SUPPLEMENTARY MATERIAL: A GENERALISATION OF IDEAL GAS MODELS FOR CAMERA TRAPS AND ACOUSTIC SENSORS

CONTENTS

1. SUPPLEMENTARY METHODS

1.1. **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 derivation of all models is included in the Python script S2.

1.2. **Gas model.** We assume that animals are in an homogeneous environment, and move in straight lines of random direction with velocity v. We allow that our sensor can detect animals at a distance r and that if an animal moves within this detection region they are detected with a probability of 1, independent of distance from the sensor while animals outside the region are never detected.

We then consider movement from the reference frame of the animals so that now, 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 study period we can estimate the number of animals it should encounter. We calculate this as the average width of the sensor region p multiplied by v. The average width of the profile is the integral of the profile width over a full circle, divided by 2π. We use x_i to denote the focal angle which is the angle we integrate over. The subscript i distinguishes different angles (see Figure S2) but here we use x_1. As all models are bilaterally symetric, we can integrate over a half circle, and divide by π.

$$p\text{Gas} = \frac{1}{\pi}\int_{\frac{\pi}{2}}^{\frac{3\pi}{2}} 2r \, \mathrm{d}x_1 \qquad \text{eqn S1}$$

$$p\text{Gas} = 2r \qquad \text{eqn S2}$$

The number of expected encounters, z, for a survey of duration t, with an animal density of D is then

$$z = 2rvtD. \qquad \text{eqn S3}$$

However, in practice we have the opposite situation. We know the number of encounters and want to estimate the density. We do this be simply rearranging to get

$$D = z/(2rvt). \qquad \text{eqn S4}$$

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

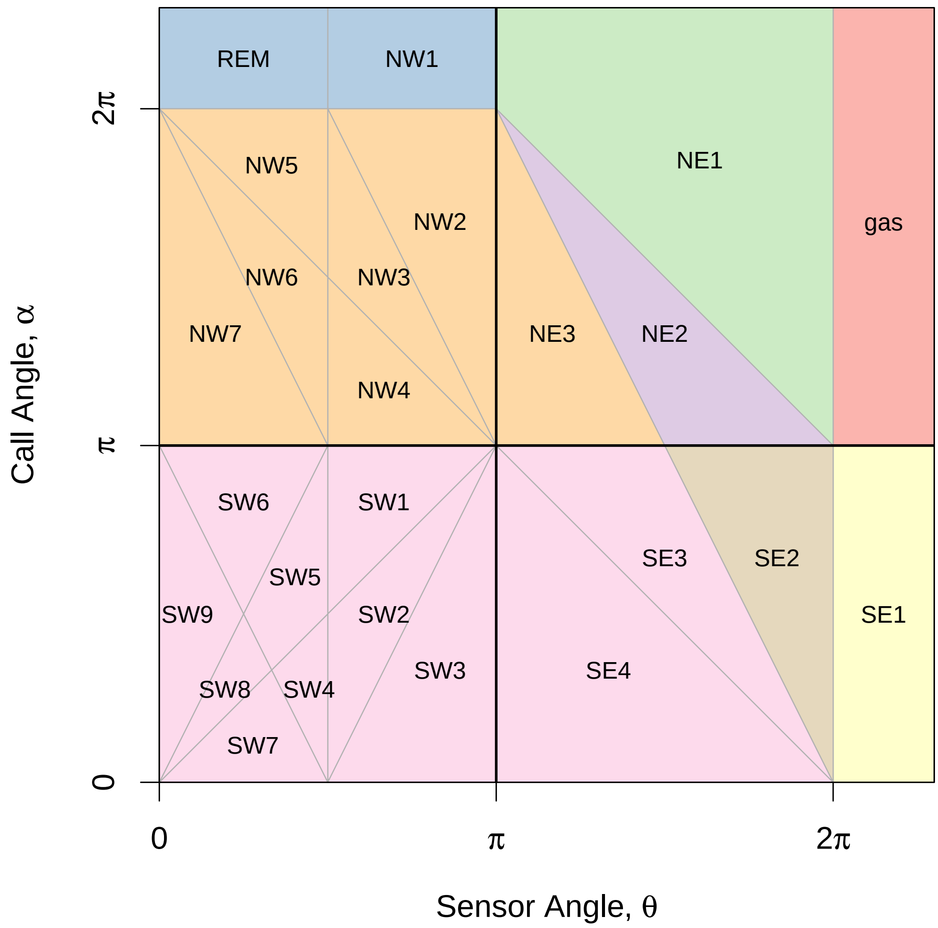


FIGURE S1. The location of each model in parameter space. Each named model must be derived separately. However, the results of the different models are often the same; areas coloured the same have the same result. Other than the gas model and th REM model, individual models are named after the compass 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 $\alpha = 2\pi$.

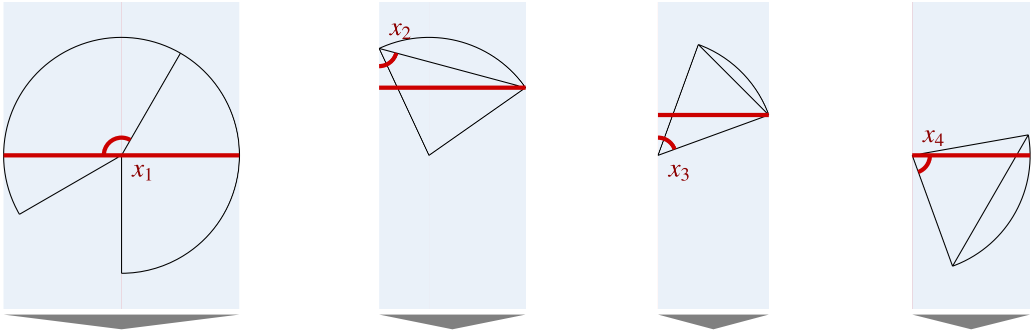
diameter that changes with approach angle. The rest of the derivation is just calculating this value for all values of α and θ. However, different regions of this two dimensional parameter space have noncontinuously different models, with different derivations. Therefore we have to identify the regions for which the derivation is the same, and then separately derive p for each region. The separate regions are shown in Figure S1.

1.3. **Model SE1.** SE1 is very similar to the gas model except that as $\alpha \leq \pi$ 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 (see Fig S3b).

$$\text{pSE1} = \frac{1}{\pi}\left(\int_{\frac{\pi}{2}}^{\frac{3\pi}{2}} 2r\sin\left(\frac{\alpha}{2}\right)\mathrm{d}x_1\right) \qquad \text{eqn S5}$$

$$\text{pSE1} = 2r\sin\left(\frac{\alpha}{2}\right) \qquad \text{eqn S6}$$

1.4. **Model NE.** For regions with profiles that are more complex than a circle we 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 for all approach angles by integrating across all 2π angles of approach and dividing by 2π.



(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 segment shaped detection region is shown in black. The width of this region is shown with a thick red line and a blue rectangle. The direction of animal movement is always downwards, as indicated by the grey arrow.

There are three regions within cell 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 regions. 1) The profile width starts, from $x_1 = \frac{\pi}{2}$ as $2r$. 2) At $x_1 = \theta/2$, the right hand side of the profile cannot be r wide as the corner of the 'blind spot' (see Fig. S3a) limits its size to being $r\cos(x_1 - \theta/2)$ wide (see Fig. S4a).

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. S4b). 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.

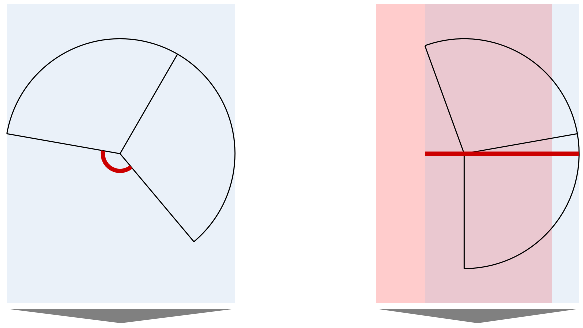
5) Finally, common to all three models, at $x_1 = 2\pi - \theta/2$ the profile becomes a full $2r$ once again.

1.4.1. *Model NE1.* Model NE1 exists within the area bounded by $\alpha \leq 2\pi$, $\theta \leq 2\pi$ and $\alpha \geq 3\pi - \theta$. It has four regions; 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

$$\begin{aligned} \text{pNE1} =& \frac{1}{\pi}\left(\int_{\frac{\pi}{2}}^{\frac{\theta}{2}} 2r \, \mathrm{d}x_1 + \int_{\frac{\theta}{2}}^{\pi} r + r\cos\left(-x_1 + \frac{\theta}{2}\right) \mathrm{d}x_1 \right. \\ & \left. + \int_{\pi}^{2\pi - \frac{\theta}{2}} r + r\cos\left(x_1 + \frac{\theta}{2}\right) \mathrm{d}x_1 + \int_{2\pi-\frac{\theta}{2}}^{\frac{3\pi}{2}} 2r \, \mathrm{d}x_1 \right) \end{aligned} \qquad \text{eqn S7}$$

$$\text{pNE1} = \frac{r}{\pi}\left(\theta + 2\sin\left(\frac{\theta}{2}\right)\right) \qquad \text{eqn S8}$$

1.4.2. *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.



