Odometry Error Covariance Estimation for Two Wheel Robot Vehicles

Lindsay KLEEMAN
Intelligent Robotics Research Centre
Department of Electrical and Computer Systems Engineering
Monash University

Technical Report MECSE-95-1
1995

ABSTRACT

This technical report develops a simple statistical error model for estimating position and orientation of a mobile robot using odometry. Once the errors are characterised, other sensor data can be combined sensibly in the estimation of position, using the Extended Kalman Filter [Kleeman, 1992 #100; Jazwinski, 1970 #117]. A closed form error covariance matrix is developed for (i) straight lines and (ii) constant curvature arcs and (iii) turning about the centre of axle of the robot. Other paths can be composed of short segments of constant curvature arcs without great loss of accuracy. The model assumes that wheel distance measurement errors are exclusively random zero mean white noise. Systematic errors due to wheel radius and wheel base measurement are ignored, since these can be removed by calibration. Previous work on developing odometry covariance relies on incrementally updating the covariance matrix in small times steps. The approach taken here integrates the noise theoretically over the entire path length to produce simple closed form expressions, allowing efficient covariance matrix updating after the completion of path segments.

1. Introduction

Odometry is the measurement of wheel rotation. When odometry measurements are known as a function of time from two wheels on a common axle, the position and orientation of the centre of the axle can be determined as a function of time. Practical implementations of this approach where both time and wheel increments are discretised are used as one estimate of a mobile robot's position and orientation [Wang, 1988 #87]. With time, however, errors accumulate in odometry localisation, due to wheel slippage and measurement error in rotation and distance between wheels, and non-continuous sampling of wheel increments. It is therefore necessary to use in addition some external referencing of position to correct for accumulating odometry errors.

This technical report develops a simple statistical model for the errors in estimation of position and orientation of a mobile robot using odometry. Once the errors are characterised, other sensor data can be combined sensibly in the estimation of position, using the Extended Kalman Filter [Kleeman, 1992 #100; Jazwinski, 1970 #117]. A closed form error covariance matrix is developed for (i) straight lines and (ii) constant curvature arcs and (iii) turning about the centre of axle of the robot. Other paths can be composed of short segments of constant curvature arcs without great loss of accuracy. The model assumes that errors are exclusively random zero mean white noise in the measurement of distance travelled by each wheel. Systematic errors due to wheel radius and wheel base measurement are ignored, since these can in theory be removed by calibration. Previous work on developing odometry covariance relies on incrementally updating the covariance matrix in small times steps. The approach taken here integrates the noise theoretically over the entire path length to produce simple closed form expressions, allowing efficient covariance matrix updating after the completion of path segments.

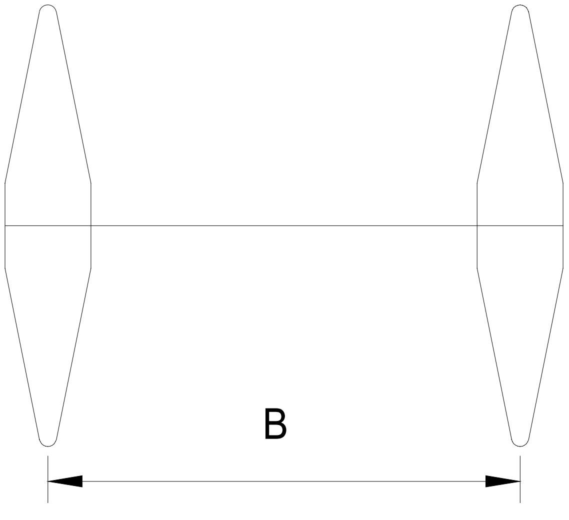

Figure 1 - Wheel Geometry.

2. Odometry Error Assumptions

The odometry error of a wheel incurred in one unit of travel is assumed to be randomly distributed with zero mean and to be independent of the odometry error incurred in the next or previous unit of travel. The variance of the cumulative error is then the sum of the variance of each independent segment and is therefore proportional to the distance travelled. The error variances for the left and right wheels σ_L^2 and σ_R^2 can then be expressed as:

$$\begin{aligned} \sigma_L^2 &= k_L^2 d_L \\ \sigma_R^2 &= k_R^2 d_R \end{aligned} \tag{1}$$

where d_L and d_R are the distances travelled by each wheel, and k_L^2 and k_R^2 are constants.

3. Straight Line Motion

The robot wheels are controlled to travel at the same speed as measured by wheel odometry with the aim of following a straight line path. In reality errors in odometry mean that the path deviates randomly from a straight line. As shown in figure 2, there are three errors in the path: (i) the error in distance travelled along the line, e_s, (ii) the error perpendicular to the line, e_p, and (iii) the error in heading angle e_θ. Each of these errors is characterised in terms of the basic odometry errors of each wheel. Initially, the errors are respectively $e_s(0)$, $e_p0)$ and $e_\theta(0)$ with an initial covariance matrix given by:

$$\mathbf{P_{sp}(0)} \equiv \begin{bmatrix} \sigma_s^2(0) & \sigma_{sp}(0) & \sigma_{s\theta}(0) \\ \sigma_{sp}(0) & \sigma_p^2(0) & \sigma_{p\theta}(0) \\ \sigma_{s\theta}(0) & \sigma_{p\theta}(0) & \sigma_\theta^2(0) \end{bmatrix} \tag{2}$$

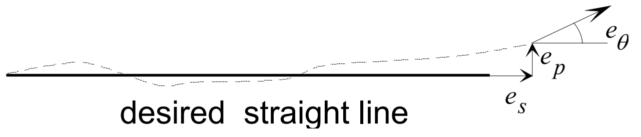

Figure 2 - Definition of Errors in Straight Line Path - the robot path is shown as the dashed line.

Below, the effect of propagating these errors and the introduction of new errors is derived for a straight line motion. The problem is tackled by breaking up the trajectory of the robot into discrete small segments where the robot travels a distance of Δd. The segments are indexed so that the robot angle error at the end of the kth segment is denoted by $e_\theta(k)$. The errors in odometry of the left and right wheels are denoted by e_L and e_R, and the distance between wheels by B as shown in Figure 1. The error introduced in θ after a segment Δd of distance is denoted by $e_{\Delta\theta}$. Therefore

$$\begin{aligned} e_\theta(k) &= e_\theta(k-1) + e_{\Delta\theta}(k) \\ &= e_\theta(0) + \sum_{i=1}^{k} e_{\Delta\theta}(i) \\ &= e_\theta(0) + \sum_{i=1}^{k}\left(\frac{e_R(i) - e_L(i)}{B}\right) \end{aligned} \tag{3}$$

From equation (3) and the assumption that odometry errors are independent, with respect to time, and with respect to wheels, the variance of the angle, σ_θ^2 is given by

$$\sigma_\theta^2(k) = \sigma_\theta^2(0) + |d|\left(\frac{k_L^2 + k_R^2}{B^2}\right) \tag{4}$$

where d is the total distance travelled and is given by $k\Delta d$. The perpendicular distance error is given approximately by

$$e_p(k) \cong e_p(k-1) + \Delta d e_\theta(k-1) \tag{5}$$

where the angle error is assumed to be small and constant throughout the segment. This approximation is good when angle errors are kept small and the segment size is small enough so that new angle errors introduced into the segment are insignificant compared to the current angle error. From equations (3) and (5) and rearranging

$$\begin{aligned} e_p(k) &= e_p(0) + \Delta d \sum_{j=1}^{k} e_\theta(j-1) \\ &= e_p(0) + \Delta d \sum_{j=1}^{k}\left(e_\theta(0) + \sum_{i=1}^{j-1} e_{\Delta\theta}(i)\right) \\ &= e_p(0) + k\Delta d e_\theta(0) + \Delta d \sum_{h=1}^{k-1}(k-h) e_{\Delta\theta}(h) \\ &= e_p(0) + d e_\theta(0) + \Delta d \sum_{g=1}^{k-1} g e_{\Delta\theta}(k-g) \end{aligned} \tag{6}$$

From the assumptions of independence of errors and equation (6), the variance of the perpendicular distance error, σ_p^2 is given by

$$\begin{aligned}\sigma_p^2(k) &= \sigma_p^2(0)+2d\sigma_{p\theta}(0)+d^2\sigma_\theta^2(0)+(\Delta d)^2\sum_{g=1}^{k-1} g^2\frac{k_L^2+k_R^2}{B^2}\Delta d \\ &= \sigma_p^2(0)+2d\sigma_{p\theta}(0)+d^2\sigma_\theta^2(0)+\frac{(\Delta d)^3(k_L^2+k_R^2)}{B^2}\sum_{g=1}^{k-1} g^2 \\ &= \sigma_p^2(0)+2d\sigma_{p\theta}(0)+d^2\sigma_\theta^2(0)+\frac{(\Delta d)^3(k_L^2+k_R^2)\big(2(k-1)^3+3(k-1)^2+(k-1)\big)}{6B^2}\end{aligned}\tag{7}$$

Note that the expression for the sum of integer squares can be proven by mathematical induction on *k*. In the limit as *Δd* approaches zero and *kΔd* is the total distance of travel *d*, the expression in equation (7) reduces to

$$\sigma_p^2 = \sigma_p^2(0)+2d\sigma_{p\theta}(0)+d^2\sigma_\theta^2(0)+\frac{|d|d^2(k_L^2+k_R^2)}{3B^2}\tag{8}$$

The variance of the perpendicular error is now seen as dependent on the cube of distance rather than just distance as is the case for the angle error. This can be understood by observing that the perpendicular errors accumulate from an ever increasing angle error translated into perpendicular distance error by the movement.

