Experiment No. 03

3.1 Experiment Name

Half-wave and Full-wave controlled rectifier circuits using Simulink

3.2 Objectives

- To learn about MATLAB tools and how they work
- To become acquainted with the Simulink platform and Simulink library
- To simulate various types of circuits using blocks from the library browser
- To design both half-wave and full-wave controlled rectifier circuits in Simulink

3.3 Theory

Simulink is a MATLAB-integrated simulation- and model-based design environment for dynamic and embedded systems.

For this experiment, half-wave and full-wave controlled rectifier circuits were designed following library functions and their output were observed,

3.3.1 MATLAB Function

The MATLAB Function blocks allow users to define custom functions in Simulink models using the MATLAB programming language. There are no continuous or discrete dynamic states in the custom functionality that you want to model.

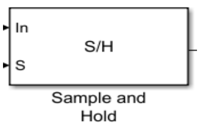
3.3.2 Mean

The Mean block computes the input signal's mean value. The mean value is calculated over a one-cycle running average window of the specified fundamental frequency:

$$Mean(f(t)) = \frac{1}{T}\int_{(t-T)}^{t} f(t)\cdot dt$$

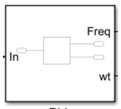
3.3.3 Sample and Hold

When a trigger event is received at the trigger port, the Sample and Hold block acquires the input at the signal port (marked by). The output is then held at the acquired input value until the next triggering event occurs.



3.3.4 Phase- Locked Loop (PLL)

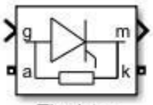
A phase-locked loop is a feedback system that combines a voltage-controlled oscillator and a phase comparator to adjust the oscillator frequency or phase to track phase-modulated signal.



PLL

3.3.5 Thyristor

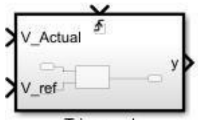
A thyristor is connected in parallel with a series RC snubber circuit. The thyristor model contains internal resistance and inductance when it is turned on. The internal inductance should be set to zero for most applications.



Thyristor

3.3.6 Triggered Subsystem

A triggered subsystem is a conditionally executed atomic subsystem that runs whenever the control signal (trigger signal) changes from a negative to a positive or zero value, or from zero to a positive value.



Triggered Subsystem

3.4 Apparatus

- Simulink

3.5 Simulink problem

The input voltage is 100sin(ωt), where the line frequency is 50 Hz. The circuit consists of a resistive load of 10Ω. Build a closed-loop control algorithm so that the output dc voltage remains constant at 40V. Show the input voltage, firing pulse, output voltage, and dc value waveforms in the scope. Simulate this in case of