(A) Blind spot (B) Directional Call

FIGURE S3. A) Shows the area referred to as the 'blind spot'. B) For directional calls, with $\alpha < \pi$, the width of the profile can be limited by the call angle or by the detector region. The detector width is shown in blue, while the call width is shown as a red rectangle. Only where the two overlap, giving a purple area, can an animal be detected. Here we would say the right side of the profile is limited by the sensor, while the left side of the profile is limited by the call angle. The terms in equations would reflect this by containing α if call limited and containing θ if detector limited.

$$\begin{aligned}\mathrm{pNE2} =& \frac{1}{\pi}\left(\int_{\frac{\pi}{2}}^{\frac{\theta}{2}} 2r \,\mathrm{d}x_1 + \int_{\frac{\theta}{2}}^{\frac{5\pi}{2}-\frac{\theta}{2}-\frac{\alpha}{2}} r + r\cos\left(-x_1 + \frac{\theta}{2}\right)\mathrm{d}x_1 \right.\\ &\left. + \int_{\frac{5\pi}{2}-\frac{\theta}{2}-\frac{\alpha}{2}}^{2\pi-\frac{\theta}{2}} r + r\cos\left(x_1 + \frac{\theta}{2}\right)\mathrm{d}x_1 + \int_{2\pi-\frac{\theta}{2}}^{\frac{3\pi}{2}} 2r\,\mathrm{d}x_1 \right) \qquad \text{eqn S9}\end{aligned}$$

$$\mathrm{pNE2} = \frac{r}{\pi}\left(\theta - \cos\left(\frac{\alpha}{2}\right) + \cos\left(\frac{\alpha}{2} + \theta\right)\right) \qquad \text{eqn S10}$$

1.4.3. *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).

$$\begin{aligned}\mathrm{pNE3} =& \frac{1}{\pi}\left(\int_{\frac{\pi}{2}}^{\frac{\theta}{2}} 2r \,\mathrm{d}x_1 + \int_{\frac{\theta}{2}}^{\frac{\pi}{2}+\frac{\theta}{2}} r + r\cos\left(-x_1 + \frac{\theta}{2}\right)\mathrm{d}x_1 \right.\\ &\left. + \int_{\frac{\pi}{2}+\frac{\theta}{2}}^{\frac{5\pi}{2}-\frac{\theta}{2}-\frac{\alpha}{2}} r\,\mathrm{d}x_1 + \int_{\frac{5\pi}{2}-\frac{\theta}{2}-\frac{\alpha}{2}}^{2\pi-\frac{\theta}{2}} r + r\cos\left(x_1 + \frac{\theta}{2}\right)\mathrm{d}x_1 + \int_{2\pi-\frac{\theta}{2}}^{\frac{3\pi}{2}} 2r\,\mathrm{d}x_1 \right) \qquad \text{eqn S11}\end{aligned}$$

$$\mathrm{pNE3} = \frac{r}{\pi}\left(\theta - \cos\left(\frac{\alpha}{2}\right) + 1\right) \qquad \text{eqn S12}$$

1.5. **Model p32.** Cell p32 contains three regions that differ in ways reminiscent of the models in NE. There are four possible profile widths. 1) As α is less than π the profile is smaller than $2r$, even when the sensor width is a full diameter. When this is the case, the profile width is instead $2r\sin(\alpha/2$. 2) Similar to NE, at a certain point the blind spot of the sensor area limits the profile width (see Fig. S5a). 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 $\theta/2 + \pi/2 + \alpha/2$ the profile width become 0 (see Fig. S5b). This inequality distinguishes between SE3 and SE4. The 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 $\alpha \leq 4\pi - 2\theta$.

1.5.1. *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.

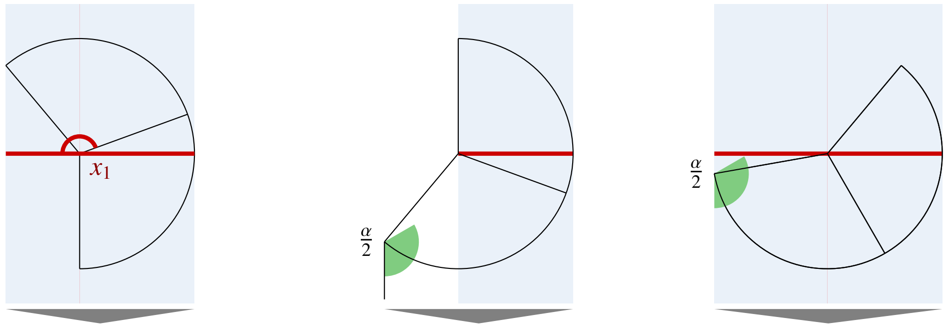


FIGURE S4. A) The second integral in NE with width $r + r\cos(x_1 - \theta/2)$ B) The third integral in NE3. The angle shown in red is $\alpha/2$. As it is small, animals to the right of the detector cannot be detected. C) After further rotation, $\alpha/2$ is now bigger than the angle shown and animals to the right of the detector can again be sensed.

$$\text{pSE2} = \frac{1}{\pi}\left(\int_{\frac{\pi}{2}}^{\frac{\pi}{2}+\frac{\theta}{2}-\frac{\alpha}{2}} 2r\sin\left(\frac{\alpha}{2}\right) \mathrm{d}x_1 + \int_{\frac{\pi}{2}+\frac{\theta}{2}-\frac{\alpha}{2}}^{\frac{5\pi}{2}-\frac{\theta}{2}-\frac{\alpha}{2}} r\cos\left(-x_1+\frac{\theta}{2}\right) + r\sin\left(\frac{\alpha}{2}\right)\mathrm{d}x_1 + \int_{\frac{5\pi}{2}-\frac{\theta}{2}-\frac{\alpha}{2}}^{\frac{3\pi}{2}} 2r\sin\left(\frac{\alpha}{2}\right)\mathrm{d}x_1 \right) \quad \text{eqn S13}$$

$$\text{pSE2} = \frac{r}{\pi}\left(\theta\sin\left(\frac{\alpha}{2}\right) - \cos\left(\frac{\alpha}{2}\right) + \cos\left(\frac{\alpha}{2}+\theta\right)\right) \quad \text{eqn S14}$$

1.5.2. *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.

$$\begin{aligned}\text{pSE3} = \frac{1}{\pi}\Bigg(&\int_{\frac{\pi}{2}}^{\frac{\pi}{2}+\frac{\theta}{2}-\frac{\alpha}{2}} 2r\sin\left(\frac{\alpha}{2}\right)\mathrm{d}x_1 + \int_{\frac{\pi}{2}+\frac{\theta}{2}-\frac{\alpha}{2}}^{\frac{\pi}{2}+\frac{\theta}{2}} r\cos\left(-x_1+\frac{\theta}{2}\right) + r\sin\left(\frac{\alpha}{2}\right)\mathrm{d}x_1 \\ &+ \int_{\frac{\pi}{2}+\frac{\theta}{2}}^{\frac{5\pi}{2}-\frac{\theta}{2}-\frac{\alpha}{2}} r\sin\left(\frac{\alpha}{2}\right)\mathrm{d}x_1 + \int_{\frac{5\pi}{2}-\frac{\theta}{2}-\frac{\alpha}{2}}^{\frac{3\pi}{2}} 2r\sin\left(\frac{\alpha}{2}\right)\mathrm{d}x_1\Bigg)\end{aligned} \quad \text{eqn S15}$$

$$\text{pSE3} = \frac{r}{\pi}\left(\theta\sin\left(\frac{\alpha}{2}\right) - \cos\left(\frac{\alpha}{2}\right) + 1\right) \quad \text{eqn S16}$$

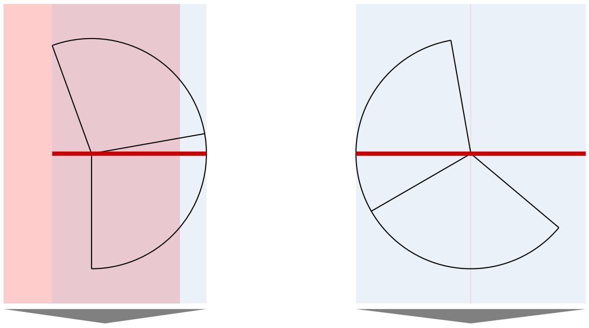
1.5.3. *Model SE4.* Finally SE4 is bounded by $\alpha \leq 4\pi - 2\theta$, $\alpha \leq \pi$ and $\theta \leq \pi$. It is the same as SE3 except that the profile becomes $2r$ rather than returning to $2r\sin(\alpha/2)$.

$$\text{pSE4} = \frac{1}{\pi}\left(\int_{\frac{\pi}{2}}^{\frac{\pi}{2}+\frac{\theta}{2}-\frac{\alpha}{2}} 2r\sin\left(\frac{\alpha}{2}\right)\mathrm{d}x_1 + \int_{\frac{\pi}{2}+\frac{\theta}{2}-\frac{\alpha}{2}}^{\frac{\pi}{2}+\frac{\theta}{2}} r\cos\left(-x_1+\frac{\theta}{2}\right) + r\sin\left(\frac{\alpha}{2}\right)\mathrm{d}x_1 + \int_{\frac{\pi}{2}+\frac{\theta}{2}}^{\frac{\pi}{2}+\frac{\theta}{2}+\frac{\alpha}{2}} r\sin\left(\frac{\alpha}{2}\right)\mathrm{d}x_1 \right) \quad \text{eqn S17}$$

$$\text{pSE4} = \frac{r}{\pi}\left(\theta\sin\left(\frac{\alpha}{2}\right) - \cos\left(\frac{\alpha}{2}\right) + 1\right) \quad \text{eqn S18}$$

1.6. **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 Rowcliffe *et al.* (2008) while x_4 is new. They are described in Fig. S2.

