SUPPLEMENTARY INFORMATION: A GENERALISED RANDOM ENCOUNTER MODEL FOR ESTIMATING ANIMAL DENSITY WITH REMOTE SENSOR DATA

S1. TABLE OF SYMBOLS

Symbol	Description	Units
θ	Sensor width	rad
α	Animal call/beam width	rad
x_i	Focal Angle $i \in \{1, 2, 3, 4\}$	rad
r	Detection distance	m
$\bar{p}$	Average profile width	m
p	A specific profile width	m
v	Velocity	$m\,s^{-1}$
t	Time	s
z	Number of detections	-
D	Animal density	m^{-2}
T	Step length	s
N	Number of steps per simulation	-
d	Distance moved in a time step	m
S	Probability of remaining stationary	-
A	Maximum turning angle	rad

TABLE S1. List of symbols used to describe the gREM

S2. SUPPLEMENTARY METHODS

S2.1. **Introduction.** This supplementary methods derives all the models used in the paper. For continuity, the gas model derivation is included here as well as in the main text. The calculation of all integrals is included in the Python script S3.

S2.2. **Gas model.** Following **?**, we derive the gas model where sensors can capture animals in any direction and animal's signal is detectable from any direction($\theta = 2\pi$ and $\alpha = 2\pi$). We assume that animals are in a homogeneous environment, and move in straight lines of random direction with velocity v. We allow that our stationary sensor can capture animals at a detection distance r and that if an animal moves within this detection zone they are captured with a probability of one, while animals outside the zone are never captured.

In order to derive animal density, we need to consider relative velocity from the reference frame of the animals. Conceptually, this requires us to imagine that all animals are stationary and randomly distributed in space, while the sensor moves with velocity v. If we calculate the area covered by the sensor during the survey period we can estimate the number of animals the sensor should capture. As a circle moving across a plane, the area covered by the sensor per unit time is $2rv$. The number of expected captures, z, for a survey period of t, with an animal density of D is $z = 2rvtD$. To estimate the density, we rearrange to get $D = z/2rvt$.

S2.2.1. *gREM derivations for different detection and signal widths.* Different combinations of θ and α would be expected to occur (e.g., sensors have different detection widths and animals have different signal widths). For different combinations θ and α, the area covered per unit time is no longer given by $2rv$. Instead of the size of the sensor detection zone having a diameter of $2r$, the size changes with the approach angle between the sensor and the animal. For any given signal width and detector width and depending on the angle that the animal approaches the sensor, the width of the area within which an animal can be detected is called the profile, p. The size of the profile (averaged across all approach angles) is defined as the average profile $\bar{p}$. However, different combinations of θ and α need different equations to calculate $\bar{p}$.

We have identified the parameter space for the combinations of θ and α for which the derivation of the equations are the same (defined as sub-models in the gREM) (Figure **??**). For example, the gas model becomes the simplest gREM sub-model (upper right in (Figure **??**) and the REM from (**?**) is another gREM sub-model where $\theta < \pi/2$ and $\alpha = 2\pi$.

For different values of θ and α, the only thing that changes is that the area covered per unit time is no longer given by $2rv$. Instead of the sensor having a diameter of $2r$, the sensor has a complex diameter that changes with approach angle. The rest of the derivation is just calculating this value for all values of α and θ. However, different regions of this two dimensional parameter space have noncontinuously different models, with different derivations. Therefore we have to identify the regions for which the derivation is the same, and then separately derive p for each region. The separate regions are shown in Figure **??**.

S2.3. **Model SE1.** SE1 is very similar to the gas model except that because $\alpha \leq \pi$ the profile width is no longer $2r$ but is instead limited by the width of the animal call. We therefore get a profile width of $2r\sin(\alpha/2)$ instead.

$$\bar{p}_{SE1} = \frac{1}{\pi} \int_{\frac{\pi}{2}}^{\frac{3\pi}{2}} 2r\sin\left(\frac{\alpha}{2}\right) dx_1 \qquad \text{eqn S1}$$

$$\bar{p}_{SE1} = 2r\sin\left(\frac{\alpha}{2}\right) \qquad \text{eqn S2}$$

S2.4. **Model NE.** When the detection zone is not a circle, we have more complex profiles and need to explicitly write functions for the width of the profile for every approach angle. We then use these functions to find the average profile width $\bar{p}$ for all approach angles by integrating across all 2π angles of approach and dividing by 2π.

There are three submodels within quadrant NE. Note that NE1 covers the area $\alpha = 2\pi$ as well as the triangle below it as these two models are specified exactly the same, rather than happening to have equal results.

These models have up to five profiles.

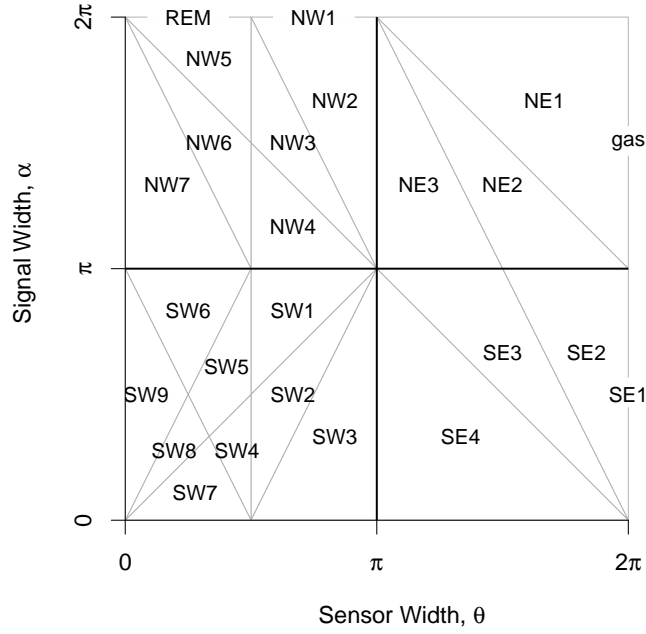(1) The profile width starts, from $x_1 = \frac{\pi}{2}$ as $2r$.

FIGURE S1. The location of each model in parameter space. Each named model must be derived separately. However, the results of the different models are often the same; areas coloured the same have the same result. Other than the gas model and th REM model, individual models are named after the compass point of the quadrant they are in. The region extends past $\alpha, \theta = 2\pi$ to clearly display the models that are defined for only $\alpha = 2\pi$ or $\theta = 2\pi$ (e.g. the REM model is only definied for $\alpha = 2\pi$.

(2) At $x_1 = \theta/2$, the right hand side of the profile cannot be r wide as the corner of the 'blind spot' limits its size to being $r\cos(x_1 - \theta/2)$ wide (see Fig. **??**).

(3) The third profile is only found in NE3. If $\alpha < 4\pi - 2\theta$, then at $x_1 = \theta/2$, when the profile is perpendicular to the edge of the blind spot, the whole right side of the profile is invisible to the sensor (see Fig. **??**). This gives a profile size of just r.

(4) At some point, the sensor can detect animals once they have passed the blind spot giving a profile width of $r + r\cos(x_1 + \theta/2)$. From $x_1 = \pi$, if the animal call is wide enough to be detected in this area, this is the wider profile. This then defines the split between NE1 and NE2. In NE1, with $\alpha > 3\pi - \theta$, the animal call is wide enough that at $x_1 = \pi$ the animal can already be detected past the blind spot and so this profile is used. In NE2, with $\alpha < 3\pi - \theta$, the latter profile is reached at $5\pi/2 - \theta/2 - \alpha/2$ and is therefore dependant on the sizes of α and θ.

(5) Finally, common to all three models, at $x_1 = 2\pi - \theta/2$ the profile becomes a full $2r$ once again.

S2.4.1. *Model NE1.* Submodel NE1 exists within the area bounded by $\alpha \leq 2\pi$, $\theta \leq 2\pi$ and $\alpha \geq 3\pi - \theta$. It has four profiles; it does not include the r profile at $x_1 = \pi$. Furthermore, θ is wide enough that the $r + r\cos(x_1 + \theta/2)$ profile starts at π. This then gives us

$$\bar{p}_{NE1} = \frac{1}{\pi}\left(\int_{\frac{\pi}{2}}^{\frac{\theta}{2}} 2r\,dx_1 + \int_{\frac{\theta}{2}}^{\pi} r\cos\left(\frac{\theta}{2} - x_1\right) + r\,dx_1 \right.$$

$$\left. + \int_{\pi}^{2\pi - \frac{\theta}{2}} r\cos\left(\frac{\theta}{2} + x_1\right) + r\,dx_1 + \int_{2\pi - \frac{\theta}{2}}^{\frac{3\pi}{2}} 2r\,dx_1 \right) \qquad \text{eqn S3}$$

$$\bar{p}_{NE1} = \frac{r}{\pi}\left(\theta + 2\sin\left(\frac{\theta}{2}\right)\right) \qquad \text{eqn S4}$$
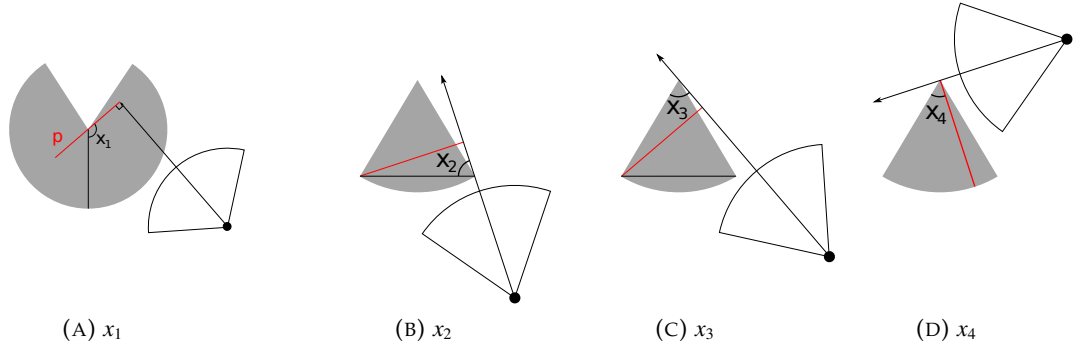
(A) x_1 (B) x_2 (C) x_3 (D) x_4

FIGURE S2. The location of the focal angles $x_{i \in [1,4]}$. In these figures, the sector shaped detection region is shown in grey. Animals are filled black circles and the animal signal is an unfilled sector. The animals direction of movement is indicated with an arrow. The profile p is shown with a red line.

S2.4.2. *Model NE2.* Model NE2 is bounded by $\alpha \leq 3\pi - \theta$, $\alpha \geq 4\pi - 2\theta$ and $\alpha \geq \pi$. It is the same as NE1 except that the third profile starts at $5\pi/2 - \theta/2 - \alpha/2$ instead of at π which is reflected in the different bounds in the second and third integral.

$$\bar{p}_{\text{NE2}} = \frac{1}{\pi} \left(\int_{\frac{\pi}{2}}^{\frac{\theta}{2}} 2r \, dx_1 + \int_{\frac{\theta}{2}}^{\frac{5\pi}{2} - \frac{\theta}{2} - \frac{\alpha}{2}} r \cos\left(\frac{\theta}{2} - x_1\right) + r \, dx_1 \right.$$

$$\left. + \int_{\frac{5\pi}{2} - \frac{\theta}{2} - \frac{\alpha}{2}}^{2\pi - \frac{\theta}{2}} r \cos\left(\frac{\theta}{2} + x_1\right) + r \, dx_1 + \int_{2\pi - \frac{\theta}{2}}^{\frac{3\pi}{2}} 2r \, dx_1 \right) \qquad \text{eqn S5}$$

$$\bar{p}_{\text{NE2}} = \frac{r}{\pi} \left(\theta - \cos\left(\frac{\alpha}{2}\right) + \cos\left(\frac{\alpha}{2} + \theta\right) \right) \qquad \text{eqn S6}$$

S2.4.3. *Model NE3.* Model NE3 is bound by $\alpha \leq 4\pi - 2\theta$, $\alpha \geq \pi$ and $\theta \geq \pi$. It is the same as NE2 except that it contains the extra profile with width r (third integral).

$$\bar{p}_{\text{NE3}} = \frac{1}{\pi} \left(\int_{\frac{\pi}{2}}^{\frac{\theta}{2}} 2r \, dx_1 + \int_{\frac{\theta}{2}}^{\frac{\theta}{2} + \frac{\pi}{2}} r \cos\left(\frac{\theta}{2} - x_1\right) + r \, dx_1 \right.$$

$$\left. + \int_{\frac{\theta}{2} + \frac{\pi}{2}}^{\frac{5\pi}{2} - \frac{\theta}{2} - \frac{\alpha}{2}} r \, dx_1 + \int_{\frac{5\pi}{2} - \frac{\theta}{2} - \frac{\alpha}{2}}^{2\pi - \frac{\theta}{2}} r \cos\left(\frac{\theta}{2} + x_1\right) + r \, dx_1 + \int_{2\pi - \frac{\theta}{2}}^{\frac{3\pi}{2}} 2r \, dx_1 \right) \qquad \text{eqn S7}$$

$$\bar{p}_{\text{NE3}} = \frac{r}{\pi} \left(\theta - \cos\left(\frac{\alpha}{2}\right) + 1 \right) \qquad \text{eqn S8}$$

S2.5. **Models SE2–4.** Quadrant SE contains three submodels (excluding SE1) that differ in ways reminiscent of the models in NE. There are four possible profiles.

