**1**

Polar NRZ-L assuming bit rate is 4 kbps.

**Code and Simulations:**

% polar NRZ-L

% 22-47048-1

% AB-CDEFG-H

clc

clear all

close all

bit\_stream = [0 0 0 0 0 1 0 0 1 0 0 0];

no\_bits = length(bit\_stream);

bit\_rate = 4000; % 1 kbps

pulse\_per\_bit = 1; % for unipolar nrz

pulse\_duration = 1/((pulse\_per\_bit)\*(bit\_rate));

no\_pulses = no\_bits\*pulse\_per\_bit;

samples\_per\_pulse = 500;

fs = (samples\_per\_pulse)/(pulse\_duration); %sampling frequency

% including pulse duration in sampling frequency

% ensures having enough samples in each pulse

t = 0:1/fs:(no\_pulses)\*(pulse\_duration); % sampling interval

% total duration = (no\_pulse)\*(pulse\_duration)

no\_samples = length(t); % total number of samples

dig\_sig = zeros(1,no\_samples);

max\_voltage = 5;

min\_voltage = -5;

for i = 1:no\_bits

if bit\_stream(i) == 0

dig\_sig(((i-1)\*(samples\_per\_pulse)+1):i\*(samples\_per\_pulse)) = max\_voltage\*ones(1,samples\_per\_pulse);

else

dig\_sig(((i-1)\*(samples\_per\_pulse)+1):i\*(samples\_per\_pulse)) = min\_voltage\*ones(1,samples\_per\_pulse);

end

end

plot(t,dig\_sig,'linewidth',1.5)

grid on

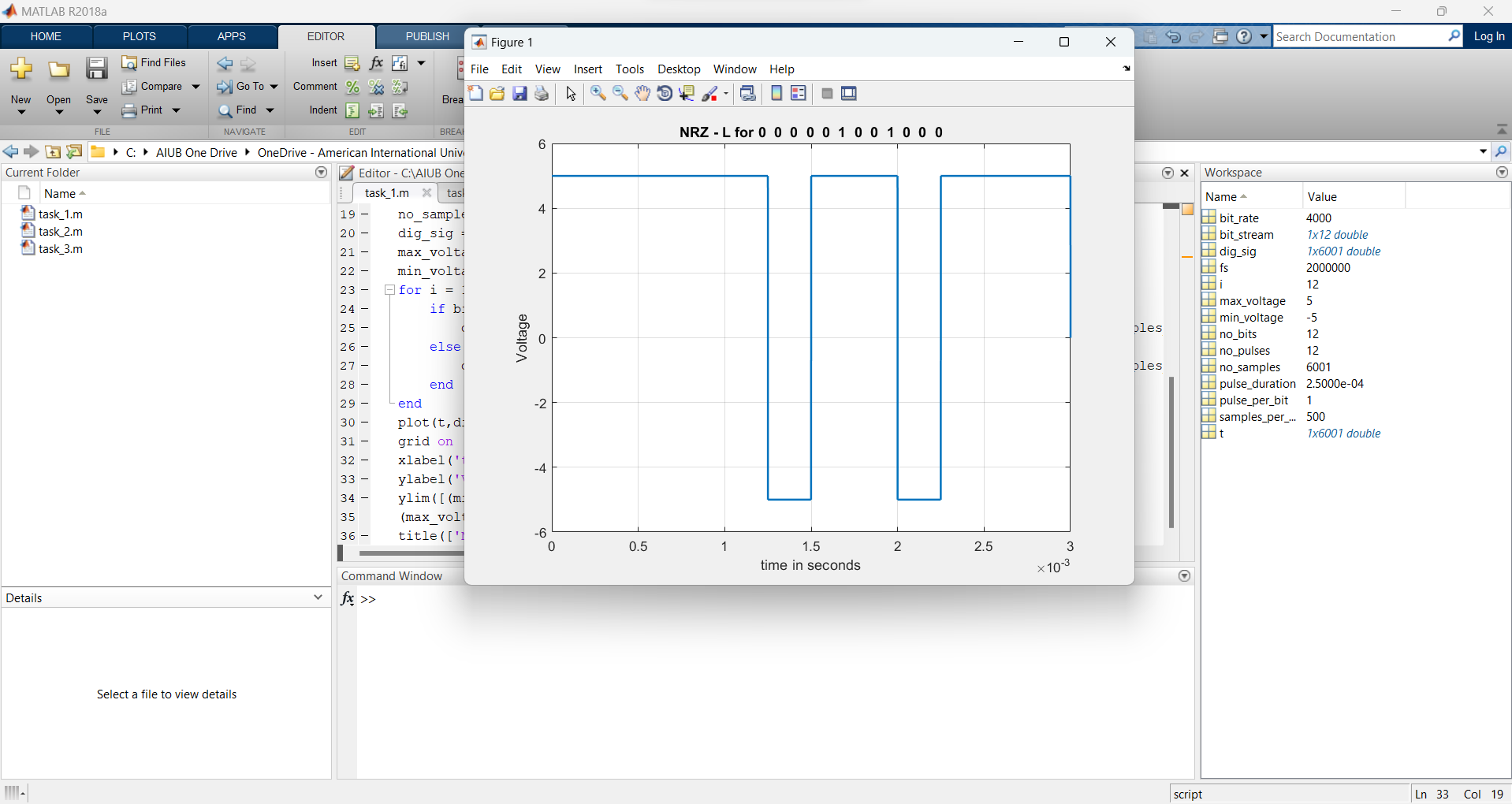
xlabel('time in seconds')

