

## ( شبیه سازی مدولاسیون APSK )

### ۱-۱ مقدمه

در سیستم‌های ارتباطی دیجیتال، داده‌ها هنگام ارسال در کانال‌های نویزی ممکن است دچار خطا شوند. یکی از روش‌های موثر برای انتقال امن و مقاوم داده‌ها، استفاده از مدولاسیون APSK (Amplitude Phase Shift Keying) است. با نگاشت داده‌ها روی حلقه‌های با دامنه و فاز متفاوت، توانایی بیشتری در مقابله با نویز نسبت به مدولاسیون‌های ساده‌تر دارد. در این شبیه‌سازی، مدولاسیون ۱۶-APSK روی دیتاست واقعی MNIST اعمال شد تا عملکرد سیستم در شرایط نویزی و میزان خطای بیت (BER) بررسی شود.

### ۱-۲ آماده‌سازی دیتاست

- ما از MNIST استفاده کردیم که تصاویر دست‌نویس اعداد ۰ تا ۹ است.
- هر تصویر  $28 \times 28$  پیکسل است و هر پیکسل ۸ بیت دارد (۰ تا ۲۵۵).
- ابتدا تصاویر را به یک بردار باینری طولانی تبدیل کردیم.
  - مثال: هر تصویر  $\rightarrow 784$  پیکسل  $\times 8$  بیت =  $6272$  بیت
  - این بیت‌ها همان «پیام» ما هستند که قرار است ارسال شود.

### ۱-۳ مدولاسیون APSK

- APSK (Amplitude Phase Shift Keying) یک روش مدولاسیون است که داده‌ها را روی حلقه‌های با دامنه متفاوت و فازهای متفاوت نگاشت می‌کند.
- در شبیه‌سازی ما از 16-APSK استفاده کردیم:
  - 2 حلقه: داخلی (۴ نماد) و خارجی (۱۲ نماد)
  - هر گروه ۴ بیتی از پیام  $\rightarrow$  یک نماد  $\rightarrow$  APSK مشخص می‌شود که روی کدام حلقه و با چه فاز قرار گیرد.

- هدف: تبدیل بیت‌ها به سیگنال‌های پیچیده (complex) که قابل ارسال روی کانال باشند.
- بیت‌ها به نقاطی روی یک صفحه مختصات (In-phase / Quadrature) تبدیل می‌شوند.

### ۴-۱ شبیه سازی کانال AWGN

- کانال واقعی نویز دارد، برای شبیه‌سازی از AWGN (Additive White Gaussian Noise) استفاده شد.
- نویز هم به قسمت In-phase و هم به Quadrature اضافه شد.
- نتیجه: هر نماد APSK در صفحه مختصات کمی جابه‌جا شد، یعنی هنگام دریافت ممکن است با نمادهای دیگر اشتباه گرفته شود.
- به زبان ساده: مثل این است که روی یک هدف نشانه گرفته‌ایم، اما باد کمی تیر را منحرف کرده است.

### ۵-۱ دمولاسیون APSK

- وظیفه دمولاسیون: تشخیص نزدیک‌ترین نماد به سیگنال دریافت‌شده و تبدیل آن به بیت‌های اصلی.
- الگوریتم:
- ۱. فاصله هر نماد دریافت‌شده تا تمام نقاط ممکن APSK را محاسبه می‌کنیم.
- ۲. نزدیک‌ترین نماد انتخاب می‌شود.
- ۳. نماد انتخاب‌شده → بیت اصلی بازسازی می‌شود.
- نتیجه: تا حد امکان خطاهای ایجاد شده توسط نویز اصلاح می‌شوند.

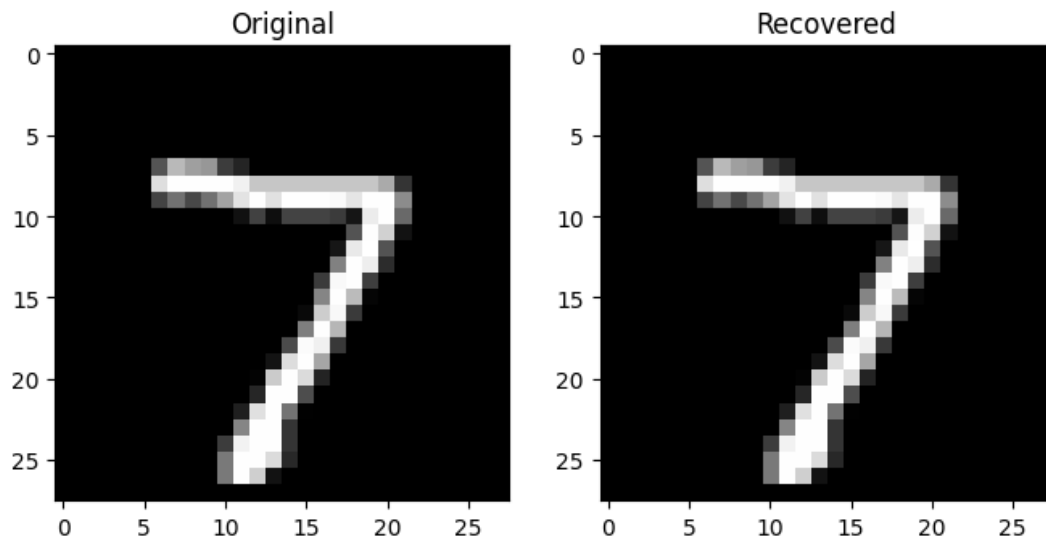
### ۶-۱ محاسبه BER (Bit Error Rate)