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 segment. During this region 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. S2b). 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. S6a). 3) At $x_4 = \theta - \pi/2$, the profile is perpendicular to the edge of the



(A) (B)

FIGURE S5. A) The third integral in p32. The right side of the profile is limited by the size of the sensor region (blue region) while the left side of the profile is limited by the size of the call angle (red region). The profile width is the purple region where these two overlap. B)

sensor area. Here, the right side of the profile is $0r$. 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. S6b). Therefore the width of the profile is $r - r\cos(x_4)$. 5) Finally, we enter the x_2 region, but from behind.

$$\begin{aligned} \text{pNW1} =& \frac{1}{\pi}\left(\int_{\frac{\theta}{2}}^{\frac{\pi}{2}} 2r\sin\left(\frac{\theta}{2}\right)\sin(x_2)\,\mathrm{d}x_2 + \int_{0}^{-\frac{\pi}{2}+\theta} r - r\cos(-x_4+\theta)\,\mathrm{d}x_4 \right. \\ & \left. + \int_{-\frac{\pi}{2}+\theta}^{\frac{\pi}{2}} r\,\mathrm{d}x_4 + \int_{\frac{\pi}{2}}^{\theta} r - r\cos(x_4)\,\mathrm{d}x_4 + \int_{\frac{\theta}{2}}^{\frac{\pi}{2}} 2r\sin\left(\frac{\theta}{2}\right)\sin(x_2)\,\mathrm{d}x_2 \right) \end{aligned} \qquad \text{eqn S19}$$

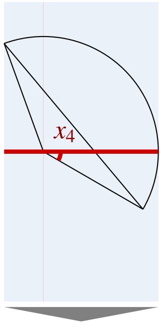
$$\text{pNW1} = \frac{r}{\pi}(\theta + 2) \qquad \text{eqn S20}$$

1.7. **Model NW2–4.** The models in cell 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. When approaching the sensor from behind, there is a period where the profile is r wide as in NW1. 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. 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.

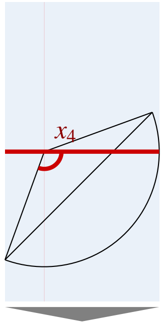
As $\alpha \leq 2\pi$ it is possible that when the angle of approach is from directly behind the sensor the animal will not be detected at all. This is the case if $\alpha/2 \leq \pi - \theta/2$ as shown in Fig. S7a. This inequality defines the boundary between NW3 and NW4.

1.7.1. *Model NW2.* NW2 is bounded by $\alpha \geq 3\pi - 2\theta$, $\alpha \leq 2\pi$ and $\theta \leq \pi$.

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 $x_4 = \theta$.



(A)



(B)

FIGURE S6. A) and B) The third and fourth profiles of NW1. The left side of of both profiles is of width r while the right side is $-r\cos(x_4-\theta)$ and $-r\cos(x_4)$ respectively.

$$\text{pNW2} = \frac{1}{\pi}\left(\int_{\frac{\theta}{2}}^{\frac{\pi}{2}} 2r\sin\left(\frac{\theta}{2}\right)\sin(x_2)\,\mathrm{d}x_2 + \int_{0}^{-\frac{\pi}{2}+\theta} r - r\cos(-x_4+\theta)\,\mathrm{d}x_4 + \int_{-\frac{\pi}{2}+\theta}^{\frac{3\pi}{2}-\frac{\alpha}{2}} r\,\mathrm{d}x_4 + \int_{\frac{3\pi}{2}-\frac{\alpha}{2}}^{\theta} r - r\cos(x_4)\,\mathrm{d}x_4 + \int_{\frac{\theta}{2}}^{\frac{\pi}{2}} 2r\sin\left(\frac{\theta}{2}\right)\sin(x_2)\,\mathrm{d}x_2 \right) \quad \text{eqn S21}$$

$$\text{pNW2} = \frac{r}{\pi}\left(\theta - \cos\left(\frac{\alpha}{2}\right) + 1\right) \quad \text{eqn S22}$$

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

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 $2r\sin(x_2)\sin(\theta/2)$.

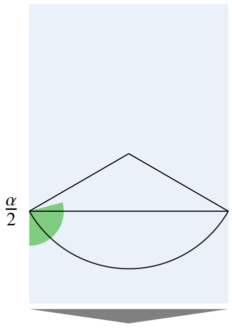
$$\text{pNW3} = \frac{1}{\pi}\left(\int_{\frac{\theta}{2}}^{\frac{\pi}{2}} 2r\sin\left(\frac{\theta}{2}\right)\sin(x_2)\,\mathrm{d}x_2 + \int_{0}^{-\frac{\pi}{2}+\theta} r - r\cos(-x_4+\theta)\,\mathrm{d}x_4 + \int_{-\frac{\pi}{2}+\theta}^{\theta} r\,\mathrm{d}x_4 + \int_{\frac{\theta}{2}}^{\frac{3\pi}{2}-\frac{\theta}{2}-\frac{\alpha}{2}} r\cos\left(-x_2+\frac{\theta}{2}\right)\mathrm{d}x_2 + \int_{\frac{3\pi}{2}-\frac{\theta}{2}-\frac{\alpha}{2}}^{\frac{\pi}{2}} 2r\sin\left(\frac{\theta}{2}\right)\sin(x_2)\,\mathrm{d}x_2 \right) \quad \text{eqn S23}$$

$$\text{pNW3} = \frac{r}{\pi}\left(\theta - \cos\left(\frac{\alpha}{2}\right) + 1\right) \quad \text{eqn S24}$$

1.7.3. *Model NW4.* Finally, NW4 is bounded by $\alpha \le \pi$, $\theta \ge \pi/2$ and $\alpha \le 3\pi - 2\theta$. NW4 is the same as NW3 except that the final profile width is zero and this profile is reached at $\alpha/2 + \theta/2 - \pi/2$.

$$\text{pNW4} = \frac{1}{\pi}\left(\int_{\frac{\theta}{2}}^{\frac{\pi}{2}} 2r\sin\left(\frac{\theta}{2}\right)\sin(x_2)\,\mathrm{d}x_2 + \int_{0}^{-\frac{\pi}{2}+\theta} r - r\cos(-x_4+\theta)\,\mathrm{d}x_4 + \int_{-\frac{\pi}{2}+\theta}^{\theta} r\,\mathrm{d}x_4 + \int_{\frac{\theta}{2}}^{-\frac{\pi}{2}+\frac{\theta}{2}+\frac{\alpha}{2}} r\cos\left(-x_2+\frac{\theta}{2}\right)\mathrm{d}x_2 \right) \quad \text{eqn S25}$$

$$\text{pNW4} = \frac{r}{\pi}\left(\theta - \cos\left(\frac{\alpha}{2}\right) + 1\right) \quad \text{eqn S26}$$



(A)

FIGURE S7. A) If $\alpha/2$, shown in green, 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.

1.8. **Model p33.** The models in p33 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 eons only once each.

There are five potential profile sizes. At the beginning of x_2, with an approach direction directly towards the sensor, the factor that limits the width of the profile can either be 1) the sensor width, in which case the profile width is $2r\sin(\theta/2)\sin(x_2)$, or 2) the call width, in which case the profile width is instead $2r\sin(\alpha/2)$ (see Figure S8)

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 $x_2 = \pi/2 + \alpha/2 - \theta/2$.

In the x_4 region the left side of the profile is always $r\sin(\alpha/2)$ while the right side is either 4) 0, giving a profile of $r\sin(\alpha/2)$, or 5) limited by the sensor giving a profile of size $r\sin(\alpha/2) - r\cos(x_4 - \theta)$.

1.8.1. *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 p33 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 $r\sin(\alpha/2) - r\cos(x_2 + \theta/2)$.

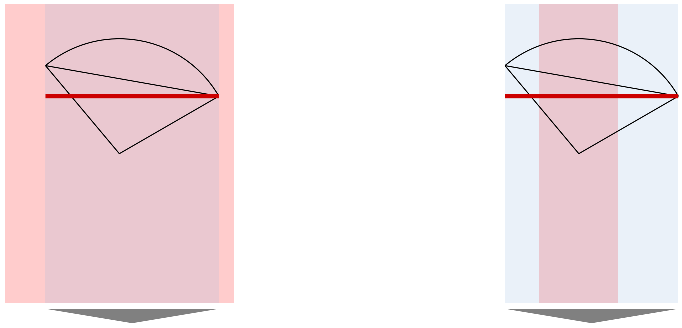
As we enter the x_4 region, the profile remains limited by the call on the left and by the sensor on the right, giving a profile width of $r\sin(\alpha/2) - r\cos(x_4 - \theta)$. Finally, at $x_4 = \theta - \pi/2$ the right side of the profile becomes zero and the profile is width is $r\sin(\alpha/2)$.

$$\begin{aligned}\text{pSW1} =& \frac{1}{\pi}\Bigg(\int_{\frac{\pi}{2}+\frac{\theta}{2}-\frac{\alpha}{2}}^{\frac{\pi}{2}} 2r\sin\left(\frac{\theta}{2}\right)\sin(x_2)\,\mathrm{d}x_2 + \int_{\frac{\theta}{2}}^{\frac{\pi}{2}+\frac{\theta}{2}-\frac{\alpha}{2}} -r\cos\left(x_2+\frac{\theta}{2}\right) + r\sin\left(\frac{\alpha}{2}\right)\mathrm{d}x_2 \\ &+ \int_{0}^{-\frac{\pi}{2}+\theta} -r\cos(-x_4+\theta) + r\sin\left(\frac{\alpha}{2}\right)\mathrm{d}x_4 + \int_{-\frac{\pi}{2}+\theta}^{-\frac{\pi}{2}+\theta+\frac{\alpha}{2}} r\sin\left(\frac{\alpha}{2}\right)\mathrm{d}x_4\Bigg) \end{aligned} \qquad \text{eqn S27}$$

$$\text{pSW1} = \frac{r}{\pi}\left(\theta\sin\left(\frac{\alpha}{2}\right) - \cos\left(\frac{\alpha}{2}\right) + 1\right) \qquad \text{eqn S28}$$

1.8.2. *Model SW2.* SW2 is bounded by $\theta \geq \pi/2$, $\alpha \leq \theta$ and $\alpha \geq 2\theta - \pi$.

