Escola de Camins
Escola Tècnica Superior d'Enginyeria de Camins, Canals i Ports
UPC BARCELONATECH

Experimental study in bed erosion induced by twin propellers in unconfined conditions

Treball realitzat per:
Pau Bayona Visiedo

Dirigit per:
Anna Mujal-Colilles i Xavier Gironella

Màster en:
Enginyeria de Camins, Canals i Ports

Barcelona, 23 de juny de 2016

Departament d'Enginyeria Hidràulica, Marítima i Ambiental

TREBALL FINAL DE MÀSTER

Contents

List of Figures

List of Tables

Abstract

Scouring processes due to the navigation and manoeuvring of ships can generate stability problems to harbour's structures. At the same time, the eroded material is settled in other areas, reducing the water level. Theoretical equations used are far from reality and only work for a single propeller. With the experiments performed at the Marine Engineering Laboratory (LIM/UPC) in LaBassA flume, new formulas are proposed to work with them to analyse the erosion effect in seabed provoked by twin propellers working at different revolutions under de scenario of unconfined conditions. The data quality is tested to examine if a new experimental system to analyse the scouring effects can be used in further experiments. A new formulation that relates erosion variables with the revolutions of the propellers system is added trough MATLAB self-made scripts.

Keywords: scouring hole, twin propellers, unconfined conditions

1 Introduction

The evolution of society and its necessities have increased the use of products all around the world. This, attached to the construction of bigger means of transportation has lead to a constant growth of the marine transportation industry. Also, cities with high rates of tourism have been demanding regular cruise lines in their harbours. Harbours and many marine infrastructures are being outdated by these changes of means of transportation that the marine industry is experiencing.

Marine transportation evolution has leaded to more powerful and bigger ships. As a consequence, the ship's draft is closer to seabed and as the engines are more powerful, the effect of those to the seabed is generating problems in harbours. This erosion of the seabed causes stability problems to the docking platforms, during the docking and undocking manoeuvres. These stability problems can, eventually, produce the collapse of the structure. Also, the eroded sediment is settled in other areas of the harbour reducing the water level of some zones of the platform. This phenomenon is even more notorious in regular cruise lines, since these types of vessels do not need tugboats and they perform regular docking manoeuvres.

There are few analytic tools to foresee the erosion in seabed generated by ship propellers. There is little experimentation in this topic and even less in the case with twin propellers. Nowadays, the equations used to compute the future erosion are based on theoretical equations that are far from reality and experimental studies using one propeller as the propulsion system.

The first parameter needed to analyse the seabed erosion is the efflux velocity. During the past century, several authors have obtained theoretical formulas based of the efflux velocity (Hamill (1987); Robakiewicz (1987)). The efflux velocity is defined as the mean axial velocity at the outlet of propeller systems without rudder, keel and wall influence. Bed velocity has always been expressed as a function of efflux velocity and is used to obtain the maximum scouring depth caused by ships propulsion systems. Therefore, all equations presented so far in literature to estimate the maximum bed erosion, are based in theoretical estimations on the efflux velocity (Hamill et al. (1999); Hong et al. (2012)).

Theoretical efflux velocity can be obtained from the efflux velocity momentum (Verheij (2010)) or from the efflux velocity mass equation (Blaauw and Van de Kaa (1978)). There are also some experimental efflux velocity equations developed after the momentum equation(Hamill (1987); Stewart (1992)).

Since all the formulas are based in single propeller studies, PIANC (2015) proposes two different approaches for twin propellers, linear and quadratic. The former is more realistic in terms of erosion but the latter is closer to real ship motion.

2 Objectives

Experiments in this field performed in laboratories are sparse and with a single propeller. Having the opportunity to work with data from twin propellers experiments opens a new line of investigation to analyse the effects of twin propellers in seabed.

The aim of this thesis is to describe the experimental results of scouring processes reproduced in a laboratory for twin propellers. This processes can be confined or unconfined. The unconfined conditions are applied when there is no interaction with any quay wall. This happens when the ship is travelling inside the canal or when the ship approaches a docking station. In that last case, when the ship is close to the dock, conditions change to confined. The scouring holes are caused by bow thrusters and this can happen with docking or undocking manoeuvres. In this thesis the data provided was in unconfined conditions.

The aim of this thesis is achieved by following two specific goals:

1. Experimental data analysis: The need of simpler and affordable ways to read the data leads to the usage of equipment not specifically created for these tasks. This thesis also studies if the equipment used in the experiments was acceptable.
2. Relate some of the propellers' variables with the characteristics of the scouring hole in different velocities all along time.

3 Experimental set-up

3.1 Model set-up

All physical experiments were performed at LaBassA, a rectangular concrete tank of 12.5m long, 4.6m wide and 2.5m high (see Figure 1). Two four-blade propellers with a diameter of $D_p = 25.4cm$ were located at the lower part of a metallic structure that was hanging from a railroad at one end of LaBassA with a distance between them of $a_p = 58cm$. The geometric scale of the model was 1/25 and the temporal scale was the square root of the geometric scale; 1/5.

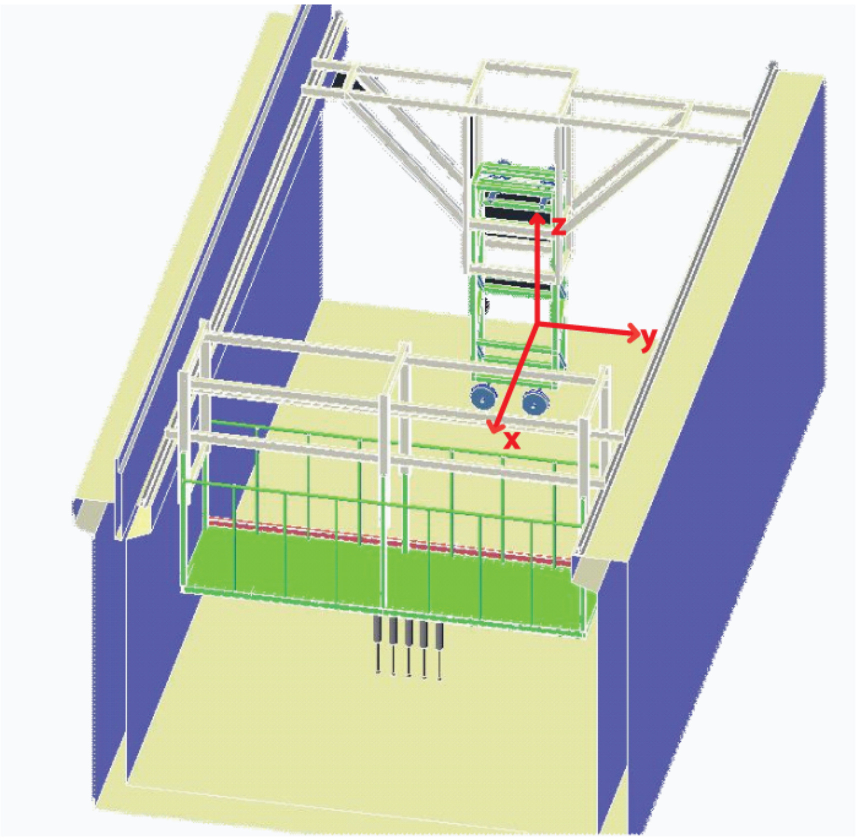


Figure 1: Sketch of LaBassA tank.

Each of the propellers was connected to its motor through a transmission chain with a gear. The air trapped in the chain was kept in a folding box to avoid the suction effect in the upstream zone of the propellers. Also, this folding box helped to simulate, with all its limitations, the hull of a vessel. Propellers were rotating in opposite directions: the left propeller rotated clockwise and the right propeller anti-clockwise, as shown in Figure 2.

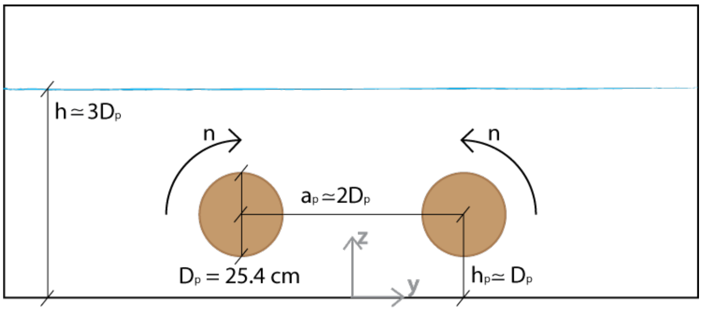


Figure 2: Sketch of the main propellers system.

The distance between the bottom of the tank to the centre of the propeller was $h_p = 26cm$. The centre of the propellers was fixed at $1.17m$ from the wall of LaBassA and the propellers were $30cm$ away from the folding box to allow its correct working. The centre of reference is located at the axis of symmetry between the two propellers at the bottom of LaBassA (see Figure 3).

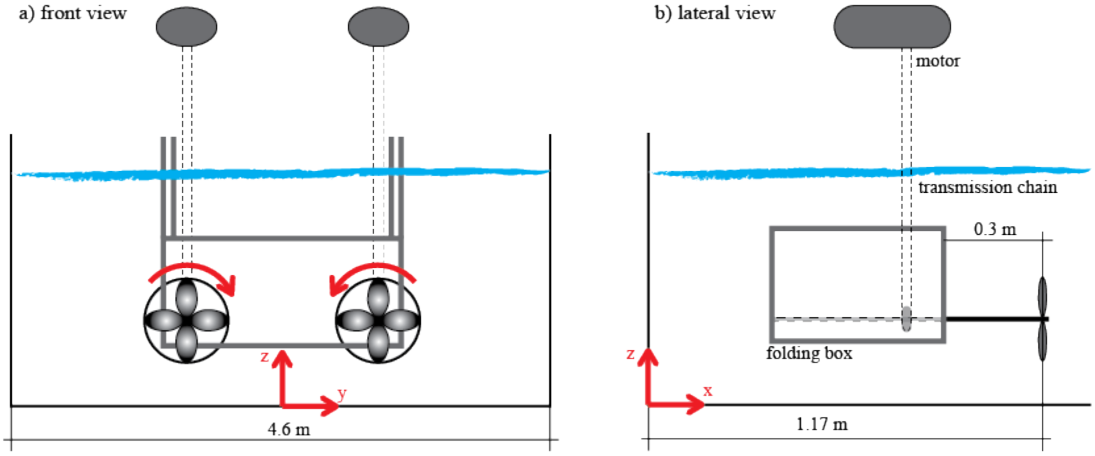


Figure 3: Diagram of the experimental set-up.

The water depth was kept constant at $H_w = 70cm$. There was a sediment layer of $H_s = 52cm$. The grain size distribution of the sediment layer was $D_{50} = 250\mu m$ and $D_{90} = 375\mu m$.

Initial tests were performed to verify that the thrusters were located far from the opposite wall in order to study unconfined scenarios (Mujal-Colilles et al. (2015)). Lateral walls were also considered far enough to let the propeller jet develop freely and obtain a minimum influence of the convective cells created in the tank. Three different rotation velocities ($n = 300, 350, 400rpm$) were used for the non-confined scenario.

