Proceedings of 2004 IEEE/RSJ International Conference on
Intelligent Robots and Systems
September 28 - October 2, 2004, Sendai, Japan

Sensor Based Robot Localisation and Navigation: Using Interval Analysis and Unscented Kalman Filter.

Immanuel Ashokaraj, Antonios Tsourdos, Peter Silson & Brian A. White
Department of Aerospace, Power and Sensors,
Cranfield University (RMCS), Shrivenham, Swindon, UK, SN6 8LA.

Abstract— **Multiple sensor fusion for robot localisation and navigation has attracted a lot of interest in recent years. This paper describes a sensor based navigation approach using an interval analysis (IA) based adaptive mechanism for an Unscented Kalman filter (UKF). The robot is equipped with inertial sensors (INS), encoders and ultrasonic sensors. An UKF is used to estimate the robots position using the inertial sensors and encoders. Since the UKF estimates may be affected by bias, drift etc. we propose an adaptive mechanism using IA to correct these defects in estimates. In the presence of land marks the complementary robot position information from the IA algorithm using ultrasonic sensors is used to estimate and bound the errors in the UKF robot position estimate.**

I. INTRODUCTION

Robot navigation is primarily about guiding a mobile robot to a desired destination or along a pre-specified path in which the robots environment consists of landmarks and obstacles. In order to achieve this objective the robot needs to be equipped with sensors suitable to localize the robot throughout the path it has to follow. Most of these sensors may give overlapping or complementary information and sometimes be redundant as well. There are many different architectures to fuse these information. Mobile robots generally carry dead reckoning sensors such as wheel encoders, inertial sensors (INS), such as accelerometers, gyroscopes, to measure acceleration and angle rate respectively, and landmark and obstacle detecting and map making sensors such as time of flight (TOF) ultrasonic sensors. All these sensor measurements can be fused to estimate the robots position by using a sensor fusion algorithm. Sensor fusion in this case is the method of integrating data from distinctly different sensors to estimate the robots position.

Classical data fusion algorithms use stochastic filters such as Kalman filters for robot position estimation [1]. But one of the main disadvantages of using Kalman filters with ultrasonic sensors for robot localisation problems is that the data association step in Kalman filters is complex and also the fact that they are often affected by bias and drift from inertial sensors. Moreover an accurate model of the robot system and accurate statistics of the sensor noises are needed, which is not available accurately in many cases.

The paper is organised as follows. This introductory section continues by presenting a background for the problem of autonomous robot localisation in section I-A, followed by a summary of previous work in robot localisation using interval analysis (IA) in section I-B. Section III explains the implementation of the UKF with INS and encoders for this problem. Section IV gives a brief explanation of the IA algorithm for robot localisation and also describes how the sensor range limitation is incorporated. In section V the implementation of the adaptive mechanism for the UKF robot position estimation using IA with ultrasonic sensors is described and the results are shown and finally in section VI the conclusions are given.

A. Background

The problem considered here is that of robot navigation and localisation using multiple low cost sensors such as INS, encoders and ultrasonic sensors. Conventionally stochastic filters such as Extended Kalman filter (EKF) or Unscented Kalman filters (UKF) are used for robot localisation [2]. One of the main prerequisites for using Kalman filters when using INS and encoders, is to have an accurate model of the robot and also accurate sensor noise statistics (i.e) bias, drift etc. But in practice it is difficult to have these parameters accurately, especially the drifts in accelerometers and gyroscopes. This affects the outcome of the UKF, there by contributing to errors in the estimated position of the robot over a period of time.

Moreover TOF sensors such as ultrasonic sensors are used to measure the distance of land marks from the robot and to recognize the presence of any obstacles in the robots path. When the 2-D map of the environment in which the robot travels is known a priori, the distance measurements from the ultrasonic sensors can be used independently to estimate the position of the robot in the map. EKF can be used for this purpose as well [3]. But one of the main limitations encountered when using this approach is the problem of data association, as the data association problem in EKF is extremely complex and is of the third order O^3. There are ways in which this problem can be simplified to O^2, but the solution may be suboptimal.

