ : مولفه نویز، تولید شده از نویز سفید گاوسی با انحراف معیار  $\sigma_n$ .

## تنظیمات وابستگی دامنه به فاز:

پارامتر  $\chi$  تأثیر قابل توجهی بر وابستگی دامنه  $x_a(t)$  به فاز فرکانس پایین ( $f_{\text{phase}}$ ) دارد:

وقتی  $\chi = 0$ :

رابطه به این شکل ساده می‌شود:

$$x_a(t) = k_a \frac{\cos(2\pi f_{\text{phase}}t) + 1}{2} \cos(2\pi f_{\text{amp}}t) + \sigma_n n_2(t) \quad (2)$$

- در این حالت، دامنه سیگنال کاملاً به فاز فرکانس پایین وابسته است.
- این وضعیت نشان‌دهنده حداکثر همبستگی بین فاز و دامنه است.

وقتی  $\chi = 1$ :

رابطه به صورت زیر تبدیل می‌شود:

$$x_a(t) = k_a \cos(2\pi f_{\text{amp}}t) + \sigma_n n_2(t) \quad (3)$$

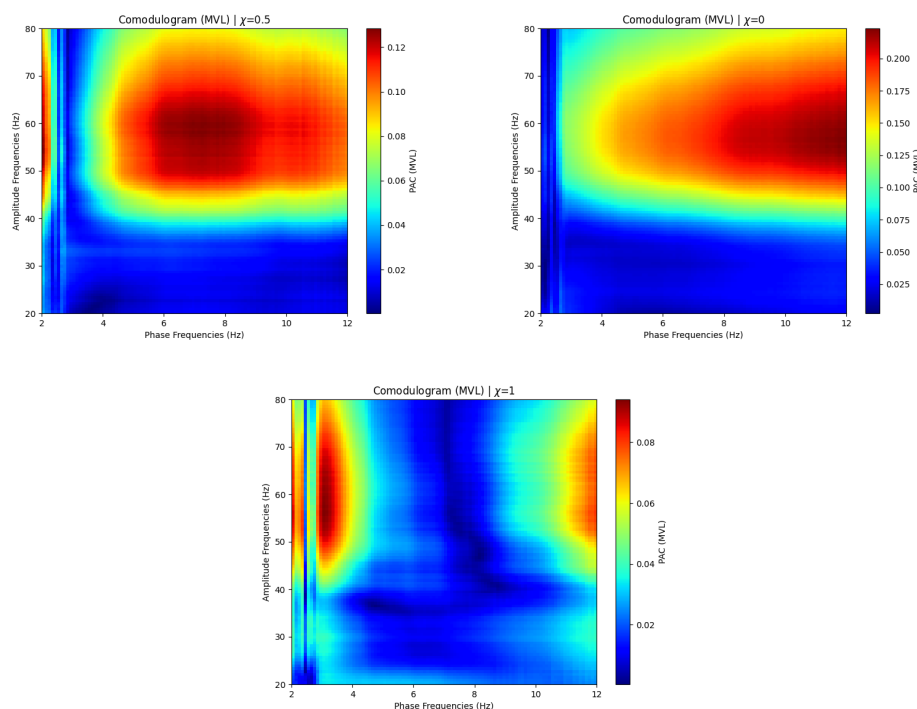
- در این حالت، دامنه سیگنال از فاز فرکانس پایین کاملاً مستقل است.
- این وضعیت نشان‌دهنده عدم وجود همبستگی بین فاز و دامنه است.

```

1 chi_values = [0, 0.5, 1]
2 for chi in chi_values:
3     t, x_p, x_a = create_signals(kp, ka, f_phase, f_amp, chi, sigma_n=0.5, fs=fs, duration=duration)
4     phase_matrix, amplitude_matrix, phase_freqs, amp_freqs = compute_phase_amplitude(x_p, x_a, fs)
5     PAC_MI = ModulationIndex(phase_matrix, amplitude_matrix, num_bins=18)
6     PAC_MVL = MVL(phase_matrix, amplitude_matrix)
7     plot_comodulogram(PAC_MVL, "MVL", chi, phase_freqs, amp_freqs, variable_name=r"$\chi$")
8     print(PAC_MI)
```

```

1 MI:
2 0.03176564242615874 # chi = 0
3 0.00633553793963841 # chi = 0.5
4 0.0007436122526502761 # chi = 1
```



