APPENDIX – III

Matlab codes used to analyse the data were developed. This code is presented as a set of two files. The variables were stored as in different formats e.g. text file, Excel file and pdf. Specific interfacial area was estimated from gas holdup and d_{32}. This part was done in Microsft's Excel. Hence, it is not in MATLAB's code. Most of the figures were plotted using MATLAB. These codes are not provided but are available on request. The code uses equal spaced font.

Anybody is free to use the codes provided here with acknowledgement to the authors and Indian Institute of Technology (BHU).

MATLAB Code 3.1: 'Acoustic_analysis.m' is used read acoustic file, divide it various segments, applying FFT to each segment, converts to bubble diameter, estimate statistical parameters and d_{32}.

```
% This program analyses acoustic data using short FFT
% Stores Bubble-Sizes into individual files
%       - Nayak, C. and Prof. Verma, A. K.
%       - Indian Institute of Technology(BHU) Varanasi,
INDIA
%       - November 12, 2018
%
% s  = amplitude of acoustic signal, a vector
% fs = frequency of acoustic signal
[s,fs]=audioread(myfile); % Reads Acoustic Signal
filesd=strcat(myfile(1:length(myfile)-4),'_total.txt')
% Here 'filesd' is name of file into which data will be
saved
% Crop the acoustic signal into many segments (each 2048
points)
```

```
index1=floor(length(s)/2048);
% Initialization:-
% sound_sample = vector of segments of acoustic signal
% time_sample  = vector of starting time of sound_sample
% time_step    = interval between two successive data
sound_sample=zeros(2048,index1);
time_sample=zeros(index1);
time_step=2048/fs;
% Cropping of acoustic signal starts
for i=1:index1
    start_data=2048*(i-1)+1;
    end_data=2048*i;
    sound_sample(:,i)=s(start_data:end_data);
    time_sample(i)=time_step*(i-0.5);
end

% Determines 10 prominent peaks separated by 100 points
% peaks    = contains spectral power arranged in a row
% loc_freq = contains frequencies at which peaks exist

% Allocating space for maximum 10 peaks
peaks=zeros(index1,10);     % Amplitude of peaks
loc_freq=zeros(index1,10);  % (Location i.e. time of peaks)
for i=1:index1
    sound_x=sound_sample(:,i);
        % FFT of acoustic signal of length 2048 data points
        y=fft(sound_x);
        y(1)=[];
        n=length(y);
        power=abs(y(1:n/2)).^2;
        nyquist=fs/2;
        freq=(1:n/2)/(n/2)*nyquist;
        % plot(freq,power);
    [peaks_power,peaks_loc] =
findpeaks(power,'minpeakdistance',100);
    for j=1:10
        if j>length(peaks_power)
            peaks(i,j)=0;
            loc_freq(i,j)=0;
        else
        peaks(i,j)=peaks_power(j);
        loc_freq(i,j)=peaks_loc(j);
        end
    end
end
```

```
% Join data of peaks and loc_freq to give peaks_data
% time_sample = time, s (column - 1)
% peaks       = column 2-11, zero means no peak
% loc_freq    = column 12-21, zero means no corresponding
freq
peaks_data=[time_sample(:,1) peaks loc_freq];
time_peak=peaks_data(:,1);            % Getting time vector
freq_peak=peaks_data(:,12:21);        % Getting frequencies
% Now convert the matrix into vector
freq_all = reshape( freq_peak.' ,1,numel(freq_peak));
% Removing zeros and all insignificant frequencies
% freq_true contains all frequencies
freq_true=[];
for counter=1:length(freq_all)
    if freq_all(counter)<120
        continue
    else freq_true=[freq_true freq_all(counter)];
    end
end
% convert frequencies to bubble size (bubble_size)
% Let den    = density of fluid
%       h    = liquid height above the sensor
%       P    = Static pressure
% bubble_dia = vector of bubble sizes
Den=1000;h=0.0582; % Change this line for each file
g=9.81;P0=den*g*h;gamma=4.186;
factor=(3*gamma*P0/den)^0.5/(2*3.14159);
bubble_dia=2*factor./freq_true;
% Saving the file
save(filesd,'bubble_dia','-ascii');
clear all
%
% Estimation of Statistical parameters
Moment_1=mean(bubble_dia);
Moment_2std=std(bubble_dia);
Moment_2var=var(bubble_dia);
Moment_3=skewness(bubble_dia);
Moment_4=kurtosis(bubble_dia);
% Sauter mean diameter
% d32 = Sauter mean diameter, m
sum_v=0;sum_a=0;
d32=mean(bubble_dia.^3)/mean(bubble_dia.^2);
analysed_data=[M_1 M_2std M_2var M_3 M_4 d32]
% Here analysed_data = calculated values as vector
```

MATLAB Code 3.2: ‘db_distribution.m’ to classify bubbles into classes separated by 0.0005 m.

```
% Reads all bubble diameters from text files
% and classifies them in intervals of 0.5 mm
%       - Nayak, C. and Prof. Verma, A. K.
%       - Indian Institute of Technology(BHU) Varanasi,
INDIA
%       - November 12, 2018
%
% Read ‘myfile’ containing sizes of all bubbles into dbi
dbi=dlmread(myfile);
% Let      dbi_max= maximum value of dbi
%          dbi_min= minimum value of dbi
%          Nb_all = Total number of bubbles
dbi_max=max(dbi);dbi_min=min(dbi);Nb_all=length(dbi);
% Defining classes for bubbles at an interval of 0.0005 m
% Let      db_group(i) being the ith class
for i=1:51
db_group(i)=0.0005+(i-1)*0.0005;
end
for i=1:50
Nb(i)=0;    %Initialization of Nb
end
clear i;
for j=1:50
for i=1:Nb_all
if dbi(i)>db_group(j) & dbi(i)<db_group(j+1)
Nb(j)=Nb(j)+1;
end
end
end
% Saving the file
% Note: file1 is myfile after removing extension
fileNb=strcat(file1,'_Nb.txt');
save(fileNb,'Nb','-ascii');
```