

Appendix 5A Matlab function for buzz click detection and level estimation

```
function [Result, cumW, cumS] = spw_click_det(x_raw, x_add, thresh, Nfilt, fs, click_duration, min_delay,
C_window)

% Function modified from Paul White, ISVR Southampton (2012)

% Function for detecting sperm whale clicks- based on energy within a frequency band, with dual threshold
% Filtering done using an FIR filter of order 128
% Energy computed using a smoothing filter of 1 ms duration
% Detection is based on a single channel - if the wav file is stereo
%
%
% Inputs:
%     x_raw           : signal pressures
%     x_add           : left-over signal pressures from signal division to 0.5 s time blocks
%     thresh          : Threshold (here, set to 40)
%     Nfilt           : number of taps (256)
%     fs              : sampling rate (Hz)
%     click_duration  : click duration - defines the length of the smoothing filter (1/1000s)
%     min_delay       : minimum delay between end of one click and the start of the next (9/1000 s)
%     C_window        : fixed window used to calculate energy (3.5 ms)

% Outputs:
%     Result          : A structure array including the fields:
%         Result(k).start      - Sample number of click start
%         Result(k).stop       - Sample number of click end
%         Result(k).rms        - rms pressure level SPL re 1?Pa
%         Result(k).rms90      - rms pressure level SPL re 1?Pa over 90% energy window
%         Result(k).sel90      - sound exposure level over 90% energy window
%         Result(k).ppeak      - peak-to-peak level
```

```

%           Result(k).duration      - 90% energy window duration
%           Result(k).addSamples    - number of added samples
%           Result(k).firstStart    - start time of the first detection in the time block
%           Result(k).peak          - sample number at which the maximum energy occurred
%           Result(k).peak_val      - peak energy during the click
%
%           cumW                    : detected signal windows concatenated
%           cumS                    : detected 90% signal windows concatenated

%% Preamble
Tblock=size(x_raw,1)./fs;

% design filters
B_det=firl(Nfilt,[f_low_det f_high]/(fs/2),'high'); % for click detection
[B, A] = butter(3, [f_low/(fs/2), f_high/(fs/2)]); % for ASL estimation

% Force filter order to be even, so number of taps is odd, so delay is an integer
if ( rem(Nfilt,2) ), Nfilt=Nfilt+1; end
delay=Nfilt/2; % Filter's group delay

min_thresh=thresh/2.5; % Threshold for ending of click

N=size(x_raw,1); % Number of samples in the wav file
Nch=size(x_raw,2); % Number of channels in the wav file

Nclick=round(fs*click_duration); % Duration of the smoothing filter in samples
% Forces smoother filter length to be odd, so its delay is an integer
if ( rem(Nclick,2)==0 ), Nclick=Nclick+1; end

Bsmth=hanning(Nclick); % Filter to be used to smoother the energy
Bsmth=Bsmth/sum(Bsmth); % Normalised so gain at dc is unity
smth_delay=(Nclick-1)/2; % Group delay of the smoothing filter

% Converting parameters in time to equivalents in terms of samples

```

```

Nblock=round(Tblock*fs);           % Number of samples in each block
delay_N=round(min_delay*fs);       % Minimum delay in samples

arg=[1 Nblock];                   % Index for each block - 2 element vector, [start_index
stop_index]

count=1;                          % Counter for the clicks

sens = -188;                      % hydrophone sensitivity in dB re 1 microPa^-1
p = x_raw * 10.^(-sens/20);        % pressure in microPa
p=filter(B,A,p);

if ~isempty(x_add)                % x_add: left over signal from signal division to time blocks
p_add = x_add * 10.^(-sens/20);    % convert to pressures
p_add=filter(B,A,p_add);
end

%% Main Loop over the blocks
while (arg(2)<=N)

    x=conv(B_det,x_raw(:,1));      % Band-pass filter time block
    % Trim the convolution, symmetrically and removes the filter delay
    x=x(delay+1:end-delay);
    if (Nch==2)
        x2=conv(B_det,x_raw(:,2)); % Band-pass filter time block
        % Trims the convolution, symmetrically and removes the filter delay
        x2=x2(delay+1:end-delay);
    end

    xsq=x.*x;
    E=conv(Bsmth,xsq);             % Smoothed energy
    E=E(smth_delay+1:end-smth_delay); % Trim symmetrically again to remove delay

    % Statistics of the energy measure

```

```

medE=median(E); % Robust estimate of the mean
rob_std=sqrt(median((E-medE).^2)); % Robust estimate of the spread

T_start=medE+thresh*rob_std; % Compute the upper threshold
T_stop=medE+min_thresh*rob_std; % Compute the lower threshold

% Clicks starts when energy exceeds the upper threshold and stops when
% energy falls below the lower threshold

above1=E>T_start; % Find the points above the upper th
above2=E>T_stop; % Find the points above the lower th

% Find the points crossing the upper threshold, from low to high
start_times=find(~above1(1:end-1) & above1(2:end) );
% Find the points crossing the lower threshold, from high to low
stop_times =find( above2(1:end-1) & ~above2(2:end));

Result=[];
cumW=[]; % sum of squared pressures of signal windows over entire block
cumS=[]; % sum of squared pressures of 90% energy windows over entire block

% Don't do any of the following if the thresholds aren't exceeded
if ( ~isempty(start_times) && ~isempty(stop_times) )

    % First crossing of the upper threshold - start of first click
    start_current=start_times(1);

    while (1)

        % Find next crossing of the lower threshold - end of the click
        ind=find(stop_times>start_current,1,'first');
        if (isempty(ind))
            break % Stop if there is none
        end
        stop_current=stop_times(ind);

```



```

% Store the data for that click
Result(count).start=start_current+arg(1)-1;
Result(count).stop=stop_current+arg(1)-1;

a1 = start_current;
a2 = start_current + C_window/1000*fs - 1;
addSamples = max(0,a2-size(x_raw,1));
a2 = min(a2,size(x_raw,1));

clickWindow = p(a1:a2,1);
if ~isempty(x_add)
clickWindow = [clickWindow; p_add(1:addSamples,1)];
end

E_a = cumsum((clickWindow).^2);
E_a = E_a./max(E_a);
s1 = find(E_a>=0.05, 1);
s2 = find(E_a>=0.95, 1);

clickSignal = clickWindow(s1:s2,1);

Result(count).rms = 10.*log10(mean(clickWindow.^2));
Result(count).rms90 = 10.*log10(mean(clickSignal.^2));
Result(count).sel90 = 10.*log10(sum(clickSignal.^2));

Result(count).ppeak = 20.*log10(max(clickWindow)+abs(min(clickWindow)));

Result(count).duration = size(clickSignal,1)./fs;
Result(count).addSamples = addSamples; % number of samples added
% start time of the first detection in the time block
Result(count).firstStart = (count==1)*(start_current+arg(1)-1);

cumW = [cumW; clickWindow];
cumS = [cumS; clickSignal];

```

```

        % Find the peak energy
        [val,ind]=max(E(start_current:stop_current));
        Result(count).peak=arg(1)+start_current-1+ind;
        Result(count).peak_val=val;

        % Update counter
        count=count+1;

        % Find next start of a click
        ind=find(start_times>stop_current+delay_N,1,'first');
        if (isempty(ind)), break, end

        start_current=start_times(ind);
    end

end
% Increment to read next block
arg=arg+Nblock;
end
end

```

Appendix 5B Buzz data and filtering

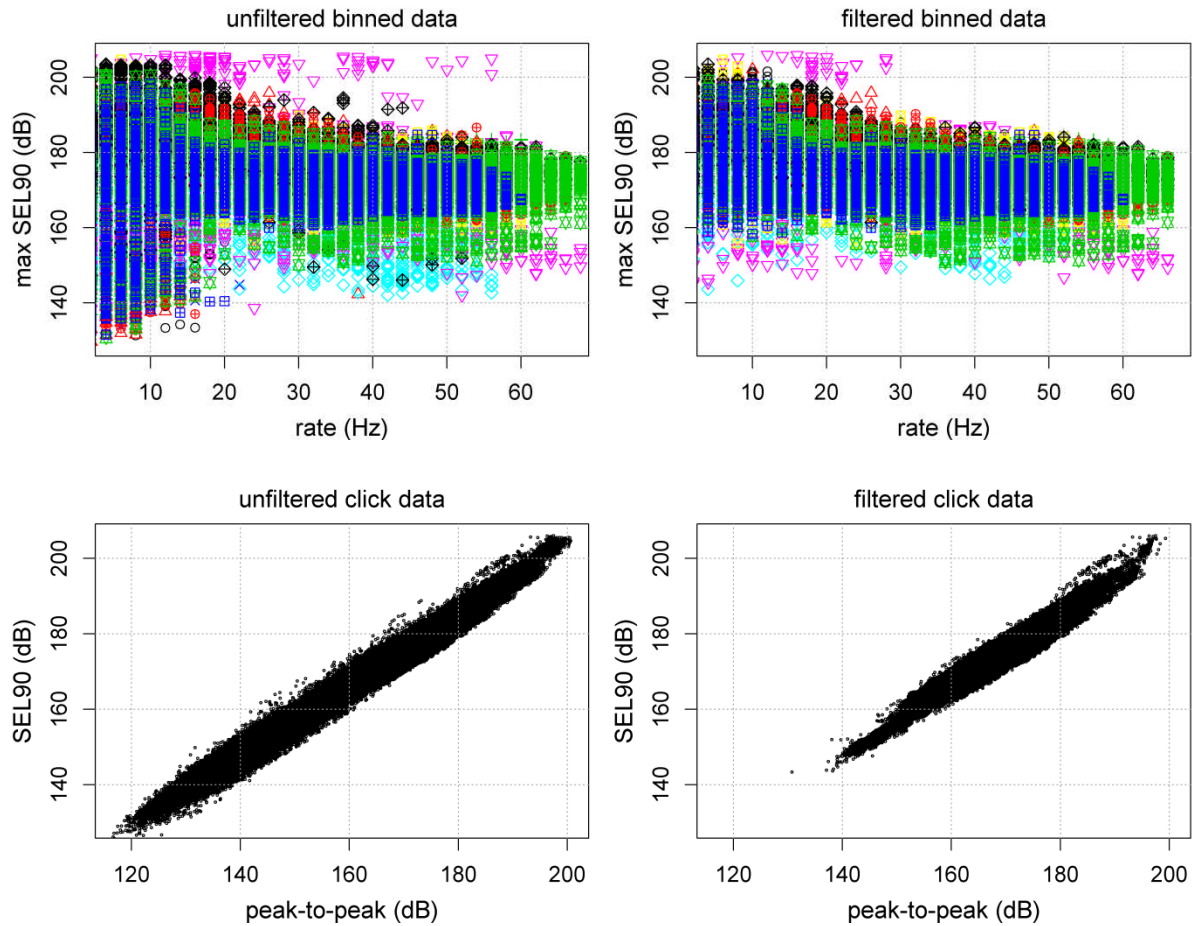


Figure 5B.1. Filtering of binned and click-by-click data.

Top: unfiltered and filtered binned data for maximum SEL 90 (dB) as a function of number of detected clicks per second. Colour-symbol combinations show data from different tags. Bottom: unfiltered and filtered click-by-click data for SEL 90 (dB) as a function of peak-to-peak values (dB). All acoustic metrics of apparent click output were strongly linearly correlated: R-squared values were 0.91 ($SPL_{rms} \sim SPL_{90}$), 0.87 ($SPL_{rms} \sim SPL_{pp}$) and 0.91 ($SPL_{90} \sim SEL_{90}$) when fitting the respective variables as Gaussian linear models to all fine-filtered click data.