The error in distance straight along the path for the *k*th segment, $e_S(k)$ is given by

$$\begin{aligned}e_s(k) &= e_s(k-1)+\cos(e_\theta(k-1))\frac{e_L(k)+e_R(k)}{2} \\ &\cong e_s(k-1)+\frac{e_L(k)+e_R(k)}{2} \\ &= e_s(0)+\tfrac{1}{2}\sum_{i=1}^{k}\big(e_L(i)+e_R(i)\big)\end{aligned}\tag{9}$$

where the approximation is valid for small heading angle errors. The corresponding variance is then

$$\sigma_s^2(k) = \sigma_s^2(0)+|d|\frac{k_L^2+k_R^2}{4}\tag{10}$$

The off-diagonal covariances are now evaluated. From equations (3) and (9)

$$\begin{aligned}\sigma_{s\theta} &= E\left[e_s e_\theta\right] \\ &= E\left[\left(e_\theta(0)+\sum_{i=1}^{k}\left(\frac{e_R(i)-e_L(i)}{B}\right)\right)\left(e_s(0)+\tfrac{1}{2}\sum_{i=1}^{k}\left(e_L(i)+e_R(i)\right)\right)\right] \\ &= \sigma_{s\theta}(0)+E\left[\sum_{i=1}^{k}\left(\frac{e_R^2(i)-e_L^2(i)}{2B}\right)\right] \\ &= \sigma_{s\theta}(0)+|d|\left(\frac{k_R^2-k_L^2}{2B}\right)\end{aligned} \tag{11}$$

and from equations (9) and (6)

$$\begin{aligned}\sigma_{sp} &= E\left[\left(e_s(0)+\tfrac{1}{2}\sum_{i=1}^{k}\left(e_L(i)+e_R(i)\right)\right)\left(e_p(0)+de_\theta(0)+\Delta d\sum_{h=1}^{k-1}(k-h)e_{\Delta\theta}(h)\right)\right] \\ &= \sigma_{sp}(0)+d\sigma_{s\theta}(0)+E\left[\frac{\Delta d}{2B}\sum_{i=1}^{k-1}(k-i)\left(e_R^2(i)-e_L^2(i)\right)\right] \\ &= \sigma_{sp}(0)+d\sigma_{s\theta}(0)+\frac{\Delta d}{2B}\sum_{i=1}^{k-1}(k-i)\left(\Delta dk_R^2-\Delta dk_L^2\right) \\ &= \sigma_{sp}(0)+d\sigma_{s\theta}(0)+\frac{\Delta d^2(k-1)k\left(k_R^2-k_L^2\right)}{4B}\end{aligned} \tag{12}$$

In the limit as *k* approaches infinity

$$\sigma_{sp} = \sigma_{sp}(0)+d\sigma_{s\theta}(0)+\frac{d^2\left(k_R^2-k_L^2\right)}{4B} \tag{13}$$

From equations (9) and (3)

$$\begin{aligned}\sigma_{p\theta} &= E\left[\left(e_p(0)+de_\theta(0)+\Delta d\sum_{h=1}^{k-1}(k-h)\left(\frac{e_R(h)-e_L(h)}{B}\right)\right)\left(e_\theta(0)+\sum_{i=1}^{k}\left(\frac{e_R(i)-e_L(i)}{B}\right)\right)\right] \\ &= \sigma_{p\theta}(0)+d\sigma_\theta^2(0)+E\left[\frac{\Delta d}{B^2}\sum_{i=1}^{k-1}(k-i)\left(e_R^2(i)+e_L^2(i)\right)\right] \\ &= \sigma_{p\theta}(0)+d\sigma_\theta^2(0)+\frac{\Delta d}{B^2}\sum_{i=1}^{k-1}(k-i)\left(\Delta dk_R^2+\Delta dk_L^2\right) \\ &= \sigma_{p\theta}(0)+d\sigma_\theta^2(0)+\frac{\Delta d^2(k-1)k\left(k_R^2+k_L^2\right)}{2B^2}\end{aligned} \tag{14}$$

In the limit as *k* approaches infinity

$$\sigma_{p\theta}=\sigma_{p\theta}(0)+d\sigma_\theta^2(0)+\frac{|d|d\left(k_R^2+k_L^2\right)}{2B^2} \tag{15}$$

In summary, the standard deviation of angle error and distance along the path increase with the square root of distance while the perpendicular standard deviation increase as a power 1.5 with distance. The constants of proportionality all can be related back to the basic odometry error statistics on each wheel. If both wheels have the same error statistics, then only the perpendicular distance error and angle errors are correlated with a constant correlation coefficient of $\sigma_{p\theta} \Big/ \sqrt{\sigma_p^2 \sigma_\theta^2} = \sqrt{3} \big/ 2 \cong 0.87$.

4. Rotation About the Centre of Axle of the Robot

The robot wheels are controlled to travel in opposite directions at equal speed as measured by the odometry, with the aim of turning through an anti-clockwise angle α at one position with respect to the centre of the axle. The starting angle with respect to the X-axis is assumed to be *β*, as shown in figure 3. Errors in odometry mean that the point of turning moves from its starting position and the total angle turned is not exactly *α*.

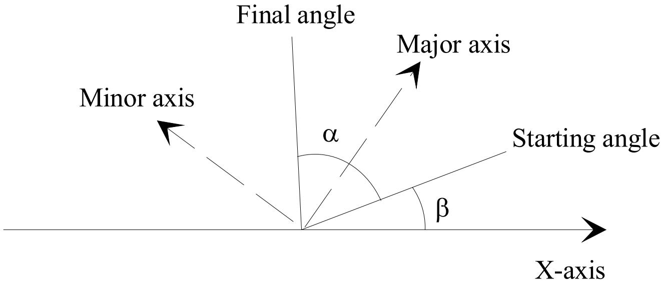

Figure 3 - Definition of Starting and Final Angles, and Major and Minor Axes.

Again small segments of motion are considered. Each segment now is a change of angle *Δθ=α/k*. The error in angle, e_α is examined first.

$$\begin{aligned} e_\alpha(k) &= e_\alpha(k-1) + \frac{e_R(k) - e_L(k)}{B} \\ &= e_\alpha(0) + \tfrac{1}{B} \sum_{i=1}^{k} \left(e_R(i) - e_L(i) \right) \end{aligned} \tag{16}$$

The variance of this error is given by

$$\begin{aligned} \sigma_\alpha^2(k) &= \sigma_\alpha^2(0) + \frac{k}{B^2}(k_R^2 + k_L^2)\Delta\theta \frac{B}{2} \\ &= \sigma_\alpha^2(0) + \frac{|\alpha|}{2B}(k_R^2 + k_L^2) \end{aligned} \tag{17}$$

To determine the covariance matrix of errors in position of the centre of axle, two orthogonal axes are first defined as shown in figure 3. The *major* axis is defined to be at an angle of $\beta+\alpha/2$ and the *minor* axis at the angle of $\beta+\alpha/2+\pi/2$. The errors of the axle centre are derived with respect to these axes, since these components will be shown to have zero covariance.

As the robot rotates, it is only the common mode or sum of wheel increments that contribute to the axle centre error. These errors are always in the direction or opposite direction to that at which the robot faces at any instant. As the angle θ moves from $-\alpha/2$ to $\alpha/2$ with respect to the major axis, the major axis error $e_M(k)$ is given by[1]

$$\begin{aligned} e_M(k) &= e_M(k-1) + \frac{e_L(k)+e_R(k)}{2}\cos\theta(k) \\ &= e_M(0) + \sum_{i=1}^{k} \frac{e_L(i)+e_R(i)}{2}\cos\theta(i) \end{aligned} \tag{18}$$

The variance of the major axis component is given by

$$\sigma_M^2(k) = \sigma_M^2(0) + \sum_{i=1}^{k} \frac{k_L^2+k_R^2}{4}\Delta\theta\frac{B}{2}\cos^2\theta(i) \tag{19}$$

In the limit as $\Delta\theta$ approaches zero, equation (19) becomes an integral as follows:

$$\begin{aligned} \sigma_M^2(k) &= \sigma_M^2(0) + \left| \int_{-\alpha/2}^{\alpha/2} \frac{B(k_L^2+k_R^2)}{8}\cos^2(\theta)d\theta \right| \\ &= \sigma_M^2(0) + \frac{B(k_L^2+k_R^2)}{8}\left| \left[\frac{\sin 2\theta}{4} + \frac{\theta}{2} \right]_{-\alpha/2}^{\alpha/2} \right| \\ &= \sigma_M^2(0) + \frac{B(k_L^2+k_R^2)\left|\sin\alpha+\alpha\right|}{16} \end{aligned} \tag{20}$$

The minor axis error, $e_m(k)$ is derived in a similar way:

$$\begin{aligned} e_m(k) &= e_m(k-1) + \frac{e_L(k)+e_R(k)}{2}\sin\theta(k) \\ &= e_m(0) + \sum_{i=1}^{k} \frac{e_L(i)+e_R(i)}{2}\sin\theta(i) \end{aligned} \tag{21}$$

Again taking limits introduces an integral:

[1] We have ignored the second order effect of angle error in *cos* $\theta(k)$.

$$
\begin{aligned}
\sigma_m^2(k) &= \sigma_m^2(0) + \left| \int_{-\alpha/2}^{\alpha/2} (k_L^2 + k_R^2)\frac{B}{2}\sin^2(\theta)d\theta \right| \\
&= \sigma_m^2(0) + (k_L^2 + k_R^2)\frac{B}{2}\left| \left[\frac{\theta}{2} - \frac{\sin 2\theta}{4} \right]_{-\alpha/2}^{\alpha/2} \right| \\
&= \sigma_m^2(0) + (k_L^2 + k_R^2)\frac{B}{2}\left| \left(\frac{\alpha - \sin\alpha}{2} \right) \right|
\end{aligned} \tag{22}
$$

