Accurate Odometry and Error Modelling for a Mobile Robot

Kok Seng CHONG
kok.seng.chong@eng.monash.edu.au

Lindsay KLEEMAN
lindsay.kleeman@eng.monash.edu.au

Intelligent Robotics Research Centre (IRRC)
Department of Electrical and Computer Systems Engineering
Monash University, Clayton Victoria 3168, Australia
http://calvin.eng.monash.edu.au/IRRC/index.html

Abstract

This paper presents a low cost novel odometry design capable of achieving high accuracy dead-reckoning. It also develops a statistical error model for estimating position and orientation errors of a mobile robot using odometry. Previous work on propagating odometry error covariance relies on incrementally updating the covariance matrix in small time steps. The approach taken here sums 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. Closed form error covariance matrix is developed for a general circular arc and two special cases : (I) straight line and (II) 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 were first calibrated with UMBmark [BorFen94]. Experimental results show that, despite its low cost, our system's performance, with regard to dead-reckoning accuracy, is comparable to some of the best, award-winning vehicles around. The statistical error model, on the other hand, needs to be improved in light of new insights.

1 Introduction

One of the major tasks of autonomous robotics navigation is localisation. In a typical indoor environment with a flat floorplan, localisation becomes a matter of determining the Cartesian coordinates (x,y) and the orientation θ, collectively known as the *state*, of the robot on a two dimensional floorplan. For a typical two wheel robot, odometry (also known as *dead-reckoning*) remains to be one of the most important means of achieving this task. Odometry is the measurement of wheel rotation as a function of time. If the two wheels of the robot are joined to a common axle, the position and orientation of the centre of the axle relative to the previous position and orientation can be determined from odometry measurements on both wheels. In practice, optical encoders that are mounted onto both drive wheels feed discretised wheel increment information to the central processor, which in turn continually updates the robot's state using geometric equations. However, with time, odometric localisation accumulates errors in an unbounded fashion due to wheel slippage, floor roughness and discretised sampling of wheel increments. A lot of research works have been undergone at both the hardware and theoretical level to improve the reliability of odometry.

At the hardware level, [BarDur95] determine the position of their robot based on inertial navigation with gyroscopes and/or accelerometers, but this method has been proven to be susceptible to drift. The remainder of this paragraph showcases the dead-reckoning implementations of a few robot vehicles from the University of Michigan: Cybermotion K2A utilises synchro-drive, which makes it insensitive to non-systematic errors. CLAPPER, consisted of two TRC LabMates connected by a compliant linkage, uses two rotary encoders to measure the rotation of the labmates relative to the compliant linkage, and a linear encoder to measure the relative distance between their centrepoints, giving it the unique ability to measure and correct non-systematic dead-reckoning errors during motion. In the attempt to improve the dead-reckoning performance of a tracked vehicle called Andros, a two-wheeled encoder trailer is attached to the back of the vehicle, which is able to freely rotate on the horizontal plane. The rotations of the trailer wheels and the trailer with respect to Andros are measured with the attached optical encoders. More details about these robots can be found in [BorFen94]. To the best of the authors' knowledge, the robot used in the experiment has a unique, lowest cost mechanical design which differs from all the existing ones while still achieve comparable accuracy.

Work done at the theoretical level normally involves error quantification via modelling, so that some kind of mathematical treatment would be possible. For instance, many robust stochastic based techniques such as the Extended Kalman Filter (EKF) require that the odometry errors be statistically quantified in the form of an *error covariance matrix*, so that it can be combined with the information provided by the external reference to produce a linear *minimum mean square* estimate of the position. Therefore, high level methods also sometimes imply the utilisation of additional external referencing of position.

Normally, odometry errors can be classified as being systematic or non-systematic, and it is a common engineering practice to first identify the sources of systematic errors and have them calibrated prior to using the system. In the work by Borenstein and Feng [BorFen94], a calibration technique called UMBmark test has been developed to calibrate out the systematic errors suffered by a typical two wheel robot. The dominant systematic error sources are identified as being the difference in wheel diameter and the uncertainty about the effective wheel base. The experiment designed requires that robot be moved around a square path in both the clockwise (CW) and counterclockwise (CCW) senses several times. The average Cartesian offsets, known as the centres of gravity, from the initial positions are assumed to be a sum (superposition) of the errors contributed by both systematic error sources. The error model parameters are then solved and incorporated into the software as tuning factors. In our work, this method has been used to calibrate the robot.

To be able to propagate the error covariance matrix of the robot's state following a change of stage is the main focus of this chapter. A commonly used method [Durra93, Jenki93] is now presented. Supposed the at stage k-1, the state of the robot is $\mathbf{S}_{k-1} = \begin{bmatrix} x_{k-1} & y_{k-1} & \theta_{k-1} \end{bmatrix}^T$, which comprises its two dimensional Cartesian coordinates (x_{k-1}, y_{k-1}) and orientation θ_{k-1} with respect to a global reference frame. It then performs a rotation α_k followed by a translation D_k to move to a new state $\mathbf{S}_k = \begin{bmatrix} x_k & y_k & \theta_k \end{bmatrix}^T$. Applying simple geometry,

$$\begin{bmatrix} X_k \\ Y_k \\ \theta_k \end{bmatrix} = \mathbf{f}\left(\begin{bmatrix} X_{k-1} \\ Y_{k-1} \\ \theta_{k-1} \end{bmatrix}, \begin{bmatrix} D_k \\ \alpha_k \end{bmatrix} \right) = \begin{bmatrix} X_{k-1} \\ Y_{k-1} \\ \theta_{k-1} \end{bmatrix} + \begin{bmatrix} D_k \cos(\theta_{k-1} + \alpha_k) \\ D_k \sin(\theta_{k-1} + \alpha_k) \\ \alpha_k \end{bmatrix} \qquad (1)$$

To propagate the error covariance matrix associated with the state matrix to the next stage, the error incurred is assumed to be small so that first order Taylor's expansion in the form of Jacobian matrix does not introduce significant higher order errors. Given the error covariance matrix of $\mathbf{S}_{k-1}$ and the input vector $\mathbf{u}_k = \begin{bmatrix} D_k & \alpha_k \end{bmatrix}^T$, and given the intuition that the error in stage *k-1* is not correlated with the error introduced by the input, the covariance matrix of the next stage, *k*, can be evaluated as follows,

$$
\begin{aligned}
\mathbf{Cov}(\mathbf{S}_k) &= E\left\{ \left(\mathbf{S}_k - \overline{\mathbf{S}}_k\right)\left(\mathbf{S}_k - \overline{\mathbf{S}}_k\right)^T \right\} \\
&\approx \begin{bmatrix} \nabla_{\mathbf{S}_k}\mathbf{f} & \nabla_{\mathbf{u}_k}\mathbf{f} \end{bmatrix} \begin{bmatrix} \mathbf{Cov}(\mathbf{S}_{k-1}) & \mathbf{0} \\ \mathbf{0} & \mathbf{Cov}(\mathbf{u}_k) \end{bmatrix} \begin{bmatrix} \nabla_{\mathbf{S}_k}\mathbf{f}^T \\ \nabla_{\mathbf{u}_k}\mathbf{f}^T \end{bmatrix} \\
&= \nabla_{\mathbf{S}_k}\mathbf{f}\mathbf{Cov}(\mathbf{S}_{k-1})\nabla_{\mathbf{S}_k}\mathbf{f}^T + \nabla_{\mathbf{u}_k}\mathbf{f}\mathbf{Cov}(\mathbf{u}_k)\nabla_{\mathbf{u}_k}\mathbf{f}^T
\end{aligned}
\qquad (2)
$$

where

$$
\nabla_{\mathbf{S}_k}\mathbf{f} = \begin{bmatrix} \frac{\partial X_k}{\partial X_{k-1}} & \frac{\partial X_k}{\partial Y_{k-1}} & \frac{\partial X_k}{\partial \theta_{k-1}} \\ \frac{\partial Y_k}{\partial X_{k-1}} & \frac{\partial Y_k}{\partial Y_{k-1}} & \frac{\partial Y_k}{\partial \theta_{k-1}} \\ \frac{\partial \theta_k}{\partial X_{k-1}} & \frac{\partial \theta_k}{\partial Y_{k-1}} & \frac{\partial \theta_k}{\partial \theta_{k-1}} \end{bmatrix}, \quad \nabla_{\mathbf{u}_k}\mathbf{f} = \begin{bmatrix} \frac{\partial X_k}{\partial D_k} & \frac{\partial X_k}{\partial \alpha_k} \\ \frac{\partial Y_k}{\partial D_k} & \frac{\partial Y_k}{\partial \alpha_k} \\ \frac{\partial \theta_k}{\partial D_k} & \frac{\partial \theta_k}{\partial \alpha_k} \end{bmatrix} \text{ and } \mathbf{Cov}(\mathbf{u}_k) = \begin{bmatrix} \sigma_D^2 & 0 \\ 0 & \sigma_\alpha^2 \end{bmatrix}
\qquad (3)
$$

In the authors' opinion, the major problem with this treatment is that there is no physical basis in assuming that the translation error is uncorrelated with the rotation error [Durra93, CheCro92, LuMil95, Nishi95]. Model parameters do not give physical insight into the characteristics of the system.

