

Bearing calculation of a single emitter sonar ring.

Damien Browne
Department of Electrical and Computer Systems Engineering
Monash University
Clayton Australia

May 31, 2011

Abstract

The equations required to localize reflectors for a single emitter sonar ring are not simple. This technical report shows the calculations required to localize a sonar target for the sonar ring proposed in [3]

An advantage of advanced sonar systems is that they produce accurate bearing as well as range measurements. In a structured environment, such as indoors, common features are edges, planes and corners. The classification and distinction of each of the three features requires measurements gathered from two different transmitter locations. Some advanced sonar systems use two transmitters to achieve this [2]. Other have used single transmitters and three receivers to classify reflectors [6]. It has been shown that the movement of a robot in an environment, in conjunction with an Extended Kalman Filter (EKF) is sufficient to identify the type of features [1]. Some work has used basic ranging sonar systems that do not produce bearing results in conjunction with robot movement to identify features [5]. The amplitude information from a received signal can be used in conjunction with a known model and a moving transmitter to classify a feature [4]. Methods of bearing calculation that rely on the amplitude of echoes are reliant on the structure of the reflector to match a known model within empirically determined bounds.

The design presented in this thesis has been constructed with the aim of using the sonar vector sensor method of bearing calculation. This method

has been shown to be accurate provided that accurate range measurements can be taken [2]. This requirement is the reason for the use of the matched filter method of time of flight estimation. The aim of this thesis is to produce an accurate sonar system that can be utilised in an existing SLAM framework such as the extended Kalman filter slam presented in [1]. Implementation of this system in the EKF framework is a subject of future work. The aim is to visualise these features and provide a framework for future SLAM implementations. To do this the equations that convert time of flight measurements to target location relative to the ring are needed. This technical report provides diagrams of the structure of the sonar ring and calculates the equations that localise each of the three most common sonar features. Features used here are planes, corners and edges. Corners are defined as the concave junction of two planes while edges are the convex junction of two planes. High curvature objects such as poles or chair legs are treated as edges.

1 Ring Geometry

The sonar ring is arranged such that pairs of receivers are intended to operate as sonar vector sensors [2]. In previous advanced sonar designs [2, 1, 6] at least one transducer of a pair would act as a transmitter as well as a receiver. The design of this ring is a complete implementation of the design published in [3]. A single near point source transmitter is located above a CNC machined conical parabolic reflector. The transmitter and reflector arrangement results in a near isotropic ultrasonic wavefront. Pairs of receivers are arranged in two tiers beneath the reflector. There is no advantage in the use of two tiers in the current design other than a reduction in radius of the sonar ring. A photograph of the sonar ring is shown in Figure 1.

A top down diagram of the sonar ring is shown in Figure 2. Pairs of receivers have been spaced at 7.5 degree intervals. The Polaroid 7000 series transducers can detect echoes from a bearing of up to 10 degrees from the normal to the surface of the transducer. The spacing of each pair of transducers allows for some overlap of echo detection between each vector sensor so that objects are detectable when moving around the sonar ring.

An elevation view of the sonar ring is shown in Figure 3. The vertical component of the sonar pulse wavefront's movement must also be taken into account. This vertical component is particularly influential at short ranges.

Given the structure of the sonar ring and the three main sonar feature

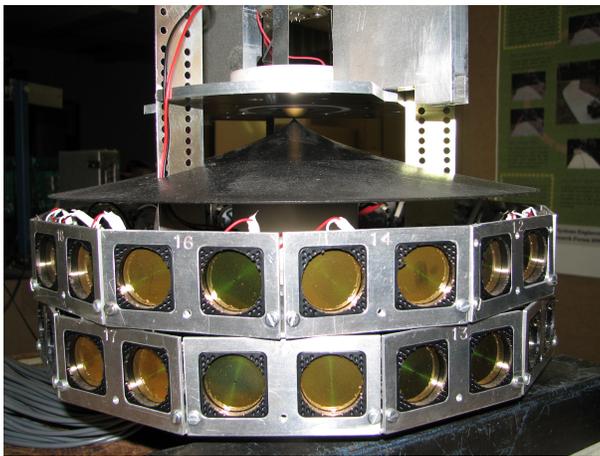


Figure 1: Photograph of the sonar ring.

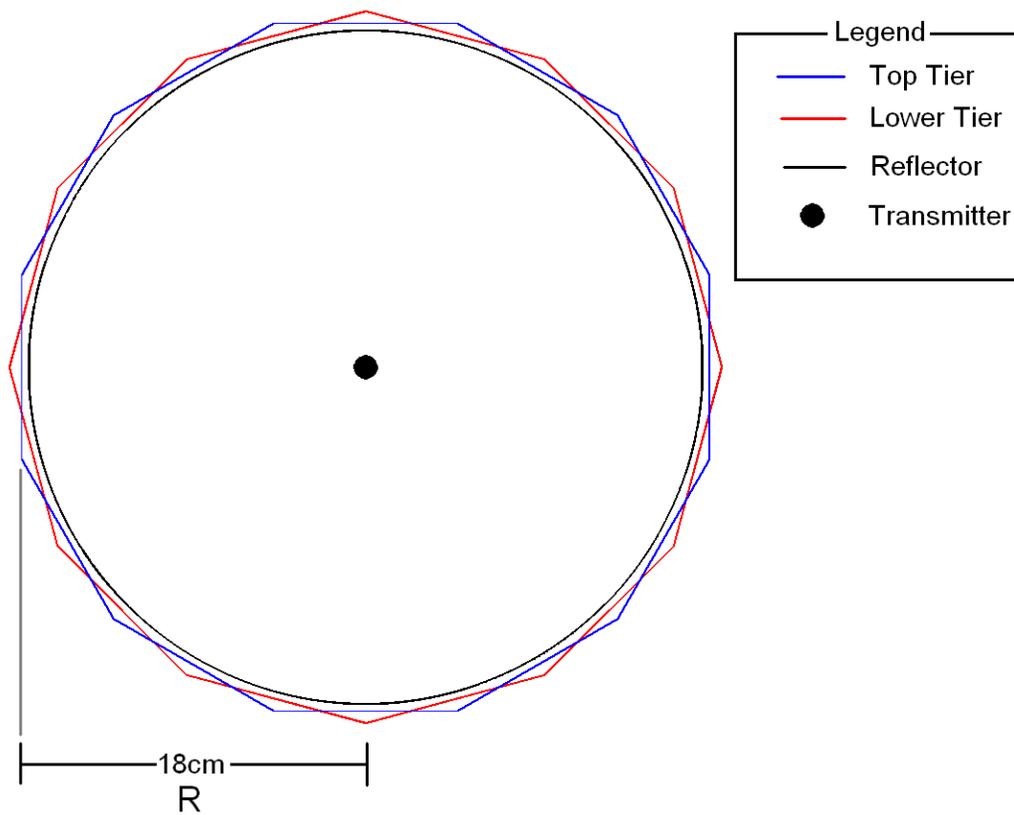


Figure 2: Top down view of the sonar ring.