To show that the major and minor components have covariance of 0, the expectation of $e_M e_m$ is examined. Note that

$$
\begin{aligned}
E[e_L(i)e_R(i)] &= 0 \\
E[e_L(i)e_L(j)] &= 0, i \neq j \\
E[e_R(i)e_R(j)] &= 0, i \neq j
\end{aligned} \tag{23}
$$

Taking expectations of equation (18) multiplied by equation (21), and taking the limit:

$$
\begin{aligned}
\sigma_{mM}(k) &= \sigma_{mM}(0) + \frac{B}{2}\int_{-\alpha/2}^{\alpha/2} \frac{k_L^2 + k_R^2}{4}\cos(\theta)\sin(\theta)d\theta \\
&= \sigma_{mM}(0) + \frac{B(k_L^2 + k_R^2)}{8}\left[\frac{\cos 2\theta}{4} \right]_{-\alpha/2}^{\alpha/2} \\
&= \sigma_{mM}(0) + 0
\end{aligned} \tag{24}
$$

That is, the covariance contribution between the major and minor axis components is zero. Symmetry arguments could be applied to achieve the same result, and in fact this was the method for choosing the axes in the first place. The covariance between the angle and the major and minor axes are now investigated:

$$
\begin{aligned}
\sigma_{\alpha M}(k) &= E\left[\left(e_\alpha(0) + \tfrac{1}{B}\sum_{i=1}^{k} \big(e_R(i) - e_L(i)\big) \right)\left(e_M(0) + \sum_{i=1}^{k} \frac{e_L(i) + e_R(i)}{2}\cos\theta(i) \right) \right] \\
&= \sigma_{\alpha M}(0) + E\left[\tfrac{1}{2B}\sum_{i=1}^{k} \big(e_R^2(i) - e_L^2(i)\big)\cos\theta(i) \right] \\
&= \sigma_{\alpha M}(0) + \tfrac{1}{2B}\sum_{i=1}^{k} \big(k_R^2 - k_L^2\big)\cos\theta(i)\Delta\theta B/2
\end{aligned} \tag{25}
$$

In the limit as $\Delta\theta$ approaches zero, the summation becomes an integral as follows:

$$\begin{aligned} \sigma_{\alpha M}(k) &= \sigma_{\alpha M}(0) + \frac{k_R^2 - k_L^2}{4} \int_{-\alpha/2}^{\alpha/2} \cos\theta d\theta \\ &= \sigma_{\alpha M}(0) + \frac{k_R^2 - k_L^2}{2} \sin(\alpha/2) \end{aligned} \tag{26}$$

Similarly, it follows that

$$\begin{aligned} \sigma_{\alpha m}(k) &= \sigma_{\alpha m}(0) + \frac{k_R^2 - k_L^2}{4} \int_{-\alpha/2}^{\alpha/2} \sin\theta d\theta \\ &= \sigma_{\alpha m}(0) + 0 \end{aligned} \tag{27}$$

That is, there is no correlation between the angle and minor axis errors.

5. Extension to Arbitrary Radius Arcs

In this section we consider the case of the wheeled vehicle traversing an arc of finite radius *R*. Figure 4 shows the propagation of an error in the direction of motion after traversing an arc of angle Δθ. The result is a perpendicular error of $-e_s \sin\Delta\theta$ and a straight ahead error of $e_s \cos\Delta\theta$.

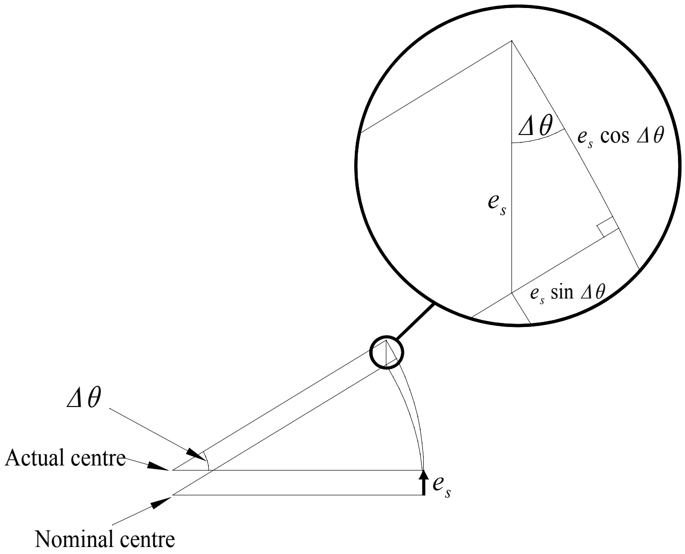

Figure 4 Propagation of straight error in arc move of angle Δθ.

The propagation of a perpendicular error e_p is shown in Figure 5. The result is a perpendicular error of $e_p \cos\Delta\theta$ and a straight ahead error of $e_p \sin\Delta\theta$. The propagation of straight and perpendicular errors can be seen to be a rotation through the angle $\Delta\theta$. The propagation of an angle error e_θ through the same arc is shown in Figure 6 and is much more complex in the geometry.

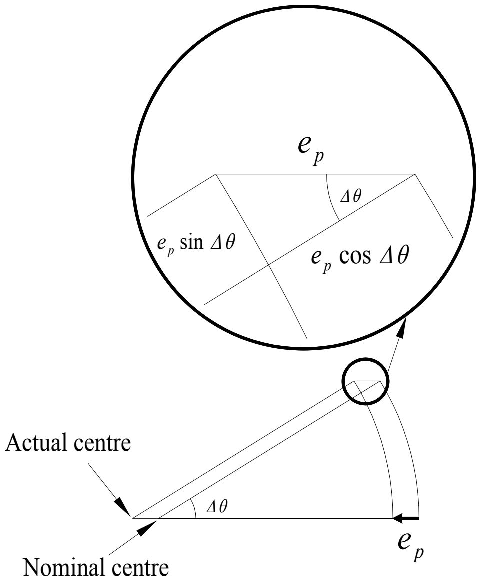

Figure 5 - Propagation of perpendicular error in arc move of angle Δθ.

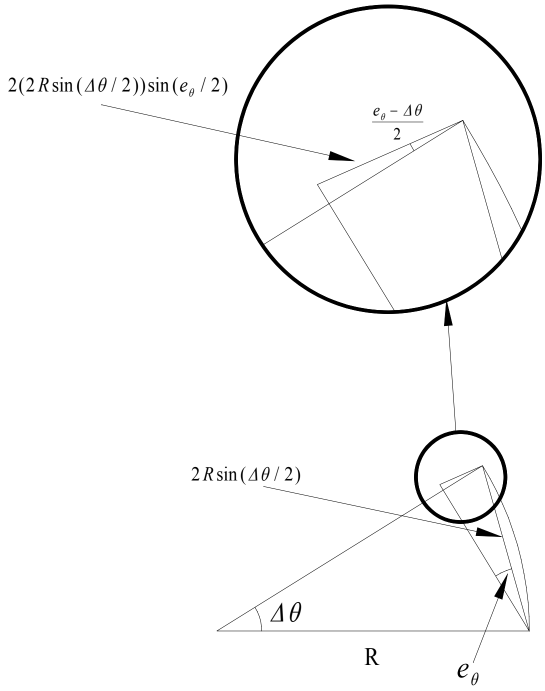
$2(2R\sin(\varDelta\theta/2))\sin(e_\theta/2)$
$\frac{e_\theta - \varDelta\theta}{2}$
$2R\sin(\varDelta\theta/2)$
$\varDelta\theta$
R
e_θ

Figure 6 - Propagation of angle error in arc move of angle Δθ and radius *R*.

On close consideration of the small triangle in the upper bubble, the error length is $4R\sin(\varDelta\theta/2)\sin(e_\theta/2)$ which is at an angle of $\frac{e_\theta - \varDelta\theta}{2}$ to the radial direction at the end point. For small angle errors and angular increments Δθ, the perpendicular error is approximated by

$$4R\sin(\Delta\theta/2)\sin(e_\theta/2)\cos((e_\theta - \Delta\theta)/2) \cong 2R\sin(\Delta\theta/2)e_\theta \tag{28}$$

and the tangential error is approximated by

$$4R\sin(\Delta\theta/2)\sin(e_\theta/2)\sin((e_\theta-\Delta\theta)/2)\cong 0$$

This gives the following iterative relationships on errors:

$$\begin{array}{lcl} e_p(k) & \cong & e_p(k-1)\cos\Delta\theta - e_s(k-1)\sin\Delta\theta + 2Re_\theta(k-1)\sin(\Delta\theta/2) \\ e_s(k) & \cong & e_p(k-1)\sin\Delta\theta + e_s(k-1)\cos\Delta\theta + \dfrac{e_L(k-1)+e_R(k-1)}{2} \\ e_\theta(k) & = & e_\theta(k-1)+e_{\Delta\theta}(k-1) = e_\theta(0)+\displaystyle\sum_{i=0}^{k-1} e_{\Delta\theta}(i) \end{array} \tag{29}$$