In order to get the best estimate of the robots position, we can use different types of sensors with different algorithms which have different sources of error. In this case we use an Unscented Kalman filter (UKF) for fusing the data from the accelerometers, gyroscopes and encoders, instead of the EKF. This is because the UKF can linearize the nonlinear models at every instant up to the 3rd order of Taylor series expansion, there by reducing the errors during linearisation, where as in the EKF the nonlinear models can be linearised only up to the 1st order. In the case of ultrasonic sensors we use an Interval Analysis (IA) algorithm for estimating the robots position. It should also be noted that the IA algorithm for ultrasonic sensors bypasses the complex data association step and handles the problem in a nonlinear way even while been robust to outliners. Thus we have two independent sets of the measurements for the robot position. The estimated robot position using UKF from INS, which might be affected by bias and drift, are then fused with the estimated interval robot position using IA from ultrasonic sensors. The fused robot position estimate is much better than either one of them by itself, since the errors in UKF estimated position are identified and corrected using the IA algorithm.

B. Prior work: Robot localisation with IA using range measurements.

Interval analysis is basically about guaranteed numerical methods for approximating sets. Guaranteed in this context means that outer (and sometimes inner) approximations of the sets of interest are obtained, which can (at least in principle), be made as precise as desired. Thus interval computation is a special case of computation on sets, and set theory provides the foundations for interval analysis.

The localisation of an autonomous robot while navigating in a known or partially known environment is an important problem in mobile robotics. In this section an approach for the localisation of the robot using IA [4] with sensor readings from ultrasonic sensors is described briefly. The main advantage of this method is that it bypasses the data-association step, which is very complex in other stochastic methods such as Extended Kalman Filters, and it handles the problem in a nonlinear way without any linearisation and it is very much robust to outliners.

The robot model is assumed to move in an known 2D environment and its motion is planned with respect to a set of landmarks. These landmarks are defined in the world reference frame "W". The robots position is described by the parameters x_c, y_c and θ,which form the configuration vector $p = (x_c, y_c, \theta)^T$ and it is shown in *Figure* 4.

So the task now is to estimate the value of the configuration vector $\mathbf{p}$, from a map representing the environment of the robot and from distance measurements provided by a belt of n_s TOF ultrasonic sensors with limited range present in the mobile robot. Moreover since it is assumed that the bounds on the distance measurement error is known, the resulting distance measurement is in terms of intervals which is stored in an interval vector

$$[d] = ([d_1], ..., [d_n]) \tag{1}$$

If a model is available to model the ultrasonic sensor interval distance measurements represented by the interval vector $d_m(p)$, when the robot configuration is $\mathbf{p}$, the localization problem now becomes a bounded error parameter estimation problem, namely that of characterizing the set

$$P = \{p \in [p_o] \mid d_m(p) \in [d]\} \tag{2}$$

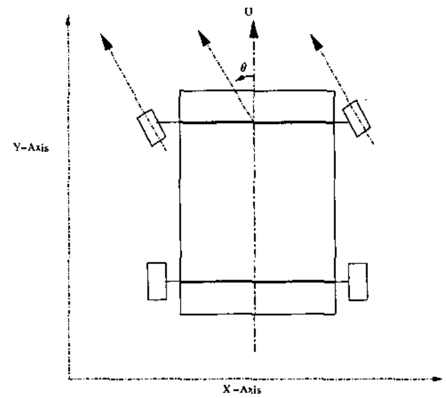

Fig. 1. Robot Model

where $[p_o]$ is an initial search box, assumed to be large enough to contain all the possible robot configurations. "P" then contains all the configurations vectors that are consistent with the given map and measurements.

But the task is to find the configuration vector $\mathbf{p}$ and so the equation given above can be rearranged as

$$= [p_o] \cap (d_m)^{-1}([d]) \tag{3}$$

(i.e.) for a given configuration vector $\mathbf{p}$ the robot evaluates the measurements that its sensors would return and compares then with the actual measurements to check whether they are consistent.

The problem described by the *Equation* 3 could then be solved using any of the two approaches namely SIVIA (Set Inversion Via Interval Analysis) [5] and ImageSP [6]. Both the above approaches have been described in detail in the book by Jaulin et al [7] and a brief introduction to both SIVIA and IMAGESP is given in Sections IV-A, IV-B.

II. Robot Model.

A kinematic representation of a 4-wheel robot that moves slowly in a 2D plane is shown in the *Figure* 1. The model for the vehicle is simplified with the "bicycle model" [8].