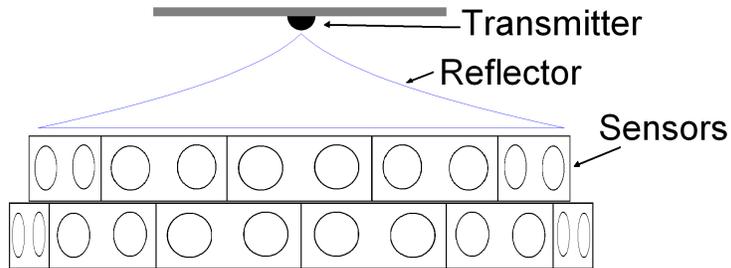


Figure 3: Elevation view of the sonar ring.

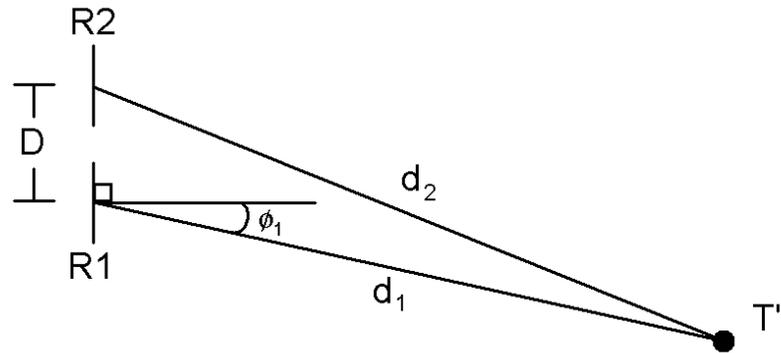


Figure 4: Receivers with virtual image geometry as seen from above.

types, the equations for target localisation can be calculated.

2 Transmitter Virtual Image

The first step in being able to localise a sonar target is to calculate the angle of reception of the sonar wave. This angle of reception can be calculated from the differences in distance of flight and the use of a virtual image [2]. The virtual image is common to both planes and corners but not edges. Localisation of the virtual image is also the same in both plane and corner features, however the localisation of the actual reflector is different in each case.

Given a plane or a corner feature, the angle of reception can be calculated. Figure 4 shows a diagram of the vector sensors localising a virtual image of the transmitter. From this diagram the angle of reception relative to one of the receivers can be calculated. In Figure 4 R1 and R2 are the two coplanar receivers. The distances d_1 and d_2 are the distances of flight calculated from the time of flight estimation, first published in [2]:

$$d_n = \frac{tof_n}{c} \quad (1)$$

where tof_n is the time of flight estimation for transducer R_n and c is the speed of sound in air. The distance D is the known separation between the centre points of the two transducers. T' is the virtual image of the transmitter and ϕ_1 is the angle between the normal of R_1 and the received echo. Applying the cosine rule to the triangle formed by R_1, R_2 and T' :

$$d_2^2 = D^2 + d_1^2 - 2Dd_1 \cos(\phi_1 + 90) \quad (2)$$

$$d_2^2 = D^2 + d_1^2 + 2Dd_1 \sin(\phi_1) \quad (3)$$

Solving for ϕ_1 yields:

$$\phi_1 = \sin^{-1} \left(\frac{d_2^2 - d_1^2 - D^2}{2d_1D} \right) \quad (4)$$

Factorising the numerator:

$$\frac{d_2^2 - d_1^2 - D^2}{2d_1D} = \frac{(d_2 - d_1)(d_2 + d_1)}{2d_1D} - \frac{D}{2d_1} \quad (5)$$

then when $d_1, d_2 \gg D$ then

$$\frac{d_2 + d_1}{2d_1} \rightarrow 1 \quad (6)$$

and

$$\frac{D}{2d_1} \rightarrow 0 \quad (7)$$

yielding the approximation:

$$\phi_1 \approx \sin^{-1} \left(\frac{d_2 - d_1}{D} \right) \quad (8)$$

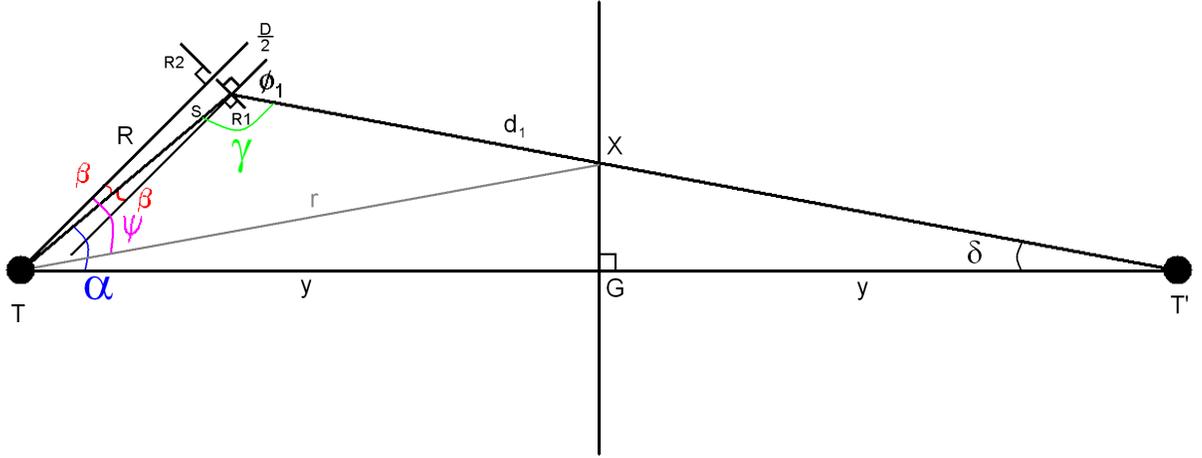


Figure 5: Geometry of a plane reflector.

Equation 4 can be used to calculate the bearing exactly or Equation 8 can be used if processing power is a concern. This approximation results in an error of approximately 2.3° at 0.5m range, 1.2° at 1m range and 0.39° at 3m range. While both plane and corner features share the same bearing to the virtual image, the localisation of the reflector is still dependent on the type of reflector. This is shown in the following sections.

3 Plane Features

So far the bearing to a virtual image has been calculated in Equation 4. Given a plane feature and a transmitter, a virtual image of the transmitter will appear behind the plane. A diagram of this is shown in Figure 5. The only point of a plane that is known to exist is the point of reflection. This point, shown as X in Figure 5, is the point that will be localised. Since a measurement was received on both receivers, there is a reflection point that exists for both receivers.

To find the point X in Figure 5 the triangle formed by the points X, G and T' must be found and by extension the triangle formed by X, G and T . To find this triangle the value of δ and y need to be found. This leads to first solving the triangle formed by $T, R1$ and T' . The triangle $T, R1, T'$ has sides of length S , d_1 and $2y$. The length d_1 is measured via Equation 1. The

length S is calculated from the triangle formed by T , R_1 and the midpoint between the two receivers is therefore:

$$S = \sqrt{\frac{D^2}{4} + R^2} \quad (9)$$

where R is the radius of the ring as shown in Figure 2 and D is still the distance between the two receivers.

