

Supplementary Materials

Fig. S1 Arena Dimensions	1
Fig. S2 Acclimation Boxplot	2
1. Ocean Acidification and Squid Egg Capsule Culture System	2
2. Recording 3D Swimming Behavior Videos	6
2.1 3D Swimming Behavior Recording Protocol	6
3. Analysis of 3D Swimming Behavior Data	7
3.1 Organism Tracking Protocol	7
3.2 Data Analysis Protocol	10
3.3 List of Model Variables	10
3.4 3D Behavior Data Correction MATLAB code	11
3.5 3D Behavior Analysis MATLAB code	12
4. References	17

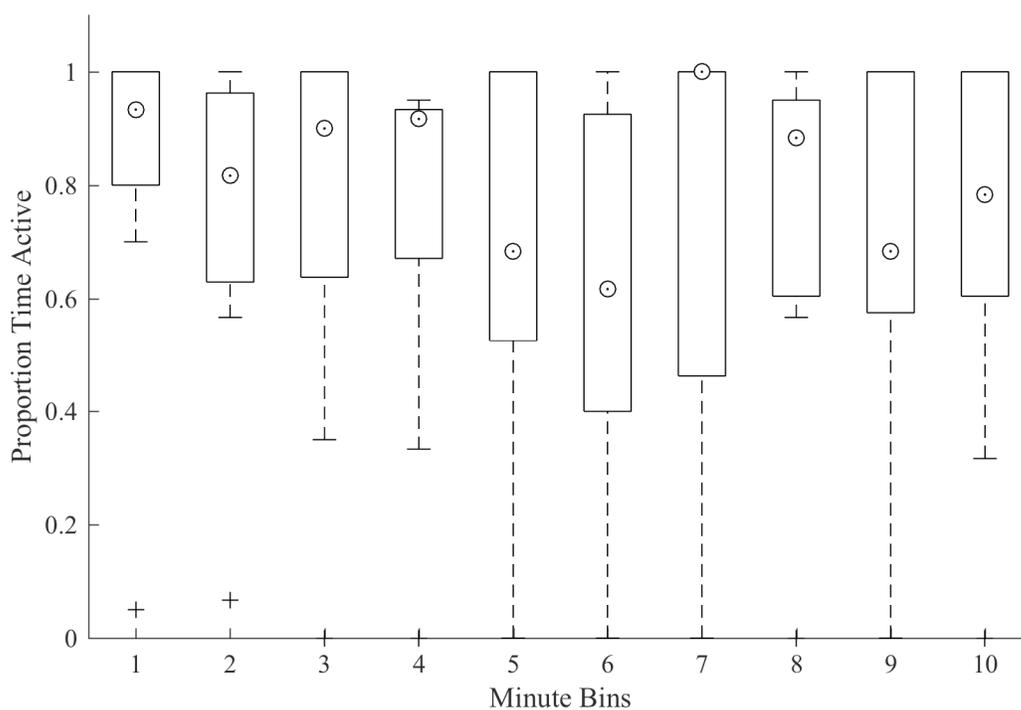


Fig. S1 The proportion of individual paralarval ($n = 9$) time spent active in one-minute bins over a 10-minute period in the 3D experimental arena. Paralarvae from all 4 CO_2 treatments were used and no consistent pattern with CO_2 exposure was seen, so all data is compiled here. Dotted circles denote medians and plus signs denote outliers

