# Báo cáo Lab 03

Môn: Truyền dữ liệu-NT105.K21.MMCL

Họ tên : Thái Công Sinh

MSSV: 18521340.

Bài 1: Hiện thực các bài thực hành mẫu từ 3.1.3 đến 3.1.8

- **3.1.3: Bipolar -AMI**

+ *Định nghĩa hàm AMI.*

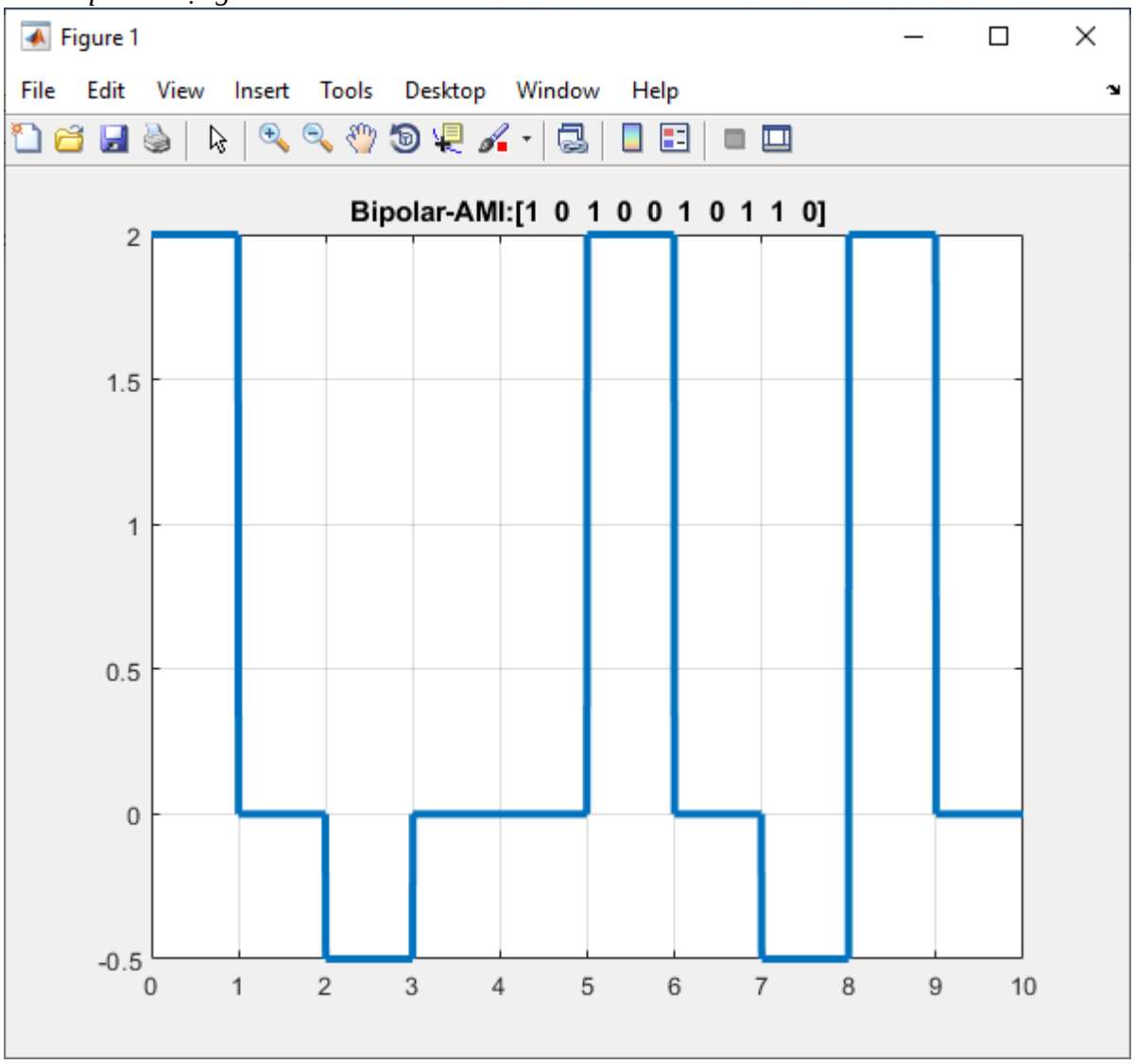
```
function [t,x]=AMI(bits,bitrate)

T=length(bits)/bitrate;
n=200;
N=n*length(bits);
dt=T/N;
t=0:dt:T;
x=zeros(1,length(t));
last=2;
for i=0:length(bits)-1
   if bits(i+1)==1
      x(i*n+1:(i+1)*n) =last;
      if (last==2 )
         last = -0.5;
      else
         if (last==-0.5)
            last=2;
         end
      end
   else
      x(i*n+1:(i+1)*n)=0;
   end
end
```

