



โครงการการสร้าง GUI Application  
โดยใช้ MATLAB App Designer  
เรื่อง การจำลองการทำงานของ graph generator  
ด้วยโปรแกรม MATLAB App Designer

รายวิชาศึกษาทั่วไป

รหัสวิชา 010123106 Intro to Signals and Systems

จัดทำโดย

นางสาวจิตฤดี ดวงดี รหัสนักศึกษา 6301012620022

นางสาวฐิตานันท์ มหาพรชัย รหัสนักศึกษา 6301012630036

นางสาวพรหมพร จูติภูมิเดชา รหัสนักศึกษา 6301012630133

เสนอ

ดร.เรวัต ศิริโกคาภิรมย์

สาขาวิชาวิศวกรรมคอมพิวเตอร์

ภาควิชาวิศวกรรมไฟฟ้าและคอมพิวเตอร์

คณะวิศวกรรมศาสตร์

พ.ศ. 2564

## บทสรุป

โครงการการสร้าง GUI Application โดยใช้ MATLAB App Designer เรื่อง การจำลองการทำงานของ graph generator ด้วย MATLAB App Designer ได้จัดทำขึ้นเพื่อศึกษาวิธีการสร้างกราฟแบบต่าง ๆ ด้วย MATLAB App Designer โดยสามารถเลือกค่าแอมพลิจูด ความถี่ รวมทั้งสามารถปรับค่าเฟส offset, Time/Div และ Volt/Div ได้

จากการสร้าง Application การจำลองการทำงานของ graph generator ด้วยโปรแกรม MATLAB App Designer ผู้ใช้งานแอปพลิเคชันสามารถปรับค่าแอมพลิจูด, ความถี่, Phase(Lagging/Leading), offset, time/div, volt/div และ plot กราฟให้อยู่ในรูปของ Square wave , Triangular wave , Sine wave , Sawtooth wave และ Full wave โดยคณะผู้จัดทำได้ทำการตรวจสอบความถูกต้อง และพบว่า Application การจำลองการทำงานของ graph generator สามารถทำงานได้อย่างถูกต้องสมบูรณ์

คณะผู้จัดทำ

ทำ

16 พฤศจิกายน 2564

## สารบัญ

บทสรุป	2
สารบัญ	3
บทที่ 1	4
1.1 ที่มาและความสำคัญ	4
1.2 วัตถุประสงค์ของกิจกรรม	4
1.3 ขอบเขตของการดำเนินการ	5
1.4 ประโยชน์ที่ได้รับจากกิจกรรม	5
บทที่ 2	6
2.1 เอกสารและความรู้ที่เกี่ยวข้อง	6
บทที่ 3	9
3.1 วิธีดำเนินการ	9
3.2 การวิเคราะห์ สร้างหรือออกแบบ	10
3.3 การประเมินผล	12
บทที่ 4	19
4.1 ผลการดำเนินการ	19
บทที่ 5	26
5.1 สรุปผลกิจกรรม อภิปรายผล ปัญหาและอุปสรรค และข้อเสนอแนะ	26
5.2 ปัญหาและอุปสรรค	27
5.3 ข้อเสนอแนะ	28
อ้างอิง	29

## บทที่ 1

### 1.1 ที่มาและความสำคัญ

เนื่องจากรายวิชา Intro to Signals and Systems มีหัวข้อ Periodic Signals and Fourier Series Analysis คณะผู้จัดทำจึงเล็งเห็นความสำคัญของหัวข้อดังกล่าว และตัดสินใจนำมาประยุกต์ใช้กับโปรแกรม MATLAB App Designer โดยจัดทำเป็นการจำลองการทำงานของ graph generator ด้วยโปรแกรม MATLAB App Designer

### 1.2 วัตถุประสงค์ของกิจกรรม

- 1.2.1 สร้าง GUI Application โดยใช้ MATLAB App Designer
- 1.2.2 ใช้งาน UI Components ชนิดต่างๆ ในการสร้าง GUI Application
- 1.2.3 ประยุกต์ใช้งานเนื้อหาที่ได้ศึกษาในรายวิชา กับการออกแบบและสร้าง GUI Application

### 1.3 ขอบเขตของการดำเนินการ

1.3.1 สร้าง GUI Application ภายในระยะเวลา 6 สัปดาห์

1.3.2 สร้าง GUI Application โดยประยุกต์ใช้งานเนื้อหาที่ได้ศึกษาในรายวิชา

Intro to Signals and Systems

1.3.3 Application สามารถ plot กราฟให้อยู่ในรูปของ Square wave, Triangular wave, Sine wave, Sawtooth wave และ Full wave

1.3.4 แอปพลิเคชันสามารถปรับค่าแอมพลิจูด, ความถี่, Phase(Lagging/Leading), offset, time/div และ volt/div

### 1.4 ประโยชน์ที่ได้รับจากกิจกรรม

1.4.1 ได้ศึกษาการสร้าง GUI Application โดยใช้ MATLAB App Designer

1.4.2 ได้ศึกษาการใช้งาน UI Components ชนิดต่าง ๆ

1.4.3 ได้ศึกษาการเขียนโค้ดแบบ OOP ด้วย MATLAB

1.4.4 ได้ออกแบบและสร้าง GUI Application เพื่อการประยุกต์ใช้งาน

1.4.5 ได้เรียนรู้การทำงานเป็นทีม

## บทที่ 2

### 2.1 เอกสารและความรู้ที่เกี่ยวข้อง

#### 2.1.1 สมการของคลื่นชนิดต่างๆ

##### 2.1.1.1 Square wave

$$X(t) = 4a \sin(k\Omega t + p\pi) / (\pi k);$$

- a controls the amplitude.
- p controls phase.

##### 2.1.1.2 Sine wave

$$f(x) = \pm a \cdot \sin(b(x+c)) + d$$

- a controls the amplitude.
- d controls the vertical shift.
- b controls the horizontal stretch.

##### 2.1.1.3 Triangular wave

$$X(t) = a/2 + ((-4a) \cos(k\Omega t + p\pi)) / (k\pi)^2$$

- a controls the amplitude.
- p controls phase.

##### 2.1.1.4 Sawtooth

$$X(t) = a/2 + (-a \sin(p\pi + k\Omega t)) / (\pi k)$$

- a controls the amplitude.
- p controls phase.

#### 2.1.1.5 Full wave

$$X(t) = 2a/\pi + (-4a\cos(k*\Omega t + p*k))/(-4\pi*(k^2))$$

- a controls the amplitude.
- p controls phase.

