MÜHENDİSLİK FAKÜLTESİ
FACULTY OF ENGINEERING
ELEKTRİK-ELEKTRONİK MÜHENDİSLİĞİ BÖLÜMÜ
DEPARTMENT OF ELECTRICAL AND
ELECTRONICS ENGINEERING

ORTA DOĞU TEKNİK ÜNİVERSİTESİ
MIDDLE EAST TECHNICAL UNIVERSITY

DUMLUPINAR BULVARI 06800
ÇANKAYA ANKARA/TURKEY
T: +90 312 210 23 02
F: +90 312 210 23 04
ee@metu.edu.tr
www.eee.metu.edu.tr

EXPERIMENT 4. DECIMATION, INTERPOLATION AND PHASE-LOCKED LOOP

PART 1

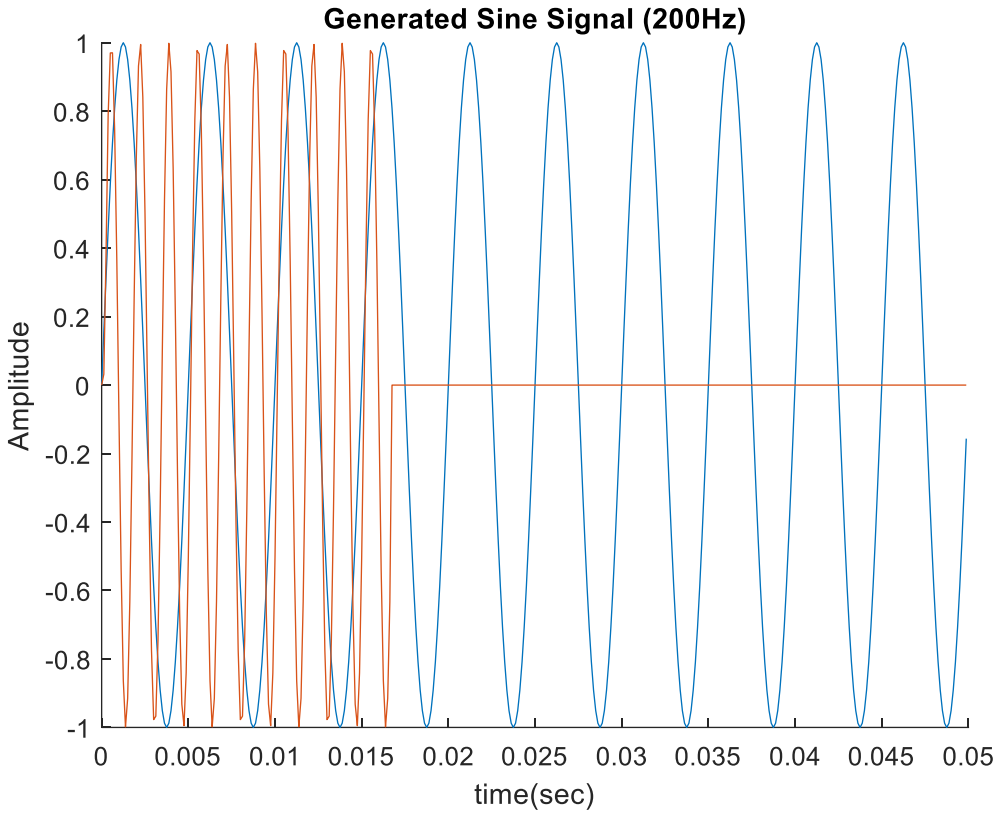
LABORATORY REPORT

Student 1: Süleyman Emre CAN-2093524
Student 2: Yekta Demirci-2093607

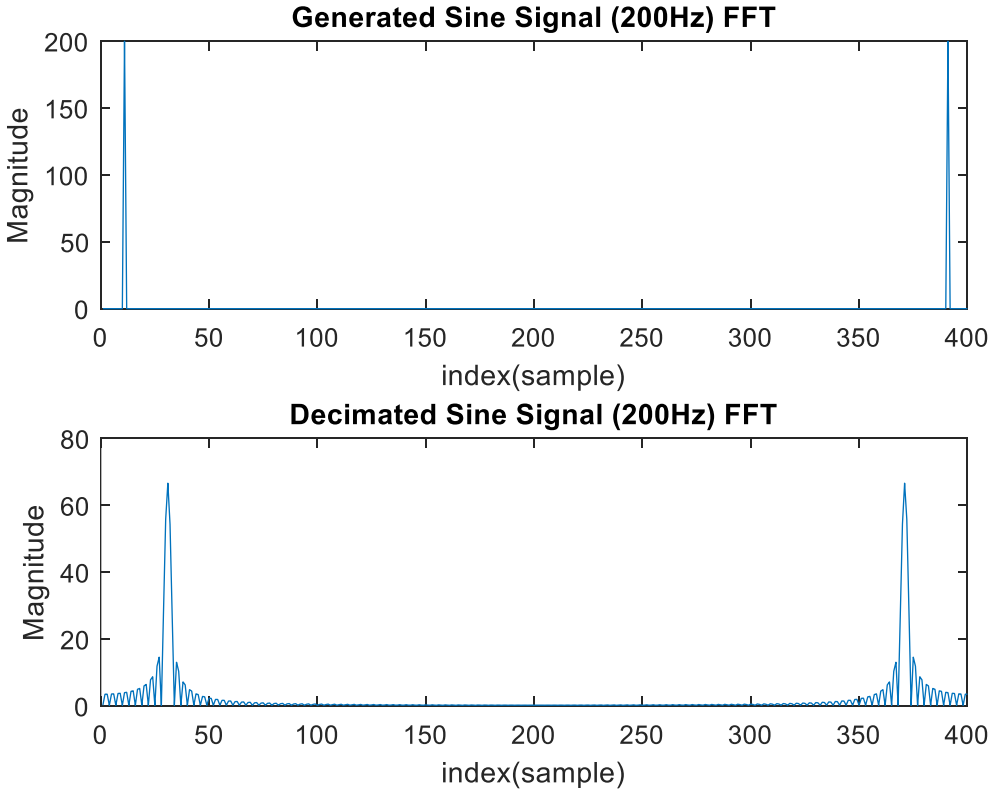
a) Write a code in MATLAB to implement **decimation** operation. Given the input signal, and decimation factor, M, the code obtains the output. The **input** and **output** signals are **plotted** both in **time** and **frequency**.