The model is also inconsistent. For the same path, if propagation of error is done in multiple parts, the model yields different solution. To illustrate, suppose that $\mathbf{S}_{k-1} = [0 \quad 0 \quad 0]^T$ and $\mathbf{Cov}(\mathbf{S}_{k-1}) = \mathbf{0}$. Compare the following two scenarios :

(i) $\mathbf{u}_{k0} = [2D \quad 0]^T$ with $\mathbf{Cov}(\mathbf{u}_{k0}) = \begin{bmatrix} \sigma_{D0}^2 & 0 \\ 0 & \sigma_{\alpha 0}^2 \end{bmatrix}$.

From equation (2), $\mathbf{Cov}(\mathbf{S}_k) = \begin{bmatrix} \sigma_{D0}^2 & 0 & 0 \\ 0 & 4D^2\sigma_{\alpha 0}^2 & 2D\sigma_{\alpha 0}^2 \\ 0 & 2D\sigma_{\alpha 0}^2 & \sigma_{\alpha 0}^2 \end{bmatrix}$

(ii) $\mathbf{u}_{k1} = [D \quad 0]^T$ followed by $\mathbf{u}_{k1} = [D \quad 0]^T$, with $\mathbf{Cov}(\mathbf{u}_{k1}) = \begin{bmatrix} \sigma_{D1}^2 & 0 \\ 0 & \sigma_{\alpha 1}^2 \end{bmatrix}$.

By applying equation (2) twice, $\mathbf{Cov}(\mathbf{S}_k) = \begin{bmatrix} 2\sigma_{D1}^2 & 0 & 0 \\ 0 & 5D^2\sigma_{\alpha 1} & 3D\sigma_{\alpha 1}^2 \\ 0 & 3D\sigma_{\alpha 1}^2 & 2\sigma_{\alpha 1}^2 \end{bmatrix}$

Even by setting $\mathbf{Cov}(\mathbf{u}_{k0})=2\cdot\mathbf{Cov}(\mathbf{u}_{k1})$, the two cases yield different final state error covariance, even though they lead to the same final state by following exactly the same path. The authors consider this model as inadequate. [CheCro92] have tried to resolve this problem by performing error propagation for every time increment on the wheel encoders. This approach is conceptually similar to numerical integration but suffers high computational cost. The physical reasoning which leads to the error model is also questionable.

The theoretical work by [Wang88] has introduced a more realistic, physically-based error model for an arbitrary circular arc motion. Furthermore, Wang has attempted to circumvent the inadequacy of first order approximation by rigorously evaluating the error covariance of the new state by integrating certain parts constituting the state change expressions. The result is a model which is very accurate for large wheel turn variance, but limited in its applicability to a range of rotation angle. For large rotation angle, the robot path has to be divided into small segments in which the total turning angle is within the limitation of the model.

Other methods of representing position error includes the 'circular-error probable' (CPE) by [Leenh85] which defines a *confidence circle* about the estimated vehicle's position, and the vehicle has a 0.5 probability of being inside the circle. This representation is questionable because it is well known that the position error is usually not equal in all directions. [Krant96] has proposed the use of equal-error probability *isoline* for a similar purpose, and has outlined some ways of growing the isoline as the vehicle moves. It remains uncertain whether any existing robust mathematical techniques could be adapted to these novel representations, and if none could, whether new and sound methods could be developed to make use of them.

The new non-systematic error model developed by the authors has a strong physical basis which is closely related to the design of the robot. The model also generates error representation in the form of a error covariance matrix, which is the standard operating block for a multitude of robust noise filtering tools. Unlike the model illustrated earlier, the new model is consistent in a multiple path segments scenario. The computational load in incrementally updating the covariance matrix in small time steps, as done in [CheCro92] has been lifted because simple close form formulae have been derived for three simple path types: (I) circular arc motion (II) straight line (III) rotation about the centre of the axle. Complex paths can be divided into sections which can be approximated by the aforementioned cases, hence the model can be applied on a section by section basis. Unlike the model by [Wang88], it is valid for arbitrary distance and rotation angle. Even though the model by [Wang88] is more accurate when the errors are large, compared to the first order accuracy of the new model, in the authors' opinion, should the robot be operating in conditions likely to incur large errors, an accurate representation of these large errors is insufficient. Instead, extra measures should be adopted to correct for such errors when they arise, such as employing external referencing.

The remainder of this chapter is organised into six sections: Section 2 summarises the principles the authors adhere to in dealing with error modelling. Section 3 presents the novel odometry system, and states, with justification, the key assumptions being incorporated into the model. Section 4 describes the UMBmark test used for reducing systematic errors and lists the key equations used. This is followed by section 5 on the derivation of the proposed non-systematic error model and highlights of its features. The details and results of calibration of systematic errors and validation of the proposed error model constitute section 6. Lastly, section 7 is the conclusion which points out some inadequacy of the model and suggests possible future work for addressing these issues.

2 Principles of Error Modelling - A Summary

In our works, error modelling has been carried out based on following steps:

Error Classification The uncertainties in data are contributed by many error sources. Error classification allows certain errors to be eliminated, and strategies to be developed to decouple the actual errors of interest from the errors introduced by the measuring devices. In practice, errors are grouped into two major categories:

- **Systematic Errors** : are errors that recur during every run of experiment. It is possible to calibrate systematic errors prior to tackling random errors. In our work, systematic errors are assumed to be predominantly caused by unequal diameters on both wheel, inaccurate estimation of effective wheel base [BorFen94], and, as the external referencing tool, the offsets of the sonar sensor from the midpoint of wheel axle and forward orientation of the robot.

- **Non-systematic Errors** : refer to the random errors and are often characterised by their mean vectors and error covariance matrices. In most cases, the mean vector is zero after calibration and therefore often ignored. In the context of our work, non-systematic errors are assumed to be primarily caused by wheel slippage and backlash, non-continuous sampling of wheel increments, the noise associated with sensor reading caused by the fluctuation of speed of sound and imperfections in reference targets [KleKuc93], and, discretisation of the pan encoder measuring the pan motor to 0.18° per step.

Error Modelling Once the error types and sources are identified, model can be developed to describe the interaction between the system's dynamic and the error sources. In the context of our work, this means

- **Parameterisation of Errors** : determination of how the errors are best quantified. For systematic errors, the parameterisation usually takes the form of a multiplicative constant to the nominal value. In our work, the real wheel base is regarded as proportional to the nominal wheel base, and the diameters of both wheel sustain a constant ratio [BorFen94]. For non-systematic errors, error covariance matrices are the preferred form of representation.

- **Process Modelling** : determination of mathematical expressions which describe the dynamic behaviour of a system. These expressions decide how the error parameters are incorporated into the state of the system after a *state transition*. For systematic errors, the expressions translate the modelling parameters into measurable experimental data after a change of state, so they can be estimated and imbued into the software as 'tuning factors'. For non-systematic errors, the expressions generate the error covariance matrices for the final state by incorporating the statistical nature (once again parameterised) of the change into the initial state.

3 Robot Design and Assumptions

The robot used in the experiment has two pairs of wheels : the actual drive wheels, and the encoder wheels that generate odometry measurements. The encoder wheels are as sharp-edged as practically possible to reduce the wheel base (B) uncertainty, and are unloaded because they are independently mounted on linear bearings which allow vertical motion, hence the problem of wheel distortion is minimised. In the authors' belief, this design greatly improves the reliability of odometry measurements since wheel slippage and load deformation are no longer significant.