At least one angle of the triangle T, R_1, T' is needed. The angle γ is simple to calculate in degrees:

$$\gamma = 180^\circ - \phi_1 + \beta \quad (10)$$

where β is calculated as:

$$\beta = \tan^{-1} \left(\frac{D}{2R} \right) \quad (11)$$

The cosine rule can now be applied to the triangle T, R_1, T'

$$4y^2 = d_1^2 + S^2 - 2d_1S \cos(\gamma) \quad (12)$$

and solving first for y :

$$y = \frac{\sqrt{d_1^2 + S^2 - 2d_1S \cos(\gamma)}}{2} \quad (13)$$

applying the sine rule:

$$\frac{\sin(\delta)}{S} = \frac{\sin(\gamma)}{2y} \quad (14)$$

and then solving for δ

$$\delta = \sin^{-1} \left(\frac{S \sin(\gamma)}{2y} \right) \quad (15)$$

and finally calculating r , the distance between T and X :

$$r = \frac{y}{\cos(\delta)} \quad (16)$$

The pair of receivers that is making the measurements is known and hence the global position angular offset of R is known. The angle ψ must be found:

$$\psi = \alpha - \delta + \beta \quad (17)$$

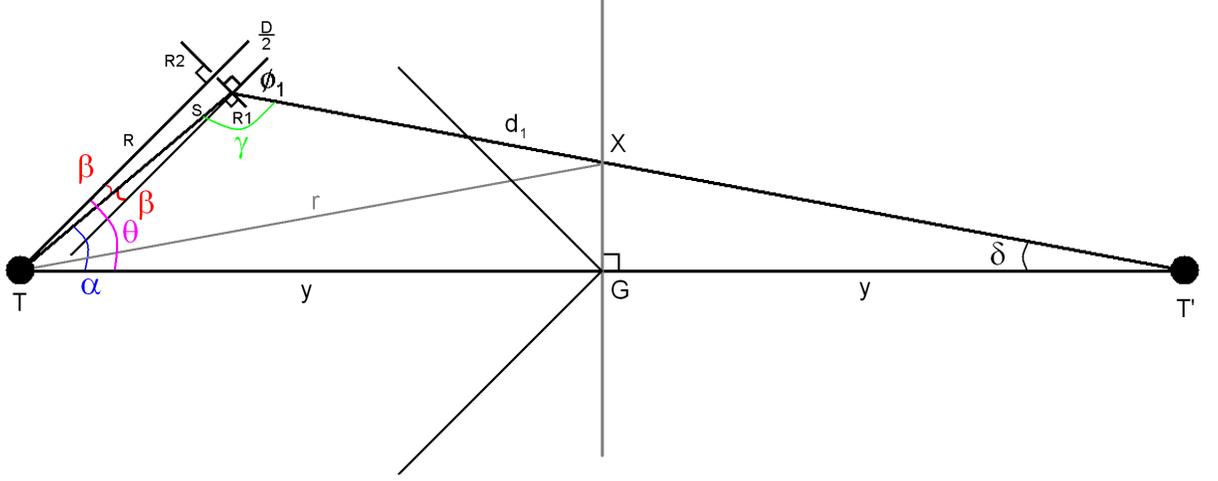


Figure 6: Geometry of a corner reflector.

where α is calculated as:

$$\alpha = \sin^{-1} \left(\frac{d_1 \sin(\gamma)}{2y} \right) \quad (18)$$

Using the value of ψ and the length r , the position X is now localised relative to the centre of the sonar ring. Alternatively it can be useful to localise the normal of the plane shown as point G . This point is localised in the following section. While planes are probably the most common feature encountered in an indoor environment there still remains the cases of the corners and edges to solve.

4 Corner Features

The geometry of a corner is very similar to that of a plane. Replicating the plane geometry for a corner using the same virtual image techniques as before yields the same triangles shown in Figure 6. Now the position of point G rather than X needs to be localised. Adding the new angle θ into the diagram:

$$\theta = \alpha + \beta \quad (19)$$

Using the value of θ and the length y , the position X is now localised

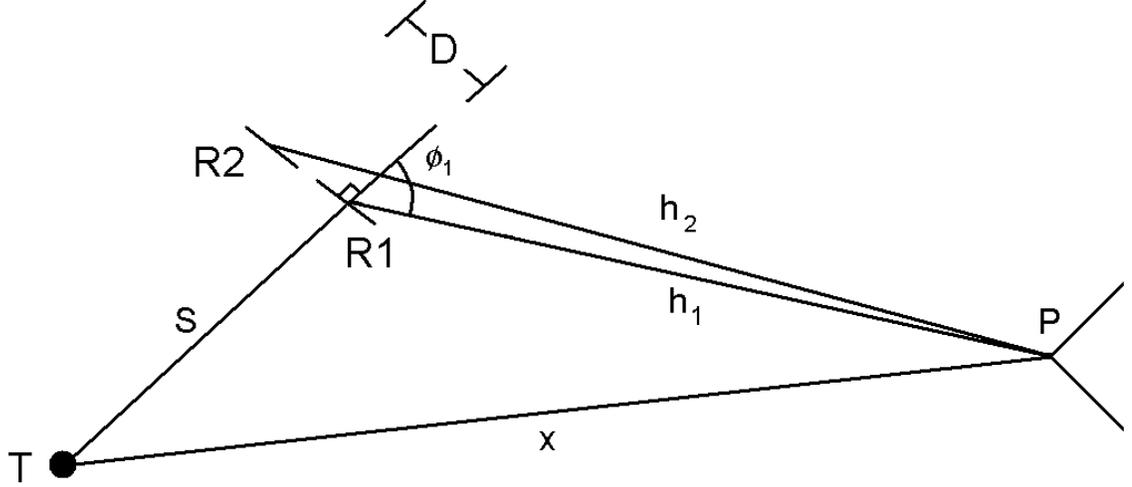


Figure 7: Geometry of an edge reflector.

relative to the centre of the sonar ring. Two features are now solved, however there remains the case of an edge feature.

5 Edge Features

Unlike the corner and plane geometry, edge features do not create a virtual image. No virtual image is created as the sound is re-radiated at the point where the edge exists. This presents a significant problem for the localisation methods used thus far. Figure 7 shows the geometry of a edge reflector.

From this diagram:

$$h_n = d_n - x \quad (20)$$

At this stage ϕ_1 and x are unknown. In systems where one receiver is also a transmitter it is trivial to calculate x as $x = \frac{d_n}{2}$, however this is not the case in this system.

Alternatively the geometry can be solved through the use of the circle equation and some boundary conditions. Figure 8 shows the geometry of the edge reflector using circle intersections to derive the geometry equations. The distance k is the distance between the transmitter and the edge. The

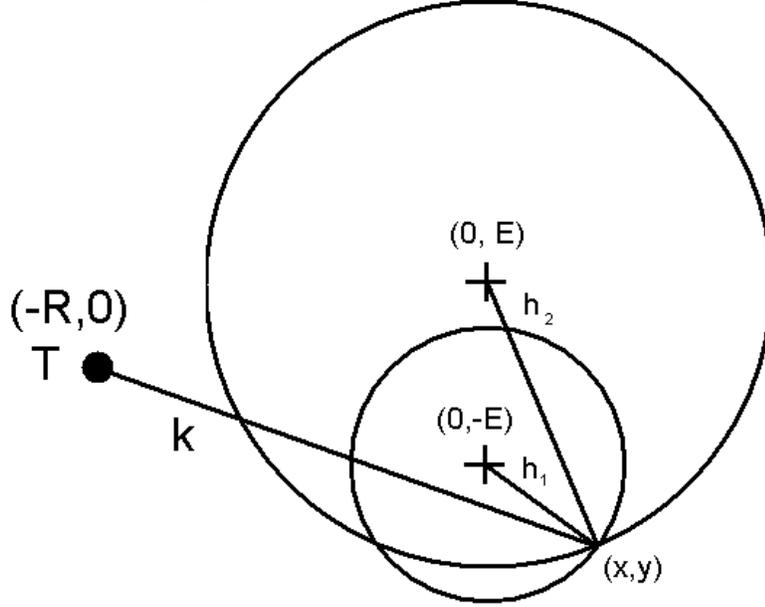


Figure 8: Geometry of an edge reflector with circles. (The origin is located between the two receivers and the receivers are oriented along the y axis).

distances h_1 and h_2 are the distances between receivers one and two and the edge. The values of x , y , k , h_1 and h_2 are unknown.