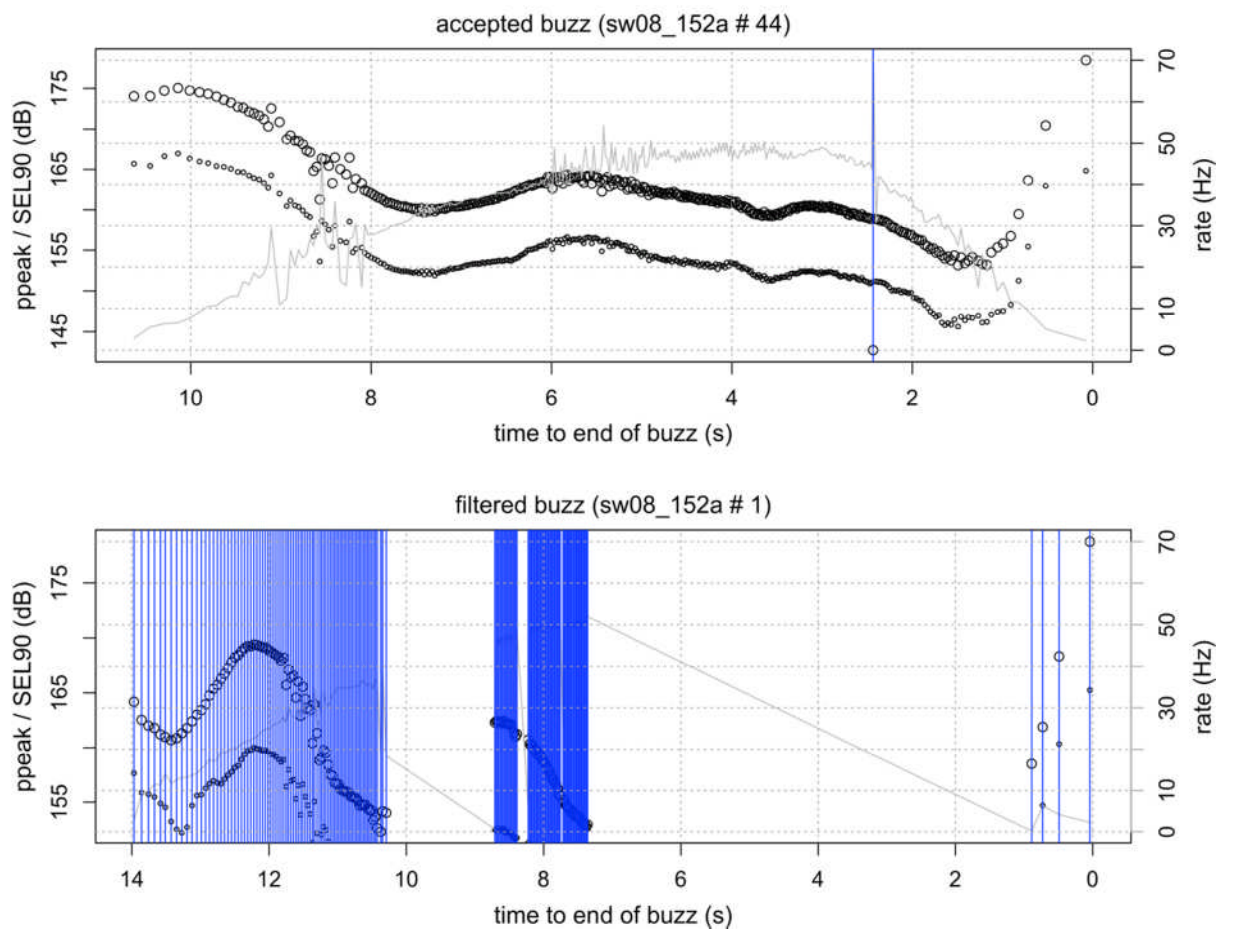


Figure 5B.2 a

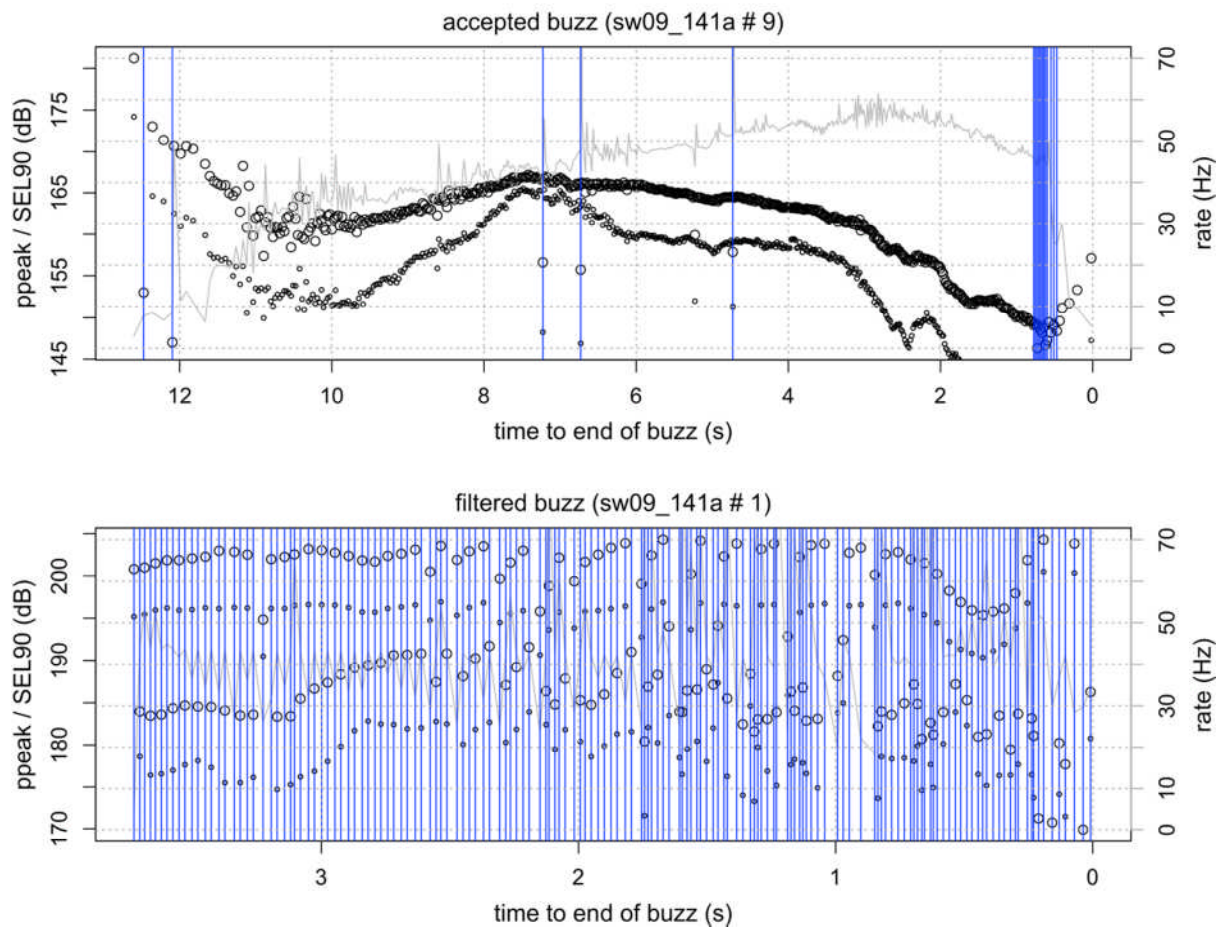


Figure 5B.2. b

Figure 5B.2. Example filtering of clicks

Top: buzz accepted as high-quality buzz that passed “fine-filtering”. Bottom: a fine-filtered buzz. Vertical blue lines: timing of individually filtered clicks, large circles: SEL90 (dB), small circles: peak-to-peak values (dB), grey line: click rate (1/ICI, s).

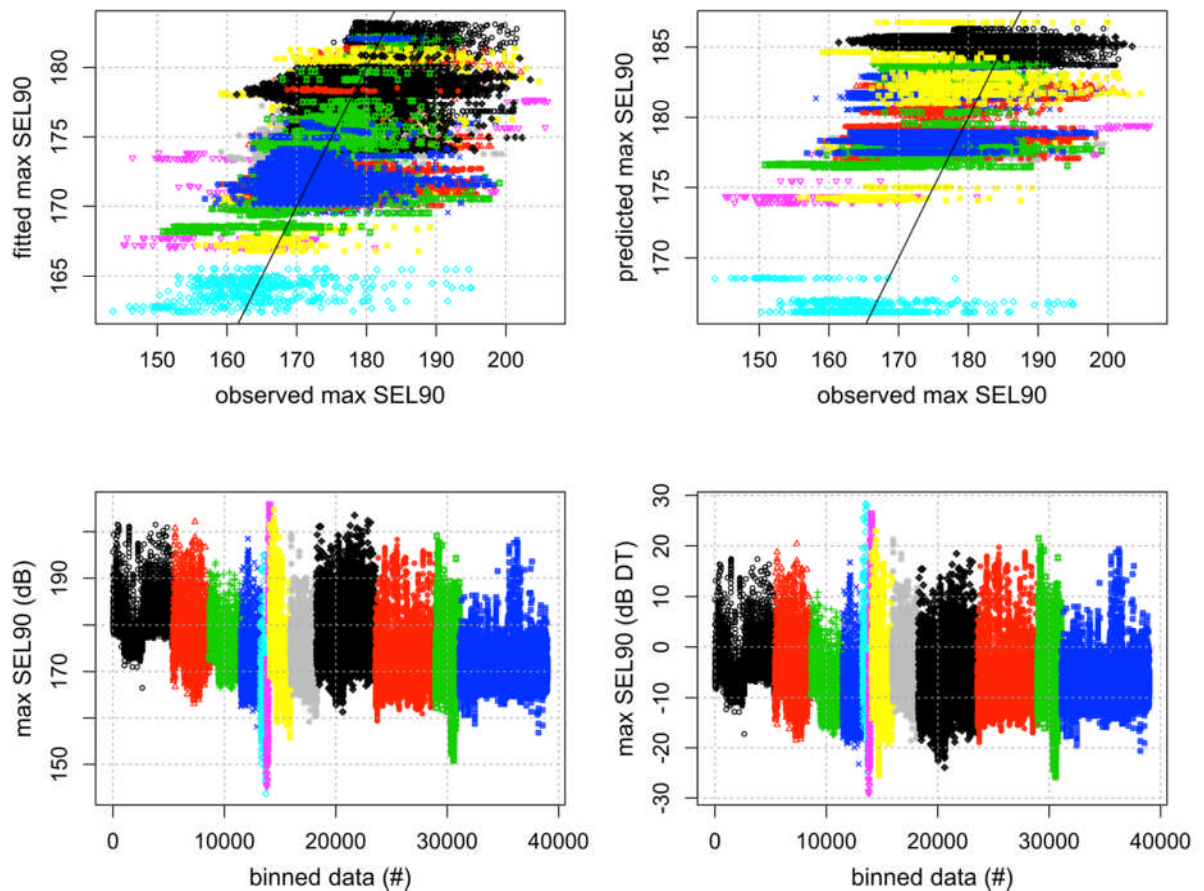


Figure 5B.3. De-trending of sound exposure levels

Generalised additive models (GAM) were used to smooth binned maximum SEL (dB) as a function of tag-on-time (interaction within each whale) and depth (main effect across whales). Top left shows fitted vs. observed values from this model. The model was then used to predict SEL for each time since tag-on with 20 metres as a reference depth. Top right shows the predicted vs. observed values. The detrended values were obtained by subtracting observed values from the predicted. Bottom left: raw values and bottom right: detrended values, shown in order of tag id. Colour-symbol combinations show data from different tags. Only fine-filtered data that were used in model fitting and prediction are shown. De-trending SEL₉₀ levels shifted the distribution of click-level SEL₉₀ from a median of 172 dB [152.0, 184.1] to a median of -2.6 dB [-15.8, 0].

