Transition Orbits of Walking Droplets

A Senior Project presented to the Faculty of the Department of Physics

California Polytechnic State University, San Luis Obispo

In Partial Fulfillment of the Requirements for the Degree

Bachelor of Science, Physics

By:

Joshua Parker

Supervised By:

Dr. Nilgun Sungar

© 2015 J. Parker

Abstract: It was recently discovered that millimeter-sized droplets bouncing on the surface of an oscillating bath of the same fluid can couple with the surface waves it produces and begin walking across the fluid bath. These walkers have been shown to behave similarly to quantum particles; a few examples include single-particle diffraction, tunneling, and quantized orbits. Such behavior occurs because the drop and surface waves depend on each other to exist, making this the first and only known macroscopic pilot-wave system. In this paper, the quantized orbits between two identical drops are explored. By sending a perturbation to a pair of orbiting walkers, the orbit can be disrupted and transition to a new orbit. The numerical results of such transitions are analyzed and discussed.

Introduction

In 2005, a group of researchers from the University of Paris, led by Yves Couder, found that drops can bounce indefinitely on the surface waves of a fluid, which are created by the droplets themselves (Protiere 2006). For this phenomenon to occur it is essential that the fluid bath be driven up and down periodically with an acceleration greater than the gravitational acceleration. Additionally, Couder's group found that these droplets can "walk" along the surface of the fluid, whereby the localized surface waves are guiding or "piloting" the droplet across the bath. This marked the discovery of the first macroscopic pilot-wave system. As pilot wave theory would predict, these walking droplets mimic behavior observed in quantum systems. For example, the droplets have quantized orbits, and exhibit behavior similar to tunneling and single-particle diffraction, along with other quantum-like behaviors (see Eddi 2009 and Couder 2006).

This new field of hydrodynamic pilot-wave theory has been developed further in recent years by other researchers, most notably by John Bush, a professor of Applied Mathematics at MIT. Bush and his colleagues have performed several experiments to both confirm and add to the findings of Couder. In 2013 Molacek and Bush published a paper entitled Droplets Walking on a Vibrating Fluid Bath in which they developed a rigorous mathematical model for walkers. This paper will henceforth be referred to as MB2. Also, Bush recently published a review paper (see Bush 2015) on the field of hydrodynamic pilot-wave theory, which is a great introductory paper for those new to the subject.

Before the model for walkers in MB2 can be discussed, we must understand how a drop becomes a bouncer and how a bouncer becomes a walker. When a drop of fluid, in this case silicone oil, collides with a bath of its own fluid, there is a thin layer of air between the drop and the surface of the fluid. The drop will coalesce into the fluid once this intervening air layer dissipates to a critical thickness where Van der Waal forces cause the drop and bath to merge (Bush 2010). However, when the bath is subject to periodic oscillations with accelerations greater than the force of gravity, the drop may bounce off the surface of the bath before the intervening layer has time to break.

When the acceleration of the forced oscillations of the bath is increased and reaches a threshold value, the surface of the bath becomes unstable and standing waves emerge. These standing waves are known as Faraday waves and have a frequency which is half of the driving frequency. It is just below this instability that a bouncing droplet can move in the horizontal direction and become a walker. In this regime, the walkers bounce off the surface only once in two driving periods and can be treated as local emitters of weakly damped Faraday waves (Bush 2010). Therefore, when the drop comes in contact with the bath surface, it will land on the crest of the wave which was created from previous impacts of the drop. If the drop is perturbed slightly in any direction, then it will land on the side of this crest which will give the drop a horizontal kick in that direction, allowing it to walk along the bath surface.

We will now adopt the notation of MB2 in which each drop is prescribed a bouncing mode, (m,n), where the drop's vertical motion has a period of m driving periods, during which the drop contacts the bath n times. Also, each bouncing mode has two energy modes: a lower-energy mode, denoted as $(m,n)^1$, and a higher-energy mode, denoted as $(m,n)^2$. The lower-energy mode corresponds to a longer contact time between the drop and the bath and is typically restricted to bigger-sized drops. Alternatively, the higher-energy mode corresponds to a shorter contact time with the bath and is usually restricted to smaller-sized drops. Since walkers are period doubled and usually on the smaller side, we can be sure that most walkers are in the $(2,1)^2$ bouncing mode.

As previously mentioned, these walkers exhibit behavior which had previously only been observed in microscopic systems and described by quantum mechanics. One example is the quantization of orbits between two walkers, which will be the focus of this paper. Experiments were performed in which orbiting pairs were given a perturbation and transitions to different orbiting distances were observed. By measuring these transitions and other characteristics of the orbitals, the mathematical models of Couder and Bush could be analyzed.