Based on the design, the following assumptions are made before proceeding to the next stage of model development.

It is reasonable to assume that for a short unit of travel, the error incurred on both wheels are uncorrelated. because the two drive wheels are driven by two different motors, and two separate optical shaft encoders are used to gather odometry information. This assumption is adopted by [BorFen94, Kleem95].

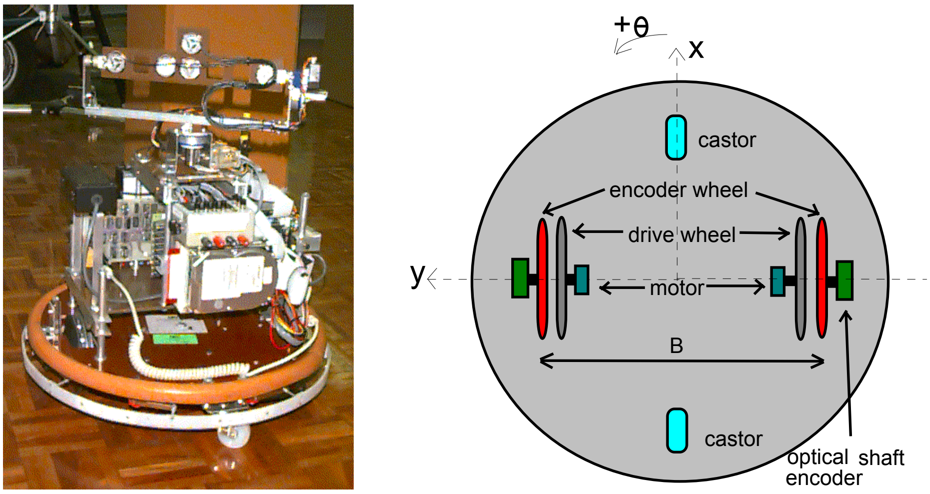


Figure 1 : Left : Sonar sensing robot with accurate odometry system. Right : Design of precise odometry system

Our work takes the assumption one step further. For a short unit of travel, the error is assumed to be zero mean, and white, that is, uncorrelated with the previous or next unit of travel. The variance of the cumulative error is then the sum of the variance of each statistically independent unit. This leads to a reasonable assumption that the variance of each unit of travel is proportional to the distance travelled

$$\begin{aligned}\sigma_L^2 &= k_L^2|d_L| \\ \sigma_R^2 &= k_R^2|d_R|\end{aligned} \qquad (4)$$

where d_L and d_R are the distances travelled by each wheel, and k_L^2 and k_R^2 are constants with unit m$^{1/2}$.

4 Calibration of Systematic Error Using UMBmark Test

UMBmark test [BorFen94] has been used for the calibration of wheel base error and unequal wheel diameter error. The principles and mathematical details of the procedures can be found in [BorFen94] so only key equations are summarised here. In short, the robot was programmed to travel a square path of side D in the clockwise sense (CW) for a number of times, say n, and the offsets of the final Cartesian coordinates from the initial Cartesian coordinates, $ex_{i,CW}$, $ey_{i,CW}$ were recorded. The experiment was repeated for the counterclockwise sense (CCW) and $ex_{i,CCW}$, $ey_{i,CCW}$ were recorded. The 'tuning factors' required to be incorporated into the software to counteract the effect of the systematic errors have been calculated from the weighted Cartesian offsets in both senses. As summary, the centres of gravity of the offsets can be computed from the their averages

$$x_{c.g.,CW/CCW} = \tfrac{1}{n}\sum_{i=1}^{n} ex_{i,CW/CCW} \tag{5}$$

$$y_{c.g.,CW/CCW} = \tfrac{1}{n}\sum_{i=1}^{n} ey_{i,CW/CCW} \tag{6}$$

With the two pairs of centres of gravity, the tuning factors for the wheel base, the radius of left wheel and the radius of right wheel, c_b, c_l and c_r, can be found by following this sequence of computations:

$$\alpha = average\left(\frac{x_{c.g.,CW} + x_{c.g.,CCW}}{-4D}, \frac{y_{c.g.,CW} - y_{c.g.,CCW}}{-4D}\right) \tag{7}$$

$$\beta = average\left(\frac{x_{c.g.,CW} - x_{c.g.,CCW}}{-4D}, \frac{y_{c.g.,CW} + y_{c.g.,CCW}}{-4D}\right) \tag{8}$$

$$E_d = \frac{D + B\sin(\beta/2)}{D - B\sin(\beta/2)} \tag{9}$$

$$c_b = \pi / (\pi - \alpha) \tag{10}$$

$$c_l = 2/(E_d + 1) \tag{11}$$

$$c_r = E_d c_l \tag{12}$$

and finally, the measure of dead-reckoning accuracy for systematic errors has been defined in [BorFen94] as

$$E_{\max,syst} = \max\left(\sqrt{x_{c.g.,CW}^2 + y_{c.g.,CW}^2}, \sqrt{x_{c.g.,CCW}^2 + y_{c.g.,CCW}^2}\right) \tag{13}$$

5 The New Non-systematic Error Model

With the new non-systematic error model, the entire path travelled by the robot is treated as consisting of k small segments. Propagation of error covariance is required to be done k times to obtain the error covariance of the final state. This section shows that it is possible to obtain a closed form solution for this model, as k approaches infinity. The solution for a general circular arc motion is first developed.

The solutions for two special cases, straight line motion and on-spot turn are then obtained by suitably taking limits.

Suppose that at segment *k-1*, the state of the robot is $\mathbf{S}_{k-1} = \begin{bmatrix} x_{k-1} & y_{k-1} & \theta_{k-1} \end{bmatrix}^T$. It then propels its left wheel by L_k metre and its right wheel by R_k metre, to bring the robot to a new state $\mathbf{S}_k = \begin{bmatrix} x_k & y_k & \theta_k \end{bmatrix}^T$. Over an infinitesimal time increment, the speed of the wheels can be assumed constant, hence the path takes on a circular arc trajectory with constant radius of curvature $\mathrm{r_k}$. Refer to Figure 2.

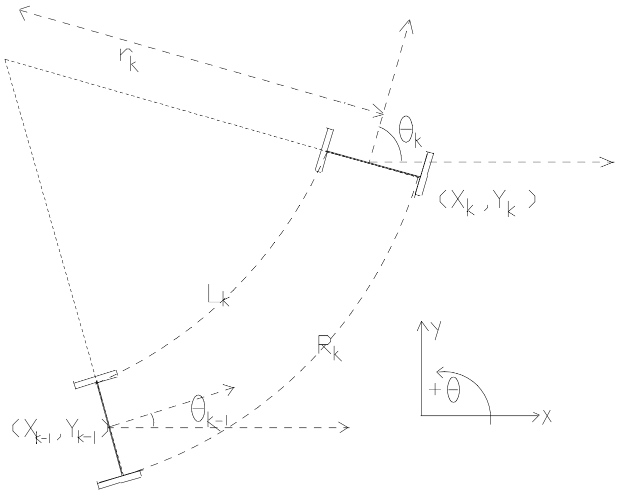


Figure 2 : Initial and final state of the robot after following a circular arc trajectory