+ *Sử dụng hàm.*

```
>> bits=[1 0 1 0 0 1 0 1 1 0];
>> bitrate=1;
>> figure;
>> [t,s]=AMI(bits,bitrate);
```

```
>> plot(t,s,'LineWidth',3);
>> axis([0 t(end) -0.5 2])
>> grid on;
>> title (['Bipolar-AMI:[' num2str(bits) ']']);
```

*+ Kết quả sử dụng hàm.*




- **3.1.4: Pseudoternary**

*+ Định nghĩa hàm Pseudo:*

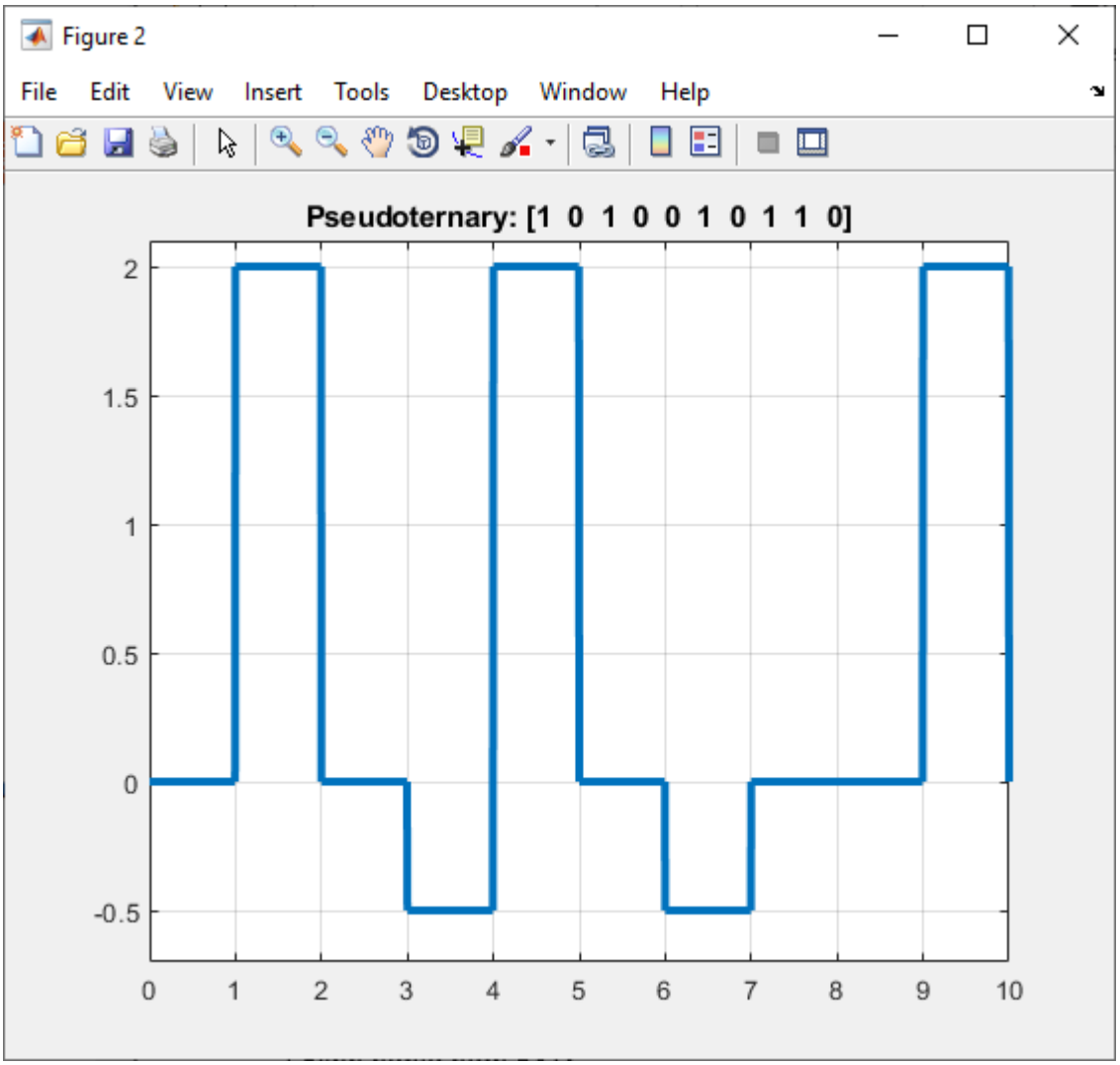
```
function [t,x]=Pseudo(bits, bitrate)
n=200;
T=length(bits)/bitrate;
N=n*length(bits);
dt=T/N;
t=0:dt:T;
```

```
x=zeros(1,length(t));
last=2;
for i=0:length(bits)-1
   if bits(i+1) ==0
      x(i*n+1:(i+1)*n)=last;
      if last==2
         last=-0.5;
      else
         if last==-0.5
            last=2;
         end
      end
   else
      x(i*n+1:(i+1)*n)=0;
   end
end
```