#### 2.1.2 Time/Div (Time/Division)

ใช้สำหรับเลือกเวลาการกวาด

ทำหน้าที่ขยายสัญญาณที่วัดเข้ามาโดยสามารถปรับคาบเวลา

#### 2.1.3 Volt/Div (Volts/Division)

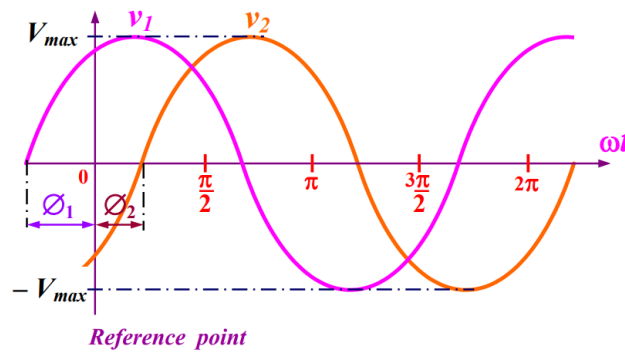
ใช้สำหรับปรับขนาด (แอมพลิจูด) ของสัญญาณทางแนวตั้ง

ทำหน้าที่ขยายสัญญาณที่วัดเข้ามาโดย สามารถปรับขนาดแรงดัน

#### 2.1.4 Offset

หากแอมพลิจูดเฉลี่ยของคลื่นเป็นศูนย์ คลื่นจะเกิดขึ้นในลักษณะที่แกน X มีค่าเป็นศูนย์สำหรับการกำหนดพิกัด (ค่าแกน Y) อย่างไรก็ตาม รูปคลื่นบางรูปถูกสร้างขึ้นเหนือแกน X นั้นเป็นเพราะแอมพลิจูดเฉลี่ยไม่ใช่ศูนย์ แต่มีค่ามากกว่าหรือน้อยกว่าศูนย์ เงื่อนไขนี้เรียกว่า DC offset

## 2.1.5 Phase Shift



การเลื่อนกราฟไปทางซ้ายหรือขวาตามหน่วยองศา โดย Lagging จะเลื่อนกราฟไปขวา Leading จะเลื่อนกราฟไปซ้าย



## บทที่ 3

### 3.1 วิธีดำเนินการ

#### 3.1.1 ขั้นตอนในการดำเนินการ

3.1.1.1 ศึกษาเนื้อหาในรายวิชา ที่ต้องการนำมาประยุกต์ใช้ในการสร้าง GUI Application

3.1.1.2 ออกแบบหน้าต่าง GUI Application ที่ต้องการ และรวมถึงเลือก UI Components ที่ต้องการใช้ตามส่วนต่าง ๆ ด้วย

3.1.1.3 ศึกษาภาษาที่ใช้บน MATLAB App Designer

3.1.1.4 ดำเนินการสร้าง GUI Application การจำลองการทำงานของ graph generator

3.1.1.5 ตรวจสอบความถูกต้องของ Application ด้วยการนำผลลัพธ์ที่ได้จาก Application มาเปรียบเทียบกับเอกสารในรายวิชา Intro to Signals and Systems

#### 3.1.2 เครื่องมือที่ใช้

3.1.2.1 MATLAB App Designer

## 3.2 การวิเคราะห์ สร้างหรือออกแบบ

### 3.2.1 Waveform

เป็นการเลือกรูปแบบของกราฟโดยจะใช้ Dropdown menu เป็นส่วนช่วยในการแสดงข้อมูลของกราฟที่โปรแกรมสามารถสร้างออกมาได้ ซึ่งจะแบ่งเป็น Square wave, Triangular wave, Sine wave, Sawtooth wave และ Full wave

### 3.2.2 Phase

เลือกใช้เป็น Dropdown menu เพื่อให้ผู้ใช้งานเลือกกว่าต้องการให้โปรแกรมแสดงกราฟออกมาในรูปแบบของ Lagging หรือ Leading โดยช่องกรอกตัวเลขสามารถกรอกได้ตั้งแต่เลข 0 จนถึง infinity

### 3.2.3 Offset

เป็นช่องกรอกตัวเลขซึ่งมีไว้กำหนดการแสดงผลการขึ้นลงของกราฟ

### 3.2.4 Amplitude

เป็นช่องกรอกตัวเลขซึ่งมีไว้กำหนดแอมพลิจูดของกราฟ

### 3.2.5 Frequency

เป็นการใส่ความถี่ให้แก่กราฟ โดยจะมีช่องใส่ค่าตัวเลขและเลือกใช้ Dropdown menu เพื่อมีส่วนช่วยในการเลือกหน่วยที่ต้องการจะใช้งาน ซึ่งหน่วยดังกล่าวแบ่งออกเป็น x1 Hz, x10 Hz, x100 Hz, x1k Hz และ x10k Hz

### 3.2.6 Time/Div

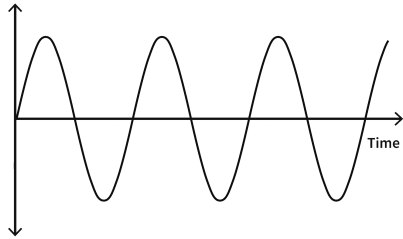
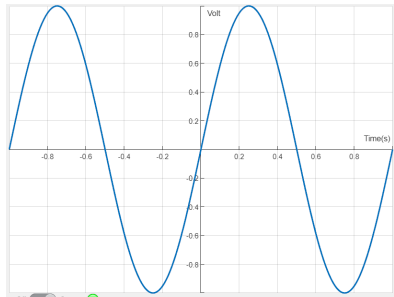
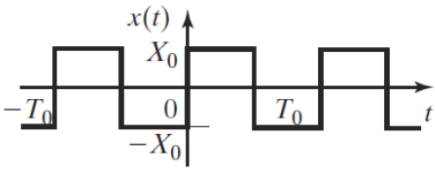
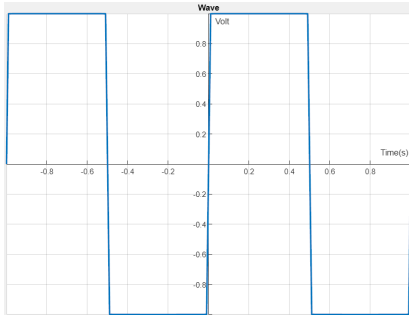
เป็นการปรับขนาดสเกลของเวลา ซึ่งมีการเลือกใช้ Knob เพื่อช่วยในการเลือกค่าที่ต้องการโดยจะมีค่าได้ตั้งแต่ 1 ถึง 10 และบางครั้งการปรับการใช้งานโดยการใช้ Knob อาจไม่สามารถแสดงค่าที่ผู้ใช้ต้องการได้อย่างแม่นยำ จึงตัดสินใจเพิ่มช่องในการใส่ค่าตัวเลขลงไปด้วยเช่นกัน และยังมีการเลือกใช้ Dropdown menu เพื่อให้ผู้ใช้สามารถเลือกหน่วยและขนาดของ Time/Div ที่ต้องการได้ โดยจะมีตัวเลือกดังต่อไปนี้ s, x1 ms, x10 ms, x100 ms, x10 us และ x100 us