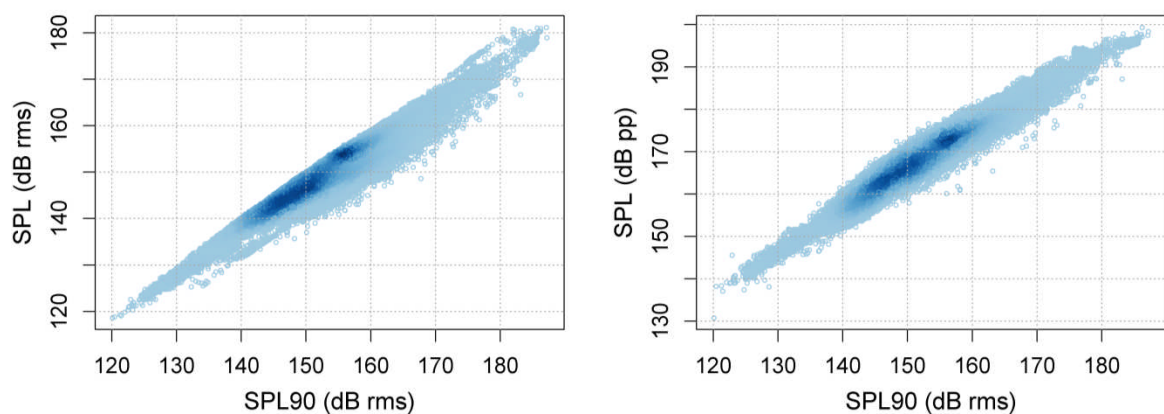


Figure 5B.4 SPL_{rms} and SEL_{90} as a function of SPL_{90} in fine-filtered click data

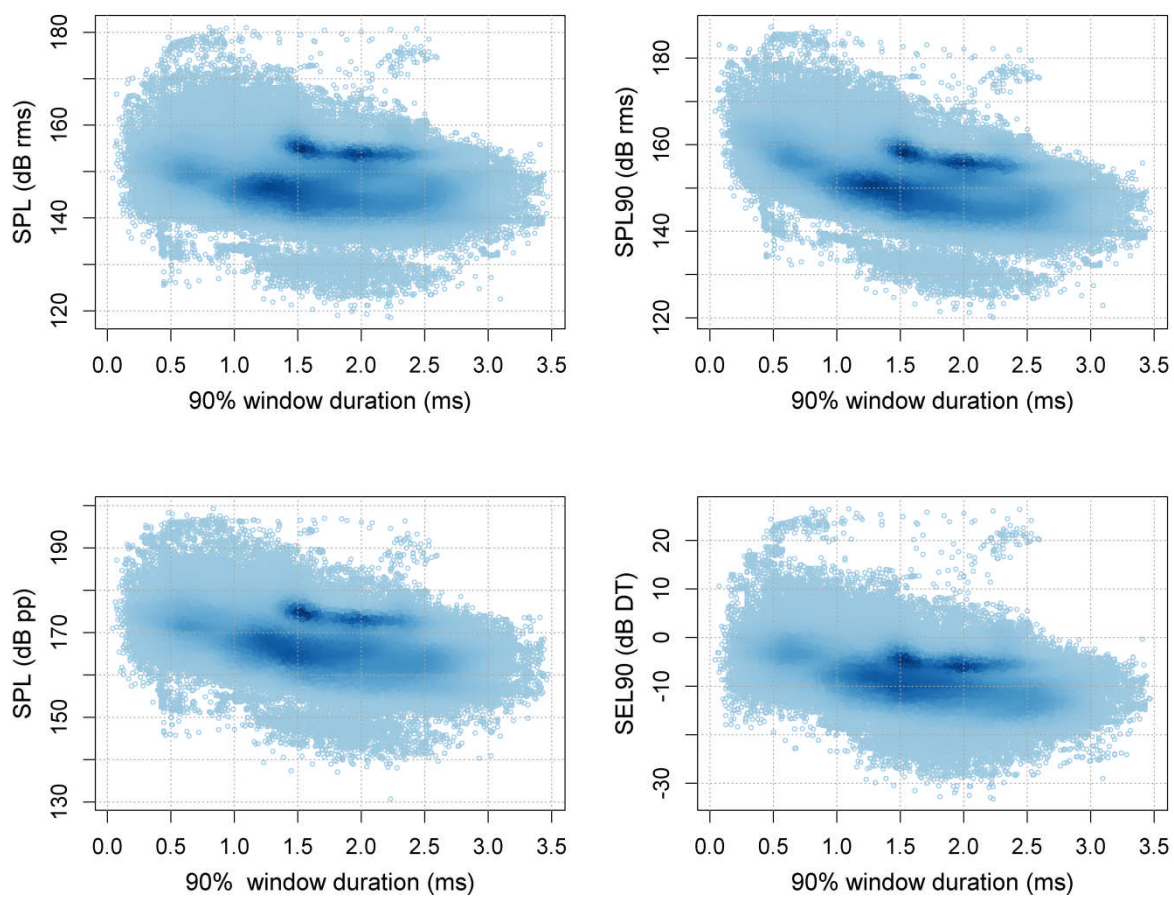


Figure 5B.5 Sound metrics as a function of 90% energy window duration in fine-filtered click data

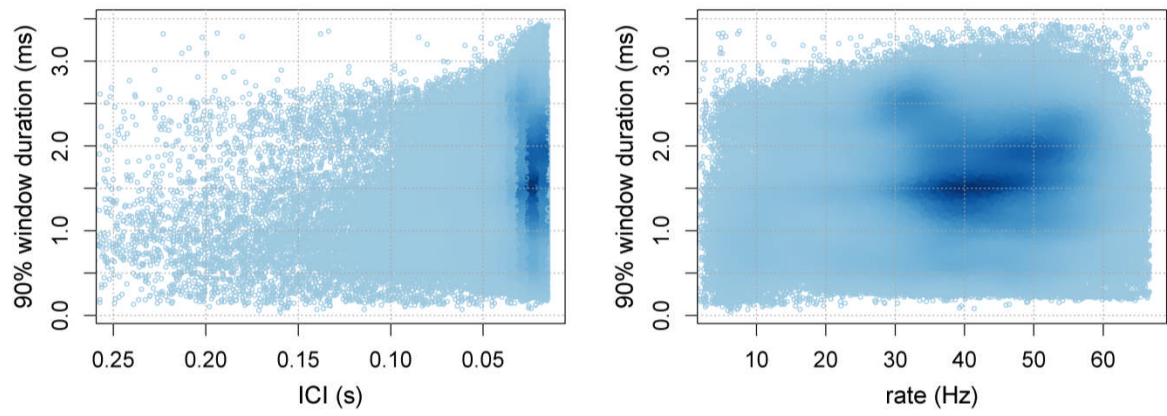


Figure 5B.6 ICI and rate ($1/ICI$) (s) as a function 90% energy window duration in fine-filtered click data