(1) As α is less than π the profile is smaller than $2r$, even when the sensor width is a full diameter. The profile width starts as $2r \sin(\alpha/2)$.
(2) Similar to NE, at a certain point the blind spot of the sensor area limits the profile width on one side. This gives a profile width of $r \sin(\alpha/2) + r \cos(x_1 - \theta/2)$.
(3) Also similar to NE, there can be a point where the right side of the profile is 0 giving a profile width of $r \sin(\alpha/2)$.
(4) If $\alpha \leq 2\pi - \theta$, then at $x_1 = \theta/2 + \pi/2 + \alpha/2$ the profile width becomes 0. This inequality distinguishes between SE3 and SE4.
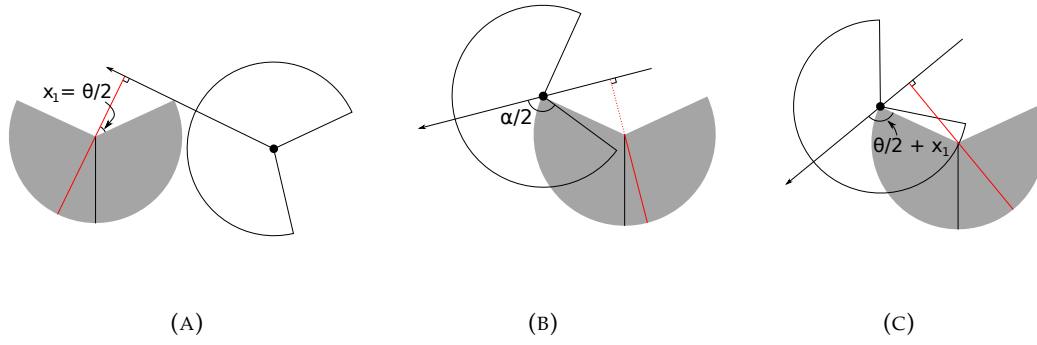
(A) (B) (C)

FIGURE S3. A) The second integral in NE with width $r + r\cos(x_1 - \theta/2)$ B) The third integral in NE3. $\alpha/2$ is labelled. As it is small, animals to the right of the detector cannot be detected (shown by a dashed red line.) C) After further rotation, $\alpha/2$ is now bigger than the angle shown and animals to the right of the detector can again be sensed.

(5) The third profile $r\sin(\alpha/2)$ starts at $\theta/2 + \pi/2$ while at $5\pi/2 - \alpha/2 - \theta/2$ the profile returns to size $2r\sin(\alpha/2)$. If $\theta/2 + \pi/2 \geq 5\pi/2 - \alpha/2 - \theta/2$ we go straight into the $2r\sin(\alpha/2)$ profile and miss the $r\sin(\alpha/2)$ profile. SE2 and SE3 are seperated by this inequality which simplifies to $\alpha \leq 4\pi - 2\theta$.

S2.5.1. *Model SE2.* SE2 is bounded by $\alpha \geq 4\pi - 2\theta$, $\alpha \leq \pi$ and $\theta \leq 2\pi$. As $\alpha \geq 4\pi - 2\theta$, there is no $r\sin(\alpha/2)$ profile. As $\alpha \leq 4\pi - 2\theta$, the profile returns to $2r\sin(\alpha/2)$ rather than going to 0. These integrals relate to profiles (1), (2) and (5) above.

$$\bar{p}_{SE2} = \frac{1}{\pi}\left(\int_{\frac{\pi}{2}}^{\frac{\pi}{2}+\frac{\theta}{2}-\frac{\alpha}{2}} 2r\sin\left(\frac{\alpha}{2}\right) dx_1 + \int_{\frac{\pi}{2}+\frac{\theta}{2}-\frac{\alpha}{2}}^{\frac{5\pi}{2}-\frac{\theta}{2}-\frac{\alpha}{2}} r\sin\left(\frac{\alpha}{2}\right) + r\cos\left(\frac{\theta}{2} - x_1\right) dx_1 + \int_{\frac{5\pi}{2}-\frac{\theta}{2}-\frac{\alpha}{2}}^{\frac{3\pi}{2}} 2r\sin\left(\frac{\alpha}{2}\right) dx_1 \right) \quad \text{eqn S9}$$

$$\bar{p}_{SE2} = \frac{r}{\pi}\left(\theta\sin\left(\frac{\alpha}{2}\right) - \cos\left(\frac{\alpha}{2}\right) + \cos\left(\frac{\alpha}{2} + \theta\right) \right) \quad \text{eqn S10}$$

S2.5.2. *Model SE3.* SE3 is bounded by $4\pi - 2\theta \leq \alpha \leq 4\pi - 2\theta$ and $\alpha \leq \pi$. Therefore there is a $r\sin(\alpha/2)$ profile but no $0r$ profile. This relates to profiles (1), (2), (3) and (5) above.

$$\bar{p}_{SE3} = \frac{1}{\pi}\left(\int_{\frac{\pi}{2}}^{\frac{\pi}{2}+\frac{\theta}{2}-\frac{\alpha}{2}} 2r\sin\left(\frac{\alpha}{2}\right) dx_1 + \int_{\frac{\pi}{2}+\frac{\theta}{2}-\frac{\alpha}{2}}^{\frac{\theta}{2}+\frac{\pi}{2}} r\sin\left(\frac{\alpha}{2}\right) + r\cos\left(\frac{\theta}{2} - x_1\right) dx_1 \right.$$

$$\left. + \int_{\frac{\theta}{2}+\frac{\pi}{2}}^{\frac{5\pi}{2}-\frac{\theta}{2}-\frac{\alpha}{2}} r\sin\left(\frac{\alpha}{2}\right) dx_1 + \int_{\frac{5\pi}{2}-\frac{\theta}{2}-\frac{\alpha}{2}}^{\frac{3\pi}{2}} 2r\sin\left(\frac{\alpha}{2}\right) dx_1 \right) \quad \text{eqn S11}$$

$$\bar{p}_{SE3} = \frac{r}{\pi}\left(\theta\sin\left(\frac{\alpha}{2}\right) - \cos\left(\frac{\alpha}{2}\right) + 1 \right) \quad \text{eqn S12}$$

S2.5.3. *Model SE4.* Finally SE4 is bounded by $\alpha \leq 4\pi - 2\theta$, $\alpha \leq \pi$ and $\theta \leq \pi$. It is the same as SE3 except that the profile becomes 0 rather than returning to $2r\sin(\alpha/2)$. This relates to profiles (1), (2), (3) and (4) above though profile (4) with width 0 is not shown.

$$\bar{p}_{SE4} = \frac{1}{\pi}\left(\int_{\frac{\pi}{2}}^{\frac{\pi}{2}+\frac{\theta}{2}-\frac{\alpha}{2}} 2r\sin\left(\frac{\alpha}{2}\right) dx_1 + \int_{\frac{\pi}{2}+\frac{\theta}{2}-\frac{\alpha}{2}}^{\frac{\theta}{2}+\frac{\pi}{2}} r\sin\left(\frac{\alpha}{2}\right) + r\cos\left(\frac{\theta}{2} - x_1\right) dx_1 + \int_{\frac{\theta}{2}+\frac{\pi}{2}}^{\frac{\alpha}{2}+\frac{\theta}{2}+\frac{\pi}{2}} r\sin\left(\frac{\alpha}{2}\right) dx_1 \right) \quad \text{eqn S13}$$

$$\bar{p}_{SE4} = \frac{r}{\pi}\left(\theta\sin\left(\frac{\alpha}{2}\right) - \cos\left(\frac{\alpha}{2}\right) + 1 \right) \quad \text{eqn S14}$$
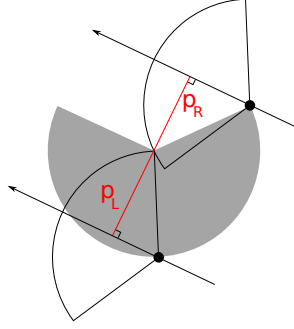
FIGURE S4. A) The second integral in SE. The right side of the profile (p_R) is limited by the size of the sensor region while the left side of the profile (p_L) is limited by the size of the call angle. The full profile has width $p = r \sin(\alpha/2) + r \cos(\theta/2 - x_1)$.

S2.6. Model NW1.

NW1 is the first model with $\theta < \pi$. Whereas previously the focal angle has always been x_1, we now use different focal angles. x_2 and x_3 correspond to γ_1 and γ_2 in **?** while x_4 is new. They are described in Fig. **??**.

There are five different profiles in NW1.

(1) x_2 has an interval of $[\pi/2, \theta/2]$ which is from the angle of approach being directly towards the sensor until the profile is parellel to the left hand radius of the sensor sector. During this interval the profile width is $2r \sin(\theta/2) \sin(x_2)$ which is calculated using the equation for the length of a chord (see Fig. **??**). Note that while rotating anti-clockwise (as usual) x_2 decreases in size.

(2) From here, we examine focal angle x_4 (note that x_3 is used in later models, but is not relevant here.) The left side of the profile is a full radius while the right side is limited to $-r \cos(x_4 - \theta)$ (see Fig. **??**).

(3) At $x_4 = \theta - \pi/2$, the profile is perpendicular to the edge of the sensor area. Here, the right side of the profile is $0r$ giving a profile size of r.

(4) When $x_4 = \pi/2$ the angle of approach is from behind the sensor, but we can once again be detected on the right side of the sensor (see Fig. **??**). Therefore the width of the profile is $r - r \cos(x_4)$.

(5) Finally, we have the x_2 profile, but from behind.

$$\bar{p}_{\text{NW1}} = \frac{1}{\pi} \left(\int_{\frac{\theta}{2}}^{\frac{\pi}{2}} 2r \sin\left(\frac{\theta}{2}\right) \sin(x_2) \, dx_2 + \int_0^{\theta - \frac{\pi}{2}} r - r \cos(-x_4 + \theta) \, dx_4 \right.$$

$$\left. + \int_{\theta - \frac{\pi}{2}}^{\frac{\pi}{2}} r \, dx_4 + \int_{\frac{\pi}{2}}^{\theta} r - r \cos(x_4) \, dx_4 + \int_{\frac{\theta}{2}}^{\frac{\pi}{2}} 2r \sin\left(\frac{\theta}{2}\right) \sin(x_2) \, dx_2 \right) \qquad \text{eqn S15}$$

$$\bar{p}_{\text{NW1}} = \frac{r}{\pi} (\theta + 2) \qquad \text{eqn S16}$$

S2.7. Models NW2–4.

The models NW2–4 have the five potential profiles in NW1 but not all profiles occur in each model, and the angle at which transitions occur are different. Furthermore, there is one extra profile possible.

(1) When approaching the sensor from behind, there is a period where the profile is r wide as in NW1 profile (3).

(2) At some point the right side of the profile becomes viable again. If this occurs in the x_4 region, the profile width becomes $r - r \cos(x_4)$ as in NW1.

(3) However, as α is now less than 2π, the right side of the profile might not be viable until we are in the second x_2 region. In this case, when we first enter the second x_2 region, the profile has a width of $r \cos(x_2 - \theta/2)$. This occurs only if $\alpha \leq 3\pi - 2\theta$. This is inequality is found by noting that the right side of the profile become viable at $x_4 = 3\pi/2 - \alpha$ but the x_2 region starts at $x_4 = \theta$. The new profile in x_2 will only occur if $\theta < 3\pi/2 - \alpha/2$ which is rearranged to find the inequality above. This defines the boundary between NW2 and NW3.
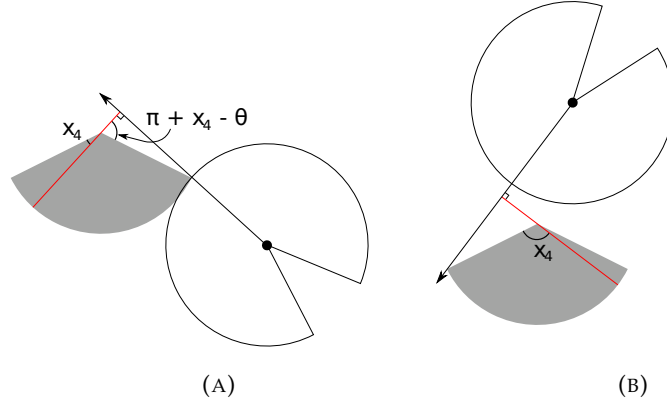
FIGURE S5. A) and B) The second and fourth profiles of NW1. The left side of of both profiles is of width r while the right side is $r\cos(\pi + x_4 - \theta) = -r\cos(\theta - x_4)$ and $r\cos(\pi - x_4) = -r\cos x_4$ respectively.