Using Figure 8 equations can be derived to solve for the unknowns in the diagram. The values of $k + h_1$ and $k + h_2$ are known as this is what the sonar ring measures. The sum of h_1 and k must equal the distance of flight as measured by the sonar ring and hence:

$$h_n = d_n - k; \quad (21)$$

The circle equations for $R1$ and $R2$ are therefore:

$$x^2 + (y + E)^2 = h_1^2 \quad (22)$$

$$x^2 + (y - E)^2 = h_2^2 \quad (23)$$

The distance between point T and point (x, y) must be consistent between both circles and therefore the equation

$$k^2 = (x + R)^2 + (y)^2 \quad (24)$$

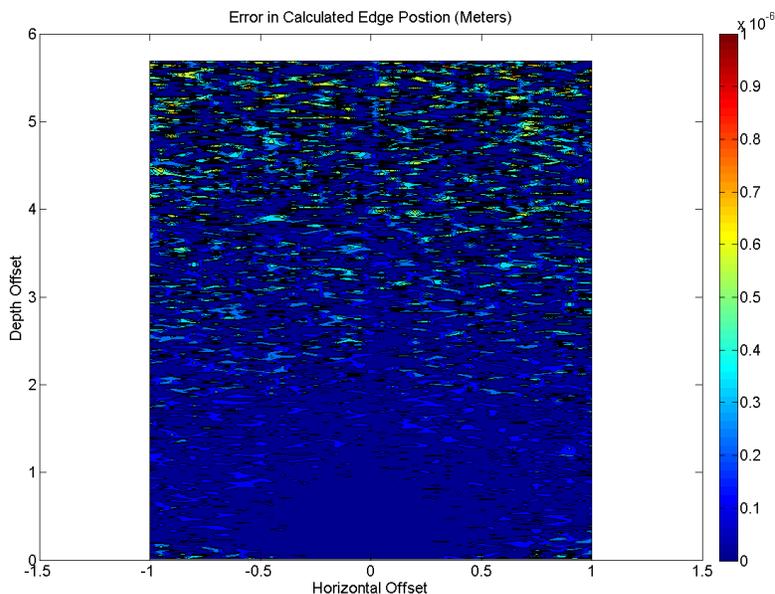


Figure 9: Error in calculated edge location.

must also be satisfied.

These three equations are sufficient to generate a unique solution (see Appendix A) provided that some additional information is included. Firstly all solutions must be real. Secondly, from the geometry in Figure 8 the value of x must always be positive. As it turns out this is sufficient to select the correct solution. As the solution to the simultaneous Equations 22, 23 and 24 is complicated, the solution was verified by exhaustively testing with simulated data within the visible range of one sensor pair. Using positions separated by one millimeter within the range of 0.01-5.7m and ± 1.0 meters left and right of the sensor pair, the distance of flight was calculated for each position. Each range of flight had its (x, y) position calculated from the distance of flight and the result was compared to the position from which the distances of flight were calculated. The result for the absolute error is shown in Figure 9. The error is such that it is unlikely the incorrect solution was selected from the two possibilities in the general solution. The remaining error can be explained as an accumulation of floating point errors due to the very complicated solution equations.

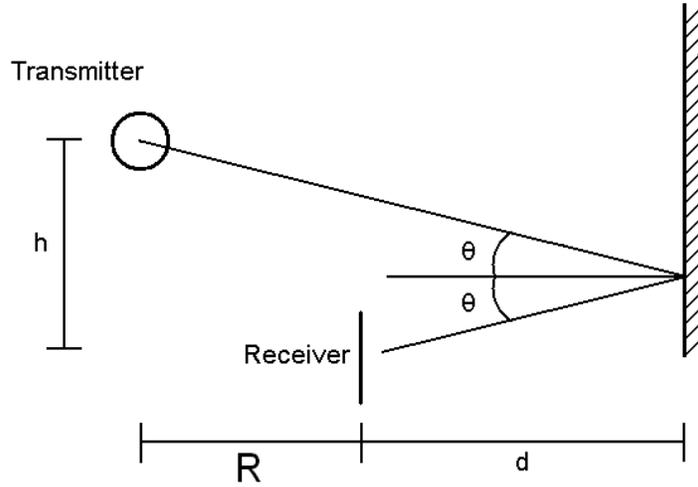


Figure 10: Vertical geometry of the sonar ring.

6 Vertical Geometry

Thus far the geometry of the sonar ring has been shown from a top down perspective. The transmitter is located significantly above the receivers. There remains a vertical component to the distances of flight that has not been accounted for. Figure 10 shows the geometry of the vertical component of the sonar ring.

The additional distance of flight incurred because this vertical offset is estimated using Figure 10. The difference between the distance covered from the transmitter to the receiver and the distance covered if there was not vertical offset is calculated and subtracted from the distances of flight. However, since the length of d (Figure 10) is not known beforehand, this is simply an estimate. The main drawback of using this estimate to resolve for range is that the two tiers of receivers become inconsistent with each other. That is, the distance of flight traveled for the lower tier is longer than the distance of flight for the upper tier when measuring an object at the same range. Ignoring the effect of horizontal positional changes, the effect of the height changes can be estimated.

Flipping the upper triangle in Figure 10 yields Figure 11 and the equation:

$$(R + 2d)^2 = t^2 - h^2 \quad (25)$$

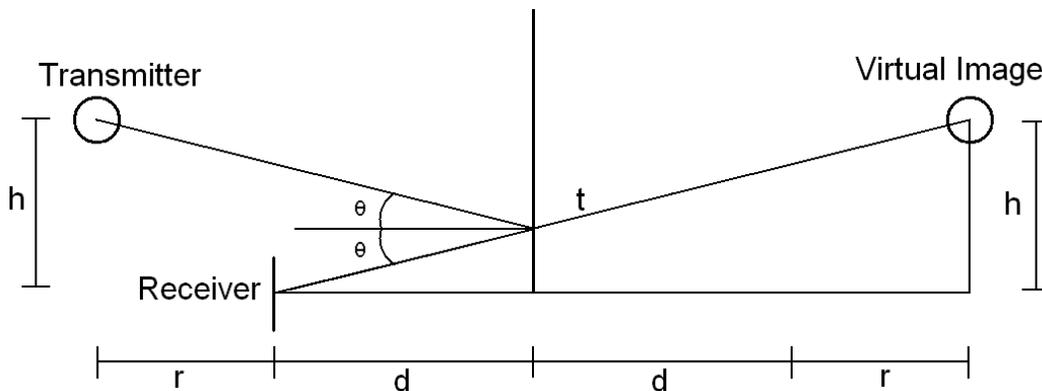


Figure 11: Vertical geometry of the sonar ring with virtual image.