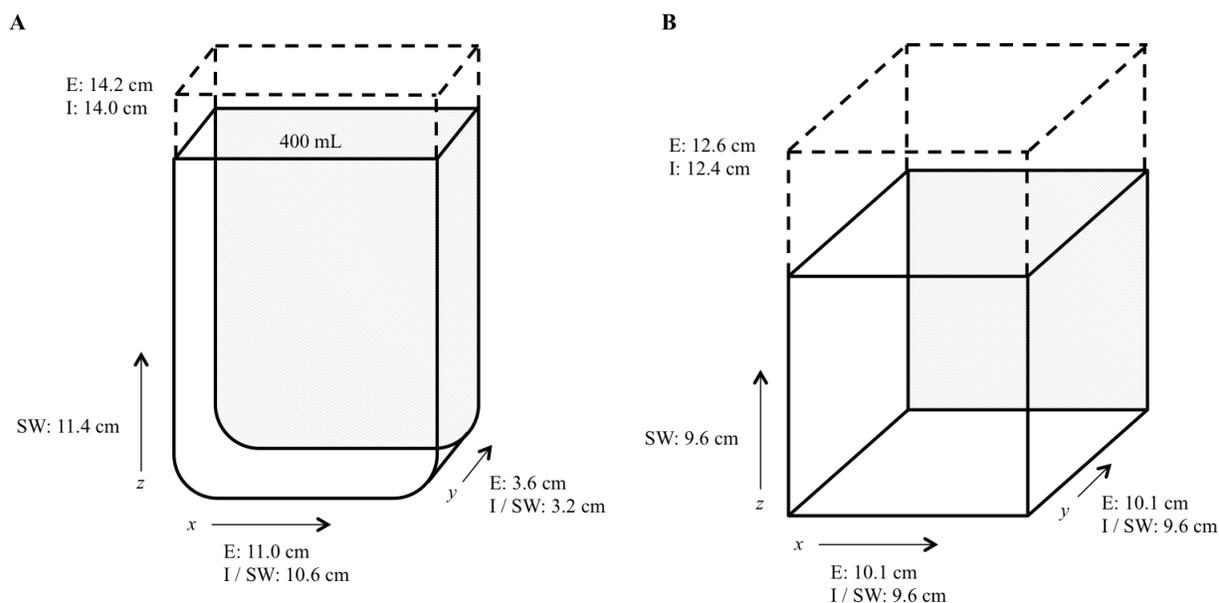


Fig. S2 (A) Dimensions and axes of the 2D arena, constructed from a tissue culture flask. 400 mL of treatment-appropriate seawater was used. (B) Dimensions and axes of the 3D arena constructed from a plastic box. Seawater was added to fill to an internal cube of 9.6 cm. For both schematics, solid lines indicate the seawater space used in the experiments, while dotted lines indicate the additional makeup of the container. Measurements are provided for both external (E) and internal (I) dimensions of the containers, and seawater volume (SW) utilized. The z axis is oriented as it was for tracking, in the bottom, front, left (0,0,0) corner. Note that this was transformed, so that 0 was at the water surface, for manuscript figures depicting depth. Figures are not drawn to scale

1. Ocean Acidification and Squid Egg Capsule Culture System

Experiments took place at the WHOI Environmental Systems Laboratory (ESL), Woods Hole, MA from May - October of 2013, 2014, and 2015. A flow-through ocean acidification (OA) system was equilibrated to different CO_2 levels for each trial. Ambient ocean water was pumped in from the ESL intake, passed through the facility's sand-filtration system, and was heated to 20°C . This temperature reflected peak values reached during the breeding season and resulted in a consistent two-week development period for the ambient paralarvae. The seawater was then passed through a $10\mu\text{m}$ filter (Hayward FLV Series industrial filter equipped with

10 μ m felt bag) to remove major particulates and a UV sterilizer (Emperor Aquatics Smart HO UV Sterilizer, Model 025150) to eliminate potentially harmful protozoans. The cleaned water subsequently flowed into the experimental system's header tank and was bubbled with compressed air. Water flowed in a gravity-based system from the header tank and was split among four treatment lines into H-shaped PVC equilibration (EQ) chambers. Each leg of an EQ chamber contained two air stones, which bubbled in the appropriate CO₂ mixture. Due to the elevated CO₂ concentration of ESL seawater (550 ppm in facility compared to 400 ppm at the pump intake in Vineyard Sound) the ambient treatment line was comprised of three EQ chambers, the first degassing with N₂ and the following two reintroducing O₂ and CO₂ at ambient concentrations. Measurements of dissolved oxygen downstream of the nitrogen system were all above 95%. Due to a facility malfunction, Trial 3 of 2013 lost oxygenation downstream of the nitrogen degassing and was not used. To reduce the growth of brown algae, which was encouraged by the nitrogen enrichment, lines were cleaned between each trial and a 1 μ m filter was placed downstream of the EQ chambers on the ambient line.

Compressed air, delivered at 30 psi from an indoor air compressor, was delivered using a six-way manifold to the header tank aeration, ambient EQ chambers' air stones, and three mass flow controllers (Aalborg GFC17, Orangeburg, NY, USA) with a flow rate of 4.5 l min⁻¹. Pure CO₂ was delivered from a cylinder at 30 psi to three further mass flow controllers (Aalborg GFC17), which were set at various flow rates to produce the desired CO₂ concentrations in the gas mixtures. Air and CO₂ lines were connected downstream of the mass flow controllers and allowed to mix before being split among the EQ chambers' air stones and experimental aquaria bubbling lines. CO₂ mixtures covered a range of values between ambient (400 ppm) and 2200 ppm (Main Text, Table 1). CO₂ concentrations for each level were measured before the start of

each trial on a Qubit Systems CO₂ Analyzer (model s151) calibrated with three commercial reference standards (0, 362, and 1036 ppm).