+ *Sử dụng hàm:*

```
>> bits=[1 0 1 0 0 1 0 1 1 0];
>> bitrate=1;
>> figure;
>> [t,s]=Pseudo(bits,bitrate);
>> plot(t,s,'LineWidth',3);
>> axis([0 t(end) -0.5 2])
>> grid on;
>> title(['Pseudoternary: [' num2str(bits) ']']);
```

+ *Kết quả sử dụng hàm:*




- **3.1.5: Manchester**

*+ Định nghĩa hàm Manchester:*

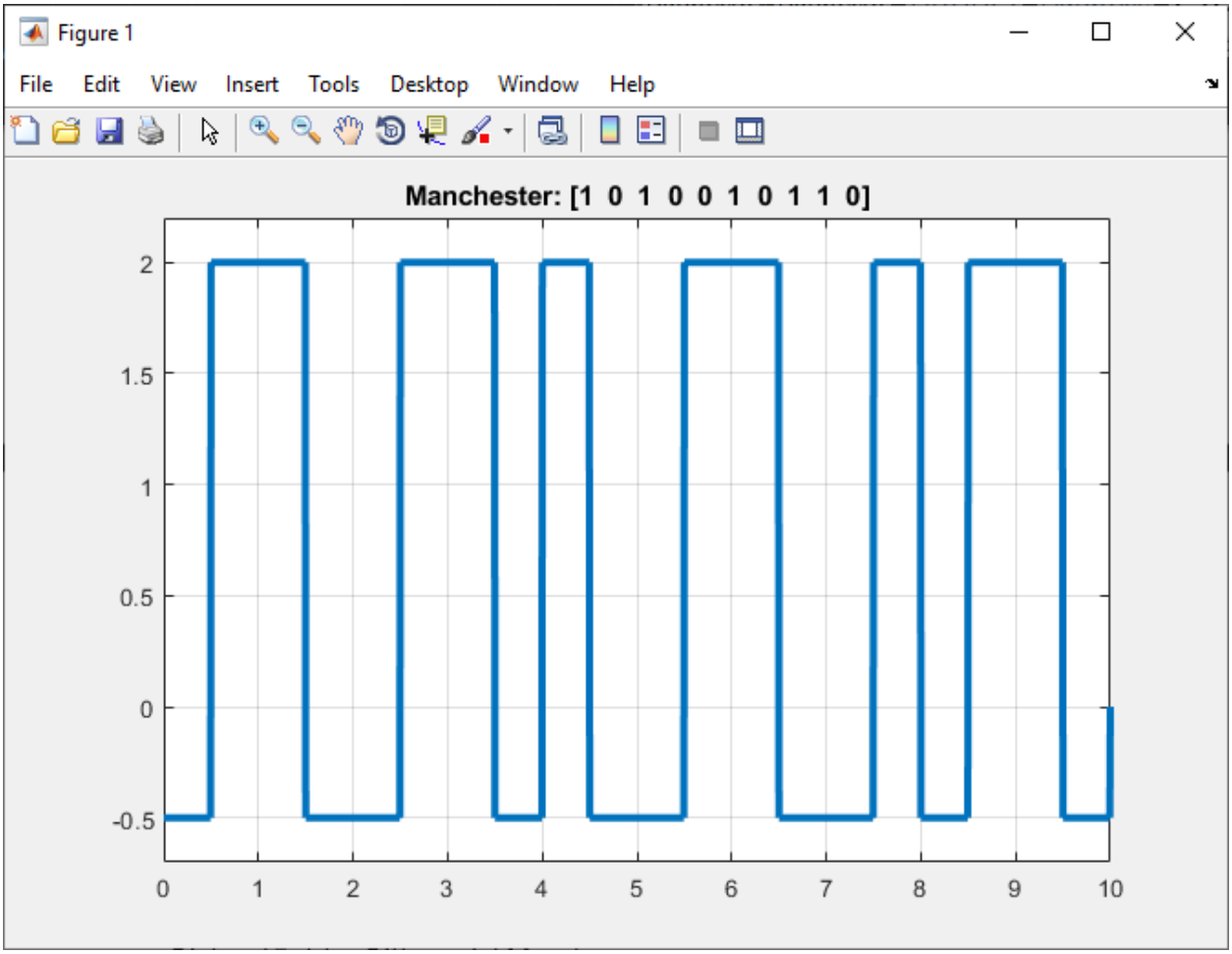
```
function [t,x]=manchester(bits, bitrate)
T=length(bits)/bitrate;
n=200;
N=n*length(bits);
dt=T/N;
t=0:dt:T;
x=zeros(1,length(t));
for i=0:length(bits)-1
   if bits(i+1)==0
      x(i*n+1:(i+0.5)*n)=2;
      x((i+0.5)*n+1:(i+1)*n)=-0.5;
   else
```

```
            x(i*n+1:(i+0.5)*n)=-0.5;
            x((i+0.5)*n+1:(i+1)*n)=2;
        end

    end
```