$$\chi \in \{0, 0.5, 1\}$$

مشاهده میکنیم که در حال وابستگی یا به عبارتی  $\chi = 0$  وابستگی بین فرکانس پایین فاز و فرکانس بالای دامنه وجود دارد و در حالتی که  $\chi = 1$  ارتباطی مشاهده نمیشود. هر دو معیار تا حد خوبی توانسته اند میزان مرتبط بودن یا نبودن را مشخص کنند.

تأثیر نویز  $\sigma_n$

نویز موجود در رابطه  $\sigma_n n_2(t)$  باعث کاهش وضوح همبستگی بین فاز و دامنه می‌شود. با افزایش مقدار  $\sigma_n$ ، نویز بیشتر شده و همبستگی ضعیف‌تر می‌گردد.

```

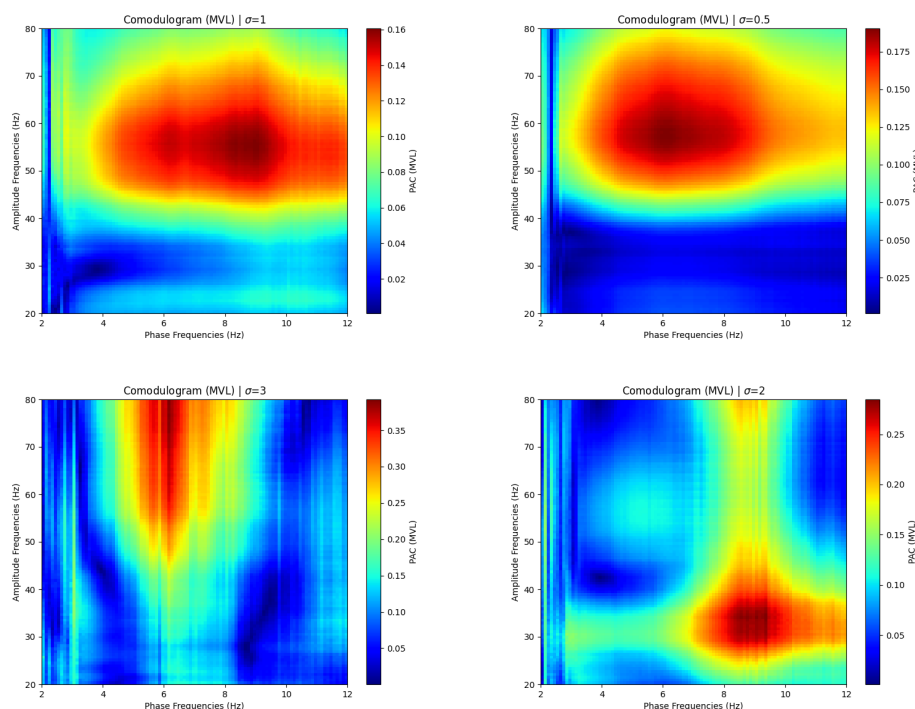
1 sigma_values = [0.5, 1, 2, 3]
2 chi = 0
3 for sigma_n in sigma_values:
4     t, x_p, x_a = create_signals(kp, ka, f_phase, f_amp, chi, sigma_n=sigma_n, fs=fs, duration=duration)
5     phase_matrix, amplitude_matrix, phase_freqs, amp_freqs = compute_phase_amplitude(x_p, x_a, fs)
6     PAC_MI = ModulationIndex(phase_matrix, amplitude_matrix, num_bins=18)
7     PAC_MVL = MVL(phase_matrix, amplitude_matrix)
8     plot_comodulogram(PAC_MVL, "MVL", sigma_n, phase_freqs, amp_freqs, variable_name=r"$\sigma_n$")
9     print(PAC_MI)

```

```

1 MI:
2 0.016868554420903795 # sigma = 0.5
3 0.007385132668591943 # sigma = 1
4 0.002530323361176201 # sigma = 2
5 0.0015002315733130486 # sigma = 3