Take M=3 in the following tasks.

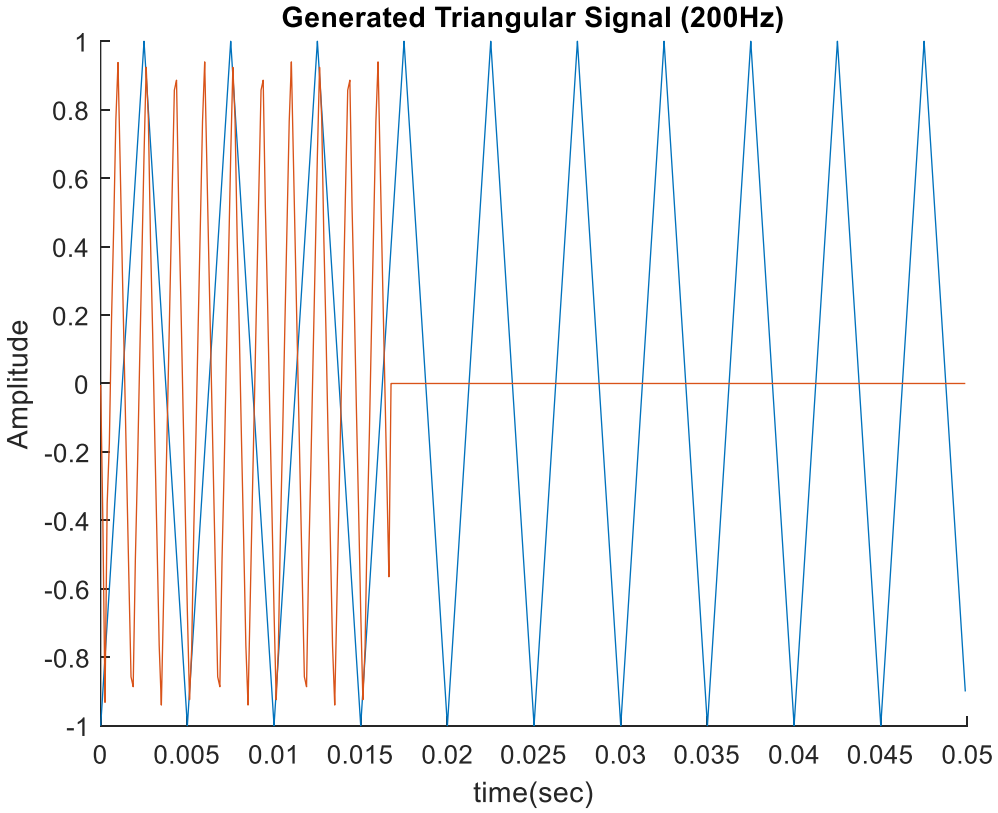
a)1) Generate a **200 Hz sine sequence** in **0-50 ms time range** with **sampling rate 8000 samples/sec**. **Plot** it and output in **time** using the **same time axis**, i.e., time difference between consecutive sequence elements is **1/8000**.