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 $x_2 = \pi/2 - \alpha/2 + \theta/2$.



(A) (B)

FIGURE S8. A) As $\alpha > \theta$ the profile width (purple) is limited by the sensor region, not the call angle (red). The profile width is $2r\sin\left(\frac{\theta}{2}\right)\sin(x_2)$. B) As $\alpha < \theta$ the profile width is limited by the call angle rather than the sensor region (blue). The profile width is $2r\sin\left(\frac{\alpha}{2}\right)$

$$\begin{aligned}\text{pSW2} =& \frac{1}{\pi}\left(\int_{\frac{\pi}{2}-\frac{\theta}{2}+\frac{\alpha}{2}}^{\frac{\pi}{2}} 2r\sin\left(\frac{\alpha}{2}\right)\mathrm{d}x_2 + \int_{\frac{\theta}{2}}^{\frac{\pi}{2}-\frac{\theta}{2}+\frac{\alpha}{2}} -r\cos\left(x_2+\frac{\theta}{2}\right) + r\sin\left(\frac{\alpha}{2}\right)\mathrm{d}x_2 \right.\\ &\left. + \int_{0}^{-\frac{\pi}{2}+\theta} -r\cos\left(-x_4+\theta\right) + r\sin\left(\frac{\alpha}{2}\right)\mathrm{d}x_4 + \int_{-\frac{\pi}{2}+\theta}^{-\frac{\pi}{2}+\theta+\frac{\alpha}{2}} r\sin\left(\frac{\alpha}{2}\right)\mathrm{d}x_4 \right) \qquad \text{eqn S29}\end{aligned}$$

$$\text{pSW2} = \frac{r}{\pi}\left(\theta\sin\left(\frac{\alpha}{2}\right) - \cos\left(\frac{\alpha}{2}\right) + 1\right) \qquad \text{eqn S30}$$

1.8.3. *Model SW3.* SW3 is bounded by $\alpha \le 2\theta - \pi$ and $\theta \le \pi$.

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.

$$\begin{aligned}\text{pSW3} =& \frac{1}{\pi}\left(\int_{\frac{\theta}{2}}^{\frac{\pi}{2}} 2r\sin\left(\frac{\alpha}{2}\right)\mathrm{d}x_2 + \int_{0}^{-\frac{\pi}{2}+\theta-\frac{\alpha}{2}} 2r\sin\left(\frac{\alpha}{2}\right)\mathrm{d}x_4 \right.\\ &\left. + \int_{-\frac{\pi}{2}+\theta-\frac{\alpha}{2}}^{-\frac{\pi}{2}+\theta} -r\cos\left(-x_4+\theta\right) + r\sin\left(\frac{\alpha}{2}\right)\mathrm{d}x_4 + \int_{-\frac{\pi}{2}+\theta}^{-\frac{\pi}{2}+\theta+\frac{\alpha}{2}} r\sin\left(\frac{\alpha}{2}\right)\mathrm{d}x_4 \right) \qquad \text{eqn S31}\end{aligned}$$

$$\text{pSW3} = \frac{r}{\pi}\left(\theta\sin\left(\frac{\alpha}{2}\right) - \cos\left(\frac{\alpha}{2}\right) + 1\right) \qquad \text{eqn S32}$$

1.9. **Model REM.** REM is the model from (Rowcliffe *et al.*, 2008). It has $\alpha = 2\pi$ and $\theta \le \pi/2$. 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.

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

$$\begin{aligned}\text{pREM} =& \frac{1}{\pi}\left(\int_{\frac{\pi}{2}-\frac{\theta}{2}}^{\frac{\pi}{2}} 2r\sin\left(\frac{\theta}{2}\right)\sin\left(x_2\right)\,\mathrm{d}x_2 + \int_{\theta}^{\frac{\pi}{2}} r\sin\left(x_3\right)\,\mathrm{d}x_3 \right.\\ &\left. + \int_{0}^{\theta} r\,\mathrm{d}x_4 + \int_{\theta}^{\frac{\pi}{2}} r\sin\left(x_3\right)\,\mathrm{d}x_3 + \int_{\frac{\pi}{2}-\frac{\theta}{2}}^{\frac{\pi}{2}} 2r\sin\left(\frac{\theta}{2}\right)\sin\left(x_2\right)\,\mathrm{d}x_2 \right) \qquad \text{eqn S33}\end{aligned}$$

$$\text{pREM} = \frac{r}{\pi}\left(\theta + 2\right) \qquad \text{eqn S34}$$

1.10. **Model NW5–7.** In the models in NW5–7, the sensor has $\theta \leq pi/2$ as in REM. As $\alpha \geq \pi/2$ a lot of the profiles are similar to REM. Specifically, the first three profiles are always the same as the first three profiles of 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 it's call is wide enough that it can be detected once it has passed the sensor.

The second x_3 profile is always the same width as in REM. This is because there is no detection region to one side of the sensor so this side is unaffected by call width, while the width of the other side of the profile is unaffected by α as when $\alpha > \pi$ the profile width will never be limited by α. 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.

As the focal angle moves from x_3 to x_2 at $x_3 = \theta$, we can see that if $\alpha \geq 2\pi + 2\theta$, then the x_2 region is reached before the animal become undetectable. When this second x_2 region is reached, the profile starts with width $r\cos(x_2 - \theta/2)$ as at the beginning of the x_2 profile as only animals approaching to the left of the sensor are detectable. The sensor is directly behind the right side of the profile.

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

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

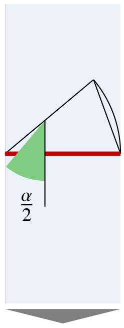
It is the same as REM except that it includes the extra profile in x_2 (the fifth integral) where only animals approaching to the left of the profile are detected.

$$\begin{aligned}\text{pNW5} =& \frac{1}{\pi}\left(\int_{\frac{\pi}{2}-\frac{\theta}{2}}^{\frac{\pi}{2}} 2r\sin\left(\frac{\theta}{2}\right)\sin\left(x_2\right)\,\mathrm{d}x_2 + \int_{\theta}^{\frac{\pi}{2}} r\sin\left(x_3\right)\,\mathrm{d}x_3 + \int_{0}^{\theta} r\,\mathrm{d}x_4 \right.\\ &\left. + \int_{\theta}^{\frac{\pi}{2}} r\sin\left(x_3\right)\,\mathrm{d}x_3 + \int_{\frac{\pi}{2}-\frac{\theta}{2}}^{\frac{3\pi}{2}-\frac{\theta}{2}-\frac{\alpha}{2}} r\cos\left(-x_2 + \frac{\theta}{2}\right)\mathrm{d}x_2 + \int_{\frac{3\pi}{2}-\frac{\theta}{2}-\frac{\alpha}{2}}^{\frac{\pi}{2}} 2r\sin\left(\frac{\theta}{2}\right)\sin\left(x_2\right)\,\mathrm{d}x_2 \right) \qquad \text{eqn S35}\end{aligned}$$

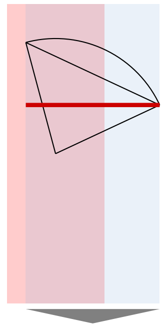
$$\text{pNW5} = \frac{r}{\pi}\left(\theta - \cos\left(\frac{\alpha}{2}\right) + 1\right) \qquad \text{eqn S36}$$

1.10.2. *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.



(A) $x_4 = 0$



(B)

FIGURE S9. A) At $x_4 = 0$, if $\alpha < \pi - 2\theta$ then $\alpha/2$ is too small for an animal to be detected at all during the x_4 profile. B) The left of the profile is limited by the call width, not the sensor (blue). On the right, the profile is limited by the sensor and not the call (red). Overall the profile width is $r\sin(\alpha/2) - r\cos(x_2 + \theta/2)$.

$$\begin{aligned}\text{pNW6} =& \frac{1}{\pi}\left(\int_{\frac{\pi}{2}-\frac{\theta}{2}}^{\frac{\pi}{2}} 2r\sin\left(\frac{\theta}{2}\right)\sin(x_2)\,\mathrm{d}x_2 + \int_{\theta}^{\frac{\pi}{2}} r\sin(x_3)\,\mathrm{d}x_3 \right.\\ &\left. + \int_0^{\theta} r\,\mathrm{d}x_4 + \int_{\theta}^{\frac{\pi}{2}} r\sin(x_3)\,\mathrm{d}x_3 + \int_{\frac{\pi}{2}-\frac{\theta}{2}}^{-\frac{\pi}{2}+\frac{\theta}{2}+\frac{\alpha}{2}} r\cos\left(-x_2+\frac{\theta}{2}\right)\mathrm{d}x_2 \right) \end{aligned} \qquad \text{eqn S37}$$

$$\text{pNW6} = \frac{r}{\pi}\left(\theta - \cos\left(\frac{\alpha}{2}\right) + 1\right) \qquad \text{eqn S38}$$

1.10.3. *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.

$$\begin{aligned}\text{pNW7} =& \frac{1}{\pi}\left(\int_{\frac{\pi}{2}-\frac{\theta}{2}}^{\frac{\pi}{2}} 2r\sin\left(\frac{\theta}{2}\right)\sin(x_2)\,\mathrm{d}x_2 + \int_{\theta}^{\frac{\pi}{2}} r\sin(x_3)\,\mathrm{d}x_3 \right.\\ &\left. + \int_0^{\theta} r\,\mathrm{d}x_4 + \int_{\pi-\frac{\alpha}{2}}^{\frac{\pi}{2}} r\sin(x_3)\,\mathrm{d}x_3 \right) \end{aligned} \qquad \text{eqn S39}$$

$$\text{pNW7} = \frac{r}{\pi}\left(\theta - \cos\left(\frac{\alpha}{2}\right) + 1\right) \qquad \text{eqn S40}$$

1.11. **Model SW4–9.** Cell SW4–9 is split into six models rather than three like most of the other cells. 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.