Based on the assumption of constant radius of curvature, the following transformation equations can be derived:

$$\begin{bmatrix} X_k \\ Y_k \\ \theta_k \end{bmatrix} = \mathbf{f}\left(\begin{bmatrix} X_{k-1} \\ Y_{k-1} \\ \theta_{k-1} \end{bmatrix}, \begin{bmatrix} L_k \\ R_k \end{bmatrix} \right) = \begin{bmatrix} X_{k-1} \\ Y_{k-1} \\ \theta_{k-1} \end{bmatrix} + \begin{bmatrix} r_k \left[\sin\theta_{k-1} - \sin\left(\theta_{k-1} + \frac{R_k - L_k}{B}\right) \right] \\ r_k \left[\cos\left(\theta_{k-1} + \frac{R_k - L_k}{B}\right) - \cos\theta_{k-1} \right] \\ \frac{R_k - L_k}{B} \end{bmatrix} \tag{14}$$

where r_k is the instantaneous radius of curvature,

$$r_k = \frac{B}{2}\left(\frac{L_k + R_k}{L_k - R_k} \right) \tag{15}$$

The next step is to propagate the error covariance matrix associated with $(k\text{-}1)^{th}$ stage to the k^{th} stage. Given the error covariance matrix of $\mathbf{S}_{k-1}$ and the input vector $\mathbf{u}_k = \begin{bmatrix} L_k & R_k \end{bmatrix}^T$, and given the intuitive assumption that the error in stage *k-1* is not correlated with the error introduced by the input, the covariance matrix of the next stage, *k*, can be evaluated using equation (2), where

$$\nabla_{\mathbf{S}_k}\mathbf{f} = \begin{bmatrix} 1 & 0 & r_k\left[\cos\theta_{k-1} - \cos\theta_k\right] \\ 0 & 1 & r_k\left[\sin_{k-1} - \sin\theta_k\right] \\ 0 & 0 & 1 \end{bmatrix} \tag{16}$$

$$\nabla_{\mathbf{u}_k}\mathbf{f} = \begin{bmatrix} \frac{-BR_k}{(L_k-R_k)^2}\left[\sin\theta_{k-1} - \sin\theta_k\right] + r_k\cos\theta_k & \frac{BL_k}{(L_k-R_k)^2}\left[\sin\theta_{k-1} - \sin\theta_k\right] - r_k\cos\theta_k \\ \frac{-BR_k}{(L_k-R_k)^2}\left[\cos\theta_k - \cos\theta_{k-1}\right] + r_k\sin\theta_k & \frac{-BL_k}{(L_k-R_k)^2}\left[\cos\theta_k - \cos\theta_{k-1}\right] - r_k\sin\theta_k \\ -\frac{1}{B} & \frac{1}{B} \end{bmatrix} \tag{17}$$

and as stated earlier,

$$\mathbf{Cov}(\mathbf{u}_k) = \begin{bmatrix} \sigma_L^2 & 0 \\ 0 & \sigma_R^2 \end{bmatrix} = \begin{bmatrix} k_L^2|L_k| & 0 \\ 0 & k_R^2|R_k| \end{bmatrix} \tag{18}$$

Refer to Figure 2 again, suppose that the arc segment is now infinitesimally small, and the full path actually comprises k such segments being concatenated from end to end. The initial state of the robot is $\mathbf{S}_0$ which is at the starting end of the first segment and the last segment is $\mathbf{S}_k$ which is the destination of the last segment.

The expression for covariance propagation can be recursively expanded like a Markov process,

$$\begin{aligned} \mathbf{Cov}(\mathbf{S}_k) &= \nabla_{\mathbf{S}_k}\mathbf{f}\mathbf{Cov}(\mathbf{S}_{k-1})\nabla_{\mathbf{S}_k}\mathbf{f}^T + \nabla_{\mathbf{u}_k}\mathbf{f}\mathbf{Cov}(\mathbf{u}_k)\nabla_{\mathbf{u}_k}\mathbf{f}^T \\ &= \nabla_{\mathbf{S}_k}\mathbf{f}\left\{\nabla_{\mathbf{S}_{k-1}}\mathbf{f}\mathbf{Cov}(\mathbf{S}_{k-2})\nabla_{\mathbf{S}_{k-1}}\mathbf{f}^T + \nabla_{\mathbf{u}_{k-1}}\mathbf{f}\mathbf{Cov}(\mathbf{u}_{k-1})\nabla_{\mathbf{u}_{k-1}}\mathbf{f}^T\right\}\nabla_{\mathbf{S}_k}\mathbf{f}^T + \nabla_{\mathbf{u}_k}\mathbf{f}\mathbf{Cov}(\mathbf{u}_k)\nabla_{\mathbf{u}_k}\mathbf{f}^T \\ &= \left(\prod_{i=1}^{k}\nabla_{\mathbf{S}_i}\mathbf{f}\right)\mathbf{Cov}(\mathbf{S}_0)\left(\prod_{i=1}^{k}\nabla_{\mathbf{S}_i}\mathbf{f}\right)^T + \sum_{i=1}^{k}\left\{\left(\prod_{j=i+1}^{k}\nabla_{\mathbf{S}_j}\mathbf{f}\right)\mathbf{Cov}(\mathbf{u}_k)\left(\prod_{j=i+1}^{k}\nabla_{\mathbf{S}_j}\mathbf{f}\right)^T\right\} \end{aligned} \tag{19}$$

Let L_i , R_i denote the small increments in wheel turn at for the i^th^ segment, and for circular arc motion, $L = kL_i$, $R = kR_i$. Let r be the radius of curvature of the circular arc.

$$\nabla_{\mathbf{S}_i}\mathbf{f} = \begin{bmatrix} 1 & 0 & r\left(\cos\theta_{i-1} - \cos\theta_i\right) \\ 0 & 1 & r\left(\sin\theta_{i-1} - \sin\theta_i\right) \\ 0 & 0 & 1 \end{bmatrix} \tag{20}$$

It is possible to show, by induction, that

$$\prod_{i=m}^{i=n}\nabla_{\mathbf{S}_i}\mathbf{f} = \begin{bmatrix} 1 & 0 & r\left(\cos\theta_m - \cos\theta_n\right) \\ 0 & 1 & r\left(\sin\theta_m - \sin\theta_n\right) \\ 0 & 0 & 1 \end{bmatrix} \tag{21}$$

and

$$\nabla_{\mathbf{u}_i}\mathbf{f} = \begin{bmatrix} \frac{-BR_i}{(L_i-R_i)^2}\left[\sin\theta_{i-1} - \sin\theta_i\right] + \frac{r}{B}\cos\theta_i & \frac{BL_i}{(L_i-R_i)^2}\left[\sin\theta_{i-1} - \sin\theta_i\right] - \frac{r}{B}\cos\theta_i \\ \frac{-BR_i}{(L_i-R_i)^2}\left[\cos\theta_i - \cos\theta_{i-1}\right] + \frac{r}{B}\sin\theta_i & \frac{BL_i}{(L_i-R_i)^2}\left[\cos\theta_i - \cos\theta_{i-1}\right] - \frac{r}{B}\sin\theta_i \\ -\frac{1}{B} & \frac{1}{B} \end{bmatrix} \quad (22)$$

Therefore the covariance of $\mathbf{S}_k$ can be further evaluated to

$$\mathbf{Cov}(\mathbf{S}_k)=\begin{bmatrix}1 & 0 & r(\cos\theta_0-\cos\theta_k)\\ 0 & 1 & r(\sin\theta_0-\sin\theta_k)\\ 0 & 0 & 1\end{bmatrix}\mathbf{Cov}(\mathbf{S}_0)\begin{bmatrix}1 & 0 & r(\cos\theta_0-\cos\theta_k)\\ 0 & 1 & r(\sin\theta_0-\sin\theta_k)\\ 0 & 0 & 1\end{bmatrix}^T$$
$$+\sum_{i=1}^{k}\left\{\left(\prod_{j=i+1}^{k}\nabla_{\mathbf{s}_j}\mathbf{f}\right)\begin{bmatrix}k_L^2|L_i| & 0\\ 0 & k_R^2|R_i|\end{bmatrix}\left(\prod_{j=i+1}^{k}\nabla_{\mathbf{s}_j}\mathbf{f}\right)^T\right\} \qquad (23)$$

where

$$\prod_{j=i+1}^{k}\nabla_{\mathbf{s}_j}\mathbf{f}=\begin{bmatrix}\frac{-BR_i}{(L_i-R_i)^2}\left[\sin\theta_{i-1}-\sin\theta_i\right]+\frac{r}{B}\cos\theta_k & \frac{BL_i}{(L_i-R_i)^2}\left[\sin\theta_{i-1}-\sin\theta_i\right]-\frac{r}{B}\cos\theta_k\\ \frac{-BR_i}{(L_i-R_i)^2}\left[\cos\theta_i-\cos\theta_{i-1}\right]+\frac{r}{B}\sin\theta_k & \frac{BL_i}{(L_i-R_i)^2}\left[\cos\theta_i-\cos\theta_{i-1}\right]-\frac{r}{B}\sin\theta_k\\ -\frac{1}{B} & \frac{1}{B}\end{bmatrix} \qquad (24)$$