Experimental Setup

In all the experiments carried out for this paper, the fluid bath used was silicone oil with a kinematic viscosity of 20 cS, which is 20 times that of water. The oil was contained in a 7 x 7 inch square aluminum tray and filled to a height of about 4 mm (the depth of the fluid was measured using a micrometer system). The tray was mounted on an electromagnetic shaker (Bruel and Kjaer 4809) which was securely attached to an optical table and driven by an amplifier (Bruel and Kjaer 2719). A signal generator (Agilent 33120A) was connected to this amplifier and used to vertically oscillate the tray and bath sinusoidally with an amplitude, A, and frequency, $f = 80$ Hz, giving the tray an acceleration, $\gamma = (2\pi f)^2 A$. To measure this acceleration, an accelerometer was attached to the side of the tray and connected to an Arduino Uno microcontroller board, which converted the analog output of the accelerometer to a 10 bit representation (a number between 0 and 1023) (Slaughter 2013). This data was read into Matlab and converted into acceleration units using the Matlab program provided in Appendix A.

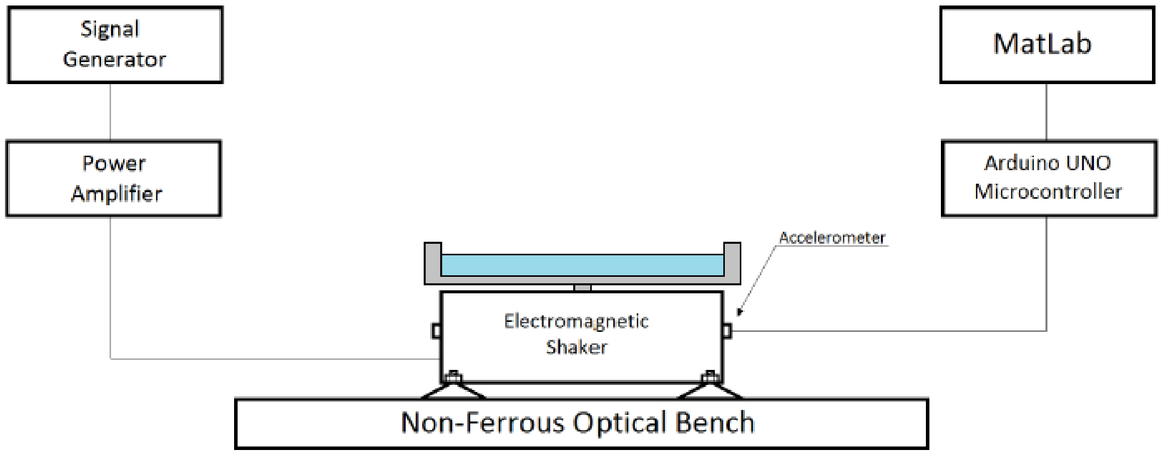

Figure 1: Diagram of experimental apparatus used to obtain data.

To observe and make measurements of the orbiting walkers, it was necessary to obtain videos that looked down on the tray. The best way to attain non-distorted images was by shining an LED lamp on the system from above, after passing through a diffuser placed below it. Then a semi-transparent mirror was placed between the diffuser and the tray at a 45 degree angle. This created a reflection of the fluid surface, which could be filmed from the side. These videos were taken at regular speed (30 frames/s). Most of the analysis for these videos were performed using Matlab's imaging toolbox. Also, in order to track the particles, a particle-tracking program was written in Matlab (see appendix B for code).

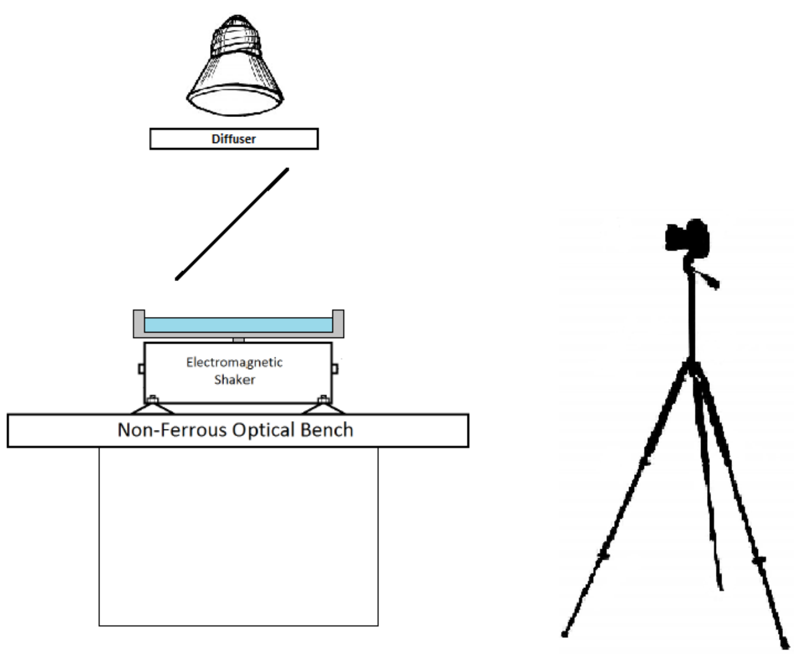

Figure 2: Experimental apparatus with imaging components.

