# Model-Based Software Development

## Computer Lab 2

Prepared by:

Madhuri Salluri

Adeyemi Mayowa

Date: 01-04-2023

# Table Of Contents

# 1PWM For Resonant Converter

## 1.10 Introduction

NCO (Numerically Controlled Oscillator) is a digital signal processing technique used to generate periodic waveforms with high precision and flexibility. PWM (Pulse Width Modulation) is a technique used to encode an analog signal into a digital signal by varying the width of the pulses in a periodic signal. NCOs can be used to generate the periodic signals required for PWM, allowing for more complex modulation schemes. NCO and PWM are commonly used in communication systems, power electronics, and audio signal processing.

## 1.11 Problem Definition

In resonant power converters based on PWM, the input voltage is applied to a resonant circuit that consists of an inductor and a capacitor. The circuit resonates at a particular frequency, and the energy is transferred to the load through a transformer. The transformer is controlled by a switching device that is turned on and off at a high frequency, which is referred to as the carrier frequency.

The carrier frequency in resonant power converters must be varied to maintain resonance, which allows for higher efficiency and reduced switching losses. In addition, the duty cycle of the control signal is kept constant, typically at 50%, to control the output voltage.

In contrast, non-resonant power converters, such as inverters and H-bridges, use a fixed carrier frequency and vary the duty cycle of the control signal to control the output voltage. These converters are simpler and less expensive than resonant converters, but they have lower efficiency and produce more switching losses.

We'll use a sawtooth carrier instead of a triangular one to simplify the problem. An NCO is a suitable way to generate a frequency-variant sawtooth signal.

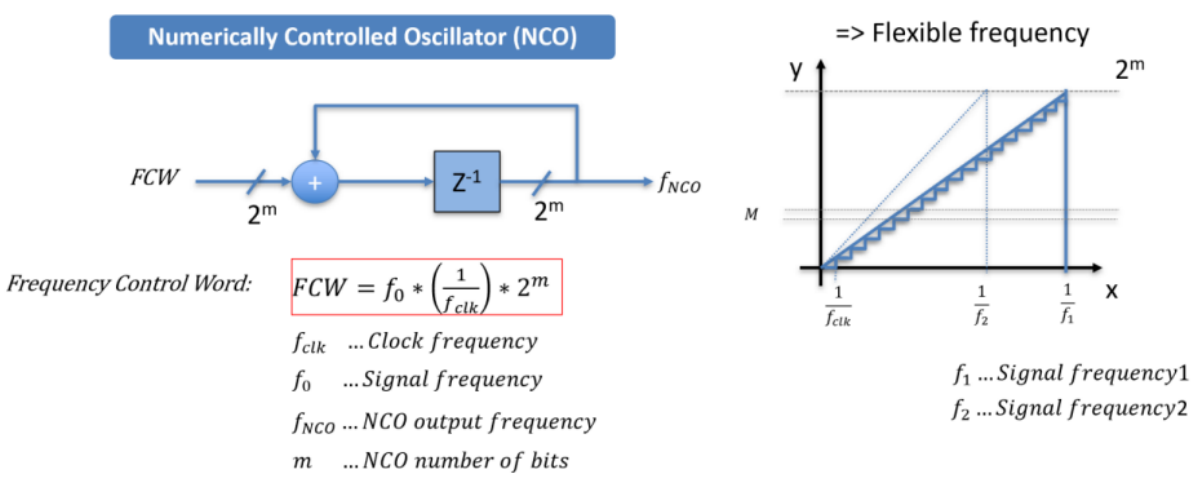


Figure 1: Saw Tooth Wave & NCO

## 1.12 Implementation Of Numerically Controlled Oscillator (NCO) & PWM

In this project, I combined a ramp system carrier frequency and duty cycle system to produce a Pulse Width Modulation (PWM) signal for a resonant converter. I used various MATLAB functions, constant blocks, compare blocks, and select blocks to achieve this.

The PWM signal is a technique used to control the power delivered to electrical devices, such as motors and LED lights. By controlling the width of the pulse in a periodic waveform, the average power delivered to the device can be adjusted. The resonant converter is a type of power converter that can efficiently convert DC power to AC power at high frequencies, making it useful for many applications, including power supplies for computers and other electronics.