We define the complex error *E(k)* as follows:

$$E(k)\equiv e_p(k)+\mathbf{j}e_s(k) \tag{30}$$

This allows rewriting of the first two iterative expressions of (29) as:

$$E(k)=e^{\mathbf{j}\Delta\theta}E(k-1)+Re_\theta(k-1)2\sin(\Delta\theta/2)+\mathbf{j}e_d(k-1) \tag{31}$$

where

$$e_d(k-1)\equiv\frac{e_L(k-1)+e_R(k-1)}{2} \tag{32}$$

After iterating on (31) we have:

$$\begin{array}{lcl} E(k) & = & e^{\mathbf{j}k\Delta\theta}E(0)+\displaystyle\sum_{i=0}^{k-1}e^{\mathbf{j}i\Delta\theta}\big(Re_\theta(k-1-i)2\sin(\Delta\theta/2)+\mathbf{j}e_d(k-1-i)\big) \\ & = & e^{\mathbf{j}\alpha}E(0)+2R\sin(\Delta\theta/2)\displaystyle\sum_{i=0}^{k-1}e^{\mathbf{j}i\Delta\theta}\left(e_\theta(0)+\sum_{g=0}^{k-i-2}e_{\Delta\theta}(g)\right)+\mathbf{j}\sum_{i=0}^{k-1}e^{\mathbf{j}i\Delta\theta}e_d(k-1-i) \\ & = & e^{\mathbf{j}\alpha}E(0)+\mathrm{Re}_\theta(0)\dfrac{e^{\mathbf{j}\alpha}-1}{\mathbf{j}}+2R\sin(\Delta\theta/2)\displaystyle\sum_{g=0}^{k-2}e_{\Delta\theta}(g)\sum_{i=0}^{k-1-g}e^{\mathbf{j}i\Delta\theta}+\mathbf{j}\sum_{i=0}^{k-1}e^{\mathbf{j}i\Delta\theta}e_d(k-1-i) \end{array} \tag{33}$$

where

$$\begin{aligned}&\alpha \equiv k\Delta\theta \\ &2\sin(\Delta\theta/2)\sum_{i=0}^{k-1} e^{\mathbf{j}i\Delta\theta} = 2\sin(\Delta\theta/2)\frac{e^{\mathbf{j}\alpha}-1}{e^{\mathbf{j}\Delta\theta}-1} \\ &\to \frac{e^{\mathbf{j}\alpha}-1}{\mathbf{j}} \text{as} \Delta\theta \to 0 \text{and} \alpha = k\Delta\theta \text{fixed.}\end{aligned} \tag{34}$$

In (34) the limit as $\Delta\theta \to 0$ is evaluated using L'Hopital's Rule[2]. Rewriting equation (33) and simplifying with the real and imaginary versions of the limits in (34) gives:

$$\begin{aligned} e_s(k) &\cong e_s(0)\cos\alpha + e_p(0)\sin\alpha + Re_\theta(0)(1-\cos\alpha) + R\sum_{g=0}^{k-2} e_{\Delta\theta}(g)\big(1-\cos((k-1-g)\Delta\theta)\big) + \sum_{i=0}^{k-1} e_d(k-i-1)\cos(i\Delta\theta) \\ e_p(k) &\cong -e_s(0)\sin\alpha + e_p(0)\cos\alpha + Re_\theta(0)\sin\alpha + R\sum_{g=0}^{k-2} e_{\Delta\theta}(g)\sin((k-1-g)\Delta\theta) - \sum_{i=0}^{k-1} e_d(k-i-1)\sin(i\Delta\theta) \\ e_\theta(k) &= e_\theta(0) + \sum_{i=0}^{k-1} e_{\Delta\theta}(i) \end{aligned} \quad . \tag{35}$$

or in matrix form:

$$\begin{aligned}&\mathbf{e(k)} = \Phi\mathbf{e}(0) + \sum_{i=1}^{k-1} \Psi(i\Delta\theta)\mathbf{en}(k-1-i) \\ &\text{where}\Phi = \begin{bmatrix} \cos\alpha & \sin\alpha & R(1-\cos\alpha) \\ -\sin\alpha & \cos\alpha & R\sin\alpha \\ 0 & 0 & 1 \end{bmatrix}, \Psi(i\Delta\theta) = \begin{bmatrix} R\big(1-\cos(i\Delta\theta)\big) & \cos(i\Delta\theta) \\ R\sin(i\Delta\theta) & \sin(i\Delta\theta) \\ 1 & 0 \end{bmatrix}, \mathbf{en}(i) = \begin{bmatrix} e_{\Delta\theta}(i) \\ e_d(i) \end{bmatrix}\end{aligned} \tag{36}$$

The error covariance matrix is then

$$\begin{aligned}\mathbf{P}(k) &= Cov(\mathbf{e}(k)) = E[\mathbf{e}(k)\mathbf{e}(k)^T] \\ &= \Phi\mathbf{(0)}\Phi^T + \sum_{i=1}^{k-1} \Psi(i\Delta\theta)\left|R\Delta\theta\right|\mathbf{Q}(i)\Psi(i\Delta\theta)^{\mathrm{T}}\end{aligned} \tag{37}$$

where

$$\left|R\Delta\theta\right|\mathbf{Q}(i) = \begin{bmatrix} E[{e_{\Delta\theta}}^2(i)] & E[e_{\Delta\theta}(i)e_d(i)] \\ E[e_{\Delta\theta}(i)e_d(i)] & E[{e_d}^2(i)] \end{bmatrix} = \left|R\Delta\theta\right| \begin{bmatrix} \dfrac{(s_R k_R^2 + s_L k_L^2)}{B^2} & \dfrac{(s_R k_R^2 - s_L k_L^2)}{2B} \\ \dfrac{(s_R k_R^2 - s_L k_L^2)}{2B} & \dfrac{(s_R k_R^2 + s_L k_L^2)}{4} \end{bmatrix}$$

(38)

and $s_L\left|R\Delta\theta\right|$ is defined to be the distance travelled by the left wheel with an angle change of $\Delta\theta$. Therefore the angle subtended by the left wheel is given by

[2] When the numerator and demominator of a limit approach zero they can be replaced by their derivatives with respect to the limiting variable.

$$\begin{aligned} &\Delta\theta = \frac{s_L R \Delta\theta}{(R - B/2)\Delta\theta} \\ &\Rightarrow s_L = \left|1 - \frac{B}{2R}\right|, similarly s_R = \left|1 + \frac{B}{2R}\right| \end{aligned} \tag{39}$$

The radius of curvature is assumed to be negative for arcs with a centre to the right of the heading direction.

In the limit as *Δθ* approaches zero and *k* approaches infinity, we can evaluate (37) as the integral:

$$\mathbf{P(k)} = \Phi\mathbf{P}(0)\Phi^T + |R| sign(\alpha)\int_0^\alpha \Psi(\theta)\mathbf{Q}\Psi(\theta)^{\mathrm{T}} d\theta \tag{40}$$

where *sign(x)* is *+1* for *x>0* and *-1* for *x<0*. The integral evaluates to the following:

$$\begin{aligned} &\int_0^\alpha \Psi(\theta)\mathbf{Q}\Psi(\theta)^T d\theta = \begin{bmatrix} \rho_s^2 & \rho_{sp} & \rho_{s\theta} \\ \rho_{sp} & \rho_p^2 & \rho_{p\theta} \\ \rho_{s\theta} & \rho_{p\theta} & \rho_\theta^2 \end{bmatrix} \\ &\rho_s^2 = \frac{s_R k_R^2 + s_L k_L^2}{4}\left(\frac{R^2}{B^2}(6\alpha - 8\sin\alpha + \sin 2\alpha) + \left(\frac{2\alpha + \sin 2\alpha}{4}\right)\right) + \left(s_R k_R^2 - s_L k_L^2\right)\frac{R}{4B}(4\sin\alpha - 2\alpha - \sin 2\alpha) \\ &\rho_p^2 = \left(\frac{2\alpha - \sin 2\alpha}{4}\right)\left\{\left(s_R k_R^2 + s_L k_L^2\right)\left(\frac{R^2}{B^2} + \frac{1}{4}\right) - \left(s_R k_R^2 - s_L k_L^2\right)\frac{R}{B}\right\} \\ &\rho_\theta^2 = \frac{\alpha\left(s_R k_R^2 + s_L k_L^2\right)}{B^2} \\ &\rho_{sp} = \left(s_R k_R^2 + s_L k_L^2\right)\left\{\frac{R^2}{B^2}\left(\frac{3 - 4\cos\alpha + \cos 2\alpha}{4}\right) + \left(\frac{\cos 2\alpha - 1}{16}\right)\right\} \\ &\rho_{s\theta} = \left(s_R k_R^2 + s_L k_L^2\right)\frac{R}{B^2}(\alpha - \sin\alpha) + \left(s_R k_R^2 - s_L k_L^2\right)\frac{\sin\alpha}{2B} \\ &\rho_{p\theta} = (1 - \cos\alpha)\left[\left(s_R k_R^2 + s_L k_L^2\right)\frac{R}{B^2} - \left(s_R k_R^2 - s_L k_L^2\right)\frac{1}{2B}\right] \\ &. \end{aligned} \tag{41}$$

It has been shown by the author that in the limit as the arc approaches a straight line (ie *R* approaches infinity, *α* approaches zero with *Rα=d* constant) the results of equation (41) approach the straight line results presented in Section 3.

6. Example Calculation of the Odometry Covariance Matrix

In this section an example robot path is used to illustrate the results of the previous sections. The path shown in Figure 7 consists of a straight line, turn on the spot, an arc of radius twice the wheel base *2B* and an arc of radius *B/4*, where *B* is 0.5 m. The covariance matrix is calculated after each move. To facilitate the calculations, we require a transformation matrix ***T(θ)*** which represents a rotation of the X-Y coordinate system through an angle *θ*. Suppose we represent our robot position and orientation with **x**=*[x y θ]*^{T}, then **T***(θ)* is given by:

$$\mathbf{T}(\theta)=\begin{bmatrix}\cos\theta & -\sin\theta & 0\\ \sin\theta & \cos\theta & 0\\ 0 & 0 & 1\end{bmatrix} \tag{42}$$

Note that the inverse and transpose of **T** are the equal.