Equilibrated water in each treatment line outflowed into a PVC manifold and entered drip lines connected to the experimental aquaria, consisting of individual 1-liter PET food service containers (Solo Foodservice, Lake Forest, IL). The containers had been seawater soaked for 24 hours and DI-water rinsed in advance to remove any potential chemical residues or toxins. Cups were sealed with fitted lids pierced with two holes for the gas and equilibrated water lines. Each cup had a small (2x4 cm), screened (5 µm mesh) hole to permit water outflow while retaining paralarvae. The bubbler of the gas line was placed under the screen to create flow and prevent paralarvae from sticking to the screening. Water outflowed from the cups into a common water bath maintained at a temperature of 20°C via aquarium heaters (JÄGER 3603, EHEIM GmbH and Co., Deizisau, DE) and chillers (Oceanic Aquarium Chiller 1/10hp, Oceanic Systems, Walnut Creek, CA, USA).

In 2013, each trial consisted of three CO₂ treatments, each with three experimental cups and one organism-free control cup, totaling twelve cups per water bath. In 2014 and 2015, each trial consisted of four CO₂ treatments, totaling sixteen cups per water bath. The system was run for several days prior to a trial to ensure stable water and gas levels. Flow rates to the cups were set at a slow drip, approximately 20 liters day⁻¹, which prevented waste accumulation. The flow rate also allowed sufficient time for bubbled gas to equilibrate with water in the EQ chambers. The ESL room containing the experimental set up was kept on a 14:10 hour light:dark photoperiod, reflecting the average photoperiod of the region, using ceiling mounted fluorescent lighting. Water bath temperature and ambient light levels were monitored using an Onset HOBO data logger (pendant model UA-004-64), one in each water bath, with recordings taken every 15

minutes. Temperatures were 20.49 ± 0.69 °C, 20.36 ± 1.80 °C, and 20.01 ± 1.00 °C (mean \pm SE) in water bath 1 and 20.26 ± 0.49 °C, 20.18 ± 1.74 °C, and 20.51 ± 1.91 °C in water bath 2 in 2013, 2014, and 2015, respectively.

Once squid were brought to the ESL holding tanks, females began laying eggs within two to three days, producing small egg mats typically found at the bottom of the tank or attached to the air hose. The morning an egg cluster was found, it was removed to a holding container and examined for quality. Individual egg capsules, each containing between 50-200 eggs, were then randomly selected and randomly assigned to each experimental cup, with two egg capsules per cup (18 egg capsules per trial for 2013, 24 for 2014 and 2015), marking the start of a trial.

During a trial, pH measurements were taken for samples from each cup every three days using a pH meter (Orion Star™ A329, Thermo Fisher Scientific Inc., Waltham, MA, USA) in order to monitor CO₂ level stability. The fourth cup from each CO₂ level contained no eggs sacs and was used as a control for water quality and carbonate chemistry measurements.

Spectrometric pH measurements were taken with a spectrophotometer using methods adapted from Clayton and Byrne (1993) and Dickson et al. (2007) Salinity was measured using a salinity probe (Orion Star™ A329, Thermo Fisher Scientific Inc., Waltham, MA, USA) in parallel to spectrophotometric pH readings. Total alkalinity (AT) samples were taken in 20mL acid-washed, glass scintillation vials and poisoned with 10 μ L saturated mercuric chloride (HgCl₂). Alkalinity samples were analyzed post-trial using an automated small volume titrator (Titrand 808, Metrohm AG, Herisau, CH) programmed to run Gran titrations of 1mL subsamples. Samples were run in duplicate and calibrated against standards of ESL water of known alkalinity. For duplicates with a difference of 4 μ mol kg⁻¹ seawater (SW) or greater, samples were rerun and an average of the four values was taken. Carbonate chemistry metrics (temperature, salinity, pH,

and alkalinity) were input into CO2SYS, using dissociation constants from Mehrbach (1973) and sulfate constants from Dickson (1990), to calculate pCO₂ and aragonite saturation state (Ω_{arag}) for each treatment of each trial. A baseline measurement of all cups was taken prior to the initiation of a trial, followed by weekly readings of only the control cups once a trial had begun. Measured seawater CO₂ concentrations fell within reasonable range of the equilibrations desired from the gas concentrations (Main Text, Table 1).

