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

Authors:
Tim C.D. Lucas[1,2,3]†, Elizabeth A. Moorcroft[1,4,5]†, Robin Freeman[5], Marcus J. Rowcliffe[5], Kate E. Jones[2,5]

Addresses:
1 CoMPLEX, University College London, Physics Building, Gower Street, London, WC1E 6BT, UK
2 Centre for Biodiversity and Environment Research, Department of Genetics, Evolution and Environment, University College London, Gower Street, London, WC1E 6BT, UK
3 Department of Statistical Science, University College London, Gower Street, London, WC1E 6BT, UK
4 Department of Computer Science, University College London, Gower Street, London, WC1E 6BT, UK
5 Institute of Zoology, Zoological Society of London, Regents Park, London, NW1 4RY, UK
† First authorship shared.

S1. TABLE OF SYMBOLS

Symbol	Description	Units
θ	Sensor width	rad
α	Animal signal/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	$\mathrm{m\,s^{-1}}$
t	Time	s
z	Number of detections	-
D	Animal density	$\mathrm{m^{-2}}$
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 and simulations. '-' means the quantity has no units.

S2. SUPPLEMENTARY METHODS

S2.1. **Introduction.** These supplementary methods derive all the models used. 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 Yapp (1956), we derive the gas model where sensors can capture animals in any direction and animal signals are 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 $\bar{p}$. This $\bar{p}$ is the only thing that changes

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) (Fig. S1). For example, the gas
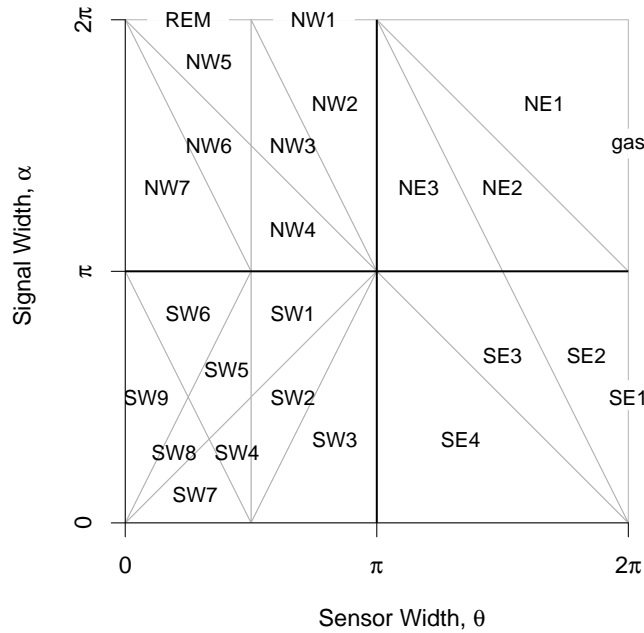

Figure S1. Locations where derivation of the average profile $\bar{p}$ is the same for different combinations of sensor detection and animal signal widths. Symbols within each polygon refer to each gREM submodel named after their compass point, except for Gas and REM which highlight the position of these previously derived models within the gREM. Symbols on the edge of the plot are for submodels where $\alpha, \theta = 2\pi$
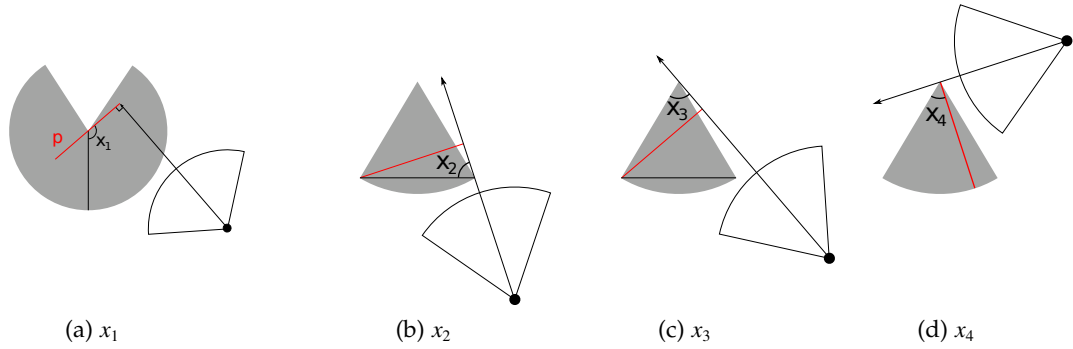
(a) x_1 (b) x_2 (c) x_3 (d) x_4

Figure S2. The location of the focal angles $x_{i \in [1,4]}$. x_1 is used in NE and SE models (including the gas model). $x_2 - x_4$ are used in NW and SW models. 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. (a) Animal is directly approaching the sensor at $x_1 = \frac{\pi}{2}$. (b) Animal is directly approaching the sensor at $x_2 = \frac{\pi}{2}$. x_2 then decreases until the profile is perpendicular to the edge of the detection region. (c) When the profile is perpendicular to the edge of the detection region, $x_3 = \theta$. (d) x_4 measures the angle between the left side of the detection region and the profile.

model becomes the simplest gREM sub-model (upper right in Fig. S1) and the REM from (Rowcliffe *et al.*, 2008) is another gREM sub-model where $\theta < \pi/2$ and $\alpha = 2\pi$.

Models with $\theta = 2\pi$ are described first (the gas model described above and SE1). Then models with $\theta > \pi$ are described (NE then SE). Finally models with $\theta < \pi$ (NW then SW) are described.

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 signal. We therefore get a profile width of $2r \sin(\alpha/2)$ instead.

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

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

This profile is integrated over the interval $[\frac{\pi}{2}, \frac{3\pi}{2}]$ which is π radians of rotation starting with the animal moving directly towards the sensor (see Fig. S2).

S2.4. **Models NE1–3.** 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 (Fig. S1). 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 $2r$.

(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. S3a).

(3) The third profile is only found in NE3. If $\alpha < 4\pi - 2\theta$, then at $x_1 = \theta/2 + \pi/2$, 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. S3b). 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 $r + r \cos(x_1 + \theta/2)$ (Fig. S3c). From $x_1 = \pi$, if the animal signal is wide enough to be detected in this area, this is the wider profile. This then defines the split between NE1
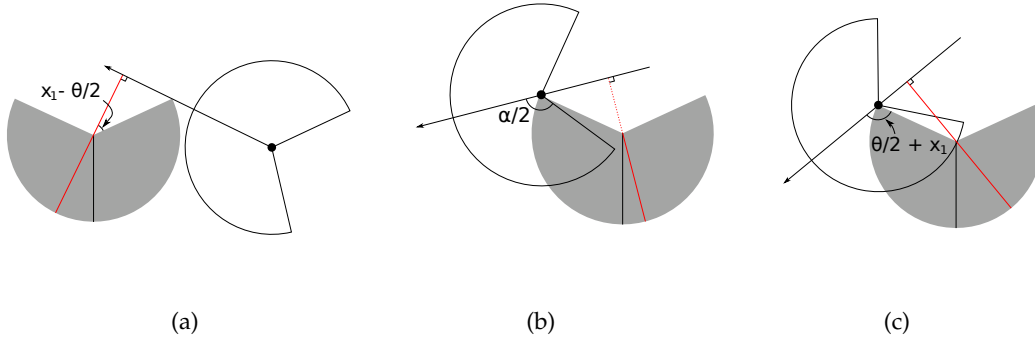
4

(a)   (b)   (c)

Figure S3. Three of the integrals in NE models. 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. Dashed red lines indicate areas where animals cannot be detected. (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. (c) After further rotation, $\alpha/2$ is now bigger than the angle shown and animals to the right of the detector can again be detected.

and NE2. In NE1, with $\alpha > 3\pi - \theta$, the animal signal is wide enough that at $x_1 = \pi$ the animal can immediately 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 $5\pi/2 - \theta/2 - \alpha/2$.

(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 $\alpha \geq 3\pi - \theta$ (Fig. S1). It has four profiles; it does not include the r profile at $x_1 = \pi$ (profile described in point (3) above). Furthermore, θ is wide enough that the $r + r\cos(x_1 + \theta/2)$ profile starts at π. This then gives us

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

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

$$\bar{p}_{\text{NE1}} = \frac{r}{\pi}\left(\theta + 2\sin\left(\frac{\theta}{2}\right)\right) \qquad \text{eqn S4}$$

S2.4.2. *Model NE2.* Model NE2 is bounded by $\alpha \leq 3\pi - \theta$, $\alpha \geq 4\pi - 2\theta$ and $\alpha \geq \pi$ (Fig. S1). 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 $\theta \geq \pi$ (Fig. S1). It is the same as NE2 except that it contains the extra profile with width r (third integral).
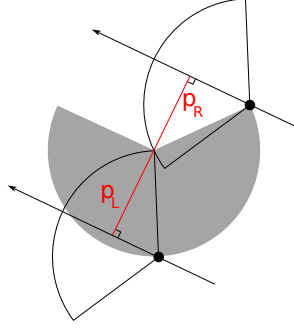
5

Figure S4. 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 signal width. The full profile has width $p = r \sin(\alpha/2) + r \cos(\theta/2 - x_1)$. 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.