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 S9a where $\alpha/2 < \pi/2 - \theta$ which simplifies to give the previous inequality.

Models with $\alpha \leq \theta$ are limited by α in the first, x_2 region (see Figure S8), rather than being limited by θ. Therefore this first profile is of width $2r\sin(\alpha/2)$ rather than $2r\sin(\theta/2)\sin(x_2)$.

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 S9b). This gives a width of $r\sin(\alpha/2) - r\cos(x_2 + \theta/2)$. This profile does not occur in models with $\alpha \geq 2\theta$.

1.11.1. *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.

$$\text{pSW4} = \frac{1}{\pi}\left(\int_{\frac{\pi}{2}-\frac{\theta}{2}+\frac{\alpha}{2}}^{\frac{\pi}{2}} 2r\sin\left(\frac{\alpha}{2}\right)\mathrm{d}x_2 + \int_{\frac{\pi}{2}-\frac{\theta}{2}}^{\frac{\pi}{2}-\frac{\theta}{2}+\frac{\alpha}{2}} -r\cos\left(x_2+\frac{\theta}{2}\right)+r\sin\left(\frac{\alpha}{2}\right)\mathrm{d}x_2 \right.$$
$$\left. + \int_{\theta}^{\frac{\pi}{2}} r\sin\left(\frac{\alpha}{2}\right)\mathrm{d}x_3 + \int_{0}^{-\frac{\pi}{2}+\theta+\frac{\alpha}{2}} r\sin\left(\frac{\alpha}{2}\right)\mathrm{d}x_4 \right) \qquad \text{eqn S41}$$

$$\text{pSW4} = \frac{r}{\pi}\left(\theta\sin\left(\frac{\alpha}{2}\right)-\cos\left(\frac{\alpha}{2}\right)+1\right) \qquad \text{eqn S42}$$

1.11.2. *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.

$$\text{pSW5} = \frac{1}{\pi}\left(\int_{\frac{\pi}{2}+\frac{\theta}{2}-\frac{\alpha}{2}}^{\frac{\pi}{2}} 2r\sin\left(\frac{\theta}{2}\right)\sin(x_2)\,\mathrm{d}x_2 + \int_{\frac{\pi}{2}-\frac{\theta}{2}}^{\frac{\pi}{2}+\frac{\theta}{2}-\frac{\alpha}{2}} -r\cos\left(x_2+\frac{\theta}{2}\right)+r\sin\left(\frac{\alpha}{2}\right)\mathrm{d}x_2 \right.$$
$$\left. + \int_{\theta}^{\frac{\pi}{2}} r\sin\left(\frac{\alpha}{2}\right)\mathrm{d}x_3 + \int_{0}^{-\frac{\pi}{2}+\theta+\frac{\alpha}{2}} r\sin\left(\frac{\alpha}{2}\right)\mathrm{d}x_4 \right) \qquad \text{eqn S43}$$

$$\text{pSW5} = \frac{r}{\pi}\left(\theta\sin\left(\frac{\alpha}{2}\right)-\cos\left(\frac{\alpha}{2}\right)+1\right) \qquad \text{eqn S44}$$

1.11.3. *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.

$$\text{pSW6} = \frac{1}{\pi}\left(\int_{\frac{\pi}{2}-\frac{\theta}{2}}^{\frac{\pi}{2}} 2r\sin\left(\frac{\theta}{2}\right)\sin(x_2)\,\mathrm{d}x_2 + \int_{\theta}^{\frac{\alpha}{2}} r\sin(x_3)\,\mathrm{d}x_3 \right.$$
$$\left. + \int_{\frac{\alpha}{2}}^{\frac{\pi}{2}} r\sin\left(\frac{\alpha}{2}\right)\mathrm{d}x_3 + \int_{0}^{-\frac{\pi}{2}+\theta+\frac{\alpha}{2}} r\sin\left(\frac{\alpha}{2}\right)\mathrm{d}x_4 \right) \qquad \text{eqn S45}$$

$$\text{pSW6} = \frac{r}{\pi}\left(\theta\sin\left(\frac{\alpha}{2}\right)-\cos\left(\frac{\alpha}{2}\right)+1\right) \qquad \text{eqn S46}$$

1.11.4. *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.

$$\text{pSW7} = \frac{1}{\pi}\left(\int_{\frac{\pi}{2}-\frac{\theta}{2}+\frac{\alpha}{2}}^{\frac{\pi}{2}} 2r\sin\left(\frac{\alpha}{2}\right)\mathrm{d}x_2 + \int_{\frac{\pi}{2}-\frac{\theta}{2}}^{\frac{\pi}{2}-\frac{\theta}{2}+\frac{\alpha}{2}} -r\cos\left(x_2+\frac{\theta}{2}\right)+r\sin\left(\frac{\alpha}{2}\right)\mathrm{d}x_2 + \int_{\theta}^{\theta+\frac{\alpha}{2}} r\sin\left(\frac{\alpha}{2}\right)\mathrm{d}x_3 \right) \qquad \text{eqn S47}$$

$$\text{pSW7} = \frac{r}{\pi}\left(\theta\sin\left(\frac{\alpha}{2}\right)-\cos\left(\frac{\alpha}{2}\right)+1\right) \qquad \text{eqn S48}$$

1.11.5. *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.

$$\text{pSW8} = \frac{1}{\pi}\left(\int_{\frac{\pi}{2}+\frac{\theta}{2}-\frac{\alpha}{2}}^{\frac{\pi}{2}} 2r\sin\left(\frac{\theta}{2}\right)\sin(x_2)\,\mathrm{d}x_2 + \int_{\frac{\pi}{2}-\frac{\theta}{2}}^{\frac{\pi}{2}+\frac{\theta}{2}-\frac{\alpha}{2}} -r\cos\left(x_2+\frac{\theta}{2}\right)+r\sin\left(\frac{\alpha}{2}\right)\mathrm{d}x_2 + \int_{\theta}^{\theta+\frac{\alpha}{2}} r\sin\left(\frac{\alpha}{2}\right)\mathrm{d}x_3 \right) \qquad \text{eqn S49}$$

$$\text{pSW8} = \frac{r}{\pi}\left(\theta\sin\left(\frac{\alpha}{2}\right)-\cos\left(\frac{\alpha}{2}\right)+1\right) \qquad \text{eqn S50}$$

1.11.6. *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.

$$\mathrm{pSW9} = \frac{1}{\pi}\left(\int_{\frac{\pi}{2}-\frac{\theta}{2}}^{\frac{\pi}{2}} 2r\sin\left(\frac{\theta}{2}\right)\sin(x_2)\,\mathrm{d}x_2 + \int_{\theta}^{\frac{\alpha}{2}} r\sin(x_3)\,\mathrm{d}x_3 + \int_{\frac{\alpha}{2}}^{\theta+\frac{\alpha}{2}} r\sin\left(\frac{\alpha}{2}\right)\mathrm{d}x_3 \right) \qquad \text{eqn S51}$$

$$\mathrm{pSW9} = \frac{r}{\pi}\left(\theta\sin\left(\frac{\alpha}{2}\right) - \cos\left(\frac{\alpha}{2}\right) + 1\right) \qquad \text{eqn S52}$$

