Personal matlab sort out about communication simulations finally

1.Basics

- ## Scalars

```
a = 10
a = 10+1i       % 推荐使用1i
```

- ## Vectors

```
a = [1 3 5]     % 1行3列
a = [1,3,5]     % 使用,与否效果相同
a = 1:2:7       % begin:step:end
a = [1;3;5]     % 3行1列
```

- ## Matrices

```
A = [1,2,3;4,5,6;7,8,9] % 三行三列矩阵
```

- ## Operations

1. `*` 和 `.*` 的区别

   > 在进行数值运算和数值乘矩阵运算，两者无区别， `a*b=a.*b; a*B=a.*B; B*a=B.*a` （其中小写字母表示数值，大写字母表示矩阵，下同）。
   > 在处理矩阵乘矩阵时， `*` 表示普通的矩阵乘法，要求前面矩阵的列数等于后面矩阵的行数； `.*` 表示两个矩阵对应元素相乘，要求两个矩阵行数列数都相等。

   ```
   >> [1,2,3]*[1,2;3,4;5,6]        % 矩阵乘法
   ans =
   22    28

   >> [1,2,3].*[4,5,6]             % 矩阵点乘
   ans =
   4    10    18
   ```

2. `/` 和 `./` 的区别

> 数值运行时，这两种没有区别， `a/b=a./b`
>
> 数值与矩阵运算，数值在前时只能用 `./` ;数值在后时 `a/b=a./b`
>
> 矩阵与矩阵， `A/B` 可以看作是 `A*inv(B)`；`A./B` 表示A矩阵和B矩阵对应元素相除，需要矩阵维度保持一致。
>
> 另外有 `\` 和 `.\` 表示前除，类似小学时候详细区的分除和被除运算，一般习惯不使用这种。

```
>> [4,5]/[1,2;3,4]                    % 矩阵除法
ans =
-0.5000    1.5000

>> [4,5,6]./[1,2,3]                   % 矩阵点除
ans =
4.0000    2.5000    2.0000
```

3. `A(1,1)` 、 `A(:,1)` 、 `A(1,:)` 、 `A(:,:)`
4. `[1+2*1i,3+4*1i]'` 和 `[1+2*1i,3+4*1i].'` 都表示矩阵的转置,实数情况下 `A'` 和 `A.'` 没有区别;复数情况下 `A'` 会产生 `A` 的转置共轭矩阵， `A.'` 产生普通的转置矩阵。

```
[1+2*1i,3+4*1i]'=[1-2*1i;3-4*1i]
[1+2*1i,3+4*1i].'=[1+2*1i;3+4*1i]
```

- ## Special number

  `pi`

  `Inf` 、 `inf` :infinity,i.e.,1/0

  `NaN` 、 `nan` :not a number,e.g.,0/0

  `eps` :accuracy of the matlab,eps= 2-52 ≈ 2.2204×10-16

- ## General functions

  - Trigonometric functions:

    `cos(x)` 、 `sin(x)` 、 `tan(x)`
  - Complex:

    z = a+bi = $re^{i\theta}$

    `real(z)` 、 `imag(z)` 、 `abs(z)` 、 `angle(z)` 、 `conj(z)`
  - Exponential and logarithm functions

Name	Description	Name	Description
`exp(x)`	e^x	`log1p(x)`	ln(1+x)
`pow2(x)`	2^x	`log2(x)`	$log_2 x$
`log(x)`	lnx	`log10(x)`	$log_{10} x、lg(x)$

- Array and matrix

Name	Description	Name	Description
`ones`	All one	`zeros`	All zeros
`length`	length	`size`	Dimension of array
`sum`	sum	`mean`	Mean of array
`reshape`	Reshape the array	`sort`	Sort the array
`min`	minimum	`max`	maximum

- Rounding functions
  `round` 、 `fix` 、 `floor` 、 `ceil`
- Figure plotting
  `plot()` 、 `subplot(n,m,i)` 、 `figure(i)` 、 `xlabel()` 、 `ylabel()`
  `title()` 、 `legend()` 、 `hold on` 、 `grid on` 、 `semilogy()` ...
- Function format
  `function [outputs]=function_name(inputs)`