Appendix 5C ICI and level adjustments

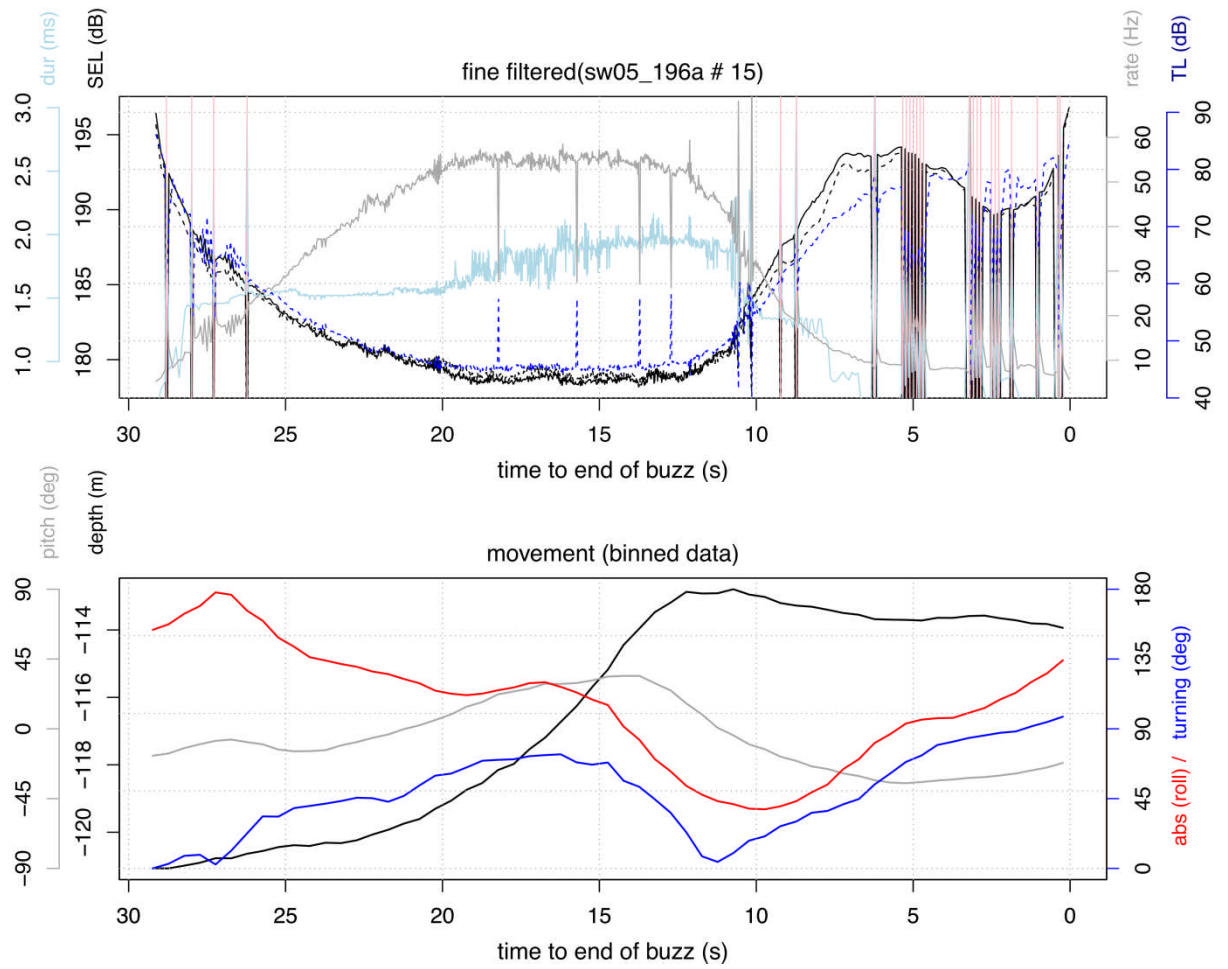


Figure 5C.1. a

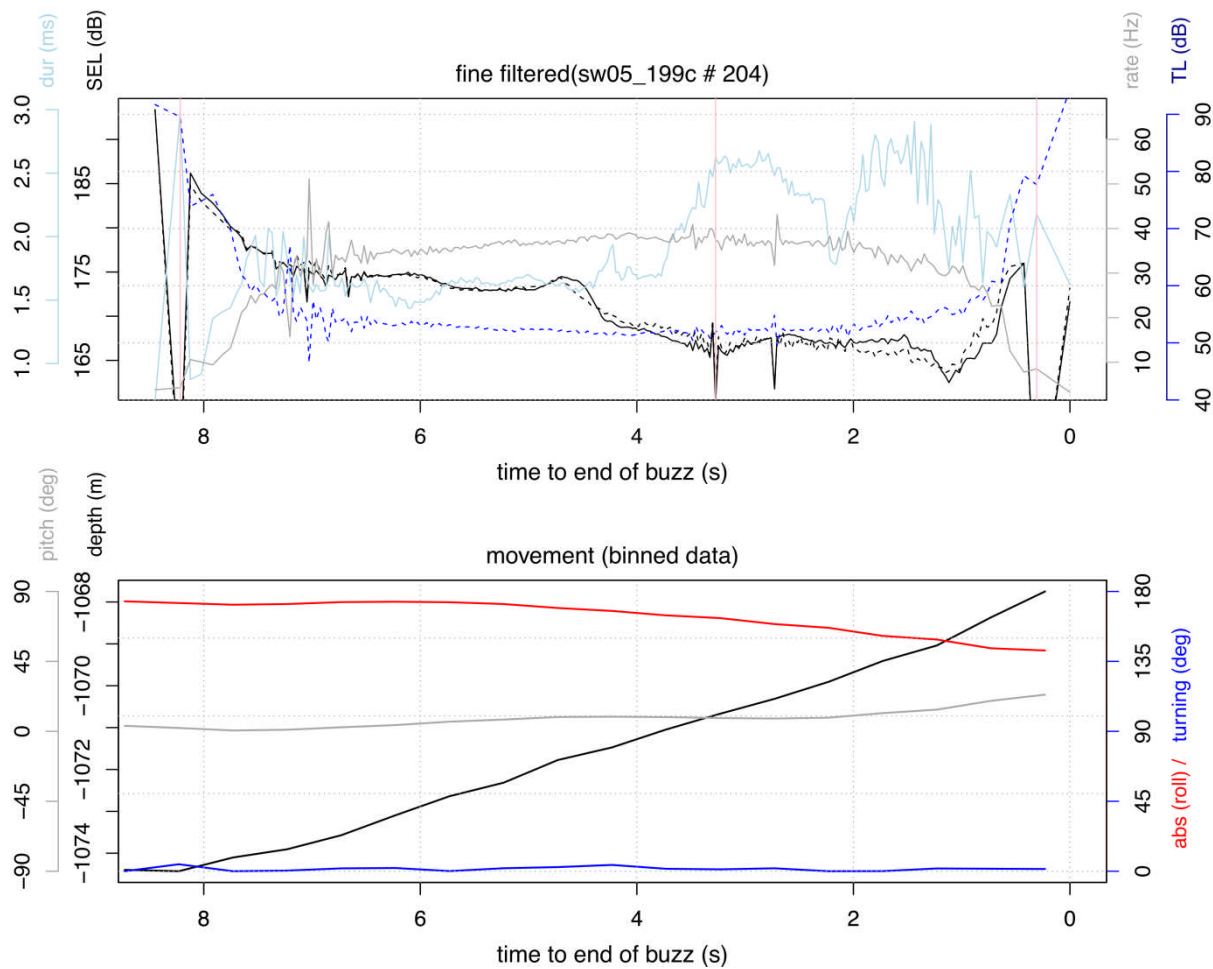


Figure 5C.1. b

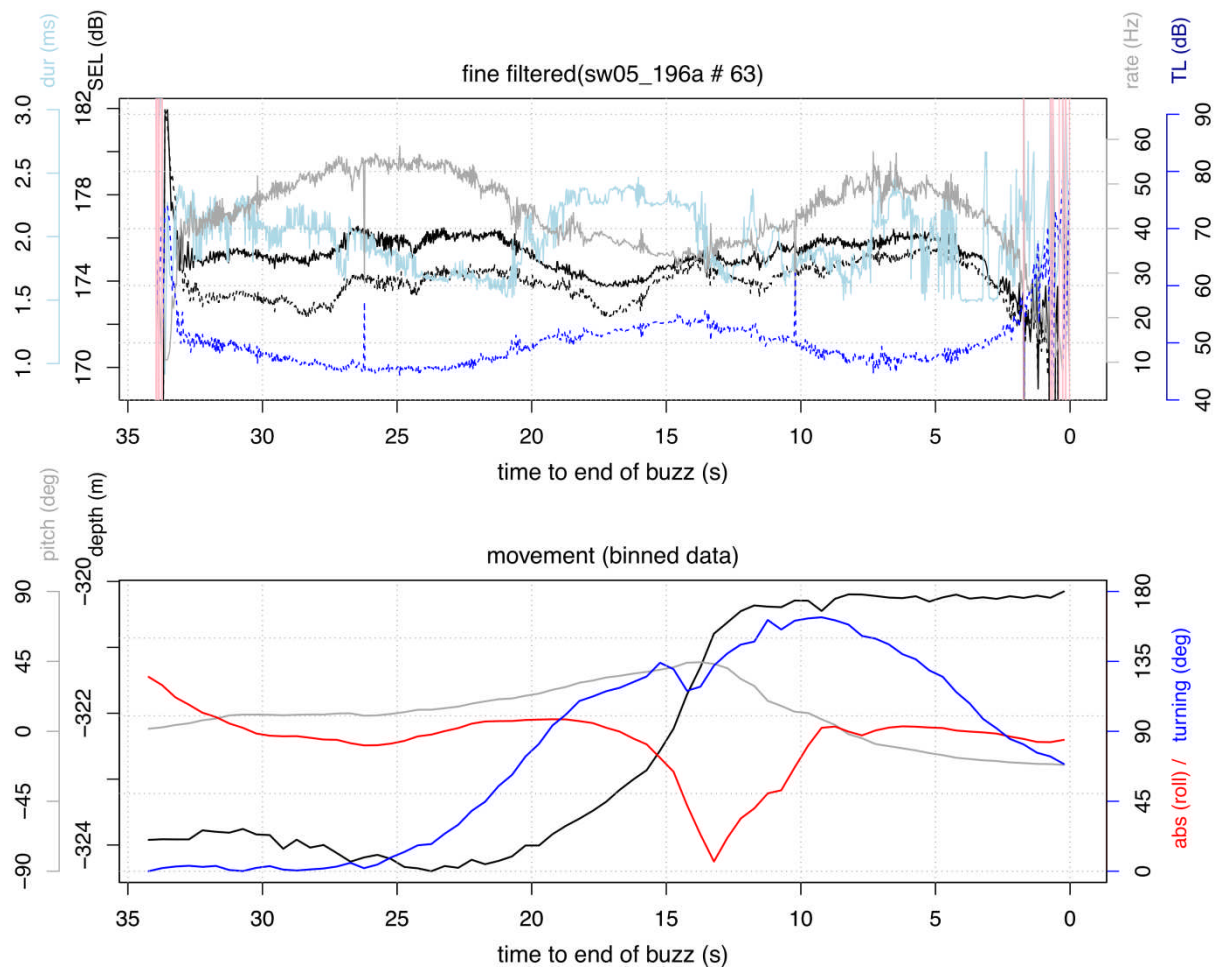


Figure 5C.1. c

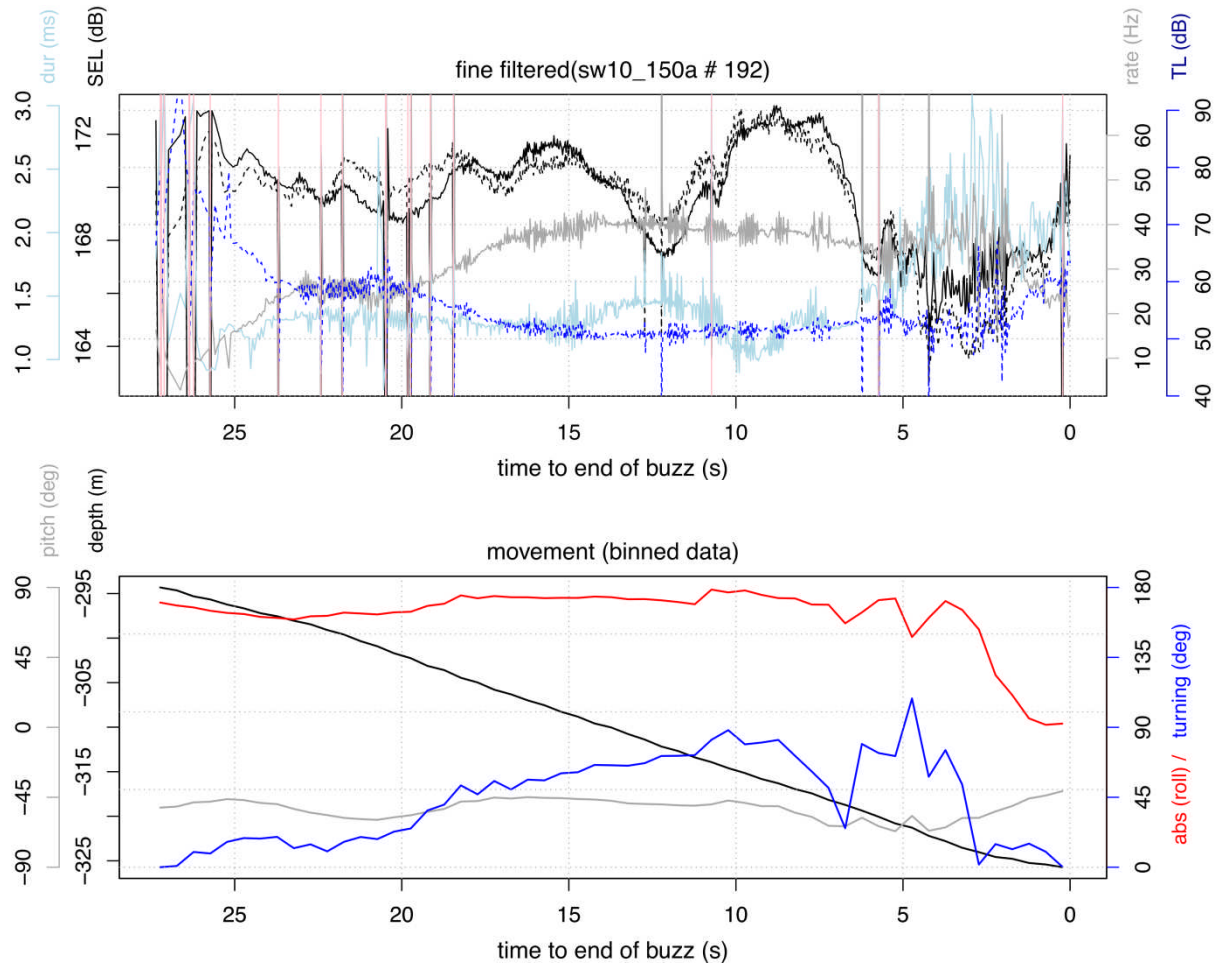


Figure 5C.1. d

Figure 5C.1 a-d Example co-variation between acoustic and movement metrics.