The sum of products part of equation (23) gives rise to an error covariance matrix henceforth known as $\mathbf{Cov}(\mathbf{U}_k)$. It is so named because it is contributed entirely by the error associated with the motion. Let $Cov(U_k)_{i,j}$ be the i$^{\text{th}}$ row, j$^{\text{th}}$ column component of $\mathbf{Cov}(\mathbf{U}_k)$, after taking k→∞

$$\begin{aligned}
Cov(U_k)_{1,1}&=\tfrac{-2r}{(L-R)^2}(k_R^2|R|L+k_L^2R|L|)(\sin\theta_0-\sin\theta_k)\cos\theta_k+\left(\tfrac{r}{B}\cos\theta_k\right)^2(k_R^2|R|+k_L^2|L|)\\
&\quad-\tfrac{B}{4(L-R)^3}\left[2(\theta_k-\theta_0)-\sin(2\theta_0)+\sin(2\theta_k)\right](k_R^2L^2|R|+k_L^2R^2|L|)\\
Cov(U_k)_{2,2}&=\tfrac{-2r}{(L-R)^2}(k_R^2|R|L+k_L^2R|L|)(\cos\theta_k-\cos\theta_0)\sin\theta_k+\left(\tfrac{r}{B}\sin\theta_k\right)^2(k_R^2|R|+k_L^2|L|)\\
&\quad-\tfrac{B}{4(L-R)^3}\left[2(\theta_k-\theta_0)+\sin(2\theta_0)-\sin(2\theta_k)\right](k_R^2L^2|R|+k_L^2R^2|L|)\\
Cov(U_k)_{3,3}&=\tfrac{1}{B^2}(k_R^2|R|+k_L^2|L|)\\
Cov(U_k)_{1,2}&=\tfrac{-r}{(L-R)^2}(k_R^2|R|L+k_L^2R|L|)(\cos(2\theta_k)-\cos(\theta_0+\theta_k))+\tfrac{1}{2}\left(\tfrac{r}{B}\right)^2\sin(2\theta_k)(k_R^2|R|+k_L^2|L|)\\
&\quad-\tfrac{B}{4(L-R)^3}\left[\cos(2\theta_0)-\cos(2\theta_k)\right](k_R^2L^2|R|+k_L^2R^2|L|)\\
Cov(U_k)_{1,3}&=\tfrac{1}{(L-R)^2}(k_R^2|R|L+k_L^2R|L|)(\sin\theta_0-\sin\theta_k)-\tfrac{r}{B^2}\cos\theta_k(k_R^2|R|+k_L^2|L|)\\
Cov(U_k)_{2,3}&=\tfrac{1}{(L-R)^2}(k_R^2|R|L+k_L^2R|L|)(\cos\theta_k-\cos\theta_0)-\tfrac{r}{B^2}\sin\theta_k(k_R^2|R|+k_L^2|L|)
\end{aligned} \qquad (25)$$

Where $sgn(x)=|x|/x$ if $x\neq0$ and 0 if $x=0$. These equations obviate the need to incrementally update the covariance matrix in small time steps. These are closed form expressions which are applicable to any circular arc motion with constant radius of curvature. For many applications, however, the robot will often need to perform two types of motion : straight line and rotation about the centre of the axle. The matrix terms can be further simplified by suitably taking limits.

5.1 Special Case 1 : Straight Line Translation

For a straight line path of length *D*, both wheels rotates by the same amount and the initial angle is approximately the same as the finally angle, hence *L,R*→*D* and $\theta_0 \to \theta_k$, The above equations can be simplified to

$$
\begin{aligned}
Cov(U_k)_{1,1} &= |D|\left[\tfrac{1}{4}\cos^2\theta_k(k_R^2+k_L^2)+\tfrac{D}{2B}\sin\theta_k\cos\theta_k(k_L^2-k_R^2)+\tfrac{D^2}{3B^2}\sin^2\theta_k(k_R^2+k_L^2)\right]\\
Cov(U_k)_{2,2} &= |D|\left[\tfrac{1}{4}\sin^2\theta_k(k_R^2+k_L^2)-\tfrac{D}{2B}\sin\theta_k\cos\theta_k(k_L^2-k_R^2)+\tfrac{D^2}{3B^2}\sin^2\theta_k(k_R^2+k_L^2)\right]\\
Cov(U_k)_{3,3} &= \tfrac{|D|}{B^2}(k_R^2+k_L^2)\\
Cov(U_k)_{1,2} = Cov(U_k)_{2,1} &= |D|\left[\left(\tfrac{1}{8}-\tfrac{D^2}{6B^2}\right)\sin(2\theta_k)(k_L^2+k_R^2)+\tfrac{D}{4B}\cos(2\theta_k)(k_R^2-k_L^2)\right]\\
Cov(U_k)_{1,3} = Cov(U_k)_{3,1} &= |D|\left[\tfrac{1}{2B}\cos\theta_k(k_R^2-k_L^2)-\tfrac{D}{2B^2}\sin\theta_k(k_R^2+k_L^2)\right]\\
Cov(U_k)_{2,3} = Cov(U_k)_{3,2} &= |D|\left[\tfrac{1}{2B}\sin\theta_k(k_R^2-k_L^2)+\tfrac{D}{2B^2}\cos\theta_k(k_R^2+k_L^2)\right] \qquad (26)
\end{aligned}
$$

An insight into the model is crucial at this point. Suppose that the initial state $\mathbf{S}_0 = \begin{bmatrix}0 & 0 & 0\end{bmatrix}^T$ and is accurately known, that is **Cov**(**S**$_0$)=**0**, so that the focus can be placed on the motion error. Physically, the robot moves in the *x* direction by *D*. The expressions for **Cov**(**U**$_k$) can be further simplified to

$$
\mathbf{Cov}(\mathbf{U}_k) = \begin{bmatrix}
\tfrac{D}{4}(k_R^2+k_L^2) & \tfrac{D^2}{4B}(k_R^2-k_L^2) & \tfrac{D}{2B}(k_R^2-k_L^2)\\
\tfrac{D^2}{4B}(k_R^2-k_L^2) & \tfrac{D^3}{3B^2}(k_R^2+k_L^2) & \tfrac{D^2}{2B^2}(k_R^2+k_L^2)\\
\tfrac{D}{2B}(k_R^2-k_L^2) & \tfrac{D^2}{2B^2}(k_R^2+k_L^2) & \tfrac{D}{B^2}(k_R^2+k_L^2)
\end{bmatrix} \qquad (27)
$$

The model predicts that variance in the direction perpendicular to the direction of motion is proportional to the cube of the distance travelled, whereas the variance in the direction of motion is only proportional to the distance travelled. Also note that the correlation coefficient between the perpendicular distance error and the orientation error is $\tfrac{D^2}{2B} / \sqrt{\tfrac{D}{B^2}\cdot\tfrac{D^3}{3B^2}} = 0.866$.

The model is also consistent. Unlike its classical counterpart, if propagation is done in multiple parts, the model generates exactly the same prediction. This is because the model itself is founded upon the concept of incrementally propagating error covariance from one infinitesimal section to the next.

5.2 Special Case 2 : Rotation about the Centre of Wheel Axle

For rotation about the centre of wheel axle, both wheels still rotate by the same amount, but in opposite direction. By letting $L \to \tfrac{B}{2}(\theta_k-\theta_0)$ and $R \to \tfrac{B}{2}(\theta_0-\theta_k)$, the above equations can be simplified to

$$
\begin{aligned}
Cov(U_k)_{1,1} &= \tfrac{B}{32}\left[2(\theta_k-\theta_0)-\sin(2\theta_0)+\sin(2\theta_k)\right](k_R^2+k_L^2)\operatorname{sgn}(\theta_k-\theta_0)\\
Cov(U_k)_{2,2} &= \tfrac{B}{32}\left[2(\theta_k-\theta_0)+\sin(2\theta_0)-\sin(2\theta_k)\right](k_R^2+k_L^2)\operatorname{sgn}(\theta_k-\theta_0)\\
Cov(U_k)_{3,3} &= \tfrac{|\theta_k-\theta_0|}{2B}(k_R^2+k_L^2)\\
Cov(U_k)_{1,2} = Cov(U_k)_{2,1} &= \tfrac{B}{32}\left[\cos(2\theta_0)-\cos(2\theta_k)\right](k_R^2+k_L^2)\operatorname{sgn}(\theta_k-\theta_0)\\
Cov(U_k)_{1,3} = Cov(U_k)_{3,1} &= \tfrac{1}{4}(k_R^2-k_L^2)(\sin\theta_k-\sin\theta_0)\operatorname{sgn}(\theta_k-\theta_0)
\end{aligned}
$$

$$Cov(U_k)_{2,3} = Cov(U_k)_{3,2} = \tfrac{1}{4}(k_R^2 - k_L^2)(\cos\theta_0 - \cos\theta_k)\operatorname{sgn}(\theta_k - \theta_0) \qquad (28)$$

6 Implementation and Results

The experimental work comprises three stages : simulation, calibration for systematic errors and analysis of non-systematic errors.