2.Signals and linear systems

- ## Basic sequence

  - Impulse signal and sequence

```
t = -5:0.01:5;
y = (t==0);
subplot(1,2,1);
plot(t, y, 'r’);

n = -5:5;
x = (n==0);
subplot(1,2,2);
stem(n, x);
```

  - Step signal and sequence

```
t =-5:0.01:5;
y=(t>=0);
subplot(1,2,1);
plot(t, y, 'r')

n = -5:5;
x = (n>=0);
subplot(1,2,2);
stem(n, x);
```

- Real exponential sequence
  x(n)=a^n, $\forall n, a \in R$

```
n=[ns:nf];
x=a.^n;
```

- Exponential sequence
  x(n)=$e^{(\delta+j\omega)n}$

```
n=[ns:nf];
x=exp((delta+jw)*n);
```

- Sinine, cosine sequence
  $x(n) = cos(\omega n + \theta)$

```
n=[ns:nf];
x=cos(w*n+sita);
```

- Other signal generation functions

Name	Description	Name	Description
`sawtooth`	Sawtooth or triangle wave	`pulstran`	Pulse train
`square`	Square wave	`rectpule`	A period square wave
`sinc`	Sinc wave	`tripuls`	A period triangle wave

- ## Signal Operations

  - Moving
    y(n)=x(n-m)

```
y(n)=x(n-m)
```

- Periodic extension
  $y(n) = x((n))_M$

```
y(n)=x(mod(n,M)+1)
```

- Flipping
  y(n)=x(-n)

```
y=fliplr(x)
```

- Correlation with two sequences
  $y(m) = \sum^{n} x_1(n+m)x_2^*(n)$

```
y=xcorr(x1,x2)
```

- Cumulative sum
  $y(n) = \sum_{i=1}^{n} x(i)$

```
y=cumsum(x)
```

- Convolution of two sequences
  $y(n) = x_1(n) * x_2(n) = \sum^{m} x_1(m)x_x(n-m)$

```
y=conv(x1,x2)
```

- Convolution of two *continuous-time* signals：
  $f(t) = \int_{-\infty}^{+\infty} f_1(\tau)f_2(t-\tau)d\tau$
  $f(t) = \sum_{k=-\infty}^{+\infty} f_1(k\Delta)f_2(t-k\Delta)\Delta$

```
y=conv(x1,x2)*dt
```

- ## Fourier transform

  - Continuous-time，continuous-frequency: FT

T2F	F2T
$X(f) = \int_{-\infty}^{+\infty} x(t)e^{-j2\pi ft}dt$	$x(t) = \int_{-\infty}^{+\infty} X(f)e^{j2\pi ft}df$

  - Discrete-time, discrete-frequency: DFT / FFT

T2F	F2T
$X(k) = \sum_{n=0}^{N-1} x(n)e^{-j\frac{2pi}{N}nk}$	$x(t) = \frac{1}{N}\sum_{n=0}^{N-1} X(k)e^{j\frac{2pi}{N}nk}$

- # Energy and Power

```
dt=t(2)-t(1);
df=f(2)-f(1);
N=length(t);
T=t(end)-t(1)+dt;   % T=N*dt
```

  - Energy
    - T-domain

      $x[n] = x(n\Delta t)$

      $E = \int_{-\infty}^{+\infty} |x(t)|^2 dt \approx \sum_{n=0}^{M} x[n] \cdot x^*[n] \cdot \Delta t$

```
% E=sum(x.*conj(x))*dt;
E=sum(abs(x).^2)*dt;
```

    - F-domain

      $X[k] = X(k\Delta f)$

      $E = \int_{-\infty}^{+\infty} |X(f)|^2 df \approx \sum_{k=0}^{k-1} X[k] \cdot X^*[k] \cdot \Delta f$

```
% E=sum(x.*conj(x))*df;
E=sum(abs(x).^2)*df;
```