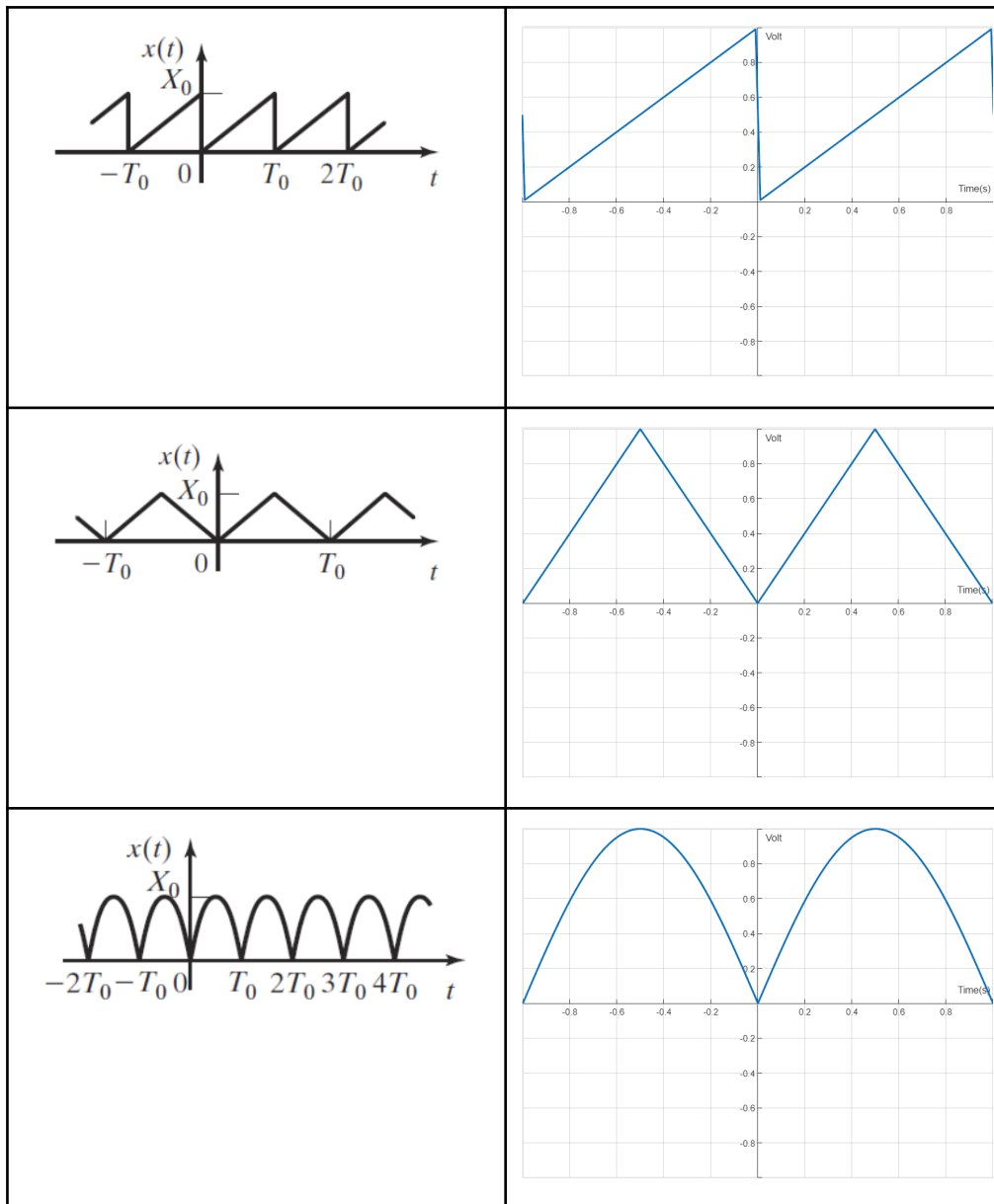
### 3.2.7 Volt/Div

มีการเลือกใช้ Knob เพื่อใช้ปรับขนาดของสเกลวัดแรงดัน โดยสามารถเลือกได้ตั้ง 1 ถึง 10 V และบางครั้งการปรับการใช้งานโดยใช้ Knob อาจไม่สามารถแสดงค่าที่ผู้ใช้ต้องการได้อย่างแม่นยำ จึงตัดสินใจเพิ่มช่องในการใส่ค่าตัวเลขลงไปด้วยเช่นกัน และยังมีให้เลือกใช้ Dropdown menu เพื่อให้ผู้ใช้สามารถเลือกหน่วยและขนาดของ Volt/Div ที่ต้องการได้ โดยจะมีตัวเลือกดังต่อไปนี้  $\times 1$  V,  $\times 10$  mV และ  $\times 100$  mV

## 3.3 การประเมินผล

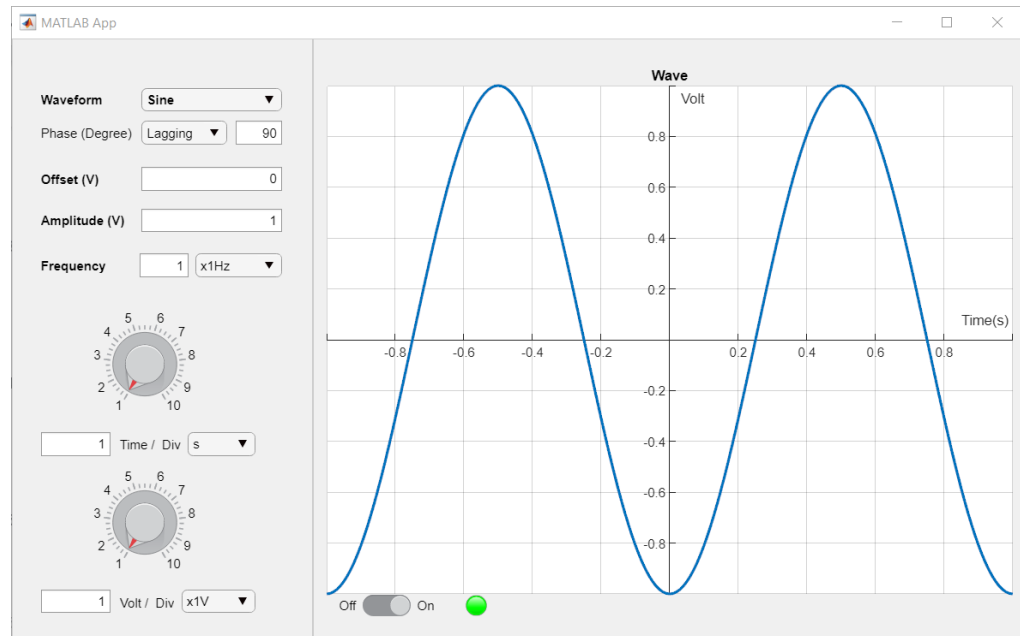
3.3.1 ประเมินผลโดยการเปรียบเทียบลักษณะของคลื่น โดยจะอ้างอิงจากเอกสารการเรียนรู้รายวิชา Intro to Signals and Systems