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

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

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

S2.5. Models SE2–4. Quadrant SE contains three submodels excluding SE1 (Fig. S1). The differences between these three models are similar to the differences between 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 $r \sin(\alpha/2) + r \cos(x_1 - \theta/2)$ (see Fig. S4).
(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.
(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 separated 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 $\theta \leq 2\pi$ (Fig. S1). 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.

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

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

S2.5.2. Model SE3. SE3 is bounded by $4\pi - 2\theta \leq \alpha \leq 4\pi - 2\theta$ and $\alpha \leq \pi$ (Fig. S1). Therefore there is a $r \sin(\alpha/2)$ profile but no $0r$ profile. This relates to profiles (1), (2), (3) and (5) above.
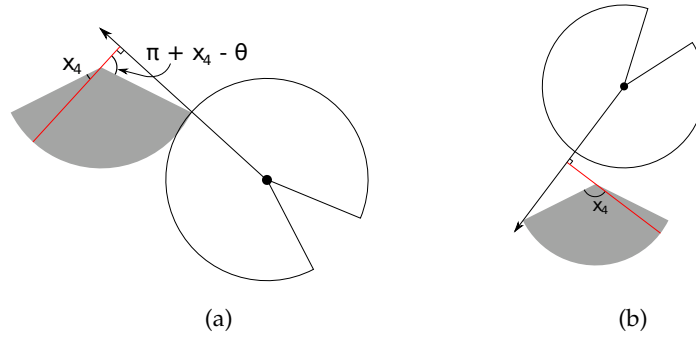
(a)   (b)

Figure S5. The second and fourth profiles of NW1. The left side of of both profiles is of width r while the right side differs. (a) The right side of the profile is $r\cos(\pi + x_4 - \theta) = -r\cos(\theta - x_4)$ (b) The right side is $r\cos(\pi - x_4) = -r\cos x_4$ respectively. In both images 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.

$$\bar{p}_{\text{SE3}} = \frac{1}{\pi}\left(\int_{\frac{\pi}{2}}^{\frac{\pi}{2}+\frac{\theta}{2}-\frac{\alpha}{2}} 2r\sin\left(\frac{\alpha}{2}\right)dx_1 + \int_{\frac{\pi}{2}+\frac{\theta}{2}-\frac{\alpha}{2}}^{\frac{\theta}{2}+\frac{\pi}{2}} r\sin\left(\frac{\alpha}{2}\right) + r\cos\left(\frac{\theta}{2}-x_1\right)dx_1 \right.$$
$$\left. + \int_{\frac{\theta}{2}+\frac{\pi}{2}}^{\frac{5\pi}{2}-\frac{\theta}{2}-\frac{\alpha}{2}} r\sin\left(\frac{\alpha}{2}\right)dx_1 + \int_{\frac{5\pi}{2}-\frac{\theta}{2}-\frac{\alpha}{2}}^{\frac{3\pi}{2}} 2r\sin\left(\frac{\alpha}{2}\right)dx_1 \right) \qquad \text{eqn S11}$$

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

S2.5.3. *Model SE4.* Finally SE4 is bounded by $\alpha \leq 4\pi - 2\theta$, $\alpha \leq \pi$ and $\theta \leq \pi$ (Fig. S1). 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.

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

$$\bar{p}_{\text{SE4}} = \frac{r}{\pi}\left(\theta\sin\left(\frac{\alpha}{2}\right) - \cos\left(\frac{\alpha}{2}\right) + 1\right) \qquad \text{eqn S14}$$

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 (Rowcliffe *et al.*, 2008) while x_4 is new. They are described in Fig. S2b–d.

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 parallel to the left hand radius of the sensor sector (see Fig. S2b). 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 . 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. S5a).

(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.

7

(a)   (b)
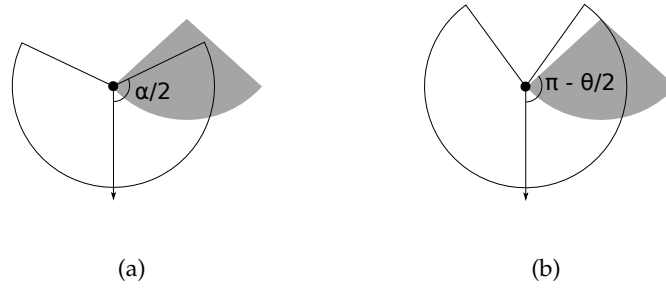
Figure S6. Profile sizes when an animal approaches from behind in models NW2–4. If α is relatively large, animals can be detected when approaching from behind. Otherwise animals cannot be detected. The sector shaped detection region is shown in grey. Animals are filled black circles and the animal signal is an unfilled sector. The animals direction of movement is indicated with an arrow. (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 $2r \sin(\theta/2) \sin(x_2)$.

(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. S5b). 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 after profile (1) animals to the right of the sensor can be detected 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, animals to the right of the sensor may be undetectable 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 inequality is found by noting that animals to the right of the sensor can be detected again 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.

(4) 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 $\alpha/2 \leq \pi - \theta/2$ (Fig. S6). This inequality (simplified as $\alpha \leq 2\pi - \theta$) defines the boundary between NW3 and NW4.

S2.7.1. *Model NW2.* NW2 is bounded by $\alpha \geq 3\pi - 2\theta$, $\alpha \leq 2\pi$ and $\theta \leq \pi$ (Fig. S1).

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 $\alpha \leq 3\pi - 2\theta$, $\alpha \geq 2\pi - \theta$ and $\theta \geq \pi/2$ (Fig. S1).

NW3 does not have the fourth integral from NW2 as animals are not detectable to the right of the sensor 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 profile 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 $\alpha \geq \pi$, $\theta \geq \pi/2$ and $\alpha \leq 2\pi - \theta$ (Fig. S1). NW4 is the same as NW3 except that the final profile width is zero and this profile is reached at $\alpha/2 + \theta/2 - \pi/2$.

$$\bar{p}_{\text{NW4}} = \frac{1}{\pi} \left(\int_{\frac{\theta}{2}}^{\frac{\pi}{2}} 2r \sin\left(\frac{\theta}{2}\right) \sin(x_2) \, dx_2 + \int_{0}^{\theta - \frac{\pi}{2}} r - r \cos(-x_4 + \theta) \, dx_4 \right.$$

$$\left. + \int_{\theta - \frac{\pi}{2}}^{\theta} r \, dx_4 + \int_{\frac{\theta}{2}}^{\frac{\alpha}{2} + \frac{\theta}{2} - \frac{\pi}{2}} r \cos\left(\frac{\theta}{2} - x_2\right) \, dx_2 \right) \qquad \text{eqn S21}$$

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

S2.8. **Model REM.** REM is the model from (Rowcliffe *et al.*, 2008). It has $\alpha = 2\pi$ and $\theta \leq \pi/2$ (Fig. S1). It has three profile widths, two of which are repeated, once as the animal approaches from in 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 on the right hand 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.

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

$$\left. + \int_{0}^{\theta} r \, dx_4 + \int_{\theta}^{\frac{\pi}{2}} r \sin(x_3) \, dx_3 + \int_{\frac{\pi}{2}-\frac{\theta}{2}}^{\frac{\pi}{2}} 2r \sin\left(\frac{\theta}{2}\right) \sin(x_2) \, dx_2 \right) \qquad \text{eqn S23}$$

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

S2.9. **Models NW5–7.** In the models NW5–7, the sensor has $\theta \le \pi/2$ as in the REM. As $\alpha \ge \pi$ 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 $\alpha \ge \pi$ signal is no different to a 2π signal. 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 $\alpha \le 2\pi + 2\theta$, 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 $\alpha \ge 2\pi + 2\theta$ 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. The x_2 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 signal width 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 $\alpha \le 2\pi - \theta$ as in NW6)
(7) or the right becomes detectable (if $\alpha \ge 2\pi - \theta$ 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 $\alpha \ge 2\pi - \theta$, $\alpha \le 2\pi$ and $\theta \le \pi/2$ (Fig. S1).
   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 $\alpha \le 2\pi - \theta$, $\alpha \ge 2\pi + 2\theta$ and $\theta \le \pi/2$ (Fig. S1).
   NW6 is the same NW5 except that as $\alpha \le 2\pi - \theta$, 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.
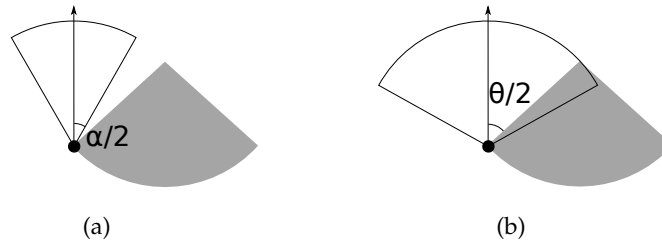
Figure S7. The first profile in SW models is limited by either α or β depending on whether $\alpha < \beta$. The sector shaped detection region is shown in grey. Animals are filled black circles and the animal signal is an unfilled sector. The animals direction of movement is indicated with an arrow. (a) As $\alpha/2 < \theta/2$ the profile width is limited by the signal width rather than the sensor region. The profile width is $2r\sin(\alpha/2)$ (b) As $\alpha/2 > \theta/2$ the profile width is limited by the sensor region, not the signal width. The profile width is $2r\sin(\theta/2)\sin(x_2)$.