The last component of this experimental setup was the device used to create a perturbation in the bath fluid to study transitions between different orbital states. This was accomplished with the use of a speaker. Attached to the speaker was a long protruding stick with a metal slab attached to the end of it. This slab was dipped partially into the fluid and a square wave pulse was sent to the speaker with a signal generator (Agilent 33220A) and an amplifier. The pulse caused the metal slab, to jump forward suddenly, creating a plane wave that propagated across the fluid bath. However, the plane wave would damp out rather quickly so the orbiting pair had to be kept close to the metal slab for the plane wave to have any effect on the system.

Soon after the experimental apparatus was set up, it was discovered that the Faraday threshold changes with time, due to heating of the system which can change the properties of the silicone oil. Figure 4 shows the threshold drop from 4.3g to 3.9g in the first 15 minutes or so, and then decrease in a more gradual manner. This issue can be mitigated by cooling the system, which can be achieved by running an air current through the shaker. For experiments where detailed calculations will be made and compared to theoretical values which depend on the properties of the fluid, the temperature of the system should be monitored and taken into account in analysis of data. For our experiments on orbiting drops this was not essential, but it was important to keep the acceleration at a level which was not susceptible to seeding Faraday waves in the middle of the experiment, due to heating of the system.

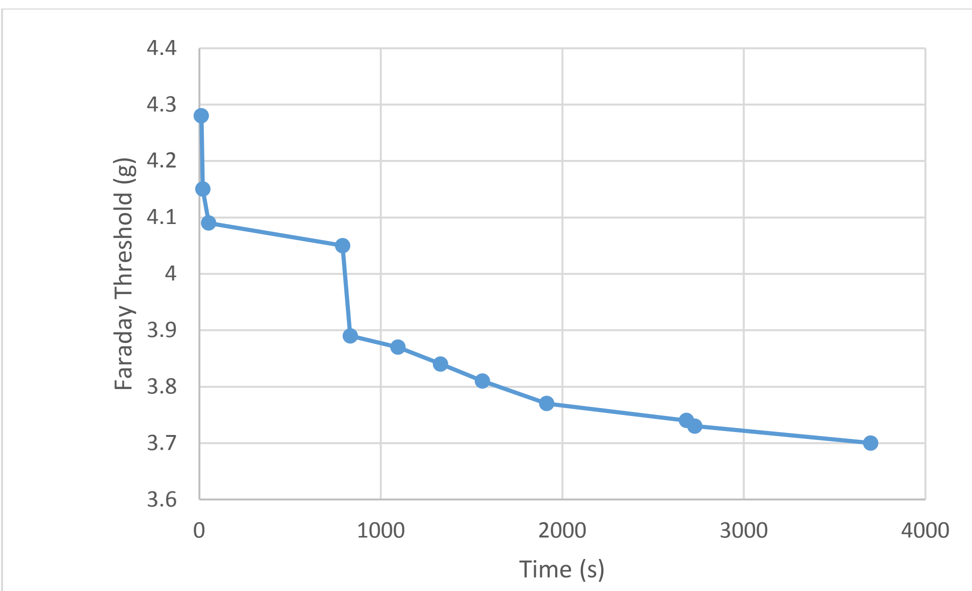

Figure 3: Faraday threshold dependence on time, due to heating of the system. The initial time of 0 corresponds to when the shaker is first turned on and the fluid bath is at or near the temperature of the room.

Theory

There are all kinds of fascinating orbits that arise when two drops have different sizes (see Protiere 2008), but in this paper only circular orbits where both drops are the same size will be considered. The theory behind orbiting pairs has been first developed by Couder's group in Paris. In their 2006 paper, *Particle-wave association on a fluid interface* they found that the diameters d_n^{orb} of circular orbits are discrete:

$$d_n^{orb} = (n - \varepsilon_0)\lambda_F \quad (1)$$

where n can have values of the whole integers, n = 1, 2, 3,… which correspond to two orbiting drops that bounce in phase with each other and values of the half integers, n = 1/2, 3/2, 5/2,… which correspond to two orbiting drops that bounce out of phase with each other. So each orbit has a diameter which is a multiple of half the Faraday wavelength, λ_F, with a slight offset ε_0. The reason for this offset is a direct consequence of the drop needing a centripetal force to keep it in orbit. This force comes from the wave propagated by the other drop. If the first drop is a wavelength away from the other drop, it will simply land on the peak of the wave where the slope is zero. However, if the drop is slightly closer to the other drop, then it will land inward of the wave crest and receive a kick toward the other drop. Therefore, this offset must exist for centripetal motion to occur.