คลื่นที่อ้างอิง	คลื่นที่ได้จาก Application
	
	

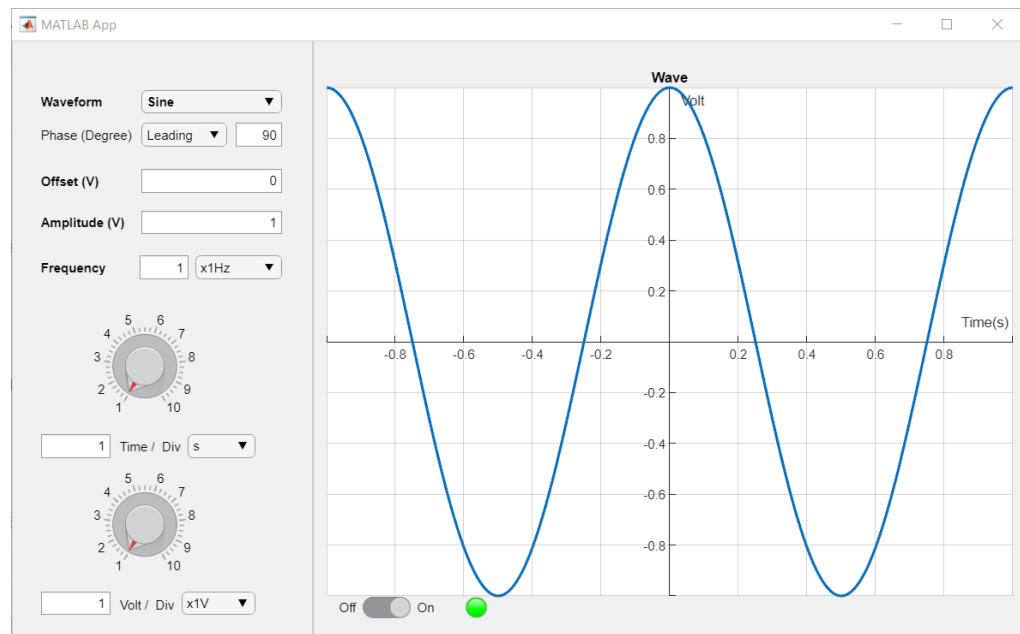


### 3.3.2 ประเมินผลโดยการลองปรับค่าต่างๆ

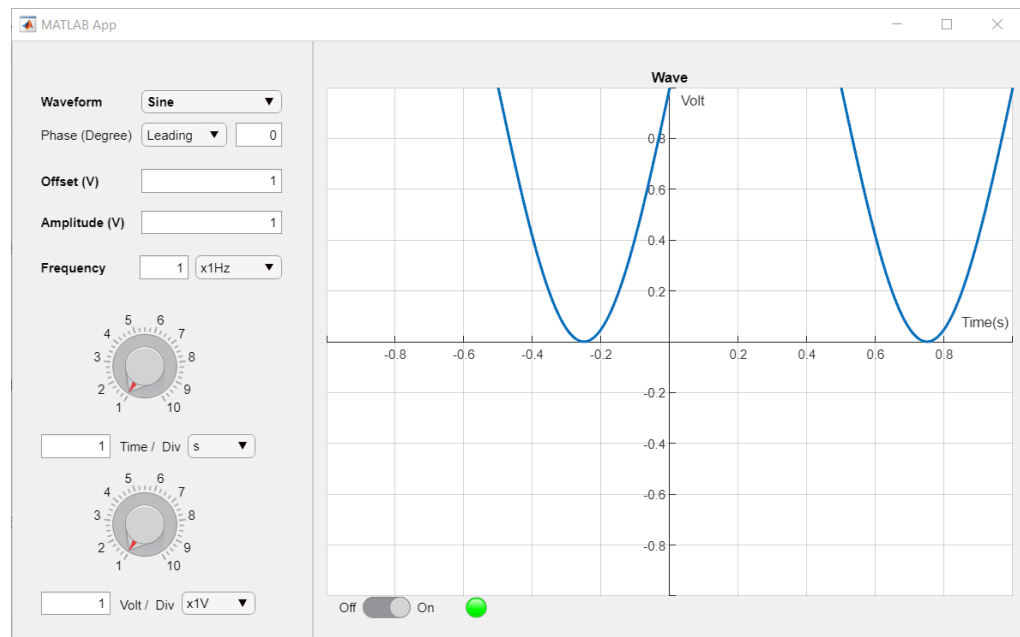
#### Lagging 90 Degree



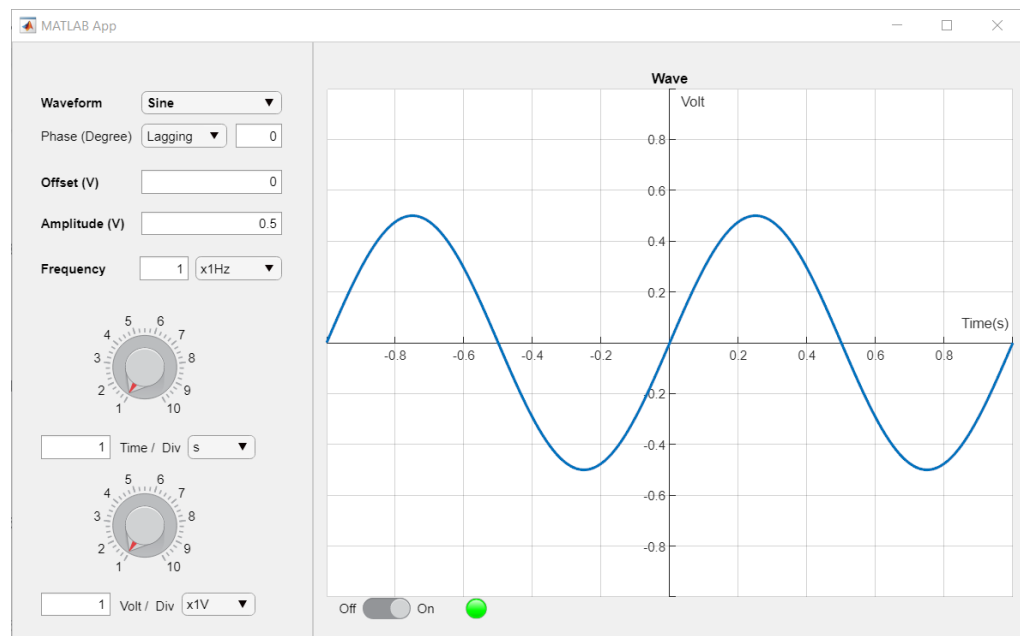
#### Leading 90 Degree



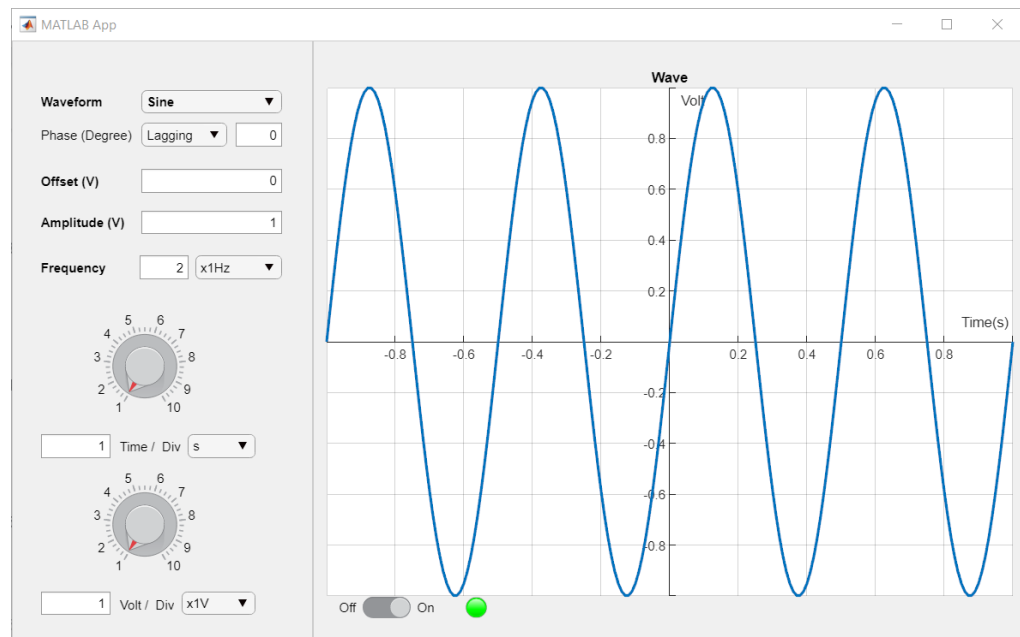
offset 1V



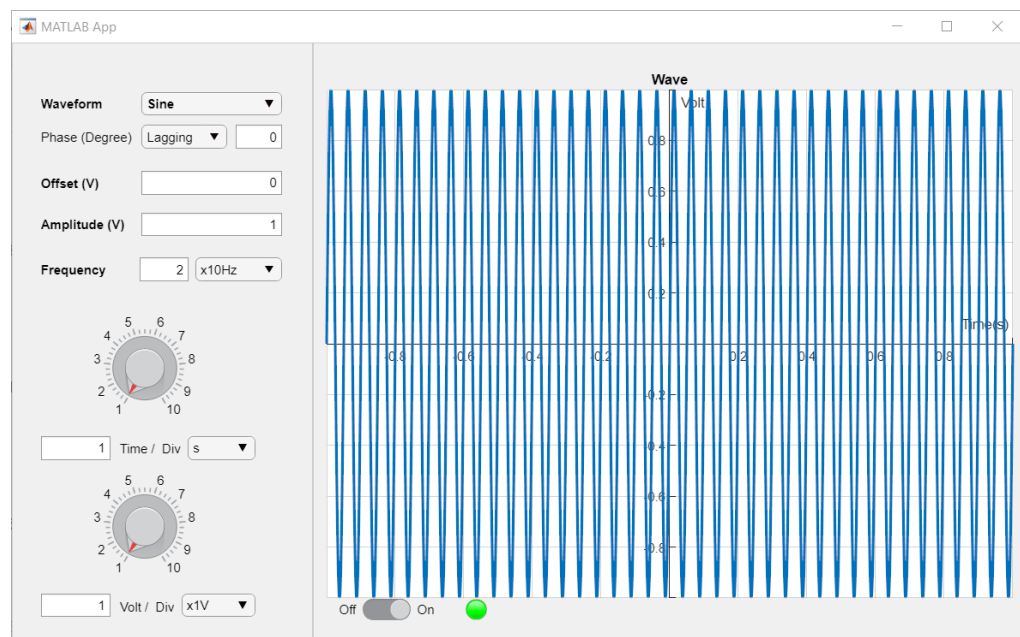
Amplitude 0.5



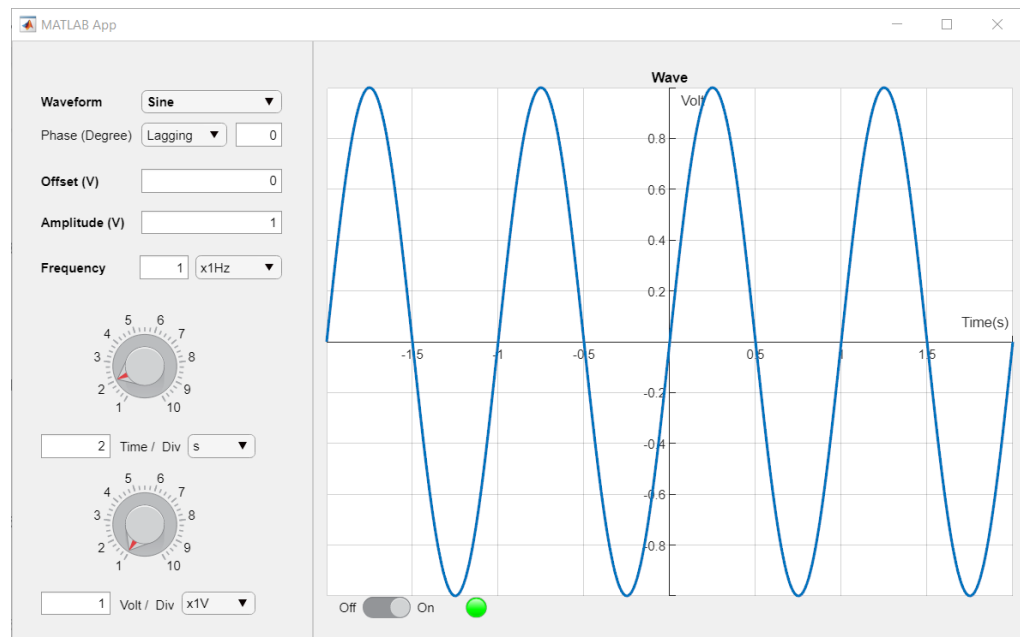
Frequency 2 x1Hz



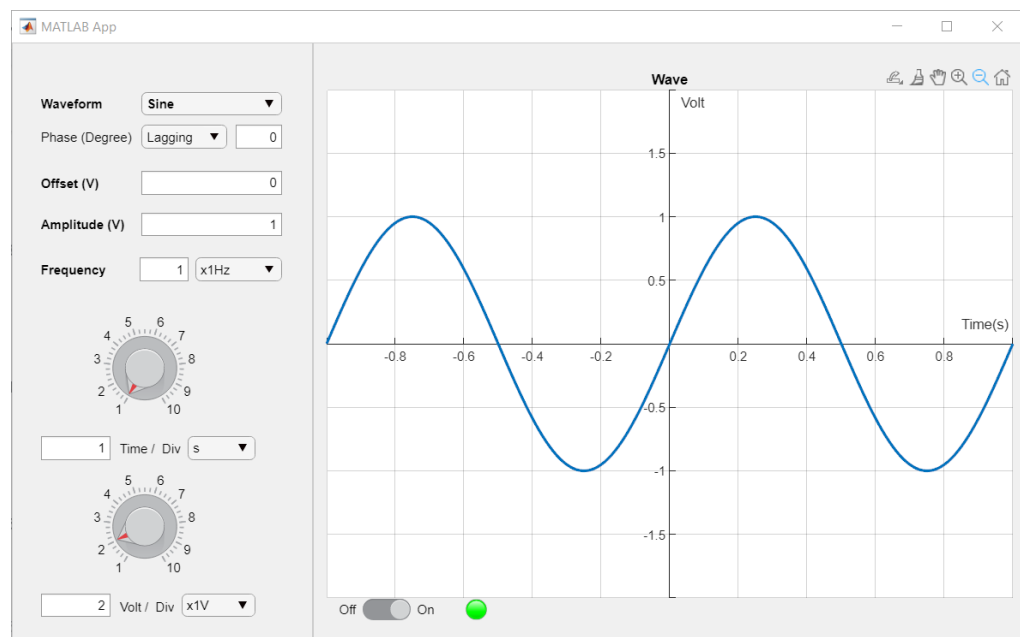
Frequency 2 x10Hz



Time/Div 2s

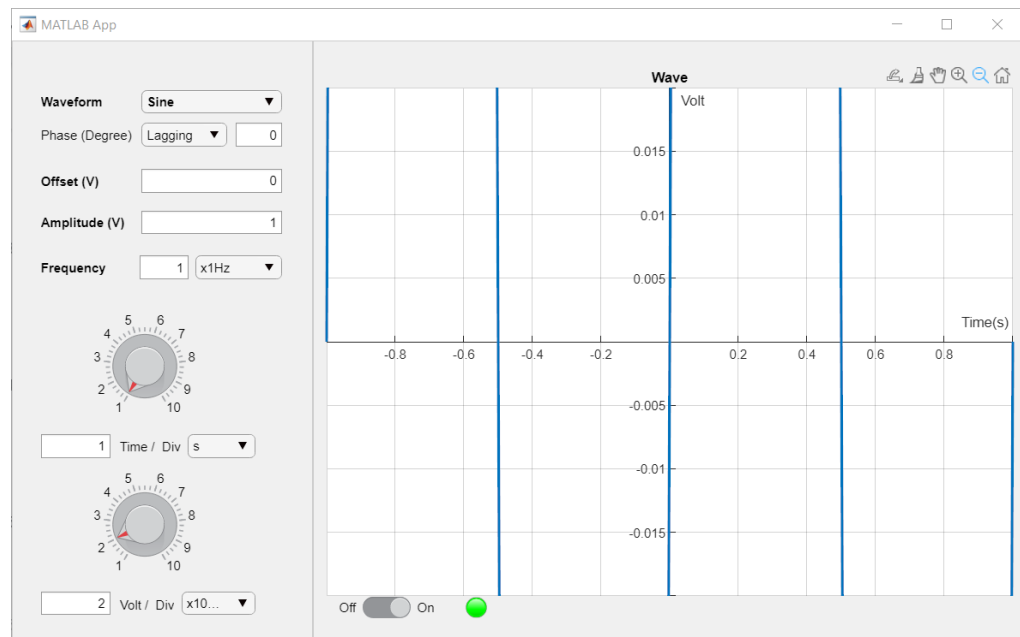


Volt/Div 2 x1V



Volt/Div 2 x10mV



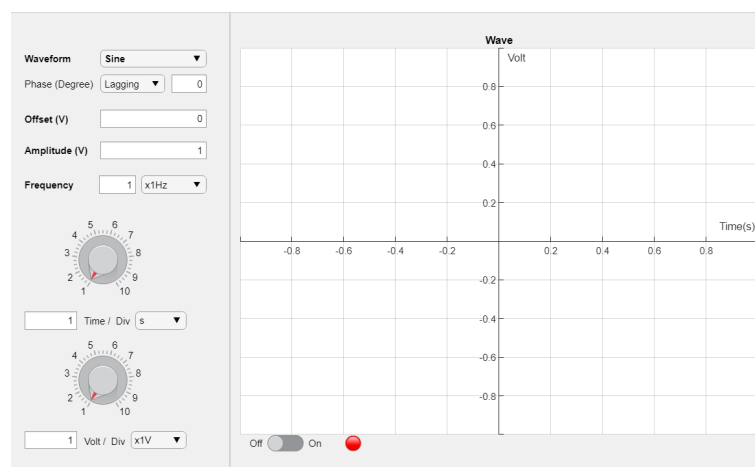


## บทที่ 4

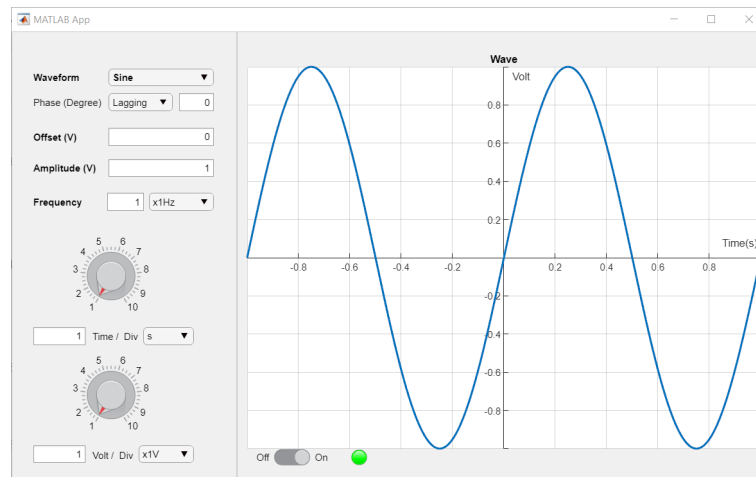
### 4.1 ผลการดำเนินการ

#### 4.1.1 GUI

off



on



## 4.1.2 Code

### 4.1.2.1 สร้างฟังก์ชันคืนค่า เมื่อกดเลือกหน่วยต่างๆ

```

function [unitvalue] = timeUnit(app)
    unit = app.TimeDivDropDown.Value;
    if(strcmp(unit,'s'))
        unitvalue = 1;
    elseif(strcmp(unit,'x1ms'))
        unitvalue = 1/10^3;
    elseif(strcmp(unit,'x10ms'))
        unitvalue = 10/10^3;
    elseif(strcmp(unit,'x100ms'))
        unitvalue = 100/10^3;
    elseif(strcmp(unit,'x10us'))
        unitvalue = 10/10^6;
    elseif(strcmp(unit,'x100us'))
        unitvalue = 100/10^6;
    end
end
function xlabelname(app)
    unit = app.TimeDivDropDown.Value;
    app.UIAxes.XLabel.String = "Time("+unit+")";