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

$$\left. + \int_{0}^{\theta} r\,\mathrm{d}x_4 + \int_{\theta}^{\frac{\pi}{2}} r\sin(x_3)\,\mathrm{d}x_3 + \int_{\frac{\pi}{2}-\frac{\theta}{2}}^{\frac{\alpha}{2}+\frac{\theta}{2}-\frac{\pi}{2}} r\cos\left(\frac{\theta}{2}-x_2\right)\mathrm{d}x_2 \right) \qquad \text{eqn S27}$$

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

S2.9.3. *Model NW7.* NW7 is bounded by $\alpha \geq 2\pi + 2\theta$, $\alpha \geq \pi$ and $\theta \geq 0$ (Fig. S1).

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 signal width is too small for any animals to be detected, so the profile width goes to zero.

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

$$\left. + \int_{0}^{\theta} r\,\mathrm{d}x_4 + \int_{\pi-\frac{\alpha}{2}}^{\frac{\pi}{2}} r\sin(x_3)\,\mathrm{d}x_3 \right) \qquad \text{eqn S29}$$

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

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 signal width can be the limiting parameter, in which case the profile width is instead $2r\sin(\alpha/2)$ (see Fig. S7)

(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 signal 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 signal 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 signal 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 $\theta \leq \pi$ (Fig. S1).

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 signal 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 signal 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 $r\sin(\alpha/2)$.

$$\bar{p}_{SW1} = \frac{1}{\pi}\left(\int_{\frac{\pi}{2}+\frac{\theta}{2}-\frac{\alpha}{2}}^{\frac{\pi}{2}} 2r\sin\left(\frac{\theta}{2}\right)\sin(x_2)\,\mathrm{d}x_2 + \int_{\frac{\theta}{2}}^{\frac{\pi}{2}+\frac{\theta}{2}-\frac{\alpha}{2}} r\sin\left(\frac{\alpha}{2}\right) - r\cos\left(\frac{\theta}{2}+x_2\right)\mathrm{d}x_2 \right.$$
$$\left. + \int_{0}^{\theta-\frac{\pi}{2}} r\sin\left(\frac{\alpha}{2}\right) - r\cos(\theta - x_4)\,\mathrm{d}x_4 + \int_{\theta-\frac{\pi}{2}}^{\frac{\alpha}{2}+\theta-\frac{\pi}{2}} r\sin\left(\frac{\alpha}{2}\right)\mathrm{d}x_4 \right) \qquad \text{eqn S31}$$

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

S2.10.2. Model SW2. SW2 is bounded by $\theta \geq \pi/2$, $\alpha \leq \theta$ and $\alpha \geq 2\theta - \pi$ (Fig. S1).

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

$$\bar{p}_{SW2} = \frac{1}{\pi}\left(\int_{\frac{\alpha}{2}-\frac{\theta}{2}+\frac{\pi}{2}}^{\frac{\pi}{2}} 2r\sin\left(\frac{\alpha}{2}\right)\,\mathrm{d}x_2 + \int_{\frac{\theta}{2}}^{\frac{\alpha}{2}-\frac{\theta}{2}+\frac{\pi}{2}} r\sin\left(\frac{\alpha}{2}\right) - r\cos\left(\frac{\theta}{2}+x_2\right)\mathrm{d}x_2 \right.$$
$$\left. + \int_{0}^{\theta-\frac{\pi}{2}} r\sin\left(\frac{\alpha}{2}\right) - r\cos(\theta - x_4)\,\mathrm{d}x_4 + \int_{\theta-\frac{\pi}{2}}^{\frac{\alpha}{2}+\theta-\frac{\pi}{2}} r\sin\left(\frac{\alpha}{2}\right)\mathrm{d}x_4 \right) \qquad \text{eqn S33}$$

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

S2.10.3. Model SW3. SW3 is bounded by $\alpha \leq 2\theta - \pi$ and $\theta \leq \pi$ (Fig. S1).

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.

$$\bar{p}_{SW3} = \frac{1}{\pi}\left(\int_{\frac{\theta}{2}}^{\frac{\pi}{2}} 2r\sin\left(\frac{\alpha}{2}\right)\,\mathrm{d}x_2 + \int_{0}^{-\frac{\pi}{2}+\theta-\frac{\alpha}{2}} 2r\sin\left(\frac{\alpha}{2}\right)\,\mathrm{d}x_4 \right.$$
$$\left. + \int_{-\frac{\pi}{2}+\theta-\frac{\alpha}{2}}^{\theta-\frac{\pi}{2}} r\sin\left(\frac{\alpha}{2}\right) - r\cos(\theta - x_4)\,\mathrm{d}x_4 + \int_{\theta-\frac{\pi}{2}}^{\frac{\alpha}{2}+\theta-\frac{\pi}{2}} r\sin\left(\frac{\alpha}{2}\right)\mathrm{d}x_4 \right) \qquad \text{eqn S35}$$

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

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.
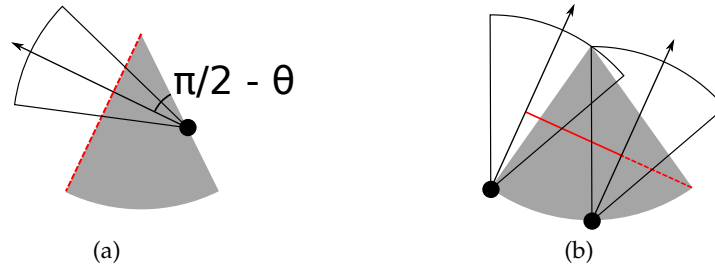
Figure S8. Description of two profiles in SW models. The sector shaped detection region is shown in grey. Animals are filled black circles and the animal signal is an unfilled sector. The animals direction of movement is indicated with an arrow. The profile p is shown with a red line. Dashed red lines indicate areas where animals cannot be detected. (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 signal width, not the sensor. On the left, the profile is limited by the sensor and not the signal. Overall the profile width is $r\sin(\alpha/2) - r\cos(x_2 + \theta/2)$.

(1) Models with $\alpha \leq \pi - 2\theta$ have no x_4 profile. This is because at $x_4 = 0$, the signal width is already too small to be detected as can be seen in Fig. S8a 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 Fig. S7), 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 Fig. S8b). 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 $\theta \leq \pi/2$ (Fig. S1). 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.

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

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

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

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 $\theta \leq \pi/2$ (Fig. S1). 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.

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

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

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

13

S2.11.3. *Model SW6.* SW6 is bounded by $\alpha \geq \pi - 2\theta$, $\alpha \geq 2\theta$ and $\alpha \leq \pi$ (Fig. S1). It starts with a θ limited profile and has a x_4 profile. However, it does not contain the $r\sin(\alpha/2) - r\cos(x_2 + \theta/2)$ profile.

$$\bar{p}_{SW6} = \frac{1}{\pi} \left(\int_{\frac{\pi}{2}-\frac{\theta}{2}}^{\frac{\pi}{2}} 2r \sin\left(\frac{\theta}{2}\right) \sin(x_2) \, dx_2 + \int_{\theta}^{\frac{\alpha}{2}} r \sin(x_3) \, dx_3 \right.$$
$$\left. + \int_{\frac{\alpha}{2}}^{\frac{\pi}{2}} r \sin\left(\frac{\alpha}{2}\right) dx_3 + \int_{0}^{\frac{\alpha}{2}+\theta-\frac{\pi}{2}} r \sin\left(\frac{\alpha}{2}\right) dx_4 \right) \qquad \text{eqn S41}$$

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

S2.11.4. *Model SW7.* SW7 is bounded by $\alpha \leq \pi - 2\theta$, $\alpha \leq \theta$ and $\alpha < 0$ (Fig. S1). 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.

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

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

S2.11.5. *Model SW8.* SW8 is bounded by $\alpha \leq \pi - 2\theta$, $\alpha \geq \theta$ and $\alpha \leq 2\theta$ (Fig. S1). 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.