A mechanized arm with three degrees of freedom was suspended from a footbridge paced at the same railroad as the propellers' structure. Three photoelectric sensors were used to locate the position in the $x - y - z$ coordinate system of the mechanized arm. The data was recorded at a frequency of $32Hz$.

The photoelectric sensors used to read the position of the point that was being studied, were Efector200-O1D100. These sensors were not meant for this experiment. However, they were more economic than the specialised ones, as well as they had a better implementation. Also, with these sensors the tank had not to be emptied to read the data points. The sensor for z component was placed inside a cylindrical Perspex tank in order to acquire data without flowing out the water in the tank, as well as, to not receive two different lectures: one from the height of sediment and another one from the reflection of the water. The working range of the sensors goes from $0.2m$ to $10m$, so it fit perfectly the tank. It had to be kept in mind that the manufacturer of the sensors gives them a $20mm - 30mm$ lecture error range.

Before every data collection session, the sensors were preheated to remove any external factors that could effect the measurements. Also, sensors had to measure two specific points inside the tank with a known height to calibrate the photoelectric sensors.

Scouring holes were measured after scanning the sediment bed with 13 longitudinal profiles and 12 transversal profiles (see Table 1).

Table 1: Position of scanning profiles.

Longitudinal		Transversal	
Name	y/a_p	Name	x/D_p
Y_5	-3.0	$X0$	0.5
Y_4	-2.5	$X1$	1.5
Y_3	-2.0	$X2$	2.7
Y_2	-1.5	$X3$	3.7
$H1$	-1.0	$X4$	4.7
Y_1	-0.5	$X5$	5.7
$Y0$	0.0	$X6$	6.6
$Y1$	0.5	$X7$	7.7
$H2$	1.0	$X8$	8.6
$Y2$	1.5	$X9$	9.6
$Y3$	2.0	$X10$	10.6
$Y4$	2.5	$X11$	11.6
$Y5$	3.0		

The scouring action was simulated by performing sequences of 5 hours run, except for the first 10 minutes, which tried to reproduce the scaled time of a docking manoeuvring. As said before, the time scale was 1/5 so the first 10 minutes were 50 minutes of real time, approximately the amount of time a ship takes to dock and undock in the harbour basin.

3.2 Data quality

The data resulting from lectures of the photoelectric sensors had noise, mainly because of the vibration of the mechanized arm when this was moved during the data acquisition process.

The cleaning of the noise involved an external software. It was decided to work with MATLAB. Files obtained from sensors were a *.dat* file. With the assistance of self-made scripts (see Appendix A), it was converted to *.mat* files to be able to work further with them and apply different tests to see how good the quality of the data points was.

To obtain the different profiles, the mechanized arm performed two movements: one was moving the mechanized arm itself, to get the longitudinal profiles; and the other one was achieved by moving the lasers locating the $y - z$ positions at the mechanized arm, to get the transversal profiles. This second movement was achieved by installing a drill connected to a chain that moved the lasers.

As the mechanized arm was hanging from a railroad, every time it was moved, it also gave noise to the data that was being recorded at that time. Also, in the transversal profiles, there was even more noise, because of the vibration of the drill.

Before filtering the noise, individual positions of the laser were recorded to obtain the PDF distribution. Given two arbitrary points, data was measured for 30 minutes in each point, getting lectures at 20, 25 and 30 minutes.

Table 2: Data statistics: mean and standard deviation.

	20min	25min	30min	20min	25min	30min
	x component			x component		
Mean	8.372	8.371	8.382	8.618	8.556	8.529
Standard deviation	0.166	0.132	0.142	0.101	0.145	0.123
	y component			y component		
Mean	1.724	1.736	1.738	1.722	1.736	1.732
Standard deviation	0.113	0.135	0.144	0.099	0.145	0.116
	z component			z component		
Mean	3.559	3.632	3.541	4.548	4.468	4.519
Standard deviation	0.028	0.026	0.030	0.029	0.023	0.028

As can be seen in Table 2, the standard deviation of the z component is substantially lower than the other components. This is due to the fact that $x - y$ components have more noise because the laser has to travel greater distances to get the lecture.

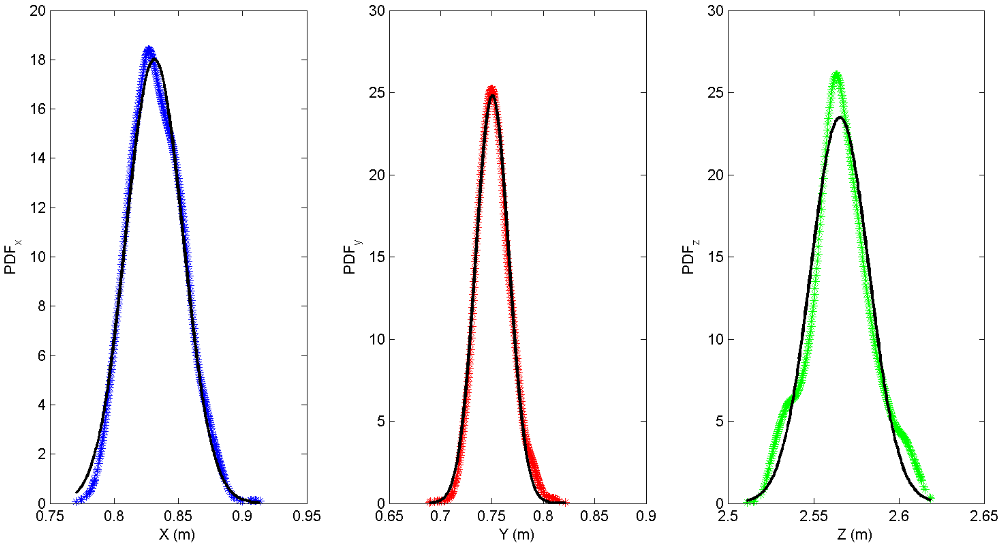


Figure 4: Distribution of data points and comparison with a normal distribution.

The distribution of the data points follows a normal distribution as expected (see figure 4). It is safe to assume that all data follows the same distribution. Consequently, to clean up the noise, two confidence intervals were proposed: 80% of the central points ($\mu \pm 1.28155\sigma$) and 90% of the central points ($\mu \pm 1.64485\sigma$).

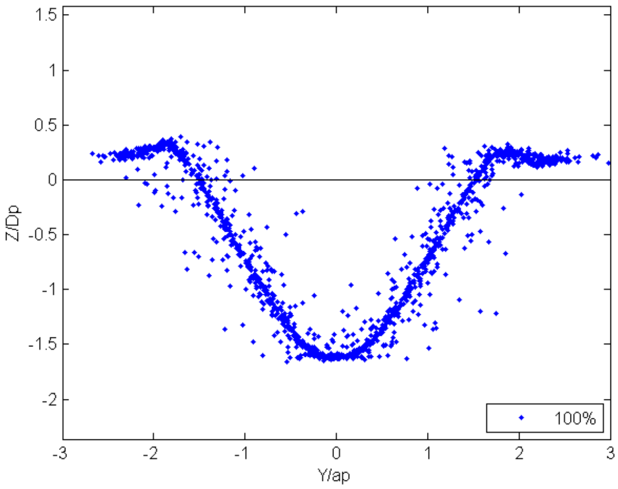


Figure 5: Transversal profile $X5$ at $300rpm$ with 100% of the data points.

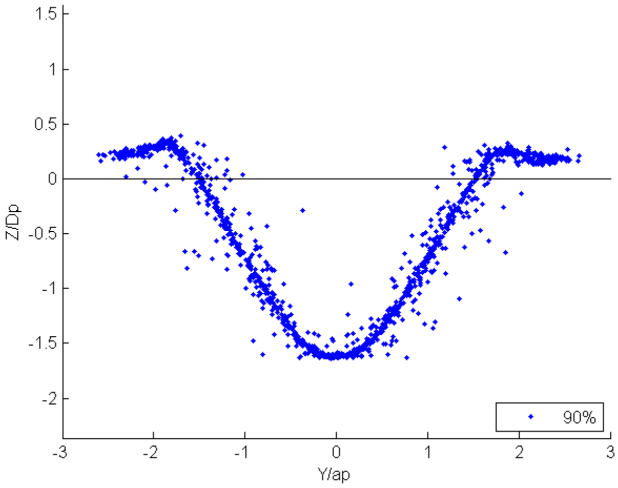


Figure 6: Transversal profile $X5$ at $300rpm$ with 90% of the data points.

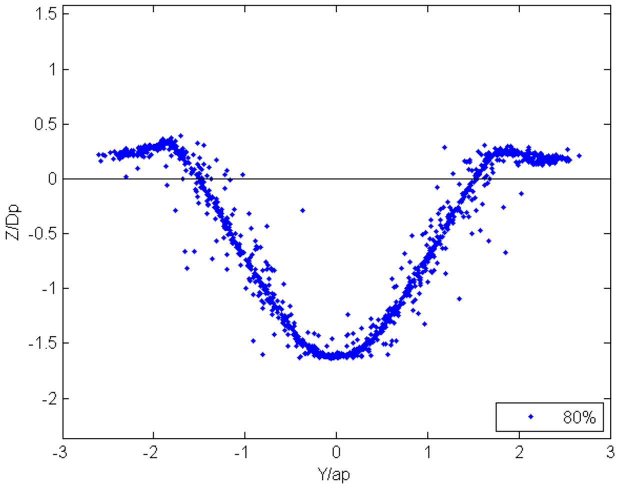


Figure 7: Transversal profile $X5$ at $300rpm$ with 80% of the data points.

Comparing Figures 6 and 7 with Figure 5, the data of the former is cleaner, but there is still some points far away from the central line that cause unrealistic results. Also, it is not good to reduce the confidence interval from 80% to lower values such as 50% or 60%, because the final data stops being a representative sample.

The other possible outcome was to implement a mobile mean in all the profiles gathered from the lasers. Knowing that the lasers had a sampling frequency of $32Hz$ and following Nyquist theorem, which states that "to be able to replicate with accuracy the shape of a wave, it is needed that the sampling frequency is greater than the twice the maximum frequency to be sampled" (Nyquist (1928)); the mobile mean cannot have more than 16 data points.

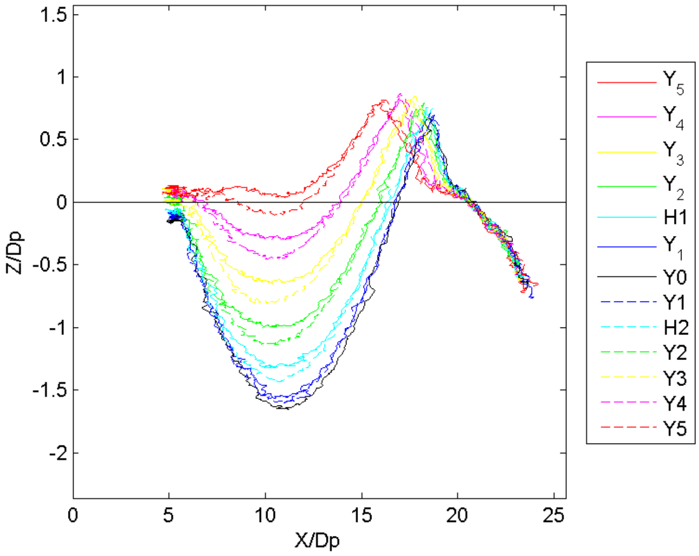


Figure 8: Longitudinal profiles measured after 10 hours running at $300 rpm$.