2. Recording 3D Swimming Behavior Videos

2.1 3D Swimming Behavior Recording Protocol

1. Before sampling your organism, make sure the setup is complete as follows (Note: all cameras and the organism aquaria/chamber were placed within a covered photography light box to block ambient light):
 - a) Fill the chamber with seawater appropriate to the treatment you are testing. The water level needed to make a perfect cube (9.6 cm for this chamber) is marked with a small black line. Remove any bubbles by scraping with a ruler.
 - b) Secure the top camera to a sturdy frame above where your chamber will be placed.
 - c) Place the chamber beneath the top camera. Adjust the top camera zoom so the entire chamber is clearly visible within the image frame.
 - d) Center the chamber in the camera view. This can be done by measuring the distance between the sides of the chamber and the edge of the image frame for each side and making this distance on both sides equal.
 - e) Ensure the camera is viewing the center of the chamber straight on and is not skewed to any angle. This can be done by measuring the length (in the camera image) from the top (front) edge of each side of the chamber to its bottom (back) edge and adjusting the position of the camera and chamber to make these distances equal.
 - f) Place the side camera on a raised block in front of the chamber, such that the camera points to the center of the chamber. Adjust the zoom so the entire chamber is clearly visible. Center the camera on the chamber and adjust skew, as above. Connect the cameras to viewing monitors outside your light box.
 - g) Place one light on either side of the chamber (perpendicular, non-biased lighting is necessary for photopositive organisms only). Set lighting so your organism is

clearly visible in both camera images and reflections from your chamber are reduced as much as possible.

2. Select one individual and place it into the center of the chamber.
3. Close the cover of the light box. Allow organism to acclimate to chamber (one minute was suitable for squid paralarvae).
4. Start timer and record for desired length (two minutes was suitable for squid paralarvae).
5. Before stopping the recording, use a light, laser, or other visual cue so that both videos can easily be synced for 3D analysis.
6. Stop the recording. Remove organism. Repeat steps 3-7 for as many individuals as you have per treatment.
7. Record file number for top and side camera and relevant experimental data/individual identification on a data sheet, along with any notes on video quality (good swimming, corner, etc).
8. Note: Remember to change water in filming chamber as needed as you progress through your treatments. The camera adjustments outlined in step 1 must be repeated/checked each time the organism chamber is removed or replaced.
9. Upload video files to an external hard drive (recommended due to size and number of videos). Save files in a dated folder, and within that, either a “top” or “side” subfolder.
10. Double check to ensure your recordings are saved and backed up before reformatting the memory cards for the next recording session.

3. Analysis of 3D Swimming Behavior Data

3.1 Organism Tracking Protocol

1. Drag your video into Tracker (available at OpenSourcePhysics.org). It is best to analyze videos one minute at a time because Tracker can slow down and can run out of memory (amount of memory available to Tracker can be adjusted by clicking **memory in use** → **Set memory size...**). Save as you go. If the memory allotment gets overloaded the program can crash. At a higher memory allotment, a full two-minute video can be analyzed in one Tracker run. In order to clear the memory cache, it is easiest to close and reopen Tracker for each video analyzed.
2. In the top right corner, click the “refresh” button then deselect **Auto-refresh**. This prevents available memory from filling up as you track, which slows down the process.
3. You may need to correct for stretching due to Tracker’s pixel size, which may differ from those used by your camera. Click **Video** → **Filters** → **New** → **Resize**, then adjust. For the Sony HD Handycams used in the squid experiment the adjustment was:

	Width	Height
Input	1440	1080

Output	1440	800
--------	------	-----

4. Adjust the axis to the desired position by clicking the magenta axis button (**show or hide the coordinate axes**) on the tool bar. From the side view, the origin (0,0) point should be in the top left corner of the chamber with the y-axis aligned with the inside of the left edge of the chamber and the x-axis aligned with the water level. You can tilt the axes, if needed, by clicking and dragging them. The axes for the top view depend on how your camera is oriented in relation to chamber. The origin for all the data should be the top, left, front corner of the chamber. Our top camera was placed such that it viewed the chamber with the front edge at the top of the image. Thus, we placed the origin for tracking top videos at the top right corner of the chamber with the y-axis aligned with the inside right edge of the chamber and the x-axis aligned with the inside top edge. This produced negative values for both x and y data that had to be later reoriented (multiply by -1) to the correct reference frame.
5. Adjust the scale by pressing the blue **Show, hide, or create calibration tools** on the toolbar. Click on **new** → **calibration tape**. A blue, double-headed arrow will appear on screen. Drag the ends of the calibration tape across a known distance, typically the length of your chamber (9.6 cm between the inside edges of the chamber). You can zoom in to be as accurate as possible placing the ends of the tape. Make sure the angle of the tape to the x-axis is 0. Double click on the box with the blue number and change it to your known length (9.6 in our case) once adjusted perfectly.
6. Frame selection:
 - a. Calculate the total number of frames analyzed for 1 minute of video. For our cameras this was 1799, since the camera filmed at 29.97 frames per second.
 - b. To calculate the frames you would like to analyze, find the frame where the laser pointer first appears signaling the end of the recording and use the frame before as the last. Subtract your desired number of frames -1 (1798, in our case) from the last frame to get the first frame (the frame range is inclusive, hence we subtracted 1798 not 1799).
7. To apply the desired frame range, click on the **Clip Settings** button on the toolbar. Enter the frame numbers in the window that appears. You can tell the range has been applied because the small black arrows beneath the video scanning bar at the bottom of the screen will have moved to enclose the desired range.
8. To start a new track, click **Create a new track** → **Point Mass**. Click on the **mass a** button in the Track Control window and click the **Autotracker...** button, which will open the Autotracker window.
9. Autotracker: (refer to **Help** → **Tracker Help** → **Autotracker** for further explanation)
 - a. Autotracker works best when the organism is a well-lit white dot and against a black background.