(4) As $\alpha \le 2\pi$ it is possible that when the angle of approach is from directly behind the sensor the animal will not be detected at all. This is the case if $\alpha/2 \le \pi - \theta/2$ as shown in Fig. **??**. This inequality (simplified as $\alpha \le 2\pi - \theta$) defines the boundary between NW3 and NW4.

S2.7.1. *Model NW2.* NW2 is bounded by $\alpha \ge 3\pi - 2\theta$, $\alpha \le 2\pi$ and $\theta \le \pi$.

NW2 has all five profiles as found in NW1. However, the change from the r profile (third integral) to the $r - r\cos(x_4)$ profile (fourth integral) occurs at $x_4 = 3\pi/2 - \alpha/2$ instead of at $x_4 = \theta$.

$$\bar{p}_{\text{NW2}} = \frac{1}{\pi}\left(\int_{\frac{\theta}{2}}^{\frac{\pi}{2}} 2r\sin\left(\frac{\theta}{2}\right)\sin(x_2)\,dx_2 + \int_0^{\theta - \frac{\pi}{2}} r - r\cos(-x_4 + \theta)\,dx_4 \right.$$

$$\left. + \int_{\theta - \frac{\pi}{2}}^{\frac{3\pi}{2} - \frac{\alpha}{2}} r\,dx_4 + \int_{\frac{3\pi}{2} - \frac{\alpha}{2}}^{\theta} r - r\cos(x_4)\,dx_4 + \int_{\frac{\theta}{2}}^{\frac{\pi}{2}} 2r\sin\left(\frac{\theta}{2}\right)\sin(x_2)\,dx_2 \right) \qquad \text{eqn S17}$$

$$\bar{p}_{\text{NW2}} = \frac{r}{\pi}\left(\theta - \cos\left(\frac{\alpha}{2}\right) + 1\right) \qquad \text{eqn S18}$$

S2.7.2. *Model NW3.* NW3 is bounded by $\alpha \le 3\pi - 2\theta$, $\alpha \ge 2\pi - \theta$ and $\theta \ge \pi/2$.

NW3 does not have the fourth integral from NW2 as the right side of the profile does not become viable until after the x_4 region has ended and the x_2 region has begun. Therefore the second x_4 integral has an upper limit of θ and the integral after has a width of $r\cos(x_2 - \theta/2)$ and is integrated with respect to x_2. The final integral starts at $x_4 = 3\pi/2 - \alpha/2 - \theta/2$ and has the full width of $2r\sin(x_2)\sin(\theta/2)$.

$$\bar{p}_{\text{NW3}} = \frac{1}{\pi}\left(\int_{\frac{\theta}{2}}^{\frac{\pi}{2}} 2r\sin\left(\frac{\theta}{2}\right)\sin(x_2)\,dx_2 + \int_0^{\theta - \frac{\pi}{2}} r - r\cos(-x_4 + \theta)\,dx_4 \right.$$

$$\left. + \int_{\theta - \frac{\pi}{2}}^{\theta} r\,dx_4 + \int_{\frac{\theta}{2}}^{\frac{3\pi}{2} - \frac{\theta}{2} - \frac{\alpha}{2}} r\cos\left(\frac{\theta}{2} - x_2\right)\,dx_2 + \int_{\frac{3\pi}{2} - \frac{\theta}{2} - \frac{\alpha}{2}}^{\frac{\pi}{2}} 2r\sin\left(\frac{\theta}{2}\right)\sin(x_2)\,dx_2 \right) \qquad \text{eqn S19}$$

$$\bar{p}_{\text{NW3}} = \frac{r}{\pi}\left(\theta - \cos\left(\frac{\alpha}{2}\right) + 1\right) \qquad \text{eqn S20}$$

S2.7.3. *Model NW4.* Finally, NW4 is bounded by $\alpha \ge \pi$, $\theta \ge \pi/2$ and $\alpha \le 2\pi - \theta$. NW4 is the same as NW3 except that the final profile width is zero and this profile is reached at $\alpha/2 + \theta/2 - \pi/2$.
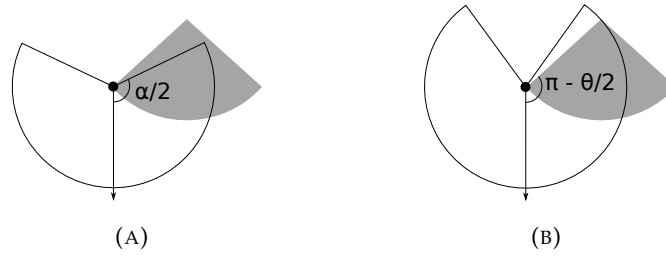
FIGURE S6. A) If $\alpha/2$ is less than $\pi - \theta/2$, as is the case here, then the width of the profile when an animal approaches directly from behind is zero. B) If $\alpha/2 > \pi - \theta/2$ the profile width from behind is $2r\sin\left(\frac{\theta}{2}\right)\sin(x_2)$.

$$\bar{p}_{\text{NW4}} = \frac{1}{\pi}\left(\int_{\frac{\theta}{2}}^{\frac{\pi}{2}} 2r\sin\left(\frac{\theta}{2}\right)\sin\left(x_2\right) \, \mathrm{d}x_2 + \int_{0}^{\theta-\frac{\pi}{2}} r - r\cos\left(-x_4 + \theta\right) \, \mathrm{d}x_4 \right.$$

$$\left. + \int_{\theta-\frac{\pi}{2}}^{\theta} r \, \mathrm{d}x_4 + \int_{\frac{\theta}{2}}^{\frac{\alpha}{2}+\frac{\theta}{2}-\frac{\pi}{2}} r\cos\left(\frac{\theta}{2} - x_2\right) \, \mathrm{d}x_2 \right) \quad \text{eqn S21}$$

$$\bar{p}_{\text{NW4}} = \frac{r}{\pi}\left(\theta - \cos\left(\frac{\alpha}{2}\right) + 1\right) \quad \text{eqn S22}$$

S2.8. Model REM.

REM is the model from (**?**). It has $\alpha = 2\pi$ and $\theta \leq \pi/2$. It has three profile widths, two of which are repeated, once as the animal approaches from on front of the sensor and once as the animal approaches from behind the sensor.

(1) Starting with an approach direction of directly towards the sensor, and examining focal angle x_2, the profile width is $2r\sin(x_2)\sin(\theta/2)$.
(2) When the profile is perpendicular to the radius edge of the sector sensor region, we instead examine x_3 where the profile width is $r\sin(x_3)$.
(3) At $x_3 = \pi/2$ the profile becomes simply r and this continues for θ radians of x_4.
(4) The x_3 profile is then repeated with an approach direction from behind the sensor.
(5) Finally the x_2 profile is repeated, again with an approach direction from behind the sensor.

$$\bar{p}_{\text{REM}} = \frac{1}{\pi}\left(\int_{\frac{\pi}{2}-\frac{\theta}{2}}^{\frac{\pi}{2}} 2r\sin\left(\frac{\theta}{2}\right)\sin\left(x_2\right) \, \mathrm{d}x_2 + \int_{\theta}^{\frac{\pi}{2}} r\sin\left(x_3\right) \, \mathrm{d}x_3 \right.$$

$$\left. + \int_{0}^{\theta} r \, \mathrm{d}x_4 + \int_{\theta}^{\frac{\pi}{2}} r\sin\left(x_3\right) \, \mathrm{d}x_3 + \int_{\frac{\pi}{2}-\frac{\theta}{2}}^{\frac{\pi}{2}} 2r\sin\left(\frac{\theta}{2}\right)\sin\left(x_2\right) \, \mathrm{d}x_2 \right) \quad \text{eqn S23}$$

$$\bar{p}_{\text{REM}} = \frac{r}{\pi}\left(\theta + 2\right) \quad \text{eqn S24}$$

S2.9. Model NW5–7.

In the models in NW5–7, the sensor has $\theta \leq \pi/2$ as in the REM. As $\alpha \geq \pi/2$ a lot of the profiles are similar to the REM. Specifically, the first three profiles are always the same as the first three profiles of the REM. This is because when an animal is moving towards the sensor, the $\alpha \geq \pi$ call is no different to a 2π call. However, when approaching the sensor from behind, things are slightly different. The animal can only be detected by the sensor if the signal width is large enough that it can be detected once it has passed the sensor.

(1) Starting with an approach direction of directly towards the sensor, and examining focal angle x_2, the profile width is $2r\sin(x_2)\sin(\theta/2)$.
(2) When the profile is perpendicular to the radius edge of the sector sensor region, we instead examine x_3 where the profile width is $r\sin(x_3)$.

(3) At $x_3 = \pi/2$ the profile becomes simply r and this continues for θ radians of x_4.

(4) If $\alpha \leq 2\pi + 2\theta$, the animal becomes undetectable during this profile when x_3 has decreased in size to $\pi - \alpha/2$. This inequality marks the boundary between NW7 and NW6.

(5) If instead $\alpha \geq 2\pi + 2\theta$ then the animal does not become undetectable during the x_3 focal angle. Instead the profile has width greater than zero for the whole of the x_3 angle and the the second x_2 angle is reached. The profile starts with width $r\cos(x_2 - \theta/2)$ as only animals approaching to the left of the sensor are detectable.

(6) During this second x_2 profile the call angle needed for animals to be detected to the left of the detector is increasing while the angle needed for animals to be detected to the right of the detector is decreasing. Therefore, either the left side becomes undetectable, making both sides undetectable (this occurs if $\alpha \leq 2\pi - \theta$ as in NW6)

(7) or the right becomes detectable (if $\alpha \geq 2\pi - \theta$ as in NW5), making both sides detectable and giving a profile width of $2r\sin(x_2)\sin(\theta/2)$.

S2.9.1. *Model NW5.* NW5 is bounded by $\alpha \geq 2\pi - \theta$, $\alpha \leq 2\pi$ and $\theta \leq \pi/2$.

It is the same as REM except that it includes the extra profile in x_2 (the fifth integral) where only animals approaching to the left of the profile are detected.

$$\bar{p}_{\text{NW5}} = \frac{1}{\pi}\left(\int_{\frac{\pi}{2} - \frac{\theta}{2}}^{\frac{\pi}{2}} 2r\sin\left(\frac{\theta}{2}\right)\sin(x_2)\, dx_2 + \int_{\theta}^{\frac{\pi}{2}} r\sin(x_3)\, dx_3 + \int_{0}^{\theta} r\, dx_4 \right.$$

$$\left. + \int_{\theta}^{\frac{\pi}{2}} r\sin(x_3)\, dx_3 + \int_{\frac{\pi}{2} - \frac{\theta}{2}}^{\frac{3\pi}{2} - \frac{\theta}{2} - \frac{\alpha}{2}} r\cos\left(\frac{\theta}{2} - x_2\right) dx_2 + \int_{\frac{3\pi}{2} - \frac{\theta}{2} - \frac{\alpha}{2}}^{\frac{\pi}{2}} 2r\sin\left(\frac{\theta}{2}\right)\sin(x_2)\, dx_2 \right) \qquad \text{eqn S25}$$

$$\bar{p}_{\text{NW5}} = \frac{r}{\pi}\left(\theta - \cos\left(\frac{\alpha}{2}\right) + 1\right) \qquad \text{eqn S26}$$

S2.9.2. *Model NW6.* NW6 is bounded by $\alpha \leq 2\pi - \theta$, $\alpha \geq 2\pi + 2\theta$ and $\theta \leq \pi/2$

NW6 is the same NW5 except that as $\alpha \leq 2\pi - \theta$, animals that approach from directly behind the detector are not detected. Therefore at $x_2 = \alpha/2 + \theta/2 - \pi/2$ the profile width goes to zero and therefore the last integral in NW5 is not included.

$$\bar{p}_{\text{NW6}} = \frac{1}{\pi}\left(\int_{\frac{\pi}{2} - \frac{\theta}{2}}^{\frac{\pi}{2}} 2r\sin\left(\frac{\theta}{2}\right)\sin(x_2)\, dx_2 + \int_{\theta}^{\frac{\pi}{2}} r\sin(x_3)\, dx_3 \right.$$

$$\left. + \int_{0}^{\theta} r\, dx_4 + \int_{\theta}^{\frac{\pi}{2}} r\sin(x_3)\, dx_3 + \int_{\frac{\pi}{2} - \frac{\theta}{2}}^{\frac{\alpha}{2} + \frac{\theta}{2} - \frac{\pi}{2}} r\cos\left(\frac{\theta}{2} - x_2\right) dx_2 \right) \qquad \text{eqn S27}$$

$$\bar{p}_{\text{NW6}} = \frac{r}{\pi}\left(\theta - \cos\left(\frac{\alpha}{2}\right) + 1\right) \qquad \text{eqn S28}$$

S2.9.3. *Model NW7.* NW7 is bounded by $\alpha \geq 2\pi + 2\theta$, $\alpha \geq \pi$ and $\theta \geq 0$.