According to the model by Couder's group, the horizontal motion of the orbiting drops can be described by a set of coupled differential equations:

$$m\frac{d^2\boldsymbol{r}_1}{dt^2} = F^b \sin\left(\frac{2\pi\|d\boldsymbol{r}_1/dt\|}{V_\phi}\right)\frac{d\boldsymbol{r}_1/dt}{\|d\boldsymbol{r}_1/dt\|} + \alpha F^b \frac{(\boldsymbol{r}_1 - \boldsymbol{r}_2)}{(\boldsymbol{r}_1 - \boldsymbol{r}_2)^2}\sin(k_F\|(\boldsymbol{r}_1 - \boldsymbol{r}_2)\|)$$

(2)

$$m\frac{d^2\boldsymbol{r}_2}{dt^2} = F^b \sin\left(\frac{2\pi\|d\boldsymbol{r}_2/dt\|}{V_\phi}\right)\frac{d\boldsymbol{r}_2/dt}{\|d\boldsymbol{r}_2/dt\|} + \alpha F^b \frac{(\boldsymbol{r}_2 - \boldsymbol{r}_1)}{(\boldsymbol{r}_2 - \boldsymbol{r}_1)^2}\sin(k_F\|(\boldsymbol{r}_2 - \boldsymbol{r}_1)\|)$$

where α is a damping term which is proportional to the Faraday wavelength and decays radially from the position of the drop . For simplicity, this decay can be ignored. Keep in mind, these two equations would be slightly different if the two drops were different sizes (see Protiere 2006). The second term on the right of each equation corresponds to the interaction of the drop with the wave created by the other drop and provides the centripetal force. Equations (2) assume the two drops are in phase, so if the drops are out of phase then this interaction term should be negated. If we let $\alpha=\lambda_F$, and let $d_n^{orb} = \|(\boldsymbol{r}_2 - \boldsymbol{r}_1)\|$ then we can set the interaction term equal to the centripetal force to get the following relation:

$$\sin\left(\frac{2\pi d_n^{orb}}{\lambda_F}\right) = -\frac{2mV_w^2}{\lambda_F F^b} \quad (3)$$

where the speed of the orbiting drop has been approximated to be V_w, the velocity it would have as a walker. A solution for the diameter d_n^{orb} can only exist if the right side of this equation has a magnitude less than or equal to 1. There are two sets of solutions in the form of Equation (1) with the following offsets values (see figure 4).

$$\varepsilon_0' = \frac{1}{2\pi} Arc\sin\left(\frac{2mV_w^2}{\lambda_F F^b}\right)$$
$$\varepsilon_0'' = \frac{1}{2} - \frac{1}{2\pi} Arc\sin\left(\frac{2mV_w^2}{\lambda_F F^b}\right) \quad (4)$$

In Protiere 2006 they found that only ε_0'' results in stable orbits. Unfortunately, they also found that this solution gives an overestimate of the offset value found experimentally. As ε_0 varies with drop size, Couder's group found that most orbits have an ε_0 in a range between 0.15 and 0.25 experimentally, while the theoretical value for the stable solution, ε_0'' is closer to 0.45.

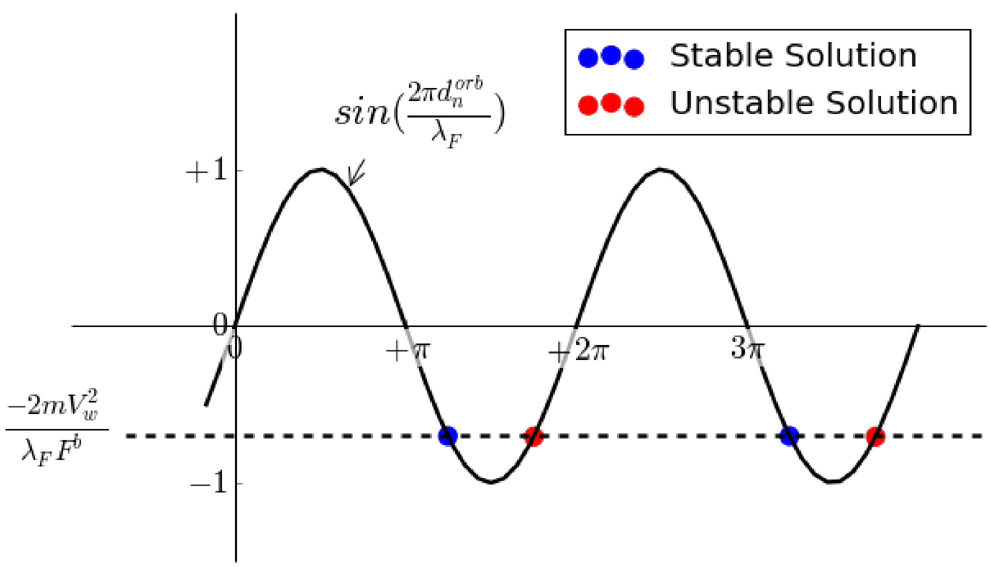

Figure 4: Graphical analysis of equation 3 which shows 2 sets of solutions. The first set, in blue, is stable while the other set, in red, is unstable. Note, if the value on the right side of equation 3 is less than -1 then there are no solutions, if it is exactly -1 then there is one set of solutions.