- **Half-wave rectifier circuit**
- **Full-wave rectifier circuit**

3.6 Simulink Block Diagram & Waveform

- **Half-wave Rectifier**

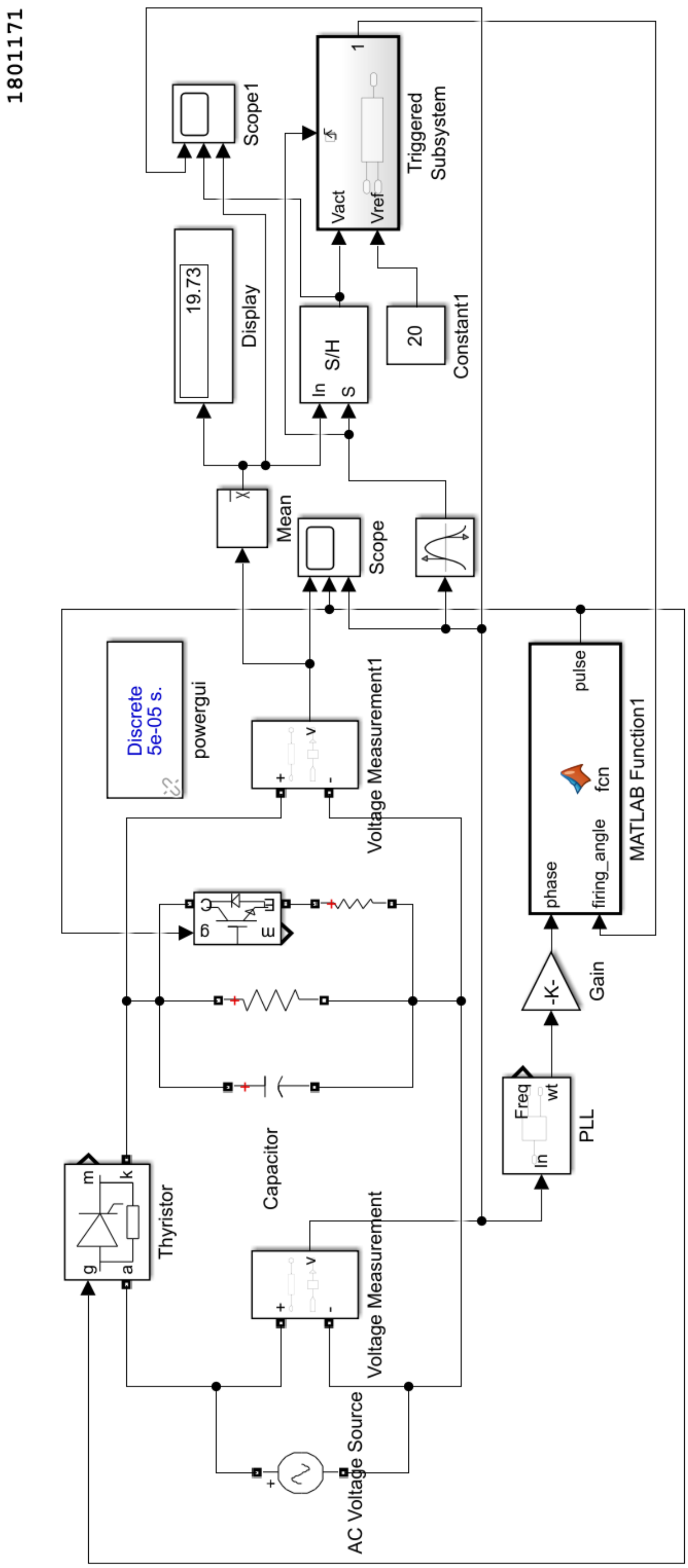


Fig. 3.1: Block diagram for controlled half- wave rectifier circuit

- **Code for triggered system**

```
function x = fcn(Vact,Vref)
%codegen
persistent firing_angle
if isempty(firing_angle),firing_angle=10,end;
step = 0.5;
if Vact<Vref
      firing_angle = firing_angle-step;
       if firing_angle<0
          firing_angle = 0;
       end
elseif Vact>Vref
         firing_angle = firing_angle+step;
          if firing_angle>180
             firing_angle = 180;
          else
              firing_angle = firing_angle;
          end
end
x = firing_angle;
end
```

- **Code for MATLAB function**

```
function pulse = fcn(phase,firing_angle)
%codegen
pulse_width = 10;
if
(phase>=firing_angle)&&(phase<=min((firing_angle+pulse_width),180))
      pulse = 1;
else
      pulse = 0;
end
end
```