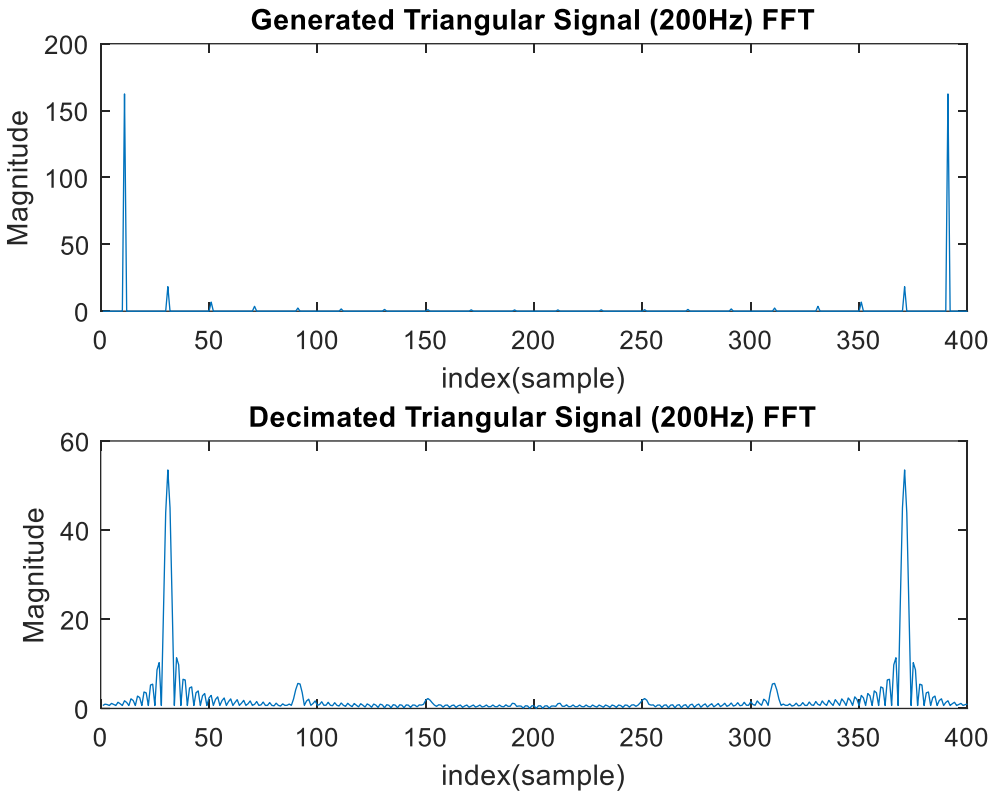
a)2) Plot the input and output in part a)1) in frequency.



a)3) Generate a **200 Hz triangle sequence** in **0-50 ms time range** with **sampling rate 8000 samples/sec**. **Plot** it and output in **time** using the **same time axis**, i.e., time difference between consecutive sequence elements is **1/8000**.



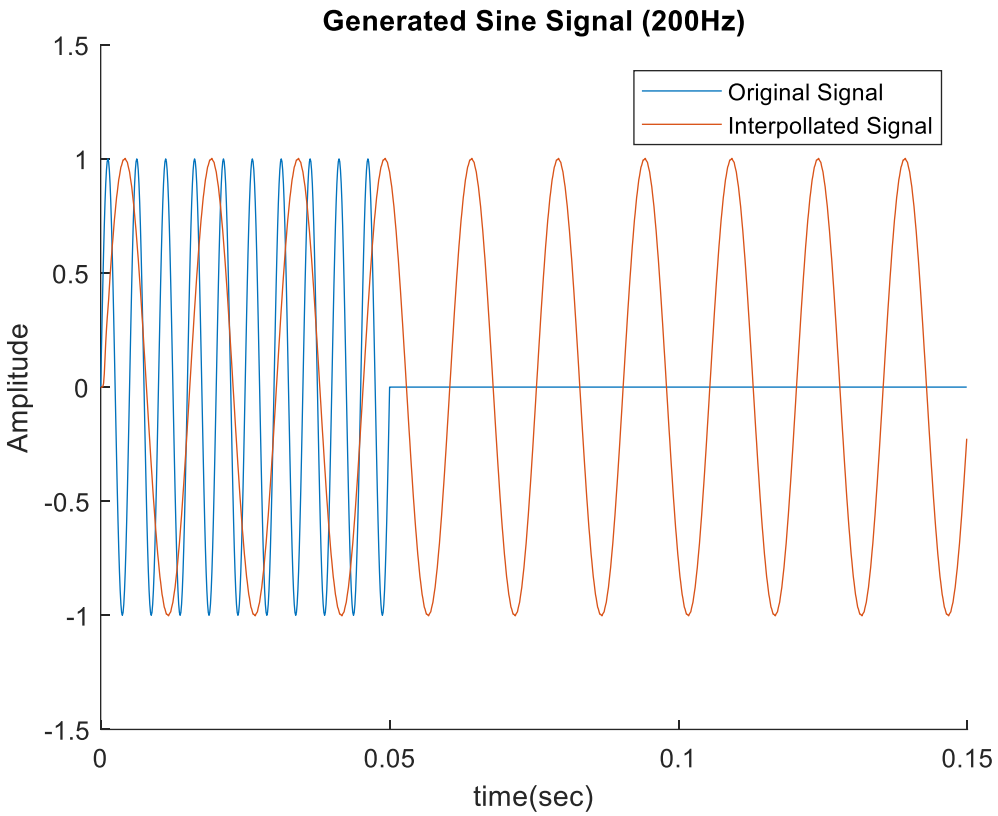
a)4) Plot the input and output in part a)3) in frequency.