*+ Sử dụng hàm:*

```
>> bits=[1 0 1 0 0 1 0 1 1 0];
>> bitrate=1;
>> [t,s]=manchester(bits,bitrate);
>> plot(t,s,'LineWidth',3);
>> axis([0 t(end) -0.7 2.2]);
>> grid on;
>> title(['Manchester: [' num2str(bits) ']']);
```

*+ Kết quả sử dụng hàm:*




- **3.1.6: Differential Manchester**

*+ Định nghĩa hàm Differential Manchester:*

```
function [t,x]= DMan(bits, bitrate)
T=length(bits)/bitrate;
n=200;
N=n*length(bits);
dt=T/N;
t=0:dt:T;
x=zeros(1,length(t));
last=[0 0];
for i=0:length(bits)-1
   if i==0
      if bits(i+1)==0
         x(i*n+1:(i+0.5)*n) = 2;
         x((i+0.5)*n+1:(i+1)*n) = -0.5;
         last(1) = 2;
         last(2) = -0.5;
```

```
            else
                x(i*n+1:(i+0.5)*n) = -0.5;
                x((i+0.5)*n+1:(i+1)*n) = 2;
                last(1) = -0.5;
                last(2) =2;
            end
        else
            if bits(i+1)==0
                x(i*n+1:(i+0.5)*n)=last(1);
                x((i+0.5)*n+1:(i+1)*n)=last(2);
            else

                if last(1)==2
                    last(1)=-0.5;
                else
                    if last(1)==-0.5
                        last(1)=2;
                    end
                end

                 if last(2)==2
                        last(2)=-0.5;
                 else
                        if last(2)==-0.5
                            last(2)=2;
                        end
                 end

                x(i*n+1:(i+0.5)*n)=last(1);
                x((i+0.5)*n+1:(i+1)*n)=last(2);
            end
        end


    end
```