- **Waveform**

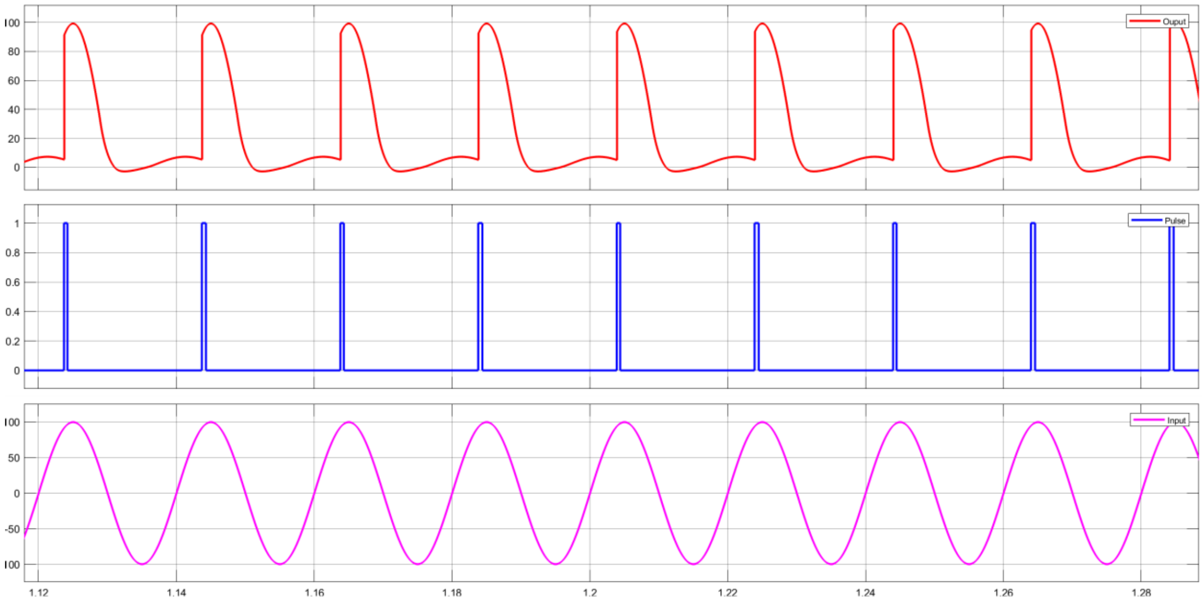


Fig. 3.2: Output, pulse, and input waveform for controlled half- wave rectifier circuit

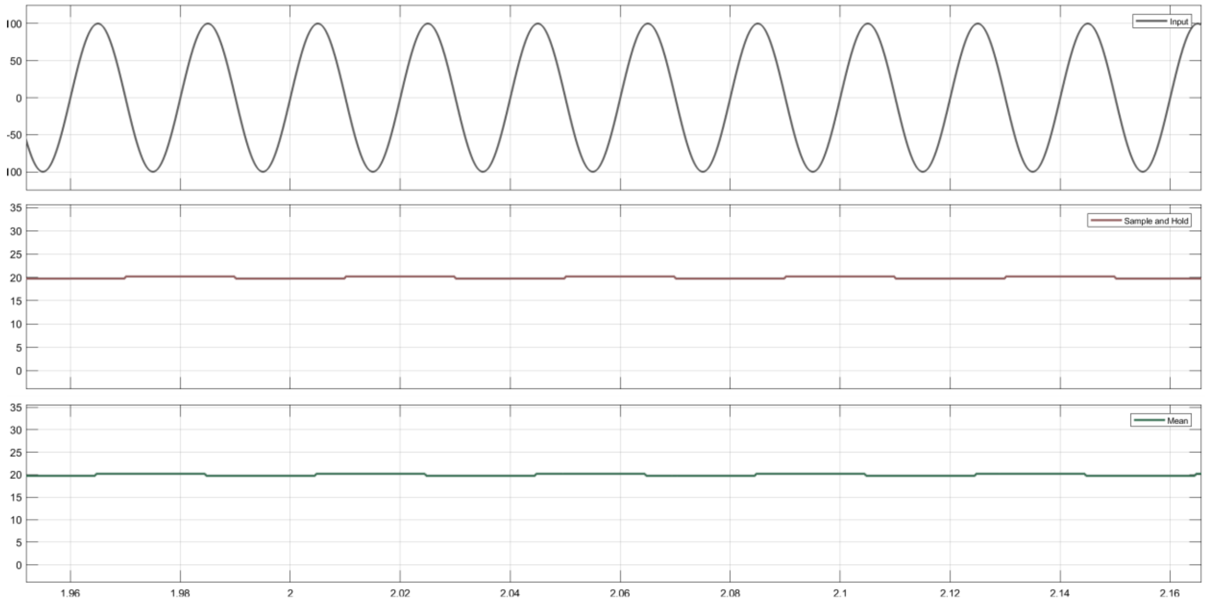


Fig. 3.3: Input, sample & hold, and mean waveform for controlled half- wave rectifier circuit