III. Robot localisation using UKF with Inertial sensors and encoders

By fusing the measurement data from the sensors - wheel encoders, gyroscopes and accelerometers - in the mobile robot, a reliable estimation of the position and heading of the robot can be obtained. There are basically two well established approaches available in literature: one is the Kalman filter and the other is the extended Kalman filter (EKF) [9]. The Kalman filters are well known methods used in the theory of stochastic dynamic systems, which can be used to improve the quality of estimates of unknown quantities. The difference between the two methods is that for the first one a linear kinematic model is used, while for the second one, the EKF a nonlinear dynamic model is used. We know that, if we use the nonlinear model, it is much more difficult to tune the performance of the filter. But in order to use all the available information, a nonlinear model is preferred. Most often in real world engineering applications, the most widespread and reliable state estimator for nonlinear systems is the extended Kalman filter [10].

The EKF is the Kalman filter of an approximate model of the nonlinear system, which is linearised to the first order around the most recent estimate. Assuming all the stochastic processes are Gaussian, the first order linearisation must be carried out at every iteration before applying the KF algorithms.

For the extended Kalman filter, the robot model equations can be rewritten as the state equation of the form shown below,

$$x_{k+1} = f(x_k, u_k, w_k) \tag{4}$$

which, when linearized will be of the form

$$x_{k+1} \approx \tilde{x}_{k+1} + A(x_k - \hat{x}_k) + Bu_k + Ww_{k-1}, \tag{5}$$

where, A and B are the jacobian matrix of partial derivatives.

This first order linearisation using the Taylor series expansion in the EKF may introduce errors in the estimated parameters which may lead to suboptimal performance and sometimes divergence of the filter.

The above described problem can be overcome to a certain extent by using a method first described by Julier and Uhlman as the unscented transform in the Kalman filter for the linearisation process [11]. This formulation of the Kalman filter for a nonlinear system is called the Unscented Kalman filter (UKF). The unscented transform is basically a deterministic sampling approach, where the state distribution is approximated by a gaussian random variable, but is now described using a minimal set of carefully chosen sample points. These sample points are chosen using the unscented transform method which completely describes the true mean and covariance of the gaussian random variable. When these chosen points are propagated through the true non-linear system, it can capture the posterior mean and covariance accurately up to the 3rd order for Taylor series expansion, where as in a EKF we can achieve only up to first-order accuracy. Also the need to compute the Jacobian matrices in the EKF is avoided when using the UKF. It should also be noted that the computational complexity of the UKF is the same order as that of EKF. The basic equations for the UKF has been given in detail in the book [12].

All the process noises are assumed to be zero mean, uncorrelated white random noises only.

In the measurement model for this robot there are four sources of observations that are considered:

1) velocity measurements from the wheel encoders,
2) acceleration from the accelerometers, which when integrated gives the velocity of the robot,
3) robot heading angle measurement from the encoders and
4) the angular velocity measurements from the rate gyroscope which when integrated once gives the heading angle of the robot.

Algorithm $[d_m]$ (in: $[\mathbf{p}]$; out: $[d_m]$)	
1	for $i := 1$ to n_s
2	$[s_i] := \begin{pmatrix} x_c \\ y_c \end{pmatrix} + \begin{pmatrix} \cos[\theta] & -\sin[\theta] \\ \sin[\theta] & \cos[\theta] \end{pmatrix} \tilde{s}_i$
3	$\overrightarrow{[u_{1i}]} := \begin{pmatrix} \cos([\theta] + \tilde{\theta}_i - \gamma) \\ \sin([\theta] + \tilde{\theta}_i - \gamma) \end{pmatrix};$
	$\overrightarrow{[u_{2i}]} := \begin{pmatrix} \cos([\theta] + \tilde{\theta}_i + \gamma) \\ \sin([\theta] + \tilde{\theta}_i + \gamma) \end{pmatrix};$
4	$[\mathrm{d_m}]_i([\mathrm{p}]) := +\infty;$
5	for $j := 1$ to n_w
	$[d_m]_i([p]) :=$
	$\min\left([d_m]_i([p]), [r]\left([s_i], \overrightarrow{[u_{1i}]}, \overrightarrow{[u_{2i}]}, a_j, b_j\right)\right).$

Fig. 2. Inclusion function for the measurement model

Thus, for both the velocity and heading angle of the robot there are two sets of measurements from two independent sensors as observations to the UKF and the UKF then estimates the best velocity and heading angle from which the robots position is calculated.