```



$$\sigma \in \{0.5, 1, 2, 3\}$$

## رفتار MI در برابر نویز

MI بر اساس انتروپی توزیع دامنه در بازه‌های مختلف فاز عمل می‌کند. نتایج نشان می‌دهد:

- مقادیر MI با افزایش شدت نویز ( $\sigma = 0.5$  تا  $\sigma = 3.0$ ) کاهش می‌یابند.
- برای  $\sigma = 0.5$ ، مقدار MI برابر 0.0168 است، در حالی که برای  $\sigma = 3.0$ ، مقدار آن به 0.0015 کاهش می‌یابد.
- این رفتار نشان می‌دهد که MI حتی در حضور نویز متوسط می‌تواند بخشی از وابستگی فاز-دامنه را شناسایی کند.
- MI از انتروپی نرمال‌شده استفاده می‌کند که تغییرات نویزی را به عنوان بخشی از توزیع یکنواخت در نظر می‌گیرد.
- نرمال‌سازی MI باعث می‌شود که اثر نویز تصادفی کمتر بر مقدار نهایی تأثیر بگذارد.

## رفتار MVL در برابر نویز

MVL طول میانگین بردارهای مختلط حاصل از دامنه و فاز را محاسبه می‌کند. این روش به دلایل زیر حساسیت بیشتری به نویز دارد:

- MVL مستقیماً به دامنه و فاز وابسته است. نویز می‌تواند باعث تغییرات تصادفی در دامنه و زاویه فاز شود.
- با افزایش شدت نویز، پراکندگی بردارهای مختلط بیشتر می‌شود که به کاهش شدید مقدار MVL می‌گردد.
- MI به دلیل ساختار مبتنی بر انتروپی، کاهش تدریجی و پایدارتری در برابر نویز نشان می‌دهد.
- MVL به دلیل حساسیت مستقیم به دامنه و فاز، کاهش سریع‌تری با افزایش نویز تجربه می‌کند.
- MI در برابر نویز مقاومت بیشتری دارد و می‌تواند برای شناسایی PAC در شرایط نویز کم تا متوسط استفاده شود.
- MVL به دلیل حساسیت بالا به نویز برای شرایطی که نویز کم است مناسب‌تر است، اما در حضور نویز زیاد کارایی آن کاهش می‌یابد.
- در شرایط نویز شدید ( $\sigma \geq 2.0$ )، هر دو روش کارایی خود را از دست می‌دهند و PAC تقریباً ۰ می‌شود.

تابع جهت اعمال شیفت چرخشی:

```

1 def circular_shift(signal, shift_ms, fs):
2     shift_samples = int(shift_ms / 1000 * fs)
3     return np.roll(signal, shift_samples)

```

```

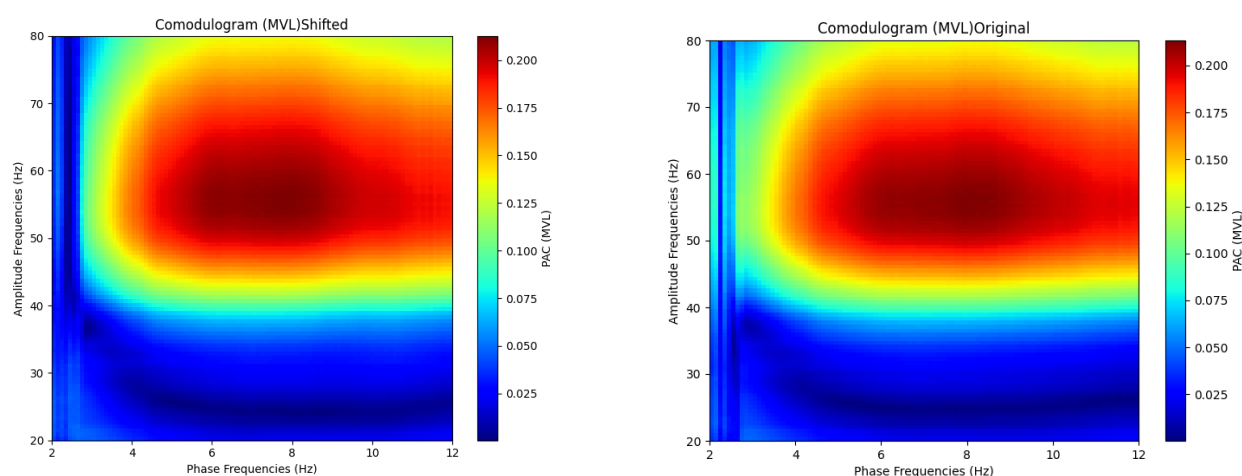
1 t, x_p, x_a = create_signals(kp, ka, f_phase, f_amp, chi, sigma_n, fs=fs, duration=duration)
2 shift_ms = 100
3 x_p_shifted = circular_shift(x_p, shift_ms, fs)