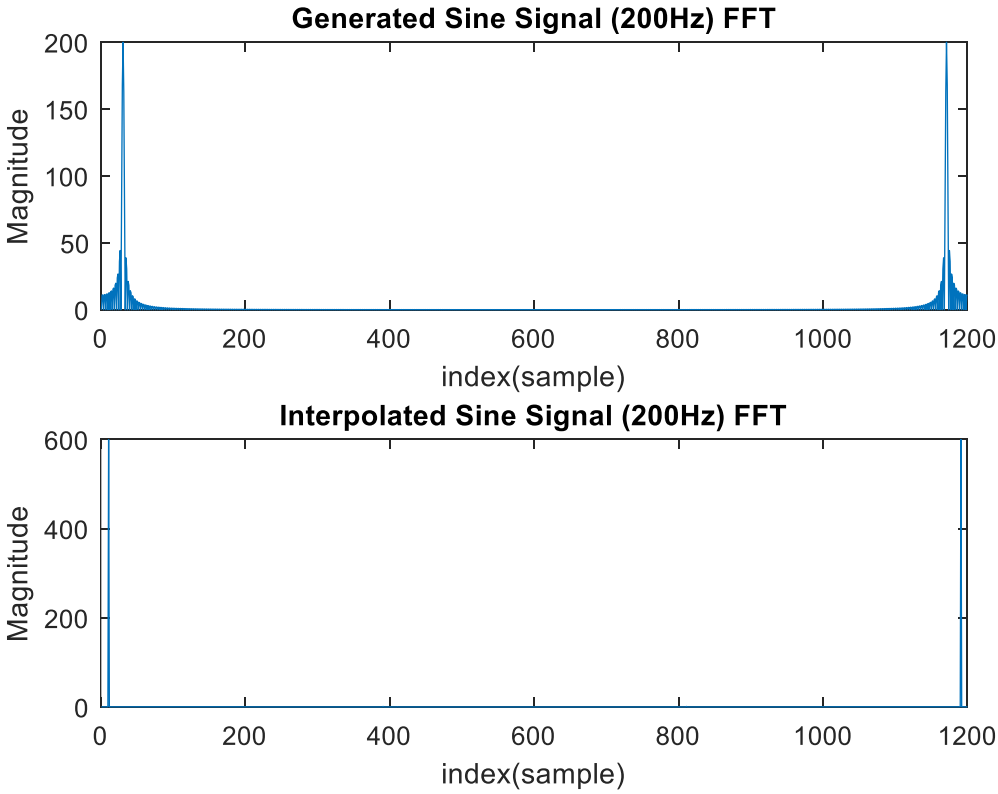
b) Repeat **a)** for **interpolation**.

Take L=3 in the following tasks.

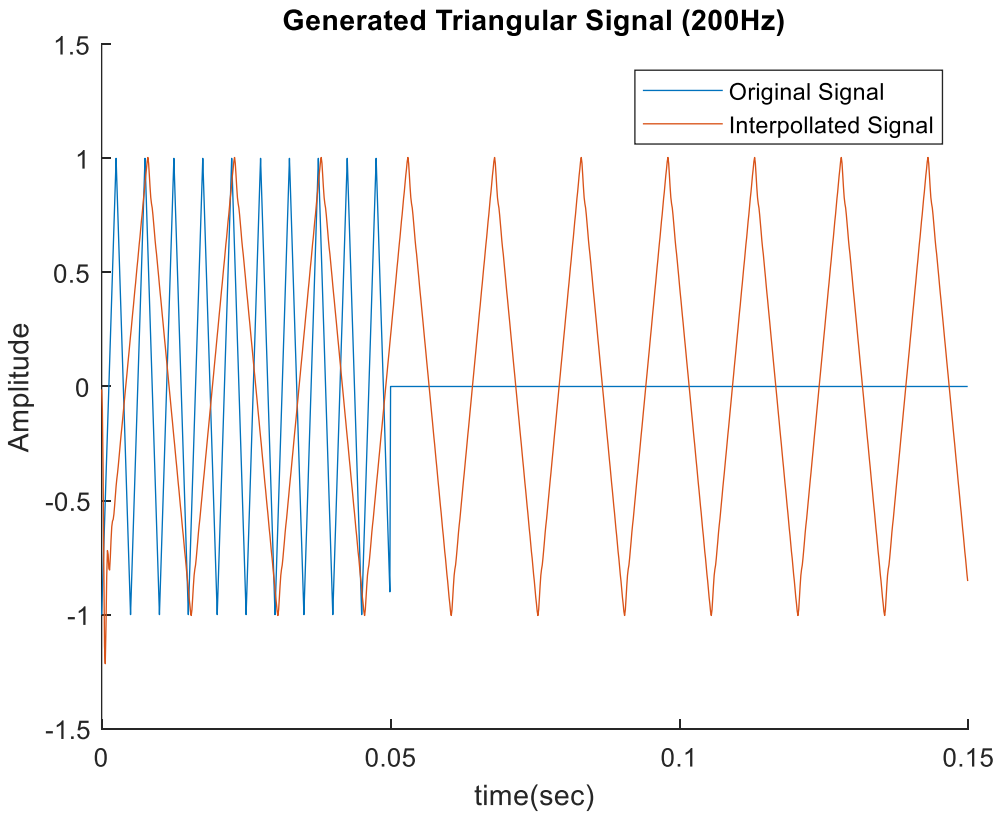
b)1) Take the **sine sequence in part a)1) as input**. **Plot** the **output** in **time** using the **same time axis**, i.e., time difference between consecutive sequence elements is **1/8000**.



b)2) Plot the output in part b)1) in frequency.