It is similar to NW6 but does not include the last integral as during the x_3 profile, at $x_3 = \pi - \alpha/2$ the call width is too small for any animals to be detected, so the profile width goes to zero.

$$\bar{p}_{\text{NW7}} = \frac{1}{\pi}\left(\int_{\frac{\pi}{2} - \frac{\theta}{2}}^{\frac{\pi}{2}} 2r\sin\left(\frac{\theta}{2}\right)\sin(x_2)\, dx_2 + \int_{\theta}^{\frac{\pi}{2}} r\sin(x_3)\, dx_3 \right.$$

$$\left. + \int_{0}^{\theta} r\, dx_4 + \int_{\pi - \frac{\alpha}{2}}^{\frac{\pi}{2}} r\sin(x_3)\, dx_3 \right) \qquad \text{eqn S29}$$

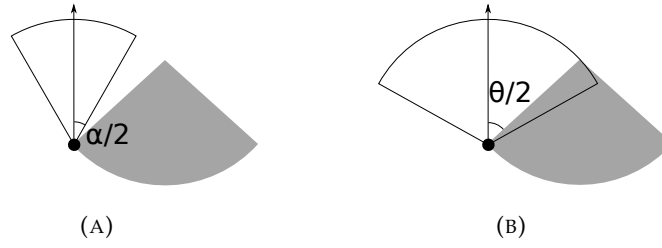$$\bar{p}_{\text{NW7}} = \frac{r}{\pi}\left(\theta - \cos\left(\frac{\alpha}{2}\right) + 1\right) \qquad \text{eqn S30}$$

FIGURE S7. A) As $\alpha/2 < \theta/2$ the profile width is limited by the call angle rather than the sensor region. The profile width is $2r\sin\left(\frac{\alpha}{2}\right)$ B) As $\alpha/2 > \theta/2$ the profile width is limited by the sensor region, not the call angle. The profile width is $2r\sin\left(\frac{\theta}{2}\right)\sin(x_2)$.

S2.10. **Model SW1–3.** The models in SW1–3 are described with the two focal angles used in models NW2–4, x_2 and x_4. As $\alpha \leq \pi$ an animal can never be detected if it is approaching the detector from behind. This makes these models simpler in that they go through the x_2 and x_4 profiles only once each.

There are five potential profile sizes.

(1) At the beginning of x_2, with an approach direction directly towards the sensor, the parameter that limits the width of the profile can either be the sensor width, in which case the profile width is $2r\sin(\theta/2)\sin(x_2)$.

(2) Or the call width can be the limiting parameter, in which case the profile width is instead $2r\sin(\alpha/2)$ (see Figure **??**)

(3) The next potential profile in x_2 has a width of $r\sin(\alpha/2) - r\cos(x_2 + \theta/2)$ as the right side of the profile is limited by the width of the sensor region while the left side is limited by the call width. However, the angle at which the profile starts depends on whether the first profile was 1) or 2) above. If the first profile is profile 1) then the profile is limited on both sides by the sensor region and then the left side of the profile becomes limited by the call width. This happens at $x_2 = \pi/2 - \alpha/2 + \theta/2$. If however the first profile was 2) then the first profile is limited by the call width. We move into the new profile when the right side of the profile becomes limited by the sensor region. This occurs at $x_2 = \pi/2 + \alpha/2 - \theta/2$.

(4) In the x_4 region the left side of the profile is always $r\sin(\alpha/2)$ while the right side is either 0, giving a profile of $r\sin(\alpha/2)$.

(5) Or limited by the sensor giving a profile of size $r\sin(\alpha/2) - r\cos(x_4 - \theta)$.

S2.10.1. *Model SW1.* SW1 is bounded by $\alpha \geq \theta$, $\alpha \leq \pi$ and $\theta \leq \pi$.

As α is large the first profile is limited by the size of the sensor region giving it a width of $2r\sin(\theta/2)\sin(x_2)$. It is the only one of the three SW models to start in this way. Later on, still with x_2 as the focal angle the left side of the profile does become limited by the call width. So at $x_2 = \pi/2 - \alpha/2 + \theta/2$ the profile width becomes $r\sin(\alpha/2) - r\cos(x_2 + \theta/2)$.

As we enter the x_4 region, the profile remains limited by the call on the left and by the sensor on the right, giving a profile width of $r\sin(\alpha/2) - r\cos(x_4 - \theta)$. Finally, at $x_4 = \theta - \pi/2$ the right side of the profile becomes zero and the profile is width is $r\sin(\alpha/2)$.

$$\bar{p}_{\text{SW1}} = \frac{1}{\pi}\left(\int_{\frac{\pi}{2}+\frac{\theta}{2}-\frac{\alpha}{2}}^{\frac{\pi}{2}} 2r\sin\left(\frac{\theta}{2}\right)\sin(x_2)\,\mathrm{d}x_2 + \int_{\frac{\theta}{2}}^{\frac{\pi}{2}+\frac{\theta}{2}-\frac{\alpha}{2}} r\sin\left(\frac{\alpha}{2}\right) - r\cos\left(\frac{\theta}{2} + x_2\right)\,\mathrm{d}x_2 \right.$$
$$\left. + \int_0^{\theta-\frac{\pi}{2}} r\sin\left(\frac{\alpha}{2}\right) - r\cos(\theta - x_4)\,\mathrm{d}x_4 + \int_{\theta-\frac{\pi}{2}}^{\frac{\alpha}{2}+\theta-\frac{\pi}{2}} r\sin\left(\frac{\alpha}{2}\right)\,\mathrm{d}x_4 \right) \qquad \text{eqn S31}$$

$$\bar{p}_{\text{SW1}} = \frac{r}{\pi}\left(\theta\sin\left(\frac{\alpha}{2}\right) - \cos\left(\frac{\alpha}{2}\right) + 1 \right) \qquad \text{eqn S32}$$

S2.10.2. *Model SW2.* SW2 is bounded by $\theta \geq \pi/2$, $\alpha \leq \theta$ and $\alpha \geq 2\theta - \pi$.
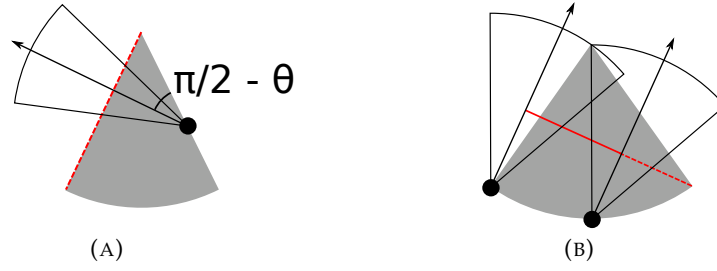
FIGURE S8. A) At $x_4 = 0$, if $\alpha/2 < \pi/2 - \theta$ then $\alpha/2$ is too small for an animal to be detected at all during the x_4 profile (shown with dashed red.) This inequality simplifies to $\alpha < \pi - 2\theta$. B) The right of the profile is limited by the call width, not the sensor. On the left, the profile is limited by the sensor and not the call. Overall the profile width is $r\sin(\alpha/2) - r\cos(x_2 + \theta/2)$.

SW2 is largely similar to SW1. However, as $\alpha \leq \theta$ the first profile is limited by α and not by the detection region. Therefore the first profile has width $2r\sin(\alpha/2)$. This also means the transition to the second profile occurs at $x_2 = \pi/2 + \alpha/2 - \theta/2$ instead of $x_2 = \pi/2 - \alpha/2 + \theta/2$.

$$\bar{p}_{\text{SW2}} = \frac{1}{\pi} \left(\int_{\frac{\alpha}{2} - \frac{\theta}{2} + \frac{\pi}{2}}^{\frac{\pi}{2}} 2r\sin\left(\frac{\alpha}{2}\right) dx_2 + \int_{\frac{\theta}{2}}^{\frac{\alpha}{2} - \frac{\theta}{2} + \frac{\pi}{2}} r\sin\left(\frac{\alpha}{2}\right) - r\cos\left(\frac{\theta}{2} + x_2\right) dx_2 \right.$$
$$\left. + \int_0^{\theta - \frac{\pi}{2}} r\sin\left(\frac{\alpha}{2}\right) - r\cos\left(\theta - x_4\right) dx_4 + \int_{\theta - \frac{\pi}{2}}^{\frac{\alpha}{2} + \theta - \frac{\pi}{2}} r\sin\left(\frac{\alpha}{2}\right) dx_4 \right) \qquad \text{eqn S33}$$

$$\bar{p}_{\text{SW2}} = \frac{r}{\pi} \left(\theta\sin\left(\frac{\alpha}{2}\right) - \cos\left(\frac{\alpha}{2}\right) + 1 \right) \qquad \text{eqn S34}$$

S2.10.3. *Model SW3.* SW3 is bounded by $\alpha \leq 2\theta - \pi$ and $\theta \leq \pi$.

SW3 is similar to SW2 except that the profile does not become limited by sensor at all during the the x_4 regions. Therefore, at $x_4 = 0$ the profile is still of width $2r\sin(\alpha/2)$. Only at $x_4 = \theta - \pi/2 - \alpha/2$ does the profile become limited on the right by the sensor region.

$$\bar{p}_{\text{SW3}} = \frac{1}{\pi} \left(\int_{\frac{\theta}{2}}^{\frac{\pi}{2}} 2r\sin\left(\frac{\alpha}{2}\right) dx_2 + \int_0^{-\frac{\pi}{2} + \theta - \frac{\alpha}{2}} 2r\sin\left(\frac{\alpha}{2}\right) dx_4 \right.$$
$$\left. + \int_{-\frac{\pi}{2} + \theta - \frac{\alpha}{2}}^{\theta - \frac{\pi}{2}} r\sin\left(\frac{\alpha}{2}\right) - r\cos\left(\theta - x_4\right) dx_4 + \int_{\theta - \frac{\pi}{2}}^{\frac{\alpha}{2} + \theta - \frac{\pi}{2}} r\sin\left(\frac{\alpha}{2}\right) dx_4 \right) \qquad \text{eqn S35}$$

$$\bar{p}_{\text{SW3}} = \frac{r}{\pi} \left(\theta\sin\left(\frac{\alpha}{2}\right) - \cos\left(\frac{\alpha}{2}\right) + 1 \right) \qquad \text{eqn S36}$$

S2.11. **Model SW4–9.** As $\alpha < \pi$, animals approaching the sensor from behind can never be detected, so unlike REM, the second x_2 and x_3 profiles are always zero. The six models are split by three inequalities that relate to the models as follows.

(1) Models with $\alpha \leq \pi - 2\theta$ have no x_4 profile. This is because at $x_4 = 0$, the call angle is already too small to be detected as can be seen in Figure **??** where $\alpha/2 < \pi/2 - \theta$ which simplifies to give the previous inequality.
(2) Models with $\alpha \leq \theta$ are limited by α in the first, x_2 region (see Figure **??**), rather than being limited by θ. Therefore this first profile is of width $2r\sin(\alpha/2)$ rather than $2r\sin(\theta/2)\sin(x_2)$.
(3) Finally, models with $\alpha \leq 2\theta$ have a second profile in x_2 where to one side of the sensor α is the limiting factor of profile width, while on the other side θ is (see Figure **??**). This gives a width of $r\sin(\alpha/2) - r\cos(x_2 + \theta/2)$. This profile does not occur in models with $\alpha \geq 2\theta$.