The ramp system is a function that generates a linearly increasing or decreasing signal over time. In this project, I used a ramp system to generate a signal that is used to modulate the carrier frequency of the PWM signal. The carrier frequency determines the frequency of the periodic waveform used to control the power delivered to the device.

The duty cycle system is another function that generates a signal that is used to control the width of the pulse in the periodic waveform. The duty cycle determines the percentage of time that the pulse is in the "on" state, which controls the amount of power delivered to the device.

To generate the PWM signal for the resonant converter, I combined the ramp system and duty cycle system using various MATLAB functions, constant blocks, compare blocks, and select blocks. The MATLAB functions were used to perform mathematical operations on the signals, such as multiplying and adding them together. The constant blocks were used to set the values of various parameters, such as the frequency and duty cycle of the PWM signal. The compare blocks were used to compare the ramp signal to the duty cycle signal, and the select blocks were used to select the appropriate value for the PWM signal based on the result of the comparison.

Overall, this project demonstrated how different systems can be combined to produce a complex signal, such as a PWM signal for a resonant converter. By using MATLAB functions, constant blocks, compare blocks, and select blocks, I was able to implement the ramp system carrier frequency and duty cycle system to generate the PWM signal. In the next subsections, I will explain each system in more detail, including how they were implemented in the project.

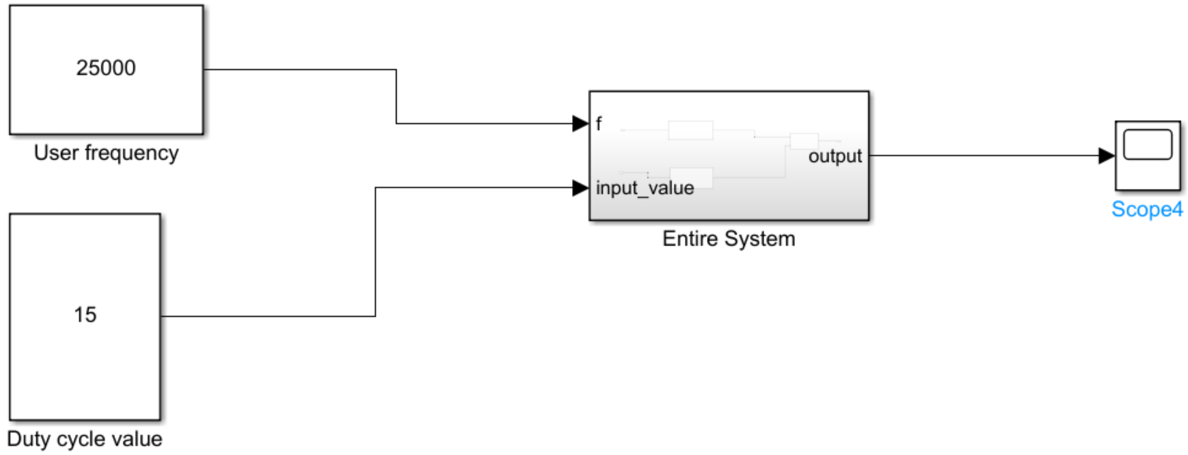


Figure2: The Entire Model for the Resonant Converter

## 1.13 The Ramp System

A ramp signal is a signal that varies linearly with time. It can be generated using a mathematical function or an electronic circuit. The ramp signal is commonly used in electronics for various applications, including generating control signals for power electronics, testing and calibration of instruments, and signal processing.

In this system, a ramp signal was generated that begins from zero, reaches a specific value, and returns back to zero. The equation **fo*(1/fclk)*2^m** was used to calculate the values of the ramp signal, where fo is the desired frequency of the ramp signal, fclk is the clock frequency of the system, and 2^m is a constant value.

To generate the ramp signal, a MATLAB function code was used that compared and selected the output of a delayed value. The delayed value was a result of the calculation of the above equation, and the compare and select blocks were used to ensure that the ramp signal started from zero, reached the desired value, and returned back to zero.

The equation **fo*(1/fclk)*2^m** was used to calculate the period of the ramp signal, which was then used to calculate the values of the ramp signal at each time step. The compare and select blocks were used to compare the current value of the ramp signal to the desired value and to select the appropriate output based on the comparison. This allowed the ramp signal to smoothly reach the desired value and then return back to zero.