Solving for d (since r , h and t are known and t is the distance measured by the sonar ring) yields an approximate correction for the vertical geometry of the sonar ring:

$$d = \frac{\sqrt{t^2 - h^2} - R}{2} \quad (26)$$

It should be noted that h is different for the two tiers of receivers. The correction and system as a whole always assumes that the object being ranged is planar in the vertical direction. A point feature (a corner or edge) in the vertical direction would cause an inaccurate correction.

7 Errors From Non Classification

As classification of features is beyond the scope of this thesis, it is worth examining the error in localisation of sonar reflectors due to non classification. It was decided that without classification all features should be localised using the corner model. The corner model was chosen as corners are a common point feature and point features are seen from many different positions. As the robot moves, the map appears the most consistent if the corner features are localised correctly. While plane features appear slightly inaccurate, this inaccuracy is only apparent at the ends of the plane feature. The error in localisation between an edge and a plane feature is therefore the distance between points X and G in Figures 5 and 6. This error was calculated for each possible location separated by 1mm. The absolute error results are shown in Figure 12.

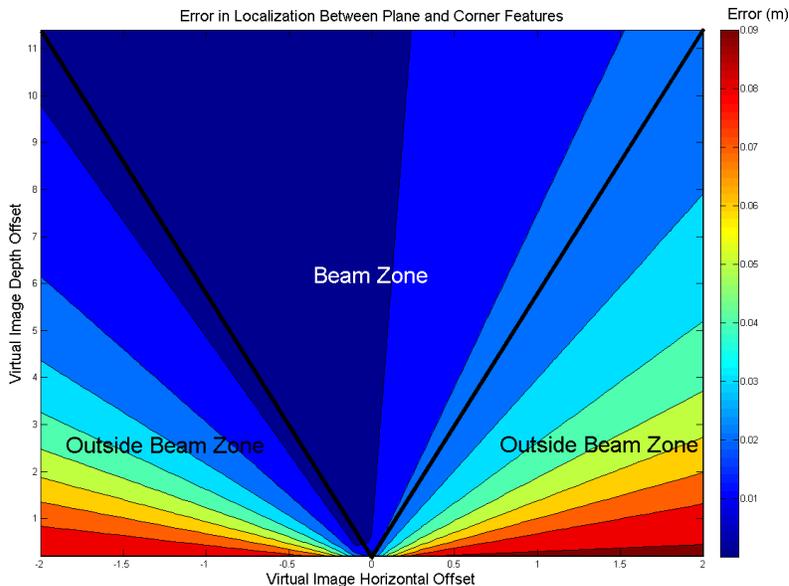


Figure 12: Difference in corner to plane location.

The difference in range becomes significant at higher angles. This is further motivation to restrict the bearing measurements. It is known that the power of the signal affects the variance of the range[2]. Longer range echoes are lower power echoes and therefore the variance in range becomes greater. Similarly higher bearing measurements are also low power. The slightly asymmetric nature of the graph is due to using the bearing referenced to one receiver.

The error in localisation due to edge features being incorrectly identified as corners is shown in Figure 13. Error within the beam zone due to this type of classification error is expected to be around 8mm.

8 Conclusions

In this technical report the equations required to localise three types of sonar features, namely planes, corners and edges are presented. The complexity of the exact edge solution may make classification of edges difficult without first finding a method to simplify the equations through approximation.

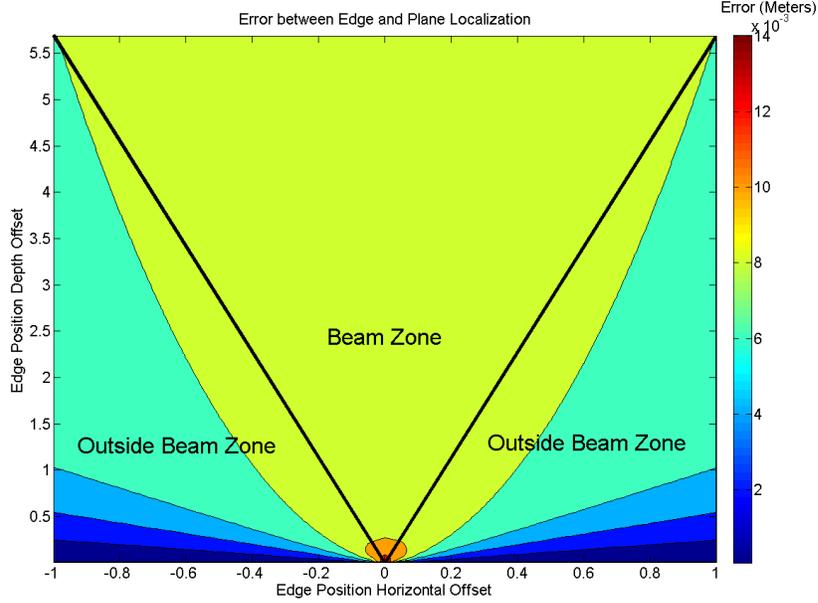


Figure 13: Difference in edge to corner location.

The errors due to non classification have been calculated and in most cases the error is no worse than approximately 2.5cm. The localisation equations presented here allow for visualisation of data.

A Edge Localisation Equation

In this appendix the solution to the edge localisation problem is presented. There are two possible solutions for x , y and h . Matlab's solve function was used to generate the result. The functionality of the solution was tested in the previous sections of this technical report. The correct answer of the alternatives is one that is real and in front of the receivers.

While these functions are long they are trivial to compute and the calculations used to generate Figure 9 completed in a fraction of the time it would have taken to gather the echoes. Therefore real time implementation of these equations is feasible.

$$x1 = 1/2/(8*d_2*d_1*R^2 - 4*d_2^2*R^2 - 4*d_1^2*R^2 - 8*E^2*d_2*d_1 + 16*R^2*E^2 - 4*E^2*d_2^2 - 4*E^2*d_1^2)*(-16*E^2*R^3 + 4*R^3*d_1^2 + 24*E^2*d_2*d_1*R - 4*E^2*d_1^2*$$