S2.11.1. *Model SW4.* SW4 is bounded by $\alpha \leq \theta$, $\alpha \geq \pi - 2\theta$ and $\theta \leq \pi/2$. Therefore it does contain a x_4 profile, starts with an α limited profile and does contain the $r\sin(\alpha/2) - r\cos(x_2 + \theta/2)$ profile in x_2.

$$\bar{p}_{\text{SW4}} = \frac{1}{\pi}\left(\int_{\frac{\alpha}{2}-\frac{\theta}{2}+\frac{\pi}{2}}^{\frac{\pi}{2}} 2r\sin\left(\frac{\alpha}{2}\right)\,dx_2 + \int_{\frac{\pi}{2}-\frac{\theta}{2}}^{\frac{\alpha}{2}-\frac{\theta}{2}+\frac{\pi}{2}} r\sin\left(\frac{\alpha}{2}\right) - r\cos\left(\frac{\theta}{2}+x_2\right)\,dx_2 \right.$$

$$\left. + \int_{\theta}^{\frac{\pi}{2}} r\sin\left(\frac{\alpha}{2}\right)\,dx_3 + \int_{0}^{\frac{\alpha}{2}+\theta-\frac{\pi}{2}} r\sin\left(\frac{\alpha}{2}\right)\,dx_4 \right) \qquad\qquad \text{eqn S37}$$

$$\bar{p}_{\text{SW4}} = \frac{r}{\pi}\left(\theta\sin\left(\frac{\alpha}{2}\right) - \cos\left(\frac{\alpha}{2}\right) + 1\right) \qquad\qquad \text{eqn S38}$$

S2.11.2. *Model SW5.* SW5 is the only model with a tetrahedral bounding region. It is bounded by $\alpha \geq \theta$, $\alpha \geq \pi - 2\theta$, $\alpha \leq 2\theta$ and $\theta \leq \pi/2$. Therefore it does contain a x_4 profile, but starts with a θ limited profile. It does contain the $r\sin(\alpha/2) - r\cos(x_2 + \theta/2)$ profile in x_2.

$$\bar{p}_{\text{SW5}} = \frac{1}{\pi}\left(\int_{\frac{\pi}{2}+\frac{\theta}{2}-\frac{\alpha}{2}}^{\frac{\pi}{2}} 2r\sin\left(\frac{\theta}{2}\right)\sin(x_2)\,dx_2 + \int_{\frac{\pi}{2}-\frac{\theta}{2}}^{\frac{\pi}{2}+\frac{\theta}{2}-\frac{\alpha}{2}} r\sin\left(\frac{\alpha}{2}\right) - r\cos\left(\frac{\theta}{2}+x_2\right)\,dx_2 \right.$$

$$\left. + \int_{\theta}^{\frac{\pi}{2}} r\sin\left(\frac{\alpha}{2}\right)\,dx_3 + \int_{0}^{\frac{\alpha}{2}+\theta-\frac{\pi}{2}} r\sin\left(\frac{\alpha}{2}\right)\,dx_4 \right) \qquad\qquad \text{eqn S39}$$

$$\bar{p}_{\text{SW5}} = \frac{r}{\pi}\left(\theta\sin\left(\frac{\alpha}{2}\right) - \cos\left(\frac{\alpha}{2}\right) + 1\right) \qquad\qquad \text{eqn S40}$$

S2.11.3. *Model SW6.* SW6 is bounded by $\alpha \geq \pi - 2\theta$, $\alpha \geq 2\theta$ and $\alpha \leq \pi$. It starts with a θ limeted profile and has a x_4 profile. However, it does not contain the $r\sin(\alpha/2) - r\cos(x_2 + \theta/2)$ profile.

$$\bar{p}_{\text{SW6}} = \frac{1}{\pi}\left(\int_{\frac{\pi}{2}-\frac{\theta}{2}}^{\frac{\pi}{2}} 2r\sin\left(\frac{\theta}{2}\right)\sin(x_2)\,dx_2 + \int_{\theta}^{\frac{\alpha}{2}} r\sin(x_3)\,dx_3 \right.$$

$$\left. + \int_{\frac{\alpha}{2}}^{\frac{\pi}{2}} r\sin\left(\frac{\alpha}{2}\right)\,dx_3 + \int_{0}^{\frac{\alpha}{2}+\theta-\frac{\pi}{2}} r\sin\left(\frac{\alpha}{2}\right)\,dx_4 \right) \qquad\qquad \text{eqn S41}$$

$$\bar{p}_{\text{SW6}} = \frac{r}{\pi}\left(\theta\sin\left(\frac{\alpha}{2}\right) - \cos\left(\frac{\alpha}{2}\right) + 1\right) \qquad\qquad \text{eqn S42}$$

S2.11.4. *Model SW7.* SW7 is bounded by $\alpha \leq \pi - 2\theta$, $\alpha \leq \theta$ and $\alpha < 0$. Therefore it does not contain a x_4 profile. It starts with an α limited profile and contains the $r\sin(\alpha/2) - r\cos(x_2 + \theta/2)$ profile in x_2.

$$\bar{p}_{\text{SW7}} = \frac{1}{\pi}\left(\int_{\frac{\alpha}{2}-\frac{\theta}{2}+\frac{\pi}{2}}^{\frac{\pi}{2}} 2r\sin\left(\frac{\alpha}{2}\right)\,dx_2 + \int_{\frac{\pi}{2}-\frac{\theta}{2}}^{\frac{\alpha}{2}-\frac{\theta}{2}+\frac{\pi}{2}} r\sin\left(\frac{\alpha}{2}\right) - r\cos\left(\frac{\theta}{2}+x_2\right)\,dx_2 + \int_{\theta}^{\frac{\alpha}{2}+\theta} r\sin\left(\frac{\alpha}{2}\right)\,dx_3 \right) \quad \text{eqn S43}$$

$$\bar{p}_{\text{SW7}} = \frac{r}{\pi}\left(\theta\sin\left(\frac{\alpha}{2}\right) - \cos\left(\frac{\alpha}{2}\right) + 1\right) \qquad\qquad \text{eqn S44}$$

S2.11.5. *Model SW8.* SW8 is bounded by $\alpha \leq \pi - 2\theta$, $\alpha \geq \theta$ and $\alpha \leq 2\theta$. It starts with a θ limited profile. It does contain the $r\sin(\alpha/2) - r\cos(x_2 + \theta/2)$ profile in x_2 but does not have a x_4 profile.

$$\bar{p}_{\text{SW8}} = \frac{1}{\pi}\left(\int_{\frac{\pi}{2}+\frac{\theta}{2}-\frac{\alpha}{2}}^{\frac{\pi}{2}} 2r\sin\left(\frac{\theta}{2}\right)\sin(x_2)\,dx_2 + \int_{\frac{\pi}{2}-\frac{\theta}{2}}^{\frac{\pi}{2}+\frac{\theta}{2}-\frac{\alpha}{2}} r\sin\left(\frac{\alpha}{2}\right) - r\cos\left(\frac{\theta}{2}+x_2\right)\,dx_2 + \int_{\theta}^{\frac{\alpha}{2}+\theta} r\sin\left(\frac{\alpha}{2}\right)\,dx_3 \right) \quad \text{eqn S45}$$

$$\bar{p}_{\text{SW8}} = \frac{r}{\pi}\left(\theta\sin\left(\frac{\alpha}{2}\right) - \cos\left(\frac{\alpha}{2}\right) + 1\right) \qquad\qquad \text{eqn S46}$$

S2.11.6. *Model SW9.* Finally, SW9, the last model, is bounded by y $\alpha \leq \pi - 2\theta$, $\alpha \geq 2\theta$ and $\theta \geq 0$. Therefore it starts with a θ limited profile. However it doesn't contain the extra x_2 profile nor a x_4 profile.

$$\bar{p}_{\text{SW9}} = \frac{1}{\pi} \left(\int_{\frac{\pi}{2}-\frac{\theta}{2}}^{\frac{\pi}{2}} 2r \sin\left(\frac{\theta}{2}\right) \sin(x_2) \, dx_2 + \int_{\theta}^{\frac{\alpha}{2}} r \sin(x_3) \, dx_3 + \int_{\frac{\alpha}{2}}^{\frac{\alpha}{2}+\theta} r \sin\left(\frac{\alpha}{2}\right) \, dx_3 \right) \qquad \text{eqn S47}$$

$$\bar{p}_{\text{SW9}} = \frac{r}{\pi} \left(\theta \sin\left(\frac{\alpha}{2}\right) - \cos\left(\frac{\alpha}{2}\right) + 1 \right) \qquad \text{eqn S48}$$