*+ Sử dụng hàm:*

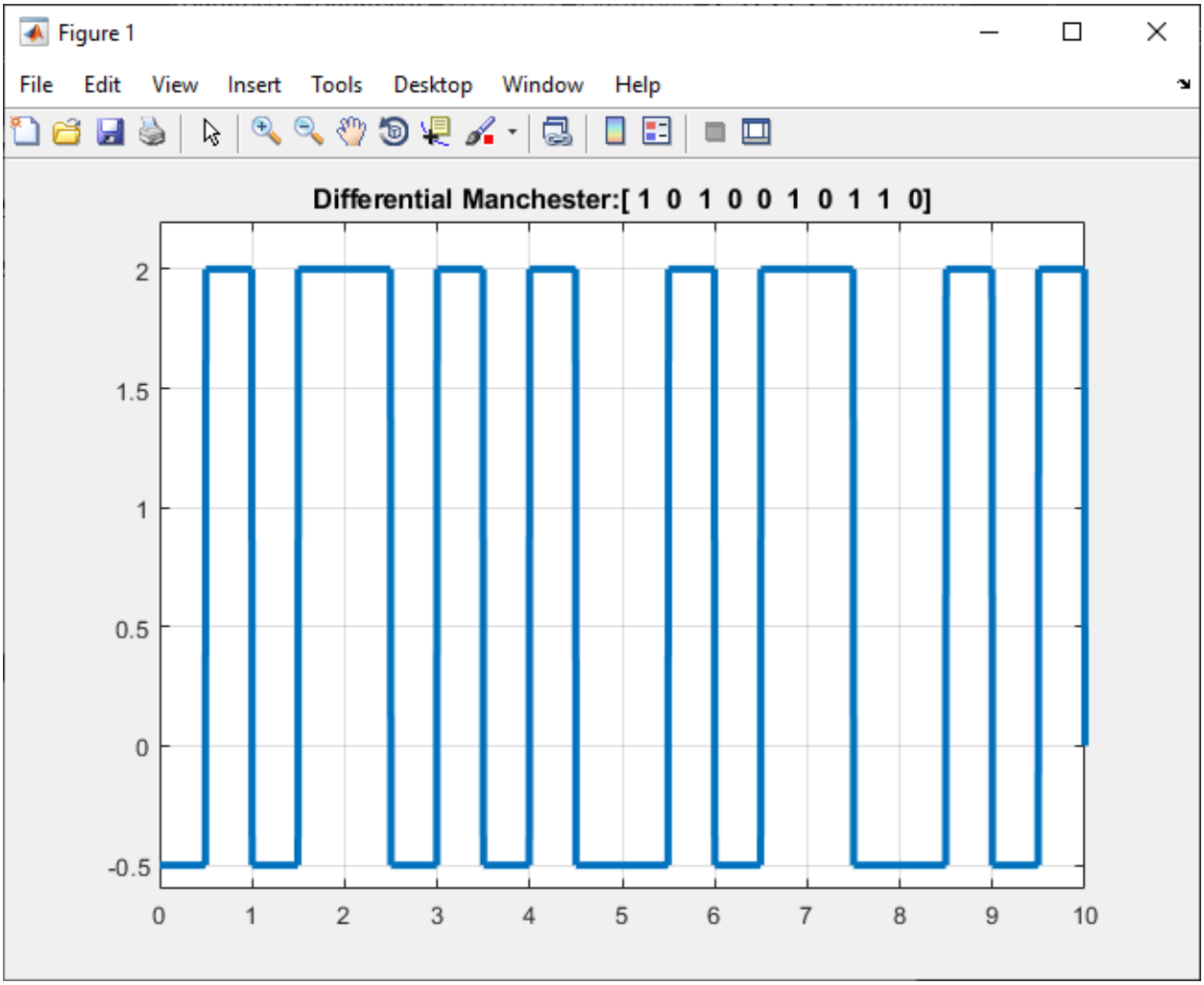
```
>> bits=[1 0 1 0 0 1 0 1 1 0];

>> bitrate =1;

>> [t,s]=DMan(bits,bitrate);

>> plot(t,s,'LineWidth',3)

>> axis([0 t(end) -0.5 2.2]);
```

*>> axis([0 t(end) -0.6 2.2]);*

*>> grid on*

*>> title(['Differential Manchester:[ ' num2str(bits) ']']);*

*+ Kết quả sử dụng hàm:*




- **3.1.7: B8ZS**

*+Định nghĩa hàm B8ZS:*

```
function [t,x]=B8ZS(bits,bitrate)
n=1000;
T=length(bits)/bitrate;
N=n*length(bits);
dt=T/N;
t=0:dt:T;
x=zeros(1,length(t));
```