Results

According to Equation 4, the value for ε_0 between two identical orbiting drops should depend only on the velocity of the walkers, the size of the drops, the acceleration, and properties of the fluid. Therefore, if an orbiting pair transitions to a different orbiting distance, then ε_0 should have the same value as before, since none of these parameters change in the process. By creating an orbiting pair and sending a plane wave across the bath surface, we occasionally observed a transition from one orbiting distance to another. Most of the time the perturbation had no effect on the system, especially if the orbiting pair was not close enough to the wave source. For the cases where the perturbation did result in a transition, videos were taken and analyzed to find the offset ε_0 for each orbiting state, as shown in Table 1.

Orbit #	n_1	n_2	ε_0 for n_1	ε_0 for n_2
1	1.5	1	0.20 ± 0.01	0.20 ±0.01
2	1.5	1	0.10 ± 0.01	0.12 ± 0.01
3	1.5	1	0.27 ± 0.02	0.26 ± 0.02
4	1.5	1	0.25 ± 0.02	0.23 ± 0.02
5	1	0.5	0.21 ± 0.05	0.16 ± 0.03
6	1	0.5	0.15 ± 0.02	0.12 ± 0.02
7	1.5	1	0.23 ± 0.02	0.24 ± 0.02
8	0.5	1	0.11 ± 0.01	0.11 ± 0.01

Table 1: Results for several orbit transitions. Values of n_1 and n_2 correspond to the original and final states of the orbit, respectively. The values for the offset ε_0 are reported for each orbital state.

The first observation to be made from these results is each orbiting distance only changed by half of a wavelength. This doesn't necessarily mean that it is impossible for an orbit to make a transition that changes its distance by more than half a wavelength, but it is highly unlikely given how difficult it was to make the orbits transition at all. This difficulty can likely be attributed to the fact that when the distance between two orbiting drops changes by half a wavelength, it means the drops either went from being in phase to out of phase or vice-versa. In Protiere 2005 they describe a process in which they had to bring an orbiting pair into their bouncing mode and then back into walking mode in order to change the phase of the drops, which shows how remarkable it is that these perturbations were able to change the phase of a droplet without any other changes in the system. When a perturbation in the fluid resulted in an orbit transition, there was usually a transient state in which the distance between the two drops oscillated before settling into a stable orbit. It seems that in order for a transition to take place, the plane wave must hit one of the drops just right so that it changes its phase, at which point the drop is unstable and must either move half a wavelength closer (assuming its orbit state is not $n = 0.5$) or half a wavelength farther away from the other drop.

It is also worth noting that the majority of the orbits started out in the $n = 1.5$ state. This means that most drops were out of phase with each other in the beginning. There is no reason to believe that one phase orientation between two drops is more likely than the other, so this result can most likely be considered a coincidence and should be ignored, especially since the sample size is so small. For two drops out of phase with each other, it makes sense that their orbiting state is more likely to be $n = 1.5$, since the only reasonable alternatives are $n = 0.5$ and $n = 2.5$. It is unlikely for a pair of orbiting drops to be found in the $n = 2.5$ state because the waves being created by each drop are damping out radially and so 2.5 wavelengths is too far for two drops to be captured into an orbit. It is also unlikely for an orbit to be in the $n = 0.5$ state because in order for this to happen, the two drops must at some point be 1.5 wavelengths away from each other, where the two drops are clearly close enough to be captured into an orbit.

Finally, looking at the measured values for ε_0 we see that the offset never made a dramatic change for an orbit transition, as predicted. In fact, it is far more likely that ε_0 didn't change at all and that any measured change is due to experimental error, which is supported by the measured uncertainties. The values for ε_0 were found by measuring the distances between the orbiting drops d_n^{orb}, the wavelength λ_F, and then plugging these values into equation (1). These measurements were all found in units of pixels which was not an issue in the measurement of ε_0, since it is a dimensionless value. After taking several measurements for each value, uncertainties were found by calculating the standard deviation of the measurements. Table 1 shows that all values for ε_0 where in the range between 0.10 and 0.27, which is roughly in agreement with the findings of Protiere 2005.