$$\begin{aligned}
& R - 4 * E^2 * d_2^2 * R + 8 * R * d_2^2 * d_1^2 + 16 * E^4 * R - 8 * R^3 * d_2 * d_1 + 4 * R^3 * d_2^2 - 4 * R * d_2^3 * \\
& d_1 - 4 * R * d_1^3 * d_2 + 4 * (4 * E^4 * R^4 * d_1^2 + 4 * E^4 * R^4 * d_2^2 - 6 * E^4 * d_1^4 * R^2 + 4 * E^8 * d_2^2 + \\
& E^4 * d_2^6 + E^4 * d_1^6 + 4 * E^8 * d_1^2 + d_2^6 * R^2 * E^2 - 8 * E^4 * d_2 * d_1^3 * R^2 - 8 * E^4 * d_2^3 * d_1 * R^2 - 4 * \\
& E^4 * d_2^2 * d_1^2 * R^2 - 6 * E^4 * d_2^4 * R^2 + d_1^6 * R^2 * E^2 + 3 * E^4 * d_2^4 * d_1^2 + 8 * E^4 * d_2^3 * d_1^3 + 8 * E^8 * \\
& d_2 * d_1 + 3 * E^4 * d_2^2 * d_1^4 - d_2^4 * E^2 * R^2 * d_1^2 + 16 * E^6 * d_2 * d_1 * R^2 + 8 * E^4 * R^4 * d_2 * d_1 - d_2^6 * \\
& E^2 * d_1^2 + 2 * d_2^2 * E^2 * d_1^4 - 5 * E^6 * d_2^4 - 5 * E^6 * d_1^4 + 2 * R^4 * E^2 * d_2^2 * d_1^2 - R^2 * E^2 * d_2^2 * d_1^4 + \\
& 8 * E^6 * d_2^2 * R^2 + 8 * E^6 * d_1^2 * R^2 - 8 * E^6 * d_1^3 * d_2 - 8 * E^6 * d_2^3 * d_1 - 6 * E^6 * d_2^2 * d_1^2 - R^4 * \\
& E^2 * d_2^4 - R^4 * E^2 * d_1^4 - E^2 * d_1^6 * d_2^2)(1/2)); x2 = 1/2/(8 * d_2 * d_1 * R^2 - 4 * d_2^2 * R^2 - 4 * \\
& d_1^2 * R^2 - 8 * E^2 * d_2 * d_1 + 16 * R^2 * E^2 - 4 * E^2 * d_2^2 - 4 * E^2 * d_1^2) * (-16 * E^2 * R^3 + 4 * R^3 * \\
& d_1^2 + 24 * E^2 * d_2 * d_1 * R - 4 * E^2 * d_1^2 * R - 4 * E^2 * d_2^2 * R + 8 * R * d_2^2 * d_1^2 + 16 * E^4 * R - 8 * \\
& R^3 * d_2 * d_1 + 4 * R^3 * d_2^2 - 4 * R * d_2^3 * d_1 - 4 * R * d_1^3 * d_2 - 4 * (4 * E^4 * R^4 * d_1^2 + 4 * E^4 * R^4 * \\
& d_2^2 - 6 * E^4 * d_1^4 * R^2 + 4 * E^8 * d_2^2 + E^4 * d_2^6 + E^4 * d_1^6 + 4 * E^8 * d_1^2 + d_2^6 * R^2 * E^2 - 8 * E^4 * \\
& d_2 * d_1^3 * R^2 - 8 * E^4 * d_2^3 * d_1 * R^2 - 4 * E^4 * d_2^2 * d_1^2 * R^2 - 6 * E^4 * d_2^4 * R^2 + d_1^6 * R^2 * E^2 + \\
& 3 * E^4 * d_2^4 * d_1^2 + 8 * E^4 * d_2^3 * d_1^3 + 8 * E^8 * d_2 * d_1 + 3 * E^4 * d_2^2 * d_1^4 - d_2^4 * E^2 * R^2 * d_1^2 + 16 * \\
& E^6 * d_2 * d_1 * R^2 + 8 * E^4 * R^4 * d_2 * d_1 - d_2^6 * E^2 * d_1^2 + 2 * d_2^4 * E^2 * d_1^4 - 5 * E^6 * d_2^4 - 5 * E^6 * \\
& d_1^4 + 2 * R^4 * E^2 * d_2^2 * d_1^2 - R^2 * E^2 * d_2^2 * d_1^4 + 8 * E^6 * d_2^2 * R^2 + 8 * E^6 * d_1^2 * R^2 - 8 * E^6 * d_1^3 * \\
& d_2 - 8 * E^6 * d_2^3 * d_1 - 6 * E^6 * d_2^2 * d_1^2 - R^4 * E^2 * d_2^4 - R^4 * E^2 * d_1^4 - E^2 * d_1^6 * d_2^2)(1/2)); \\
& y1 = -1/2 * (d_2^2 * d_1 - d_2 * R / (8 * d_2 * d_1 * R^2 - 4 * d_2^2 * R^2 - 4 * d_1^2 * R^2 - 8 * E^2 * d_2 * \\
& d_1 + 16 * R^2 * E^2 - 4 * E^2 * d_2^2 - 4 * E^2 * d_1^2)) * (-16 * E^2 * R^3 + 4 * R^3 * d_1^2 + 24 * E^2 * d_2 * \\
& d_1 * R - 4 * E^2 * d_1^2 * R - 4 * E^2 * d_2^2 * R + 8 * R * d_2^2 * d_1^2 + 16 * E^4 * R - 8 * R^3 * d_2 * d_1 + 4 * \\
& R^3 * d_2^2 - 4 * R * d_2^3 * d_1 - 4 * R * d_1^3 * d_2 + 4 * (4 * E^4 * R^4 * d_1^2 + 4 * E^4 * R^4 * d_2^2 - 6 * E^4 * d_1^4 * \\
& R^2 + 4 * E^8 * d_2^2 + E^4 * d_2^6 + E^4 * d_1^6 + 4 * E^8 * d_1^2 + d_2^6 * R^2 * E^2 - 8 * E^4 * d_2 * d_1^3 * R^2 - 8 * \\
& E^4 * d_2^3 * d_1 * R^2 - 4 * E^4 * d_2^2 * d_1^2 * R^2 - 6 * E^4 * d_2^4 * R^2 + d_1^6 * R^2 * E^2 + 3 * E^4 * d_2^4 * d_1^2 + 8 * \\
& E^4 * d_2^3 * d_1^3 + 8 * E^8 * d_2 * d_1 + 3 * E^4 * d_2^2 * d_1^4 - d_2^4 * E^2 * R^2 * d_1^2 + 16 * E^6 * d_2 * d_1 * R^2 + 8 * \\
& E^4 * R^4 * d_2 * d_1 - d_2^6 * E^2 * d_1^2 + 2 * d_2^4 * E^2 * d_1^4 - 5 * E^6 * d_2^4 - 5 * E^6 * d_1^4 + 2 * R^4 * E^2 * d_2^2 * \\
& d_1^2 - R^2 * E^2 * d_2^2 * d_1^4 + 8 * E^6 * d_2^2 * R^2 + 8 * E^6 * d_1^2 * R^2 - 8 * E^6 * d_1^3 * d_2 - 8 * E^6 * d_2^3 * d_1 - \\
& 6 * E^6 * d_2^2 * d_1^2 - R^4 * E^2 * d_2^4 - R^4 * E^2 * d_1^4 - E^2 * d_1^6 * d_2^2)(1/2)) - d_2 * d_1^2 + E^2 * d_2 - R^2 * \\
& d_2 + d_1 * R / (8 * d_2 * d_1 * R^2 - 4 * d_2^2 * R^2 - 4 * d_1^2 * R^2 - 8 * E^2 * d_2 * d_1 + 16 * R^2 * E^2 - 4 * \\
& E^2 * d_2^2 - 4 * E^2 * d_1^2) * (-16 * E^2 * R^3 + 4 * R^3 * d_1^2 + 24 * E^2 * d_2 * d_1 * R - 4 * E^2 * d_1^2 * R - \\
& 4 * E^2 * d_2^2 * R + 8 * R * d_2^2 * d_1^2 + 16 * E^4 * R - 8 * R^3 * d_2 * d_1 + 4 * R^3 * d_2^2 - 4 * R * d_2^3 * d_1 - \\
& 4 * R * d_1^3 * d_2 + 4 * (4 * E^4 * R^4 * d_1^2 + 4 * E^4 * R^4 * d_2^2 - 6 * E^4 * d_1^4 * R^2 + 4 * E^8 * d_2^2 + E^4 * \\
& d_2^6 + E^4 * d_1^6 + 4 * E^8 * d_1^2 + d_2^6 * R^2 * E^2 - 8 * E^4 * d_2 * d_1^3 * R^2 - 8 * E^4 * d_2^3 * d_1 * R^2 - 4 * \\
& E^4 * d_2^2 * d_1^2 * R^2 - 6 * E^4 * d_2^4 * R^2 + d_1^6 * R^2 * E^2 + 3 * E^4 * d_2^4 * d_1^2 + 8 * E^4 * d_2^3 * d_1^3 + 8 * E^8 * \\
& d_2 * d_1 + 3 * E^4 * d_2^2 * d_1^4 - d_2^4 * E^2 * R^2 * d_1^2 + 16 * E^6 * d_2 * d_1 * R^2 + 8 * E^4 * R^4 * d_2 * d_1 - \\
& d_2^6 * E^2 * d_1^2 + 2 * d_2^4 * E^2 * d_1^4 - 5 * E^6 * d_2^4 - 5 * E^6 * d_1^4 + 2 * R^4 * E^2 * d_2^2 * d_1^2 - R^2 * E^2 * d_2^2 * \\
& d_1^4 + 8 * E^6 * d_2^2 * R^2 + 8 * E^6 * d_1^2 * R^2 - 8 * E^6 * d_1^3 * d_2 - 8 * E^6 * d_2^3 * d_1 - 6 * E^6 * d_2^2 * d_1^2 - \\
& R^4 * E^2 * d_2^4 - R^4 * E^2 * d_1^4 - E^2 * d_1^6 * d_2^2)(1/2)) - d_1 * E^2 + R^2 * d_1) / (d_1 + d_2) / E; y2 = \\
& -1/2 * (d_2^2 * d_1 - d_2 * R / (8 * d_2 * d_1 * R^2 - 4 * d_2^2 * R^2 - 4 * d_1^2 * R^2 - 8 * E^2 * d_2 * d_1 + 16 *
\end{aligned}$$