Figure 8 shows longitudinal profiles after using a mobile mean of 16 points was used to clean up the data. As it can be seen, there was still noise, so an assumption was made to surpass the limit of 16 data points on the mobile mean. The output of the mobile mean was not reproducing $16Hz$ signals anymore, but the data presented herein does not need such a detailed profile. Therefore, higher frequency mobile means were tested.

As stated before, transversal profiles had more noise due to the vibration from the mechanized arm and from the drill, whereas longitudinal profiles only had noise from the vibration of the mechanized arm. Also, data retrieving for longitudinal profiles was faster than for transversal profiles, as for the former it was done by manually pushing the mechanized arm, whereas the later was done throughout the drill. This lead to more data for each point in the transversal profile, which involved to more noise to the data.

Two different mobile means were considered, one for longitudinal profiles and one for transversal profiles. The mobile mean for transversal profiles had twice as much points as the mobile mean for longitudinal profiles.

Two different scenarios were approached: using 32 and 16 points for the mobile mean (transversal and longitudinal profiles, respectively) and using 64 and 32 points.

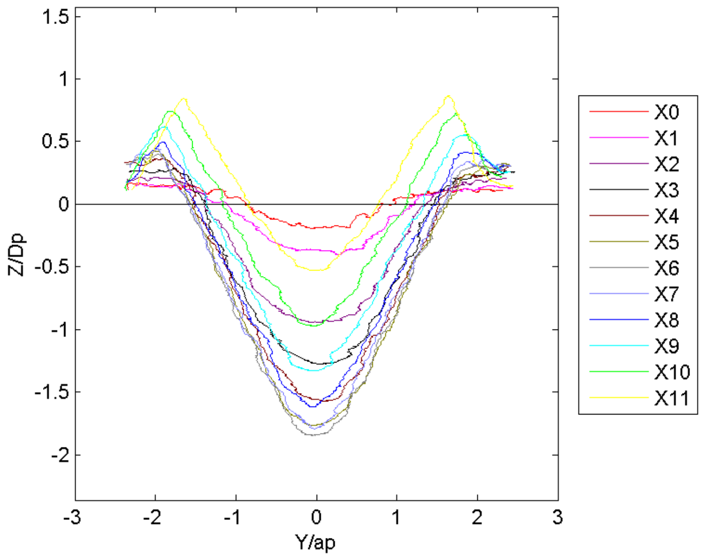


Figure 9: Transversal profiles with 32 points in mobile mean at 300rpm.

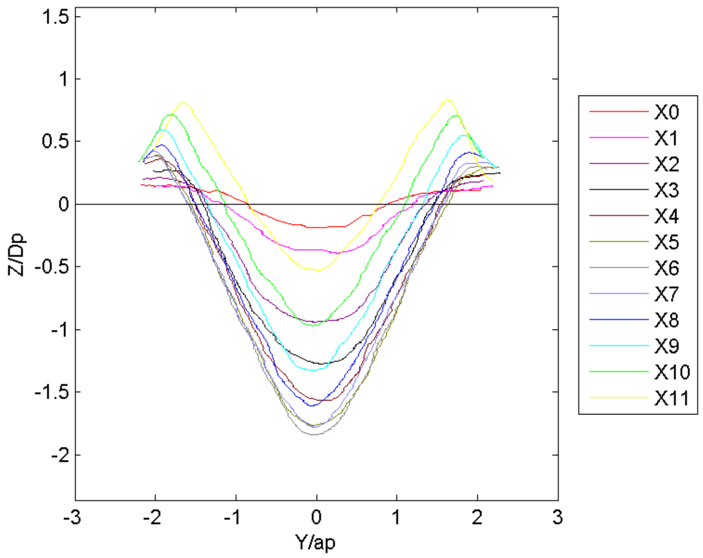


Figure 10: Transversal profiles with 64 points in mobile mean at 300rpm.

Looking at Figures 9 and 10, where the plot of the transversal profiles after 15 hours running at 300rpm is shown, there is a subtle difference in them. In Figure 10, lines are smoother than in Figure 9. Knowing that the representative sample of the data was lost due to working with mobile means higher than 16 points, the chosen pair of points for the mobile mean was 64 points for transversal profiles and 32 points for longitudinal profiles over the 32/16 pair.

4 Experimental results

The data analysis was made through self-made scripts of MATLAB. Results found after analysing the experiments revealed that the stationary time was not reached before 20 hours run. However, experiments were stopped after 20 hours run because at some points of the sediment layer, the whole sediment was totally eroded.

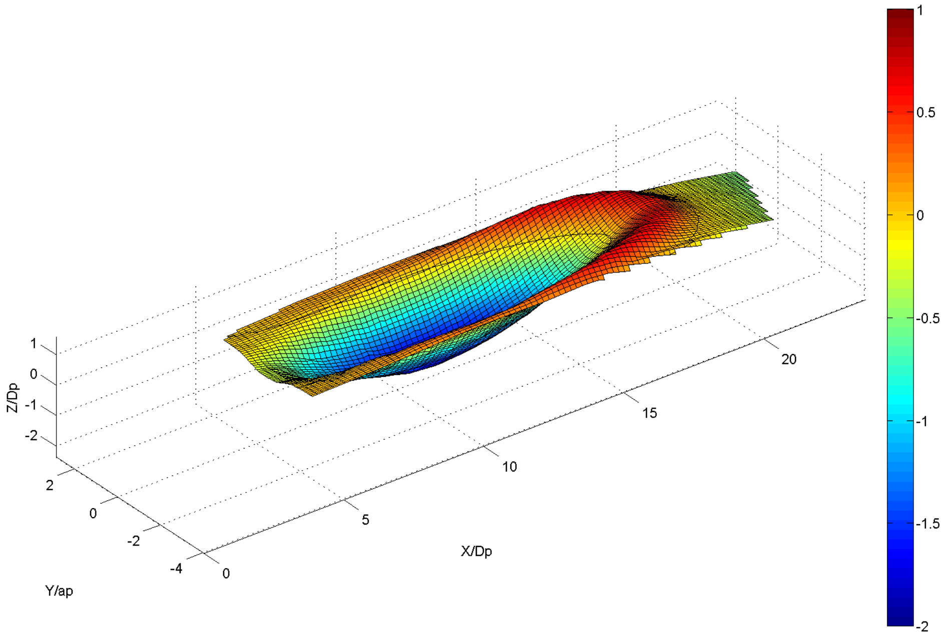


Figure 11: Scouring hole after 15 hours running at $300rpm$.

Figure 11 plots a 3D rendering reproduction of the scouring hole created by the propellers running for 15 hours at $300rpm$. The scouring pattern turns out to be almost symmetric since the rotating effect for one propeller experiments is compensated with the second propeller (Hamill (1988)).

As stated before, the experiments were stopped after 20 hours running because of some areas having the whole sediment layer eroded. This can be seen in Figure 12, that shows the evolution of the centreline when the propellers are running at $400rpm$. The 20 hour plot line is horizontal between 10 and 15 x/D_p.

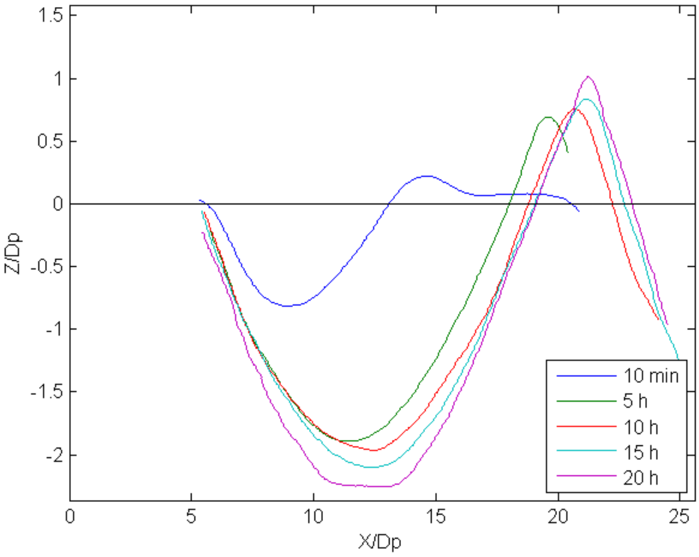
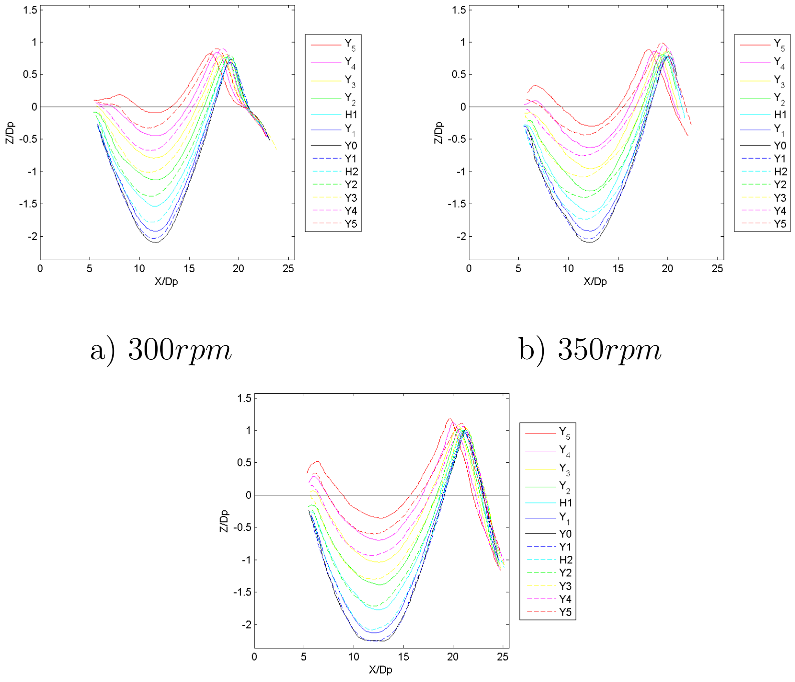


Figure 12: Evolution of the centreline $Y0$ at $400rpm$.

Figure 13 shows the longitudinal profiles for all three scenarios: 300, 350 and $400rpm$. In all of them, the continuum lines are higher than the dashed ones meaning that the right propeller was working at a higher speed rotation than the left one. This error has been corrected for future experiments.



a) $300rpm$

b) $350rpm$

c) $400rpm$

Figure 13: Longitudinal profiles after 20 hours running.

Figure 14 shows the evolution of the transversal profile $X1$ when the propellers are running at $400rpm$. As noticed, the effect of the twin propeller is only visible in the $10min$ line. After that, the influence of the twin propeller disappears completely. It is also only visible in the profiles closer to the propellers. Figure 15 shows the evolution of the transversal profile $X6$ at $400rpm$ confirming that there is no evidence of the twin propellers in any of the different times.