end

function [VDvalue] = VDUnit(app)
    unit = app.VoltDivDropDown.Value;
    if(strcmp(unit,'x1V'))
        VDvalue = 1;
    elseif(strcmp(unit,'x10mV'))
        VDvalue = 10/1000;
    elseif(strcmp(unit,'x100mV'))
        VDvalue = 100/1000;
    end
end

function [frevalue] = FreUnit(app)
    unit = app.FreDropDown.Value;
    if(strcmp(unit,'x1Hz'))
        frevalue = 1;
    elseif(strcmp(unit,'x10Hz'))
        frevalue = 10;
    elseif(strcmp(unit,'x100Hz'))
        frevalue = 100;
    elseif(strcmp(unit,'x1kHz'))
        frevalue = 1000;
    elseif(strcmp(unit,'x10kHz'))
        frevalue = 10000;
    end
end

```

4.1.2.2 สร้างฟังก์ชันกำหนดค่าเริ่มต้นเมื่อเปิด Application จะถูกเรียกใช้  
เมื่อ กดปุ่มจาก on เป็น off

```
function setDefault(app)
    cla(app.UIAxes);
    app.AmplitudeVEditField.Value = 1;
    app.FrequencyEditField.Value = 1;
    app.PhaseDegreeEditField.Value = 0;
    app.OffsetVEditField.Value = 0;
    app.TimeDivDropDown.Value = 's';
    app.VoltDivDropDown.Value = 'x1V';
    app.FreDropDown.Value = 'x1Hz';
    app.TimeDivKnob.Value = 1;
    app.VoltDivKnob.Value = 1;
    app.VoltdivEditField.Value=1;
    app.TimedivEditField.Value=1;
    app.LagLead.Value='Lagging';

    app.UIAxes.YLim = [-1 1];
    app.UIAxes.XLim = [-1 1];
    app.UIAxes.XAxisLocation = 'origin';