ylabel('Voltage')

ylim([(min\_voltage - (max\_voltage)\*0.2)

(max\_voltage+max\_voltage\*0.2)])

title(['NRZ - L for ',num2str(bit\_stream),''])



**2**

Manchester assuming bit rate is 2 kbps.

**Code and Simulations:**

% Manchester

% 22-47048-1

% AB-CDEFG-H

clc

clear all

close all

bit\_stream = [0 0 0 0 0 1 0 0 1 0 0 0];

no\_bits = length(bit\_stream);

bit\_rate = 2000; % 1 kbps

pulse\_per\_bit = 2; % for unipolar rz

pulse\_duration = 1/((pulse\_per\_bit)\*(bit\_rate));

no\_pulses = no\_bits\*pulse\_per\_bit;

samples\_per\_pulse = 500;

fs = (samples\_per\_pulse)/(pulse\_duration); %sampling frequency

% including pulse duration in sampling frequency

% ensures having enough samples in each pulse

t = 0:1/fs:(no\_pulses)\*(pulse\_duration); % sampling interval

% total duration = (no\_pulse)\*(pulse\_duration)

no\_samples = length(t); % total number of samples

dig\_sig = zeros(1,no\_samples);

max\_voltage = 5;

min\_voltage = -5;

for i = 1:no\_bits

j = (i-1)\*2;

if bit\_stream(i) == 1

dig\_sig((j\*(samples\_per\_pulse)+1):(j+1)\*(samples\_per\_pulse)) = min\_voltage\*ones(1,samples\_per\_pulse);

dig\_sig(((j+1)\*(samples\_per\_pulse)+1):(j+2)\*(samples\_per\_pulse)) = max\_voltage\*ones(1,samples\_per\_pulse);

else

dig\_sig((j\*(samples\_per\_pulse)+1):(j+1)\*(samples\_per\_pulse)) = max\_voltage\*ones(1,samples\_per\_pulse);

dig\_sig(((j+1)\*(samples\_per\_pulse)+1):(j+2)\*(samples\_per\_pulse)) = min\_voltage\*ones(1,samples\_per\_pulse);

end

end

plot(t,dig\_sig,'linewidth',1.5)