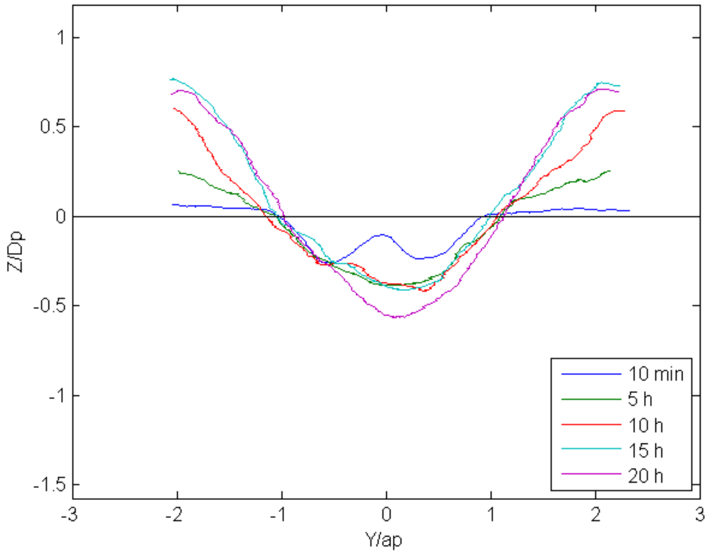


Figure 14: Evolution of the profile $X1$ at $400rpm$.

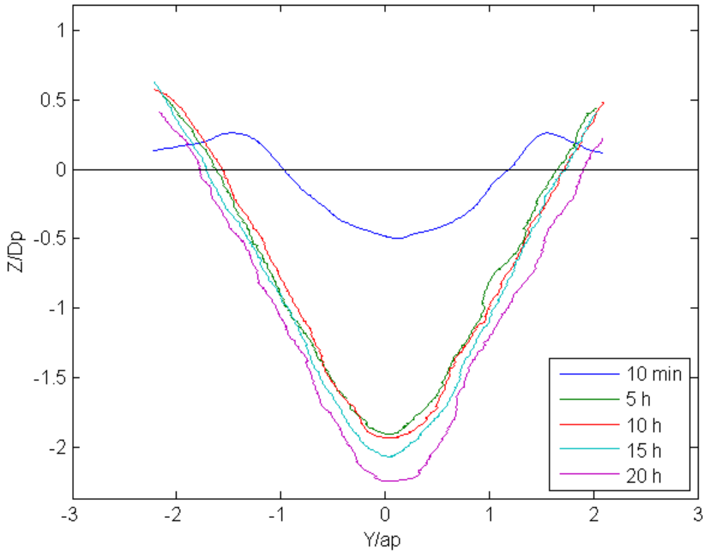


Figure 15: Evolution of the profile $X6$ at $400rpm$.

There are different problems, for harbour authorities, associated with the scouring holes. Not only the maximum scouring depth is important to prevent the structural problems, but also the deposition of the sediment in other areas affect the manoeuvring of the basin and its total use.

Figure 16 shows the evolution of the maximum depth of the scouring hole respect to the time. The maximum depth is achieved at the centreline of the scouring hole $Y0$ due to being symmetric as stated before.

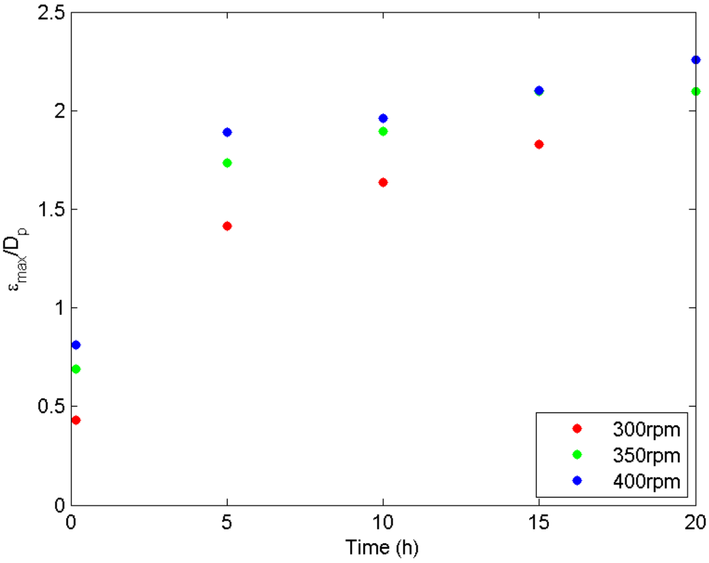


Figure 16: Evolution of the maximum scouring depth, ϵ_{max}.

Note that ϵ_{max}/D_p at $350rpm$ and $400rpm$ after 15 hours, as well as ϵ_{max}/D_p at $300rpm$ and $350rpm$ after 20 hours, are similar and not really well seen at Figure 16.

The behaviour of the three scenarios is consistent with qualitative previous experiments (Nardone et al. (2014); Hamill (1988); PIANC (2015)) and can be fitted within a log-log profile.

One step further is to create a dimensionless variable for the x axis. Having three scenarios working at different rpm, the x axis is multiplied by three different constants that correspond to the different velocities:

$$300\frac{rev}{min}\frac{2\pi rad}{1rev}\frac{60min}{1h} \tag{1}$$

$$350\frac{rev}{min}\frac{2\pi rad}{1rev}\frac{60min}{1h} \tag{2}$$

$$400\frac{rev}{min}\frac{2\pi rad}{1rev}\frac{60min}{1h} \tag{3}$$

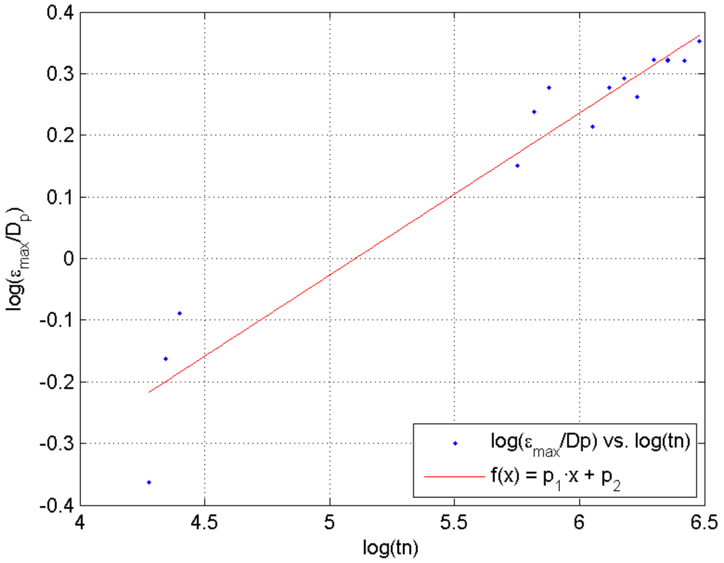


Figure 17: Fitting of the evolution of the maximum scouring depth.

The coefficient of determination R^2 for the fitting formula for Figure 17 is $R^2 = 0.9316$. The formula that relates the scouring depth with the diameter of the propeller is:

$$\frac{\epsilon_{max}}{D_p} = 0.2613tn^{0.2629} \tag{4}$$

Figure 18 shows the evolution of the maximum deposition. In this case, the maximum deposition was not on the centreline $Y0$ but on the further profiles ($Y5$ and Y_5) from the centreline. As explained before, the right propeller was working at a higher speed, so the profile that had the maximum deposition was $Y5$.

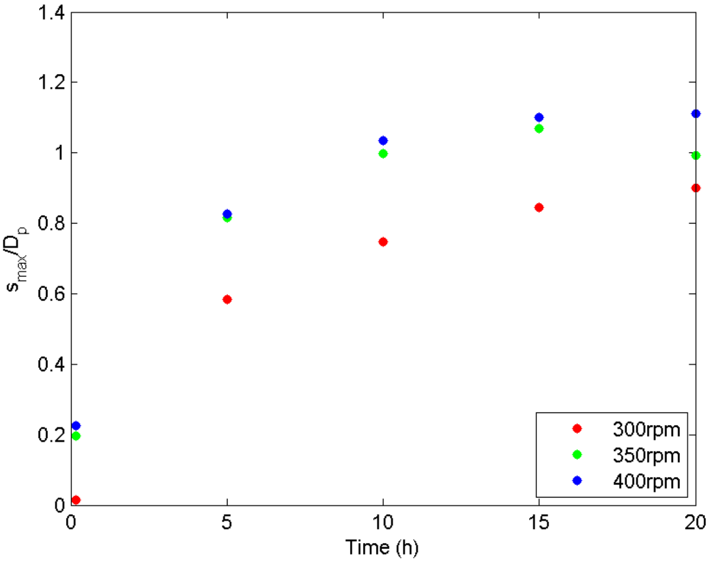


Figure 18: Evolution of the maximum deposition, s_{max}.

Also, notice that when the propellers were running at 300rpm and the data was read at 10min, the deposition was negligible (≈ 0). Using a log-log profile and fitting an equation to the data gave a coefficient of determination R^2 lower than expected: $R^2 = 0.7436$ (Figure 19)

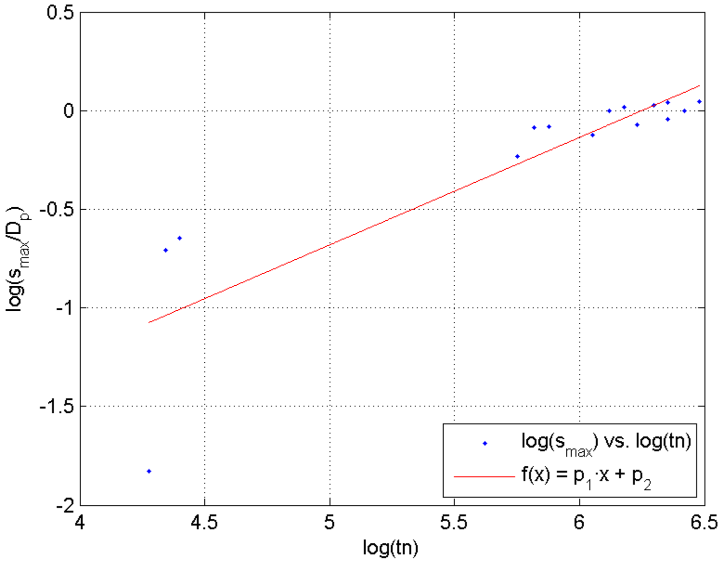


Figure 19: Fitting of the evolution of the maximum deposition.

If the value when the propellers were running at 300rpm at 10min was deleted from the series, the coefficient of determination R^2 gave the expected result(Figure 20).