Overall, this project demonstrated how a ramp signal can be generated using a mathematical equation and MATLAB function code. By using the compare and select blocks, the ramp signal was able to smoothly reach the desired value and return back to zero, making it suitable for a wide range of applications, including generating control signals for power electronics and testing and calibration of instruments.

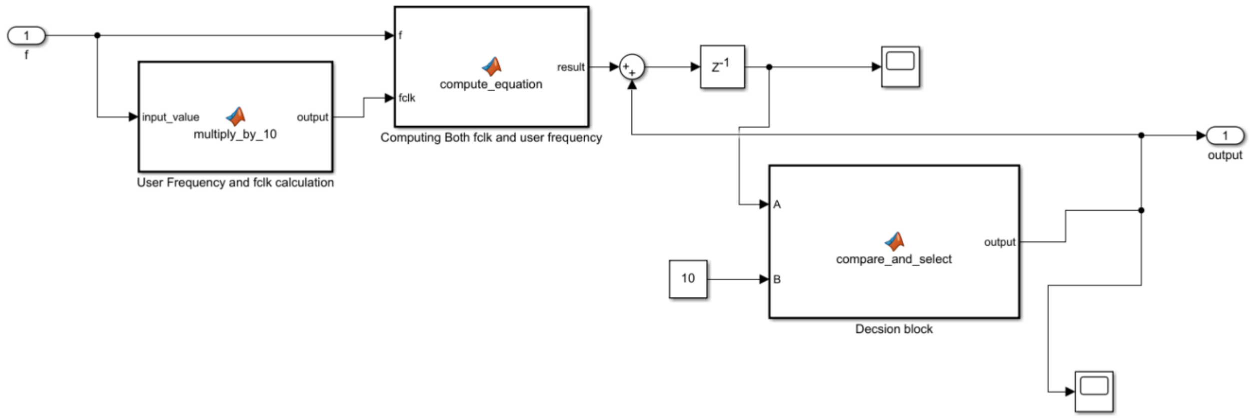


Figure 3: The ramp system Model in Simulink

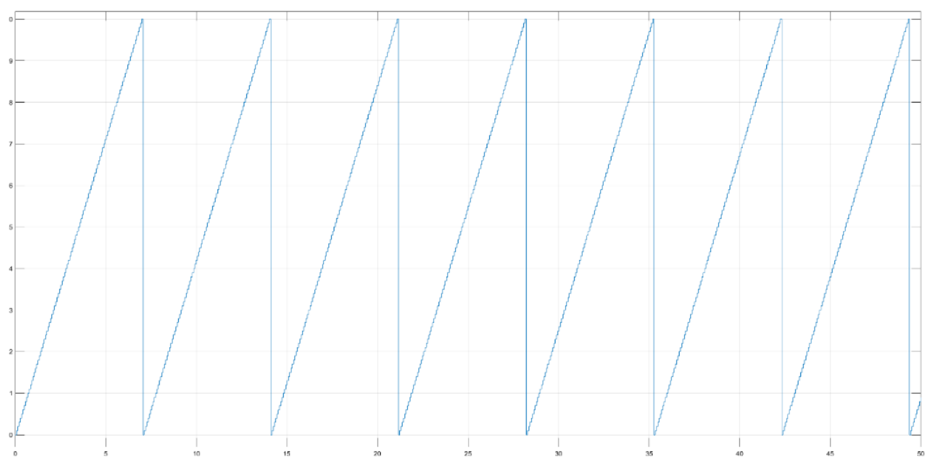

Figure 4: Generation of the ramp Signal

## 1.5 MATLAB Code behind the ramp system

### User frequency and fclk calculation

```
function output = multiply_by_10(input_value)
% MULTIPLY_BY_10 Multiplies an input value by 10 and returns the result
%   OUTPUT = MULTIPLY_BY_10(INPUT_VALUE) multiplies the input value by
10 and
%   returns the result.

output = input_value * 10;
```

### User frequency and fclk computation

```
function result = compute_equation(f, fclk)
% Computes f*(1/fclk)*2^m
%
% Inputs:
```

```
%   f: frequency value
%   fclk: clock frequency value
%
% Output:
%   result: the value of f*(1/fclk)

result = f * (1/fclk);
end
```