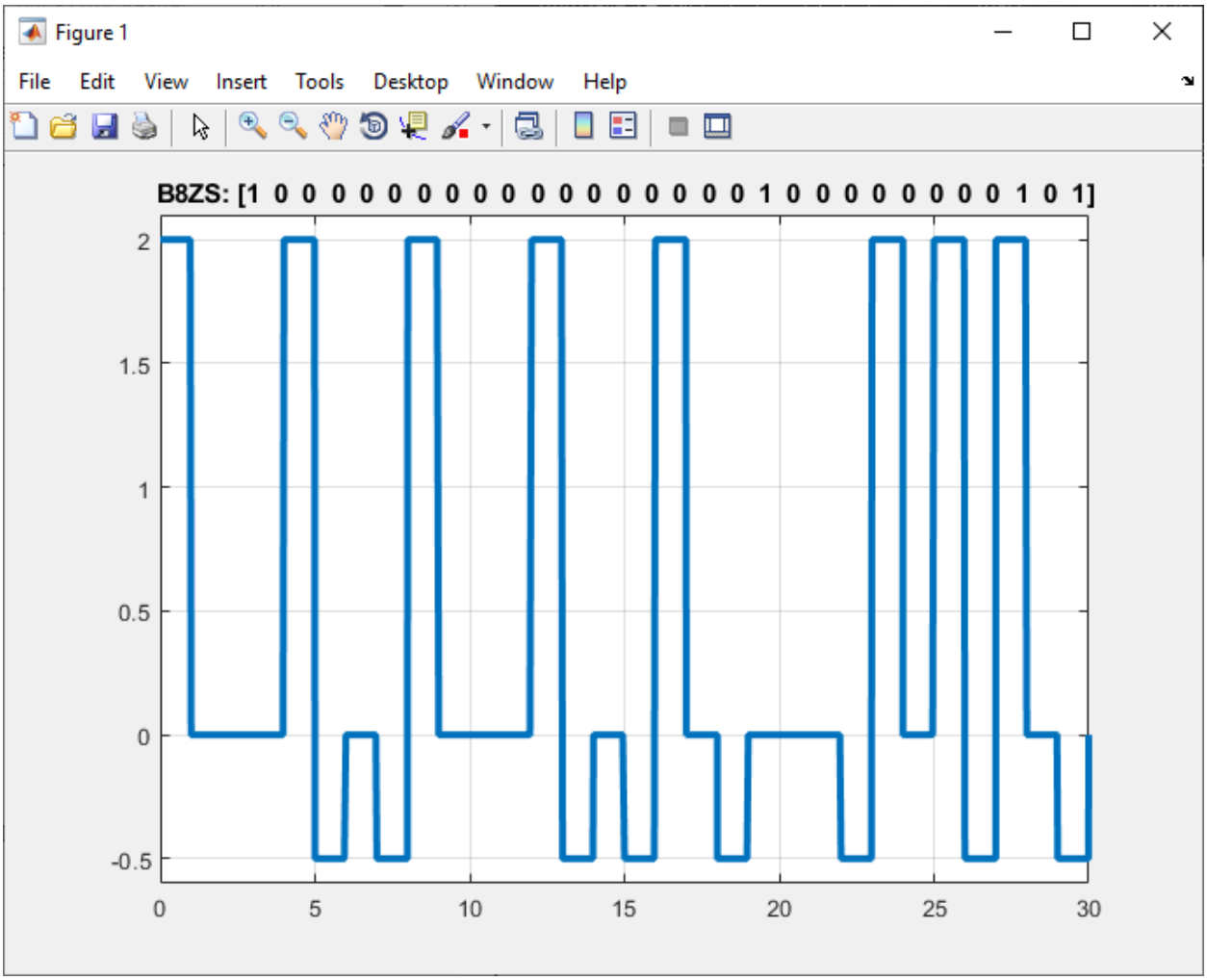
```
counter=0;
lastbit=-0.5;
for i=1:length(bits)
   if bits(i)==0
      counter = counter + 1;
      if counter==8
         x((i-1-7)*n+1:(i-7)*n) = 0;
         x((i-1-6)*n+1:(i-6)*n) = 0;
         x((i-1-5)*n+1:(i-5)*n) = 0;
         x((i-1-4)*n+1:(i-4)*n) = lastbit;
         if lastbit==2
            x((i-1-3)*n+1:(i-3)*n)=-0.5;
         else
            if lastbit==-0.5
            x((i-1-3)*n+1:(i-3)*n)=2;
            end
         end

         lastbit =x((i-1-3)*n+1);

         x((i-1-2)*n+1:(i-2)*n) = 0;
         x((i-1-1)*n+1:(i-1)*n) = lastbit;
         if lastbit==2
             x((i-1)*n+1:i*n)=-0.5;
         else
            if lastbit==-0.5
                x((i-1)*n+1:i*n)=2;
            end
         end

         lastbit =  x((i-1)*n+1);
         counter = 0;
      end
   else
      counter = 0;

      if lastbit==2
         x((i-1)*n+1:i*n)=-0.5;
      else
         if lastbit==-0.5
            x((i-1)*n+1:i*n)=2;
         end
      end

      lastbit = x((i-1)*n+1);
   end
```

```
end
```