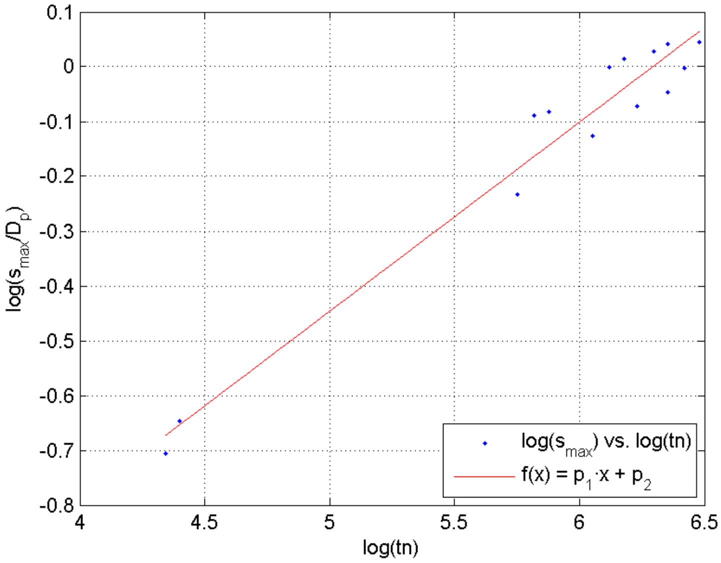


Figure 20: Fitting of the evolution of the maximum deposition (not using 300rpm 10min).

The coefficient of determination R^2 for the fitting formula for Figure 17 is $R^2 = 0.9584$. The formula that relates the scouring depth with the diameter of the propeller is:

$$\frac{s_{max}}{D_p} = 0.1139tn^{0.3452} \tag{5}$$

In Figure 21, the evolution of the maximum length of the scouring hole is plotted. As the scouring hole is symmetrical, the maximum length is always achieved in the centreline $Y0$.

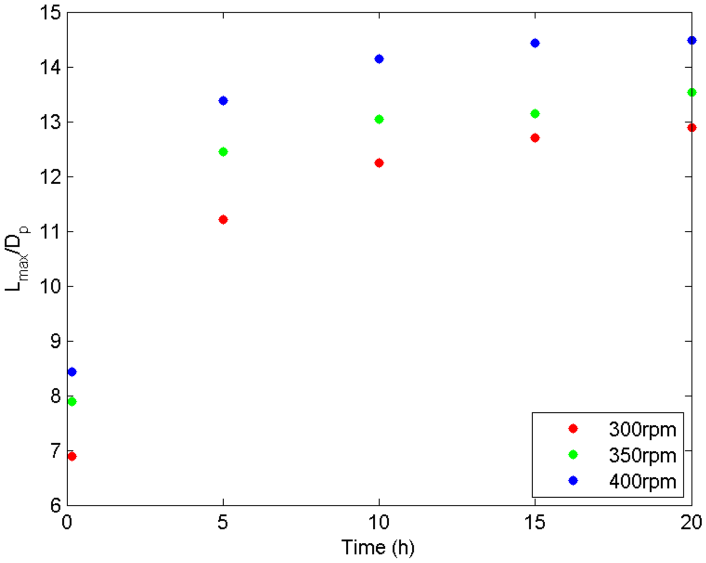


Figure 21: Evolution of the maximum scouring length, L_{max}.

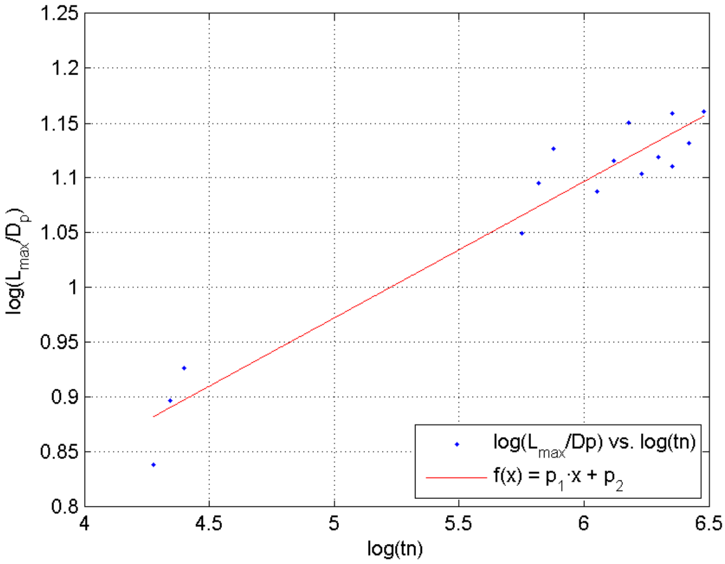


Figure 22: Fitting of the evolution of the maximum scouring length.

The coefficient of determination of the curve fitting (22)is $R^2 = 0.9366$ and the formula itself:

$$\frac{L_{max}}{D_p} = 1.4196tn^{0.1244} \tag{6}$$

Figure 23 shows the evolution of the maximum scouring width. This variable did not appear in the same profiles. It was dependant of the measuring time. For the

$10min$ scenario, the maximum scouring width was in the profile $X4$, whereas for all the other scenarios (5 hours to 20 hours), it was in the $X7$.

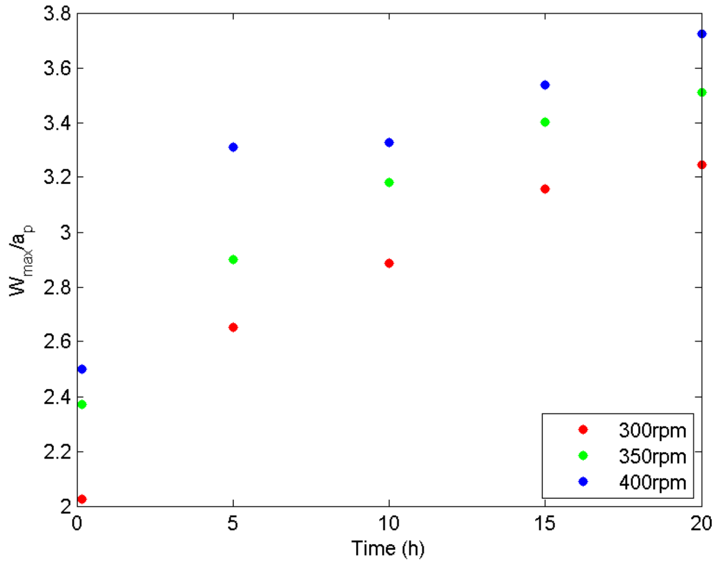


Figure 23: Evolution of the maximum scouring width.

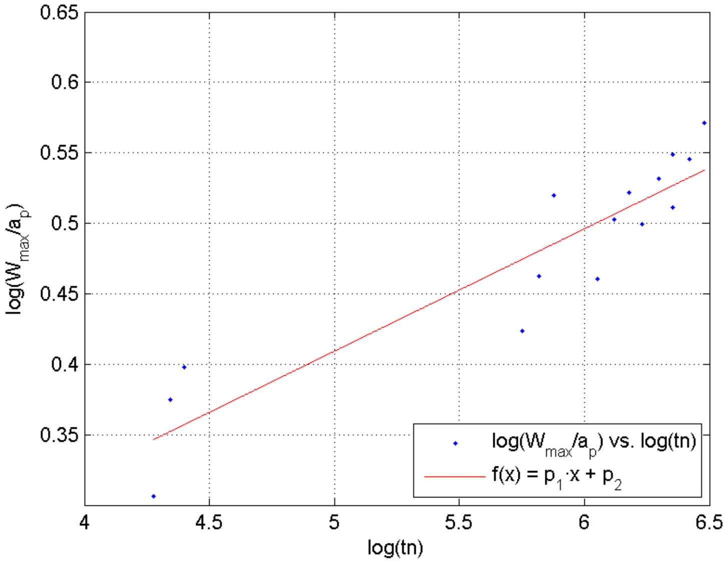


Figure 24: Fitting of the evolution of the maximum scouring width.

R^2 for Figure 24 is $R^2 = 0.8441$. The coefficient of determination for the curve fitting of the scouring width is lower than the ones from other scouring variables. At first it could be the same problem as in Figure 19, that one of the data points at $10min$ time was negligible and it affected the whole curve fitting. But in this case, that is not what happens. In a case like this, the data lecturing should be redone to have another set of points and try the self-made MATLAB script.

Nonetheless, the equation is shown, as R^2 is not that low, but not as high as the ones from other variables:

$$\frac{W_{max}}{D_p} = 0.9754tn^{0.0868} \tag{7}$$

As a general discussion, all variables are related logarithmically with velocity and time. These equations can be given to harbour authorities to prevent damages to the structures and to estimate protections. This variables are subjected to the material, and all the experiments should be redone with another D_{50}.

5 Conclusions

Physical experiments are a good way to approach the problem, but they require of big facilities and they take a long time to test them. Even the finest error can bargain away the whole experiment. The scale effect produced with the use of sediments limits the applicability of the results, than can be used as an upper bound for harbour authorities and port's projects engineers. To get rid of scale effects

The new experimental system composed by photoelectric sensors Efector200-O1D100 was revealed as a good and cheap choice to analyse scouring effects due to ship propellers.

Results obtained after modelling the propulsion system are found to be different than the previous data that used a single propeller (Hamill et al. (1999); Hong et al. (2012)). Former experiments were not realistic in terms of real ship motion, as it was done with a single propeller and the approach to relate twin propellers was to double the results of single propeller.

As the twin propellers are used as a main propulsion system, these experiments can be used to obtain the erosion caused by vessels that are manoeuvring in navigation channels or in areas that have no influence of a wall. As seen before, the sediment will accumulate in the laterals of the navigation channel, thus reducing the depth of that area.

Also, the effect of the twin propeller is only visible in the $10min$ time step. After that, the influence disappears completely. Moreover, this effect is only visible in the profiles closer to the propellers ($X1, ..., X3$).

This thesis has allowed to add new formulation that directly relates erosion variables (ϵ_{max}, s_{max}, L_{max}, W_{max}) with the revolution of the propulsion systems, without the necessity of approximating the efflux velocity.

Further investigations and analysis should change D_{50} of the sediment, because in this thesis it always been kept constant and the new formulation, that directly relates the erosion variables with the revolution of the propulsion systems, have not been related to such an important variable as D_{50}.

Acknowledgements

This research has been carried out as a Master's Thesis in Civil Engineering at ETSECCPB, Barcelona.

This thesis has been supported by MINECO (Ministerio de Economía y Competitividad) and FEDER (Unión Europea - Fondo Europeo de Desarrollo Regional "Una Manera de hacer Europa") from Spanish Government through projects BIA2012-38676-C03-01 and TRA2015-70473-R.

I would like to give special thanks to Prof. X. Gironella and Dr. A. Mujal-Colilles as they have guided me throughout all the process.

References

Blaauw, H. and Van de Kaa, E. (1978). *Erosion of bottom and sloping banks caused by the screw race of manoevring ships.* Delft Hydraulics.

Hamill, G., Johnston, H., and Stewart, D. (1999). Propeller wash scour near quay walls. *Journal of waterway, port, coastal, and ocean engineering*, 125(4):170–175.

Hamill, G. A. (1987). *Characteristics of the screw wash of a manoeuvring ship and the resulting bed scour.* PhD thesis, Queen's University of Belfast.

Hamill, G. A. (1988). The scouring action of the propeller jet produced by a slowly manoeuvring ship. *Bulletin of the Permanent International Association of Navigation Congresses [PIANC]*, (62).

Hong, J.-H., Chiew, Y.-M., Susanto, I., and Cheng, N.-S. (2012). Evolution of scour induced by propeller wash. *ICSE6*, page 147.