S3. SUPPLEMENTARY SCRIPT: SYMBOLIC ALGEBRA PYTHON SCRIPT

This script uses the SymPy package **?**, a computer algebra system to calculate the equations for *p* in the various models and to perform unit checks on the results.

```python
"""
Systematic analysis of REM models
Tim Lucas
01/10/13
"""


from sympy import *
import numpy as np
import matplotlib.pyplot as pl
from datetime import datetime


# Use LaTeX printing
from sympy import init_printing ;
init_printing()
# Make LaTeX output white. Because I use a dark theme
init_printing(forecolor="White")


# Load symbols used for symbolic maths
t, a, r, x2, x3, x4, x1 = symbols('theta alpha r x_2 x_3 x_4 x_1', positive=True)
r1 = {r:1} # useful for lots of checks


# Define functions
# Calculate the final profile averaged over pi.
def calcModel(model):
        x = pi**-1 * sum( [integrate(m[0], m[1:]) for m in model] ).simplify().trigsimp()
        return x

# Do the replacements fit within the area defined by the conditions?
def confirmReplacements(conds, reps):
        if not all([c.subs(reps) for c in eval(conds)]):
                print('reps' + conds[4:] + ' incorrect')

# is average profile in range 0r-2r?
def profileRange(prof, reps):
        if not 0 <= eval(prof).subs(dict(reps, **r1)) <= 2:
                print('Total ' + prof + ' not in 0, 2r')

# Are the individuals integrals >0r
def intsPositive(model, reps):
        m = eval(model)
        for i in range(len(m)):
                if not integrate(m[i][0], m[i][1:]).subs(dict(reps, **r1)) > 0:
                        print('Integral ' + str(i+1) + ' in ' + model + ' is negative')

# Are the individual averaged integrals between 0 and  2r
def intsRange(model, reps):
        m = eval(model)
        for i in range(len(m)):
                if not 0 <= (integrate(m[i][0], m[i][1:])/(m[i][3]-m[i][2])).subs(dict(reps, **r1)) <=
                    2:
                        print('Integral ' + str(i+1) + ' in ' + model + ' has averaged integral outside
                            0<p<2r')

# Are the bounds the correct way around
def checkBounds(model, reps):
        m = eval(model)
        for i in range(len(m)):
                if not (m[i][3]-m[i][2]).subs(reps) > 0:
                        print('Bounds ' + str(i+1) + ' in ' + model + ' has lower bounds bigger than
                            upper bounds')

# create latex strings with the 1) the integral equation that defines it and 2)  the final calculated
    model.
# There's some if statements to split longer equations on two lines and get +s in the right place.
def parseLaTeX(prof):
        m = eval( 'm' + prof[1:] )

        f = open('/home/tim/Dropbox/liz-paper/lucasMoorcroftManuscript/supplementary-material/latexFiles
            /'+prof+'.tex', 'w')
        f.write('\\begin{align}\n    \\bar{p}_{\\text{\\tiny{' + prof[1:] + '}}} =&\\frac{1}{\\pi} \\left
            (\;\;')
        for i in range(len(m)):
    # Roughly try and prevent expressions beginning with minus signs.
    if latex(m[i][2])[0]=='-':
      o1 = 'rev-lex'
    else:
      o1 = 'lex'
```

```python
 77        if latex(m[i][3])[0]=='-':
 78          o2 = 'rev-lex'
 79        else:
 80          o2 = 'lex'
 81
 82        if latex(m[i][0])[0]=='-':
 83          o3 = 'rev-lex'
 84        else:
 85          o3 = 'lex'
 86
 87        if latex(m[i][1])[0]=='-':
 88          o4 = 'rev-lex'
 89        else:
 90          o4 = 'lex'
 91
 92                  f.write('\int\limits_{'+latex(m[i][2], order=o1)+'}^{'+latex(m[i][3], order=o2)+'}'+
 93                      latex(m[i][0], order=o3)+'\;\mathrm{d}' +latex(m[i][1], order=o4))
 93            if len(m)>3 and i==(len(m)/2)-1:
 94                  f.write( '\\right.\\notag\\\\\n &\left.' )
 95            if i<len(m)-1:
 96                  f.write('+')
 97        f.write('\\right)\label{' + prof + 'Def}\\\\\n    ')
 98        f.write('\\bar{p}_{\\text{\\tiny{' + prof[1:] + '}}}  =& ' + latex(eval(prof)) + '\label{' +
 99            prof + 'Sln}\n\\end{align}')
 99        f.close()
100
101
102 # Apply all checks.
103 def allChecks(prof):
104        model = 'm' + prof[1:]
105        reps = eval('rep' + prof[1:])
106        conds = 'cond' + prof[1:]
107        confirmReplacements(conds, reps)
108        profileRange(prof, reps)
109        intsPositive(model, reps)
110        intsRange(model, reps)
111        checkBounds(model, reps)
112
113 #######################################################
114 ### Define and solve all models                     ###
115 #######################################################
116
117 # NE1 animal: a = 2*pi.  sensor: t > pi, a > 3pi - t  #
118
119 mNE1 = [ [2*r,                 x1, pi/2, t/2         ],
120          [r + r*cos(x1 - t/2), x1, t/2,  pi          ],
121          [r + r*cos(x1 + t/2), x1, pi,    2*pi-t/2   ],
122          [2*r,                 x1, 2*pi-t/2, 3*pi/2 ] ]
123
124 # Replacement values in range
125 repNE1 = {t:3*pi/2, a:2*pi}
126
127 # Define conditions for model
128 condNE1 = [pi <= t, a >= 3*pi - t]
129
130 # Calculate model, run checks, write output.
131 pNE1 = calcModel(mNE1)
132 allChecks('pNE1')
133 parseLaTeX('pNE1')
134
135
136 # NE2 animal: a > pi.  sensor: t > pi Condition: a < 3pi - t, a > 4pi - 2t  #
137
138 mNE2 = [ [2*r,                 x1, pi/2, t/2           ],
139          [r + r*cos(x1 - t/2), x1, t/2,  5*pi/2 - t/2 - a/2 ],
140          [r + r*cos(x1 + t/2), x1, 5*pi/2 - t/2 - a/2,   2*pi-t/2 ],
141          [2*r,                 x1, 2*pi-t/2, 3*pi/2 ] ]
142
143 # Replacement values in range
144 repNE2 = {t:5*pi/3, a:4*pi/3-0.1}
145
146 # Define conditions for model
147 condNE2 = [pi <= t, a >= pi, a <= 3*pi - t, a >= 4*pi - 2*t]
148
149 # Calculate model, run checks, write output.
150 pNE2 = calcModel(mNE2)
151 allChecks('pNE2')
152 parseLaTeX('pNE2')
153
154
155 # NE3 animal: a > pi.  sensor: t > pi Condition: a < 4pi - 2t  #
156
157 mNE3 = [ [2*r,                 x1, pi/2, t/2           ],
158          [r + r*cos(x1 - t/2), x1, t/2,  t/2 + pi/2          ],
159          [r                  , x1, t/2 + pi/2,   5*pi/2 - t/2 - a/2 ],
160          [r + r*cos(x1 + t/2), x1, 5*pi/2 - t/2 - a/2,   2*pi-t/2 ],
161          [2*r,                 x1, 2*pi-t/2, 3*pi/2 ] ]
```

```
162
163 # Replacement values in range
164 repNE3 = {t:5*pi/4-0.1, a:3*pi/2}
165
166 # Define conditions for model
167 condNE3 = [pi <= t, a >= pi, a <= 4*pi - 2*t]
168
169 # Calculate model, run checks, write output.
170 pNE3 = calcModel(mNE3)
171 allChecks('pNE3')
172 parseLaTeX('pNE3')
173
174
175 # NW1 animal: a = 2*pi.   sensor:  pi/2 <= t <= pi       #
176
177 mNW1 = [ [2*r*sin(t/2)*sin(x2), x2, t/2,       pi/2      ],
178          [r - r*cos(x4 - t),    x4, 0,         t - pi/2 ],
179          [r,                    x4, t - pi/2, pi/2      ],
180          [r - r*cos(x4),        x4, pi/2,      t        ],
181          [2*r*sin(t/2)*sin(x2), x2, t/2,       pi/2     ] ]
182
183 # Replacement values in range
184 repNW1 = {t:3*pi/4}
185
186 # Define conditions for model
187 condNW1 = [pi/2 <= t, t <= pi]
188
189 # Calculate model, run checks, write output.
190 pNW1 = calcModel(mNW1)
191 allChecks('pNW1')
192 parseLaTeX('pNW1')
193
194
195
196
197 # NW2 animal: a > pi.  Sensor: pi/2 <= t <= pi. Condition: a > 2pi - t          #
198
199 mNW2 = [ [2*r*sin(t/2)*sin(x2), x2, t/2,           pi/2        ],
200          [r - r*cos(x4 - t),    x4, 0,             t - pi/2    ],
201          [r,                    x4, t - pi/2,      3*pi/2 - a/2],
202          [r - r*cos(x4),        x4, 3*pi/2 - a/2, t           ],
203          [2*r*sin(t/2)*sin(x2), x2, t/2,           pi/2        ] ]
204
205
206 repNW2 = {t:3*pi/4, a:15*pi/8} # Replacement values in range
207
208 # Define conditions for model
209 condNW2 = [a > pi, pi/2 <= t, t <= pi, a >= 3*pi - 2*t]
210
211 # Calculate model, run checks, write output.
212 pNW2 = calcModel(mNW2)
213 allChecks('pNW2')
214 parseLaTeX('pNW2')
215
216
217
218 # NW3 animal: a > pi.  Sensor: pi/2 <= t <= pi. Cond: 2pi - t < a < 3pi - 2t    #
219
220 mNW3 = [ [2*r*sin(t/2)*sin(x2), x2, t/2,                 pi/2             ],
221          [r - r*cos(x4 - t),    x4, 0,                   t - pi/2         ],
222          [r,                    x4, t - pi/2,            t                ],
223          [r*cos(x2 - t/2),      x2, t/2,                 3*pi/2 - a/2 - t/2],
224          [2*r*sin(t/2)*sin(x2), x2, 3*pi/2 - a/2 - t/2, pi/2             ] ]
225
226
227 repNW3 = {t:5*pi/8, a:6*pi/4} # Replacement values in range
228
229 # Define conditions for model
230 condNW3 = [a > pi, pi/2 <= t, t <= pi, 2*pi - t <= a, a <= 3*pi - 2*t]
231
232 # Calculate model, run checks, write output.
233 pNW3 = calcModel(mNW3)
234 allChecks('pNW3')
235 parseLaTeX('pNW3')
236
237
238
239 # NW4 animal:  a > pi.  Sensor: pi/2 <= t <= pi. Condition: a <= 2pi - t       #
240
241 mNW4 = [ [2*r*sin(t/2)*sin(x2), x2, t/2, pi/2],
242          [r - r*cos(x4 - t),    x4, 0, t - pi/2],
243          [r,                    x4, t - pi/2, t],
244          [r*cos(x2 - t/2),      x2, t/2, a/2 + t/2 - pi/2] ]
245
246 repNW4 = {t:3*pi/4, a:9*pi/8} # Replacement values in range
247
248 # Define conditions for model
```

```
249  condNW4 = [a > pi,  pi/2 <= t, t <= pi, a <= 2*pi - t]
250
251  # Calculate model, run checks, write output.
252  pNW4 = calcModel(mNW4)
253  allChecks('pNW4')
254  parseLaTeX('pNW4')
255
256
257  # REM animal: a=2pi.  Sensor: t <= pi/2.                                    #
258
259  mREM = [ [2*r*sin(t/2)*sin(x2), x2, pi/2 - t/2, pi/2],
260           [r*sin(x3),            x3, t,          pi/2],
261           [r,                    x4, 0*t,           t],
262           [r*sin(x3),            x3, t,          pi/2],
263           [2*r*sin(t/2)*sin(x2), x2, pi/2 - t/2, pi/2] ]
264
265
266  repREM = {t:3*pi/8, a:2*pi} # Replacement values in range
267
268  # Define conditions for model
269  condREM = [ t <= pi/2  ]
270
271  # Calculate model, run checks, write output.
272  pREM = calcModel(mREM)
273  allChecks('pREM')
274  parseLaTeX('pREM')
275
276
277
278  # NW5 animal: a>pi.  Sensor: t <= pi/2. Condition: 2*pi - t < a            #
279
280
281  mNW5 = [ [2*r*sin(t/2)*sin(x2), x2, pi/2 - t/2, pi/2],
282           [r*sin(x3),            x3, t,          pi/2],
283           [r,                    x4, 0,             t],
284           [r*sin(x3),            x3, t,          pi/2],
285           [r*cos(x2 - t/2),      x2, pi/2 - t/2, 3*pi/2 - t/2 - a/2],
286           [2*r*sin(t/2)*sin(x2), x2, 3*pi/2 - t/2 - a/2, pi/2] ]
287
288
289  repNW5 = {t:3*pi/8, a:29*pi/16} # Replacement values in range
290
291  # Define conditions for model
292  condNW5 = [a >= pi, t <= pi/2, 2*pi - t <= a  ]
293
294  # Calculate model, run checks, write output.
295  pNW5 = calcModel(mNW5)
296  allChecks('pNW5')
297  parseLaTeX('pNW5')
298
299
300  # NW6 animal: a>pi.  Sensor: t <= pi/2. Condition:  2*pi - 2*t <= a <= 2*pi - t  #
301
302
303  mNW6 = [ [2*r*sin(t/2)*sin(x2), x2, pi/2 - t/2, pi/2],
304           [r*sin(x3),            x3, t,          pi/2],
305           [r,                    x4, 0,             t],
306           [r*sin(x3),            x3, t,          pi/2],
307           [r*cos(x2 - t/2),      x2, pi/2 - t/2, a/2 + t/2 - pi/2] ]
308
309  repNW6 = {t:3*pi/8, a:3*pi/2} # Replacement values in range
310
311  # Define conditions for model
312  condNW6 = [a >= pi, t <= pi/2, 2*pi - 2*t <= a, a <= 2*pi - t]
313
314  # Calculate model, run checks, write output.
315  pNW6 = calcModel(mNW6)
316  allChecks('pNW6')
317  parseLaTeX('pNW6')
318
319
320
321  # NW7 animal: a>pi.  Sensor: t <= pi/2. Condition: a <= 2pi - 2t  #
322
323
324  mNW7 = [ [2*r*sin(t/2)*sin(x2), x2, pi/2 - t/2, pi/2],
325           [r*sin(x3),            x3, t,          pi/2],
326           [r,                    x4, 0,         t   ],
327           [r*sin(x3),            x3, pi - a/2,  pi/2] ]
328
329
330  repNW7 = {t:pi/9, a:10*pi/9} # Replacement values in range
331
332  # Define conditions for model
333  condNW7 = [t <= pi/2, a >= pi, a <= 2*pi - 2*t]
334
335  # Calculate model, run checks, write output.
```

```
336  pNW7 = calcModel(mNW7)
337  allChecks('pNW7')
338  parseLaTeX('pNW7')
339
340
341
342  # SE1 animal: a <= pi.  Sensor: t =2pi.              #
343
344  mSE1 = [ [ 2*r*sin(a/2),x1, pi/2, 3*pi/2        ],
345            ]
346
347
348  repSE1 = {a:pi/4} # Replacement values in range
349
350  # Define conditions for model
351  condSE1 = [a <= pi]
352
353  # Calculate model, run checks, write output.
354  pSE1 = calcModel(mSE1)
355  allChecks('pSE1')
356  parseLaTeX('pSE1')
357
358
359
360
361  # SE2 animal: a <= pi.  Sensor: t > pi. Condition: a > 2pi - t, a > 4pi - 2t    #
362
363  mSE2 = [ [ 2*r*sin(a/2),                      x1, pi/2,             t/2 + pi/2 - a/2       ],
364           [ r*sin(a/2) + r*cos(x1 - t/2),      x1, t/2 + pi/2 - a/2,  5*pi/2 - a/2 - t/2 ],
365           [ 2*r*sin(a/2),                      x1, 5*pi/2 - a/2 - t/2, 3*pi/2]   ]
366
367
368  repSE2 = {t:19*pi/10, a:pi/2} # Replacement values in range
369
370  # Define conditions for model
371  condSE2 = [a <= pi, t >= pi,  a >= 4*pi - 2*t]
372
373  # Calculate model, run checks, write output.
374  pSE2 = calcModel(mSE2)
375  allChecks('pSE2')
376  parseLaTeX('pSE2')
377
378
379  # SE3 animal: a <= pi.  Sensor: t > pi. Condition: 2pi - t < a < 4pi - 2t #
380
381  mSE3 = [ [ 2*r*sin(a/2),                      x1, pi/2,             t/2 + pi/2 - a/2  ],
382           [ r*sin(a/2) + r*cos(x1 - t/2),      x1, t/2 + pi/2 - a/2, t/2 + pi/2        ],
383           [ r*sin(a/2),                        x1, t/2 + pi/2,        5*pi/2 - a/2 - t/2],
384           [ 2*r*sin(a/2),                      x1, 5*pi/2 - a/2 - t/2, 3*pi/2           ] ]
385
386  repSE3 = {t:3*pi/2 + 0.1, a:pi/2} # Replacement values in range
387
388  # Define conditions for model
389  condSE3 = [a <= pi, t >= pi,  a >= 2*pi - t, a <= 4*pi - 2*t]
390
391  # Calculate model, run checks, write output.
392  pSE3 = calcModel(mSE3)
393  allChecks('pSE3')
394  parseLaTeX('pSE3')
395
396
397  # SE4 animal: a <= pi.  Sensor: t > pi. Condition: a <= 4*pi - 2*t and a < 2*pi - t #
398
399
400  mSE4 = [ [ 2*r*sin(a/2),                      x1, pi/2,             t/2 + pi/2 - a/2  ],
401           [ r*sin(a/2) + r*cos(x1 - t/2),      x1, t/2 + pi/2 - a/2, t/2 + pi/2        ],
402           [ r*sin(a/2),                        x1, t/2 + pi/2,        t/2 + pi/2 + a/2  ] ]
403
404
405  repSE4 = {t:3*pi/2, a:pi/3} # Replacement values in range
406
407
408  # Define conditions for model
409  condSE4 = [a <= pi, t >= pi/2, a <= 4*pi - 2*t , a <= 2*pi - t]
410
411  # Calculate model, run checks, write output.
412  pSE4 = calcModel(mSE4)
413  allChecks('pSE4')
414  parseLaTeX('pSE4')
415
416
417  # SW1 animal: a <= pi.  Sensor: pi/2 <= t <= pi. Condition: a >= t and a/2 >= t - pi/2 #
418
419  mSW1 =  [ [2*r*sin(t/2)*sin(x2),            x2, pi/2 - a/2 + t/2, pi/2            ],
420            [r*sin(a/2) - r*cos(x2 + t/2),      x2, t/2,               pi/2 - a/2 + t/2],
421            [r*sin(a/2) - r*cos(x4 - t),        x4, 0,                 t - pi/2        ],
422            [r*sin(a/2),                        x4, t-pi/2,            t - pi/2 + a/2  ] ]
```

```
423
424
425  repSW1 = {t:5*pi/8, a:6*pi/8} # Replacement values in range
426
427  # Define conditions for model