    app.UIAxes.YAxisLocation = 'origin';
    xlabelname(app);
end
end
```

4.1.2.3 function SwitchValueChanged(app, event) เป็นฟังก์ชันหลักของ Application graph generator นี้ โดยมีการแบ่งเป็น 2 ส่วนหลักๆ ดังนี้

1. ส่วนแรก จะสร้างตัวแปรต่างๆ และกำหนดค่า เพื่อนำมาใช้งานในการ plot กราฟ

```
% Value changed function: Switch
function SwitchValueChanged(app, event)
    app.UIAxes.XAxisLocation = 'origin';
    app.UIAxes.YAxisLocation = 'origin';
    value = app.Switch.Value;
    amp = app.AmplitudeVEditField.Value ;
    F = app.FrequencyEditField.Value;
    FU = FreUnit(app);
    F = F*FU;
    Td = app.TimeDivKnob.Value;
    t = -1*Td:1/(100*FU): Td;
    TU = timeUnit(app);
    t = t*TU;
    Tperiod = 1/F;
    Omega = (2*pi)/Tperiod; % w
    PhaseAngle = app.PhaseDegreeEditField.Value;
    PhaseAngle = (PhaseAngle*Omega)/360;
    LL = app.LagLead.Value;
    if strcmp(LL, 'Lagging')
        PhaseAngle = PhaseAngle*(-1);
    end
    offset = app.OffsetVEditField.Value;
    Wtype = app.WaveformDropDown.Value;
    VoltDiv = app.VoltDivKnob.Value;
    TimeDive = app.TimeDivKnob.Value;
```

2. ส่วนที่ 2 จะตรวจสอบเงื่อนไขของสวิตช์ on/off

- a. หากเงื่อนไขตรวจสอบว่า สวิตช์มีค่าเป็น off หลอด LED เป็นสีแดง และเรียกใช้ฟังก์ชัน `setDefault()` เพื่อการคืนค่าตัวแปรต่างๆ และหากสวิตช์มีค่าเป็น off จะไม่สามารถเปลี่ยนแปลงค่าต่างๆได้

```
if strcmp(value, 'Off')
    app.Lamp.Color = 'r';