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

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

S2.11.6. *Model SW9.* Finally, SW9, the last model, is bounded by y $\alpha \leq \pi - 2\theta$, $\alpha \geq 2\theta$ and $\theta \geq 0$ (Fig. S1). Therefore it starts with a θ limited profile. However it does not contain the extra x_2 profile nor a x_4 profile.

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

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

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. 26

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. 4, 7, 9

SymPy Development Team (2014) *SymPy: Python library for symbolic mathematics*. 15

Yapp, W. (1956) The theory of line transects. *Bird study*, **3**, 93–104. 3

S3. 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.

The script can also be found in the plain text file supplementaryS3Python.py.

```python
1  """
2  S3
3  Supplementary Python script from
4  A generalised random encounter model for estimating animal density with remote sensor data
5  Tim C.D. Lucas, Elizabeth A. Moorcroft, Robin Freeman, Marcus J. Rowcliffe, Kate E. Jones
6  """
7
8  """
9  Systematic analysis of REM models
10 Tim Lucas
11 01/10/13
12 """
13
14
15
16 """
17
18 This script contains:
19 1. The integration of all models (lines 24 - 603)
20 2. Some tests that the models are correct (606 - 738)
21 3. A python function to calculate \bar{p} given any parameters (743 - 763)
22 4. Creation of a plot of \bar{p} over all parameter space (767 - 794)
23 5. Code to create supplementaryS4.R, an R implementation of the model (797 - 873)
24
25 """
26
27 from sympy import *
28 import numpy as np
29 import matplotlib.pyplot as pl
30 from datetime import datetime
31 import os as os
32
33 # Set working directory
34 os.chdir('supplementary-material')
35
36 # Use LaTeX printing
37 from sympy import init_printing ;
38 init_printing()
39 # Make LaTeX output white. Because I use a dark theme
40 init_printing(forecolor="White")
41
42
43 # Load symbols used for symbolic maths
44 t, a, r, x2, x3, x4, x1 = symbols('theta alpha r x_2 x_3 x_4 x_1', positive=True)
45 r1 = {r:1} # useful for lots of checks
46
47
48
49 # Define functions
50 # Calculate the final profile averaged over pi.
51 def calcModel(model):
52         x = pi**-1 * sum( [integrate(m[0], m[1:]) for m in model] ).simplify().trigsimp()
53         return x
54
55 # Do the replacements fit within the area defined by the conditions?
56 def confirmReplacements(conds, reps):
57         if not all([c.subs(reps) for c in eval(conds)]):
58                 print('reps' + conds[4:] + ' incorrect')
59
60 # is average profile in range 0r-2r?
61 def profileRange(prof, reps):
62         if not 0 <= eval(prof).subs(dict(reps, **r1)) <= 2:
63                 print('Total ' + prof + ' not in 0, 2r')
64
65 # Are the individuals integrals >0r
66 def intsPositive(model, reps):
67         m = eval(model)
68         for i in range(len(m)):
69                 if not integrate(m[i][0], m[i][1:]).subs(dict(reps, **r1)) > 0:
70                     print('Integral ' + str(i+1) + ' in ' + model + ' is negative')
71
72 # Are the individual averaged integrals between 0 and  2r
73 def intsRange(model, reps):
74         m = eval(model)
75         for i in range(len(m)):
76                 if not 0 <= (integrate(m[i][0], m[i][1:])/(m[i][3]-m[i][2])).subs(dict(reps, **r1)) <=
                        2:
```