IV. Robot Localisation with Interval Analysis using Range Measurements.

In this section the robot localisation using IA as first described briefly in section I-B is further elaborated here.

A brief overview of the algorithm for a single measurement process has been given in the *Figures* 2 and 3. The *Figure* 2 is basically an interval inclusion function (i.e.) a mathematical model which models all the possible distance measurements expected from all the ultrasonic sensors when the configuration is $[\mathbf{p}]$, where a_j,b_j are the two extreme points of a segment of the land marks as shown in *Figure* 4. The *Figure* 3 is the interval mathematical model of a single ultrasonic sensor distance measurement. Its a simple sensor model with interval values which models the distance measurement as the smallest distance between the sensor vertex "s_i" (as shown in *Figure* 4) and the segment of a land mark $[\mathbf{a},\mathbf{b}]$, in four simple scenarios which is described in detail in [7].

The main improvement in this version of the algorithm in the descriptions in the tables is that the range of the ultrasonic sensor has been limited to 3 meters, where as in [4] the range was unlimited. This is implemented by identifying the sensors n_i that only give readings less than 3 meters (which is done by setting all the interval ranges greater than 3 meters to infinity in *Figure* 3) and substituting them for instead of n_s (where the inclusion function was calculated for all the n_s number of sensors) in the *Figure* 2.

Additionally the land marks which are visible to the robot in the 3 meter radius are only given to the inclusion function in *Figure* 2 instead of all the n_w segments, there by saving computational time.

This problem as given in *equation* 3 is then solved using any of the two approaches namely SIVIA (Set Inversion Via Interval Analysis) [5] and ImageSP. A brief

Algorithm $[r]$ (in : $[s], [\overrightarrow{u_1}], [\overrightarrow{u_2}], \mathbf{a}, \mathbf{b}$; out : $[r]$);

1. $[t_r] := \left(\det\left(\overrightarrow{\mathrm{ab}}, \overrightarrow{\mathrm{as}}\right) \geq 0\right)$
   if $[t_r] = 0$ then $[r] := +\infty$;
   return;
2. $[t_h] := \left(\left\langle \overrightarrow{\mathrm{ab}}, \overrightarrow{\mathrm{a}[\mathrm{s}]} \right\rangle \geq 0\right) \wedge \left(\left\langle \overrightarrow{\mathrm{ba}}, \overrightarrow{\mathrm{b}[\mathrm{s}]} \right\rangle \geq 0\right)$
   $\wedge \left(\left\langle \overrightarrow{\mathrm{ab}}, \overrightarrow{[\mathrm{u}_1]} \right\rangle \leq 0\right) \wedge \left(\left\langle \overrightarrow{\mathrm{ab}}, \overrightarrow{[\mathrm{u}_2]} \right\rangle \geq 0\right)$;
3. $[r_h] := [\chi]\left([t_h], [l]([s], (a, b)), +\infty\right)$;
4. $[t_a] := \left(\det\left(\overrightarrow{[\mathrm{u}_1]}, \overrightarrow{[\mathrm{s}]\mathrm{a}}\right) \geq 0\right) \wedge \left(\det\left(\overrightarrow{[\mathrm{u}_2]}, \overrightarrow{[\mathrm{s}]\mathrm{a}}\right) \leq 0\right)$;
5. $[r_a] := [\chi]\left([t_a], \left\|\overrightarrow{[s]a}\right\|, +\infty\right)$;
6. $[t_b] := \left(\det\left(\overrightarrow{[\mathrm{u}_1]}, \overrightarrow{[\mathrm{s}]\mathrm{b}}\right) \geq 0\right) \wedge \left(\det\left(\overrightarrow{[\mathrm{u}_2]}, \overrightarrow{[\mathrm{s}]\mathrm{b}}\right) \leq 0\right)$;
7. $[r_b] := [\chi]\left([t_b], \left\|\overrightarrow{[s]b}\right\|, +\infty\right)$;
8. for $i := 1$ to 2
   $[t_{h_i}] := \left(\det\left(\overrightarrow{[\mathrm{s}]\mathrm{a}}, \overrightarrow{[\mathrm{u_i}]}\right) \geq 0\right) \wedge \left(\det\left(\overrightarrow{[\mathrm{s}]\mathrm{b}}, \overrightarrow{[\mathrm{u_i}]}\right) \leq 0\right)$;
   $[r_{h_i}] := [\chi]\left([t_{h_i}], \left[l_{\overrightarrow{[\mathrm{u_j}]}}\right]([s], (\mathrm{a}, \mathrm{b})), +\infty\right)$;
9. $[r] := \min\left([r_h], [r_a], [r_b], [r_{h_1}], [r_{h_2}]\right)$;
10. $[r] := [\chi]\left([t_r], [r], +\infty\right)$