### Decision Block

```
function output = compare_and_select(A, B)
% Compares A and B to 0 and returns A if A is less than B, or 0
otherwise
%
% Inputs:
%   A: the first value to compare
%   B: the second value to compare
%
% Output:
%   output: the value of A if A is less than B, or 0 otherwise

if A < B
    output = A;
else
    output = 0;
end
end
```

## 1.6 The Duty Cycle System

The below figure 5 shows the duty cycle system of the Resonant Converter

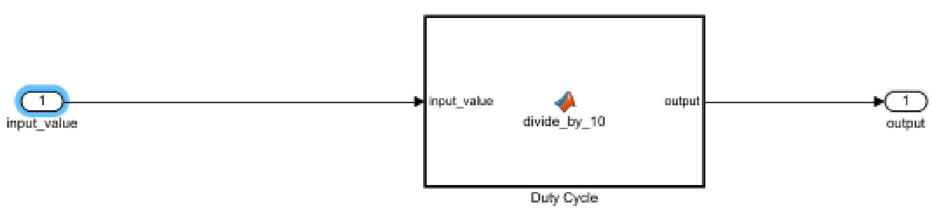


Figure 5: The Duty Cycle System

**About a duty cycle**

Duty cycle is a measure of the percentage of time that a signal is in the "on" or "active" state during a given period. It is often used in electronics and signal processing to control the amount of power delivered to a device or to control the timing of a circuit.

**Details about the duty cycle system**

In the system developed, a user-defined duty cycle value is taken and used to affect the ramp signal being produced by the ramp system. The duty cycle represents the percentage of time that the signal is in the "on" state, and therefore affects the amount of power delivered to a device or the timing of a circuit.

To compute the duty cycle, the ramp signal is compared to the user-defined duty cycle value using a compare block. If the ramp signal is less than the duty cycle value, the output of the compare block is "on", and if it is greater than or equal to the duty cycle value, the output is "off". This affects the ramp signal being produced, and allows for precise control over the amount of power delivered or the timing of a circuit.

**Matlab code for the duty cycle system**

```
function output = divide_by_10(input_value)
% DIVIDE_BY_10 Divides an input value by 10 and returns the result
%   OUTPUT = DIVIDE_BY_10(INPUT_VALUE) divides the input value by 10 and
%   returns the result.

output = input_value / 10;
```

## 1.7 The carrier Signal System

**About A Carrier Signal**

The carrier signal serves as a "carrier" for the information signal, which is imposed on it by varying its amplitude, frequency or phase. Carrier signals are commonly used in radio communications, television broadcasting, and cellular phone systems.

**Details About the Carrier Signal system**

In the system developed to generate a Pulse Width Modulation (PWM) signal, the input signal of the ramp signal and the duty cycle is compared together by a compare and select MATLAB function code. The ramp signal is generated using an equation, and the duty cycle is defined by the user as a percentage of the signal cycle.

The compare and select function code compares the output of the ramp signal with the user-defined duty cycle value. If the ramp signal is less than the duty cycle value, the output of the compare and select block is a "high" or "on" signal. If it is greater than or equal to the duty cycle value, the output is a "low" or "off" signal. This process generates a PWM signal with a frequency determined by the user input value and a duty cycle determined by the user-defined value.

The carrier signal is used in this process to help create a PWM signal that can then be used by the user for various purposes, such as controlling the speed of a motor or the brightness of an LED. The values assigned to the frequency and duty cycle inputs determine the characteristics of the PWM signal, and therefore affect how the signal is used in practical applications.

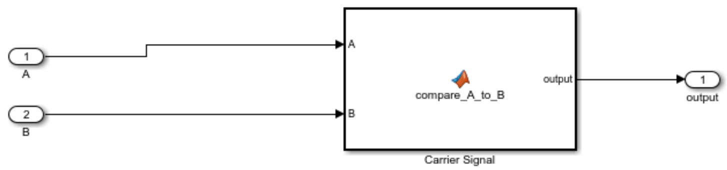


Figure 6: Carrier Signal System

Matlab function code for the Carrier System

```
function output = compare_A_to_B(A, B)
% COMPARE_A_TO_B Compares A to B and returns 1 if A <= B, 0 otherwise.
%   OUTPUT = COMPARE_A_TO_B(A, B) returns 1 if A is less than or equal
to B,
%   and 0 otherwise.

if A <= B
    output = 1;
else
    output = 0;
end
```

## 1.8 The Functional requirements

The PWM function developed using MATLAB satisfies the specified requirements. The pulse train generated by the PWM function is normalized between 0 to 1, providing a standard range for the user to work with. The carrier frequency range is set between 10 kHz to 100 kHz, giving the user a wide range of frequencies to choose from. Additionally, the duty cycle range is between 1% to 99%, and the minimum resolution is 1%. This ensures that the user has precise control over the duty cycle of the PWM signal.