Mujal-Colilles, A., Gironella, X., Jaquet, A., Gomez-Gesteira, R., and Sanchez-Arcilla, A. (2015). Study of the efflux velocity induced by two propellers. *SCACR, Conference on Applied Coastal Research.*

Nardone, P., Geisenhainer, P., Koll, K., and Di Cristo, C. (2014). Experimental investigation of a propeller jet induced velocity field.

Nyquist, H. (1928). Certain topics in telegraph transmission theory. *Trans. AIEE*, 47:617–644.

PIANC (2015). Guidelines for protecting berthing structures from scour caused by ships. Technical Report 180.

Robakiewicz, M. (1987). Bottom erosion as an effect of ship propeller action near the harbour quays. *Bulletin of the Permanent International Association of Navigation Congresses [PIANC]*, (58).

Stewart, D. P. J. (1992). *Characteristics of a ship's screw wash and the influence of quay wall proximity.* PhD thesis, Queen's University of Belfast.

Verheij, H. (2010). Comparison of water jets and conventional propeller jets. In *Port Infrastructure Seminar 2010, Delft, The Netherlands, June 22-23, 2010.* TU Delft, Section Hydraulic Engineering.

Appendix A. Source code

Description

Some code in MATLAB language had to be written to get the results and the plots throughout the whole thesis.

The first two codes were used to calibrate the data received from the sensors in Volts and transform it to meters, and to produce all the matrices of the different profiles, as well as the graphics.

The other codes were used to relate erosion variables with the revolution if the propulsion systems.

Source code files

There are six source code files:

- plot_scouring_hole.m
- calibrationV_M.m
- depth.m
- deposition.m
- length_.m
- width_.m

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                            PLOT SCOURING HOLE                           %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
clear all; clc; close all

%% FIXED VARIABLES
Dp = 0.254;     %m
hp = 0.26;      %m
ap = 0.58;      %m
ch = 2.35;      %m distance to the centre of the helices
Hw = 0.70;      %m
Hs = 0.52;      %m

%% LOADING DATA
exp_name = input('Experiment Name: 300, 350, 400: ','s');
exp_time = input('Experiment Time: ','s');
```

```
calib = input('Calibrate from Volts to M? (y/n): ','s');
quality_size = input('Quality size: 32, 64: ','s');

if isempty(calib)
   calib = 'y';
end

exp_name = strcat(exp_name,'rpm');

rootDir = uigetdir('D:\TFM\');
openDir = strcat(rootDir,'\',exp_name,'\');
openDir = strcat(openDir,exp_time,'\');
openDir2 = strcat(openDir,quality_size,'\');

saveDir = strcat(rootDir, '\Results\plot_scouring_hole\',exp_name,'\');

mkdir(saveDir);
mkdir(strcat(saveDir, '\longitudinal\'));
mkdir(strcat(saveDir, '\transversal\'));
mkdir(strcat(saveDir, '\3D\'));
mkdir(strcat(saveDir, '\contour\'));

% File Names and order
long_names = {'Y_5';'Y_4';'Y_3';'Y_2';'H1';'Y_1';'Y0';'Y1';'H2';...
    'Y2';'Y3';'Y4';'Y5'};
trans_names = {'X0';'X1';'X2';'X3';'X4';'X5';'X6';'X7';'X8';'X9';...
    'X10';'X11'};

% Calibration and mobile mean
if strcmp(calib, 'y');
   dataDir = strcat(openDir, 'original_data\');
   load(strcat(dataDir, 'cal.txt'));
   files = dir(strcat(dataDir,'*.dat'));

   qsize = str2num(quality_size); %Number of points in mobile mean (trans)
   t_lag = qsize/2; %Number of points in mobile mean (long)

   % Load files
   for i =1:size(files, 1);
       V = load(strcat(dataDir,files(i,1).name));
       if files(i,1).name(1)=='X'
          V = calibrationV_M(V,cal,qsize);
       else
          V = calibrationV_M(V,cal,t_lag);
       end

       varname = genvarname(files(i,1).name(1:end-4));
       eval([varname '=V;'])

       save(strcat(openDir2, varname,'.mat'),files(i,1).name(1:end-4))
   end

else

    % Load files
    files = dir(strcat(openDir2,'*.mat'));
    for i =1:size(files, 1);
```

```
       load(strcat(openDir2,files(i,1).name(1:end-4),'.mat'));
    end
end

% Figure parameters
positionVect = [6    1    30   20];

%% TRANSVERSAL (X) & LONGITUDINAL (Y)

% Longitudinal
figure(1)
set(0,'DefaultAxesColorOrder',[1 0 0; 1 0 1;1 1 0;0 1 0;0 1 1;...
    0 0 1;0 0 0 ;0 0 1;0 1 1;0 1 0;1 1 0;1 0 1;1 0 0])

for i = 1:size(long_names,1)
    V = eval(long_names{i});
    if i>7
        plot(V(:,1)/Dp,(V(:,3)-Hs)/Dp,'LineStyle','--')
    else
        plot(V(:,1)/Dp,(V(:,3)-Hs)/Dp)
    end
    hold all
    clear V
end

plot([0,7]/Dp,[0,0],'k')

legend(long_names,'Location','EastOutside')
xlabel('X/Dp')
ylabel('Z/Dp')
xlim([0 6.5]/Dp)
ylim([-0.6 0.4]/Dp)

figName = strcat(saveDir,'\longitudinal\',exp_name,'_',exp_time,...
    '_',quality_size,'_long');
        set(gcf,'PaperPositionMode','Auto')
        print ('-dpng','-zbuffer','-r300',figName)
        saveas(gcf,figName, 'png');

% Transversal
figure(2)

set(0,'DefaultAxesColorOrder',[1 0 0; 1 0 1;0.5 0 0.5;0 0 0;0.5 0 0;...
    0.5 0.5 0;0.5 0.5 0.5;0.5 0.5 1;0 0 1;0 1 1;0 1 0;1 1 0;])

for i = 1:size(trans_names,1)
    V = eval(trans_names{i});
    plot((V(:,2)-ch)/ap,(V(:,3)-Hs)/Dp)
    hold all
    clear V
end

plot([-5,5],[0,0],'k')

legend(trans_names,'Location','EastOutside')
xlabel('Y/ap')
ylabel('Z/Dp')
```

```
xlim([-3 3])
ylim([-0.6 0.4]/Dp)

figName = strcat(saveDir,'\transversal\',exp_name,'_',exp_time,...
    '_',quality_size,'_trans');
        set(gcf,'PaperPositionMode','Auto')
        print ('-dpng','-zbuffer','-r300',figName)
        saveas(gcf,figName, 'png');

%% VECTOR CONSTRUCTOR

V = eval(long_names{1});

for i = 2:size(long_names,1);
    V = [V;eval(long_names{i})];
end

for i = 1:size(trans_names,1)
    V = [V;eval(trans_names{i})];
end

%% 3D PLOT

stepsX = 0:0.05:6;
stepsY = 0:0.05:4;
[XI,YI]= meshgrid(stepsX,stepsY);

DI = griddata(V(:,1),V(:,2),V(:,3)-Hs,XI, YI);

save(strcat(openDir2, 'DI.mat'),'DI')

% Set figure parameters
fig = figure(3);

% Figure background color
set(fig,'Color',[1 1 1]);

% Set figure size
set(fig,'units','centimeters');
set(fig,'position',positionVect);
set(fig, 'PaperUnits','centimeters');
set(fig, 'PaperPosition',positionVect);

h = surf(XI/Dp,(YI-ch)/ap,DI/Dp);
hold on
contour3(XI/Dp,(YI-ch)/ap,DI/Dp,[0,0],'k')
hold on

% Plot measuring lines
for i=1:length(trans_names)
    V = eval(trans_names{i});
    plot3(V(:,1)/Dp,(V(:,2)-ch)/ap,(V(:,3)-Hs)/Dp,'r');
    hold all
end
for i=1:length(long_names)
    V = eval(long_names{i});
    plot3(V(:,1)/Dp,(V(:,2)-ch)/ap,(V(:,3)-Hs)/Dp,'r');
    hold all
end

% Axis characteristics
xlim([0 6]/Dp)
ylim([-6 6])
axis equal
zlim([-0.6 0.4]/Dp)
caxis([-2 1])

% Colormap gray
colorbar

xlabel('X/Dp')
ylabel('Y/ap')
zlabel('Z/Dp')
figName = strcat(saveDir,'\3D\',exp_name,'_',exp_time,...
    '_',quality_size,'_3D');
        set(gcf,'PaperPositionMode','Auto')
        print ('-dpng','-zbuffer','-r300',figName)
        saveas(gcf,figName, 'png');

%% CONTOUR

% Setting figure parameters
fig = figure(4);

% Figure background color
set(fig,'Color',[1 1 1]);

% Set figure size
set(fig,'units','centimeters');
set(fig,'position',positionVect);
set(fig, 'PaperUnits','centimeters');
set(fig, 'PaperPosition',positionVect);

contour(XI/Dp,(YI-ch)/ap,DI/Dp,40)
hold on
contour(XI/Dp,(YI-ch)/ap,DI/Dp,[0,0],'k')

% Plot measuring lines
for i=1:length(trans_names)
    V = eval(trans_names{i});
    plot(V(:,1)/Dp,(V(:,2)-ch)/ap,'r');
    hold all
end
for i=1:length(long_names)
    V = eval(long_names{i});
    plot(V(:,1)/Dp,(V(:,2)-ch)/ap,'r');
    hold all
    end

% Axis characteristics
xlim([1.2 6]/Dp)
ylim([-3 3])
axis equal
zlim([-0.6 0.4]/Dp)
```

```
caxis([-2 1])

% Colormap gray
colorbar

xlabel('X/Dp')
ylabel('Y/ap')
zlabel('Z/Dp')
figName = strcat(saveDir,'\contour\',exp_name,'_',exp_time,...
    '_',quality_size,'_contour');
        set(gcf,'PaperPositionMode','Auto')
        print ('-dpng','-zbuffer','-r300',figName)
        saveas(gcf,figName, 'png');
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                              CALIBRATION                                %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% This function transforms .dat files in Volts to .txt files in Meters.
% V = data in volts
% C = matrix with the calibration values C(1,:): X calibration parameters
                                        %C(2,:): Y calibration parameters
                                        %C(3,:): Z calibration parameters
% M = data in meters M = C(:,1)+Volts*C(:,2);
% f = moving average filtering

function M = calibrationV_M(V,C,f)

    % Cut initial and final data
       V = V(2*f:end-2*f,:);

       M(:,1) = C(1,1)+C(1,2)*V(:,1);
       M(:,2) = C(2,1)+C(2,2)*V(:,2);
       M(:,3) = C(3,1)+C(3,2)*V(:,3);

    % Moving average

       M = tsmovavg(M,'s',f,1);
       M = M(f:end,:);  % Cut NaN variables
end
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                                 DEPTH                                   %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
clear all; clc; close all

%% FIXED VARIABLES
Dp = 0.254;     %m
hp = 0.26;      %m
ap = 0.58;      %m
ch = 2.35;      %m distance to the centre of the helices
Hw = 0.70;      %m
Hs = 0.52;      %m
% Number of points in mobile mean = 64
```