Top: acoustic metrics from click data. Light blue: duration of 90% energy window (ms). Black: click level (SEL_{90} : solid line, SPL_{pp} : dashed line), grey: click rate ($1/ICI$ s), dark blue: expected transmission loss TL (assuming $ICI = \text{two-way travel time}$, $40 \log_{10}(\frac{ICI}{2} * 1490 \frac{m}{s})$). Vertical pink lines: fine-filtered clicks. Bottom: movement metrics for the same buzz from binned data. Pitch, depth, absolute value of roll and turning angle are shown in degrees.

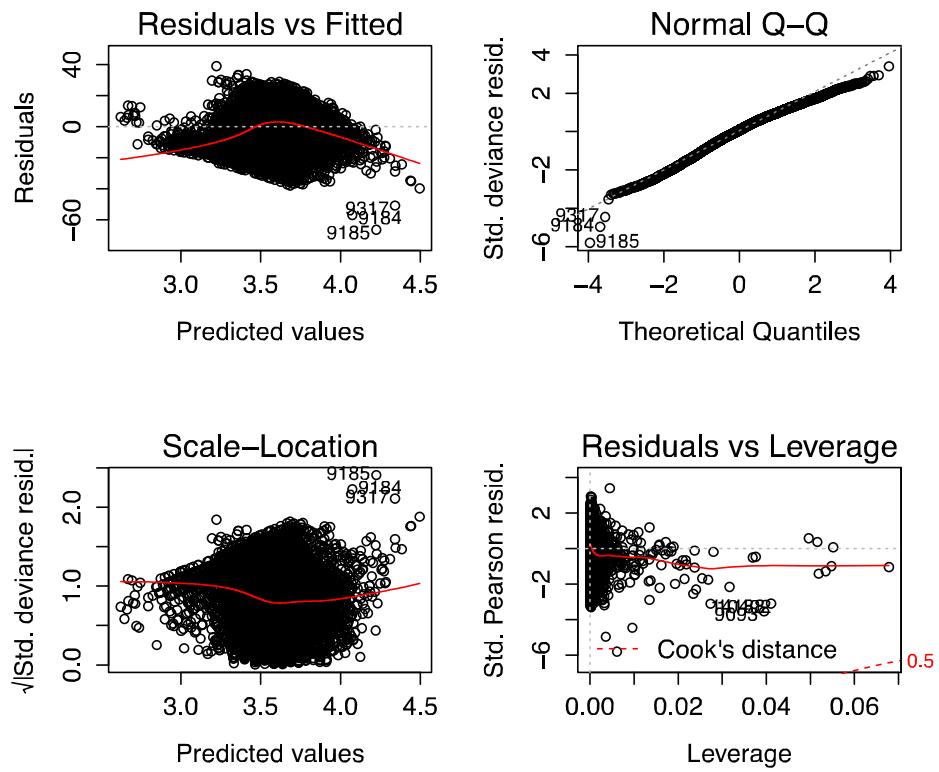


Figure 5C.2 Diagnostic plots for the GLM Gaussian model for binned rate

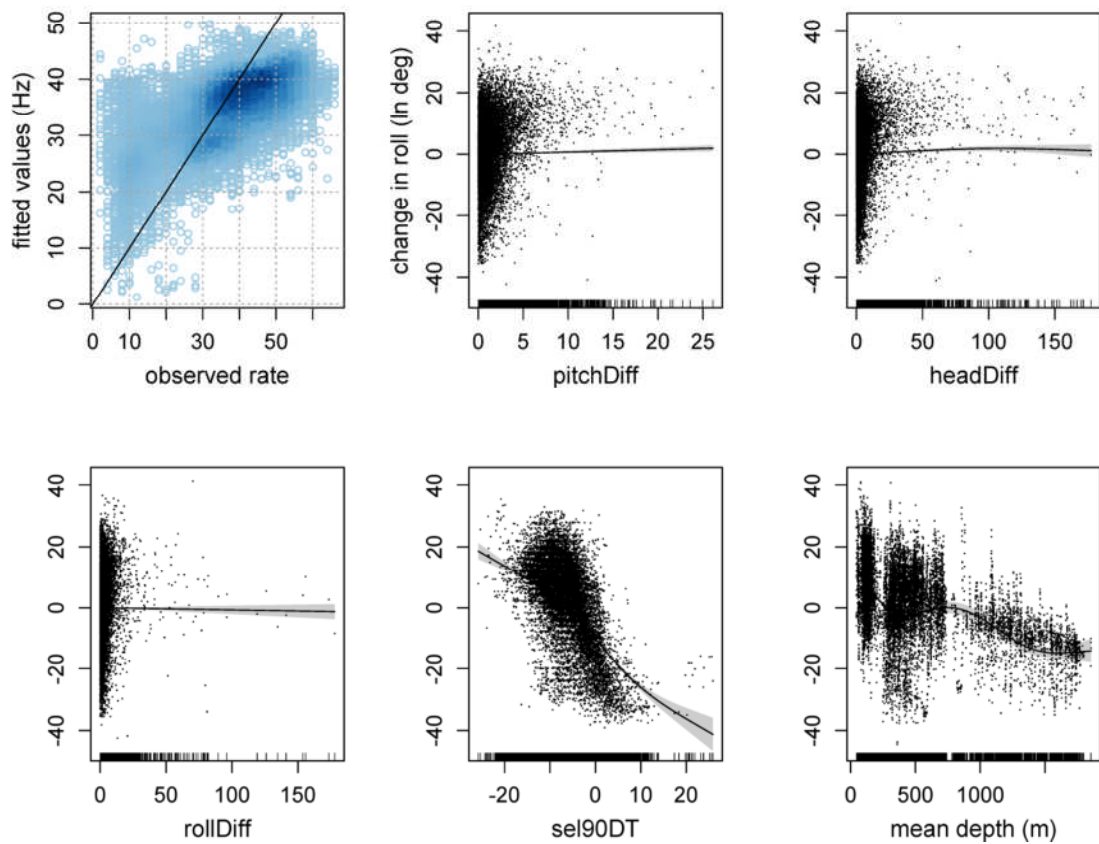


Figure 5C.3. GAMM models for binned rate

Top left shows kernel density (darker: denser) of observed vs. fitted values, with black line showing one-to-one relationship as a reference. Remaining plots show component smooths for each explanatory variable; scatter shows residuals, and shaded area \pm SE. Binned rate was assumed normal distribution, and the complexity of each explanatory smooth was limited by setting maximum number of knots to five. Buzz id was included as a random effect and autocorrelation was modelled using first-order autoregression (AR1).