```

```

1 MI:
2 0.026766930591953307 # original
3 0.026829240192476532 # shifted

```

تأثیر شیفت  $x_p$ 

با بررسی دو تصویر کومدلوگرام بر اساس معیار MVL و معیار MI تغییرات با وجود تاخیر کم هستند لذا میتوانیم PAC را یک معیار مقاوم حداقل از بررسی این چینی بنامیم.

```

1 def add_cs(x_p, x_a, alpha, beta, f_c, t):
2     ct = np.cos(2 * np.pi * f_c * t)
3     x_p_c = x_p + alpha * ct
4     x_a_c = x_a + beta * ct
5     return x_p_c, x_a_c

```

```

1 kp = 1
2 ka = 1
3 f_phase = 5
4 f_amp = 60
5 chi = 0
6 sigma_n = 0.5
7 fs = 500
8 duration = 1
9 f_c = 40
10 a_bvalues = [0, 5, 10, 50]
11 t, x_p, x_a = create_signals(kp, ka, f_phase, f_amp, chi, sigma_n, fs=fs, duration=duration)
12 for a_b in a_bvalues:
13     x_p_c, x_a_c = add_cs(x_p, x_a, a_b, a_b, f_c, t)
14     phase_matrix, amplitude_matrix, phase_freqs, amp_freqs = compute_phase_amplitude(x_p_c, x_a_c, fs)
15     PAC_MI = MI(phase_matrix, amplitude_matrix, num_bins=18)
16     PAC_MVL = MVL(phase_matrix, amplitude_matrix)