- بیت‌های بازسازی شده با بیت‌های اصلی مقایسه می‌شوند.
- درصد بیت‌های اشتباه BER →

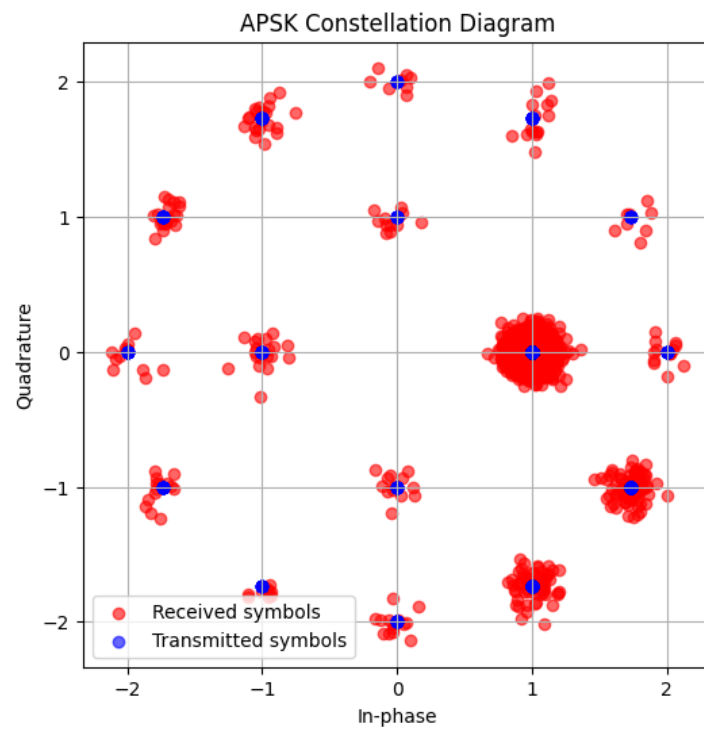
- مثال:
- 62720 بیت پردازش شد
- 11680 بیت قبل از دمدولاسیون اشتباه → به دلیل نویز
- 5619 بیت بعد از دمدولاسیون اشتباه APSK → توانست حدود نصف خطاها را اصلاح کند

## ۷-۱ نمایش نتایج

- تصویر اصلی VS تصویر بازیابی شده: تقریباً شبیه هستند، اما ممکن است نویز اندکی روی پیکسل‌ها باقی مانده باشد.
- Constellation diagram:
- نقاط آبی → نمادهای ارسال شده
- نقاط قرمز → نمادهای دریافت شده بعد از نویز
- هر چقدر نقاط قرمز نزدیک نقاط آبی باشند → دمدولاسیون دقیق‌تر
- به زبان ساده: نمودار constellation مانند نقشه‌ای است که نشان می‌دهد چقدر سیگنال‌ها بعد از نویز منحرف شدند و APSK توانست آنها را به درستی برگرداند.



شکل یک ( تصویر اصلی در سمت راست و تصویر باز یابی شده در سمت چپ



شکل ۲ ( دیاگرام مدولاسیون APSK

## ۸-۱ جمع بندی عملکرد

- Bit  $\rightarrow$  APSK symbol  $\rightarrow$  Channel  $\rightarrow$  Noisy symbol  $\rightarrow$  Bits
- APSK اجازه می‌دهد بیت‌ها روی حلقه و فاز کدگذاری شوند  $\rightarrow$  توانایی بهتر در مقابل نویز.
- دمدولاسیون سعی می‌کند نزدیک‌ترین نماد را پیدا کند  $\rightarrow$  بخشی از خطاها اصلاح می‌شود.
- BER کاهش می‌یابد، ولی تمام خطاها اصلاح نمی‌شوند  $\rightarrow$  به دلیل نویز زیاد یا تشابه نمادها.