Conclusion

When this project was undertaken in the summer of 2014 by Bryce Parry and myself, under the advisement of Dr. Nilgun Sungar, the first and primary goal was simply to have the experiment set up and working. The aluminum tray, which contained the fluid bath, had to be designed and machined, so having this done was a feat in and of itself. The electronics, which consisted of the components needed to oscillate the tray vertically and the devices used to read the acceleration of these oscillations, were fairly easy to implement since the setup was nearly identical to the setup used on a previous version of the experiment, which was developed by Lisa Slaughter and Zech Thurman. Over the course of about two months, all of this was accomplished along with the development of an imaging system and some data collecting tools which included a particle tracking program in matlab.

Soon after the experimental apparatus was completely set up, we focused our attention on the quantized orbits between two identical drops. We found that by perturbing their motion, orbits can transition to other orbital states where the distance between the drops changes. In this process the phase of one of the drops changes, which is somewhat surprising given that all the parameters of the vibrating tray stay the same. If this process could be filmed with a high speed camera, it could potentially make it possible to understand how the drop changes phase and what criteria are necessary for such a phenomenon to occur.

The values we found for the orbital offsets, ε_0 were very similar to the values found by the research group in Paris. This is very reassuring for future experiments with this apparatus in which similar measurements will be made. The theoretical framework developed by Couder's group in Paris is very useful in understanding the behavior of two orbiting drops. However, there is the one issue where the theoretical predictions for ε_0 did not match the experimental values. It is possible that such an issue can be resolved, at least partially, by using a more sophisticated theoretical approach and incorporating the travelling waves of previous impacts of the drops, which would contribute to the interaction terms in equations (7). In MB2, their model for the height of the fluid bath, $h(x,t)$ incorporates the previous impacts that a drop has with the fluid, and does so by taking a sum of Bessel functions of zero order, where the terms corresponding to longer time intervals are smaller since the waves decay with time. This model, however, only incorporates the motion of the standing waves and does not include the travelling wave which precedes the stationary waves. Therefore, this approach is not good for two interacting drops and taking an approach similar to Fort 2010 (see equation 10) would be better.

In the process of conducting these experiments, there were a few needs of improvement found. First, there was the issue of the Faraday threshold changing with time due to the heating of the electromagnetic shaker. This issue was partially fixed with a cooling system. As previously mentioned, measuring the temperature and using this to make any necessary adjustments in calculations would make measurements more reliable. Also, it is important that measurements using the imaging system can be converted from pixels to length. One way of doing this would be to make marks at the bottom of the tray with known proximities which can be identified in a photo or video and then used for a length calibration. Additionally, there were instances where the particle tracking program did not work due to poor image quality. Therefore, either imaging techniques need to be improved or the particle tracking program needs to be modified so that even videos with poor image quality can be analyzed, which is not an unreasonable task. Lastly, it is worth noting that the group of researchers at MIT working with John Bush recently published a paper on the implementation of a drop maker for studies in hydrodynamic pilot-wave systems. This device is capable of making a drop of fluid with a prescribed diameter. This makes it much easier to make several drops of the same size, which is especially useful in studies of orbits and arrays. Adding such a device to this system would be highly advantageous and is currently being undertaken by Dr. Nilgun Sungar and students. So, while it is still a work in progress, the current experimental setup is clearly capable of making valuable measurements and with a few adjustments and additions, the system will be fully prepared to perform repeatable experiments and make reliable measurements, worthy of contribution to the relatively new field of hydrodynamic pilot-wave theory.

References

Bush, J. W. M. 2010 Quantum mechanics writ large. Proc. Natl Acad. Sci. 107, 17 455–17 456.

Bush, J. W. M. 2015 Pilot-Wave Hydrodynamics. Annu. Rev. Fluid Mech. 2015. 47:269-92

Couder, Y. & Fort, E. 2006 Single-particle diffraction and interference at macroscopic scale. Phys.Rev. Lett. 97, 154101.

Eddi, A., Fort, E., Moisy, F. & Couder, Y. 2009 Unpredictable tunneling of a classical wave-particle association. Phys. Rev. Lett. 102, 240401.

Fort, E., Eddi, A., Boudaoud, A., Moukhtar, J. & Couder, Y. 2010 Path-memory induced quantization of classical orbits. Proc. Natl Acad. Sci. 107, 17 515–17 520.

Protiere, S., Boudaoud, A. & Couder, Y. 2006 Particle-wave association on a fluid interface. J. Fluid Mech. 554, 85–108.

Molacek,J. & Bush,J.W.M. 2013 Drops walking on a vibrating bath: towards a hydrodynamic pilot- wave theory. J. Fluid Mech. 727, 612–647

Slaughter, Lisa. 2013 Viscosity dependence of faraday wave formation thresholds. Senior Project.