- b. To start, **shift + control + click** on the animal to be tracked. The numbered point is the **target**; its position is what will be marked in the data column. The circle selects the **template image** autotracker will search for, it will appear in the autotracker window. It can be manually resized and moved in relation to the **target**, but the target will stay wherever you have placed it in relation to the **template image** while tracking (should be inside the circle).
 - c. The dashed rectangle is the **search area**, in which autotracker will search for the **template image** in the next frame. Depending on how fast/far per frame the animal is moving, make this as small as possible.
 - d. Press the **search** button for autotracker to automatically check each frame for the image.
 - e. Watch Autotracker for the entire time! If it is no longer following the animal, push the stop button and backtrack to the frame where the **target** lost the animal.
 - f. To create a new template image, delete the current point and **shift + control + click** on the animal again to create a new **key frame**. If you do this without deleting the current point you will just create a new template image, the **target** will not be moved and will place data points incorrectly. Always make sure the target is not placing data points outside the template image.
 - g. You can also try dragging the **search area** box back over the organism and then clicking **search**.
 - h. If tracker is having a hard time, you can manually track by repeatedly using **shift + click** to place the target on the animal frame-by-frame until autotracker can be used again.
 - i. Use the **evolution rate** to tell autotracker how much the template image will change frame-to-frame; usually it can be lower than default. Use the **automark** to set the minimum match score needed to mark points. Lower numbers result in more false-matches but fewer stoppages.
10. Autotracker data:
- a. Adjust the **frame range** in **Clip Settings** to include the entire tracked time, including first and last frames. Select your point mass (usually mass A) from the Plot or Table dropdown menu, and then click the **Refresh data and views** button on the right side of the toolbar.
 - b. Copy and paste the table data into an Excel document.
 - c. Autotracker occasionally skips frames, so it is necessary to check for missing data points. In Excel, select all the data and then use the **Go To** function (Fn+F5 on PC), then click **Special** and check **Blanks**. This highlights all blank cells. You can then change fill color to red to be more visible, and then scroll through to find missing data. Then, go back into the Tracker file and find the skipped frame, **shift-click** to manually mark the position of the organism and copy the needed data back into Excel.

11. Save the tracker file and the Excel file of your data.
12. In Tracker, use a calibration tape (**Show, hide, or create calibration tools** → **new** → **calibration tape**) to measure the length of the back of the chamber for q_{bs} or q_{bt} (See list of model variables below). These values are the same throughout a certain placement and orientation of the cameras and chamber, so only need to be measured once for each block of recording (i.e. if the cameras or chamber get moved/readjusted, then measure these values again).

3.2 3D Data Analysis Protocol

1. Compile the side and top data for each individual into a single Excel file with the columns: time (s), x_s , z_s , x_t , y_t (time can be converted to seconds from frame number, x_s and z_s are the x and y data from the side view while x_t and y_t are the x and y data from the top view, respectively). Ensure that your data is oriented to an origin (0,0,0) at the top, left, front corner of the chamber (particularly your top view data).
2. Open Matlab and run the *Behavior3DCorrection* code (below). Enter your Excel file name, sheet name, data range, q_{bs} and q_{bt} for the video you are analyzing. The code usually takes several minutes (5 - 10) to run for two minutes of data. This code outputs your corrected time, x, y, z position data in a new .xls or .csv file and automatically calls *Behavior3DAnalysis* to calculate metrics of movement.
3. *Behavior3DAnalysis* calculates a number of 3D movement metrics and prints the summary results of this data to a results text file as well as full data to individual .xls or .csv files. It also produces several preliminary plots of the individual organism's movement to aid in visualization and further analysis.