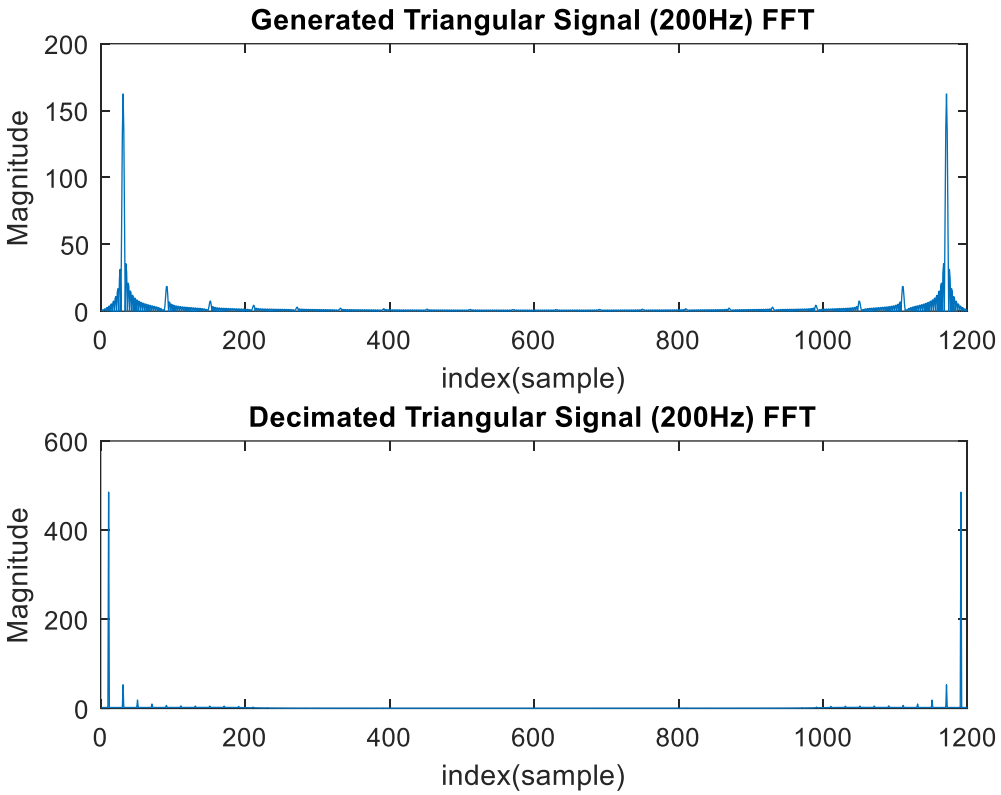


b)3) Take the **triangle sequence in part a)3) as input**. **Plot** the **output** in **time** using the **same time axis**, i.e., time difference between consecutive sequence elements is **1/8000**.



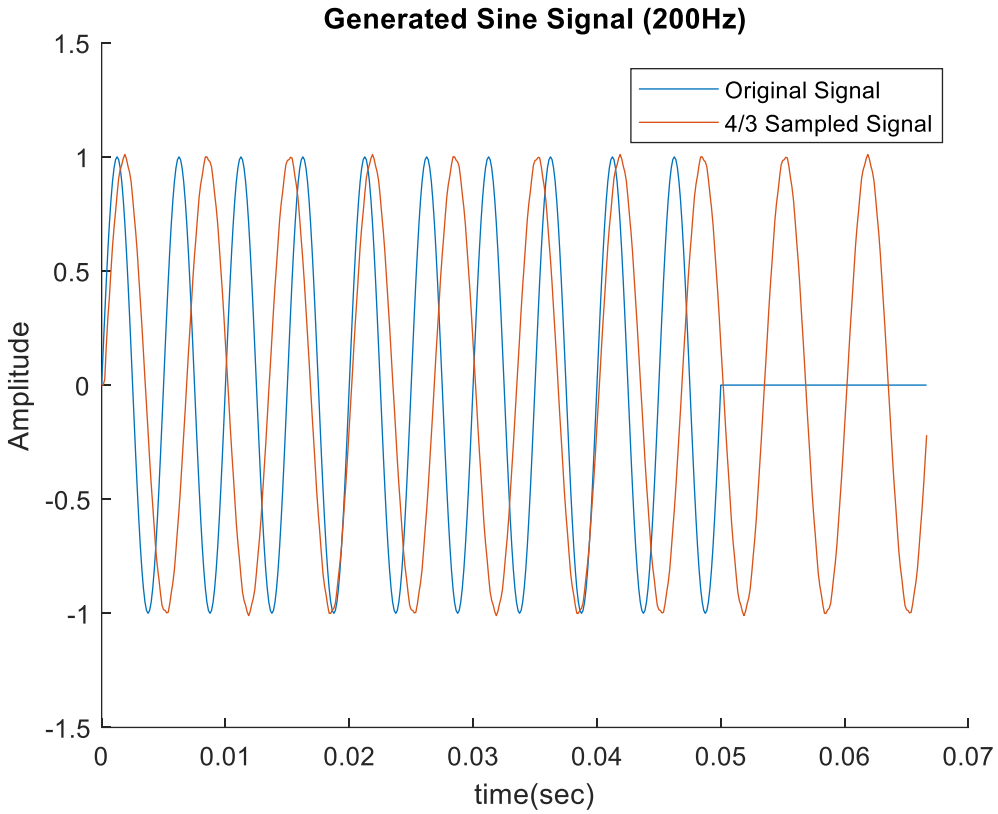
b)4) Plot the output in part b)3) in frequency.

c) Write a code for sampling rate change with a **rational factor**. **Plot** the **input** and **output** both in **time** and **frequency**.