2. Supplementary Script: Symbolic algebra Python Script

This script uses the SymPy package SymPy Development Team (2014), a computer algebra system to calculate the equations for p in the various models and to perform unit checks on the results.

```
"""
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
import Image as Im


# 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 to neaten up later code.

# 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    \\mathrm{' + prof + '} =&\\frac{1}{\pi} \left(\;\;')
        for i in range(len(m)):
                f.write('\int\limits_{'+latex(m[i][2], order='rev-lex')+'}^{'+latex(m[i][3], order='rev-lex')+'}'+latex(m[i][0], order='rev-lex')+'\;\mathrm{d}' +latex(m[i][1]))
                if len(m)>3 and i==(len(m)/2)-1:
                        f.write( '\\right.\\notag\\\\\n &\left.' )
                if i<len(m)-1:
                        f.write('+')
```

```
        f.write('\\right)\label{' + prof + 'Def}\\\\\n    ')
        f.write('\\mathrm{' + prof + '} =& ' + latex(eval(prof)) + '\label{' + prof + 'Sln}\n\\end{align}')
        f.close()


# Apply all checks.
def allChecks(prof):
        model = 'm' + prof[1:]
        reps = eval('rep' + prof[1:])
        conds = 'cond' + prof[1:]
        confirmReplacements(conds, reps)
        profileRange(prof, reps)
        intsPositive(model, reps)
        intsRange(model, reps)
        checkBounds(model, reps)

###########################################################
# NE1 animal: a = 2*pi.  sensor: t > pi, a > 3pi - t  #
###########################################################



mNE1 = [ [2*r,                 x1, pi/2, t/2        ],
         [r + r*cos(x1 - t/2), x1, t/2,  pi         ],
         [r + r*cos(x1 + t/2), x1, pi,   2*pi-t/2   ],
         [2*r,                 x1, 2*pi-t/2, 3*pi/2 ] ]

# Replacement values in range
repNE1 = {t:3*pi/2, a:2*pi}

# Define conditions for model
condNE1 = [pi <= t, a >= 3*pi - t]

# Calculate model, run checks, write output.
pNE1 = calcModel(mNE1)
allChecks('pNE1')
parseLaTeX('pNE1')

#################################################################################
# NE2 animal: a > pi.  sensor: t > pi Condition: a < 3pi - t, a > 4pi - 2t  #
#################################################################################



mNE2 = [ [2*r,                 x1, pi/2, t/2        ],
         [r + r*cos(x1 - t/2), x1, t/2,  5*pi/2 - t/2 - a/2 ],
         [r + r*cos(x1 + t/2), x1, 5*pi/2 - t/2 - a/2,   2*pi-t/2 ],
         [2*r,                 x1, 2*pi-t/2, 3*pi/2 ] ]


# Replacement values in range
repNE2 = {t:5*pi/3, a:4*pi/3-0.1}

# Define conditions for model
condNE2 = [pi <= t, a >= pi, a <= 3*pi - t, a >= 4*pi - 2*t]

# Calculate model, run checks, write output.
pNE2 = calcModel(mNE2)
allChecks('pNE2')
parseLaTeX('pNE2')


####################################################################
# NE3 animal: a > pi.  sensor: t > pi Condition: a < 4pi - 2t  #
####################################################################



mNE3 = [ [2*r,                 x1, pi/2, t/2        ],
         [r + r*cos(x1 - t/2), x1, t/2,  t/2 + pi/2        ],
         [r                  , x1, t/2 + pi/2,   5*pi/2 - t/2 - a/2 ],
         [r + r*cos(x1 + t/2), x1, 5*pi/2 - t/2 - a/2,   2*pi-t/2 ],
         [2*r,                 x1, 2*pi-t/2, 3*pi/2 ] ]


# Replacement values in range
repNE3 = {t:5*pi/4-0.1, a:3*pi/2}

# Define conditions for model
condNE3 = [pi <= t, a >= pi, a <= 4*pi - 2*t]

# Calculate model, run checks, write output.
pNE3 = calcModel(mNE3)
allChecks('pNE3')
parseLaTeX('pNE3')
```

```
##############################################################
# NW1 animal: a = 2*pi.   sensor:  pi/2 <= t <= pi       #
##############################################################

mNW1 = [ [2*r*sin(t/2)*sin(x2), x2, t/2,      pi/2     ],
         [r - r*cos(x4 - t),    x4, 0,        t - pi/2 ],
         [r,                    x4, t - pi/2, pi/2     ],
         [r - r*cos(x4),        x4, pi/2,     t        ],
         [2*r*sin(t/2)*sin(x2), x2, t/2,      pi/2     ] ]

# Replacement values in range
repNW1 = {t:3*pi/4}

# Define conditions for model
condNW1 = [pi/2 <= t, t <= pi]

# Calculate model, run checks, write output.
pNW1 = calcModel(mNW1)
allChecks('pNW1')
parseLaTeX('pNW1')



#########################################################################################
# NW2 animal: a > pi.  Sensor: pi/2 <= t <= pi. Condition: a > 2pi - t                #
#########################################################################################


mNW2 = [ [2*r*sin(t/2)*sin(x2), x2, t/2,           pi/2        ],
          [r - r*cos(x4 - t),    x4, 0,             t - pi/2    ],
          [r,                    x4, t - pi/2,      3*pi/2 - a/2],
          [r - r*cos(x4),        x4, 3*pi/2 - a/2, t            ],
          [2*r*sin(t/2)*sin(x2), x2, t/2,           pi/2        ] ]


repNW2 = {t:3*pi/4, a:15*pi/8} # Replacement values in range

# Define conditions for model
condNW2 = [a > pi, pi/2 <= t, t <= pi, a >= 3*pi - 2*t]

# Calculate model, run checks, write output.
pNW2 = calcModel(mNW2)
allChecks('pNW2')
parseLaTeX('pNW2')


#########################################################################################
# NW3 animal: a > pi.  Sensor: pi/2 <= t <= pi. Cond: 2pi - t < a < 3pi - 2t     #
#########################################################################################


mNW3 = [ [2*r*sin(t/2)*sin(x2), x2, t/2,                 pi/2               ],
          [r - r*cos(x4 - t),    x4, 0,                   t - pi/2           ],
          [r,                    x4, t - pi/2,            t                  ],
          [r*cos(x2 - t/2),      x2, t/2,                 3*pi/2 - a/2 - t/2],
          [2*r*sin(t/2)*sin(x2), x2, 3*pi/2 - a/2 - t/2, pi/2               ] ]


repNW3 = {t:5*pi/8, a:6*pi/4} # Replacement values in range

# Define conditions for model
condNW3 = [a > pi, pi/2 <= t, t <= pi, 2*pi - t <= a, a <= 3*pi - 2*t]

# Calculate model, run checks, write output.
pNW3 = calcModel(mNW3)
allChecks('pNW3')
parseLaTeX('pNW3')


#########################################################################################
# NW4 animal:  a > pi.  Sensor: pi/2 <= t <= pi. Condition: a <= 2pi - t        #
#########################################################################################

mNW4 = [ [2*r*sin(t/2)*sin(x2), x2, t/2, pi/2],
          [r - r*cos(x4 - t),    x4, 0, t - pi/2],
          [r,                    x4, t - pi/2, t],
          [r*cos(x2 - t/2),      x2, t/2, a/2 + t/2 - pi/2] ]

repNW4 = {t:3*pi/4, a:9*pi/8} # Replacement values in range

# Define conditions for model
condNW4 = [a > pi,  pi/2 <= t, t <= pi, a <= 2*pi - t]

# Calculate model, run checks, write output.
```

```
pNW4 = calcModel(mNW4)
allChecks('pNW4')
parseLaTeX('pNW4')

###########################################################################
# REM animal: a=2pi.  Sensor: t <= pi/2.                                  #
###########################################################################

mREM = [ [2*r*sin(t/2)*sin(x2), x2, pi/2 - t/2, pi/2],
         [r*sin(x3),            x3, t,          pi/2],
         [r,                    x4, 0*t,          t],
         [r*sin(x3),            x3, t,          pi/2],
         [2*r*sin(t/2)*sin(x2), x2, pi/2 - t/2, pi/2] ]


repREM = {t:3*pi/8, a:2*pi} # Replacement values in range

# Define conditions for model
condREM = [ t <= pi/2  ]

# Calculate model, run checks, write output.
pREM = calcModel(mREM)
allChecks('pREM')
parseLaTeX('pREM')


###########################################################################
# NW5 animal: a>pi.  Sensor: t <= pi/2. Condition: 2*pi - t < a           #
###########################################################################

mNW5 = [ [2*r*sin(t/2)*sin(x2), x2, pi/2 - t/2, pi/2],
         [r*sin(x3),            x3, t,          pi/2],
         [r,                    x4, 0,          t],
         [r*sin(x3),            x3, t,          pi/2],
         [r*cos(x2 - t/2),      x2, pi/2 - t/2, 3*pi/2 - t/2 - a/2],
         [2*r*sin(t/2)*sin(x2), x2, 3*pi/2 - t/2 - a/2, pi/2] ]


repNW5 = {t:3*pi/8, a:29*pi/16} # Replacement values in range

# Define conditions for model
condNW5 = [a >= pi, t <= pi/2, 2*pi - t <= a  ]

# Calculate model, run checks, write output.
pNW5 = calcModel(mNW5)
allChecks('pNW5')
parseLaTeX('pNW5')

################################################################################
# NW6 animal: a>pi.  Sensor: t <= pi/2. Condition:  2*pi - 2*t <= a <= 2*pi - t  #
################################################################################

mNW6 = [ [2*r*sin(t/2)*sin(x2), x2, pi/2 - t/2, pi/2],
         [r*sin(x3),            x3, t,          pi/2],
         [r,                    x4, 0,          t],
         [r*sin(x3),            x3, t,          pi/2],
         [r*cos(x2 - t/2),      x2, pi/2 - t/2, a/2 + t/2 - pi/2] ]

repNW6 = {t:3*pi/8, a:3*pi/2} # Replacement values in range

# Define conditions for model
condNW6 = [a >= pi, t <= pi/2, 2*pi - 2*t <= a, a <= 2*pi - t]

# Calculate model, run checks, write output.
pNW6 = calcModel(mNW6)
allChecks('pNW6')
parseLaTeX('pNW6')


#################################################################
# NW7 animal: a>pi.  Sensor: t <= pi/2. Condition: a <= 2pi - 2t   #
#################################################################

mNW7 = [ [2*r*sin(t/2)*sin(x2), x2, pi/2 - t/2, pi/2],