15

```
 77                              print('Integral ' + str(i+1) + ' in ' + model + ' has averaged integral outside
                                    0<p<2r')
 78
 79 # Are the bounds the correct way around
 80 def checkBounds(model, reps):
 81         m = eval(model)
 82         for i in range(len(m)):
 83                 if not (m[i][3]-m[i][2]).subs(reps) > 0:
 84                         print('Bounds ' + str(i+1) + ' in ' + model + ' has lower bounds bigger than
                                upper bounds')
 85
 86 # create latex strings with the 1) the integral equation that defines it and 2) the final calculated
        model.
 87 # There's some if statements to split longer equations on two lines and get +s in the right place.
 88 def parseLaTeX(prof):
 89         m = eval( 'm' + prof[1:] )
 90
 91         f = open('/latexFiles/'+prof+'.tex', 'w')
 92         f.write('\\begin{align}\n    \\bar{p}_{\\text{\\tiny{' + prof[1:] + '}}} =&\\frac{1}{\pi} \left
                (\;\;')
 93         for i in range(len(m)):
 94     # Roughly try and prevent expressions beginning with minus signs.
 95     if latex(m[i][2])[0]=='-':
 96       o1 = 'rev-lex'
 97     else:
 98       o1 = 'lex'
 99
100     if latex(m[i][3])[0]=='-':
101       o2 = 'rev-lex'
102     else:
103       o2 = 'lex'
104
105     if latex(m[i][0])[0]=='-':
106       o3 = 'rev-lex'
107     else:
108       o3 = 'lex'
109
110     if latex(m[i][1])[0]=='-':
111       o4 = 'rev-lex'
112     else:
113       o4 = 'lex'
114
115                 f.write('\int\limits_{'+latex(m[i][2], order=o1)+'}^{'+latex(m[i][3], order=o2)+'}'+
                        latex(m[i][0], order=o3)+'\;\mathrm{d}' +latex(m[i][1], order=o4))
116                 if len(m)>3 and i==(len(m)/2)-1:
117                         f.write( '\\right.\\notag\\\\\n &\left.' )
118                 if i<len(m)-1:
119                         f.write('+')
120         f.write('\\right)\label{' + prof + 'Def}\\\\\n    ')
121         f.write('\\bar{p}_{\\text{\\tiny{' + prof[1:] + '}}}  =& ' + latex(eval(prof)) + '\label{' +
                prof + 'Sln}\n\\end{align}')
122         f.close()
123
124
125 # Apply all checks.
126 def allChecks(prof):
127         model = 'm' + prof[1:]
128         reps = eval('rep' + prof[1:])
129         conds = 'cond' + prof[1:]
130         confirmReplacements(conds, reps)
131         profileRange(prof, reps)
132         intsPositive(model, reps)
133         intsRange(model, reps)
134         checkBounds(model, reps)
135
136 ######################################################
137 ### Define and solve all models                   ###
138 ######################################################
139
140 # NE1 animal: a = 2*pi.  sensor: t > pi, a > 3pi - t  #
141
142 mNE1 = [ [2*r,                x1, pi/2, t/2          ],
143          [r + r*cos(x1 - t/2), x1, t/2,  pi          ],
144          [r + r*cos(x1 + t/2), x1, pi,    2*pi-t/2   ],
145          [2*r,                x1, 2*pi-t/2, 3*pi/2 ] ]
146
147 # Replacement values in range
148 repNE1 = {t:3*pi/2, a:2*pi}
149
150 # Define conditions for model
151 condNE1 = [pi <= t, a >= 3*pi - t]
152
153 # Calculate model, run checks, write output.
154 pNE1 = calcModel(mNE1)
155 allChecks('pNE1')
156 parseLaTeX('pNE1')
157
```

```
158
159 # NE2 animal: a > pi.  sensor: t > pi Condition: a < 3pi - t, a > 4pi - 2t  #
160
161 mNE2 = [ [2*r,                 x1, pi/2, t/2           ],
162          [r + r*cos(x1 - t/2), x1, t/2,  5*pi/2 - t/2 - a/2 ],
163          [r + r*cos(x1 + t/2), x1, 5*pi/2 - t/2 - a/2,   2*pi-t/2 ],
164          [2*r,                 x1, 2*pi-t/2, 3*pi/2 ] ]
165
166 # Replacement values in range
167 repNE2 = {t:5*pi/3, a:4*pi/3-0.1}
168
169 # Define conditions for model
170 condNE2 = [pi <= t, a >= pi, a <= 3*pi - t, a >= 4*pi - 2*t]
171
172 # Calculate model, run checks, write output.
173 pNE2 = calcModel(mNE2)
174 allChecks('pNE2')
175 parseLaTeX('pNE2')
176
177
178 # NE3 animal: a > pi.  sensor: t > pi Condition: a < 4pi - 2t  #
179
180 mNE3 = [ [2*r,                 x1, pi/2, t/2           ],
181          [r + r*cos(x1 - t/2), x1, t/2,  t/2 + pi/2       ],
182          [r                  , x1, t/2 + pi/2,    5*pi/2 - t/2 - a/2 ],
183          [r + r*cos(x1 + t/2), x1, 5*pi/2 - t/2 - a/2,   2*pi-t/2 ],
184          [2*r,                 x1, 2*pi-t/2, 3*pi/2 ] ]
185
186 # Replacement values in range
187 repNE3 = {t:5*pi/4-0.1, a:3*pi/2}
188
189 # Define conditions for model
190 condNE3 = [pi <= t, a >= pi, a <= 4*pi - 2*t]
191
192 # Calculate model, run checks, write output.
193 pNE3 = calcModel(mNE3)
194 allChecks('pNE3')
195 parseLaTeX('pNE3')
196
197
198 # NW1 animal: a = 2*pi.   sensor:  pi/2 <= t <= pi      #
199
200 mNW1 = [ [2*r*sin(t/2)*sin(x2), x2, t/2,      pi/2      ],
201          [r - r*cos(x4 - t),    x4, 0,        t - pi/2 ],
202          [r,                    x4, t - pi/2, pi/2      ],
203          [r - r*cos(x4),        x4, pi/2,     t         ],
204          [2*r*sin(t/2)*sin(x2), x2, t/2,      pi/2     ] ]
205
206 # Replacement values in range
207 repNW1 = {t:3*pi/4}
208
209 # Define conditions for model
210 condNW1 = [pi/2 <= t, t <= pi]
211
212 # Calculate model, run checks, write output.
213 pNW1 = calcModel(mNW1)
214 allChecks('pNW1')
215 parseLaTeX('pNW1')
216
217
218
219
220 # NW2 animal: a > pi.  Sensor: pi/2 <= t <= pi. Condition: a > 2pi - t        #
221
222 mNW2 = [ [2*r*sin(t/2)*sin(x2), x2, t/2,         pi/2      ],
223          [r - r*cos(x4 - t),    x4, 0,           t - pi/2 ],
224          [r,                    x4, t - pi/2,    3*pi/2 - a/2],
225          [r - r*cos(x4),        x4, 3*pi/2 - a/2, t         ],
226          [2*r*sin(t/2)*sin(x2), x2, t/2,         pi/2     ] ]
227
228
229 repNW2 = {t:3*pi/4, a:15*pi/8} # Replacement values in range
230
231 # Define conditions for model
232 condNW2 = [a > pi, pi/2 <= t, t <= pi, a >= 3*pi - 2*t]
233
234 # Calculate model, run checks, write output.
235 pNW2 = calcModel(mNW2)
236 allChecks('pNW2')
237 parseLaTeX('pNW2')
238
239
240
241 # NW3 animal: a > pi.  Sensor: pi/2 <= t <= pi. Cond: 2pi - t < a < 3pi - 2t     #
242
243 mNW3 = [ [2*r*sin(t/2)*sin(x2), x2, t/2,                pi/2              ],
244          [r - r*cos(x4 - t),    x4, 0,                  t - pi/2           ],
```

```
245|               [r,                      x4, t - pi/2,              t                    ],
246|               [r*cos(x2 - t/2),        x2, t/2,                   3*pi/2 - a/2 - t/2],
247|               [2*r*sin(t/2)*sin(x2), x2, 3*pi/2 - a/2 - t/2, pi/2             ] ]
248|
249|
250| repNW3 = {t:5*pi/8, a:6*pi/4} # Replacement values in range
251|
252| # Define conditions for model
253| condNW3 = [a > pi, pi/2 <= t, t <= pi, 2*pi - t <= a, a <= 3*pi - 2*t]
254|
255| # Calculate model, run checks, write output.
256| pNW3 = calcModel(mNW3)
257| allChecks('pNW3')
258| parseLaTeX('pNW3')
259|
260|
261|
262| # NW4 animal:  a > pi.  Sensor: pi/2 <= t <= pi. Condition: a <= 2pi - t      #
263|
264| mNW4 = [ [2*r*sin(t/2)*sin(x2), x2, t/2, pi/2],
265|          [r - r*cos(x4 - t),     x4, 0, t - pi/2],
266|          [r,                      x4, t - pi/2, t],
267|          [r*cos(x2 - t/2),        x2, t/2, a/2 + t/2 - pi/2] ]
268|
269| repNW4 = {t:3*pi/4, a:9*pi/8} # Replacement values in range
270|
271| # Define conditions for model
272| condNW4 = [a > pi,  pi/2 <= t, t <= pi, a <= 2*pi - t]
273|
274| # Calculate model, run checks, write output.
275| pNW4 = calcModel(mNW4)
276| allChecks('pNW4')
277| parseLaTeX('pNW4')
278|
279|
280| # REM animal: a=2pi.  Sensor: t <= pi/2.                                        #
281|
282| mREM = [ [2*r*sin(t/2)*sin(x2), x2, pi/2 - t/2, pi/2],
283|          [r*sin(x3),            x3, t,          pi/2],
284|          [r,                      x4, 0*t,         t],
285|          [r*sin(x3),            x3, t,          pi/2],
286|          [2*r*sin(t/2)*sin(x2), x2, pi/2 - t/2, pi/2] ]
287|
288|
289| repREM = {t:3*pi/8, a:2*pi} # Replacement values in range
290|
291| # Define conditions for model
292| condREM = [ t <= pi/2   ]
293|
294| # Calculate model, run checks, write output.
295| pREM = calcModel(mREM)
296| allChecks('pREM')
297| parseLaTeX('pREM')
298|
299|
300|
301| # NW5 animal: a>pi.  Sensor: t <= pi/2. Condition: 2*pi - t < a             #
302|
303|
304| mNW5 = [ [2*r*sin(t/2)*sin(x2), x2, pi/2 - t/2, pi/2],
305|          [r*sin(x3),            x3, t,          pi/2],
306|          [r,                      x4, 0,          t],
307|          [r*sin(x3),            x3, t,          pi/2],
308|          [r*cos(x2 - t/2),        x2, pi/2 - t/2, 3*pi/2 - t/2 - a/2],
309|          [2*r*sin(t/2)*sin(x2), x2, 3*pi/2 - t/2 - a/2, pi/2] ]
310|
311|
312| repNW5 = {t:3*pi/8, a:29*pi/16} # Replacement values in range
313|
314| # Define conditions for model
315| condNW5 = [a >= pi, t <= pi/2, 2*pi - t <= a  ]
316|
317| # Calculate model, run checks, write output.
318| pNW5 = calcModel(mNW5)
319| allChecks('pNW5')
320| parseLaTeX('pNW5')
321|
322|
323| # NW6 animal: a>pi.  Sensor: t <= pi/2. Condition:  2*pi - 2*t <= a <= 2*pi - t  #
324|
325|
326| mNW6 = [ [2*r*sin(t/2)*sin(x2), x2, pi/2 - t/2, pi/2],
327|          [r*sin(x3),            x3, t,          pi/2],
328|          [r,                      x4, 0,          t],
329|          [r*sin(x3),            x3, t,          pi/2],