```

```

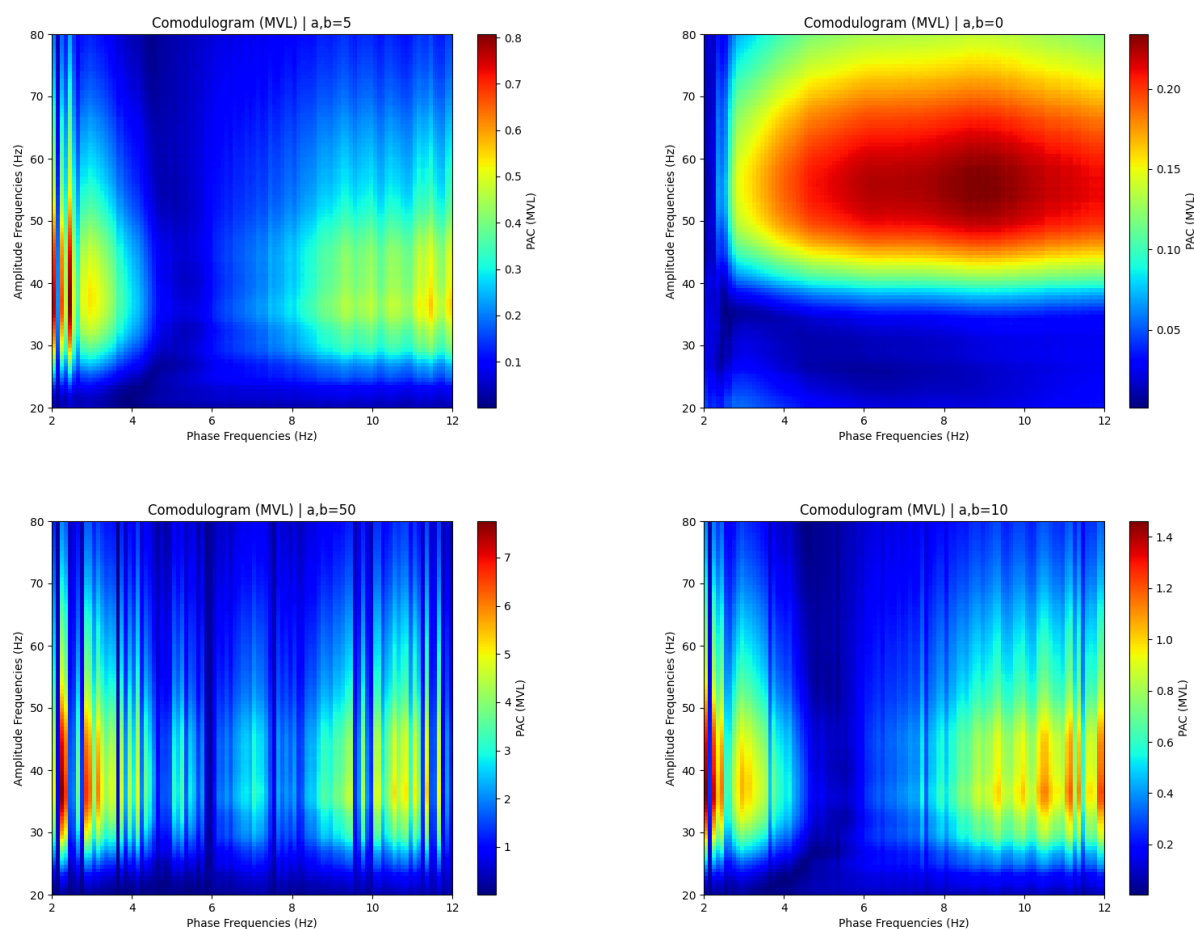
۱۷ plot_comodulogram(PAC_MVL, "MVL", a_b, phase_freqs, amp_freqs, variable_name="a,b")
۱۸ print(f"{PAC_MI}")

```

```

۱ MI :
۲ 0.03460151813026085 # a,b = 0
۳ 0.0004912686273834117 # a,b = 5
۴ 0.0005685966866890023 # a,b = 10
۵ 0.016746939595327363 # a,b = 50

```



اعمال منبع مشترک

## تحلیل رفتار MI در حضور منبع مشترک

ابتدا معیار MI را بررسی میکنیم در حالت عادی و بدون در نظر گرفتن اثر منبع مشترک، خروجی این معیار تقریباً برابر 0.035 است. با افزایش منبع مشترک این میزان کاهش می‌یابد اما اگر این مقدار بر سیگنال غلبه کند مانند  $a, b = 10, 50$  معیار دچار افزایش و اطلاعات نامعتبری را خروجی می‌دهد. در این صورت منبع مشترک به صورت غالب عمل میکند. این نشان دهنده این است که این معیار در محیط‌های دارای منبع مشترک رفتار خوبی از خود نمی‌دهد و باید در مراحل پیش پردازش اثر منبع مشترک را استخراج و حذف کرد تا داده‌های خروجی برای این معیار تولید خروجی معتبری کنند. پس به طور کلی میتوان گفت که اضافه شدن منبع مشترک باعث همبستگی کاذب و از بین رفتن الگوهای واقعی هماهنگی بین فرکانس‌ها می‌شود.

## تحلیل رفتار MVL در حضور منبع مشترک

معیار MVL تغییرات الگوی هماهنگی بین فرکانس‌های فاز و دامنه را در حضور منبع مشترک نشان می‌دهد. رفتار این معیار نیز با افزایش مقادیر  $a, b$  (شدت منبع مشترک) ابتدا دچار کاهش و سپس افزایش می‌شود. دلیل این تغییرات به مانند MI در دو مرحله توضیح داده می‌شود:



در مقادیر کم  $a, b$ ، اضافه شدن منبع مشترک باعث کاهش هماهنگی واقعی بین فرکانس‌ها می‌شود. این تغییرات در کومودولوگرام‌ها به صورت کاهش شدت نواحی هماهنگی و پراکندگی آن‌ها دیده می‌شود.