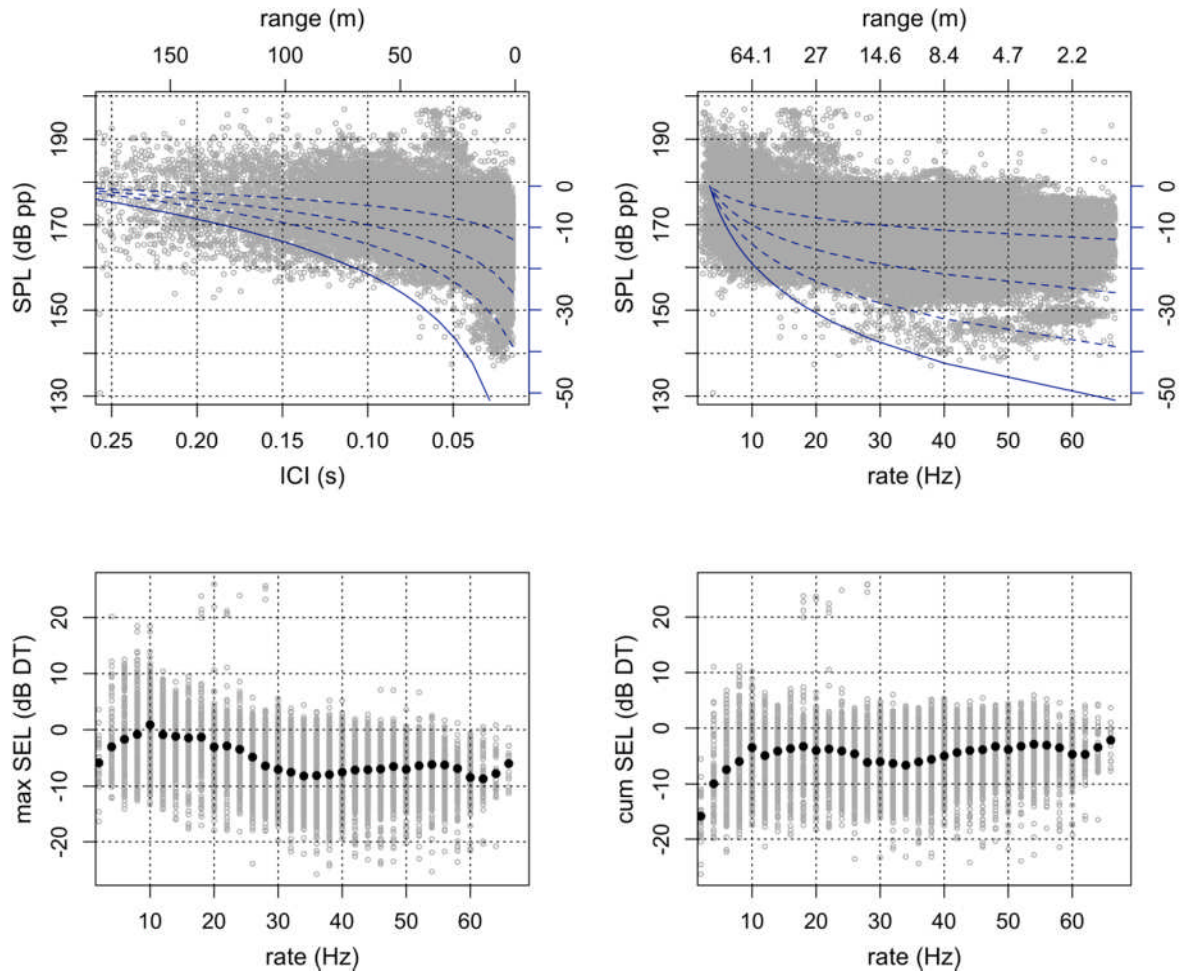


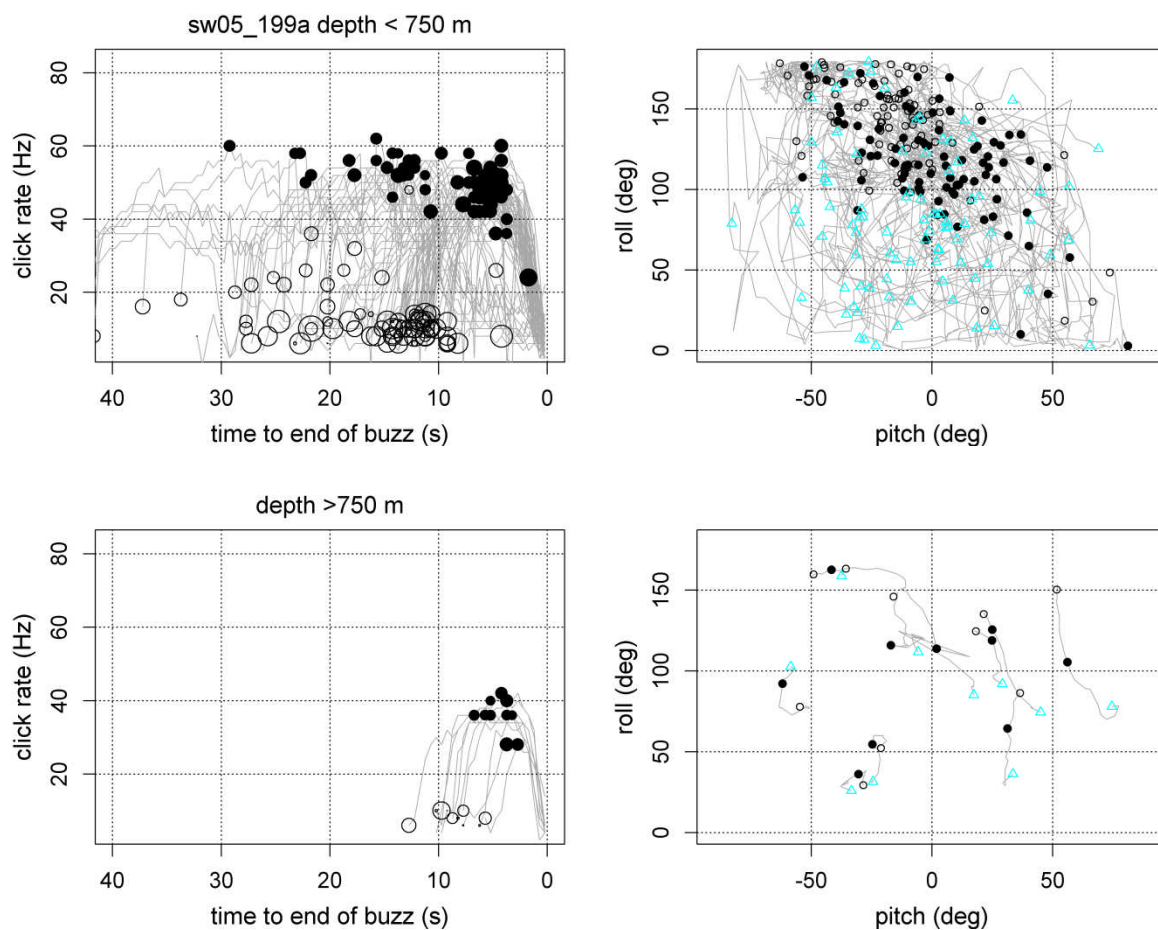
Figure 5C.4. Apparent output as a function of ICI pooling across approaches

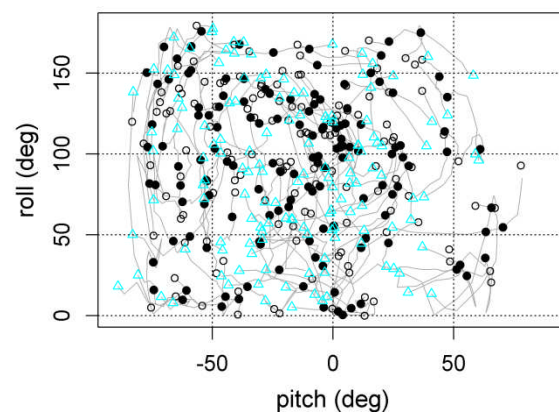
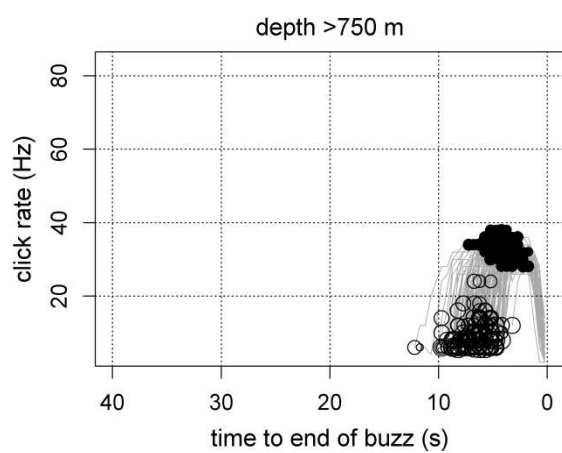
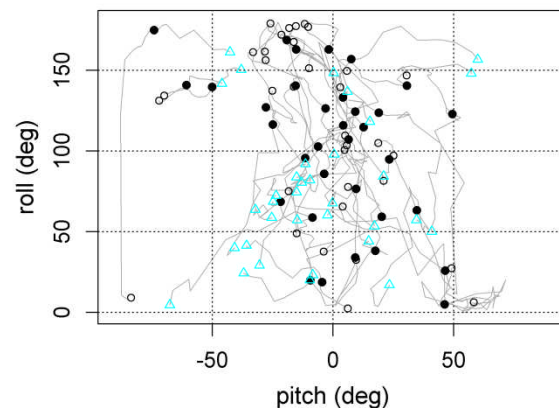
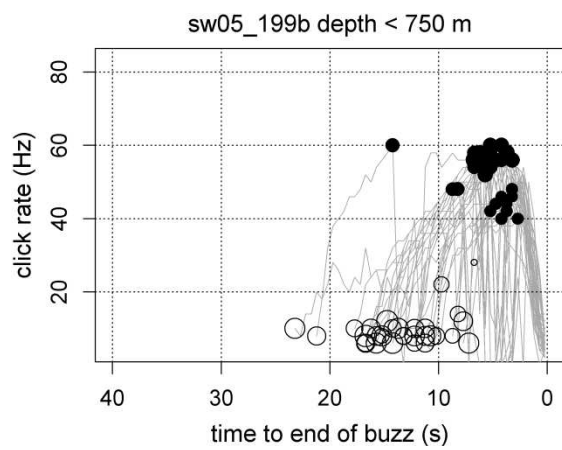
Top: click peak-to-peak levels as a function of ICI (s), and $1/\text{ICI}$ (Hz). Top axis shows respective ranges if ICI equalled two-way travel time (TWTT, for sound speed 1485 m/s). Fine-filtered “approach” clicks are shown (Section 5.3.4; $n = 22\,000$ clicks, 910 buzzes). Solid dark blue lines show expected transmission $40 \cdot \log(\text{range})$, given an arbitrary initial level at ICI of 0.3 s. Dashed lines show $30 \cdot \log_{10}$, $20 \cdot \log_{10}$ and $10 \cdot \log_{10}$ curves. Bottom: maximum and cumulative SEL (dB, detrended) as a function of detection rate in binned data (fine-filtered, $n = 168$ min).

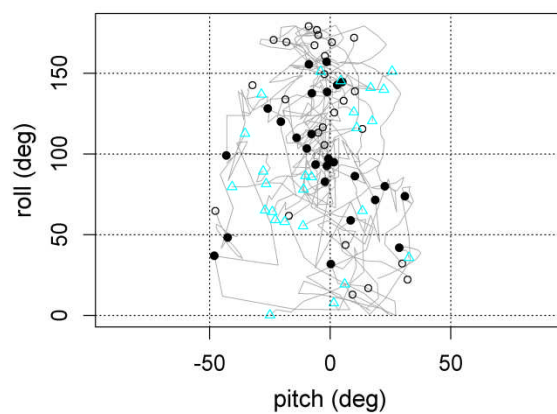
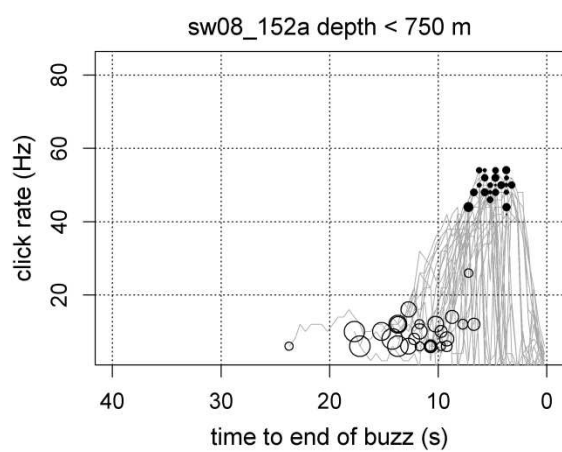
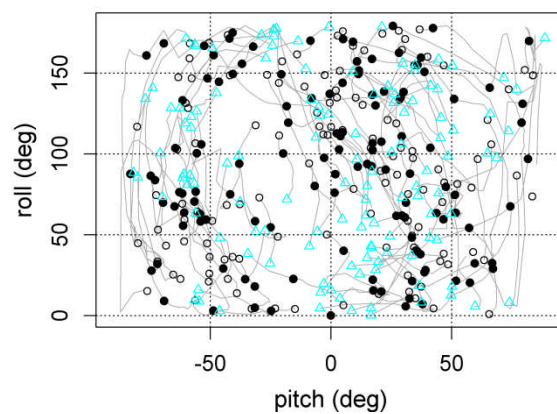
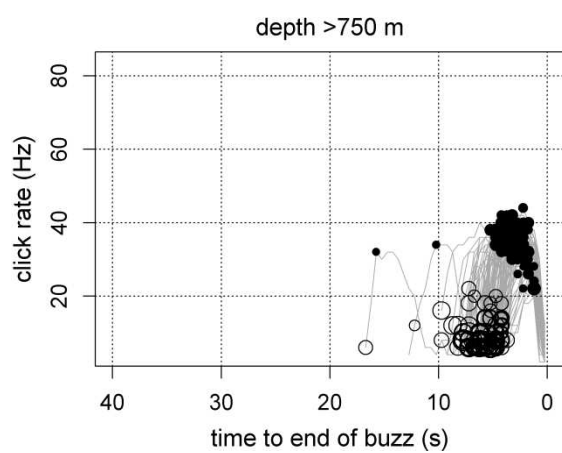
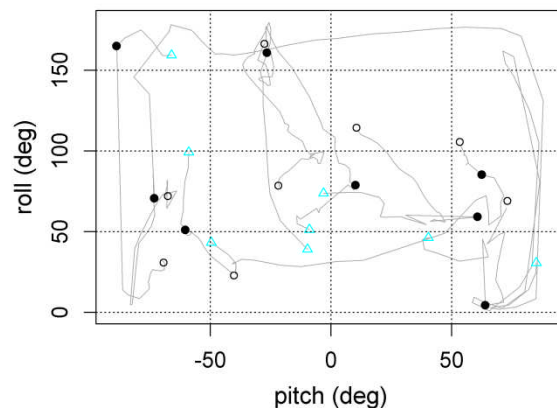
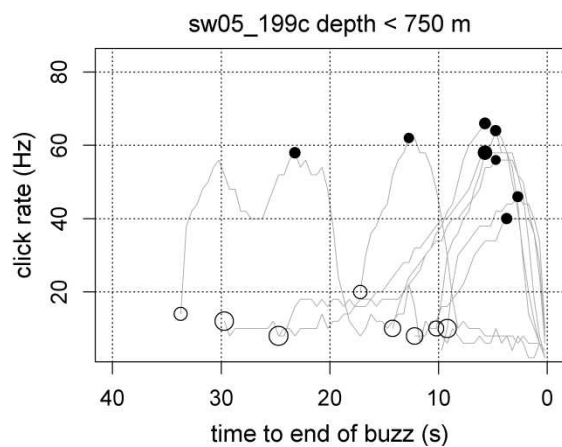
Appendix 5D Whale-by-whale time-series