330|          [r*cos(x2 - t/2),        x2, pi/2 - t/2, a/2 + t/2 - pi/2] ]
331|
```

```
332  repNW6 = {t:3*pi/8, a:3*pi/2} # Replacement values in range
333
334  # Define conditions for model
335  condNW6 = [a >= pi, t <= pi/2, 2*pi - 2*t <= a, a <= 2*pi - t]
336
337  # Calculate model, run checks, write output.
338  pNW6 = calcModel(mNW6)
339  allChecks('pNW6')
340  parseLaTeX('pNW6')
341
342
343
344  # NW7 animal: a>pi.  Sensor: t <= pi/2. Condition: a <= 2pi - 2t  #
345
346
347  mNW7 = [ [2*r*sin(t/2)*sin(x2), x2, pi/2 - t/2, pi/2],
348           [r*sin(x3),            x3, t,          pi/2],
349           [r,                    x4, 0,          t   ],
350           [r*sin(x3),            x3, pi - a/2,   pi/2] ]
351
352
353  repNW7 = {t:pi/9, a:10*pi/9} # Replacement values in range
354
355  # Define conditions for model
356  condNW7 = [t <= pi/2, a >= pi, a <= 2*pi - 2*t]
357
358  # Calculate model, run checks, write output.
359  pNW7 = calcModel(mNW7)
360  allChecks('pNW7')
361  parseLaTeX('pNW7')
362
363
364
365  # SE1 animal: a <= pi.  Sensor: t =2pi.            #
366
367  mSE1 = [ [ 2*r*sin(a/2),x1, pi/2, 3*pi/2        ],
368           ]
369
370
371  repSE1 = {a:pi/4} # Replacement values in range
372
373  # Define conditions for model
374  condSE1 = [a <= pi]
375
376  # Calculate model, run checks, write output.
377  pSE1 = calcModel(mSE1)
378  allChecks('pSE1')
379  parseLaTeX('pSE1')
380
381
382
383
384  # SE2 animal: a <= pi.  Sensor: t > pi. Condition: a > 2pi - t, a > 4pi - 2t     #
385
386  mSE2 = [ [ 2*r*sin(a/2),                    x1, pi/2,                t/2 + pi/2 - a/2        ],
387           [ r*sin(a/2) + r*cos(x1 - t/2),    x1, t/2 + pi/2 - a/2,   5*pi/2 - a/2 - t/2 ],
388           [ 2*r*sin(a/2),                    x1, 5*pi/2 - a/2 - t/2, 3*pi/2]  ]
389
390
391  repSE2 = {t:19*pi/10, a:pi/2} # Replacement values in range
392
393  # Define conditions for model
394  condSE2 = [a <= pi, t >= pi,  a >= 4*pi - 2*t]
395
396  # Calculate model, run checks, write output.
397  pSE2 = calcModel(mSE2)
398  allChecks('pSE2')
399  parseLaTeX('pSE2')
400
401
402  # SE3 animal: a <= pi.  Sensor: t > pi. Condition: 2pi - t < a < 4pi - 2t #
403
404  mSE3 = [ [ 2*r*sin(a/2),                    x1, pi/2,                t/2 + pi/2 - a/2  ],
405           [ r*sin(a/2) + r*cos(x1 - t/2),    x1, t/2 + pi/2 - a/2,   t/2 + pi/2        ],
406           [ r*sin(a/2),                      x1, t/2 + pi/2,         5*pi/2 - a/2 - t/2],
407           [ 2*r*sin(a/2),                    x1, 5*pi/2 - a/2 - t/2, 3*pi/2            ] ]
408
409  repSE3 = {t:3*pi/2 + 0.1, a:pi/2} # Replacement values in range
410
411  # Define conditions for model
412  condSE3 = [a <= pi, t >= pi,  a >= 2*pi - t, a <= 4*pi - 2*t]
413
414  # Calculate model, run checks, write output.
415  pSE3 = calcModel(mSE3)
416  allChecks('pSE3')
417  parseLaTeX('pSE3')
418
```

```
419
420 # SE4 animal: a <= pi.  Sensor: t > pi. Condition: a <= 4*pi - 2*t and a < 2*pi - t #
421
422
423 mSE4 = [ [ 2*r*sin(a/2),                  x1, pi/2,          t/2 + pi/2 - a/2  ],
424         [ r*sin(a/2) + r*cos(x1 - t/2),   x1, t/2 + pi/2 - a/2, t/2 + pi/2    ],
425         [ r*sin(a/2),                     x1, t/2 + pi/2,    t/2 + pi/2 + a/2  ] ]
426
427
428 repSE4 = {t:3*pi/2, a:pi/3} # Replacement values in range
429
430
431 # Define conditions for model
432 condSE4 = [a <= pi, t >= pi/2, a <= 4*pi - 2*t , a <= 2*pi - t]
433
434 # Calculate model, run checks, write output.
435 pSE4 = calcModel(mSE4)
436 allChecks('pSE4')
437 parseLaTeX('pSE4')
438
439
440 # SW1 animal: a <= pi.  Sensor: pi/2 <= t <= pi. Condition: a >= t and a/2 >= t - pi/2 #
441
442 mSW1 =  [ [2*r*sin(t/2)*sin(x2),          x2, pi/2 - a/2 + t/2, pi/2           ],
443          [r*sin(a/2) - r*cos(x2 + t/2),   x2, t/2,           pi/2 - a/2 + t/2],
444          [r*sin(a/2) - r*cos(x4 - t),     x4, 0,             t - pi/2        ],
445          [r*sin(a/2),                     x4, t-pi/2,        t - pi/2 + a/2  ] ]
446
447
448 repSW1 = {t:5*pi/8, a:6*pi/8} # Replacement values in range
449
450 # Define conditions for model
451 condSW1 = [a <= pi, pi/2 <= t, t <= pi, a >= t, a/2 >= t - pi/2]
452
453 # Calculate model, run checks, write output.
454 pSW1 = calcModel(mSW1)
455 allChecks('pSW1')
456 parseLaTeX('pSW1')
457
458
459 # SW2 animal: a <= pi.  Sensor: pi/2 <= t <= pi. Condition: a <= t and a/2 >= t- pi/2 #
460
461 mSW2 =  [ [2*r*sin(a/2),                  x2, pi/2 + a/2 - t/2, pi/2           ],
462          [r*sin(a/2) - r*cos(x2 + t/2),   x2, t/2,           pi/2 + a/2 - t/2],
463          [r*sin(a/2) - r*cos(x4 - t),     x4, 0*t,          t - pi/2        ],
464          [r*sin(a/2),                     x4, t - pi/2,      t - pi/2 + a/2  ] ]
465
466
467 repSW2 = {t:7*pi/8, a:7*pi/8-0.1} # Replacement values in range
468
469 # Define conditions for model
470 condSW2 = [a <= pi, pi/2 <= t, t <= pi, a/2 <= t/2, a/2 >= t - pi/2]
471
472 # Calculate model, run checks, write output.
473 pSW2 = calcModel(mSW2)
474 allChecks('pSW2')
475 parseLaTeX('pSW2')
476
477
478
479 # SW3 animal: a <= pi.  Sensor: pi/2 <= t <= pi. Condition: a <= t and a/2 <= t- pi/2 #
480
481 mSW3 =  [ [2*r*sin(a/2),                  x2, t/2,           pi/2           ],
482          [2*r*sin(a/2),                   x4, 0,             t - pi/2 - a/2 ],
483          [r*sin(a/2) - r*cos(x4 - t),     x4, t - pi/2 - a/2, t - pi/2       ],
484          [r*sin(a/2),                     x4, t - pi/2,      t - pi/2 + a/2  ] ]
485
486
487 repSW3 = {t:7*pi/8, a:2*pi/8} # Replacement values in range
488
489 # Define conditions for model
490 condSW3 = [a <= pi, pi/2 <= t, t <= pi, a/2 <= t/2, a/2 <= t - pi/2]
491
492 # Calculate model, run checks, write output.
493 pSW3 = calcModel(mSW3)
494 allChecks('pSW3')
495 parseLaTeX('pSW3')
496
497
498 # SW4 animal: a <= pi.  Sensor: t <= pi/2. Condition: a > pi - 2t &  a <= t      #
499
500 mSW4 = [ [2*r*sin(a/2),                  x2, pi/2 - t/2 + a/2, pi/2          ],
501         [r*sin(a/2) - r*cos(x2 + t/2),   x2, pi/2 - t/2,    pi/2 - t/2 + a/2],
502         [r*sin(a/2),                     x3, t,             pi/2            ],
503         [r*sin(a/2),                     x4, 0,             a/2 + t - pi/2  ] ]
504
505 repSW4 = {t:pi/2-0.1, a:pi/4} # Replacement values in range
```

```
506
507  # Define conditions for model
508  condSW4 = [a <= pi,  t <= pi/2,  a >= pi - 2*t,  a <= t]
509
510  # Calculate model, run checks, write output.
511  pSW4 = calcModel(mSW4)
512  allChecks('pSW4')
513  parseLaTeX('pSW4')
514
515
516  # SW5 animal: a <= pi.  Sensor: t <= pi/2. Condition: a > pi - 2t &  t <= a <= 2t    #
517
518  mSW5 = [ [2*r*sin(t/2)*sin(x2),        x2, pi/2 + t/2 - a/2, pi/2           ],
519           [r*sin(a/2) - r*cos(x2 + t/2), x2, pi/2 - t/2,      pi/2 + t/2 - a/2],
520           [r*sin(a/2),                  x3, t,               pi/2           ],
521           [r*sin(a/2),                  x4, 0,               a/2 + t -pi/2   ] ]
522
523
524  repSW5 = {t:pi/2-0.1, a:pi/2} # Replacement values in range
525
526  # define conditions for model
527  condSW5 = [a <= pi,  t <= pi/2,  a >= pi - 2*t,  t <= a, a <= 2*t]
528
529
530  # Calculate model, run checks, write output.
531  pSW5 = calcModel(mSW5)
532  allChecks('pSW5')
533  parseLaTeX('pSW5')
534
535
536  # SW6 animal: a <= pi.  Sensor: t <= pi/2. Condition: a > pi - 2t &  a > 2t      #
537
538  mSW6 = [ [2*r*sin(t/2)*sin(x2), x2, pi/2 - t/2, pi/2          ],
539           [r*sin(x3),           x3, t,          a/2           ],
540           [r*sin(a/2),          x3, a/2,        pi/2          ],
541           [r*sin(a/2),          x4, 0,          a/2 + t -pi/2   ] ]
542
543
544  repSW6 = {t:pi/4, a:3*pi/4} # Replacement values in range
545
546
547  # Define conditions for model
548  condSW6 = [a <= pi,  t <= pi/2,  a >= pi - 2*t,  a > 2*t]
549
550  # Calculate model, run checks, write output.
551  pSW6 = calcModel(mSW6)
552  allChecks('pSW6')
553  parseLaTeX('pSW6')
554
555
556  # SW7 animal: a <= pi.  Sensor: t <= pi/2. Condition: a <= pi - 2t & a <= t     #
557
558  mSW7 = [ [2*r*sin(a/2),                x2, pi/2 - t/2 + a/2, pi/2          ],
559           [r*sin(a/2) - r*cos(x2 + t/2), x2, pi/2 - t/2,      pi/2 - t/2 + a/2],
560           [r*sin(a/2),                  x3, t,               t + a/2         ] ]
561
562
563  repSW7 = {t:2*pi/8, a:pi/8} # Replacement values in range
564
565  # Define conditions for model
566  condSW7 = [a <= pi, t <= pi/2, a <= pi - 2*t, a <= t]
567
568  # Calculate model, run checks, write output.
569  pSW7 = calcModel(mSW7)
570  allChecks('pSW7')
571  parseLaTeX('pSW7')
572
573
574  # SW8 animal: a <= pi.  Sensor: t <= pi/2. Condition: a <= pi - 2t & t <= a <= 2t   #
575
576  mSW8 = [ [2*r*sin(t/2)*sin(x2),        x2, pi/2 + t/2 - a/2, pi/2          ],
577           [r*sin(a/2) - r*cos(x2 + t/2), x2, pi/2 - t/2,      pi/2 + t/2 - a/2],
578           [r*sin(a/2),                  x3, t,               t + a/2         ] ]
579
580  repSW8 = {t:2*pi/8, a:pi/2-0.1} # Replacement values in range
581
582  # Define conditions for model
583  condSW8 = [a <= pi, t <= pi/2, a <= pi - 2*t, t <= a, a <= 2*t]
584
585  # Calculate model, run checks, write output.
586  pSW8 = calcModel(mSW8)
587  allChecks('pSW8')
588  parseLaTeX('pSW8')
589
590
591  # SW9 animal: a <= pi.  Sensor: t <= pi/2. Condition: a <= pi - 2t &  2t <= a      #
592
```