428  condSW1 = [a <= pi, pi/2 <= t, t <= pi, a >= t, a/2 >= t - pi/2]
429
430  # Calculate model, run checks, write output.
431  pSW1 = calcModel(mSW1)
432  allChecks('pSW1')
433  parseLaTeX('pSW1')
434
435
436  # SW2 animal: a <= pi.  Sensor: pi/2 <= t <= pi. Condition: a <= t and a/2 >= t- pi/2 #
437
438  mSW2 =  [ [2*r*sin(a/2),                    x2, pi/2 + a/2 - t/2, pi/2             ],
439          [r*sin(a/2) - r*cos(x2 + t/2), x2, t/2,              pi/2 + a/2 - t/2],
440          [r*sin(a/2) - r*cos(x4 - t),   x4, 0*t,              t - pi/2         ],
441          [r*sin(a/2),                    x4, t - pi/2,         t - pi/2 + a/2 ] ]
442
443
444  repSW2 = {t:7*pi/8, a:7*pi/8-0.1} # Replacement values in range
445
446  # Define conditions for model
447  condSW2 = [a <= pi, pi/2 <= t, t <= pi, a/2 <= t/2, a/2 >= t - pi/2]
448
449  # Calculate model, run checks, write output.
450  pSW2 = calcModel(mSW2)
451  allChecks('pSW2')
452  parseLaTeX('pSW2')
453
454
455
456  # SW3 animal: a <= pi.  Sensor: pi/2 <= t <= pi. Condition: a <= t and a/2 <= t- pi/2 #
457
458  mSW3 =  [ [2*r*sin(a/2),                    x2, t/2,             pi/2 - t/2        ],
459          [2*r*sin(a/2),                    x4, 0,               t - pi/2 - a/2 ],
460          [r*sin(a/2) - r*cos(x4 - t),   x4, t - pi/2 - a/2, t - pi/2          ],
461          [r*sin(a/2),                    x4, t - pi/2,        t - pi/2 + a/2 ] ]
462
463
464  repSW3 = {t:7*pi/8, a:2*pi/8} # Replacement values in range
465
466  # Define conditions for model
467  condSW3 = [a <= pi, pi/2 <= t, t <= pi, a/2 <= t/2, a/2 <= t - pi/2]
468
469  # Calculate model, run checks, write output.
470  pSW3 = calcModel(mSW3)
471  allChecks('pSW3')
472  parseLaTeX('pSW3')
473
474
475  # SW4 animal: a <= pi.  Sensor: t <= pi/2. Condition: a > pi - 2t &  a <= t       #
476
477  mSW4 = [ [2*r*sin(a/2),                x2, pi/2 - t/2 + a/2, pi/2            ],
478         [r*sin(a/2) - r*cos(x2 + t/2), x2, pi/2 - t/2,      pi/2 - t/2 + a/2],
479         [r*sin(a/2),                x3, t,               pi/2            ],
480         [r*sin(a/2),                x4, 0,               a/2 + t - pi/2  ] ]
481
482  repSW4 = {t:pi/2-0.1, a:pi/4} # Replacement values in range
483
484  # Define conditions for model
485  condSW4 = [a <= pi,  t <= pi/2,  a >= pi - 2*t,  a <= t]
486
487  # Calculate model, run checks, write output.
488  pSW4 = calcModel(mSW4)
489  allChecks('pSW4')
490  parseLaTeX('pSW4')
491
492
493  # SW5 animal: a <= pi.  Sensor: t <= pi/2. Condition: a > pi - 2t &  t <= a <= 2t    #
494
495  mSW5 = [ [2*r*sin(t/2)*sin(x2),          x2, pi/2 + t/2 - a/2, pi/2             ],
496         [r*sin(a/2) - r*cos(x2 + t/2), x2, pi/2 - t/2,      pi/2 + t/2 - a/2],
497         [r*sin(a/2),                x3, t,               pi/2         ],
498         [r*sin(a/2),                x4, 0,               a/2 + t -pi/2   ] ]
499
500
501  repSW5 = {t:pi/2-0.1, a:pi/2} # Replacement values in range
502
503  # define conditions for model
504  condSW5 = [a <= pi,  t <= pi/2,  a >= pi - 2*t,  t <= a, a <= 2*t]
505
506
507  # Calculate model, run checks, write output.
508  pSW5 = calcModel(mSW5)
509  allChecks('pSW5')
```

```
510  parseLaTeX('pSW5')
511
512
513  # SW6 animal: a <= pi.  Sensor: t <= pi/2. Condition: a > pi - 2t  &  a > 2t      #
514
515  mSW6 = [ [2*r*sin(t/2)*sin(x2), x2, pi/2 - t/2, pi/2                ],
516           [r*sin(x3),            x3, t,          a/2                 ],
517           [r*sin(a/2),           x3, a/2,        pi/2                ],
518           [r*sin(a/2),           x4, 0,          a/2 + t -pi/2   ] ]
519
520
521  repSW6 = {t:pi/4, a:3*pi/4} # Replacement values in range
522
523
524  # Define conditions for model
525  condSW6 = [a <= pi,  t <= pi/2,  a >= pi - 2*t,  a > 2*t]
526
527  # Calculate model, run checks, write output.
528  pSW6 = calcModel(mSW6)
529  allChecks('pSW6')
530  parseLaTeX('pSW6')
531
532
533  # SW7 animal: a <= pi.  Sensor: t <= pi/2. Condition: a <= pi - 2t & a <= t      #
534
535  mSW7 = [ [2*r*sin(a/2),                 x2, pi/2 - t/2 + a/2, pi/2            ],
536           [r*sin(a/2) - r*cos(x2 + t/2), x2, pi/2 - t/2,       pi/2 - t/2 + a/2],
537           [r*sin(a/2),                   x3, t,                t + a/2         ] ]
538
539
540  repSW7 = {t:2*pi/8, a:pi/8} # Replacement values in range
541
542  # Define conditions for model
543  condSW7 = [a <= pi, t <= pi/2, a <= pi - 2*t, a <= t]
544
545  # Calculate model, run checks, write output.
546  pSW7 = calcModel(mSW7)
547  allChecks('pSW7')
548  parseLaTeX('pSW7')
549
550
551  # SW8 animal: a <= pi.  Sensor: t <= pi/2. Condition: a <= pi - 2t & t <= a <= 2t   #
552
553  mSW8 = [ [2*r*sin(t/2)*sin(x2),         x2, pi/2 + t/2 - a/2, pi/2            ],
554           [r*sin(a/2) - r*cos(x2 + t/2), x2, pi/2 - t/2,       pi/2 + t/2 - a/2],
555           [r*sin(a/2),                   x3, t,                t + a/2         ] ]
556
557  repSW8 = {t:2*pi/8, a:pi/2-0.1} # Replacement values in range
558
559  # Define conditions for model
560  condSW8 = [a <= pi, t <= pi/2, a <= pi - 2*t, t <= a, a <= 2*t]
561
562  # Calculate model, run checks, write output.
563  pSW8 = calcModel(mSW8)
564  allChecks('pSW8')
565  parseLaTeX('pSW8')
566
567
568  # SW9 animal: a <= pi.  Sensor: t <= pi/2. Condition: a <= pi - 2t &  2t <= a      #
569
570  mSW9 = [ [2*r*sin(t/2)*sin(x2), x2, pi/2 - t/2, pi/2    ],
571           [r*sin(x3),           x3, t,          a/2     ],
572           [r*sin(a/2),          x3, a/2,        t + a/2 ] ]
573
574
575  repSW9 = {t:1*pi/8, a:pi/2} # Replacement values in range
576
577  # Define conditions for model
578  condSW9 = [a <= pi,  t <= pi/2,  a <= pi - 2*t,  2*t <= a]
579
580  # Calculate model, run checks, write output.
581  pSW9 = calcModel(mSW9)
582  allChecks('pSW9')
583  parseLaTeX('pSW9')
584
585
586  ####################
587  ## Run tests     ###
588  ####################
589
590  # create gas model object
591  gas = 2*r
592
593
594  # for each model run through every adjacent model.
595  # Contains duplicatea but better for avoiding missed comparisons.
596  # Also contains replacement t->a and a->t just in case.
```

```
allComps = [
['gas', 'pNE1', {t:2*pi}], ['gas', 'pSE1', {a:pi}],

['pNE1', 'gas', {t:2*pi}], ['pNE1', 'pNW1', {t:pi}],
['pNE1', 'pNE2',{a:3*pi-t}], ['pNE1', 'pNE2',{t:3*pi-a}],

['pNE2', 'pNE1',{a:3*pi-t}], ['pNE2', 'pNE1',{t:3*pi-a}],
['pNE2', 'pNE3',{a:4*pi-2*t}], ['pNE2', 'pNE3',{t:2*pi-a/2}],
['pNE2', 'pSE2',{a:pi}],

['pNE3', 'pNE2',{a:4*pi-2*t}], ['pNE3', 'pNE2',{t:2*pi-a/2}],
['pNE3', 'pSE3',{a:pi}], ['pNE3', 'pNW2',{t:pi}],

['pNW1','pNE1', {t:pi}], ['pNW1','pNW2',{a:2*pi}],

['pNW2','pNE3',{t:pi}], ['pNW2','pNW3',{a:3*pi-2*t}],
['pNW2','pNW3',{t:3*pi/2-a/2}], ['pNW2','pNW1',{a:2*pi}],

['pNW3','pNW5',{t:pi/2}], ['pNW3','pNW4',{a:2*pi-t}],
['pNW3','pNW4',{t:2*pi-a}], ['pNW3','pNW2',{a:3*pi-2*t}],
['pNW3','pNW2',{t:3*pi/2-a/2}],

['pNW4','pNW6',{t:pi/2}], ['pNW4','pNW3',{t:2*pi-a}],
['pNW4','pNW3',{a:2*pi-t}], ['pNW4','pSW1',{a:pi}],

['pREM','pNW1', {t:pi/2}], ['pREM','pNW5',{a:2*pi}],

['pNW5','pREM',{a:2*pi}], ['pNW5','pNW6',{a:2*pi-t}],
['pNW5','pNW6',{t:2*pi-a}], ['pNW5','pNW3',{t:pi/2}],

['pNW6','pNW5',{a:2*pi-t}], ['pNW6','pNW5',{t:2*pi-a}],
['pNW6','pNW7',{t:pi-a/2}], ['pNW6','pNW7',{a:2*pi-2*t}],
['pNW5','pNW4',{t:pi/2}],

['pNW7','pNW6',{t:2*pi-2*a}], ['pNW7','pNW6',{a:2*pi-2*t}],
['pNW7','pSW6',{a:pi}],

['pSE1','pSE2',{t:2*pi}], ['pSE1','gas',{a:pi}],

['pSE2','pSE3',{t:2*pi-a/2}], ['pSE2','pSE3',{a:4*pi-2*t}],
['pSE2','pSE1',{t:2*pi}], ['pSE2','pNE2',{a:pi}],

['pSE3','pSE2',{a:4*pi-2*t}], ['pSE3','pSE2',{t:2*pi-a/2}],
['pSE3','pSE4',{a:2*pi-t}], ['pSE3','pSE4',{t:2*pi-a}],
['pSE3','pNE3',{a:pi}],

['pSE4','pSE3',{t:2*pi-a}], ['pSE4','pSE3',{a:2*pi-t}],
['pSE4','pSW3',{t:pi}],

['pSW1','pSW5',{t:pi/2}], ['pSW1','pSW2',{a:t}],
['pSW1','pSW2',{t:a}], ['pSW1','pNW4',{a:pi}],

['pSW2','pSW1',{a:t}], ['pSW2','pSW1',{t:a}],
['pSW2','pSW4',{t:pi/2}], ['pSW2','pSW3',{a:2*t-pi}],
['pSW2','pSW3',{t:a/2+pi/2}],

['pSW3','pSW2',{t:a/2+pi/2}], ['pSW3','pSW2',{a:2*t-pi}],
['pSW3','pSE4',{t:pi}],


['pSW4','pSW7',{a:pi-2*t}], ['pSW4','pSW7',{t:pi/2-a/2}],
['pSW4','pSW5',{t:a}], ['pSW4','pSW5',{a:t}],
['pSW4','pSW2',{t:pi/2}],

['pSW5','pSW4',{t:a}], ['pSW5','pSW4',{a:t}],
['pSW5','pSW8',{t:pi/2-a/2}], ['pSW5','pSW8',{a:pi-2*t}],
['pSW5','pSW6',{a:2*t}], ['pSW5','pSW6',{t:a/2}],
['pSW5','pSW1',{t:pi/2}],

['pSW6','pSW9',{t:pi/2-a/2}], ['pSW6','pSW9',{a:pi-2*t}],
['pSW6','pSW5',{a:2*t}], ['pSW6','pSW5',{t:a/2}],
['pSW6','pNW7',{a:pi}],


['pSW7','pSW8',{t:a}], ['pSW7','pSW8',{a:t}],
['pSW7','pSW4',{t:pi/2-a/2}], ['pSW7','pSW4',{a:pi-2*t}],

['pSW8','pSW7',{a:t}], ['pSW8','pSW7',{t:a}],
['pSW8','pSW9',{a:2*t}], ['pSW8','pSW9',{t:a/2}],
['pSW8','pSW5',{a:pi-2*t}], ['pSW8','pSW5',{t:pi/2-a/2}],

['pSW9','pSW8',{a:2*t}], ['pSW9','pSW8',{t:a/2}],
['pSW9','pSW6',{a:pi-2*t}], ['pSW9','pSW6',{t:pi/2-a/2}]
]
```

```python
684
685 # List of regions that touch a=0. Should equal 0 when a=0.
686 zeroRegions = ['pSW9', 'pSW8', 'pSW7', 'pSW4', 'pSW2', 'pSW3', 'pSE4', 'pSE3', 'pSE2', 'pSE1']
687
688 # Run through all the comparisons. Need simplify(). Even together() gives some false negatives.
689
690 checkFile = open('/home/tim/Dropbox/phd/Analysis/REM-chapter/checksFile.tex','w')
691
692 checkFile.write('All checks evaluated.\nTim Lucas - ' + str(datetime.now()) + '\n')
693 for i in range(len(allComps)):
694         if (eval(allComps[i][0]).subs(allComps[i][2]) - eval(allComps[i][1]).subs(allComps[i][2])).
                simplify() == 0:
695                 checkFile.write(str(i) + ': ' + allComps[i][0]+ ' and ' +allComps[i][1]+': OK\n')
696         else:
697                 checkFile.write(str(i) + ': ' + allComps[i][0]+ ' and ' +allComps[i][1]+': Incorrect\n')
698
699 for i in range(len(zeroRegions)):
700         if eval(zeroRegions[i]).subs({a:0}).simplify() == 0:
701                 checkFile.write(zeroRegions[i] + ' at a=0: OK\n')
702         else:
703                 checkFile.write(zeroRegions[i] + ' at a=0: Incorrect\n')
704
705 checkFile.close()
706
707
708 # And print to terminal
709 #for i in range(len(allComps)):
710 #        if not (eval(allComps[i][0]).subs(allComps[i][2]) - eval(allComps[i][1]).subs(allComps[i][2])).
        simplify() == 0:
711 #                print allComps[i][0] + ' and ' + allComps[i][1]+': Incorrect\n'
712
713
714 ######################################################################
715 ### Define a a function that calculates p bar answer.         ####
716 ######################################################################
717
718 def calcP(A, T, R):
719   assert (A <= 2*pi and A >= 0), "a is out of bounds. Should be in 0<a<2*pi"
720   assert (T <= 2*pi and T >= 0), "s is out of bounds. Should be in 0<s<2*pi"
721
722   if A > pi:
723     if A < 4*pi - 2* T:
724       p = pNW7.subs({a:A,  t:T, r:R}).n()
725     elif A <= 3*pi -  T:
726                      p = pNE2.subs({a:A,  t:T, r:R}).n()
727     else:
728                      p = pNE1.subs({a:A,  t:T, r:R}).n()
729   else:
730     if A < 4*pi - 2* T:
731                      p = pSE3.subs({a:A,  t:T, r:R}).n()
732     else:
733                      p = pSE2.subs({a:A,  t:T, r:R}).n()
734         return p
735
736
737 ##############################
738 ## Apply to entire grid   ###
739 ##############################
740
741 # How many values for each parameter
742 nParas = 100
743
744 # Make a vector for a and s. Make an empty nParas x nParas array.
745 # Calculated profile sizes will go in pArray
746 tVec = np.linspace(0, 2*pi, nParas)
747 aVec = np.linspace(0, 2*pi, nParas)
748 pArray = np.zeros((nParas,nParas))
749
750 # Calculate profile size for each combination of parameters
751 for i in range(nParas):
752         for j in range(nParas):
753                 pArray[i][j] = calcP(aVec[i], tVec[j], 1)
754
755 # Turn the array upside down so origin is at bottom left.
756 pImage = np.flipud(pArray)
757
758 # Plot and save.
759 pl.imshow(pImage, interpolation='none', cmap=pl.get_cmap('Blues') )
760 #pl.show()
761
762 pl.savefig('/home/tim/Dropbox/phd/Analysis/REM-chapter/imgs/profilesCalculated.png')
763
764
765
766 ##############################
767 #### Output R function.  ###
768 ##############################
```

```
769
770  # To reduce mistakes, output R function directly from python.
771  # However, the if statements, which correspond to the bounds of each model, are not automatic.
772
773  Rfunc = open('/home/tim/Dropbox/phd/Analysis/REM-chapter/supplementaryRscript.R', 'w')
774
775  Rfunc.write("""
776  # Functions to calculate density.
777  #
778  # Tim C.D. Lucas, Elizabeth Moorcroft, Robin Freeman, Marcus J. Rowcliffe, Kate E. Jones.
779  #
780  # calcDensity is the main function to calculate density.
781  # It takes parameters z, alpha, theta, r, animalSpeed, t
782  # z - The number of camera/acoustic counts or captures.
783  # alpha - Call width in radians.
784  # theta - Sensor width in radians.
785  # r - Sensor range in metres.
786  # animalSpeed - Average animal speed in metres per second.
787  # t - Length of survey in sensor seconds i.e. number of sensors x survey duration.
788  #
789  # calcAbundance calculates abundance rather than density and requires an extra parameter
790  # area - In metres squared. The size of the region being examined.
791
792
793  # Internal function to calculate profile width as described in the text
794  calcProfileWidth <- function(alpha, theta, r){
795          if(alpha > 2*pi | alpha < 0)
796      stop('alpha is out of bounds. alpha should be in interval 0<a<2*pi')
797          if(theta > 2*pi | theta < 0)
798      stop('theta is out of bounds. theta should be in interval 0<a<2*pi')
799
800    if(alpha > pi){
801           if(alpha < 4*pi - 2*theta){
802  """ +
803  '           p <- ' + str(pNW7) +
804  '\n               } else if(alpha <= 3*pi - theta){'
805  '\n                       p <- ' + str(pNE2) +
806  '\n               } else {'
807  '\n                       p <- ' + str(pNE1) +
808  '\n               }'
809  '\n         } else {'
810  '\n          if(alpha < 4*pi - 2*theta){'
811  '\n                       p <- ' + str(pSE3) +
812  '\n      } else {'
813  '\n                       p <- ' + str(pSE2) +
814  '\n               }'
815  '\n         }'
816  '\n       return(p)'
817  '\n}' +
818  """
819  # Calculate a population density. See above for units etc.
820  calcDensity <- function(z, alpha, theta, r, animalSpeed, t){
821          # Check the parameters are suitable.
822          if(z <= 0 | !is.numeric(z)) stop('Counts, z, must be a positive number.')
823          if(animalSpeed <= 0 | !is.numeric(animalSpeed)) stop('animalSpeed must be a positive number.')
824          if(t <= 0 | !is.numeric(t)) stop('Time, t, must be a positive number.')
825
826          # Calculate profile width, then density.
827          p <- calcProfileWidth(alpha, theta, r)
828          D <- z/{animalSpeed*t*p}
829          return(D)
830  }
831
832  # Calculate abundance rather than density.
833  calcAbundance <- function(z, alpha, theta, r, animalSpeed, t, area){
834          if(area <= 0 | !is.numeric(area)) stop('Area must be a positive number')
835          D <- calcDensity(z, alpha, theta, r, animalSpeed, t)
836          A <- D*area
837          return(A)
838  }
839  """
840  )
841
842  Rfunc.close()
```