Fig. 3. Inclusion function for the remoteness of the cone from a segment

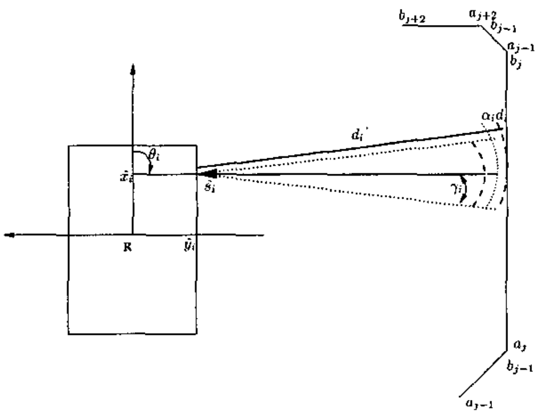

Fig. 4. Robot and Sensor model

introduction to both SIVIA and IMAGESP is given in the next two subsections.

A. Set Inversion Via Interval Analysis (SIVIA)

Set inversion is the computation of the reciprocal image

$$X = \{x \in R^n \mid f(x) \in Y\} = f^{-1}(Y) \qquad (6)$$

of a regular subpaving Y of R^m by a possibly nonlinear function $f : R^n \rightarrow R^m$ and SIVIA is a method to compute two subpavings $\underline{X}$ and $\overline{X}$ of R^n such that

$$\underline{X} \subset X \subset \overline{X} \qquad (7)$$

A subpaving is a finite set of non-overlapping boxes that are all included in the same root box. It is called regular if each of the boxes can be obtained by a finite succession of bisections and selections [7].

In this problem for robot localisation, $P = \{\mathbf{p} \in [p_o] \mid t(p) = 1\}$, SIVIA can be applied [4]. Therefore in this case, if $t_{[]}([p_o]) = 1$, p_o is in the solution set P and is stored in $\widehat{P}$. If $t_{[]}([p_o]) = 0$ then $[p_o]$ has an empty intersection with $\mathbf{P}$ and is dropped from further consideration. If $t_{[]}([p_o]) = [0, 1]$ and if the width of $[p_o]$ is larger than the pre-specified precision parameter ϵ , then p_o is bisected, leading to two child sub boxes $L(p)$ and $R(p)$, and the test $t_{[]}(.)$ is recursively applied to both of them. Any box with width smaller than ϵ is considered to be small enough and it is added to $\widehat{P}$. A diagram explaining SIVIA is given in *Figure* 5.

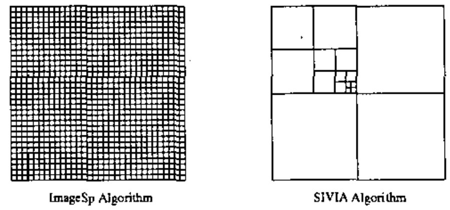

Fig. 5. A schematic representation of ImageSp and SIVIA algorithm

B. IMAGE SubPaving (IMAGESP)

When $\mathbf{f}$ is not invertible, a specific and computationally more demanding procedure is used. The basic idea of IMAGESP algorithm is to describe the initial search box p_o using a subpaving consisting of p boxes whose width are less than or equal to ϵ. Then IMAGESP evaluates the image of each of these p boxes using an inclusion function $f_{[]}$ of f and stores them on a list. Therefore we will be getting p image boxes, each of which contains the true image set of the associated initial box. At last, IMAGESP merges all these image boxes into a subpaving to allow further processing [7]. Thus when using the IMAGESP algorithm for the problem of robot localisation, it basically checks all the possible robot poses to obtain the true robot position. A diagrammatic representation of IMAGESP is given in *Figure* 5.