    setDefault(app);
```

- b. หากเงื่อนไขตรวจสอบว่า สวิตช์มีค่าเป็น on หลอด LED เป็นสีเขียว

```
else
    xlabelname(app);
    app.Lamp.Color = 'g';
    cla(app.UIAxes);
    VU = VUnit(app);
    VoltDiv = VoltDiv*VU;

    TimeDiv = TimeDiv*TU;
    app.UIAxes.YLim = [(-1)*VoltDiv VoltDiv];
    app.UIAxes.XLim = [(-1)*TimeDiv TimeDiv];
```

ตรวจสอบค่าของ Wtype แล้วนำมาใช้ เลือกชนิดกราฟที่ต้องการ

plot

```

if strcmp(Wtype, 'Sine')
    y = amp*(sin(Omega*t+PhaseAngle));
    plot(app.UIAxes,t,y+offset, 'LineWidth',2.0);
elseif strcmp(Wtype, 'Square')
    sq = 0;
    for k=1:2:10000
        sq = sq + (4*amp*sin(k*Omega*t+PhaseAngle*k))/(pi*k);
    end
    plot(app.UIAxes,t,sq+offset, 'LineWidth',2.0);

elseif strcmp(Wtype, 'Triangular')
    tri = amp/4;
    for k = 1:2:10000
        tri = tri + ((-2*amp/(k*pi)^2)*cos(k*Omega*t+PhaseAngle*k));
    end
    plot(app.UIAxes,t,(tri*2)+offset, 'LineWidth',2.0);

elseif strcmp(Wtype, 'Sawtooth')
    sa = amp/2;%0;
    for k=1:1:10000
        sa = sa + (amp*-1*sin(k*Omega*t+PhaseAngle*k))/(pi*k);
    end
    plot(app.UIAxes,t,sa+offset, 'LineWidth',2.0);

elseif strcmp(Wtype, 'Full-wave rectified')
    full = (2*amp)/pi;
    for k=1:1:10000
        full = full + ((-4*amp)*cos(k*Omega*t+PhaseAngle*k))/(pi*((4*k*k)-1));
    end
    plot(app.UIAxes,t,full+offset, 'LineWidth',2.0);
end