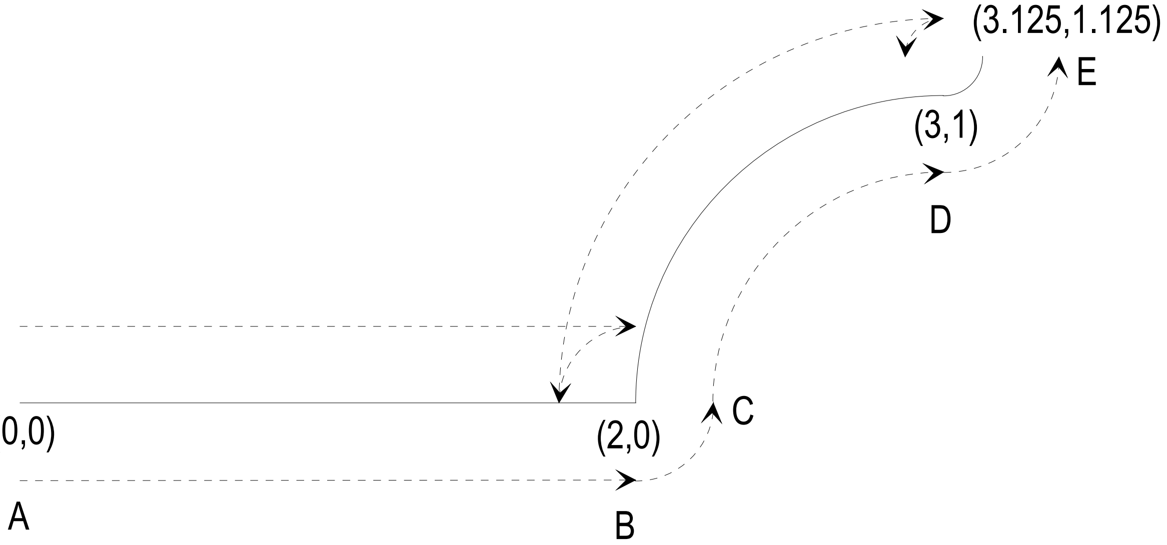

Figure 7 - Example robot path with (x,y) coordinates marked (in metres). The left and right wheel paths are shown dotted.

The covariance matrix results presented in Sections 3 and 5 are with respect to tangential coordinates for lines and arcs. Therefore coordinates need to be rotated by $-\theta$ to convert to X-Y cartesian coordinates, where θ is the heading angle of the robot after the path segment. Thus if the covariance matrix $\mathbf{P_{sp}}$ is with respect to tangential coordinates, then the cartesian covariance matrix, $\mathbf{P_{xy}}$ is given by

$$\mathbf{P_{xy}}=Cov[\mathbf{T}(-\theta)\mathbf{e_{sp}}(\mathbf{T}(-\theta)\mathbf{e_{sp}})^T]=Cov[\mathbf{T}(-\theta)\mathbf{e_{sp}}\mathbf{e_{sp}}^T\mathbf{T}(-\theta)^T]=\mathbf{T}(-\theta)\mathbf{P_{sp}}\mathbf{T}(-\theta)^T \tag{43}$$

The results for straight line paths of Section 3 are summarised here in matrix form. The distance *d* is positive when moving in the direction of the robot and negative when the robot moves backwards. The covariance after the straight line move is the addition of the propagation of the initial covariance $\mathbf{P_{sp}}(0)$ and the contribution of noise along the path itself, $\mathbf{Q_{line}}(d)$:

$$
\begin{aligned}
&\mathbf{P_{sp}}(k) = \mathbf{\Phi_{line}}(d)\mathbf{P_{sp}}(0)\mathbf{\Phi_{line}}(d)^T + \mathbf{Q_{line}}(d) \\
&\text{where } \mathbf{\Phi_{line}}(d) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & d \\ 0 & 0 & 1 \end{bmatrix} \\
&\text{and } \mathbf{Q_{line}}(d) = |d| \begin{bmatrix} \frac{k_R^2 + k_L^2}{4} & \frac{d(k_R^2 - k_L^2)}{4B} & \frac{(k_R^2 - k_L^2)}{2B} \\ \frac{d(k_R^2 - k_L^2)}{4B} & \frac{d^2(k_R^2 + k_L^2)}{3B^2} & \frac{d(k_R^2 + k_L^2)}{2B^2} \\ \frac{(k_R^2 - k_L^2)}{2B} & \frac{d(k_R^2 + k_L^2)}{2B^2} & \frac{k_R^2 + k_L^2}{B^2} \end{bmatrix}
\end{aligned}
\qquad (44)
$$

The results are generated from MATLAB scripts which are included at the end of this section. Error ellipses are generated for X-Y errors at the end of each segment and due to the noise introduced along each segment and are plotted in Figures 8 and 9. Covariance matrices are symmetric and therefore, from linear algebra, can be diagonalised using an orthonormal basis (ie by changing coordinate system to these unity orthogonal vectors we achieve decoupling of the error contributions since their covariance matrix is diagonal). The basis vectors are the eigenvectors and these form the axes of the error ellipses. The lengths of the axes are the square root of the eigenvalues and correspond to standard deviations of the independent noise contribution in the direction of the eigenvector. For the purposes of displaying the results,only X-Y errors are shown.

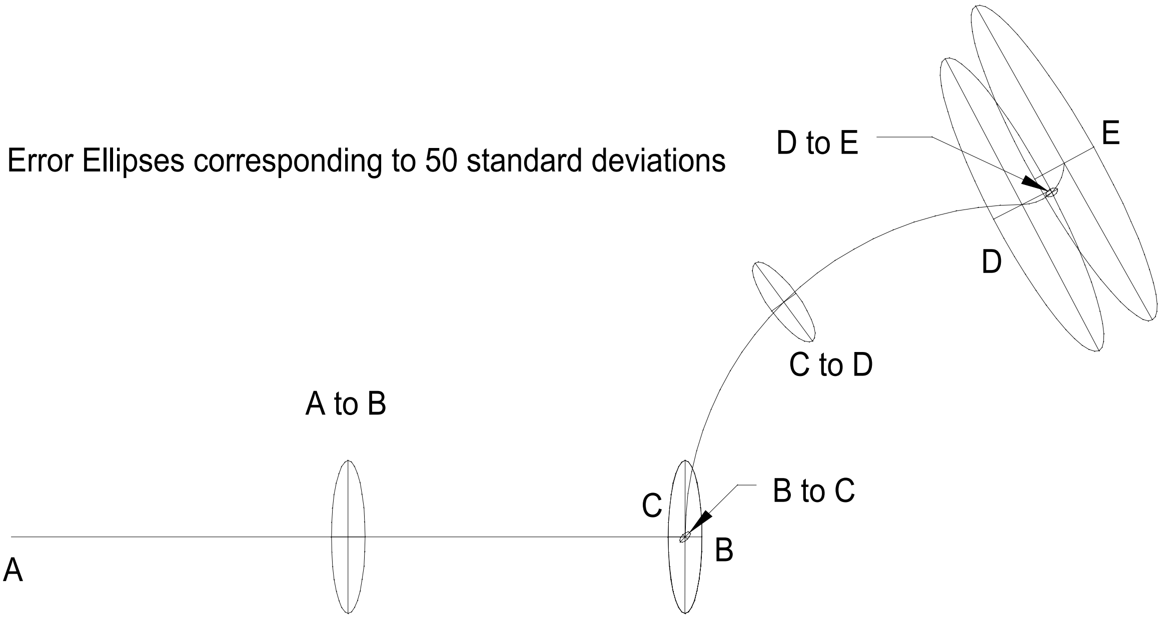

Figure 8 - Error Ellipses for each path segment and at the end of each segment.
$k_L = k_R = 10^{-3} m^{\frac{1}{2}}$, *B=0.5 m.*

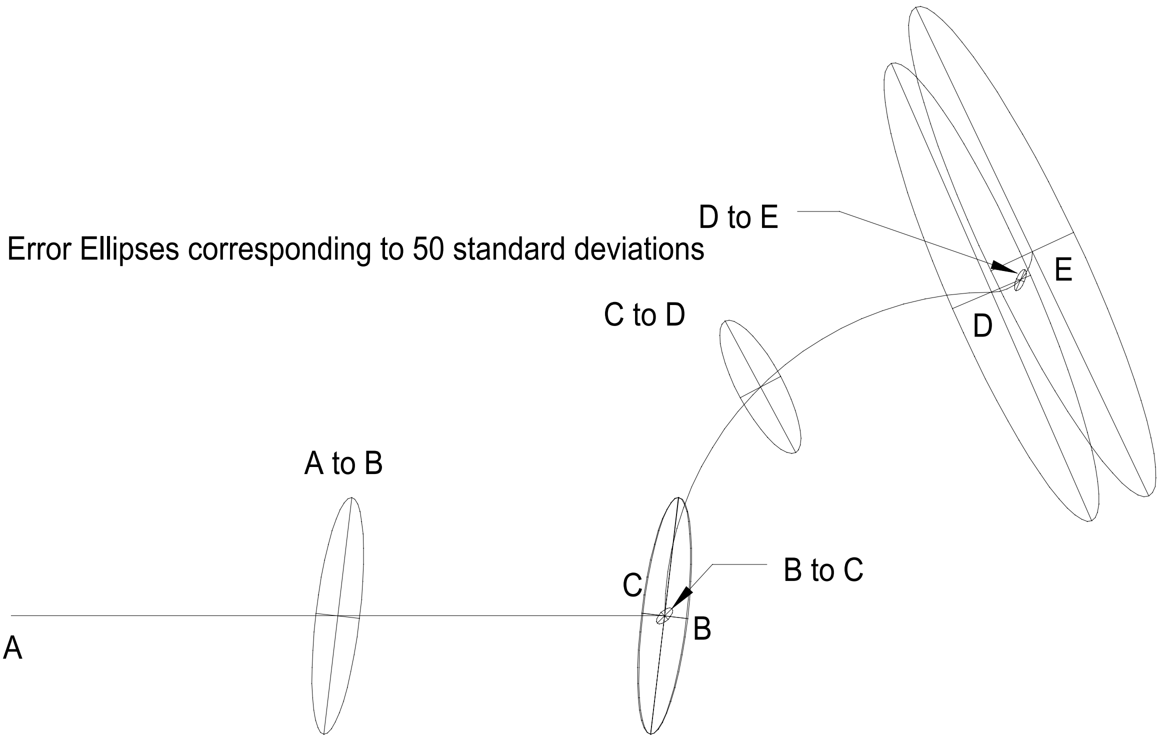

Figure 9 - Error Ellipses for each path segment and at the end of each segment. $k_L = 10^{-3} m^{1/2} \quad k_R = 2 \times 10^{-3} m^{1/2}$ *B=0.5 m.*

