**Processing of OsCaR flume measurements**

Deriving resuspension thresholds from turbidity timeseries and / or visual observations

**Part 1 -- loading datafiles & setting up filters**

close all

clear variables

Copy the path with the Delf measure files

cd('Folder with .raw files')

Load observed timepoints (in seconds after start flume) of incipient motion. This should be a textfile with a single column

t = load('times.txt');

Load Delft measure files. For these to work in Matlab, make sure to change the comma's into periods first

files = dir('\*.raw');

Appending some constants

freq = 25; % frequency of your measurements

[A, B] = butter(5,0.05/(freq/2),'high'); % low frequency filter for pressure data

[C, D] = butter(5,0.5/(freq/2),'low'); % noise filter for pressure data

[E, F] = butter(5,0.1/(freq/2),'low'); % high frequency filter for turbidity timeseries

calibration\_pressure = 690; % calibration factor for pressure data for turning voltage into cm water level \*SET THIS TO 86 FOR THE DEFAUNATED SEDIMENTS\*

calibration\_turbidity = 300; % calibration factor for turbidity data for turning voltage into NTU

**Part 2 -- assessing the timeseries**

Extracting relevant data from files

for i = 1:length(files)

names{i,1} = files(i).name; % obtaining filenames

data{i} = load(files(i).name); % obtaining data

% appending a time axis

timeax{i} = [1/freq:1/freq:length(data{i})/freq]'-1/freq;

timeax{i} = round(timeax{i},2);

% filtering pressure and turbidity data

pressure{i} = filtfilt(A,B,data{i}(:,1));

pressure{i} = filtfilt(C,D,pressure{i});

turbidity{i} = filtfilt(E,F,data{i}(:,2));

end

Extracting peak orbital velocity from pressure timeseries

for i = 1:length(files)

pressure{i} = pressure{i}.\*calibration\_pressure; % water pressure in cm

dpdt{i} = gradient(pressure{i})./(1/freq); % pressure gradient

[upeak{i},locs{i}] = findpeaks(dpdt{i},'MinPeakHeight',5); % peak orbital velocity

[apeakpos{i},locs2{i}] = findpeaks(pressure{i},'MinPeakHeight',5);

[apeakneg{i},locs3{i}] = findpeaks(-pressure{i},'MinPeakHeight',5);

T(i) = (locs{i}(end)-locs{i}(end-9))./9./freq;

end

**Part 3 -- calculating the resuspension threshold from the rate of increase in turbidity**

Turbidity increases when sediments are resuspended. The rate of sediment resuspension determines the rate of the increase in turbidity. Hence, the resuspesion threshold can be defined as the point where the turbidity increases at a sufficient rate. This can be compared with visual observations.

for i = 1:length(files)

turbidity{i} = turbidity{i}.\*calibration\_turbidity; % turbidity in NTU

% increase in turbidity per second over a 1 minute moving window (to filter out noise)

dTdt{i} = movmean(gradient(turbidity{i})./(1/freq),60\*freq);

end

Assessment

for i = 1:length(files)

figure;

title(names{i}(1:end-4))

yyaxis left

hold on

plot(timeax{i},dTdt{i},'-','Color',[0.2 0.5 1])

plot([min(timeax{i}) max(timeax{i})],[0.06 0.06],'-k')

%plot([t(i) t(i)],[0 0.3],'-k')

%text(t(i) - max(timeax{i}/45),0.22,'visual','Rotation',90)

ylabel('Sediment resuspension rate (NTU s^{-1})')

set(gca,'YColor',[0.2 0.5 1])

axis([0 max(timeax{i}) 0 0.3])

yyaxis right

plot(locs{i}./freq,upeak{i}./100,'-','Color',[1 0 0.2])

ylabel('u\_{max} (m s^{-1})')

xlabel('Time (s)')

set(gca,'YColor',[1 0 0.2])

axis([0 max(timeax{i}) 0 0.8])

end

Figures show peak orbital velocity and sediment resuspension rate (as an increase in mud concentration).

In sand there is no material to be suspended. Hence, only visual observations can be used. As soon as mud is present in the sediment mixture, even at low concentrations, there is a clear turbidity signal from which the resuspension threshold can be derived.

In this case, observations generally match with a sediment resuspension rate exeeding ~0.06 NTU/s. Hence, we choose this as a threshold. This number depends on the properties of the mud fraction, and should therefore be estimated based on observations from the timeseries and how it compares to vissual observations of resuspension.

t\_lim = 0.06;

for i = 1:length(files)

% we only consider turbidity after the first minute, to remove possible sensor errors at start of experiments (see graphs)

dTdt{i}(1:60\*freq) = 0;

% when sediment resuspension has reached 2e-4 NTU/s, the velocity threshold can be determined automatically

if max(dTdt{i}) > t\_lim

umax(i,1) = median(upeak{i}(locs{i} > find(dTdt{i} > t\_lim,1) & locs{i} < find(dTdt{i} > t\_lim,1)+(60\*freq)));

umax(i,3) = 1;

% if not, visual observation is used

else

umax(i,1) = median(upeak{i}(locs{i} > find(timeax{i} == t(i),1) & locs{i} < find(timeax{i} == t(i)+60)));

umax(i,3) = 2;

end

umax(i,2) = median(upeak{i}(locs{i} > find(timeax{i} == t(i),1) & locs{i} < find(timeax{i} == t(i)+60)));

% we also calculate the orbital magnitude

amaxpos(i,1) = median(apeakpos{i}(locs2{i} > find(timeax{i} == t(i),1) & locs2{i} < find(timeax{i} == t(i)+60)));

amaxneg(i,1) = median(apeakneg{i}(locs3{i} > find(timeax{i} == t(i),1) & locs3{i} < find(timeax{i} == t(i)+60)));

end

amax = amaxpos+amaxneg;

dlmwrite('umax.txt',umax,'delimiter','\t');

dlmwrite('amax.txt',amax);

dlmwrite('waveperiod.txt',T');