- **Full-wave Rectifier**

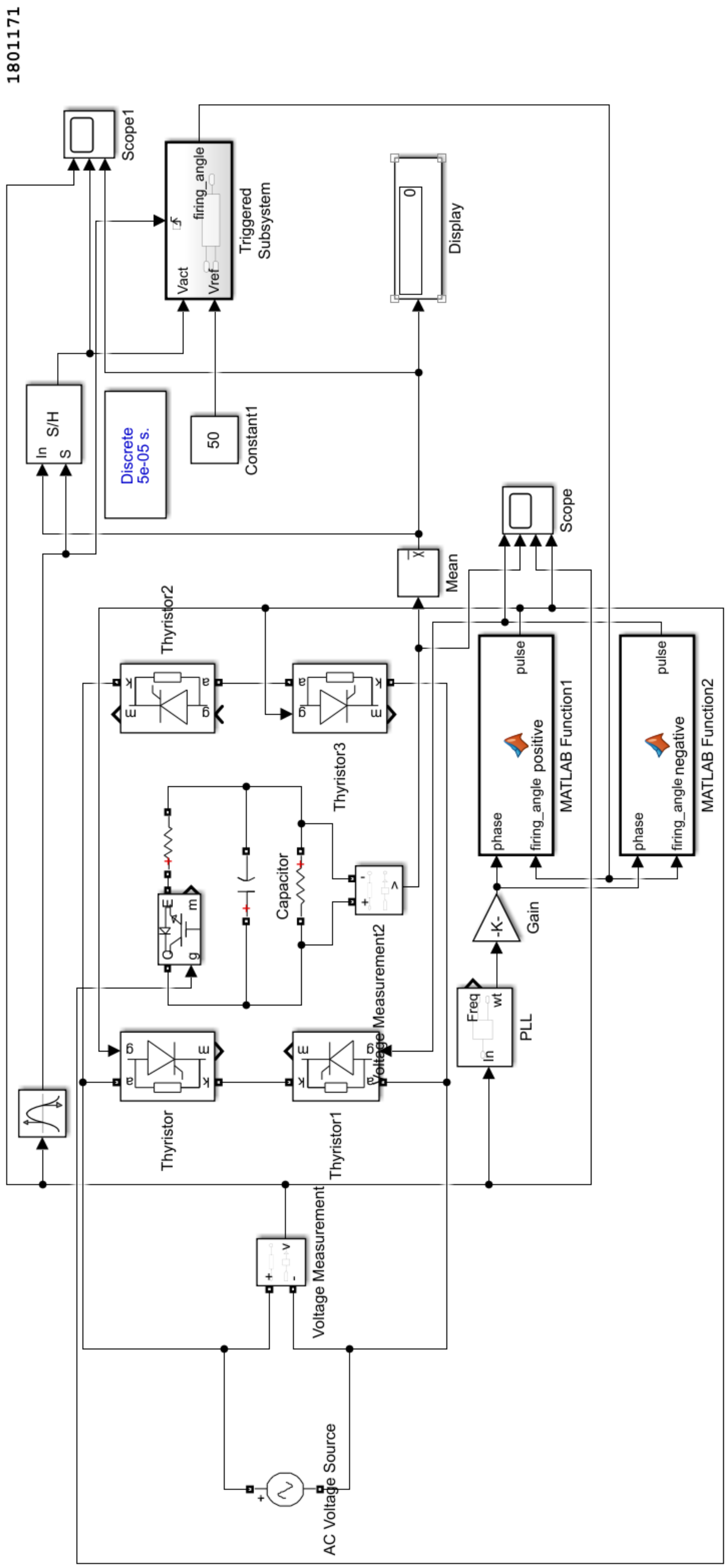


Fig. 3.4: Block diagram for controlled full- wave rectifier circuit

- **Code for triggered system**

```
function x = fcn(Vact,Vref)
%codegen
persistent firing_angle
if isempty(firing_angle),firing_angle=10,end;
step = 0.5;
if Vact<Vref
      firing_angle = firing_angle-step;
       if firing_angle<0
          firing_angle = 0;
       end
elseif Vact>Vref
         firing_angle = firing_angle+step;
          if firing_angle>180
             firing_angle = 180;
          else
              firing_angle = firing_angle;
          end
end
x = firing_angle;
end
```

- **Code for MATLAB function (positive half cycle)**

```
function pulse = positive(phase,firing_angle)
%codegen
pulse_width = 10;
if
(phase>=firing_angle)&&(phase<=min((firing_angle+pulse_width),180))
      pulse = 1;
else
      pulse = 0;
end
end
```

- **Code for MATLAB function (negative half cycle)**

```
function pulse = negative(phase,firing_angle)
%codegen
pulse_width = 10;
if
(phase>=(firing_angle+180))&&(phase<=min((firing_angle+pulse_width+1
80),360))
      pulse = 1;
else
      pulse = 0;
end
end
```