- در مقادیر کوچک  $a, b$ ، اثر منبع مشترک منجر به افزایش نویز و تداخل می‌شود، که باعث کاهش مقدار MVL و کم‌رنگ شدن نواحی با شدت بالا می‌شود.
- در این مرحله، MVL نمی‌تواند به خوبی اثر منبع مشترک را تفکیک کند و شدت هماهنگی واقعی کاهش می‌یابد.

در شکل نخست رنگ قرمز نشان دهنده همبستگی می‌باشد با افزایش ضرائب در شکل دوم می‌بینیم که بخشی از اطلاعات از بین رفته اند. در مقادیر بالای  $a, b$  (مانند  $a, b = 50$ )، شدت منبع مشترک به قدری زیاد می‌شود که الگوهای یکنواخت و غالب ایجاد می‌کند. این شرایط باعث می‌شود برخی نواحی از هماهنگی دوباره در نمودار MVL آشکار شوند.

- در مقادیر بسیار بزرگ  $a, b$ ، منبع مشترک به صورت غالب بر کل سیگنال تأثیر می‌گذارد، که باعث بهبود افزایش هماهنگی در برخی محدوده‌های فرکانسی می‌شود.
- البته این بهبود ناشی از واقعی شدن سیگنال‌ها نیست، بلکه به دلیل افزایش اثر منبع مشترک و یکنواخت شدن سیگنال‌هاست.

رفتار MVL در برابر اثر منبع مشترک به صورت زیر خلاصه می‌شود:

۱. در مقادیر کم  $a, b$ : هماهنگی واقعی کاهش می‌یابد و MVL پراکندگی بیشتری در نمودار نشان می‌دهد.

۲. در مقادیر بالای  $a, b$ : نواحی شدت هماهنگی به صورت مصنوعی ظاهر می‌شوند، که ناشی از افزایش منبع مشترک است.

به طور کلی MVL مقاوم‌تر است. در حالی که هر دو معیار در مواجهه با منبع مشترک دچار مشکلاتی می‌شوند، MVL قادر است اطلاعات واقعی را بهتر حفظ کند و تغییرات کاذب کمتری را در مقایسه با MI نشان دهد. MI به شدت تحت تأثیر منبع مشترک قرار می‌گیرد و سریع کاهش می‌یابد و اطلاعات مفید خود را سریعاً از دست می‌دهد و با حضور منبع مشترک دیتای کاذب تولید می‌کند.

## ۵ مقایسه معیارهای ارتباطی

### ۱۰۵ تأثیر تاخیر در نوسانات ناشی از فرستنده

تاخیر زمانی یکی از ویژگی‌های کلیدی تعاملات سیگنال‌های زیستی است که بر عملکرد معیارها تأثیر می‌گذارد.

#### ۱۰۱.۵ dPLI

این معیار به دلیل تمرکز بر اختلاف فاز بین سیگنال‌ها، در شناسایی ارتباطات با تاخیر زمانی بسیار موثر است و می‌تواند جهت ارتباط را شناسایی کند. این ویژگی باعث می‌شود که dPLI در تعاملات غیرهم‌زمان که اختلاف فاز مشخصی دارند، عملکرد بهتری داشته باشد. با این حال، در تاخیرهای بسیار کوتاه یا هم‌زمانی زیاد (نزدیک به صفر)، ممکن است دقت آن کاهش یابد.

#### ۲۰۱.۵ PPC

این معیار که بر همبستگی توان سیگنال‌ها مبتنی است، نسبت به تغییرات زمانی حساسیت کمتری دارد. تاخیر در دامنه سیگنال‌ها تأثیر چندانی بر آن ندارد، اما همین ویژگی می‌تواند باعث نادیده‌گرفتن تعاملات علی شود. بنابراین، PPC برای شناسایی تعاملات زمانی مناسب نیست. البته این نتیجه با در نظر گرفتن همبستگی توان به توان بررسی شده است در تمرین، همبستگی دامنه به دامنه بررسی شده است که مشاهده می‌شود با این تعریف تاخیر در همبستگی دارای اثر است و گاهی ممکن است دوسیگنالی که هیچ همبستگی ندارند، همبسته و بالعکس شناسایی شوند.