  - Power
    - T-domain

      $P = \lim_{T\to\infty} \frac{1}{T} \int_{0}^{T} |x(t)|^2 dt \approx \frac{1}{N} \sum_{n=0}^{N-1} |x[n]|^2$

```
% P=sum(x.*conj(x))/N;
P=sum(abs(x).^2)/N;
```

    - F-domain

      $X_T f$ *is the spectrum of x(t) within [0,T]*

      $P = \lim_{T\to\infty} \frac{1}{T} \int_{-\infty}^{+\infty} |X_T(f)|^2 df \approx \frac{1}{N} \sum_{k=0}^{K-1} |X[k]|^2 \Delta f$

```
% P=sum(x.*conj(x))*df/T;
P=sum(abs(x).^2)*df/T;
```

- # Autocorrelation and Power spectral density(PSD)

  - Autocorrelation

    $x[n] = x(n\Delta t)$

    $R(\tau) = \int_{-\infty}^{+\infty} x(t)x^*(t+\tau)dt \approx \sum_{n=0}^{M} x[n] \cdot x^*[n+\tau] \cdot \Delta t$

    $R_T(\tau) = \lim_{T\to\infty} \frac{R(\tau)}{T}$

```
R(tau)=sum(x(t).*conj(x(t+tau))*dt/(N*dt);

%% or
R=xcorr(x);
R=R*Ts/T;    % R=R/N
```

- PSD
  $P_T(f) = \lim_{T\to\infty} \frac{|X_T(f)|^2}{T}$
- Wiener-Khinchin theorem
  $FT(R_T(\tau)) = P_T(f)$

- ## Component

  - hilbert change
    通过希尔伯特变换，返回的实部是本身即同向分量(quadrature component)，虚部是延迟90°后的信号即正交分量(in-phase component)

```
x_a = hilbert(x);
fc = 50;                      % Carrier fc=50Hz
x_l = x_a.*exp(-1i*2*pi*fc*t); % Lowpass equivalent
x_i = real(x_l);              % In-phase component
x_q = imag(x_l);              % Quadrature component
```

3.Randon process and analog modulation

For a random process $x(t)$, for an arbitrary t_1, $x(t_1)$ is a random variable.

- ## Variables and Distributions

  Function:$F_x(x) = Pr(X \le x)$
  - Probability density function:$p(x) = \frac{dF_x(x)}{dx}$
  - Expectation:$E[X] = \int_{-\infty}^{+\infty} xp(x)dx$
  - Moment:$E[X^n] = \int_{-\infty}^{+\infty} x^n p(x)dx$
  - Variance:$E[(X - m_x)^2] = E[X^2] - m_x^2$

- ## Ergodicity and Stationary

  - Ergodicity
    > 1.The expectation of x(t) is a constant.
    >
    > `mean(x)=constant`
    >
    > 2.Its autocorrelation only depends on the time difference.

> `Ry is autocorrelation of {Yn}.`
>
> `After 100 times, Ry is has nothing to do with time.`

- Stationary

  > 1.The expectation of x(t) equals the time-average. An arbitrary realization of the random process will go through all the possible states.
  >
  > If one line has 1000 components, use `mean` to calculate expectation.
  >
  > Use the first component of each line, the `mean` of one row is time-average.

- e.g. Modulate and demodulate

- # Analog modulation

  - Amplitude modulation(mainly)

    `m(t)` means the message signal.

    - DSB-Am

Time-domain	Frequency-domain
$s(t) = A_c m(t) cos(2\pi f_c t)$	$S(f) = \frac{A_c}{2}[M(f - f_c) + M(f + f_c)]$

    - Conventional AM

Time-domain	Frequency-domain
$s(t) = A_c(1 + am(t))cos(2\pi f_c t)$	$S(f) = \frac{A_c}{2}[\delta(f - f_c) + aM(f - f_c) + aM(f + f_c)]$

    - SSB-AM(T-domain)

      $s(t) = \frac{A_c}{2} m(t) cos(2\pi f_c t) \pm \frac{A_c}{2} m(t) sin(2\pi f_c t)$