grid on

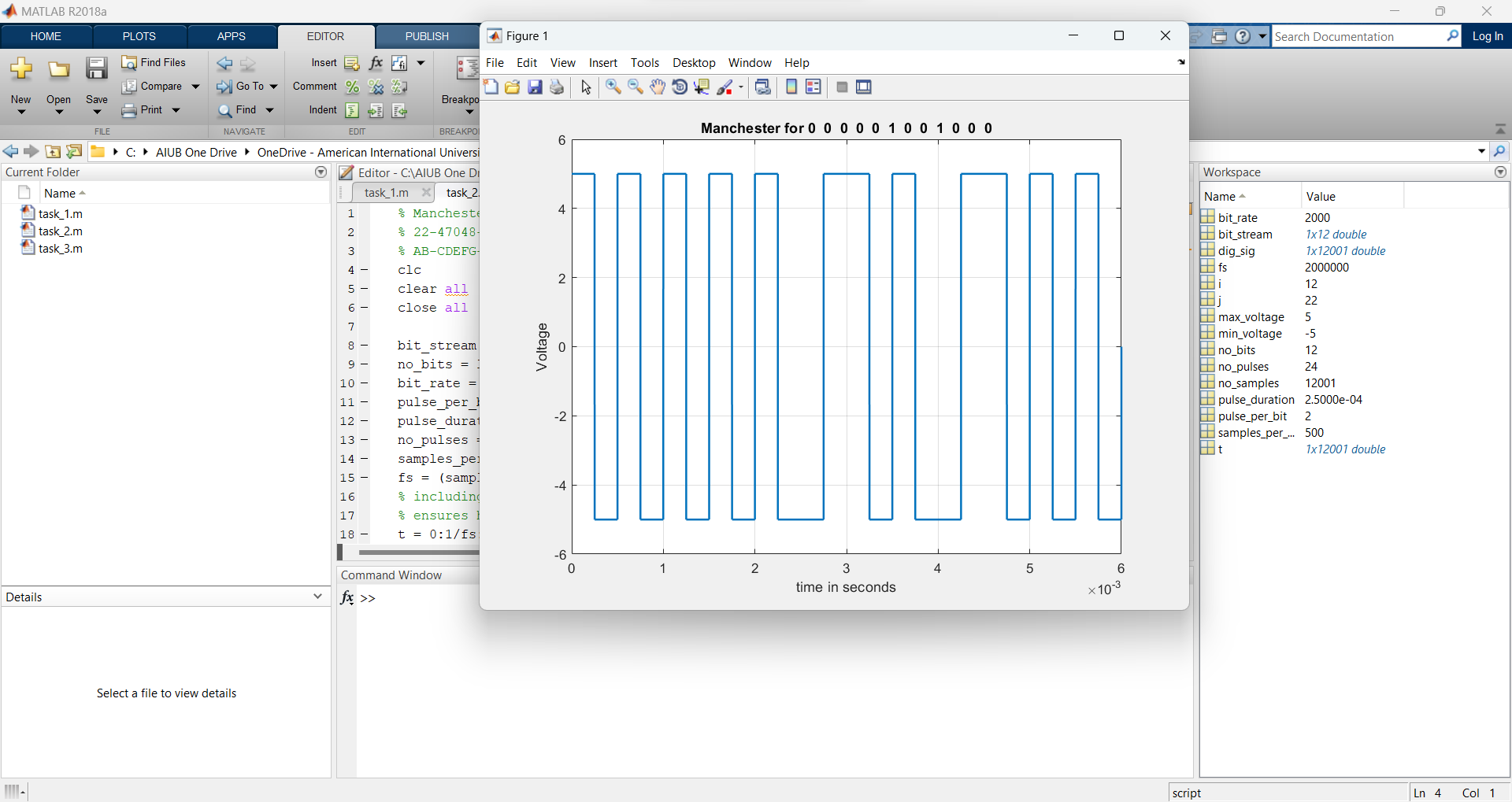
xlabel('time in seconds')

ylabel('Voltage')

ylim([(min\_voltage - (max\_voltage)\*0.2)

(max\_voltage+max\_voltage\*0.2)])

title(['Manchester for ',num2str(bit\_stream),''])



**3**

AMI assuming bit rate is 5 kbps.

**Code and Simulations:**

% AMI

% 22-47048-1

% AB-CDEFG-H

clc

clear all

close all

bit\_stream = [0 0 0 0 0 1 0 0 1 0 0 0];

no\_bits = length(bit\_stream);

bit\_rate = 5000; % 1 kbps

pulse\_per\_bit = 1; % for unipolar nrz

pulse\_duration = 1/((pulse\_per\_bit)\*(bit\_rate));

no\_pulses = no\_bits\*pulse\_per\_bit;

samples\_per\_pulse = 500;

fs = (samples\_per\_pulse)/(pulse\_duration); %sampling frequency

% including pulse duration in sampling frequency

% ensures having enough samples in each pulse

t = 0:1/fs:(no\_pulses)\*(pulse\_duration); % sampling interval

% total duration = (no\_pulse)\*(pulse\_duration)

no\_samples = length(t); % total number of samples

dig\_sig = zeros(1,no\_samples);

max\_voltage = 5;

min\_voltage = 0;

sign = 1;

for i = 1:no\_bits

if bit\_stream(i) == 1

dig\_sig(((i-1)\*(samples\_per\_pulse)+1):i\*(samples\_per\_pulse)) = sign\*max\_voltage\*ones(1,samples\_per\_pulse);

sign = (-1)\*sign;

else

dig\_sig(((i-1)\*(samples\_per\_pulse)+1):i\*(samples\_per\_pulse)) = min\_voltage\*ones(1,samples\_per\_pulse);

end

end

plot(t,dig\_sig,'linewidth',1.5)

grid on

xlabel('time in seconds')

ylabel('Voltage')

% ylim([(min\_voltage - (max\_voltage)\*0.2)

% (max\_voltage+max\_voltage\*0.2)])

title(['AMI for ',num2str(bit\_stream),''])

A screenshot of a computer

Description automatically generated