To allow for user flexibility, the carrier frequency and duty cycle inputs can be specified in any clock cycle. However, any changes made to these inputs are held in standby until the next carrier period begins. This ensures that there are no glitches or abrupt changes in the PWM signal, which could result in undesirable effects in practical applications.

The system accomplishes these requirements by using a combination of different MATLAB function codes, constant blocks, compare and select blocks. The ramp signal is generated using a MATLAB function code, while the duty cycle is computed using a separate code. The compare and select block is used to compare the ramp signal and duty cycle, and generate the PWM signal accordingly.

Overall, the system developed provides the user with a precise and flexible way of generating PWM signals, while also ensuring that the signal is stable and glitch-free.

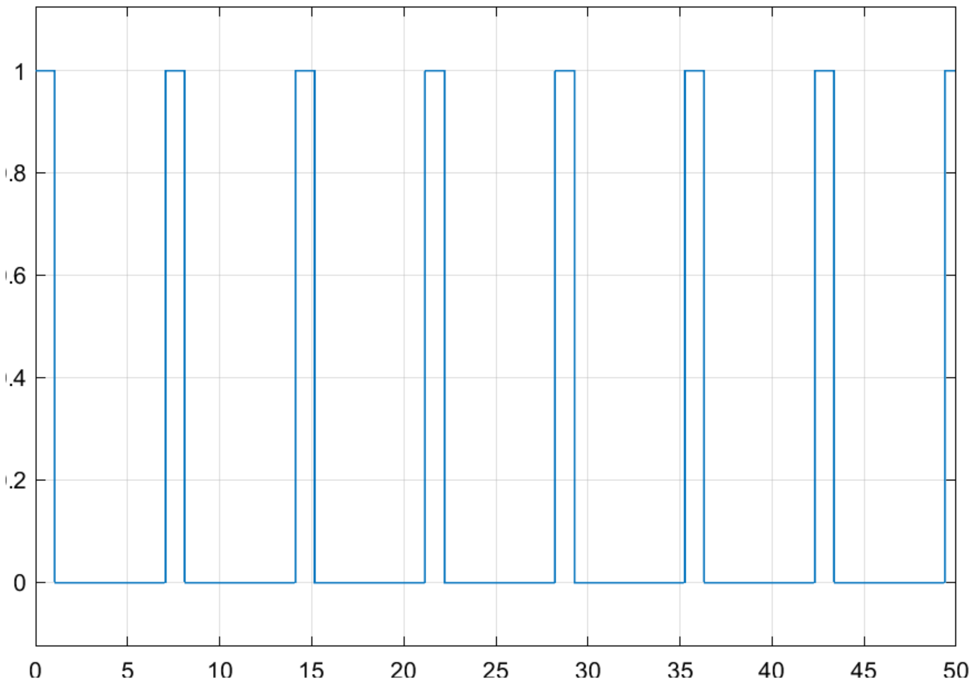


Figure 7: PWM generated by the system as the frequency 25khz and duty cycle 15

### 1.9 Discussion on the entire system

The PWM technique is used in the control of many power electronic converters, including resonant converters. In generating the PWM signal, a ramp signal is produced using a MATLAB function code, which starts from zero and increases to a specific value before returning to zero. A user-defined duty cycle value is used to compute the ramp signal and affect its shape. The ramp signal and the duty cycle are then compared using a compare and select MATLAB function code, which generates a Pulse Width Modulation (PWM) signal. The carrier signal is used to help create the PWM signal, which can then be used by the user for various applications such as controlling the speed of a motor or the brightness of an LED. The values assigned to the frequency and duty cycle inputs determine the characteristics of the PWM signal, and therefore affect how the signal is used in practical applications. Overall, the process involves generating a ramp signal, computing a duty cycle value, comparing them, and generating a PWM signal for use in various applications.