*+Sử dụng hàm:*

```
>> bits = [1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 1 0 1];
>>bitrate=1;
>> [t,x]=B8ZS(bits,bitrate);
>> plot (t,x,'LineWidth',3);
>> axis([0 t(end) -0.6 2.1]);
>> grid on
>> title(['B8ZS: [',num2str(bits) ,']']);
```

*+Kết quả:*




- **3.1.8: HDB3**

*+ Định nghĩa hàm HDB3:*

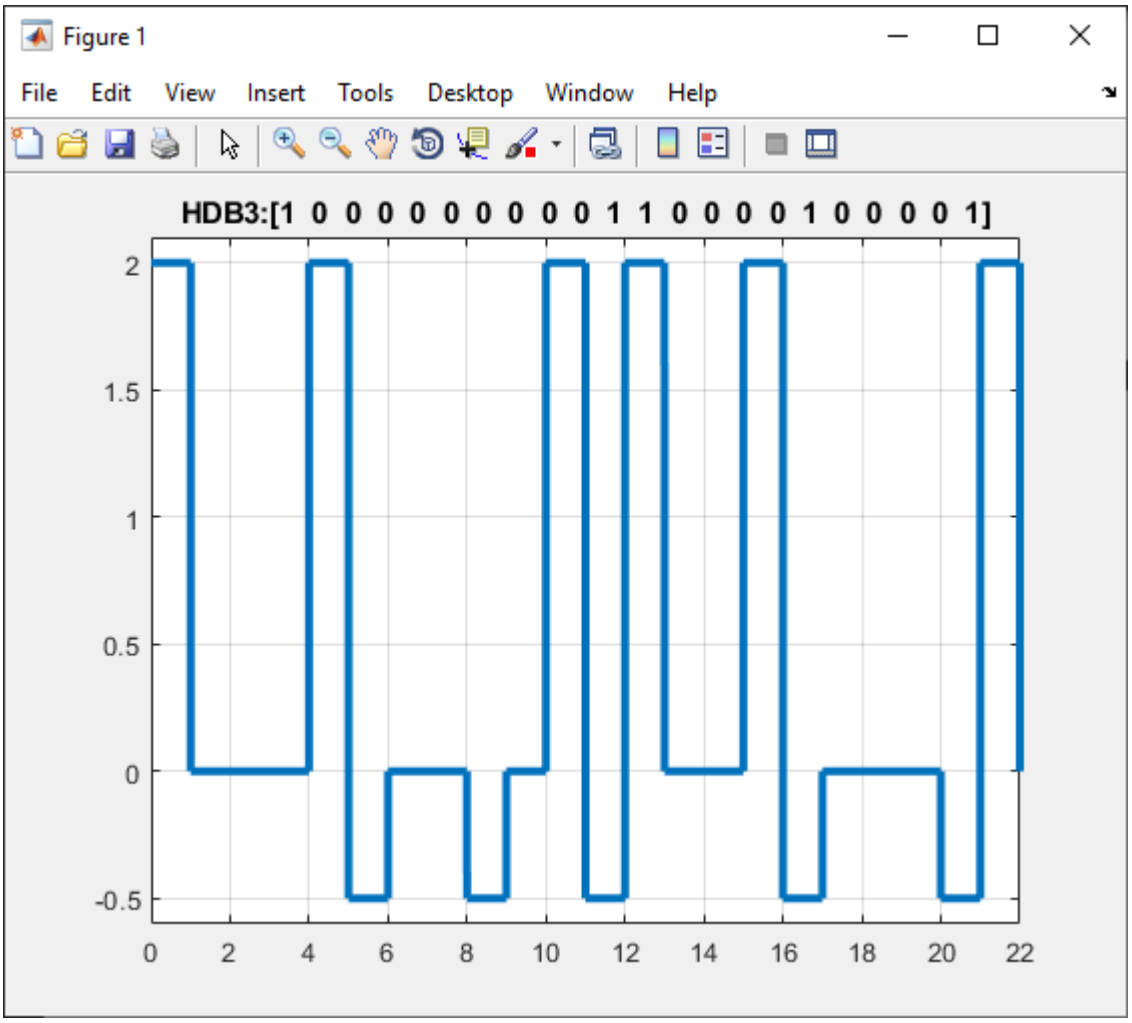
```
function [t,x]=HDB3(bits, bitrate)
n=1000;
T=length(bits)/bitrate;
N=n*length(bits);
dt=T/N;
t=0:dt:T;
x=zeros(1,length(t));
counter=0;
counter1=0;
lastbit=2;
for i=1:length(bits)
   if bits(i)==0
      counter=counter+1;
      if counter==4
         if mod(counter1,2)==0
            x((i-1-3)*n+1:(i-3)*n) = lastbit;
            x((i-1-2)*n+1:(i-2)*n) = 0;
            x((i-1-1)*n+1:(i-1)*n) = 0;
            x((i-1)*n+1:i*n) = lastbit;
            if lastbit==2
               lastbit = -0.5;
            else
               if lastbit==-0.5
                  lastbit=2;
               end
            end

            counter=0;
            counter1=0;
         else
            x((i-1-3)*n+1:(i-3)*n) = 0;
            x((i-1-2)*n+1:(i-2)*n) = 0;
            x((i-1-1)*n+1:(i-1)*n) = 0;

              if lastbit==2
                  lastbit = -0.5;
               else
                  if lastbit==-0.5
                     lastbit=2;
                  end
              end

            x((i-1)*n+1:i*n)=lastbit;

              if lastbit==2
```

```
                    lastbit = -0.5;
                else
                    if lastbit==-0.5
                        lastbit=2;
                    end
                end
                counter =0;
                counter1=0;
            end
        end
    else
        counter=0;
        counter1=counter1+1;
        x((i-1)*n+1:i*n)=lastbit;
        if lastbit==2
            lastbit = -0.5;
        else
            if lastbit==-0.5
                lastbit=2;
            end
        end
    end

end
```