REM–Analysis.py

S4. SUPPLEMENTARY SCRIPT: R IMPLEMENTATION OF MODELS

This is a simple implementation of the models derived in the paper in R (**?**). Once given the parameters θ and α it automatically selects the correct model to apply.

```r
# Functions to calculate density.
#
# Tim C.D. Lucas, Elizabeth Moorcroft, Robin Freeman, Marcus J. Rowcliffe, Kate E. Jones.
#
# calcDensity is the main function to calculate density.
# It takes parameters z, alpha, theta, r, animalSpeed, t
# z - The number of camera/acoustic counts or captures.
# alpha - Call width in radians.
# theta - Sensor width in radians.
# r - Sensor range in metres.
# animalSpeed - Average animal speed in metres per second.
# t - Length of survey in sensor seconds i.e. number of sensors x survey duration.
#
# calcAbundance calculates abundance rather than density and requires an extra parameter
# area - In metres squared. The size of the region being examined.


# Internal function to calculate profile width as described in the text
calcProfileWidth <- function(alpha, theta, r){
        if(alpha > 2*pi | alpha < 0)
    stop('alpha is out of bounds. alpha should be in interval 0<a<2*pi')
        if(theta > 2*pi | theta < 0)
    stop('theta is out of bounds. theta should be in interval 0<a<2*pi')

  if(alpha > pi){
         if(alpha < 4*pi - 2*theta){
           p <- r*(theta - cos(alpha/2) + 1)/pi
             } else if(alpha <= 3*pi - theta){
                     p <- r*(theta - cos(alpha/2) + cos(alpha/2 + theta))/pi
             } else {
                     p <- r*(theta + 2*sin(theta/2))/pi
             }
       } else {
         if(alpha < 4*pi - 2*theta){
                     p <- r*(theta*sin(alpha/2) - cos(alpha/2) + 1)/pi
    } else {
                     p <- r*(theta*sin(alpha/2) - cos(alpha/2) + cos(alpha/2 + theta))/pi
             }
        }
        return(p)
}
# Calculate a population density. See above for units etc.
calcDensity <- function(z, alpha, theta, r, animalSpeed, t){
        # Check the parameters are suitable.
        if(z <= 0 | !is.numeric(z)) stop('Counts, z, must be a positive number.')
        if(animalSpeed <= 0 | !is.numeric(animalSpeed)) stop('animalSpeed must be a positive number.')
        if(t <= 0 | !is.numeric(t)) stop('Time, t, must be a positive number.')

        # Calculate profile width, then density.
        p <- calcProfileWidth(alpha, theta, r)
        if(p <= 0) stop('Calculated profile width is 0. We would therefore expect 0 captures. If z is
            not zero, then the density is undefined.')
        D <- z/{animalSpeed*t*p}
        return(D)
}

# Calculate abundance rather than density.
calcAbundance <- function(z, alpha, theta, r, animalSpeed, t, area){
        if(area <= 0 | !is.numeric(area)) stop('Area must be a positive number')
        D <- calcDensity(z, alpha, theta, r, animalSpeed, t)
        A <- D*area
        return(A)
}
```

supplementaryRscript.R