Tables I and II list the covariance matrices corresponding to Figures 8 and 9.

TABLE I $k_L = k_R = 10^{-3} m^{1/2}$, *B=0.5 m*

Location	$\mathbf{P}_{\mathbf{xy}}$
A	$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$
A to B	$10^{-4} \times \begin{bmatrix} 0.01 & 0 & 0 \\ 0 & .2133 & .1600 \\ 0 & .1600 & .1600 \end{bmatrix}$
B	$10^{-4} \times \begin{bmatrix} 0.01 & 0 & 0 \\ 0 & .2133 & .1600 \\ 0 & .1600 & .1600 \end{bmatrix}$
B to C	$10^{-5} \times \begin{bmatrix} 0.0098 & 0.0062 & 0 \\ 0.0062 & 0.0098 & 0 \\ 0 & 0 & .3142 \end{bmatrix}$
C	$10^{-4} \times \begin{bmatrix} 0.0110 & 0.0006 & 0 \\ 0.0006 & 0.2143 & -0.1600 \\ 0 & -0.1600 & .1914 \end{bmatrix}$

C to D	$10^{-4}\times\begin{bmatrix}0.0346 & -.0375 & -0.0507\\ 0.0006 & 0.0589 & 0.0750\\ 0 & 0.0750 & 0.1257\end{bmatrix}$
D	$10^{-4}\times\begin{bmatrix}0.2370 & -0.3883 & -0.2421\\ -0.3883 & 0.7846 & 0.4264\\ -0.2421 & 0.4264 & 0.3171\end{bmatrix}$
D to E	$10^{-5}\times\begin{bmatrix}0.0074 & 0.0047 & 0\\ 0.0047 & 0.0123 & -0.0393\\ 0 & -0.0393 & 0.3142\end{bmatrix}$
E	$10^{-4}\times\begin{bmatrix}0.3032 & -0.4763 & 0.2817\\ -0.4763 & 0.8974 & -0.4700\\ 0.2817 & -0.4700 & 0.3485\end{bmatrix}$

TABLE II $k_L = 10^{-3} m^{\frac{1}{2}}$ $k_R = 2\times 10^{-3} m^{\frac{1}{2}}$**,** *B=0.5 m*

Location	$\mathbf{P}_{\mathbf{xy}}$
A	$\begin{bmatrix}0 & 0 & 0\\ 0 & 0 & 0\\ 0 & 0 & 0\end{bmatrix}$
A to B	$10^{-3}\times\begin{bmatrix}0.0250 & 0.0600 & 0.0600\\ 0.0600 & 0.5333 & 0.4000\\ 0.0600 & 0.4000 & 0.4000\end{bmatrix}$
B	$10^{-4}\times\begin{bmatrix}0.01 & 0 & 0\\ 0 & .2133 & .1600\\ 0 & .1600 & .1600\end{bmatrix}$
B to C	$10^{-5}\times\begin{bmatrix}0.0245 & 0.0156 & -0.0750\\ 0.0156 & 0.0245 & -0.0750\\ -0.0750 & -0.0750 & 0.7854\end{bmatrix}$
C	$10^{-4}\times\begin{bmatrix}0.0275 & 0.0616 & -0.0675\\ 0.0616 & 0.5358 & -0.4075\\ -0.0675 & -0.4075 & 0.4785\end{bmatrix}$
C to D	$10^{-4}\times\begin{bmatrix}0.0614 & -0.0797 & -0.0795\\ -0.0797 & 0.1694 & 0.1875\\ -0.0795 & 0.1875 & 0.2670\end{bmatrix}$
D	$10^{-3}\times\begin{bmatrix}0.0432 & -0.0837 & -0.0002\\ -0.0837 & 0.1999 & 0.1074\\ -0.0491 & 0.1074 & 0.0746\end{bmatrix}$

D to E	$10^{-4}\times\begin{bmatrix}0.0013 & 0.0015 & -0.0056\\ 0.0015 & 0.0043 & -0.0184\\ -0.0056 & -0.0184 & 0.1021\end{bmatrix}$
E	$10^{-3}\times\begin{bmatrix}0.0568 & -0.1042 & 0.0578\\ -0.1042 & 0.2283 & -0.1185\\ 0.0578 & -0.1185 & 0.0848\end{bmatrix}$

Shown below is a MATLAB script RUN.M and functions to evaluate the covariance matrices. Output from RUN.M is also shown for different sets of *kl* and *kr*.

```
;********* start of RUN.M ******
kl = 1e-3
kr = 2e-3
B = 0.5
d = 2
angle = 0
P=zeros(3,3);
P = line_cov(P,kl, kr, B, d);
%[V,D] = eig(P)
Pxy = P
[V,D] = eig(Pxy(1:2, 1:2))
sqrtD=sqrt(D)

a = pi/2;
dl= ra_to_dl(1e-6, pi/2, B);
dr =ra_to_dr(1e-6, pi/2, B);
P = arc_cov(P, dl, dr, kl, kr, B);
angle = angle + a;
Pxy = T(-angle)*P*T(-angle)'
[V,D] = eig(Pxy(1:2, 1:2))
sqrtD=sqrt(D)
Pdiff = arc_cov(zeros(3,3), dl, dr, kl, kr, B);
Pdiffxy = T(-angle)*Pdiff*T(-angle)'
[V,D] = eig(Pdiffxy(1:2, 1:2))
sqrtD=sqrt(D)

a=-pi/2;
dl= ra_to_dl(-1, a, B);
dr =ra_to_dr(-1, a, B);
P = arc_cov(P, dl, dr, kl, kr, B)
angle = angle + a;
Pxy = T(-angle)*P*T(-angle)'
[V,D] = eig(Pxy(1:2, 1:2))
sqrtD=sqrt(D)
Pdiff = arc_cov(zeros(3,3), dl, dr, kl, kr, B);
Pdiffxy = T(-angle)*Pdiff*T(-angle)'
[V,D] = eig(Pdiffxy(1:2, 1:2))
sqrtD=sqrt(D)


a = pi/2;
dl= ra_to_dl(0.125, a, B);
dr =ra_to_dr(0.125, a, B);
P = arc_cov(P, dl, dr, kl, kr, B);
angle = angle + a;
Pxy = T(-angle)*P*T(-angle)'
[V,D] = eig(Pxy(1:2, 1:2))
sqrtD=sqrt(D)
Pdiff = arc_cov(zeros(3,3), dl, dr, kl, kr, B);
Pdiffxy = T(-angle)*Pdiff*T(-angle)'
[V,D] = eig(Pdiffxy(1:2, 1:2))
sqrtD=sqrt(D)

;************ end of RUN.M *************
function Transf=t(theta);
```

```
Transf = eye(3);
Transf(1,1) = cos(theta);
Transf(2,1) = sin(theta);
Transf(1,2) = -Transf(2,1);
Transf(2,2) = Transf(1,1);
;***********************************
function P=line_cov(P, kl, kr, B, d);
%function P=line_cov(P, kl, kr, B, d);
C=eye(3);
kl2 = kl*kl;
kr2 = kr*kr;
C(1,1) = d*(kl2+kr2)/4;
C(1,2) = d^2*(kr2-kl2)/4/B;
C(1,3) = d*(kr2-kl2)/2/B;
C(2,1) = C(1,2);
C(2,2) = d^3*(kl2+kr2)/3/B/B;
C(2,3) = d^2*(kr2+kl2)/2/B/B;
C(3,1) = C(1,3);
C(3,2) = C(2,3);
C(3,3) = d*(kl2+kr2)/B/B;

phi=eye(3);
phi(2,3) = d;
P = phi*P*phi'+C;


function P=arc_cov(P, dl, dr, kl, kr, B);
%function P=arc_cov(P, dl, dr, kl, kr, B);
% update covariance matrix P for arc of left distance dl, and right dr
% with robot parameters kl, kr, B wheel base
% Assumes that P is in tangential coordinates and is left in tangential
% coordinates on return

C=eye(3);
R = B*(dr+dl)/2/(dr-dl);
a = (dr-dl)/B;
kRsr = abs(B*dr/(dr-dl)*kr*kr);
kRsl = abs(B*dl/(dr-dl)*kl*kl);
ksum = kRsr + kRsl;
kdiff = kRsr - kRsl;

    C(1,1) = ksum*R*R/B/B*(3*a/2-2*sin(a)+sin(2*a)/4)+ksum/4*(a/2+sin(2*a)/4) + kdiff*R/B*(sin(a)-a/2-sin(2*a)/4);
    C(1,2) = ksum*(R*R/B/B*(3-4*cos(a)+cos(2*a))/4+(cos(2*a)-1)/16)+kdiff*R/4/B*(2*cos(a)-cos(2*a)-1);
      C(1,3) = ksum*R/B/B*(a-sin(a))+kdiff*sin(a)/2/B;
      C(2,1) = C(1,2);
      C(2,2) = (a/2-sin(2*a)/4)*(ksum*(R*R/B/B+0.25)-kdiff*R/B);
      C(2,3) = (ksum*R/B/B-kdiff/2/B)*(1-cos(a));
      C(3,1) = C(1,3);
      C(3,2) = C(2,3);
      C(3,3) = ksum/B/B*a;

      if (a < 0) C = -1 * C;
    end

% propagate P
    Phi=eye(3);
    Phi(2,2)=cos(a);
    Phi(1,1)=Phi(2,2);
    Phi(1,2) = sin(a);
      Phi(1,3) = R*(1-cos(a));
      Phi(2,1) = -sin(a);
      Phi(2,3) = R*sin(a);
      P = Phi*P*Phi'+C;

function dl = ra_to_dl(R, a, B);
%function dl = ra_to_dl(R, a, B);
%converts radius and angle to left distance dl
```

```
dl = a*(2*R-B)/2;

function dr = ra_to_dr(R, a, B);
%function dr = ra_to_dr(R, a);
%converts radius and angle to left distance dr
dr = a*(2*R+B)/2;
```