21

```
593  mSW9 = [ [2*r*sin(t/2)*sin(x2), x2, pi/2 - t/2, pi/2   ],
594          [r*sin(x3),             x3, t,          a/2    ],
595          [r*sin(a/2),            x3, a/2,        t + a/2 ] ]
596
597
598  repSW9 = {t:1*pi/8, a:pi/2} # Replacement values in range
599
600  # Define conditions for model
601  condSW9 = [a <= pi,  t <= pi/2,  a <= pi - 2*t,  2*t <= a]
602
603  # Calculate model, run checks, write output.
604  pSW9 = calcModel(mSW9)
605  allChecks('pSW9')
606  parseLaTeX('pSW9')
607
608
609  ####################
610  ## Run tests     ###
611  ####################
612
613  # create gas model object
614  gas = 2*r
615
616
617  # for each model run through every adjacent model.
618  # Contains duplicates but better for avoiding missed comparisons.
619  # Also contains replacement t->a and a->t just in case.
620
621
622  allComps = [
623  ['gas', 'pNE1', {t:2*pi}], ['gas', 'pSE1', {a:pi}],
624
625  ['pNE1', 'gas', {t:2*pi}], ['pNE1', 'pNW1', {t:pi}],
626  ['pNE1', 'pNE2',{a:3*pi-t}], ['pNE1', 'pNE2',{t:3*pi-a}],
627
628  ['pNE2', 'pNE1',{a:3*pi-t}], ['pNE2', 'pNE1',{t:3*pi-a}],
629  ['pNE2', 'pNE3',{a:4*pi-2*t}], ['pNE2', 'pNE3',{t:2*pi-a/2}],
630  ['pNE2', 'pSE2',{a:pi}],
631
632  ['pNE3', 'pNE2',{a:4*pi-2*t}], ['pNE3', 'pNE2',{t:2*pi-a/2}],
633  ['pNE3', 'pSE3',{a:pi}], ['pNE3', 'pNW2',{t:pi}],
634
635  ['pNW1','pNE1', {t:pi}], ['pNW1','pNW2',{a:2*pi}],
636
637  ['pNW2','pNE3',{t:pi}], ['pNW2','pNW3',{a:3*pi-2*t}],
638  ['pNW2','pNW3',{t:3*pi/2-a/2}], ['pNW2','pNW1',{a:2*pi}],
639
640  ['pNW3','pNW5',{t:pi/2}], ['pNW3','pNW4',{a:2*pi-t}],
641  ['pNW3','pNW4',{t:2*pi-a}], ['pNW3','pNW2',{a:3*pi-2*t}],
642  ['pNW3','pNW2',{t:3*pi/2-a/2}],
643
644  ['pNW4','pNW6',{t:pi/2}], ['pNW4','pNW3',{t:2*pi-a}],
645  ['pNW4','pNW3',{a:2*pi-t}], ['pNW4','pSW1',{a:pi}],
646
647  ['pREM','pNW1', {t:pi/2}], ['pREM','pNW5',{a:2*pi}],
648
649  ['pNW5','pREM',{a:2*pi}], ['pNW5','pNW6',{a:2*pi-t}],
650  ['pNW5','pNW6',{t:2*pi-a}], ['pNW5','pNW3',{t:pi/2}],
651
652  ['pNW6','pNW5',{a:2*pi-t}], ['pNW6','pNW5',{t:2*pi-a}],
653  ['pNW6','pNW7',{t:pi-a/2}], ['pNW6','pNW7',{a:2*pi-2*t}],
654  ['pNW5','pNW4',{t:pi/2}],
655
656  ['pNW7','pNW6',{t:2*pi-2*a}], ['pNW7','pNW6',{a:2*pi-2*t}],
657  ['pNW7','pSW6',{a:pi}],
658
659  ['pSE1','pSE2',{t:2*pi}], ['pSE1','gas',{a:pi}],
660
661  ['pSE2','pSE3',{t:2*pi-a/2}], ['pSE2','pSE3',{a:4*pi-2*t}],
662  ['pSE2','pSE1',{t:2*pi}], ['pSE2','pNE2',{a:pi}],
663
664  ['pSE3','pSE2',{a:4*pi-2*t}], ['pSE3','pSE2',{t:2*pi-a/2}],
665  ['pSE3','pSE4',{a:2*pi-t}], ['pSE3','pSE4',{t:2*pi-a}],
666  ['pSE3','pNE3',{a:pi}],
667
668  ['pSE4','pSE3',{t:2*pi-a}], ['pSE4','pSE3',{a:2*pi-t}],
669  ['pSE4','pSW3',{t:pi}],
670
671  ['pSW1','pSW5',{t:pi/2}], ['pSW1','pSW2',{a:t}],
672  ['pSW1','pSW2',{t:a}], ['pSW1','pNW4',{a:pi}],
673
674  ['pSW2','pSW1',{a:t}], ['pSW2','pSW1',{t:a}],
675  ['pSW2','pSW4',{t:pi/2}], ['pSW2','pSW3',{a:2*t-pi}],
676  ['pSW2','pSW3',{t:a/2+pi/2}],
677
678  ['pSW3','pSW2',{t:a/2+pi/2}], ['pSW3','pSW2',{a:2*t-pi}],
679  ['pSW3','pSE4',{t:pi}],
```

```
680
681
682 ['pSW4','pSW7',{a:pi-2*t}], ['pSW4','pSW7',{t:pi/2-a/2}],
683 ['pSW4','pSW5',{t:a}], ['pSW4','pSW5',{a:t}],
684 ['pSW4','pSW2',{t:pi/2}],
685
686 ['pSW5','pSW4',{t:a}], ['pSW5','pSW4',{a:t}],
687 ['pSW5','pSW8',{t:pi/2-a/2}], ['pSW5','pSW8',{a:pi-2*t}],
688 ['pSW5','pSW6',{a:2*t}], ['pSW5','pSW6',{t:a/2}],
689 ['pSW5','pSW1',{t:pi/2}],
690
691 ['pSW6','pSW9',{t:pi/2-a/2}], ['pSW6','pSW9',{a:pi-2*t}],
692 ['pSW6','pSW5',{a:2*t}], ['pSW6','pSW5',{t:a/2}],
693 ['pSW6','pNW7',{a:pi}],
694
695
696 ['pSW7','pSW8',{t:a}], ['pSW7','pSW8',{a:t}],
697 ['pSW7','pSW4',{t:pi/2-a/2}], ['pSW7','pSW4',{a:pi-2*t}],
698
699 ['pSW8','pSW7',{a:t}], ['pSW8','pSW7',{t:a}],
700 ['pSW8','pSW9',{a:2*t}], ['pSW8','pSW9',{t:a/2}],
701 ['pSW8','pSW5',{a:pi-2*t}], ['pSW8','pSW5',{t:pi/2-a/2}],
702
703 ['pSW9','pSW8',{a:2*t}], ['pSW9','pSW8',{t:a/2}],
704 ['pSW9','pSW6',{a:pi-2*t}], ['pSW9','pSW6',{t:pi/2-a/2}]
705 ]
706
707
708 # List of regions that touch a=0. Should equal 0 when a=0.
709 zeroRegions = ['pSW9', 'pSW8', 'pSW7', 'pSW4', 'pSW2', 'pSW3', 'pSE4', 'pSE3',  'pSE1']
710
711
712 # Run through all the comparisons. Need simplify(). Even together() gives some false negatives.
713
714 checkFile = open('checksFile.tex','w')
715
716 checkFile.write('All checks evaluated.\nTim Lucas - ' + str(datetime.now()) + '\n')
717 for i in range(len(allComps)):
718         if (eval(allComps[i][0]).subs(allComps[i][2]) - eval(allComps[i][1]).subs(allComps[i][2])).\
               simplify() == 0:
719                 checkFile.write(str(i) + ': ' + allComps[i][0]+ ' and ' +allComps[i][1]+': OK\n')
720         else:
721                 checkFile.write(str(i) + ': ' + allComps[i][0]+ ' and ' +allComps[i][1]+': Incorrect\n')
722
723 for i in range(len(zeroRegions)):
724         if eval(zeroRegions[i]).subs({a:0}).simplify() == 0:
725                 checkFile.write(zeroRegions[i] + ' at a = 0: OK\n')
726         else:
727                 checkFile.write(zeroRegions[i] + ' at a = 0: Incorrect\n')
728
729 # pSE2 is slightly different. Only one corner touches a=0, so need theta value as well. I'm not sure why
        this isn't
730 # A problem for some other regions.
731 if pSE2.subs({a:0, t:2*pi}) == 0:
732         checkFile.write('pSE2 at a = 0, t = 2pi: OK\n')
733 else:
734         checkFile.write('pSE2 at a = 0, t = 2pi: Incorrect\n')
735 checkFile.close()
736
737
738 # And print to terminal
739 #for i in range(len(allComps)):
740 #       if not (eval(allComps[i][0]).subs(allComps[i][2]) - eval(allComps[i][1]).subs(allComps[i][2])).\
        simplify() == 0:
741 #               print allComps[i][0] + ' and ' + allComps[i][1]+': Incorrect\n'
742
743
744 ####################################################################
745 ### Define a a function that calculates p bar answer.          ####
746 ####################################################################
747
748 def calcP(A, T, R):
749   assert (A <= 2*pi and A >= 0), "a is out of bounds. Should be in 0<a<2*pi"
750   assert (T <= 2*pi and T >= 0), "s is out of bounds. Should be in 0<s<2*pi"
751
752   if A > pi:
753     if A < 4*pi - 2* T:
754       p = pNW7.subs({a:A,  t:T, r:R}).n()
755     elif A <= 3*pi - T:
756                         p = pNE2.subs({a:A,  t:T, r:R}).n()
757     else:
758                         p = pNE1.subs({a:A,  t:T, r:R}).n()
759   else:
760     if A < 4*pi - 2* T:
761                         p = pSE3.subs({a:A,  t:T, r:R}).n()
762     else:
763                         p = pSE2.subs({a:A,  t:T, r:R}).n()
```

```
764         return p
765
766
767 #############################
768 ## Apply to entire grid   ###
769 #############################
770
771 # How many values for each parameter
772 nParas = 100
773
774 # Make a vector for a and s. Make an empty nParas x nParas array.
775 # Calculated profile sizes will go in pArray
776 tVec = np.linspace(0, 2*pi, nParas)
777 aVec = np.linspace(0, 2*pi, nParas)
778 pArray = np.zeros((nParas,nParas))
779
780 # Calculate profile size for each combination of parameters
781 for i in range(nParas):
782         for j in range(nParas):
783                 pArray[i][j] = calcP(aVec[i], tVec[j], 1)
784
785 # Turn the array upside down so origin is at bottom left.
786 pImage = np.flipud(pArray)
787
788 # Plot and save.
789 pl.imshow(pImage, interpolation='none', cmap=pl.get_cmap('Blues') )
790
791 # Show or save image.
792 # pl.show()
793 # pl.savefig('/imgs/profilesCalculated.png')
794
795
796
797 #############################
798 #### Output R function.  ###
799 #############################
800
801 # To reduce mistakes, output R function directly from python.
802 # However, the if statements, which correspond to the bounds of each model, are not automatic.
803
804 Rfunc = open('supplementaryRscript.R', 'w')
805
806 Rfunc.write("""
807 # S4
808 # Supplementary R script from
809 # A generalised random encounter model for estimating animal density with remote sensor data
810 # Tim C.D. Lucas, Elizabeth A. Moorcroft, Robin Freeman, Marcus J. Rowcliffe, Kate E. Jones
811 #
812 # calcDensity is the main function to calculate density.
813 # It takes parameters z, alpha, theta, r, animalSpeed, t
814 # z - The number of camera/acoustic counts or captures.
815 # alpha - Call width in radians.
816 # theta - Sensor width in radians.
817 # r - Sensor range in metres.
818 # animalSpeed - Average animal speed in metres per second.
819 # t - Length of survey in sensor seconds i.e. number of sensors x survey duration.
820 #
821 # calcAbundance calculates abundance rather than density and requires an extra parameter
822 # area - In metres squared. The size of the region being examined.
823
824
825 # Internal function to calculate profile width as described in the text
826 calcProfileWidth <- function(alpha, theta, r){
827         if(alpha > 2*pi | alpha < 0)
828             stop('alpha is out of bounds. alpha should be in interval 0<a<2*pi')
829         if(theta > 2*pi | theta < 0)
830             stop('theta is out of bounds. theta should be in interval 0<a<2*pi')
831
832   if(alpha > pi){
833         if(alpha < 4*pi - 2*theta){
834 """ +
835 '             p <- ' + str(pNW7) +
836 '\n                } else if(alpha <= 3*pi - theta){'
837 '\n                     p <- ' + str(pNE2) +
838 '\n                } else {'
839 '\n                     p <- ' + str(pNE1) +
840 '\n                }'
841 '\n         } else {'
842 '\n          if(alpha < 4*pi - 2*theta){'
843 '\n                     p <- ' + str(pSE3) +
844 '\n      } else {'
845 '\n                     p <- ' + str(pSE2) +
846 '\n                }'
847 '\n         }'
848 '\n       return(p)'
849 '\n}' +
850 """
```

```
851  # Calculate a population density. See above for units etc.
852  calcDensity <- function(z, alpha, theta, r, animalSpeed, t){
853          # Check the parameters are suitable.
854          if(z <= 0 | !is.numeric(z)) stop('Counts, z, must be a positive number.')
855          if(animalSpeed <= 0 | !is.numeric(animalSpeed)) stop('animalSpeed must be a positive number.')
856          if(t <= 0 | !is.numeric(t)) stop('Time, t, must be a positive number.')
857
858          # Calculate profile width, then density.
859          p <- calcProfileWidth(alpha, theta, r)
860          D <- z/{animalSpeed*t*p}
861          return(D)
862  }
863
864  # Calculate abundance rather than density.
865  calcAbundance <- function(z, alpha, theta, r, animalSpeed, t, area){
866          if(area <= 0 | !is.numeric(area)) stop('Area must be a positive number')
867          D <- calcDensity(z, alpha, theta, r, animalSpeed, t)
868          A <- D*area
869          return(A)
870  }
871  """
872  )
873
874  Rfunc.close()
```