Appendix A: Accelerometer Acquisition Code

```
% Accelerometer data acquisition code
% By Bryce Parry and Joshua Parker, physics
% California Polytechnic State University, SLO
% Summer 2014
% Public domain
% To be used with Arduino code 'AccelerometerDataInput.ino'
%
% **Ensure that the same COM port is being used here as the Arduino
% **Ensure that baud rates also match (max: 115200)
%
% This code takes in data from an accelerometer, calibrates the system
% and then outputs the acceleration recorded as both a graph and a specific
% maximum acceleration.
%% ----------Calibration----------
clear all
close all
clc

% The system was calibrated by placing the accelerometer on a level surface
% and recording the voltages at what should be observed as zero g.

x0g = 510.3;
y0g = 510.3;

% The accelerometer was then positioned on one side of the test tray and
% the active voltage readings were used to level the instrument by
% comparing with the zero g readings above. The negative one g reading was
% observed here.

x1g = 531;
y1g = 531;

% Now that the controlled voltages have been recorded and the accelerometer
% has been glued on the side of the test tray, data can be taken.
%% ----------Data Acquisition----------

numpts = 100;               % Number of data points
delete(instrfindall)        % Frees COM port
arduino = serial('COM3','BaudRate',115200);     % Addresses the device
data = ones(numpts,3);          % Initializes the data array.
fopen(arduino);          % Opens serial object
k = 0;                   % Data point index initialized
for i = 1:120
 [temp,count] = fscanf(arduino);     % Get the data
 if i>10
 if count ~= 0
 commas = 0;
 for j=1:length(temp)                % Count the commas
 if temp(j) == ','
 commas = commas + 1;
 end
```

```
 end
 else
 disp('No values read');
 end
 if commas == 2
 k = k+1;                                       % Adds to the data index
 predata = textscan(temp,'%u,%u,%u'); % Convert to cell
 data(k,:) = cell2mat(predata);                 % De-cell into vector.
 if k == numpts                         % Limits number of data points
    break
 end
 end
 end
end
fclose(arduino);                % Close serial object
delete(instrfindall);           % Free up com port

% Data organization/conversion
data(:,1) = data(:,1) - data(1,1);  % Starts time at 0 ms
t = data(:,1);
accx = (data(:,2) - x0g) / (x0g - x1g);       % Conversion to g's
accy = (data(:,3) - y1g) / (y0g - y1g);       %   "    "    "    "

% Plot the data on a plane
figure(1)
hold on
plot(t,accx,'r-');
%plot(t,accy,'g-');
hold off
%% Peak test for amplification
Y = accy(1:100);
Yabs = abs(Y);
k = 0;
for i = 2:99
    if Yabs(i)>Yabs(i-1) && Yabs(i)>Yabs(i+1)
        k = k+1;
        peak(k) = Yabs(i);
    end
end
mpeak = mean(peak)
%% Line of best fit
X = t(1:100);
Y = accy(1:100);
B0 = mean(Y);                   % Vertical shift
B1 = (max(Y) - min(Y))/2;       % Amplitude
B2 = 2*pi/1000*50;              % Frequency
B3 = 1;                         % Phase shift (eyeball the Curve)
myFit = NonLinearModel.fit(X,Y, 'y ~ b0 + b1*sin(b2*x1 + b3)', [B0, B1, B2,
B3])

% Note that all the coefficient estimates are very good except for b3 where
% any even integer is equally valid

%Generate a plot
hold on
scatter(X,Y)
```

```
plot(X, myFit.Fitted)
hold off
```