;********* Results from running RUN.M**

```
run
kl =  1.0000e-003
kr =  1.0000e-003
B =  0.5000
d =     2
angle =     0
Pxy =  1.0e-004 *
    0.0100         0         0
         0    0.2133    0.1600
         0    0.1600    0.1600
V =
     1     0
     0     1
D =  1.0e-004 *
    0.0100         0
         0    0.2133
sqrtD =
    0.0010         0
         0    0.0046
Pxy =  1.0e-004 *
    0.0110    0.0006    0.0000
    0.0006    0.2143   -0.1600
    0.0000   -0.1600    0.1914
V =
    1.0000    0.0031
   -0.0031    1.0000
D =  1.0e-004 *
    0.0110         0
         0    0.2143
sqrtD =
    0.0010         0
         0    0.0046
Pdiffxy =  1.0e-005 *
    0.0098    0.0062    0.0000
    0.0062    0.0098   -0.0000
    0.0000   -0.0000    0.3142
V =
    0.7071    0.7071
   -0.7071    0.7071

D =  1.0e-006 *
    0.0357         0
         0    0.1607
sqrtD =  1.0e-003 *
    0.1889         0
         0    0.4008
P =  1.0e-004 *
    0.2370   -0.3883   -0.2421
   -0.3883    0.7846    0.4264
   -0.2421    0.4264    0.3171
Pxy =  1.0e-004 *
    0.2370   -0.3883   -0.2421
   -0.3883    0.7846    0.4264
   -0.2421    0.4264    0.3171
V =
    0.8878   -0.4603
    0.4603    0.8878
D =  1.0e-004 *
    0.0357         0
         0    0.9859
sqrtD =
    0.0019         0
         0    0.0099
Pdiffxy =  1.0e-004 *
    0.0346   -0.0375   -0.0507
   -0.0375    0.0589    0.0750
   -0.0507    0.0750    0.1257
V =
    0.8089   -0.5879
    0.5879    0.8089
D =  1.0e-005 *
    0.0731         0
         0    0.8616
sqrtD =
    0.0009         0
         0    0.0029
Pxy =  1.0e-004 *
    0.3032   -0.4763    0.2817
   -0.4763    0.8974   -0.4700
    0.2817   -0.4700    0.3485
V =
    0.8744   -0.4852
    0.4852    0.8744
D =  1.0e-003 *
    0.0039         0
         0    0.1162
sqrtD =
    0.0020         0
         0    0.0108
Pdiffxy =  1.0e-005 *
    0.0074    0.0047   -0.0000
    0.0047    0.0123   -0.0393
   -0.0000   -0.0393    0.3142
V =
    0.8555    0.5178
   -0.5178    0.8555
D =  1.0e-006 *
    0.0453         0
         0    0.1511
sqrtD =  1.0e-003 *
    0.2128         0
         0    0.3887
¯ diary
¯ run
kl =  1.0000e-003
kr =    0.0050
```

```
B =    0.5000
d =     2
angle =     0
Pxy =  1.0e-003 *
    0.0130    0.0480    0.0480
    0.0480    0.2773    0.2080
    0.0480    0.2080    0.2080
V =
    0.9849    0.1733
   -0.1733    0.9849
D =  1.0e-003 *
    0.0046         0
         0    0.2858
sqrtD =
    0.0021         0
         0    0.0169
Pxy =  1.0e-003 *
    0.0143    0.0488   -0.0540
    0.0488    0.2786   -0.2140
   -0.0540   -0.2140    0.2488
V =
   -0.9844   -0.1760
    0.1760   -0.9844
D =  1.0e-003 *
    0.0056         0
         0    0.2873
sqrtD =
    0.0024         0
         0    0.0170
Pdiffxy =  1.0e-004 *
    0.0128    0.0081   -0.0600
    0.0081    0.0128   -0.0600
   -0.0600   -0.0600    0.4084
V =
    0.7071    0.7071
   -0.7071    0.7071
D =  1.0e-005 *
    0.0464         0
```

```
         0    0.2089

sqrtD =

    0.0007         0

         0    0.0014

P =

    0.0002   -0.0004   -0.0002

   -0.0004    0.0010    0.0006

   -0.0002    0.0006    0.0004

Pxy =

    0.0002   -0.0004   -0.0002

   -0.0004    0.0010    0.0006

   -0.0002    0.0006    0.0004

V =

    0.9322   -0.3619

    0.3619    0.9322

D =

    0.0000         0

         0    0.0012

sqrtD =

    0.0051         0

         0    0.0347

Pdiffxy =

  1.0e-003 *

    0.0249   -0.0375   -0.0282

   -0.0375    0.0942    0.0975

   -0.0282    0.0975    0.1257

V =

    0.9162   -0.4007

    0.4007    0.9162

D =  1.0e-003 *

    0.0085         0

         0    0.1106

sqrtD =

    0.0029         0

         0    0.0105

Pxy =

    0.0002   -0.0005    0.0003

   -0.0005    0.0012   -0.0006
```

```
    0.0003   -0.0006    0.0004

V =

   -0.9194    0.3932

   -0.3932   -0.9194

D =

    0.0000         0

         0    0.0014

sqrtD =

    0.0053         0

         0    0.0376

Pdiffxy =  1.0e-004 *

    0.0052    0.0089   -0.0450

    0.0089    0.0257   -0.1196

   -0.0450   -0.1196    0.5969

V =

    0.9370    0.3492

   -0.3492    0.9370

D =  1.0e-005 *

    0.0183         0

         0    0.2905

sqrtD =

    0.0004         0

         0    0.0017

¯ quit

 5091 flops.

¯ run

kl =  1.0000e-003

kr =    0.0020

B =    0.5000

d =     2

angle =     0

Pxy =  1.0e-004 *

    0.0250    0.0600    0.0600

    0.0600    0.5333    0.4000

    0.0600    0.4000    0.4000

V =

    0.9933    0.1157

   -0.1157    0.9933
```

```
D =  1.0e-004 *

    0.0180         0

         0    0.5403

sqrtD =

    0.0013         0

         0    0.0074

Pxy =  1.0e-004 *

    0.0275    0.0616   -0.0675

    0.0616    0.5358   -0.4075

   -0.0675   -0.4075    0.4785

V =

    0.9929    0.1185

   -0.1185    0.9929

D =  1.0e-004 *

    0.0201         0

         0    0.5431

sqrtD =

    0.0014         0

         0    0.0074

Pdiffxy =

  1.0e-005 *

    0.0245    0.0156   -0.0750

    0.0156    0.0245   -0.0750

   -0.0750   -0.0750    0.7854

V =

    0.7071    0.7071

   -0.7071    0.7071

D =  1.0e-006 *

    0.0892         0

         0    0.4017

sqrtD =  1.0e-003 *

    0.2986         0

         0    0.6338

P =  1.0e-003 *

    0.0432   -0.0837   -0.0491

   -0.0837    0.1999    0.1074

   -0.0491    0.1074    0.0746

Pxy =  1.0e-003 *
```

```
    0.0432   -0.0837   -0.0491

   -0.0837    0.1999    0.1074

   -0.0491    0.1074    0.0746

V =

   -0.9174    0.3979

   -0.3979   -0.9174

D =  1.0e-003 *

    0.0070         0

         0    0.2362

sqrtD =

    0.0026         0

         0    0.0154

Pdiffxy =  1.0e-004 *

    0.0614   -0.0797   -0.0795

   -0.0797    0.1694    0.1875

   -0.0795    0.1875    0.2670

V =

    0.8834   -0.4686

    0.4686    0.8834

D =  1.0e-004 *

    0.0191         0

         0    0.2116

sqrtD =

    0.0014         0

         0    0.0046

Pxy =  1.0e-003 *

    0.0568   -0.1042    0.0578

   -0.1042    0.2283   -0.1185

    0.0578   -0.1185    0.0848

V =

   -0.9042    0.4270

   -0.4270   -0.9042

D =  1.0e-003 *

    0.0076         0

         0    0.2775

sqrtD =

    0.0028         0

         0    0.0167
```

```
Pdiffxy =  1.0e-004 *

    0.0013    0.0015   -0.0056

    0.0015    0.0043   -0.0184

   -0.0056   -0.0184    0.1021

V =

    0.9224    0.3861

   -0.3861    0.9224
```

```
D =  1.0e-006 *

    0.0651         0

         0    0.4928

sqrtD =  1.0e-003 *

    0.2551         0

         0    0.7020

¯ quit

 2622 flops.
```

7. Simulation Results

Monte-Carlo simulations studies have been performed on the odometry error covariance estimation results for two main reasons. Firstly, there are approximations involved in the derivation of the results and the simulations will be able to verify that the approximations are reasonable. Secondly, the errors need to be eliminated from the theoretical derivations and the simulation can verify the correctness of the derivations.