3.3 List of Model Variables

List of Model Variables

q_{bs}	length of back plane of chamber in side video image
q_{bt}	length of back plane of chamber in top video image
q_{is}	hypotenuse formed between front plane and side view image plane
q_{it}	hypotenuse formed between front plane and top view image plane
Q	length of cubic chamber side
Q_s	length of image plane of organism in side view
Q_t	length of image plane of organism in top view
x	true x position of organism
x_{is}	x position in side view image plane
x_{it}	x position in top view image plane
x_s	measured x position in side view tracking
x_t	measured x position in top view tracking
y	true y position of organism
y_{it}	y position in organism's top view image plane
y_t	measured y position in top view tracking
z	true z position of organism

z_{is} z position in side view image plane
 z_s measured z position in side view tracking

3.4 3D Behavior Data Correction MATLAB code

```
function Behavior3DCorrection
%{Behavior3DCorrection is designed to merge and correct x,z (side view) and
%x,y (top view) tracking data derived from video of an organism (a squid
%paralarvae) swimming in a cubic arena taken with two perpendicular cameras.
%This code assumes the data is stored in a single Excel spreadsheet where
%the columns are: time (s), xs, zs, xt, yt. These known data are input into
%an overconstrained system of equations and assessed using least sum of
%squares to approximate the true x, y, and z values with the most minimal
%error. This code accompanies the publication: Zakroff, C. Mooney, TA,
%Wirth, C. Ocean Acidification Responses in Paralarval Squid Swimming
%Behavior Using a Novel 3D Tracking System. (2017). Hydrobiologia Vol: Pages.
%DOI:
%
%Version 1.5 written by Casey Zakroff (czakroff@whoi.edu) May 11 2017
%in MATLAB version 2016b on Mac. Code and protocols available at:
%https://github.com/czakroff/3D-Swimming-Behavior
%}

%User Input of Excel Data Sheet
filename = input('\nEnter the name of the Excel file: ', 's');
sheet = input('Enter the name of the Excel sheet: ', 's');
dataRange = input('Enter the range of the data cells, i.e. "A5:E3602":',
's'); %This data range works for 2 minute videos (3598 frames)
pLcode = strcat(filename, '_', sheet); %ID code for each individual squid
paralarvae (pL)

%Read in t, xs, zs, xt, yt
rawData = xlsread(filename, sheet, dataRange);

%Establish known values of from your arena
Q = 9.6; %Q is the length of the side of your cubic arena (in cm)
%Read in Qbs - the length of the back of the arena in the side video image
Qbs = input('Enter Qbs value: ');
%Read in Qbt - the length of the back of the arena in the top video image
Qbt = input('Enter Qbt value: ');

%Calculate the hypoteneuse of the right triangle between Q and Q back for
both the side and top.
qbs = (Q-Qbs)/2*sqrt(2);
qbt = (Q-Qbt)/2*sqrt(2);

%Set up output data array and assign time to first column
correctedData = zeros(size(rawData,1),4);
correctedData(:,1) = rawData(:,1);

%Assign xs, zs, xt, and yt
xs = rawData(1:size(rawData,1),2);
zs = rawData(1:size(rawData,1),3);
xt = rawData(1:size(rawData,1),4);
```

This is a post-peer-review, pre-copyedit version of an article published in *Hydrobiologia*. The final authenticated version is available online at: <http://dx.doi.org/10.1007/s10750-017-3342-9>

```

yt = rawData(1:size(rawData,1),5);

%Assign limits and options for fmincon
A = ones(1,11)*0.001;
B = ones(1,11)*Q;
options = optimoptions('fmincon','Display','final');

for i = 1:size(rawData,1)
    %Assign guesses for fmincon
    % x0 = [x, y, z, xis, zis, xit, yit, qis, qit, Qs, Qt]
    x0 = [xs(i), yt(i), zs(i), xs(i), zs(i), xt(i), yt(i), 0, 0, Q, Q];

    %Run least sum of squares using fmincon
    [p,fval,exitflag,output] =
fmincon(@leastSumSqr,x0,[],[],[],[],A,B,[],options);
    correctedData(i,2:4) = p(1:3); %store corrected t,x,y,z data
end

%Output corrected data. Creates CSV on Macs
xlswrite(pLcode, correctedData);

%Call analysis code to get 3D metrics and basic plots
Behavior3DAnalysis(correctedData,pLcode);

%System of equations assessed by fmincon
function fVal = leastSumSqr(x0)

    F = zeros(1,12);

    F(1) = x0(1)*x0(10)/x0(4)-Q;
    F(2) = x0(3)*x0(10)/x0(5)-Q;
    F(3) = x0(1)*x0(11)/x0(6)-Q;
    F(4) = x0(2)*x0(11)/x0(7)-Q;
    F(5) = (x0(4)+x0(8)/sqrt(2))-xs(i);
    F(6) = (x0(5)+x0(8)/sqrt(2))-zs(i);
    F(7) = (x0(6)+x0(9)/sqrt(2))-xt(i);
    F(8) = (x0(7)+x0(9)/sqrt(2))-yt(i);
    F(9) = (x0(10)+2*x0(8)/sqrt(2))-Q;
    F(10) = (x0(11)+2*x0(9)/sqrt(2))-Q;
    F(11) = x0(8)/x0(2)-qbs/Q;
    F(12) = x0(9)/x0(3)-qbt/Q;

    fVal = sum(F.^2);
end

end