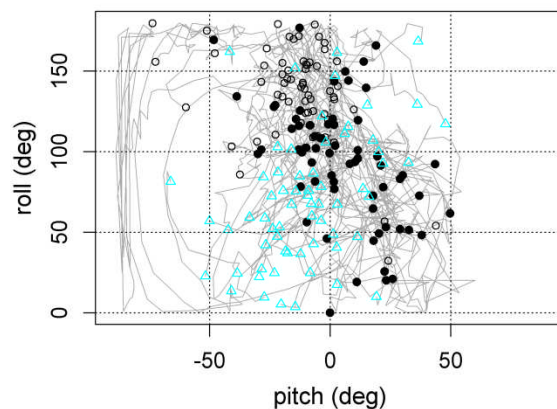
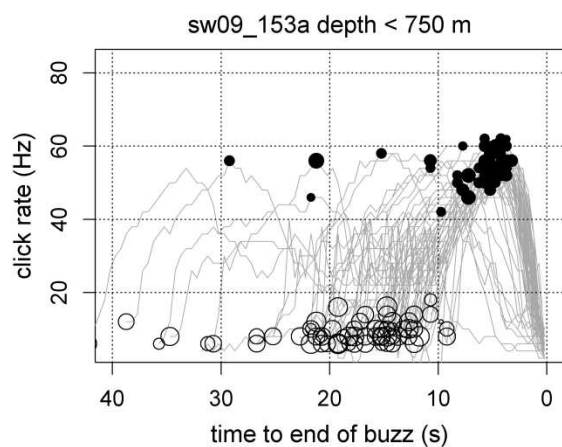
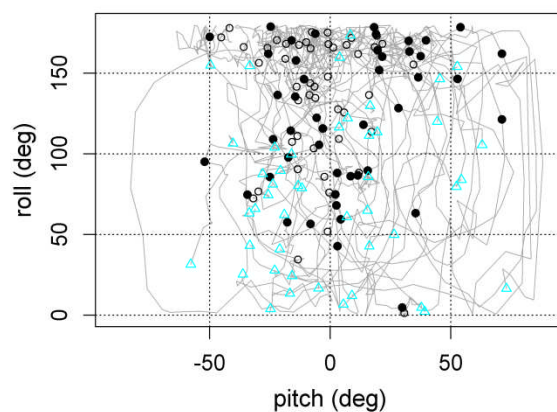
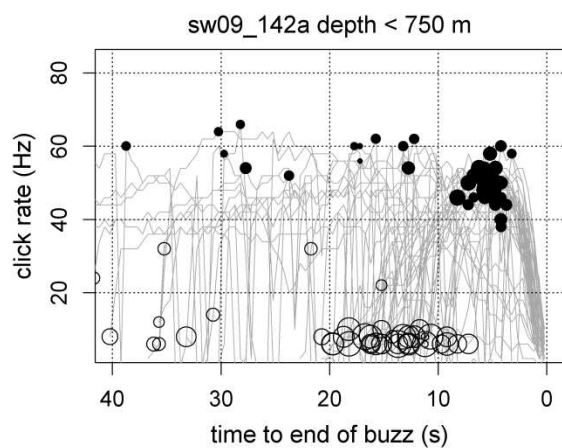
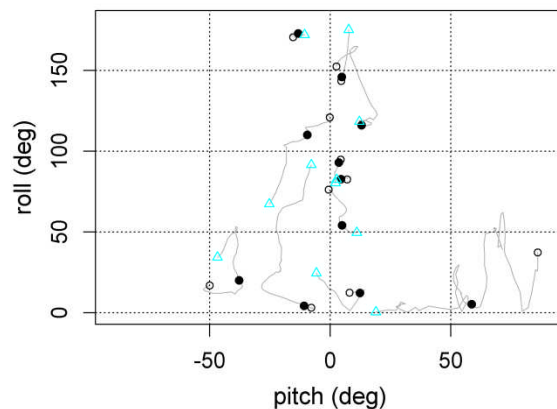
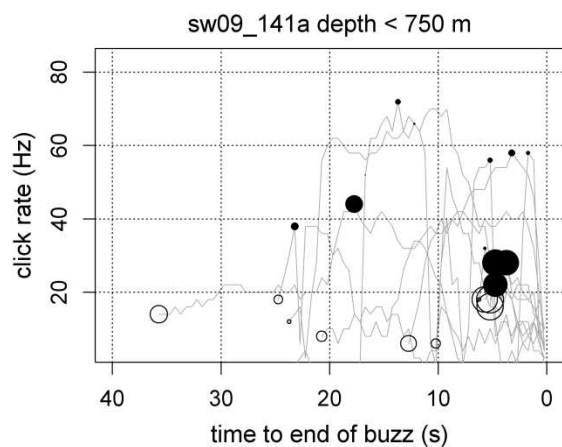
Figure 5D.1 Binned click rate and movement data for each buzz

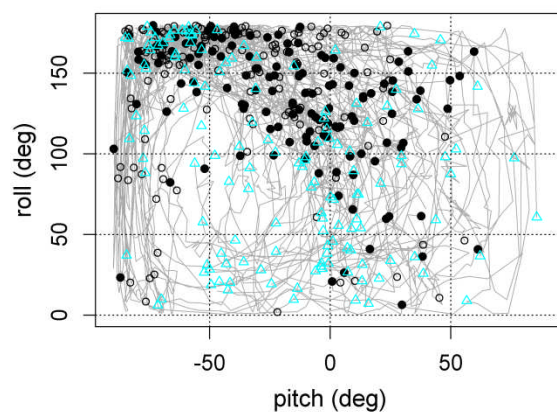
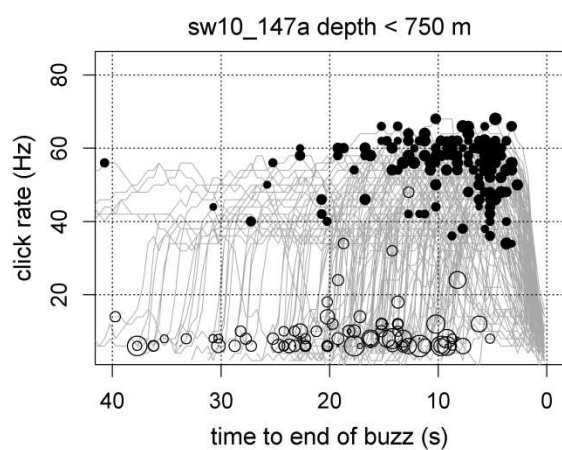
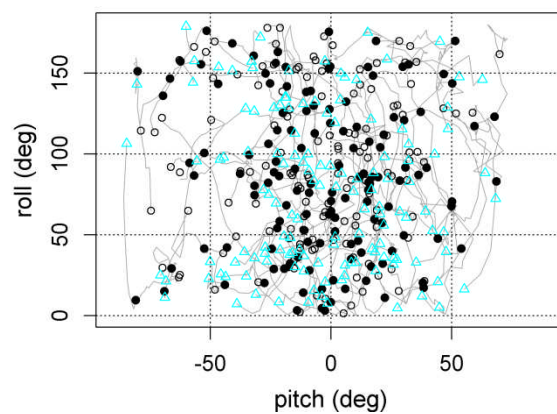
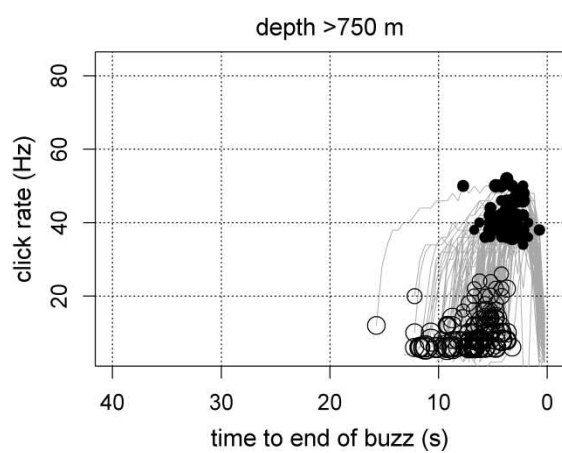
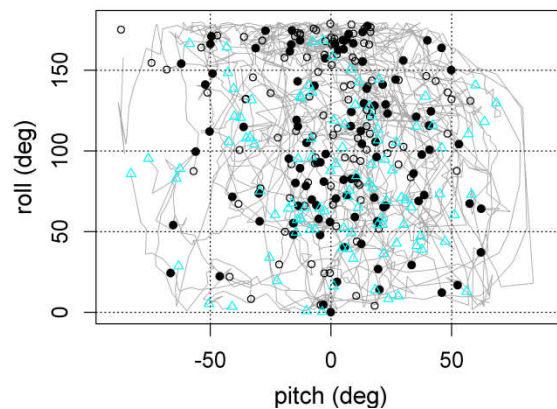
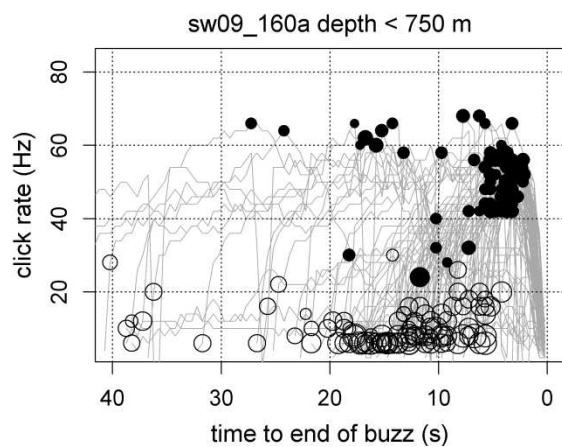
Open circles show start of buzz, solid circles show 0.5 time bin with maximum click rate; cyan triangles show end of buzz.