         [r*sin(x3),            x3, t,          pi/2],
         [r,                    x4, 0,          t   ],
         [r*sin(x3),            x3, pi - a/2,   pi/2] ]


repNW7 = {t:pi/9, a:10*pi/9} # Replacement values in range

# Define conditions for model
condNW7 = [t <= pi/2, a >= pi, a <= 2*pi - 2*t]

# Calculate model, run checks, write output.
pNW7 = calcModel(mNW7)
allChecks('pNW7')
```

```
parseLaTeX('pNW7')

#####################################################
# SE1 animal: a <= pi.  Sensor: t =2pi.             #
#####################################################


mSE1 = [ [ 2*r*sin(a/2),x1, pi/2, 3*pi/2       ],
         ]


repSE1 = {a:pi/4} # Replacement values in range

# Define conditions for model
condSE1 = [a <= pi]

# Calculate model, run checks, write output.
pSE1 = calcModel(mSE1)
allChecks('pSE1')
parseLaTeX('pSE1')



##################################################################################
# SE2 animal: a <= pi.  Sensor: t > pi. Condition: a > 2pi - t, a > 4pi - 2t    #
##################################################################################


mSE2 = [ [ 2*r*sin(a/2),                        x1, pi/2,                t/2 + pi/2 - a/2       ],
         [ r*sin(a/2) + r*cos(x1 - t/2),        x1, t/2 + pi/2 - a/2,    5*pi/2 - a/2 - t/2 ],
         [ 2*r*sin(a/2),                        x1, 5*pi/2 - a/2 - t/2, 3*pi/2]  ]


repSE2 = {t:19*pi/10, a:pi/2} # Replacement values in range

# Define conditions for model
condSE2 = [a <= pi, t >= pi,  a >= 4*pi - 2*t]

# Calculate model, run checks, write output.
pSE2 = calcModel(mSE2)
allChecks('pSE2')
parseLaTeX('pSE2')



##########################################################################################
# SE3 animal: a <= pi.  Sensor: t > pi. Condition: 2pi - t < a < 4pi - 2t #
##########################################################################################

mSE3 = [ [ 2*r*sin(a/2),                        x1, pi/2,                t/2 + pi/2 - a/2  ],
         [ r*sin(a/2) + r*cos(x1 - t/2),        x1, t/2 + pi/2 - a/2,    t/2 + pi/2        ],
         [ r*sin(a/2),                          x1, t/2 + pi/2,          5*pi/2 - a/2 - t/2],
         [ 2*r*sin(a/2),                        x1, 5*pi/2 - a/2 - t/2, 3*pi/2             ] ]

repSE3 = {t:3*pi/2 + 0.1, a:pi/2} # Replacement values in range

# Define conditions for model
condSE3 = [a <= pi, t >= pi,  a >= 2*pi - t, a <= 4*pi - 2*t]

# Calculate model, run checks, write output.
pSE3 = calcModel(mSE3)
allChecks('pSE3')
parseLaTeX('pSE3')



####################################################################################
# SE4 animal: a <= pi.  Sensor: t > pi. Condition: a <= 4*pi - 2*t and a < 2*pi - t #
####################################################################################

mSE4 = [ [ 2*r*sin(a/2),                       x1, pi/2,              t/2 + pi/2 - a/2  ],
         [ r*sin(a/2) + r*cos(x1 - t/2),       x1, t/2 + pi/2 - a/2, t/2 + pi/2        ],
         [ r*sin(a/2),                         x1, t/2 + pi/2,       t/2 + pi/2 + a/2  ] ]


repSE4 = {t:3*pi/2, a:pi/3} # Replacement values in range


# Define conditions for model
condSE4 = [a <= pi, t >= pi/2, a <= 4*pi - 2*t , a <= 2*pi - t]

# Calculate model, run checks, write output.
pSE4 = calcModel(mSE4)
allChecks('pSE4')
parseLaTeX('pSE4')
```

```
##################################################################################


"""
Ccomplex profiles for a <= pi/2
These were specified using a very roundabout way that I realised isn't necessary.
Worth keeping them here just for the record.

# p-l-r for x2 profil. Calculated by AE in fig 22.4 minus AE in fig 22.3
p1 = (2*r*sin(t/4 - x2/2 + pi/4 + a/4)*sin(a/4 + pi/4 + x2/2 - t/4) - \
     2*r*sin((pi - a - 2*x2 + t)/4)*sin((pi - a + 2*x2 - t)/4)).simplify()

# p-l for x2 profiles
p2 = (2*r*sin(t/2)*sin(x2) - 2*r*sin((pi - a - 2*x2 + t)/4)*sin((pi - a + 2*x2 - t)/4)).simplify()

# p-l for x3 profile.
p3 = (r*sin(x3) - (2*r*sin(x3/2 - a/4)*sin(pi/2 - x3/2 - a/4)).simplify()).trigsimp()
"""


###############################################################################################
# SW1 animal: a <= pi.  Sensor: pi/2 <= t <= pi. Condition: a >= t and a/2 >= t - pi/2 #
###############################################################################################


mSW1 =  [ [2*r*sin(t/2)*sin(x2),             x2, pi/2 - a/2 + t/2, pi/2            ],
          [r*sin(a/2) - r*cos(x2 + t/2),     x2, t/2,              pi/2 - a/2 + t/2],
          [r*sin(a/2) - r*cos(x4 - t),       x4, 0,                t - pi/2        ],
          [r*sin(a/2),                       x4, t-pi/2,           t - pi/2 + a/2  ] ]


repSW1 = {t:5*pi/8, a:6*pi/8} # Replacement values in range

# Define conditions for model
condSW1 = [a <= pi, pi/2 <= t, t <= pi, a >= t, a/2 >= t - pi/2]

# Calculate model, run checks, write output.
pSW1 = calcModel(mSW1)
allChecks('pSW1')
parseLaTeX('pSW1')




##############################################################################################
# SW2 animal: a <= pi.  Sensor: pi/2 <= t <= pi. Condition: a <= t and a/2 >= t- pi/2 #
##############################################################################################


mSW2 =  [ [2*r*sin(a/2),                x2, pi/2 + a/2 - t/2, pi/2             ],
          [r*sin(a/2) - r*cos(x2 + t/2), x2, t/2,              pi/2 + a/2 - t/2],
          [r*sin(a/2) - r*cos(x4 - t),  x4, 0*t,              t - pi/2        ],
          [r*sin(a/2),                  x4, t - pi/2,         t - pi/2 + a/2 ] ]


repSW2 = {t:7*pi/8, a:7*pi/8-0.1} # Replacement values in range

# Define conditions for model
condSW2 = [a <= pi, pi/2 <= t, t <= pi, a/2 <= t/2, a/2 >= t - pi/2]

# Calculate model, run checks, write output.
pSW2 = calcModel(mSW2)
allChecks('pSW2')
parseLaTeX('pSW2')







##############################################################################################
# SW3 animal: a <= pi.  Sensor: pi/2 <= t <= pi. Condition: a <= t and a/2 <= t- pi/2 #
##############################################################################################



mSW3 =  [ [2*r*sin(a/2),                     x2, t/2,             pi/2           ],
          [2*r*sin(a/2),                     x4, 0,               t - pi/2 - a/2 ],
          [r*sin(a/2) - r*cos(x4 - t),       x4, t - pi/2 - a/2, t - pi/2       ],
          [r*sin(a/2),                       x4, t - pi/2,       t - pi/2 + a/2 ] ]


repSW3 = {t:7*pi/8, a:2*pi/8} # Replacement values in range

# Define conditions for model
condSW3 = [a <= pi, pi/2 <= t, t <= pi, a/2 <= t/2, a/2 <= t - pi/2]
```

```
# Calculate model, run checks, write output.
pSW3 = calcModel(mSW3)
allChecks('pSW3')
parseLaTeX('pSW3')




#####################################################################################
# SW4 animal: a <= pi.  Sensor: t <= pi/2. Condition: a > pi - 2t &  a <= t       #
#####################################################################################


mSW4 = [ [2*r*sin(a/2),                   x2, pi/2 - t/2 + a/2, pi/2            ],
         [r*sin(a/2) - r*cos(x2 + t/2), x2, pi/2 - t/2,       pi/2 - t/2 + a/2],
         [r*sin(a/2),                   x3, t,                pi/2            ],
         [r*sin(a/2),                   x4, 0,                a/2 + t - pi/2  ] ]

repSW4 = {t:pi/2-0.1, a:pi/4} # Replacement values in range

# Define conditions for model
condSW4 = [a <= pi,  t <= pi/2,  a >= pi - 2*t,  a <= t]

# Calculate model, run checks, write output.
pSW4 = calcModel(mSW4)
allChecks('pSW4')
parseLaTeX('pSW4')


#########################################################################################
# SW5 animal: a <= pi.  Sensor: t <= pi/2. Condition: a > pi - 2t &  t <= a <= 2t     #
#########################################################################################


mSW5 = [ [2*r*sin(t/2)*sin(x2),         x2, pi/2 + t/2 - a/2, pi/2            ],
         [r*sin(a/2) - r*cos(x2 + t/2), x2, pi/2 - t/2,       pi/2 + t/2 - a/2],
         [r*sin(a/2),                   x3, t,                pi/2        ],
         [r*sin(a/2),                   x4, 0,                a/2 + t -pi/2   ] ]


repSW5 = {t:pi/2-0.1, a:pi/2} # Replacement values in range

# define conditions for model
condSW5 = [a <= pi,  t <= pi/2,  a >= pi - 2*t,  t <= a, a <= 2*t]


# Calculate model, run checks, write output.
pSW5 = calcModel(mSW5)
allChecks('pSW5')
parseLaTeX('pSW5')





#####################################################################################
# SW6 animal: a <= pi.  Sensor: t <= pi/2. Condition: a > pi - 2t &  a > 2t       #
#####################################################################################



mSW6 = [ [2*r*sin(t/2)*sin(x2), x2, pi/2 - t/2, pi/2            ],
         [r*sin(x3),            x3, t,          a/2             ],
         [r*sin(a/2),           x3, a/2,        pi/2            ],
         [r*sin(a/2),           x4, 0,          a/2 + t -pi/2   ] ]


repSW6 = {t:pi/4, a:3*pi/4} # Replacement values in range


# Define conditions for model
condSW6 = [a <= pi,  t <= pi/2,  a >= pi - 2*t,  a > 2*t]

# Calculate model, run checks, write output.
pSW6 = calcModel(mSW6)
allChecks('pSW6')
parseLaTeX('pSW6')




######################################################################################
# SW7 animal: a <= pi.  Sensor: t <= pi/2. Condition: a <= pi - 2t & a <= t      #
######################################################################################