      $\textbf{matlab code:}\ hilbert(m) = m(t) + j\hat{m}(t)$

U_SSB	L_SSB
$s(t) = \frac{A_c}{2} m(t) cos(2\pi f_c t) - \frac{A_c}{2}\hat{m}(t) sin(2\pi f_c t)$	$s(t) = \frac{A_c}{2} m(t) cos(2\pi f_c t) + \frac{A_c}{2}\hat{m}(t) sin(2\pi f_c t)$
$\frac{A_c}{2} Re[(m(t) + j\hat{m}(t))e^{j2\pi f_c t}]$	$\frac{A_c}{2} Re[(m(t) + j\hat{m}(t))e^{-j2\pi f_c t}]$

    - e.g.

```
Fs = 1500;               % Sampling rate
Ts = 1/Fs;               % Sampling interval
Fc = 20;                 % Carrier rate
t = 0:Ts:1;              % Time vector

m = sin(2*pi*t);         % Message signal
c = cos(2*pi*Fc*t);      % Carrier signal
Ac_DSB_AM = 1;
Ac_USSB_AM = sqrt(2);

% DBS_AM modulated signal.
u_dsb = Ac_DSB_AM*m.*c;
% SSB_AM modulated signal.
u_ussb = 1/2*Ac_USSB_AM*real(hilbert(m).*exp(1i*2*pi*Fc*t));
```

4.Baseband signal transmission

AWGN: add white gauss noise
Two optimum receivers for AWGN: signal correlator and matched filter.
For AWGN, the noise N_i is Gaussian distributed with mean of zero and variance of $\frac{EN_0}{2}$.

- ## Binary modulations
  - Received signal
    - $r(t) = s_i(t) + n(t),\ \ 0 \le t \le T_b,\ \ i = 0, 1$
    - Determine whether a 0 or 1 was transmitted.
  - Signal correlator
    - $r_0 = \int_0^{T_b} r(t)s_0(t)dt = E + n_0$
    - $r_1 = \int_0^{T_b} r(t)s_1(t)dt = n_1$
  - Match filter
    - Sample at $t = T_b$
    - $h_i(t) = s_i(T_b - t),\ \ 0 \le t \le T_b,\ \ i = 0, 1$
  - Detector
    - The detector observe the correlator or matched filter output and decides on whether the transmitted signal waveform is either $s_0(t)$ or $s_1(t)$.
    - Comparing r_0 or r_1. If $r_0 > r_1$, it decides 0 is transmitted.
    - $P_e = Q(\frac{E}{\delta}) = Q(\frac{E}{\sqrt{\frac{EN_0}{2}}}) = Q(\sqrt{\frac{2E}{N_0}})$

- ## Other binary modulations

- Antipodal signal
  - $P_e(\alpha_{opt}) = Q(\frac{\frac{E}{2}}{\delta}) = Q(\frac{\frac{E}{2}}{\sqrt{\frac{EN_0}{2}}}) = Q(\sqrt{\frac{E}{2N_0}})$

- ## Monto Carlo simulation
  - Source output `dsource=0` or `dsource=1`
  - Detection
    - Match filter output `r=E+gngauss(sgma)` or `r=-E+gngauss(sgma)`
    - Detector `r<0` ? `decis=0` or `decis=1`
    - Error counter `decis!=dsource` ? `numoferr+=1`

- ## Constellation diagram
  - $x_1 = \sqrt{E} + n_1$
  - $n_1 = \sqrt{\frac{N_0}{2}} * randn(100,1)$
  - $x_0 = \sqrt{E} + n_0$
  - $n_0 = \sqrt{\frac{N_0}{2}} * randn(100,1)$

5.Pulse Amplitude Modulation(PAM)

- ## Theoretical symbol error rate
  - $P_M = \frac{2(M-1)}{M} Q(\sqrt{\frac{6(log_2 M)E_{avb}}{(M^2-1)N_0}})$
  - `SNR=exp(snr_in_dB*log(10)/10)` equals to `SNR=10^(snr_in_dB/10)`