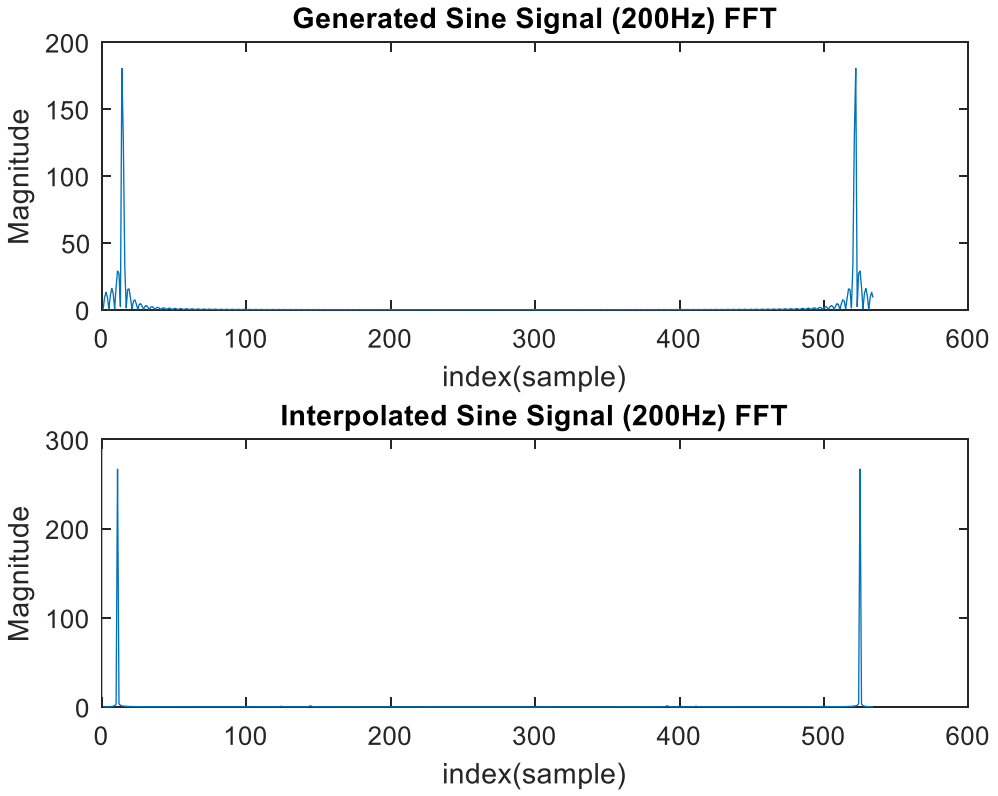


Increase the sampling rate by 4/3 in the following tasks.

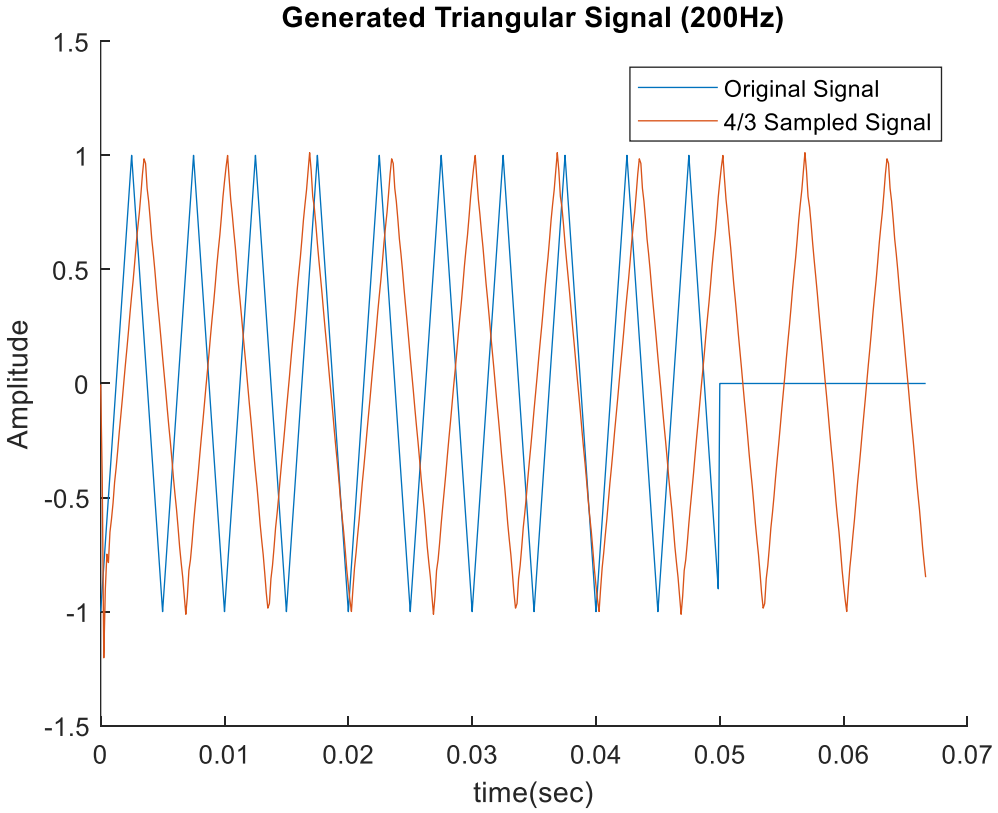
c)1) Take the **sine sequence in part a)1) as input**. **Plot** the **output** in **time** using the **same time axis**, i.e., time difference between consecutive sequence elements is **1/8000**.