*+Sử dụng hàm:*

```
>> bits = [1 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 1 0 0 0 0 1];

>> bitrate=1;

>> [t,x]=HDB3(bits, bitrate);

>> plot(t,x,'LineWidth',3);

>> axis([0 t(end) -0.6 2.1]);

>> grid on

>> title (['HDB3:[' num2str(bits) ']']);
```

*+ Kết quả:*




Bài 2: Phân biệt sự khác nhau giữa kĩ thuật mã hóa NRZ -L và NRZI:

| NRZ-L | NRZI |
|---|---|
| – Điện áp không thay đổi khi không có sự thay đổi tín hiệu<br>– Điện áp thay đổi khi có sự thay đổi điện áp (từ 0->1 hoặc từ 1->0) | – Dữ liệu được mã hóa dựa vào việc có hay không sự thay đổi đầu thời khoảng bit.<br>– Bit 1: có thay đổi điện áp.<br>– Bit 0: không thay đổi điện áp. |

Bài 3: Phân tích ưu nhược điểm của kỹ thuật mã hóa B8ZS và HDB3:

| | B8ZS | HDB3 |
|---|---|---|
| Ưu điểm | Đồng bộ tốt các bit 0 liên tiếp<br>Thành phần DC bằng 0 | Không bị mất tín hiệu đồng bộ<br>Đảm bảo tính trong suốt |

| | | |
|---|---|---|
| Nhược điểm | Mã hóa và giải mã phức tạp<br>Khó khăn trong việc đồng bộ khi số lượng các bit 0 liên tiếp không phải là bội số của 8. | Mã hóa và giải mã phức tạp |