Appendix B: Particle-Tracking Code

```
      %% Multidrop Particle Tracker

% By Bryce Parry and Joshua Parker, physics
% California Polytechnic State University, SLO
% Summer 2014
% Public domain
%
% ** It is necessary that the drops of interest are illuminated more than
%  the background of the video.
%
% This code takes in a video file and requires that the user clicks on the
% the drops of interest (must be the bright points in the field of view),
% then tracks the motion, recording their positions (in terms of the pixel
% position) in an array. It then graphs the data and plays the video back
% to assure that the data taken was accurate to the motion of the particle.

%% Particle Tracking
clear all
tic

vid = VideoReader('Droplet_movies/DSCF0494.mov');          % Imports video
nFrames = vid.NumberOfFrames;                   % Takes the number of frames
% nFrames = 900;
vidFrame = read(vid,1);                                % Reads the first frame
imagesize = size(vidFrame);                   % Takes in the size of the image

figure(1)
imagesc(vidFrame);                                         % Scales the image
pos = ginput;                        % Initializes the drops via clicking them

psize = size(pos);                  % Returns the size of the position array
arraylen = 1 + 2*psize(1);             % Array length for any # of particles
postrack = zeros([nFrames,arraylen]);          % Data array of correct size.

for a = 2:2:arraylen          % Inputing initialized drops into data array
    index = a/2;
    postrack(1,a) = round(pos(index,1));                        % x-coordinate
    postrack(1,a+1) = round(pos(index,2));                      % y-coordinate
end

for i = 2:(nFrames-200)                             % Looks at every frame of
video
    vidFrame = read(vid,i);                            % Reads the given frame
    for p = 2:2:arraylen                    % Runs a loop 'for' every particle
        init = 0;                           % Creates brightest pixel variable
        max_xdisp = 20;                          % Max pixel displacement in x
        max_ydisp = 20;                          % Max pixel displacement in y
```

```
        postrack(i,1) = i-1;                        % Appends frame number
        xlast = postrack(i-1,p);            % Takes in the last x position
        ylast = postrack(i-1,p+1);          % Takes in the last y position

        % The following code is for debugging. When the particle approaches
        % the edge of the frame, this piece of code will prevent searching
        % outside of the image (at an invalid array index).

        if ylast < max_ydisp
            ylast = round((ylast+max_ydisp)/2);
            max_ydisp = ylast;
        elseif ylast > imagesize(1)- max_ydisp
            ylast = round((ylast-max_ydisp+imagesize(1))/2);
            max_ydisp = imagesize(1)-ylast;
        end

        if xlast < max_xdisp
            xlast = round((xlast+max_xdisp)/2);
            max_xdisp = xlast;
        elseif xlast > imagesize(2)- max_xdisp
            xlast = round((xlast-max_xdisp+imagesize(2))/2);
            max_xdisp = imagesize(2)-xlast;
        end

        % The next piece of code finds the brightest point in an area
        % (defined by the max x/y displacement) by running a for loop
        % through the square/rectangle of pixels. If the current pixel
        % brightness is brighter than the running variable (called 'init'),
        % it will append that (x,y) position to the data array and set
        % the init variable to that pixel brightness.

        for x = xlast - max_xdisp: xlast + max_xdisp     % Set of x-values
            for y = ylast - max_ydisp: ylast + max_ydisp        % y-values
                if vidFrame(y,x) > init     % Current pixel > previous max
                    init = vidFrame(y,x);                 % Reassigns init
                    postrack(i,p) = x;                  % Appends x-position
                    postrack(i,p+1) = y;                % Appends y-position
                end
            end
        end
    end
end

% Necessary for graphing the position, is a consequence of the way pixels
% are located in images, in MATLAB.

% for d = 2:2:arraylen
%     postrack(:,d+1) = postrack(:,d+1)*(-1) + 1080;
% end

%% Graphing the images
xl = imagesize(2);                                          % X axis limit
yl = imagesize(1);                                          % Y axis limit
maxtime = nFrames/30;
maxx = max(max(postrack(:,2:2:arraylen)));
```

```
maxy = max(max(postrack(:,3:2:arraylen)));
minx = min(min(postrack(:,2:2:arraylen)));
miny = min(min(postrack(:,3:2:arraylen)));
fit = 20;

figure(2)
set(axes,'YDir','reverse')
hold on
for b = 2:2:arraylen
    plot(postrack(:,b),postrack(:,b+1))                    % Graphs y vs x
end

axis([0 xl 0 yl])
title('Graphing Y vs X of all particles')
xlabel('X')
ylabel('Y')
hold off

figure(3)
subplot(2,1,1)
hold on
for b = 2:2:arraylen
    time = postrack(:,1)/30;
    plot(time,postrack(:,b))                               % Graphs x vs t
end
axis([0 maxtime minx-fit maxx+fit])
title('Graphing X vs Time of all particles')
xlabel('Time(sec)')
ylabel('X')
hold off

subplot(2,1,2)
hold on
for b = 2:2:arraylen
    time = postrack(:,1)/30;
    plot(time,postrack(:,b+1))                    % Graphs y vs t
end
axis([0 maxtime miny-fit maxy+fit])
title('Graphing Y vs Time of all particles')
xlabel('Time(sec)')
ylabel('Y')
hold off

%% Two droplet analysis
% ** To be used only after analyzing two body interactions
% (postrack size = nFrames x 5)
if arraylen == 5
    xdif = postrack(:,2) - postrack(:,4);
    ydif = postrack(:,3) - postrack(:,5);
    r = sqrt(xdif.^2 + ydif.^2);
    maxr = max(r);
    meanr = mean(r);

    figure(4)
    hold on
    for b = 2:2:arraylen
```

```
        time = postrack(:,1)/30;
        plot(time,r)
    end
    axis([0 maxtime 0 maxr+fit])
    title('Separation Distance vs Time')
    xlabel('Time(sec)')
    ylabel('Separation Distance')
    hold off

    hold on
    rf = r - mean(r);
    ft = fft(rf);
    ps = ft.*conj(ft);
    a = size(ps);
    f = 0:1:a(1)-1;
    f = f/maxtime;
    fmax = max(f)/2;
    psmax = max(ps)+max(ps)*.1;
    figure(5)
    plot(f,ps);
    axis([0 fmax 0 psmax])
    title('Power Spectrum')
    xlabel('Frequency(Hz)')
    ylabel('Power')
    hold off
end
toc
```