```

4.1.2.4 function อื่นๆเป็นการเขียนขึ้น เพื่อให้การเปลี่ยนแปลงค่าต่างๆ ของโปรแกรมส่งผลต่อกราฟทันที โดยกำหนดในทุก UI Components

```
function UIAxesButtonDown(app, event)
    SwitchValueChanged(app, event);
end

% Callback function
function PlotButtonPushed(app, event)

end

% Value changed function: TimeDivKnob
function TimeDivKnobValueChanged(app, event)
    app.TimedivEditField.Value = app.TimeDivKnob.Value;
    SwitchValueChanged(app, event);
end

% Button down function: LeftPanel
function LeftPanelButtonDown(app, event)

end

% Value changed function: PhaseDegreeEditField
function PhaseDegreeEditFieldValueChanged(app, event)
    SwitchValueChanged(app, event);
end

% Value changed function: WaveformDropDown
function WaveformDropDownValueChanged(app, event)
    SwitchValueChanged(app, event);
end

% Value changed function: OffsetVEditField
function OffsetVEditFieldValueChanged(app, event)
    SwitchValueChanged(app, event);
end

% Value changed function: AmplitudeVEditField
function AmplitudeVEditFieldValueChanged(app, event)
    SwitchValueChanged(app, event);
```



## บทที่ 5

### 5.1 สรุปผลกิจกรรม อภิปรายผล ปัญหาและอุปสรรค และข้อเสนอแนะ

#### 5.1.1 สรุปผลกิจกรรม

จากการสร้าง Application การจำลองการทำงานของ graph generator ด้วยโปรแกรม MATLAB App Designer ผู้ใช้งานแอปพลิเคชันสามารถปรับค่า แอมพลิจูด, ความถี่, Phase(Lagging/Leading), offset, time/div, volt/div และ plot กราฟให้อยู่ในรูปของ Square wave , Triangular wave , Sine wave , Sawtooth wave และ Full wave ได้อย่างถูกต้อง

#### 5.1.2 อภิปรายผลกิจกรรม

จากการศึกษาค้นคว้าเกี่ยวกับเนื้อหาที่สนใจ และการใช้งาน MATLAB App Designer คณะผู้จัดทำจึงได้ออกแบบและสร้าง Application การจำลองการทำงานของ graph generator จนเสร็จสิ้น และได้ทำการตรวจสอบความถูกต้อง และพบว่า Application การจำลองการทำงานของ graph generator สามารถทำงานได้อย่างถูกต้องสมบูรณ์

## 5.2 ปัญหาและอุปสรรค

5.2.1 อุปกรณ์คอมพิวเตอร์ที่ใช้ในการจัดทำ Application มีปัญหาทั้งทางด้านอินเทอร์เน็ต และการประมวลผลของอุปกรณ์ จึงส่งผลให้ใช้เวลานานในการเปิดโปรแกรม MATLAB App Designer ขึ้น และบางครั้งก็พบว่าโปรแกรม MATLAB App Designer ปิดการทำงานด้วยตัวเองอัตโนมัติ

5.2.2 เนื่องจากใช้ระยะเวลาในการศึกษาวิธีใช้ MATLAB App Designer มากกว่าที่คาดไว้ จึงส่งผลให้การทำงานมีความล่าช้า กว่าที่วางแผนไว้

5.2.3 เนื่องด้วยสถานการณ์ covid-19 ทำให้ไม่สามารถนัดหมายมาทำงานที่เดียวกันได้ การสื่อสารและการนัดช่วงเวลาจึงมีอุปสรรคเล็กน้อย ทำให้บางครั้งการแบ่งงานที่ไม่ชัดเจน ทำให้เกิดการทำงานที่ทับซ้อนกัน จากการที่ต้องมีการส่ง code ไปมาระหว่างผู้จัดทำ และมีบางคนทำงานซ้ำในจุดเดียวกัน ทำให้ต้องมีการแก้ไขที่ซ้ำซ้อนจึงส่งผลให้การทำงานมีความล่าช้าในการทำงาน

### 5.3 ข้อเสนอแนะ

5.3.1 เนื่องจากปัญหาความล่าช้าในการเปิดการใช้งานโปรแกรม MATLAB App Designer จึงควรตรวจสอบพื้นที่ภายในคอมพิวเตอร์ก่อนติดตั้งโปรแกรม จัดการกับโปรแกรมที่ไม่ได้ใช้งานเพื่อให้คอมพิวเตอร์มีพื้นที่มากพอสำหรับการเปิดโปรแกรม และหากไม่สามารถเปิด MATLAB App Designer บนอุปกรณ์คอมพิวเตอร์ได้ สามารถเปิด MATLAB App Designer ผ่านเว็บไซต์ได้เช่นกัน

5.3.2 ปัญหาทางด้านอินเทอร์เน็ต ควรติดต่อพนักงานเพื่อตรวจสอบความผิดปกติในการทำงาน

5.3.3 ปัญหาทางการศึกษาวิธีใช้ MATLAB App Designer หากใช้เวลาในการฝึกฝนหาความรู้มากขึ้น ความเร็วในการใช้งาน ก็จะเร็วและมีประสิทธิภาพมากขึ้น

5.3.4 ปัญหาทางการแบ่งงานที่ไม่ชัดเจนนั้น อาจเกิดได้จากการสื่อสารที่ไม่ชัดเจน หรือการสื่อสารที่น้อยเกินไป หากมีการสื่อสารที่มากขึ้นและชัดเจนมากขึ้นจะทำให้การทำงานมีประสิทธิภาพมากขึ้น

## อ้างอิง

<http://www.rmutphysics.com/charud/virtualexperiment/labphysics2/oscilloscope/oscilloscope1/index4.htm>

[https://toolsdoctor.com/th/how-to-read-an-oscilloscope-screen/#What\\_is\\_DC\\_Offset\\_on\\_An\\_Oscilloscope](https://toolsdoctor.com/th/how-to-read-an-oscilloscope-screen/#What_is_DC_Offset_on_An_Oscilloscope)

เรวัต ศิริโกคาภิรมย์. (2564) .เอกสารประกอบการสอน Fourier series signal and system 2