```

3.5 3D Behavior Analysis MATLAB code

```

function Behavior3DAnalysis(data, filename)
%{Behavior3DAnalysis takes the corrected x, y, z positional data from
%Behavior3DCorrection and calculates total distance traveled, average,
%peak, min, max, and median of 3D, horizontal, and vertical velocity, 3D
%volume covered, 2D turning angles, and path tortuosity (a ratio of the

```

This is a post-peer-review, pre-copyedit version of an article published in *Hydrobiologia*. The final authenticated version is available online at: <http://dx.doi.org/10.1007/s10750-017-3342-9>

```

%distance transitted between two points and the true distance between those
%points) of an individual swimming organism. These data are output to a
%'results' text file and to an Excel file named with the individual
%organism ID code. A 3D plot of the organism's swimming track, a plot of
%the tortuosity of that path, and a 3D polygon of the volume covered by the
%organism are produced and output as tiff files. A histogram of the
%organism's turning angles over the entire track is output as a png.
%This code accompanies the publication: Zakroff, C. Mooney, TA,
%Wirth, C. Ocean Acidification Responses in Paralarval Squid Swimming
%Behavior Using a Novel 3D Tracking System. (2017). Hydrobiologia. Vol:
Pages.
%DOI:
%
%Version 1.5 written by Casey Zakroff (czakroff@whoi.edu) May 11 2017
%in MATLAB version 2016b on Mac. Code and protocols available at:
%https://github.com/czakroff/3D-Swimming-Behavior
%}

%Set variables to set sampling resolution for tortuosity and turning angle
res = 30; %Sampling resolution (value is in frames; 30 fps = 1 s segments)
numPts = floor(size(data,1)/res); %Number of points sampled from the data

%Create arrays to store 3D metrics
distance = zeros(size(data,1)-1,1);
vel = zeros(size(data,1)-1,1); %3D velocity
vertVel = zeros(size(data,1)-1,1); %vertical velocity
horVel = zeros(size(data,1)-1,1); %horizontal velocity
angles = zeros(numPts-1,1); %turning angles
tort = zeros(size(data,1)-res-1,1); %tortuosity

%Calculate total distance traveled (cm) and velocities (cm/s)
%by looping through all data and recording values between successive
%positions.
for i = 1:(size(data,1)-1)
    tDiff = data(i+1,1)-data(i,1); %difference in time (s)
    xDiff = data(i+1,2)-data(i,2); %difference in x (cm)
    yDiff = data(i+1,3)-data(i,3); %difference in y (cm)
    zDiff = data(i+1,4)-data(i,4); %difference in z (cm)
    distance(i) = sqrt(xDiff^2+yDiff^2+zDiff^2);
    vel(i) = distance(i)/tDiff;
    vertVel(i) = zDiff/tDiff;
    horVel(i) = sqrt(xDiff^2+yDiff^2)/tDiff;
end

%Store distance and speed data
totDist = sum(distance);
avgVel = mean(vel);
peakVel = max(vel);
minVel = min(vel);
medVel = median(vel);
avgVVel = mean(vertVel);
peakVVel = max(vertVel);
minVVel = min(vertVel);
medVVel = median(vertVel);
avgHVel = mean(horVel);
peakHVel = max(horVel);
minHVel = min(horVel);

```

This is a post-peer-review, pre-copyedit version of an article published in *Hydrobiologia*. The final authenticated version is available online at: <http://dx.doi.org/10.1007/s10750-017-3342-9>

```

medHVel = median(horVel);

%Calculate hull (for polygon) and 3D volume covered
[K,vol3D] = convhull(data(:,2),data(:,3),data(:,4));

%Calculate 2D turning angles between successive 1-second steps (res)
j = 1;
for i = 1:(numPts-1)
    xdiffl = data((j+res),2)-data(j,2);
    ydiffl = data((j+res),3)-data(j,3);
    xdiffr = data((j+2*res),2)-data((j+res),2);
    ydiffr = data((j+2*res),3)-data((j+res),3);
    a = [xdiffl,ydiffl];
    b = [xdiffr,ydiffr];
    angle = atan2(abs(det([a;b])),dot(a,b));
    angles(i) = angle*180/pi;
    j = j+res;
end

%Store turning angle data
avgAngl = mean(angles);
maxAngl = max(angles);
minAngl = min(angles);
medAngl = median(angles);

%Tortuosity
%Loop for calculating tortuosity over 1-second steps (res)
for i = 1:(size(data,1)-res-1)
    xDifft = data((i+res),2)-data(i,2);
    yDifft = data((i+res),3)-data(i,3);
    zDifft = data((i+res),4)-data(i,4);
    distPts = sqrt(xDifft^2+yDifft^2+zDifft^2); %distance between points
    distTort = sum(distance(i:(i+res))); %distance covered by organism
    tort(i) = distTort/distPts;
end

%Store tortuosity data
avgTort = mean(tort);
peakTort = max(tort);
minTort = min(tort);
medTort = median(tort);

figure
fig = gcf;
fig.PaperUnits = 'centimeters';
fig.PaperPositionMode = 'manual';
fig.PaperSize = [33.8 33.8];

%Plot 3D track and output tiff
plot3(data(:,2),data(:,3),data(:,4)-9.6);
ax = gca;
ax.FontUnits = 'centimeters';
ax.FontName = 'Times New Roman';
ax.FontSize = 0.35;
ax.XTickMode = 'manual';
ax.YTickMode = 'manual';