6.1 Simulation

Monte Carlo simulations have been carried out to examine the validity of approximations being made in the course of deriving the model, and to envisage the vulnerability of its first order basis to various combinations of wheel variances, initial covariance and travel distance. Altogether seven combinations of conditions have been simulated and the comparison of mean and covariance are tabulated in Table 1 to Table 7.

Number of Monte-Carlo Runs	**= 10000**
Wheel Base	**= 0.4m**
Number of Partitions of Path	**= 10000**
Initial State	**= [0 0 0]^T**

Table 1 : Simulation results with L=3m, R=1m, $[\sigma_x\ \sigma_y\ \sigma_\theta]^T=[0\ 0\ 0]^T$

	(I) $k_L=0.001m^{1/2}$ $k_R=0.001m^{1/2}$		(II) $k_L=0.003m^{1/2}$ $k_R=0.002m^{1/2}$		(III) $k_L=0.01m^{1/2}$ $k_R=0.02m^{1/2}$	
	Theory	Simul	Theory	Simul	Theory	Simul
mean x (m)	-0.38357	-0.383646	-0.38357	-0.383626	-0.38357	-0.383321
mean y (m)	-0.286535	-0.286721	-0.286535	-0.286685	-0.286535	-0.28738
mean θ (rad)	1.28319	1.28368	1.28319	1.28357	1.28319	1.28458
stddev x (m)	1.4376e-3	1.4253e-3	3.473e-3	3.4643e-3	2.4473e-2	2.4104e-2
stddev y (m)	2.4666e-3	2.4172e-3	6.494e-3	6.402e-3	3.6930e-2	3.5939e-2
stddev θ (rad)	5.0000e-3	4.9823e-3	0.01392	0.01392	6.6144e-2	6.5469e-2
corr xθ (%)	-59.47	-59.45	-64.81	-64.998	-52.882	-52.162
corr yθ (%)	-86.46	-86.19	-89.95	-89.875	-79.698	-78.803
corr xy (%)	41.90	42.43	50.38	51.148	29.783	29.471

Table 2 : Simulation results with L=15m, R=10m, $[\sigma_x\ \sigma_y\ \sigma_\theta]^T=[0\ 0\ 0]^T$

	(I) $k_L=0.001m^{1/2}$ $k_R=0.001m^{1/2}$		(II) $k_L=0.003m^{1/2}$ $k_R=0.002m^{1/2}$		(III) $k_L=0.01m^{1/2}$ $k_R=0.02m^{1/2}$	
	Theory	Simul	Theory	Simul	Theory	Simul
mean x (m)	-6.6322e-2	-6.7663e-2	-6.6322e-2	-6.748e-2	-6.6322e-2	-7.092e-2
mean y (m)	-2.2017e-3	-2.6005e-3	-2.2017e-3	-3.446e-3	-2.2017e-3	-2.249e-2
mean θ (rad)	6.637e-2	6.767e-2	6.637e-2	6.757e-2	6.637e-2	7.137e-2
stddev x (m)	1.5223e-2	1.5192e-2	3.933e-2	3.9251e-2	0.235605	0.23263
stddev y (m)	8.723e-3	8.704e-3	2.1383e-2	2.1352e-2	0.145819	0.147936
stddev θ (rad)	1.25e-2	1.2413e-2	3.3072e-2	3.2973e-2	0.185405	0.183095
corr xθ (%)	-82.348	-82.341	-84.3	-84.276	-78.975	-78.542
corr yθ (%)	-9.528	-9.686	-10.282	-10.424	-8.457	-9.375
corr xy (%)	7.826	6.992	8.6489	8.045	6.657	6.015

Table 3 : Simulation results with L=3m, R=1m, $[\sigma_x\ \sigma_y\ \sigma_\theta]^T=[0.05m\ 0.05m\ 2°]^T$

	(I) $k_L=0.001m^{1/2}$ $k_R=0.001m^{1/2}$		(II) $k_L=0.003m^{1/2}$ $k_R=0.002m^{1/2}$		(III) $k_L=0.01m^{1/2}$ $k_R=0.02m^{1/2}$	
	Theory	Simul	Theory	Simul	Theory	Simul
mean x (m)	-0.38357	-0.38294	-0.38357	-0.383273	-0.38357	-0.382952
mean y (m)	-0.286535	-0.287202	-0.286535	-0.287167	-0.286535	-0.287857
mean θ (rad)	1.28319	1.283435	1.28319	1.2833	1.28319	1.28434
stddev x (m)	5.099e-2	5.2362e-2	5.1094e-2	5.2508e-2	5.6546e-2	5.8017e-2
stddev y (m)	5.1794e-2	5.1077e-2	5.2141e-2	5.1425e-2	6.3564e-2	6.2373e-2
stddev θ (rad)	3.5e-2	3.5206e-2	3.7333e-2	3.7516e-2	7.4666e-2	7.4344e-2
corr xθ (%)	19.025	19.083	16.383	15.9996	-12.131	-11.71
corr yθ (%)	-25.979	-26.72	-27.822	-28.434	-50.717	-49.487
corr xy (%)	-4.937	-3.317	-4.524	-2.715	3.819	5.185

Table 4 : Simulation results with L=15m, R=10m, $[\sigma_x\ \sigma_y\ \sigma_\theta]^T$=[0.05m 0.05m 2°]^T

	(I) k_L=0.001m$^{1/2}$ k_R=0.001m$^{1/2}$		(II) k_L=0.003m$^{1/2}$ k_R=0.002m$^{1/2}$		(III) k_L=0.01m$^{1/2}$ k_R=0.02m$^{1/2}$	
	Theory	Simul	Theory	Simul	Theory	Simul
mean x (m)	-6.6322e-2	-6.7422e-2	-6.6322e-2	-6.723e-2	-6.6322e-2	-7.0508e-2
mean y (m)	-2.2017e-3	-3.3320e-3	-2.2017e-3	-4.175e-3	-2.2017e-3	-2.3202e-2
mean θ (rad)	6.63706e-2	6.7471e-2	6.63706e-2	6.727e-2	6.63706e-2	7.117e-2
stddev x (m)	5.2266e-2	5.3774e-2	6.3615e-2	6.5274e-2	0.240852	0.238806
stddev y (m)	5.0807e-2	5.0081e-2	5.4429e-2	5.3753e-2	0.15417	0.156657
stddev θ (rad)	3.6827e-2	3.7032e-2	4.7893e-2	4.8004e-2	0.18861	0.186645
corr xθ (%)	-8.004	-7.548	-35.903	-35.278	-75.935	-75.161
corr yθ (%)	-4.809	-5.509	-5.842	-6.722	-8.137	-9.165
corr xy (%)	0.3848	0.2199	2.0956	3.1899	6.159	5.8689

Table 5 : Simulation results with L=3m, R=1m, $[\sigma_x\ \sigma_y\ \sigma_\theta]^T$=[0.1m 0.1m 5°]^T

	(I) k_L=0.001m$^{1/2}$ k_R=0.001m$^{1/2}$		(II) k_L=0.003m$^{1/2}$ k_R=0.002m$^{1/2}$		(III) k_L=0.01m$^{1/2}$ k_R=0.02m$^{1/2}$	
	Theory	Simul	Theory	Simul	Theory	Simul
mean x (m)	-0.38357	-0.381971	-0.38357	-0.381949	-0.38357	-0.381608
mean y (m)	-0.286535	-0.286882	-0.286535	-0.286846	-0.286535	-0.287527
mean θ (rad)	1.28319	1.28306	1.28319	1.28296	1.28319	1.28398
stddev x (m)	0.103082	0.105786	0.103131	0.105885	0.105938	0.108937
stddev y (m)	0.105471	0.104072	0.110046	0.104245	0.111722	0.110046
stddev θ (rad)	0.087321	8.7836e-2	0.109738	8.8878e-2	0.10943	0.109738
corr xθ (%)	24.145	24.212	12.089	23.4391	11.401	11.444
corr yθ (%)	-31.768	-32.551	-39.425	-32.865	-39.767	-39.425
corr xy (%)	-7.669	-6.159	-3.428	-5.954	-4.783	-3.245