The simulation studies read a configuration file which specifies the left and right wheel movements defining the path, the wheel base *B*, the wheel noise constants k_L and k_R and the sample frequency of the odometry position update. A noise free path is calculated first to determine the end point position and orientation. Statistics are calculated for errors in end points of paths where measurement noise is added to the wheel measurements on each sample. The error covariance matrix for all the samples is calculated and printed. Also the mean of the errors is calculated to determine any bias that has been introduced. A convenient form of the covariance matrix is also used whereby the square root of diagonal elements and the correlation coefficient of diagonal elements[3] are displayed. This allows easy comparison of the simulation results and theoretical results.

An example output file corresponding to Figure 8 follows:

```
SampleRate=200.000000,    B=0.500000,    kl=0.001000,    kr=    0.001000,
MonteCarloRuns= 10000
     time Leftdist Rightdist:
   0.000000 0.000000 0.000000
   2.000000 2.000000 2.000000
   3.000000 1.607300 2.392700
   4.000000 3.570800 3.570800
   5.000000 3.374450 4.159850
```

[3] The correlation coefficient is $\frac{\sigma_{ij}}{\sqrt{\sigma_i^2 \sigma_j^2}}$ and lies between -1 and +1.

```
number of samples: 10000
mean is:
   -1.997124e-05
     5.32083e-05
    3.183715e-05
covariance matrix is:
     2.97987e-05   -4.650919e-05    -2.76081e-05
   -4.650919e-05    8.708272e-05     4.58468e-05
    -2.76081e-05     4.58468e-05    3.436819e-05
stand dev, correlation matrix is:
        0.005459       -0.913007         -0.8627
       -0.913007        0.009332         0.83804
         -0.8627         0.83804        0.005862
========== LK theoretical cov matrix:
    3.031791e-05   -4.763405e-05   -2.817159e-05
   -4.763405e-05    8.974219e-05    4.699783e-05
   -2.817159e-05    4.699783e-05     3.48496e-05
stand dev, correlation matrix is:
        0.005506       -0.913208       -0.866688
       -0.913208        0.009473        0.840389
       -0.866688        0.840389        0.005903
```

The results for 3 simulation runs are shown in Table III for the conditions matching Figure 8. The agreement between theory and simulation is within the statistical variation displayed by three runs.

TABLE III - 10000 Monte-Carlo runs for $k_L = k_R = 10^{-3} m^{½}$, *B=0.5 m*

	Means	**Covariance Matrix**	**Stand dev/ Corr Matrix**
Theoretical results	0 0 0	3.032e-5 -4.763e-5 -2.817e-5 -4.763e-5 8.974e-5 4.700e-5 -2.817e-5 4.700e-5 3.48496e-5	0.005506 -0.913208 -0.8667 -0.9132 0.009473 0.8404 -0.8667 0.8404 0.005903
Run 1 (10000 samples)	-1.997e-5 5.321e-5 3.184e-5	2.980e-5 -4.651e-5 2.761e-5 -4.651e-5 8.708e-5 4.585e-5 -2.761e-5 4.585e-5 3.437e-5	0.005459 -0.9130 -0.8627 -0.9130 0.009332 0.8380 0.8627 0.8380 0.005862
Run 2 (10000 samples)	1.992e-5 -2.702e-5 -5.061e-5	3.018e-5 -4.715e-5 -2.813e-5 -4.715e-5 8.884e-5 4.657e-5 -2.813e-5 4.657e-5 3.476e-5	0.005494 -0.9106 -0.8684 -0.9106 0.009426 0.8380 -0.8684 0.8380 0.005896

Run 3 (10000 samples)	-3.199e-5 -2.236e-5 -4.512e-5	3.040e-5 -4.774e-5 -2.823e-5 -4.774e-5 8.995e-5 4.726e-5 -2.823e-5 4.726e-5 3.499e-5	0.005513 -0.9131 -0.8657 -0.9131 0.009484 0.8424 -0.8657 0.8424 0.005915

Three more runs were conducted with the conditions of Figure 9 and the results are summarised in Table IV:

TABLE IV - 10000 Monte-Carlo runs for $k_L = 0.001 m^{\frac{1}{2}}$ $k_R = 0.002 m^{\frac{1}{2}}$, *B=0.5 m*

	Means	**Covariance Matrix**	**Stand dev/ Corr Matrix**
Theoretical results	0 0 0	5.680e-5 -0.000104 -5.782e-5 -0.000104 0.000228 0.000119 -5.782e-5 0.000119 8.477e-5	0.007536 -0.9153 -0.8332 -0.9153 0.01511 0.8519 -0.8332 0.8519 0.009207
Run 1 (10000 samples)	-3.005e-5 -0.000125 -2.289e-5	5.656e-5 -0.000104 -5.743e-5 -0.000104 0.000228 0.000118 -5.743e-5 0.000118 8.398e-5	0.00752 -0.9129 -0.8334 -0.9129 0.01508 0.8519 -0.8334 0.8519 0.009164
Run 2 (10000 samples)	-0.000119 7.654e-5 1.374e-5	5.536e-5 -0.000102 -5.667e-5 -0.000102 0.000225 0.000117 -5.667e-5 0.000117 8.427e-5	0.00744 -0.9130 -0.8297 -0.9130 0.0150 0.8512 -0.8297 0.8512 0.00918
Run 3 (10000 samples)	-1.774e-5 -7.531e-5 2.308e-5	5.676e-5 -0.000104 -5.765e-5 -0.000104 0.000227 0.000118 -5.765e-5 0.000118 8.4560e-5	0.007534 -0.9137 -0.8320 -0.9137 0.01507 0.8511 -0.8320 0.8511 0.009196

The theoretical results derived in Sections 3, 4 and 5 depend on the absolute values of the errors being small. To study the effect of larger errors, the noise variance has been increased to values of $k_L = 0.01 m^{\frac{1}{2}}$ $k_R = 0.02 m^{\frac{1}{2}}$. Table V summarises the results obtained.

TABLE V - 10000 Monte-Carlo runs for $k_L = 0.01 m^{\frac{1}{2}}$ $k_R = 0.02 m^{\frac{1}{2}}$, *B=0.5 m*

	Means	**Covariance Matrix**	**Stand dev/ Corr Matrix**
Theoretical results	0 0 0	0.00568 -0.0104 -0.005782 -0.0104 0.02283 0.01185 -0.005782 0.01185 0.008477	0.07536 -0.9153 -0.8333 -0.9153 0.1511 0.8519 -0.8333 0.8519 0.092069
Run 1 (10000 samples)	-0.005359 -0.003173 -8.374e-5	0.005586 -0.010132 -0.005629 -0.010132 0.02239 0.011583 -0.005629 0.011583 0.008284	0.07474 -0.9060 -0.8274 -0.9060 0.1496 0.8505 -0.8274 0.8505 0.09102

Run 2 (10000 samples)	-0.006619 -0.001239 0.000884	0.005623 -0.010224 -0.00566 -0.010224 0.022542 0.011663 -0.00566 0.011663 0.008374	0.07499 -0.9081 -0.8248 -0.9081 0.1501 0.8489 -0.8248 0.8489 0.09151
Run 3 (10000 samples)	-0.005757 -0.003603 -0.000882	0.005695 -0.010359 -0.005843 -0.010359 0.022763 0.011943 -0.005843 0.011943 0.008611	0.07546 -0.9098 -0.8344 -0.9098 0.1509 0.8530 -0.8344 0.8530 0.09280

8. Conclusion and Future Work

Based on a simple model for odometry errors in each wheel of a robot, results have been derived that describe the error statistics of robot trajectories in straight lines, arcs and turns. The results allow efficient computation of the odometry covariance matrix at the end of each path segment of the robot. Previous approaches rely on updating the covariance matrix after small time increments [Wang, 1988 #87].

The model of odometry errors allows sensible fusing of localisation data from other sensors, since the relative errors of each can be assessed. A Kalman Filter can be employed in the merging process.

The results of the covariance estimates for constant curvature paths can be applied to a varying curvature path by breaking the path up into small segments of approximately constant curvature.

Simulation studies have been employed to confirm the mathematical correctness of the derivations and study the limitations of approximations employed in the derivations. Future work will attempt to experimentally verify the results derived in this report using a mobile robot. A calibration technique for determining the error model parameters will be investigated.

References

[1] L Kleeman. "Optimal estimation of position and heading for mobile robots using ultrasonic beacons and dead-reckoning", in *IEEE International Conference on Robotics and Automation*. 1992. Nice FRANCE: pp. 2582-2587

[2] A H Jazwinski, *Stochastic Processes and Filtering Theory*. **1970, New York: Academic Press.**

[3] C M Wang. "Location estimation and uncertainty analysis for mobile robots", in *IEEE Conference on Robotics and Automation*. 1988. pp. 1230-1235