- ## Bit error rate and energy
  - `smld_err_pb=smld_err_p/M`
  - Energy(M-PAM)(N symbols)
    $E_{av} = \frac{1}{N} \sum_{k=1}^{N} \int_0^T s_k^2(t)dt$
    $E = \frac{E_{av}}{M}$

```
%% 8PAM,M=3;there is 8 symbols.
Eav = sum(Am.^2)/8       % Average energy per symbol.
E = Eav/3                % Average energy per bit.
```

- ## Raised-cosine and ISI

- RC
  - $x_{rc}(t) = \frac{sin(\pi t/T_s)}{\pi t/T_s} \frac{cos(\alpha\pi t/T_s))}{1-4\alpha^2 t^2/T_s^2}$
- ISI
  - $x(nT) = \begin{cases} 1, & n = 0 \\ 0, & others \end{cases}$
  - F-domain: $\sum_{m=-\infty}^{+\infty} X(f + \frac{m}{T}) = T$
- Under a band-limited noiseless channel, the larger the passband, the smoother the signal.
  - e.g. Band-limited noiseless channel

6. 💬 Digital transmission via carrier modulation

- ## Carrier amplitude modulation (ASK)

  - In baseband digital PAM, the signal waveforms are: $s_m(t) = A_m g_T(t)$
  - $A_m = (2m - 1 - M)d, m = 1, 2, ..., M$
  - Multiplied by a sinusoidal carrier: $u_m(t) = A_m g_T(t) cos(2\pi f_c t)$
  - When the pulse shape is rectangular: $x(nT) = \begin{cases} \sqrt{\frac{2}{T}}, & 0 \leq t \leq T \\ 0, & otherwise \end{cases}$
  - Usually called amplitude shift keying, which is not bandlimited.

- ## Carrier phase modulation(PSK)

  - The information is impressed on the phase of the carrier.
  - The range of the phase: $0 \leq \theta \leq 2\pi$
  - $\theta_m = \frac{2\pi m}{M}, m = 0, 1, ..., M - 1$
  - Modulated signal waveform: $u_m(t) = A g_T(t) cos(2\pi f_c t + \frac{2\pi m}{M}), m = 0, 1, ..., M - 1$
  - Usually called phase shift keying.
  - e.g. ASK and PSK

- ## Quadrature amplitude modulation(QAM)

  - Two quadrature carriers, $sin(2\pi f_c t)$ and $cos(2\pi f_c t)$
  - Each is modulated by independent information bits.
  - $u_m(t) = A_{mc} g_T(t) cos(2\pi f_c t) + A_{ms} g_T(t) sin(2\pi f_c t), m = 0, 1, ..., M$
  - Carried bits per symbol: $log_2 M$

- Can be viewed as a form of combined amplitude and digital-phase modulation.
- Rewrite: $u_{mn}(t) = A_m g_T(t) cos(2\pi f_c t + \theta_n), m = 1, 2, ..., M_1, n = 1, 2, ..., M_2$
- This time, carried bits per symbol: $log_2 M_1 + log_2 M_2$

Carrier frequency modulation(FSK)

- For channel lack of phase stability, digital transmission by frequency modulation can be applied.
- M-ary FSK can be used to transmit a block of $k = log_2 M$ bits per symbol.
- $u_m(t) = \sqrt{\frac{2E_s}{T}} cos(2\pi f_c t + 2\pi m \Delta f t), m = 0, 1, ..., M-1, 0 \leq t \leq T$
- To guarantee orthogonality, Δf is a multiple of 1/2T.

Sampling

- sampling.m

```
function [T,Samp_Sig]=Sampling(t,Fs,sig)
%Fucntion Name:Sampling
%Input: T,Fs:sig OutPut:Samp_Sig
%When you call the Function ,u input the time for a symbol,the
%Sampling rate and the source signal,then output the Samplint Signal.
Ts=1/Fs;
Sig=sig;
len=length(Sig);
T=0:Ts:len*t-Ts;
Samp_Sig=T;
for i=0:1:len-1
  for j=1:1:t/Ts
      Samp_Sig(i*t/Ts+j)=Sig(i+1);
  end
end
```

🗁 Source file.zip