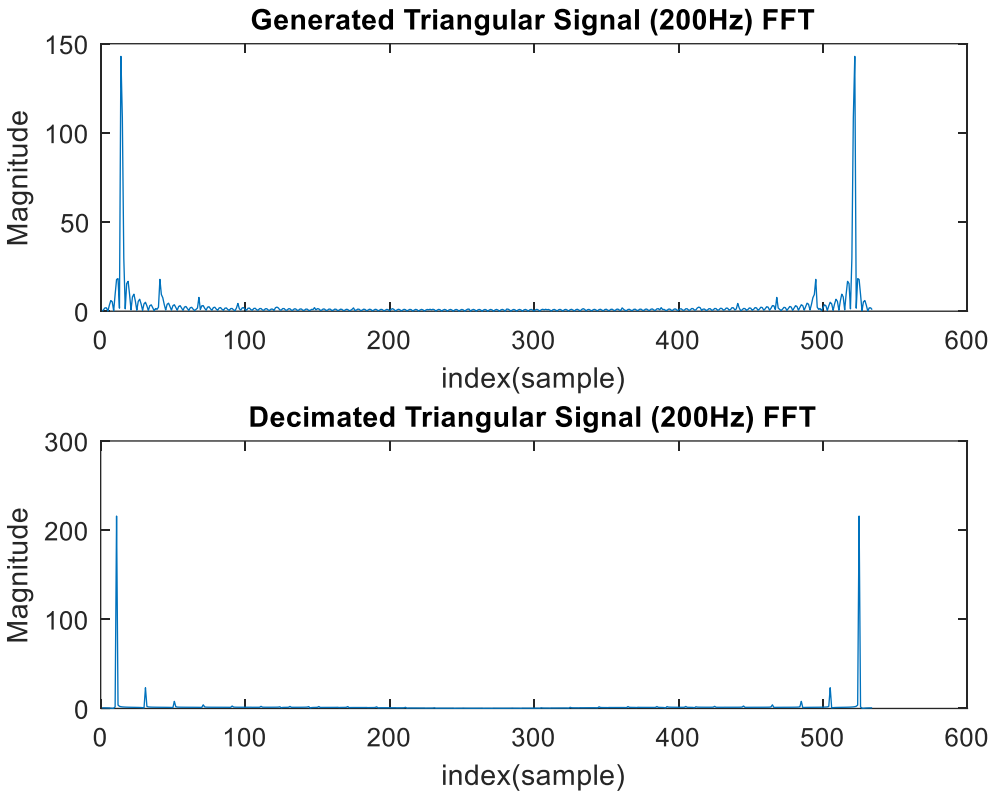
c)2) Plot the output in part c)1) in frequency.



c)3) Take the **triangle sequence in part a)3) as input**. **Plot** the **output** in **time** using the **same time axis**, i.e., time difference between consecutive sequence elements is **1/8000**.