#### ۳۰۱.۵ PAC

- MI: شاخص MI از تحلیل اطلاعات استفاده می‌کند و تاخیر را به‌عنوان بخشی از عدم قطعیت سیگنال در نظر می‌گیرد. این ویژگی باعث می‌شود MI در برابر تأخیر مقاوم‌تر باشد، به‌خصوص در سیگنال‌هایی با هم‌پوشانی زمانی.

- MLV: شاخص MLV وابستگی دقیق بین فاز و دامنه را تحلیل می‌کند. تاخیر زمانی ممکن است این وابستگی را مختل کند، به‌ویژه اگر تاخیر باعث کاهش هماهنگی بین فاز و دامنه شود.

معیار dPLI در شناسایی اثرات تاخیر عملکرد بهتری دارد. در بین شاخص‌های PAC، MI نسبت به MLV در مواجهه با تاخیر مقاوم‌تر است.

## ۲۰۵ اثر منبع مشترک

وجود منبع مشترک در سیستم‌های زیستی می‌تواند ارتباطات غیرواقعی ایجاد کند.

## ۱۰.۲.۵ dPLI

این معیار به دلیل تمرکز بر ترتیب زمانی بین سیگنال‌ها، اثر منبع مشترک را کاهش می‌دهد. منبع مشترک نمی‌تواند به تنهایی باعث ایجاد جهت مشخص در ارتباط شود، بنابراین تاثیر آن بر dPLI محدود است.

## ۲۰.۲.۵ PPC

این معیار به شدت به منبع مشترک حساس است. منبع مشترک می‌تواند دامنه‌های مشابه ایجاد کند، حتی اگر ارتباط واقعی بین سیگنال‌ها وجود نداشته باشد. همبستگی دامنه تنها ارتباط سطحی را نشان می‌دهد و نمی‌تواند اثر منبع مشترک را از سیگنال اصلی جدا کند.

## ۳۰.۲.۵ PAC

## • MI:

شاخص MI به دلیل تحلیل توزیع اطلاعات، اثر منبع مشترک را کاهش می‌دهد. این ویژگی به آن امکان می‌دهد الگوهای اطلاعاتی واقعی را از ارتباطات غیرواقعی جدا کند.

## • MLV:

وابستگی فاز دامنه در MLV ممکن است تحت تاثیر منبع مشترک قرار بگیرد، زیرا الگوهای فاز دامنه از منبع مشترک می‌توانند باعث ایجاد ارتباط غیرواقعی شوند.

معیار dPLI مقاوم‌ترین معیار در برابر اثرات منبع مشترک است، در حالی که PPC بیشترین آسیب‌پذیری را دارد. در بین شاخص‌های MI PAC، نسبت به MLV مقاوم‌تر است.

## ۳.۵ اثر نویز

نویز یکی از مهم‌ترین عوامل مخرب و غیرقابل اجتناب در تحلیل سیگنال‌هاست.

## ۱۰.۳.۵ dPLI

این معیار به دلیل تمرکز بر فاز، اثر نویز دامنه‌ای را کاهش می‌دهد. با این حال، در شرایط نویزی شدید عملکرد آن ممکن است کاهش یابد.

## ۲۰.۳.۵ PPC

نویز دامنه می‌تواند عملکرد این معیار را به شدت مختل کرده و خروجی غیرواقعی تولید کند، این اثر را در بررسی تاثیر نویز دامنه بررسی کردیم.

## ۳۰.۳.۵ PAC

## • MI:

تحلیل اطلاعاتی MI به آن کمک می‌کند تا اثر نویز را کاهش دهد، زیرا نویز را به عنوان بخشی از عدم قطعیت در نظر می‌گیرد.

## • MLV:

این شاخص به دلیل وابستگی دقیق به فاز و دامنه، نسبت به نویز حساس‌تر است.

معیارهای MI و dPLI در شرایط نویزی مقاوم‌تر از PPC و MLV عمل می‌کنند، اما شدت نویز بر عملکرد همه معیارها تاثیر می‌گذارد.