Table 6 : Simulation results with L=15m, R=10m, $[\sigma_x\ \sigma_y\ \sigma_\theta]^T$=[0.1m 0.1m 5°]^T

	(I) k_L=0.001m$^{1/2}$ k_R=0.001m$^{1/2}$		(II) k_L=0.003m$^{1/2}$ k_R=0.002m$^{1/2}$		(III) k_L=0.01m$^{1/2}$ k_R=0.02m$^{1/2}$	
	Theory	Simul	Theory	Simul	Theory	Simul
mean x (m)	-6.6322e-2	-6.6998e-2	-6.6322e-2	-6.678e-2	-6.6322e-2	-6.9807e-2
mean y (m)	-2.2017e-3	-4.0477e-3	-2.2017e-3	-4.887e-3	-2.2017e-3	-2.3840e-2
mean θ (rad)	6.6391e-2	6.7071e-2	6.6391e-2	6.6971e-2	6.6391e-2	7.0771e-2
stddev x (m)	0.101152	0.10396	0.107457	0.110578	0.255949	0.254892
stddev y (m)	0.100546	9.9042e-2	0.102424	0.100984	0.176909	0.179717
stddev θ (rad)	8.807e-2	8.8607e-2	9.324e-2	9.3723e-2	0.20488	0.203619
corr xθ (%)	-1.571	-1.003	-10.777	-10.228	-65.756	-64.149
corr yθ (%)	-5.81	-6.53	-6.039	-6.956	-7.699	-9.134
corr xy (%)	0.0913	1.9125	0.6508	-2.2438	5.0488	5.2486

Table 7 : Simulation results with L=15m, R=10m, $[\sigma_x\ \sigma_y\ \sigma_\theta]^T=[0\ 0\ 0]^T$

	(I) $k_L=0.05m^{1/2}$ $k_R=0.05m^{1/2}$		(II) $k_L=0.1m^{1/2}$ $k_R=0.1m^{1/2}$		(III) $k_L=0.5m^{1/2}$ $k_R=0.5m^{1/2}$	
	Theory	Simul	Theory	Simul	Theory	Simul
mean x (m)	-6.6322e-2	-5.7167e-2	-6.6322e-2	-2.795e-3	-6.6322e-2	0.372037
mean y (m)	-2.2017e-3	-0.189423	-2.2017e-3	-0.556837	-2.2017e-3	-0.310966
mean θ (rad)	0.0663706	0.0714706	0.0663706	0.0753706	0.0663706	0.181841
stddev x (m)	0.761173	0.67708	1.522347	1.1008	7.611734	2.436149
stddev y (m)	0.436142	0.491106	0.872283	1.03646	4.361422	2.401108
stddev θ (rad)	0.625000	0.62063	1.250000	1.241261	6.250000	6.206303
corr xθ (%)	-0.82348	-0.75888	-0.82348	-0.50629	-0.82348	0.057035
corr yθ (%)	-0.09528	-0.11516	-0.09528	-0.12235	-0.09528	0.041792
corr xy (%)	0.078263	0.068023	0.078263	0.020512	0.078263	0.019032

The values of k_L and k_R are not normally higher than 0.02m$^{1/2}$ (amounts to a wheel standard deviation of 0.02m in 1m) because in practice better odometry is achievable by most existing odometry systems. In the first six cases, the standard deviations are at least an order of magnitude larger than the difference between the theoretical mean and the simulated mean. This bias in mean is in turn a lot smaller than the distance travelled and initial covariance. Despite going as high as 0.02m for the worst k_L and k_R in the sixth case, it has little impact on the prediction of standard deviation. The discrepancy in the estimation of standard deviation is a lot less than the initial error and distance travelled, on average 0.005% to 0.3% from the shortest test distance to the longest test distance for moderate k_L and k_R . The prediction of correlation coefficient is generally good except for *corr xy*, which shows significant discrepancy at large initial covariance. Expectably, standard deviation is dominated by initial errors when they are large. Therefore, Table 7 investigates the limitation of the model alone at large k_R and k_L by setting the initial errors to zero. For distance around 10m, the model performs marginally good at k_R=k_L=0.05m$^{1/2}$. Beyond that, the performance deteriorates rapidly and a second order model becomes necessary. A orientation standard deviation of almost 2π for the largest wheel variances tested gives absolutely no angle information.

In practice, the estimation of mean is believed to be more superior because it is calculated with the data from the wheel encoders. Therefore, the model for predicting the non-systematic error covariance can be regarded as statistically sufficient.

6.2 Calibration for Systematic Error

In this part of the experiment, the wheel encoder measurements were used to calculated the perceived final state of the robot, whereas the sonar array mounted on top of the robot was to used to estimate its actual state by sensing some reference walls placed close to the initial state. Two reference walls were used to establish the robot's coordinates and orientation. Unfortunately, in collecting such statistics, errors were introduced by the sensor in two ways :

- **Systematic errors** : Horizontal offsets of the sonar array from the midpoint of the wheel axle, and the misalignment of sonar array 'zero' direction with the normal of the wheel axle.

- **Non-systematic errors** : Discretisation of pan mechanism into steps of 0.18°, and the fluctuation of speed of sound.

It remains unknown at this stage how these offset errors could be removed. The sonar array was positioned and aligned as accurately as possible, and the offsets were set to zero. The distance travelled was made large so that the odometry random errors outweigh the sensor errors. It is also justifiable to state that the non-systematic errors caused by the sonar sensor can be assumed small compared to the odometry errors.

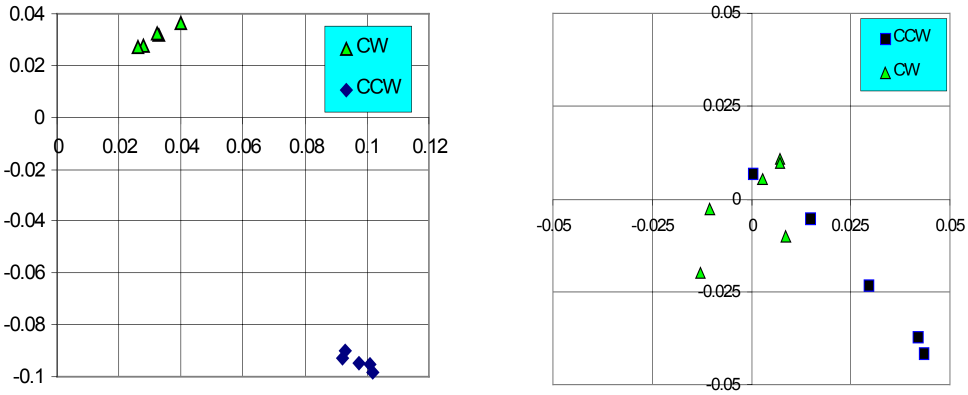


Figure 3 : Result of UMBmark test, before and after calibration.

The distribution of Cartesian offsets after the completion of *D*=4m square path for 5 runs in each sense (clockwise and counterclockwise), before and after calibration, are shown in Figure 3. The value of D has been chosen as such in order to make comparison with the results presented in [BorFen94].