- **Waveform**

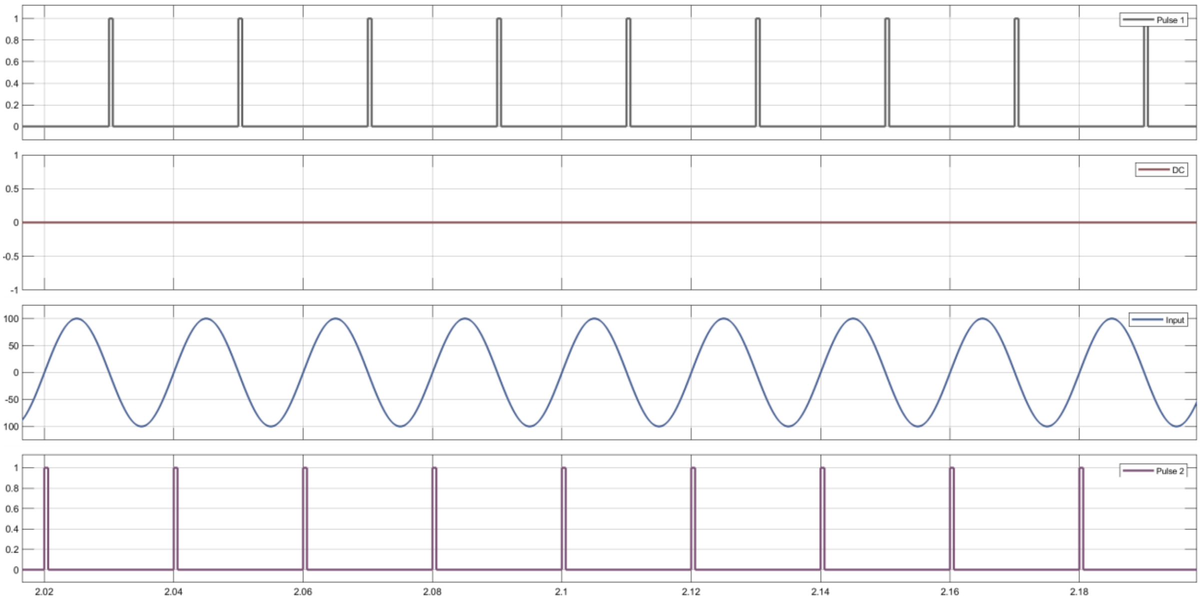


Fig. 3.2: Pulse1, DC, pulse2, and input waveform for controlled half- wave rectifier circuit

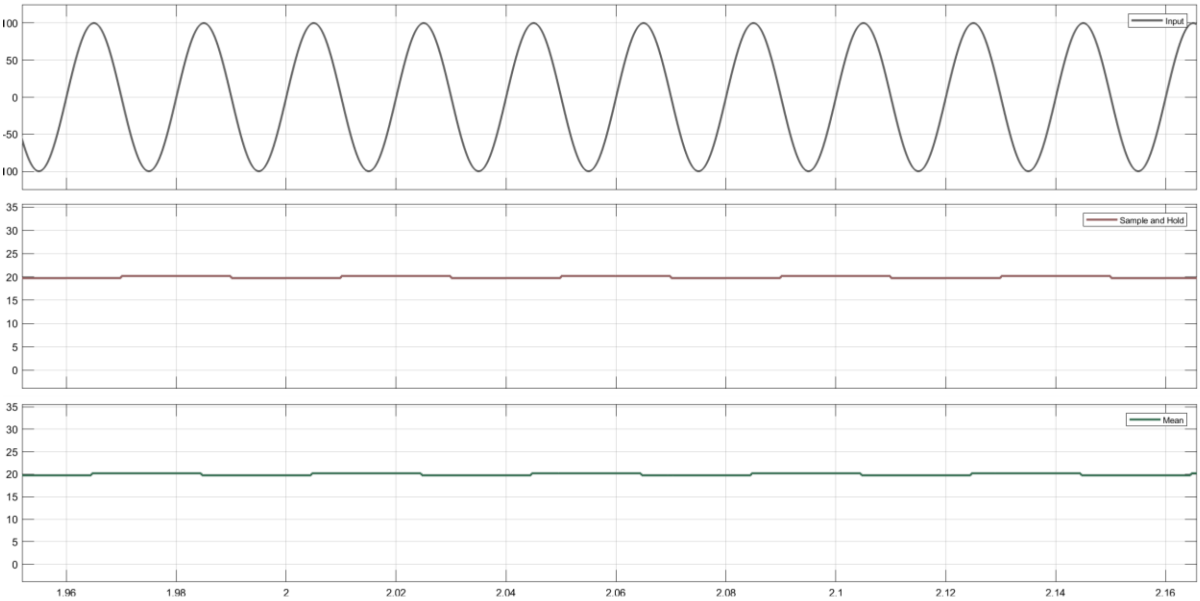


Fig. 3.3: Input, sample & hold, and mean waveform for controlled half- wave rectifier circuit

3.7 Discussion & Conclusion

In this experiment, triggered subsystem block was used. Additionally, to decrease the amount of block components in the Simulink platform, we employed MATLAB function blocks and codes. In the experiment, half-wave and full-wave controlled rectifier circuits are developed and studied.

Moreover, expected outputs were observed at scope. Thus, the experiment was a success.