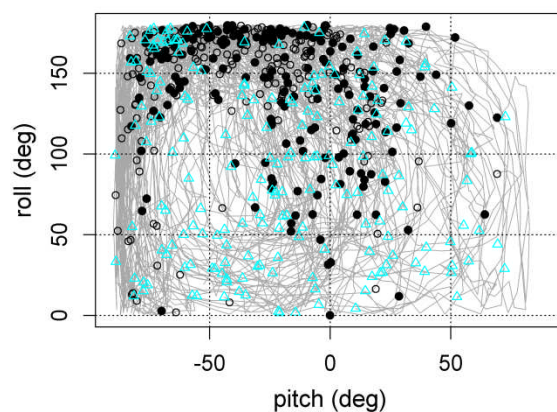
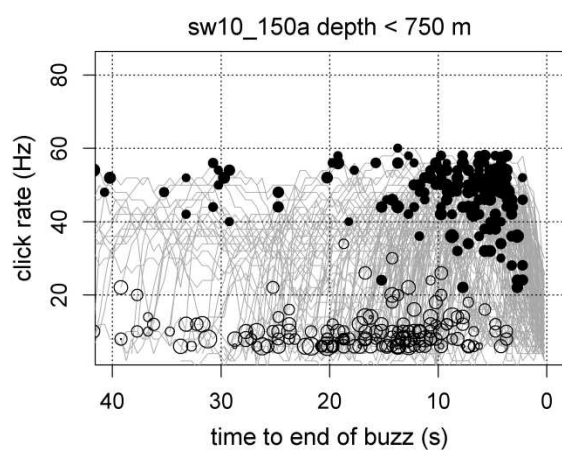
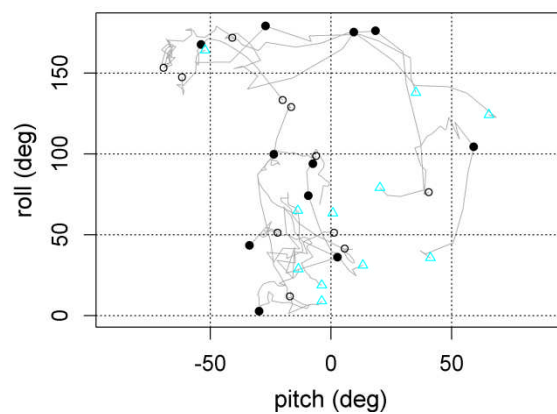
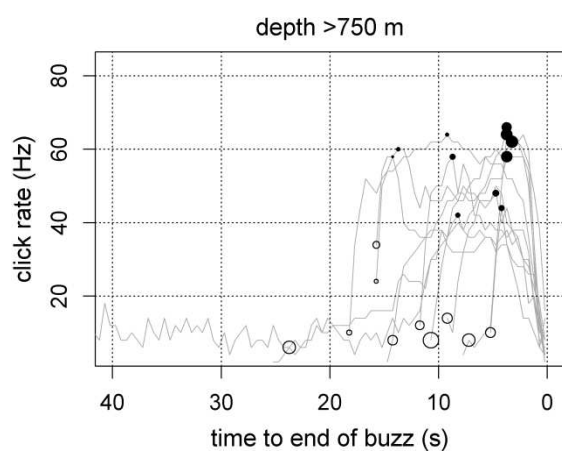
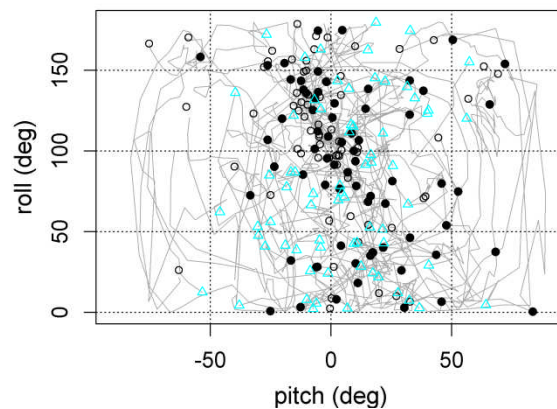
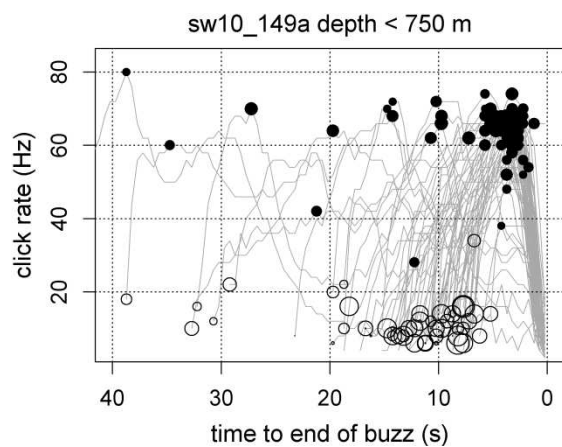












Appendix 5E Across-buzz models for acoustic metrics

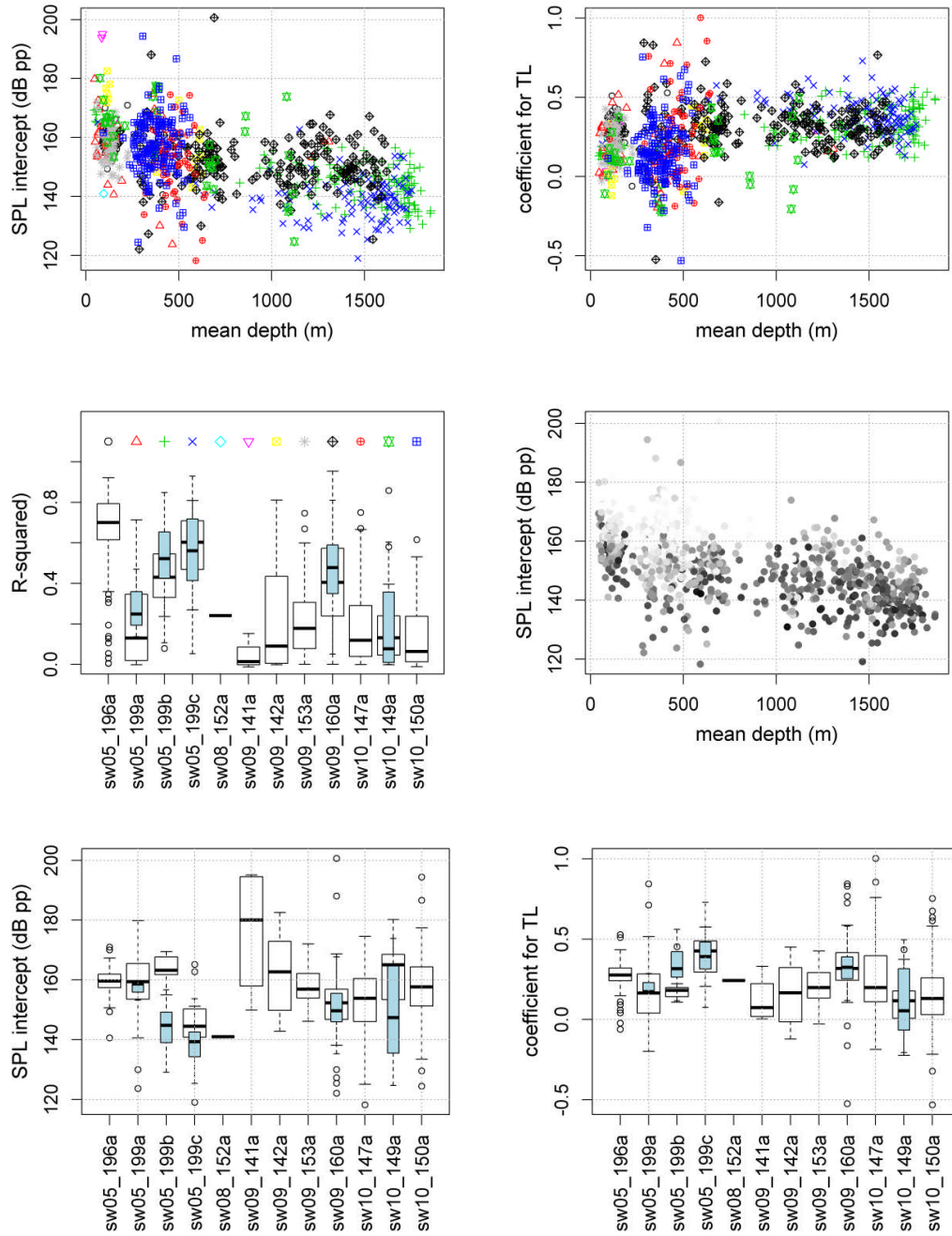


Figure 5E.1. Linear regression coefficients between SPL_{pp} and ICI as a function of depth

Top: intercepts and coefficients when each fine-filtered buzz ($n=910$) was fitted a linear regression with SPL_{pp} as a response variable and expected transmission loss (TL) as a covariate. Middle right: as top left, but coloured by R-squared value (White 0%, Black 100%). Boxplots: narrow blue boxes show data from deep buzzes (>750 m), and white boxes show data from shallower buzzes (<750 m) for each individual.

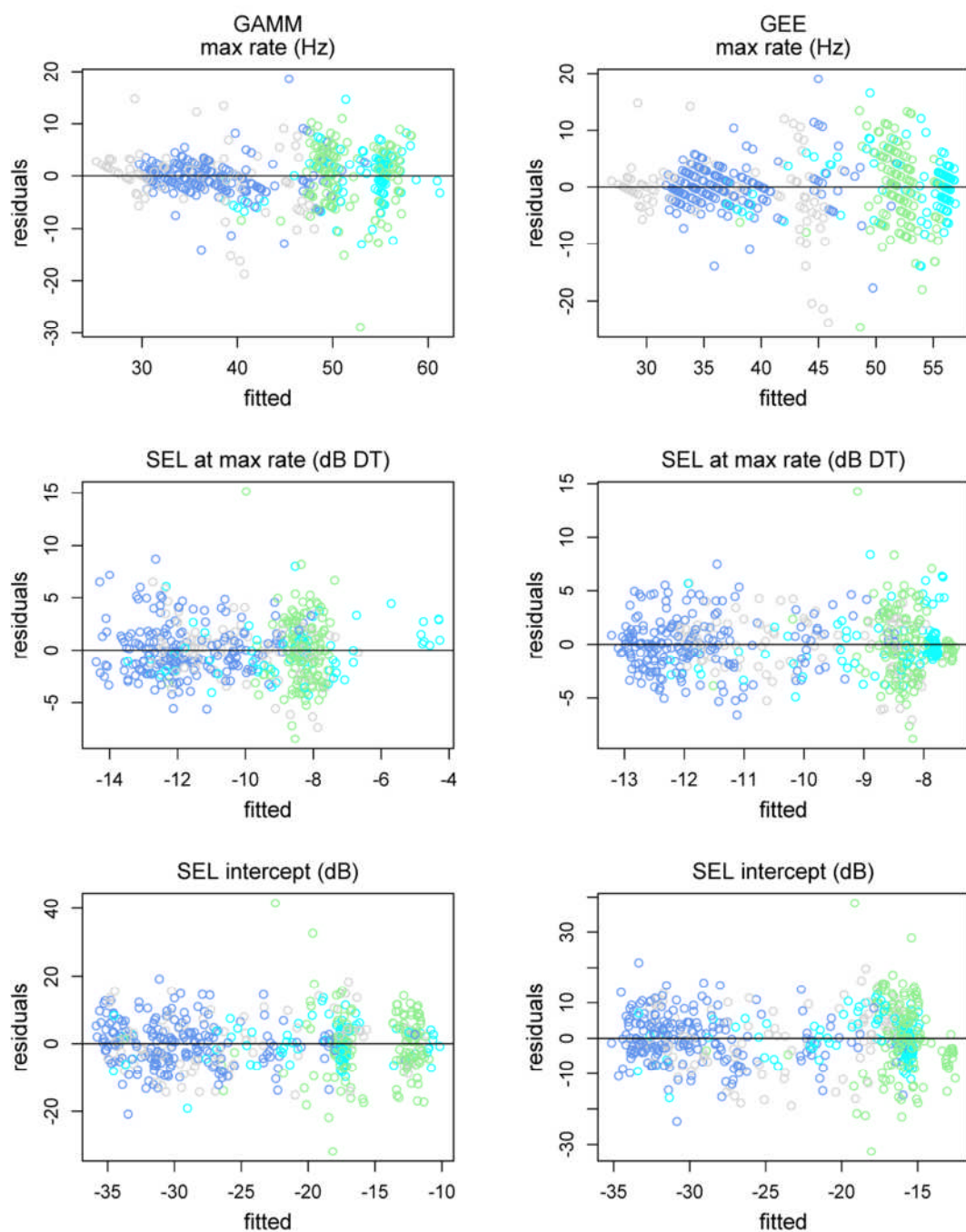


Figure 5E.2 Residuals as a function of fitted values for across-buzz GEE and GAMM models

x-axis: fitted values, y-axis: residuals, left column: GAMM, right column: GEE. Data points are coloured by cluster: blue: deep, grey: deep-descent, green: mid-active, and cyan: shallow-sparse.