$$\begin{aligned}
& R^2 * E^2 - 4 * E^2 * d_2^2 - 4 * E^2 * d_1^2) * (-16 * E^2 * R^3 + 4 * R^3 * d_1^2 + 24 * E^2 * d_2 * d_1 * R - \\
& 4 * E^2 * d_1^2 * R - 4 * E^2 * d_2^2 * R + 8 * R * d_2^2 * d_1^2 + 16 * E^4 * R - 8 * R^3 * d_2 * d_1 + 4 * R^3 * d_2^2 - \\
& 4 * R * d_2^3 * d_1 - 4 * R * d_1^3 * d_2 - 4 * (4 * E^4 * R^4 * d_1^2 + 4 * E^4 * R^4 * d_2^2 - 6 * E^4 * d_1^4 * R^2 + 4 * \\
& E^8 * d_2^2 + E^4 * d_1^6 + E^4 * d_1^6 + 4 * E^8 * d_1^2 + d_2^6 * R^2 * E^2 - 8 * E^4 * d_2 * d_1^3 * R^2 - 8 * E^4 * d_2^3 * \\
& d_1 * R^2 - 4 * E^4 * d_2^2 * d_1^2 * R^2 - 6 * E^4 * d_2^2 * R^2 + d_1^6 * R^2 * E^2 + 3 * E^4 * d_2^4 * d_1^2 + 8 * E^4 * \\
& d_2^3 * d_1^3 + 8 * E^8 * d_2 * d_1 + 3 * E^4 * d_2^2 * d_1^4 - d_2^4 * E^2 * R^2 * d_1^2 + 16 * E^6 * d_2 * d_1 * R^2 + 8 * E^4 * \\
& R^4 * d_2 * d_1 - d_2^6 * E^2 * d_1^2 + 2 * d_2^4 * E^2 * d_1^4 - 5 * E^6 * d_2^4 - 5 * E^6 * d_1^4 + 2 * R^4 * E^2 * d_2^2 * d_1^2 - \\
& R^2 * E^2 * d_2^2 * d_1^4 + 8 * E^6 * d_2^2 * R^2 + 8 * E^6 * d_1^2 * R^2 - 8 * E^6 * d_1^3 * d_2 - 8 * E^6 * d_2^3 * d_1 - 6 * \\
& E^6 * d_2^2 * d_1^2 - R^4 * E^2 * d_2^4 - R^4 * E^2 * d_1^4 - E^2 * d_1^6 * d_2^2) (1/2)) - d_2 * d_1^2 + E^2 * d_2 - R^2 * \\
& d_2 + d_1 * R / (8 * d_2 * d_1 * R^2 - 4 * d_2^2 * R^2 - 4 * d_1^2 * R^2 - 8 * E^2 * d_2 * d_1 + 16 * R^2 * E^2 - 4 * \\
& E^2 * d_2^2 - 4 * E^2 * d_1^2) * (-16 * E^2 * R^3 + 4 * R^3 * d_1^2 + 24 * E^2 * d_2 * d_1 * R - 4 * E^2 * d_1^2 * \\
& R - 4 * E^2 * d_2^2 * R + 8 * R * d_2^2 * d_1^2 + 16 * E^4 * R - 8 * R^3 * d_2 * d_1 + 4 * R^3 * d_2^2 - 4 * R * d_2^3 * \\
& d_1 - 4 * R * d_1^3 * d_2 - 4 * (4 * E^4 * R^4 * d_1^2 + 4 * E^4 * R^4 * d_2^2 - 6 * E^4 * d_1^4 * R^2 + 4 * E^8 * d_2^2 + \\
& E^4 * d_1^6 + E^4 * d_1^6 + 4 * E^8 * d_1^2 + d_2^6 * R^2 * E^2 - 8 * E^4 * d_2 * d_1^3 * R^2 - 8 * E^4 * d_2^3 * d_1 * R^2 - \\
& 4 * E^4 * d_2^2 * d_1^2 * R^2 - 6 * E^4 * d_2^2 * R^2 + d_1^6 * R^2 * E^2 + 3 * E^4 * d_2^4 * d_1^2 + 8 * E^4 * d_2^3 * d_1^3 + 8 * \\
& E^8 * d_2 * d_1 + 3 * E^4 * d_2^2 * d_1^4 - d_2^4 * E^2 * R^2 * d_1^2 + 16 * E^6 * d_2 * d_1 * R^2 + 8 * E^4 * R^4 * d_2 * \\
& d_1 - d_2^6 * E^2 * d_1^2 + 2 * d_2^4 * E^2 * d_1^4 - 5 * E^6 * d_2^4 - 5 * E^6 * d_1^4 + 2 * R^4 * E^2 * d_2^2 * d_1^2 - R^2 * E^2 * \\
& d_2^2 * d_1^4 + 8 * E^6 * d_2^2 * R^2 + 8 * E^6 * d_1^2 * R^2 - 8 * E^6 * d_1^3 * d_2 - 8 * E^6 * d_2^3 * d_1 - 6 * E^6 * d_2^2 * \\
& d_1^2 - R^4 * E^2 * d_2^4 - R^4 * E^2 * d_1^4 - E^2 * d_1^6 * d_2^2) (1/2)) - d_1 * E^2 + R^2 * d_1) / (d_1 + d_2) / E; \\
& h1 = -1/2 * (-2 * R / (8 * d_2 * d_1 * R^2 - 4 * d_2^2 * R^2 - 4 * d_1^2 * R^2 - 8 * E^2 * d_2 * d_1 + 16 * \\
& R^2 * E^2 - 4 * E^2 * d_2^2 - 4 * E^2 * d_1^2) * (-16 * E^2 * R^3 + 4 * R^3 * d_1^2 + 24 * E^2 * d_2 * d_1 * R - 4 * \\
& E^2 * d_1^2 * R - 4 * E^2 * d_2^2 * R + 8 * R * d_2^2 * d_1^2 + 16 * E^4 * R - 8 * R^3 * d_2 * d_1 + 4 * R^3 * d_2^2 - 4 * R * \\
& d_2^3 * d_1 - 4 * R * d_1^3 * d_2 + 4 * (4 * E^4 * R^4 * d_1^2 + 4 * E^4 * R^4 * d_2^2 - 6 * E^4 * d_1^4 * R^2 + 4 * E^8 * d_2^2 + \\
& E^4 * d_1^6 + E^4 * d_1^6 + 4 * E^8 * d_1^2 + d_2^6 * R^2 * E^2 - 8 * E^4 * d_2 * d_1^3 * R^2 - 8 * E^4 * d_2^3 * d_1 * R^2 - 4 * \\
& E^4 * d_2^2 * d_1^2 * R^2 - 6 * E^4 * d_2^2 * R^2 + d_1^6 * R^2 * E^2 + 3 * E^4 * d_2^4 * d_1^2 + 8 * E^4 * d_2^3 * d_1^3 + 8 * E^8 * \\
& d_2 * d_1 + 3 * E^4 * d_2^2 * d_1^4 - d_2^4 * E^2 * R^2 * d_1^2 + 16 * E^6 * d_2 * d_1 * R^2 + 8 * E^4 * R^4 * d_2 * d_1 - d_2^6 * \\
& E^2 * d_1^2 + 2 * d_2^4 * E^2 * d_1^4 - 5 * E^6 * d_2^4 - 5 * E^6 * d_1^4 + 2 * R^4 * E^2 * d_2^2 * d_1^2 - R^2 * E^2 * d_2^2 * d_1^4 + \\
& 8 * E^6 * d_2^2 * R^2 + 8 * E^6 * d_1^2 * R^2 - 8 * E^6 * d_1^3 * d_2 - 8 * E^6 * d_2^3 * d_1 - 6 * E^6 * d_2^2 * d_1^2 - R^4 * \\
& E^2 * d_2^4 - R^4 * E^2 * d_1^4 - E^2 * d_1^6 * d_2^2) (1/2)) - d_2^2 - d_1^2 + 2 * E^2 - 2 * R^2) / (d_1 + d_2); h2 = \\
& -1/2 * (-2 * R / (8 * d_2 * d_1 * R^2 - 4 * d_2^2 * R^2 - 4 * d_1^2 * R^2 - 8 * E^2 * d_2 * d_1 + 16 * R^2 * E^2 - \\
& 4 * E^2 * d_2^2 - 4 * E^2 * d_1^2) * (-16 * E^2 * R^3 + 4 * R^3 * d_1^2 + 24 * E^2 * d_2 * d_1 * R - 4 * E^2 * d_1^2 * \\
& R - 4 * E^2 * d_2^2 * R + 8 * R * d_2^2 * d_1^2 + 16 * E^4 * R - 8 * R^3 * d_2 * d_1 + 4 * R^3 * d_2^2 - 4 * R * d_2^3 * \\
& d_1 - 4 * R * d_1^3 * d_2 + 4 * (4 * E^4 * R^4 * d_1^2 + 4 * E^4 * R^4 * d_2^2 - 6 * E^4 * d_1^4 * R^2 + 4 * E^8 * d_2^2 + \\
& E^4 * d_1^6 + E^4 * d_1^6 + 4 * E^8 * d_1^2 + d_2^6 * R^2 * E^2 - 8 * E^4 * d_2 * d_1^3 * R^2 - 8 * E^4 * d_2^3 * d_1 * R^2 - 4 * \\
& E^4 * d_2^2 * d_1^2 * R^2 - 6 * E^4 * d_2^2 * R^2 + d_1^6 * R^2 * E^2 + 3 * E^4 * d_2^4 * d_1^2 + 8 * E^4 * d_2^3 * d_1^3 + 8 * E^8 * \\
& d_2 * d_1 + 3 * E^4 * d_2^2 * d_1^4 - d_2^4 * E^2 * R^2 * d_1^2 + 16 * E^6 * d_2 * d_1 * R^2 + 8 * E^4 * R^4 * d_2 * d_1 - \\
& d_2^6 * E^2 * d_1^2 + 2 * d_2^4 * E^2 * d_1^4 - 5 * E^6 * d_2^4 - 5 * E^6 * d_1^4 + 2 * R^4 * E^2 * d_2^2 * d_1^2 - R^2 * E^2 * d_2^2 * \\
& d_1^4 + 8 * E^6 * d_2^2 * R^2 + 8 * E^6 * d_1^2 * R^2 - 8 * E^6 * d_1^3 * d_2 - 8 * E^6 * d_2^3 * d_1 - 6 * E^6 * d_2^2 * d_1^2 - \\
& R^4 * E^2 * d_2^4 - R^4 * E^2 * d_1^4 - E^2 * d_1^6 * d_2^2) (1/2)) - d_2^2 - d_1^2 + 2 * E^2 - 2 * R^2) / (d_1 + d_2) / E;
\end{aligned}$$