```matlab

time = [10/60; 5; 10; 15; 20];

%% LOADING DATA
rootDir = uigetdir('D:\TFM\');
saveDir = strcat(rootDir,'\Results\depth\');
mkdir(saveDir);

k = 0; %Counter
for exp_name = [300 350 400]
    i = num2str(exp_name);
    kk = 0; %Counter
    for exp_time = [10 5 10 15 20]
        j = num2str(exp_time);
        k = k+1;
        kk = kk+1;
        if kk == 1;
            openDir = strcat(rootDir,'\',i,'rpm\',j,'min\64\Y0.mat');
        else
            openDir = strcat(rootDir,'\',i,'rpm\',j,'H\64\Y0.mat');
        end

        % Load files
        load(openDir);
        eval(['V' num2str(k) '=' 'Y0']);
        clear Y0
    end
end
clc;

% Figure parameters
positionVect = [6    1    30   20];

names300 = {'V1';'V2';'V3';'V4';'V5'};
names350 = {'V6';'V7';'V8';'V9';'V10'};
names400 = {'V11';'V12';'V13';'V14';'V15'};

for i = 1:size(names300,1)
V = eval(names300{i});
depth300(i,:) = (-min(V(:,3))+Hs)/Dp;
end

for i = 1:size(names350,1)
V = eval(names350{i});
depth350(i,:) = (-min(V(:,3))+Hs)/Dp;
end

for i = 1:size(names400,1)
V = eval(names400{i});
depth400(i,:) = (-min(V(:,3))+Hs)/Dp;
end

%% NORMAL PLOT

figure (1)
plot(time, depth300,'.','MarkerSize',15,'Color','r')
hold on
```

```
plot(time, depth350,'.','MarkerSize',15,'Color','g')
hold on
plot(time, depth400,'.','MarkerSize',15,'Color','b')

legend('300rpm','350rpm','400rpm','Location','SouthEast')

% Axis characteristics
xlabel('Time (h)')
ylabel('\epsilon_{max}/D_{p}')
xlim([0.0 20.0])
ylim([0.0 2.5])

figName = strcat(saveDir,'depth_normal');
        set(gcf,'PaperPositionMode','Auto')
        print ('-dpng','-zbuffer','-r300',figName)
        saveas(gcf,figName, 'png');

%% LOG-LOG PLOT

figure(2)
loglog(time, depth300,'.','MarkerSize',15,'Color','r')
hold on
loglog(time, depth350,'.','MarkerSize',15,'Color','g')
hold on
loglog(time, depth400,'.','MarkerSize',15,'Color','b')

legend('300rpm','350rpm','400rpm','Location','SouthEast')

% Axis characteristics
xlabel('log(Time) (h)')
ylabel('log(\epsilon_{max}/D_{p})')
xlim([0.0 100.0])
ylim([0.1 10.0])

figName = strcat(saveDir,'depth_loglog');
        set(gcf,'PaperPositionMode','Auto')
        print ('-dpng','-zbuffer','-r300',figName)
        saveas(gcf,figName, 'png');

%% DIMENSIONLESS LOG-LOG PLOT

% New vectors of time
t300 = time*300*2*pi()*60;
t350 = time*350*2*pi()*60;
t400 = time*400*2*pi()*60;

% Creation of dimensionless vectors
j = 1; % Counter
k = 1; % Counter
for i = 1:15
    if i < 6
        log_depth(i,:) = log10(depth300(i,:));
        log_time_dimensionless (i,:) = log10(t300(i,:));
    elseif i > 5 && i < 11
        log_depth(i,:) = log10(depth350(j,:));
        log_time_dimensionless (i,:) = log10(t350(j,:));
        j = j+1;
    else
        log_depth(i,:) = log10(depth400(k,:));
        log_time_dimensionless (i,:) = log10(t400(k,:));
        k = k+1;
    end
end

% Fit: 'f(x) = p1 x + p2'.
[xData, yData] = prepareCurveData( log_time_dimensionless, log_depth );

% Set up fittype and options.
ft = fittype( 'poly1' );
opts = fitoptions( ft );
opts.Lower = [-Inf -Inf];
opts.Upper = [Inf Inf];

% Fit model to data.
[fitresult, gof] = fit( xData, yData, ft, opts );

% Plot fit with data.
figure( 'Name', 'f(x) = p_{1} x + p_{2}' );
h = plot( fitresult, xData, yData);
legend( h, 'log(\epsilon_{max}/Dp) vs. log(tn)',...
    'f(x) = p_{1} x + p_{2}', 'Location', 'SouthEast' );
% Label axes
xlabel( 'log(tn)' );
ylabel( 'log(\epsilon_{max}/D_{p})' );
grid on

fitresult
gof

figName = strcat(saveDir,'depth_loglog_adim');
        set(gcf,'PaperPositionMode','Auto')
        print ('-dpng','-zbuffer','-r300',figName)
        saveas(gcf,figName, 'png');
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                              DEPOSITION                                 %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
clear all; clc; close all

%% FIXED VARIABLES
Dp = 0.254;     %m
hp = 0.26;      %m
ap = 0.58;      %m
ch = 2.35;      %m distance to the centre of the helices
Hw = 0.70;      %m
Hs = 0.52;      %m
% Number of points in mobile mean = 64

time = [10/60; 5; 10; 15; 20];

%% LOADING DATA
rootDir = uigetdir('D:\TFM\');
saveDir = strcat(rootDir,'\Results\deposition\');
```

```
mkdir(saveDir);

k = 0; %Counter
for exp_name = [300 350 400]
    i = num2str(exp_name);
    kk = 0; %Counter
    for exp_time = [10 5 10 15 20]
        j = num2str(exp_time);
        k = k+1;
        kk = kk+1;
        if kk == 1;
            openDir = strcat(rootDir,'\',i,'rpm\',j,'min\64\Y5.mat');
        else
            openDir = strcat(rootDir,'\',i,'rpm\',j,'H\64\Y5.mat');
        end

        % Load files
        load(openDir);
        eval(['V' num2str(k) '=' 'Y5']);
        clear Y5
    end
end
clc;

% Figure parameters
positionVect = [6    1    30   20];

names300 = {'V1';'V2';'V3';'V4';'V5'};
names350 = {'V6';'V7';'V8';'V9';'V10'};
names400 = {'V11';'V12';'V13';'V14';'V15'};

for i = 1:size(names300,1)
    V = eval(names300{i});
    deposition300(i,:) = (max(V(:,3))-Hs)/Dp;
end

for i = 1:size(names350,1)
    V = eval(names350{i});
    deposition350(i,:) = (max(V(:,3))-Hs)/Dp;
end

for i = 1:size(names400,1)
    V = eval(names400{i});
    deposition400(i,:) = (max(V(:,3))-Hs)/Dp;
end

%% NORMAL PLOT

figure (1)
plot(time, deposition300,'.','MarkerSize',15,'Color','r')
hold on
plot(time, deposition350,'.','MarkerSize',15,'Color','g')
hold on
plot(time, deposition400,'.','MarkerSize',15,'Color','b')

legend('300rpm','350rpm','400rpm','Location','SouthEast')

```

```matlab
% Axis characteristics
xlabel('Time (h)')
ylabel('s_{max}/D_{p}')
% xlim([0.0 20.0])
% ylim([0.0 1.5])

figName = strcat(saveDir,'deposition_normal');
        set(gcf,'PaperPositionMode','Auto')
        print ('-dpng','-zbuffer','-r300',figName)
        saveas(gcf,figName, 'png');

%% LOG-LOG PLOT

figure(2)
loglog(time, deposition300,'.','MarkerSize',15,'Color','r')
hold on
loglog(time, deposition350,'.','MarkerSize',15,'Color','g')
hold on
loglog(time, deposition400,'.','MarkerSize',15,'Color','b')

legend('300rpm','350rpm','400rpm','Location','SouthEast')

% Axis characteristics
xlabel('log(Time) (h)')
ylabel('log(s_{max}/D_{p})')
xlim([0.0 100.0])
ylim([0.0 5.0])

figName = strcat(saveDir,'deposition_loglog');
        set(gcf,'PaperPositionMode','Auto')
        print ('-dpng','-zbuffer','-r300',figName)
        saveas(gcf,figName, 'png');

%% DIMENSIONLESS LOG-LOG PLOT

% New vectors of time
t300 = time*300*2*pi()*60;
t350 = time*350*2*pi()*60;
t400 = time*400*2*pi()*60;

% Creation of dimensionless vectors
j = 1; % Counter
k = 1; % Counter
for i = 1:15
    if i < 6
        log_deposition(i,:) = log10(deposition300(i,:));
        log_time_dimensionless (i,:) = log10(t300(i,:));
    elseif i > 5 && i < 11
        log_deposition(i,:) = log10(deposition350(j,:));
        log_time_dimensionless (i,:) = log10(t350(j,:));
        j = j+1;
    else
        log_deposition(i,:) = log10(deposition400(k,:));
        log_time_dimensionless (i,:) = log10(t400(k,:));
        k = k+1;
    end
end
```

```

% Vectors without the first component (300rpm 10min value)
% Use (uncomment) for better fitting
% log_time_dimensionless(1,:) = [];
% log_deposition(1,:) = [];


% Fit: 'f(x) = p1 x + p2'.
[xData, yData] = prepareCurveData( log_time_dimensionless, log_deposition );

% Set up fittype and options.
ft = fittype( 'poly1' );
opts = fitoptions( ft );
opts.Lower = [-Inf -Inf];
opts.Upper = [Inf Inf];

% Fit model to data.
[fitresult, gof] = fit( xData, yData, ft, opts );

% Plot fit with data.
figure( 'Name', 'f(x) = p_{1} x + p_{2}' );
h = plot( fitresult, xData, yData );
legend( h, 'log(s_{max}) vs. log(tn)', 'f(x) = p_{1} x + p_{2}',...
    'Location', 'SouthEast' );

% Label axes
xlabel( 'log(tn)' );
ylabel( 'log(s_{max}/D_{p})' );
grid on

fitresult
gof

figName = strcat(saveDir,'deposition_loglog_adim');
        set(gcf,'PaperPositionMode','Auto')
        print ('-dpng','-zbuffer','-r300',figName)
        saveas(gcf,figName, 'png');
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                                 LENGTH                                   %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
clear all; clc; close all

%% FIXED VARIABLES
Dp = 0.254;     %m
hp = 0.26;      %m
ap = 0.58;      %m
ch = 2.35;      %m distance to the centre of the helices
Hw = 0.70;      %m
Hs = 0.52;      %m
% Number of points in mobile mean = 64

time = [10/60; 5; 10; 15; 20];

%% LOADING DATA
rootDir = uigetdir('D:\TFM\');
```

```
saveDir = strcat(rootDir,'\Results\length\');
mkdir(saveDir);

k = 0; %Counter
for exp_name = [300 350 400];
    i = num2str(exp_name);
    kk = 0; %Counter
    for exp_time = [10 5 10 15 20];
        j = num2str(exp_time);
        k = k+1;
        kk = kk+1;
        if kk == 1;
            openDir = strcat(rootDir,'\',i,'rpm\',j,'min\64\Y0.mat');
        else
            openDir = strcat(rootDir,'\',i,'rpm\',j,'H\64\Y0.mat');
        end