supplementaryS3Python.py

S4. 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.

The script can also be found in the plain text file supplementaryS4R.R.

```r
# S4
# Supplementary R script from
# A generalised random encounter model for estimating animal density with remote sensor data
# Tim C.D. Lucas, Elizabeth A. 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.numeric(area)) stop('Area must be a positive number')
        D <- calcDensity(z, alpha, theta, r, animalSpeed, t)
        A <- D*area
        return(A)
}
```

supplementaryS4R.R

S5. Supplementary Information: Simulation model results of the gREM precision
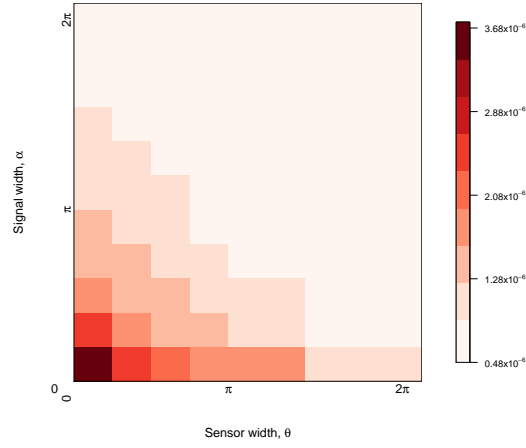

Figure S1. Simulation model results of the gREM precision given a range of sensor and signal widths, shown by the standard deviation of the error between the estimated and true densities. Standard deviations are shown from deep red to pink, representing high to low values between 0.483×10^{-6} to 3.74×10^{-6}.