$$R^4 * E^2 * d_2^4 - R^4 * E^2 * d_1^4 - E^2 * d_1^6 * d_2^2)(1/2)) - d_2^2 - d_1^2 + 2 * E^2 - 2 * R^2) / (d_1 + d_2);$$

References

- [1] S. Fazli and L. Kleeman. Simultaneous landmark classification, localization and map building for an advanced sonar ring. *Robotica*, 25(03):283–296, 2007.
- [2] L. Kleeman and R. Kuc. Mobile robot sonar for target localization and classification. *The International Journal of Robotics Research*, 14:295–318, 1995.
- [3] L. Kleeman and A. Ohya. The design of a transmitter with a parabolic conical reflector for a sonar ring. *Australasian Conference on Robotics and Automation 2006*, 2006.
- [4] R. Kuc. Forward model for sonar maps produced with the polaroid ranging module. *Robotics and Automation, IEEE Transactions on*, 19(2):358–362, Apr. 2003.
- [5] S.-J. Lee, J.-H. Lim, and D.-W. Cho. General feature extraction for mapping and localization of a mobile robot using sparsely sampled sonar data. *Advanced Robotics*, 23:1601–1616(16), September 2009.
- [6] H. Peremans, K. Audenaert, and J. Van Campenhout. A high-resolution sensor based on tri-aural perception. *Robotics and Automation, IEEE Transactions on*, 9(1):36–48, Feb. 1993.