        % Load files
        load(openDir);
        eval(['V' num2str(k) '=' 'Y0']);
        clear Y0
    end
end
clc;

% Figure parameters
positionVect = [6    1    30   20];

names300 = {'V1';'V2';'V3';'V4';'V5'};
names350 = {'V6';'V7';'V8';'V9';'V10'};
names400 = {'V11';'V12';'V13';'V14';'V15'};

start = 1.17; % The start of the scouring hole is under the propellers

for i = 1:size(names300,1)
    V = eval(names300{i});
    for m = 1:(size(V)-1);
        if (V(m,3)-Hs)<0 && (V(m+1,3)-Hs)>0;
            finish = (V(m,1)+V(m+1,1))/2;
            length300(i,:) = (finish - start)/Dp;
        else
        end
    end
end

for i = 1:size(names350,1)
    V = eval(names350{i});
    for m = 1:(size(V)-1);
        if (V(m,3)-Hs)<0 && (V(m+1,3)-Hs)>0;
            finish = (V(m,1)+V(m+1,1))/2;
            length350(i,:) = (finish - start)/Dp;
        else
        end
    end
end

for i = 1:size(names400,1)
```

```
    V = eval(names400{i});
    for m = 1:(size(V)-1);
        if (V(m,3)-Hs)<0 && (V(m+1,3)-Hs)>0;
            finish = (V(m,1)+V(m+1,1))/2;
            length400(i,:) = (finish - start)/Dp;
        else
        end
    end
end

%% NORMAL PLOT

figure(1)

plot(time, length300,'.','MarkerSize',15,'Color','r')
hold on
plot(time, length350,'.','MarkerSize',15,'Color','g')
hold on
plot(time, length400,'.','MarkerSize',15,'Color','b')

legend('300rpm','350rpm','400rpm','Location','SouthEast')

% Axis characteristics
xlabel('Time (h)')
ylabel('L_{max}/D_{p}')

figName = strcat(saveDir,'length_normal');
        set(gcf,'PaperPositionMode','Auto')
        print ('-dpng','-zbuffer','-r300',figName)
        saveas(gcf,figName, 'png');

%% LOG-LOG PLOT

figure(2)

loglog(time, length300,'.','MarkerSize',15,'Color','r')
hold on
loglog(time, length350,'.','MarkerSize',15,'Color','g')
hold on
loglog(time, length400,'.','MarkerSize',15,'Color','b')

legend(exp_name,'Location','EastOutside')

% Axis characteristics
xlabel('log(Time) (h)')
ylabel('log(L_{max}/D_{p})')

figName = strcat(saveDir,'length_loglog');
        set(gcf,'PaperPositionMode','Auto')
        print ('-dpng','-zbuffer','-r300',figName)
        saveas(gcf,figName, 'png');

%% DIMENSIONLESS LOG-LOG PLOT

% New vectors of time
t300 = time*300*2*pi()*60;
t350 = time*350*2*pi()*60;
```

```
t400 = time*400*2*pi()*60;

% Creation of dimensionless vectors
j = 1; % Counter
k = 1; % Counter
for i = 1:15
    if i < 6
        log_length(i,:) = log10(length300(i,:));
        log_time_dimensionless (i,:) = log10(t300(i,:));
    elseif i > 5 && i < 11
        log_length(i,:) = log10(length350(j,:));
        log_time_dimensionless (i,:) = log10(t350(j,:));
        j = j+1;
    else
        log_length(i,:) = log10(length400(k,:));
        log_time_dimensionless (i,:) = log10(t400(k,:));
        k = k+1;
    end
end

% Fit: 'f(x) = p1*x + p2'.
[xData, yData] = prepareCurveData( log_time_dimensionless, log_length );

% Set up fittype and options.
ft = fittype( 'poly1' );
opts = fitoptions( ft );
opts.Lower = [-Inf -Inf];
opts.Upper = [Inf Inf];

% Fit model to data.
[fitresult, gof] = fit( xData, yData, ft, opts );

% Plot fit with data.
figure( 'Name', 'f(x) = p_{1} x + p_{2}' );
h = plot( fitresult, xData, yData);
legend( h, 'log(L_{max}/Dp) vs. log(tn)',...
    'f(x) = p_{1} x + p_{2}', 'Location', 'SouthEast' );
% Label axes
xlabel( 'log(tn)' );
ylabel( 'log(L_{max}/D_{p})' );
grid on

fitresult
gof

figName = strcat(saveDir,'length_loglog_adim');
        set(gcf,'PaperPositionMode','Auto')
        print ('-dpng','-zbuffer','-r300',figName)
        saveas(gcf,figName, 'png');
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                                  WIDTH                                  %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
clear all; clc; close all

%% FIXED VARIABLES
```

```
Dp = 0.254;       %m
hp = 0.26;        %m
ap = 0.58;        %m
ch = 2.35;        %m distance to the centre of the helices
Hw = 0.70;        %m
Hs = 0.52;        %m
% Number of points in mobile mean = 64

time = [10/60; 5; 10; 15; 20];

%% LOADING DATA
rootDir = uigetdir('D:\TFM\');
saveDir = strcat(rootDir,'\Results\width\');
mkdir(saveDir);

k = 0; %Counter
for exp_name = [300 350 400];
    i = num2str(exp_name);
    kk = 0; %Counter
    for exp_time = [10 5 10 15 20];
        j = num2str(exp_time);
        k = k+1;
        kk = kk+1;
        finish = 0;
        if kk == 1;
            openDir = strcat(rootDir,'\',i,'rpm\',j,'min\64\X4.mat');
            load(openDir);
            eval(['V' num2str(k) '=' 'X4']);
            clear X4
        else
            openDir = strcat(rootDir,'\',i,'rpm\',j,'H\64\X7.mat');
            load(openDir);
            eval(['V' num2str(k) '=' 'X7']);
            clear X7
        end
    end
end
clc;

% Figure parameters
positionVect = [6    1    30   20];

names300 = {'V1';'V2';'V3';'V4';'V5'};
names350 = {'V6';'V7';'V8';'V9';'V10'};
names400 = {'V11';'V12';'V13';'V14';'V15'};

for i = 1:size(names300,1)
    finish = 0;
    start = 0;
    V = eval(names300{i});
    for m = 1:(size(V)-1);
        if finish == 0;
            if (V(m,3)-Hs)>0 && (V(m+1,3)-Hs)<0;
                start = ((V(m,2)-ch)+(V(m+1,2)-ch))/2;
            elseif (V(m,3)-Hs)<0 && (V(m+1,3)-Hs)>0;
                finish = ((V(m,2)-ch)+(V(m+1,2)-ch))/2;
            else
            end
        else
        end
    end
    width300(i,:) = (abs(finish - start))/ap;
end

for i = 1:size(names350,1)
    finish = 0;
    start = 0;
    V = eval(names350{i});
    for m = 1:(size(V)-1);
        if finish == 0;
            if (V(m,3)-Hs)>0 && (V(m+1,3)-Hs)<0;
                start = ((V(m,2)-ch)+(V(m+1,2)-ch))/2;
            elseif (V(m,3)-Hs)<0 && (V(m+1,3)-Hs)>0;
                finish = ((V(m,2)-ch)+(V(m+1,2)-ch))/2;
            else
            end
        else
        end
    end
    width350(i,:) = (abs(finish - start))/ap;
end

for i = 1:size(names400,1)
    finish = 0;
    start = 0;
    V = eval(names400{i});
    for m = 1:(size(V)-1);
        if finish == 0;
            if (V(m,3)-Hs)>0 && (V(m+1,3)-Hs)<0;
                start = ((V(m,2)-ch)+(V(m+1,2)-ch))/2;
            elseif (V(m,3)-Hs)<0 && (V(m+1,3)-Hs)>0;
                finish = ((V(m,2)-ch)+(V(m+1,2)-ch))/2;
            else
            end
        else
        end
    end
    width400(i,:) = (abs(finish - start))/ap;
end

%% NORMAL PLOT

figure(1)

plot(time, width300,'.','MarkerSize',15,'Color','r')
hold on
plot(time, width350,'.','MarkerSize',15,'Color','g')
hold on
plot(time, width400,'.','MarkerSize',15,'Color','b')

legend('300rpm','350rpm','400rpm','Location','SouthEast')

% Axis characteristics
xlabel('Time (h)')
```

```
ylabel('W_{max}/a_{p}')

figName = strcat(saveDir,'width_normal');
        set(gcf,'PaperPositionMode','Auto')
        print ('-dpng','-zbuffer','-r300',figName)
        saveas(gcf,figName, 'png');

%% LOG-LOG PLOT

figure(2)

loglog(time, width300,'.','MarkerSize',15,'Color','r')
hold on
loglog(time, width350,'.','MarkerSize',15,'Color','g')
hold on
loglog(time, width400,'.','MarkerSize',15,'Color','b')

legend(exp_name,'Location','EastOutside')

% Axis characteristics
xlabel('log(Time) (h)')
ylabel('log(W_{max}/a_{p})')

figName = strcat(saveDir,'width_loglog');
        set(gcf,'PaperPositionMode','Auto')
        print ('-dpng','-zbuffer','-r300',figName)
        saveas(gcf,figName, 'png');

%% DIMENSIONLESS LOG-LOG PLOT

% New vectors of time
t300 = time*300*2*pi()*60;
t350 = time*350*2*pi()*60;
t400 = time*400*2*pi()*60;

% Creation of dimensionless vectors
j = 1; % Counter
k = 1; % Counter
for i = 1:15
    if i < 6
        log_width(i,:) = log10(width300(i,:));
        log_time_dimensionless (i,:) = log10(t300(i,:));
    elseif i > 5 && i < 11
        log_width(i,:) = log10(width350(j,:));
        log_time_dimensionless (i,:) = log10(t350(j,:));
        j = j+1;
    else
        log_width(i,:) = log10(width400(k,:));
        log_time_dimensionless (i,:) = log10(t400(k,:));
        k = k+1;
    end
end

% Fit: 'f(x) = p1 x + p2'.
[xData, yData] = prepareCurveData( log_time_dimensionless, log_width );

% Set up fittype and options.
```

```
ft = fittype( 'poly1' );
opts = fitoptions( ft );
opts.Lower = [−Inf −Inf];
opts.Upper = [Inf Inf];

% Fit model to data.
[fitresult, gof] = fit( xData, yData, ft, opts );

% Plot fit with data.
figure( 'Name', 'f(x) = p_{1} x + p_{2}' );
h = plot( fitresult, xData, yData);
legend( h, 'log(W_{max}/a_{p}) vs. log(tn)',...
    'f(x) = p_{1} x + p_{2}', 'Location', 'SouthEast' );
% Label axes
xlabel( 'log(tn)' );
ylabel( 'log(W_{max}/a_{p})' );
grid on

fitresult
gof

figName = strcat(saveDir,'width_loglog_adim');
        set(gcf,'PaperPositionMode','Auto')
        print ('−dpng','−zbuffer','−r300',figName)
        saveas(gcf,figName, 'png');
```