Table 8 : Key results before and after calibration

	Before Calibration	After Calibration
$x_{c.g.,CCW}$ (mm)	97	26
$y_{c.g.,CCW}$ (mm)	-94	-20
$x_{c.g.,CW}$ (mm)	32	0.4
$y_{c.g.,CW}$ (mm)	31	-1
$E_{max,syst}$ (mm)	135	33 (4 folds)
wheel base (m)	0.37100	0.36898
left wheel radius (m)	0.06890	0.068662
right wheel radius (m)	0.06884	0.068871

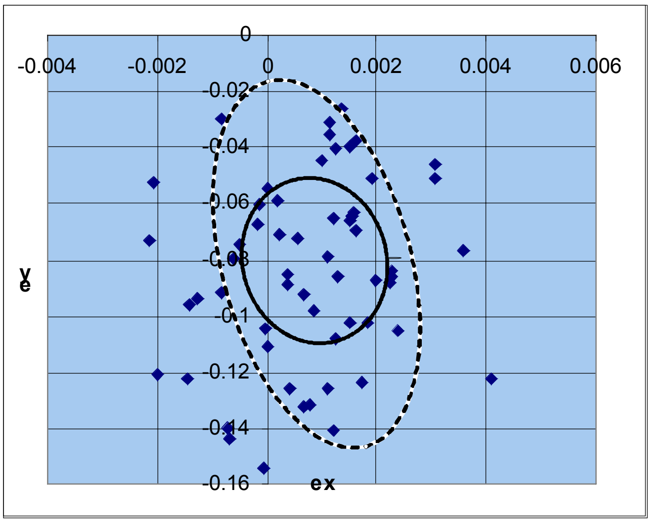
The calibration results are presented in Table 8. Comparison with other robot vehicles are made in Table 9, it can be seen that the measure of dead-reckoning accuracy for systematic errors, $E_{max,syst}$, is comparable to those achievable by many advanced but more costly odometry systems. To seek further improvement, further calibration has been carried out with the compensated parameters. It has been found that the parameter values fluctuated a little but insignificantly. The residual systematic errors could not be thoroughly removed despite repetitive trials.

Table 9 : Comparison of dead-reckoning accuracy and approximate cost of our design with four different vehicles. The first four sets of figures are obtained from [BorFen94]

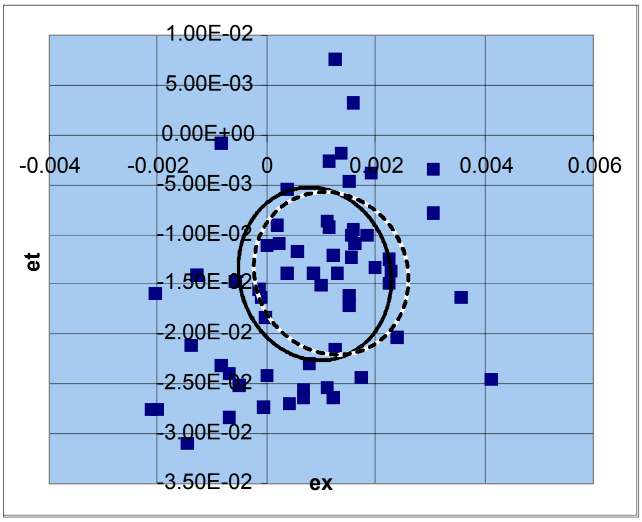
Tested Vehicle	Calibrated ?	$E_{max,syst}$ (mm)	Cost (US$)
TRC LabMate	yes	average 27	$10K
Cybermotion K2A (CARMEL)	unknown	63	<$50K
CLAPPER	yes	22	$30K
Andros with Encoder Trailer	yes	74	unknown
Ours	yes	33	$4K

6.3 Analysis of Non-systematic Error Source

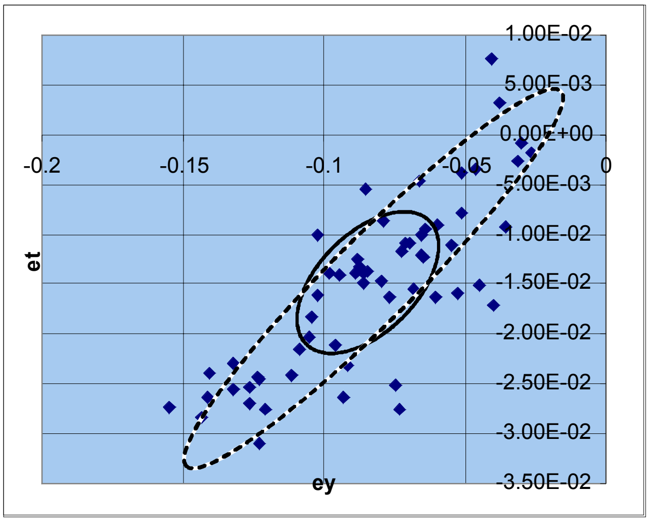
After calibration, the robot was programmed such that it scanned two reference walls, travelled forward 10 metres, travelled backward 10 metres, re-scanned the two reference walls and compared the position estimation from sonar sensing and odometry reading. The process was executed autonomously 60 times and the difference in the Cartesian coordinates, (*ex*, *ey*) and the difference in orientation *et* for all 60 runs are plotted against each other in Figure 4(a)-(c). By trial and error, it has been found that, if the robot is navigating on a flat parquetry floor, k_L=0.00040m$^{1/2}$, k_R=0.00058m$^{1/2}$, Expectably, these values are dependent upon the material of the floor. Despite various attempts, the author was unable to find the k_L and k_R that yield close approximation to all six elements in the actual covariance matrix. At this stage, it is believed that the y component, being the 'off track' error is very small, therefore highly sensitive to residual biases and outliers. Another possible cause is the misalignment of wheels. These likely factors have been taken into account while fitting k_L and k_R to the results by putting stronger emphasis on the 'along track' error and orientation error. In Figure 4, both the 95% confidence error ellipses of the actual data and the 95% confidence error ellipses generated with k_L and k_R are overlayed on the plots for comparison.



(a) *ex* versus *ey*



(b) ***ex*** **versus** ***et***



(c) ***ey*** **versus** ***et***

Figure 4 : Errors in the along-track direction (*ex*), the off-track direction (*ey*) and orientation (*et*), for 60 autonomous runs, are plotted against each other. The solid ellipses belong to the real data and the dashed ellipses are generated with the k_R and k_L obtained by trial and error.

This part of the experiment is yet to be perfected. More data points are required to obtain large enough clusters for fitting error ellipses. As the next stage, many more experiments will be conducted to obtain statistically significant validation.

7 Conclusion and Future Work

This chapter draws together the key considerations and procedures involved in the calibration and statistical quantification of odometry error for a novel robot design. A precise odometry system is presented which is comparable to the best reported system but can be fabricated at half the cost. A new first order odometry error model has been derived to propagate the state error covariance following a motion. The error model takes the form of a covariance matrix which is prevalent in statistics and filtering theory, hence could easily fit into many powerful tools such as the Extended Kalman Filter. The limitation of the linearisation has been investigated in simulation. This model cannot account for unexpected errors such as hitting a bump on the floor, which violates the flat floorplan assumption. Extra consideration is necessary. For certain applications such as mapping, external referencing should be deployed to correct such errors instead of treating them as systematic errors as done in [BorFen94]. Moreover, the values of k_L and k_R depend critically on the interaction between the encoder wheels and the floor. As future work, a new technique which employs laser beams will be developed to further validate the non-systematic error model.

8 Reference

[1] Wang, C.M., “Location Estimation and Uncertainty Analysis for Mobile Robots”, in IEEE Conference on Robotics and Automation, 1988, pp. 1230-1235

[2] Bar-Shalom, Y. and Li, X.R. "Estimation and Tracking: Principles, Techniques and software", Boston, London: Artech House Inc., 1993

[3] Barshan, B. and Durrant-Whyte, H. “An Inertial Navigation System for a Mobile Robot”, IEEE Transactions on Robotics and Automation, Vol 11, No. 3, June 1995, pp. 328-342.

[4] Borenstein, J. and Feng, L. “UMBmark - A method for Measuring, Comparing, and Correcting Dead-reckoning Errors in Mobile Robots.”, Technical Report UM-MEAM-94-22, University of Michigan

[5] Kleeman, L, “Odometry Error Covariance Estimation for Two Wheel Robot Vehicles”, Technical Report MECSE-95-1, 1995, Intelligent Robotics Research Centre, Department of Electrical and Computer Systems Engineering, Monash University

[6] Krantz, D. “Non-Uniform Dead-Reckoning Position Estimate Updates”, IEEE International Conferencve on Robotics and Automation, April 1996, Vol 2, pp 2061-2066.

[7] Leenhouts, P. “On the Computation of Bi-Normal Radial Error”, Navigation: Journal of The Institute of Navigation, Vol 32, No 1, Spring 1985, pp. 16-28.