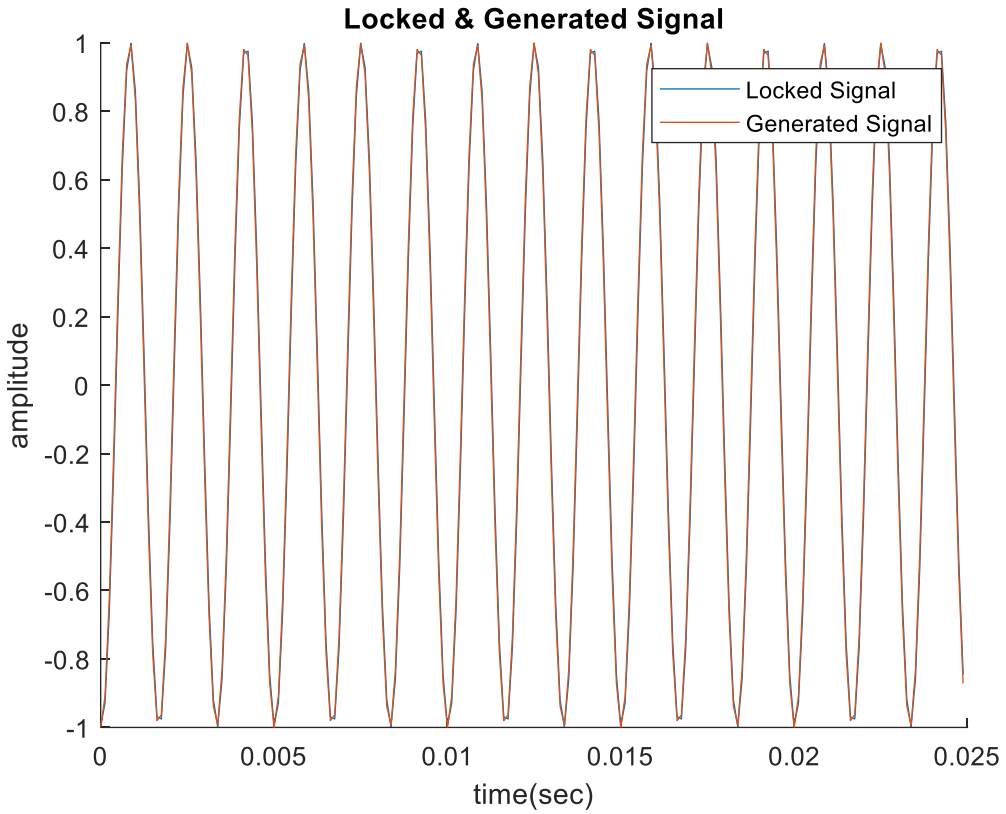


c)4) Plot the **output in part c)3) in frequency**.

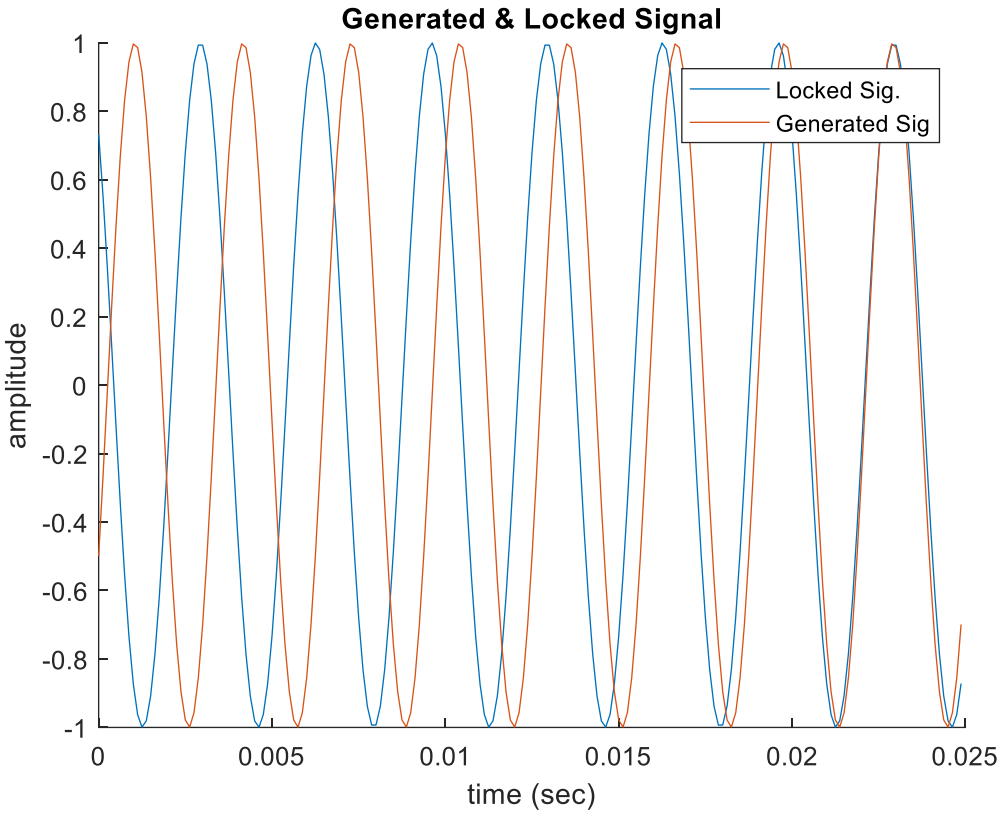


d) Implement a **digital PLL** structure. In this case, assume that there is a **sinusoid** where **you would lock onto its frequency and phase.** You should be able to **change the frequency and phase of this sinusoid.** The second sinusoid is an **internally produced** one. It may have a constant frequency and you can match its frequency with the first sinusoid. For this purpose, you need to **detect the frequency of the first sinusoid by using FFT magnitude.** After the frequency is of the second sinusoid is brought to the lock range of the PLL, PLL works on lock onto the fine frequency and phase.

d)1) Take the **sampling rate as 8000 samples/sec**. Select the **frequency of internally produced sinusoid** as **600 Hz** and create a **sine sequence in 0-25 ms range as input**. **Detect the frequency of this input sinusoid** by using **FFT magnitude and let the internally produced sinusoid have this frequency**. **Plot** the **input and locked signal on the same figure** with **different colors**.



d)2) Repeat d)1) with the **frequency of sine as 300 Hz**. **Observe** whether locking occurs or not.



Since the generated data array is not sufficient enough to estimate frequency, which doesn't involve enough period of the generated signal, we cannot match the generated and locked signals. However, when we increase the time range, they matched perfectly.