```

```

ax.XLimMode = 'manual';
ax.YLimMode = 'manual';
ax.XLim = [0,9.600];
ax.YLim = [0,9.600];
ax.ZLim = [-9.600,0];
ax.XTick = [0,1,2,3,4,5,6,7,8,9];
ax.XTickLabel = {' ','1',' ','3',' ','5',' ','7',' ','9'};
ax.YTick = [0,1,2,3,4,5,6,7,8,9];
ax.YTickLabel = {' ','1',' ','3',' ','5',' ','7',' ','9'};
ax.ZTick = [-9,-8,-7,-6,-5,-4,-3,-2,-1,0];
ax.ZTickLabel = {'-9',' ','-7',' ','-5',' ','-3',' ','-1',' '};
ax.XGrid = 'on';
ax.YGrid = 'on';
ax.ZGrid = 'on';
ax.XLabel.String = 'Length (cm)';
ax.YLabel.String = 'Length (cm)';
ax.ZLabel.String = 'Depth (cm)';
print('-dtiff','-r600',filename);

%Calculate colormap for tortuosity plot
[t1,t2] = size(tort);
[d1,d2] = size(data(:,2));
col = tort;
for i = t1:d1-1
    col = [col; tort(t1)];
end

%Plot tortuosity as thin 3D surface and output tiff
h = surface([data(:,2), data(:,2)], [data(:,3), data(:,3)],...
    [data(:,4)-9.6, data(:,4)-9.6],[col,col],...
    'FaceColor','none','EdgeColor','interp','LineWidth',2);
colormap(jet);
caxis([1 5]);
c = colorbar;
text('String','Tortuosity','Rotation',90,...
    'Position',[242.25 302.5 -227.2],...
    'FontUnits','centimeters','FontSize',0.35,...
    'FontName','Times New Roman','color','k');
print('-dtiff','-r400',strcat(filename,'_tortuosity'));

%Create 3D volume polygon and output tiff
colormap(parula);
trisurf(K,data(:,2),data(:,3),data(:,4)-9.6);
ax = gca;
ax.FontUnits = 'centimeters';
ax.FontName = 'Times New Roman';
ax.FontSize = 0.35;
ax.XTickMode = 'manual';
ax.YTickMode = 'manual';
ax.XLimMode = 'manual';
ax.YLimMode = 'manual';
ax.XLim = [0,9.600];
ax.YLim = [0,9.600];
ax.ZLim = [-9.600,0];
ax.XTick = [0,1,2,3,4,5,6,7,8,9];
ax.XTickLabel = {' ','1',' ','3',' ','5',' ','7',' ','9'};
ax.YTick = [0,1,2,3,4,5,6,7,8,9];

```

This is a post-peer-review, pre-copyedit version of an article published in *Hydrobiologia*. The final authenticated version is available online at: <http://dx.doi.org/10.1007/s10750-017-3342-9>


```
xlswrite(strcat(filename, '_distance'), distance);  
xlswrite(strcat(filename, '_3Dvelocity'), vel);  
xlswrite(strcat(filename, '_verticalVelocity'), vertVel);  
xlswrite(strcat(filename, '_horizontalVelocity'), horVel);  
xlswrite(strcat(filename, '_angles'), angles);  
xlswrite(strcat(filename, '_tortuosity'), tort);
```

end

4. References

- Clayton TD, Byrne RH (1993) Spectrophotometric seawater pH measurements : total hydrogen results. *Deep Res* 40:2115–2129
- Dickson AG (1990) Standard potential of the reaction: , and and the standard acidity constant of the ion HSO₄⁻ in synthetic sea water from 273.15 to 318.15 K. *J Chem Thermodyn* 22:113–127
- Dickson AG, Sabine CL, Christian JR (2007) Guide to best practices for ocean CO₂ measurements. *PICES Spec Publ* 3:p191
- Mehrbach C, Culberson CH, Hawley JE, Pytkowicz RM (1973) Measurement of the apparent dissociation constants of carbonic acid in seawater at atmospheric pressure. *Limnol Oceanogr* 18:897–907