```

```
mSW7 = [ [2*r*sin(a/2),                x2, pi/2 - t/2 + a/2, pi/2             ],
         [r*sin(a/2) - r*cos(x2 + t/2), x2, pi/2 - t/2,       pi/2 - t/2 + a/2],
         [r*sin(a/2),                  x3, t,                t + a/2          ] ]


repSW7 = {t:2*pi/8, a:pi/8} # Replacement values in range

# Define conditions for model
condSW7 = [a <= pi, t <= pi/2, a <= pi - 2*t, a <= t]

# Calculate model, run checks, write output.
pSW7 = calcModel(mSW7)
allChecks('pSW7')
parseLaTeX('pSW7')




#######################################################################################
# SW8 animal: a <= pi.  Sensor: t <= pi/2. Condition: a <= pi - 2t & t <= a <= 2t   #
#######################################################################################


mSW8 = [ [2*r*sin(t/2)*sin(x2),        x2, pi/2 + t/2 - a/2, pi/2             ],
         [r*sin(a/2) - r*cos(x2 + t/2), x2, pi/2 - t/2,       pi/2 + t/2 - a/2],
         [r*sin(a/2),                  x3, t,                t + a/2          ] ]

repSW8 = {t:2*pi/8, a:pi/2-0.1} # Replacement values in range

# Define conditions for model
condSW8 = [a <= pi, t <= pi/2, a <= pi - 2*t, t <= a, a <= 2*t]

# Calculate model, run checks, write output.
pSW8 = calcModel(mSW8)
allChecks('pSW8')
parseLaTeX('pSW8')



##################################################################################
# SW9 animal: a <= pi.  Sensor: t <= pi/2. Condition: a <= pi - 2t &  2t <= a      #
##################################################################################

mSW9 = [ [2*r*sin(t/2)*sin(x2), x2, pi/2 - t/2, pi/2    ],
         [r*sin(x3),            x3, t,          a/2     ],
         [r*sin(a/2),           x3, a/2,        t + a/2 ] ]


repSW9 = {t:1*pi/8, a:pi/2} # Replacement values in range

# Define conditions for model
condSW9 = [a <= pi,  t <= pi/2,  a <= pi - 2*t,  2*t <= a]

# Calculate model, run checks, write output.
pSW9 = calcModel(mSW9)
allChecks('pSW9')
parseLaTeX('pSW9')




#
    ###################################################################################################################

####################
## Run tests     ###
####################

# create gas model object
gas = 2*r


# for each model run through every adjacent model.
# Contains duplicatea but better for avoiding missed comparisons.
# 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}]
]


# List of regions that cover a=0. Should equal 0 when a=0.
zeroRegions = [’pSW9’, ’pSW8’, ’pSW7’, ’pSW4’, ’pSW2’, ’pSW3’, ’pSE4’, ’pSE3’, ’pSE2’, ’pSE1’]

# Run through all the comparisons. Need simplify(). Even together() gives some false negatives.

checkFile = open(’/home/tim/Dropbox/phd/Analysis/REM-chapter/checksFile.tex’,’w’)

checkFile.write(’All checks evaluated.\nTim Lucas - ’ + str(datetime.now()) + ’\n’)
for i in range(len(allComps)):
        if (eval(allComps[i][0]).subs(allComps[i][2]) - eval(allComps[i][1]).subs(allComps[i][2])).simplify() == 0:
                checkFile.write(str(i) + ’: ’ + allComps[i][0]+ ’ and ’ +allComps[i][1]+’: OK\n’)
        else:
                checkFile.write(str(i) + ’: ’ + allComps[i][0]+ ’ and ’ +allComps[i][1]+’: Incorrect\n’)

for i in range(len(zeroRegions)):
        if eval(zeroRegions[i]).subs({a:0}).simplify() == 0:
                checkFile.write(zeroRegions[i] + ’ at a=0: OK\n’)
        else:
                checkFile.write(zeroRegions[i] + ’ at a=0: Incorrect\n’)

checkFile.close()


# And print to terminal
#for i in range(len(allComps)):
#        if not (eval(allComps[i][0]).subs(allComps[i][2]) - eval(allComps[i][1]).subs(allComps[i][2])).simplify() == 0:
#                print allComps[i][0] + ’ and ’ + allComps[i][1]+’: Incorrect\n’

####################################
## Check some that don’t work well ##
####################################

xRange = np.arange(0,pi/2, 0.01)
ySW2Range = [pSW2.subs({r:1,  t:pi/2, a:i}).n() for i in xRange]
plotSW2 = pl.plot(xRange, ySW2Range)
pl.savefig(’/home/tim/Dropbox/phd/Analysis/REM-chapter/imgs/pSW2Profile.pdf’)
pl.close()

ySW4Range = [pSW4.subs({r:1,  t:pi/2, a:i}).n() for i in xRange]
plotSW4 = pl.plot(xRange, ySW4Range)
pl.savefig(’/home/tim/Dropbox/phd/Analysis/REM-chapter/imgs/pSW4Profile.pdf’)
pl.close()



#pl.savefig(’/home/tim/Dropbox/phd/Analysis/REM-chapter/imgs/pNE1Profile.pdf’)
#pl.close()
```

```
########################################################################
### Define a a function that calculates your answer.               ####
########################################################################

def calcP(A, T, R):
  assert (A <= 2*pi and A >= 0), "a is out of bounds. Should be in 0<a<2*pi"
  assert (T <= 2*pi and T >= 0), "s is out of bounds. Should be in 0<s<2*pi"

  if A > pi:
    if A < 4*pi - 2* T:
      p = pNW7.subs({a:A,  t:T, r:R}).n()
    elif A <= 3*pi -  T:
                    p = pNE2.subs({a:A,  t:T, r:R}).n()
    else:
                    p = pNE1.subs({a:A,  t:T, r:R}).n()
  else:
    if A < 4*pi - 2* T:
                    p = pSE3.subs({a:A,  t:T, r:R}).n()
    else:
                    p = pSE2.subs({a:A,  t:T, r:R}).n()
        return p


##############################
## Apply to entire grid   ###
##############################

# How many values for each parameter
nParas = 100

# Make a vector for a and s. Make an empty nParas x nParas array.
# Calculated profile sizes will go in pArray
tVec = np.linspace(0, 2*pi, nParas)
aVec = np.linspace(0, 2*pi, nParas)
pArray = np.zeros((nParas,nParas))

# Calculate profile size for each combination of parameters
for i in range(nParas):
        for j in range(nParas):
                pArray[i][j] = calcP(aVec[i], tVec[j], 1)

# Turn the array upside down so origin is at bottom left.
pImage = np.flipud(pArray)

# Plot and save.
pl.imshow(pImage, interpolation='none', cmap=pl.get_cmap('Blues') )
#pl.show()

pl.savefig('/home/tim/Dropbox/phd/Analysis/REM-chapter/imgs/profilesCalculated.png')



#############################
#### Output R function.   ###
#############################

# To reduce mistakes, output R function directly from python.
# However, the if statements, which correspond to the bounds of each model, are not automatic.

Rfunc = open('/home/tim/Dropbox/phd/Analysis/REM-chapter/supplementaryRscript.R', 'w')

Rfunc.write("""
# 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 <- ' + str(pNW7) +
'\n                  } else if(alpha <= 3*pi - theta){'
'\n                          p <- ' + str(pNE2) +
'\n                  } else {'
'\n                          p <- ' + str(pNE1) +
'\n                  }'
'\n          } else {'
'\n           if(alpha < 4*pi - 2*theta){'
'\n                          p <- ' + str(pSE3) +
'\n       } else {'
'\n                          p <- ' + str(pSE2) +
'\n                  }'
'\n          }'
'\n          return(p)'
'\n}' +
"""
# 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)
        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.numer(area)) stop('Area must be a positive number')
        D <- calcDensity(z, alpha, theta, r, animalSpeed, t)
        A <- D*area
        return(A)
}
"""
)

Rfunc.close()
```

REM–Analysis.py

3. Supplementary Script: R implementation of models

This is a simple implementation of the models derived in the paper in R R Development Core Team (2010). Once given the parameters θ and α it automatically selects the correct model to apply.

```

# 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)
        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.numer(area)) stop('Area must be a positive number')
        D <- calcDensity(z, alpha, theta, r, animalSpeed, t)
        A <- D*area
        return(A)
}
```

supplementaryRscript.R

References

R Development Core Team (2010) *R: A Language And Environment For Statistical Computing*. R Foundation For Statistical Computing, Vienna, Austria. ISBN 3-900051-07-0. 25

Rowcliffe, J., Field, J., Turvey, S. & Carbone, C. (2008) Estimating animal density using camera traps without the need for individual recognition. *Journal of Applied Ecology*, **45**, 1228–1236. 5, 9

SymPy Development Team (2014) *SymPy: Python library for symbolic mathematics*. 14