Thus the actual position of the robot represented here by the configuration vector $\mathbf{p}$ at any given instant of time can be found using SIVIA or IMAGESP algorithms, where basically the subpavings in SIVIA and IMAGESP represent the configuration vectors $\mathbf{p}$ in the robot localisation problem.

The *Figure* 4 gives a brief description of the sensor model and also the measurement process.

A detailed description of the above inclusion functions has been provided by [4]. Also a better version of the above algorithm in terms of computational time, incorporating the interval elementary tests to eliminate some of the infeasible configurations in the configuration vector has been described in [4], in which the problem is reformulated as $P = \{\mathbf{p} \in [p_o] \mid t(p) \text{ holdstrue}\}$ (i.e.) $P = \{\mathbf{p} \in [p_o] \mid t(p) = 1\}$, where $t(p)$ is a global test. The global test $t(p)$ consists of various elementary tests (three tests [4]) and they are robust to outliners as well as described in [4]. Also the purpose of the first two tests is to eliminate some infeasible configurations there by saving computational time. But if all the three tests are used when the range of the sensor is limited, it leads to scenarios in which some feasible configurations are ignored prematurely. Therefore only two tests were used (first test (inroom test) and the third test (data test (i.e.) $(d_m) \in [d])$ in [4]). The main consequence of not using the second test (i.e. leg in test) when the sensor range is limited is that it may increase the computation time.

In the case when the robot is moving the robots position needs to be tracked, in which case the robots position needs to predicted at the next instant to estimate the robots position at that instant. This is done by using the physical limitations of the robot based on the maximum possible speed and heading angle rate of the robot, instead of using a dynamic robot model with interval values [7] or position information from other sensors namely INS and encoders in this case. Thus we obtain an independent interval position of the robot from the ultrasonic sensors.

V. Interval Analysis based Adaptive Mechanism for UKF.

Sensor fusion is a very important and keenly researched topic in the domain of mobile robotics. This is due to the fact that, instead of using bespoke expensive sensors for estimating the robots position, multiple low cost sensors can be used, there by reducing the cost of developing the robot. Moreover these same sensors can be used to do other tasks other than estimating the robots position such as building the map of the robots environment using ultrasonic sensors etc. Also the source of errors in one sensor may be different from another one and this fact can be exploited to eliminate the errors in the measurements.

For the problem of sensor fusion stochastic filters, such as Kalman filters, are commonly used. But they suffer from the same disadvantages described before (i.e.) an accurate model of the system and statistics are needed. In order to overcome these difficulties and obtain a guaranteed position of the robot while using the UKF, a new approach has been introduced in this paper. As described above the robots position are estimated using two independent sources namely, the robot position from inertial sensors and the interval robot position from ultrasonic sensors.

The position obtained using IA is updated only once every second (due to computation time), where as the position from inertial sensors and encoders are updated 100 times per second. Moreover we know that the robots interval position to be guaranteed. The interval robot position thus obtained using ultrasonic sensors is like a plane or rectangle (i.e) basically an uniform distribution. Then the estimated position using the INS is checked whether they are inside this rectangle. In case they are present inside the rectangle then they are trusted to be accurate and used. Alternatively if they are not then both the measurements are fused by selecting the point on the rectangle (box) (i.e.) the interval robot position, which is geometrically closest to the robot position estimated using UKF with inertial sensors, thereby bounding the error in the UKF estimates. A block diagram of the sensor fusion method is given in Figure 6.

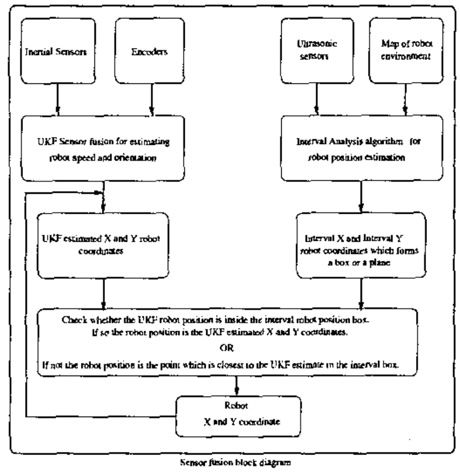

Fig. 6. Block diagram of IA based adaptive mechanism for UKF

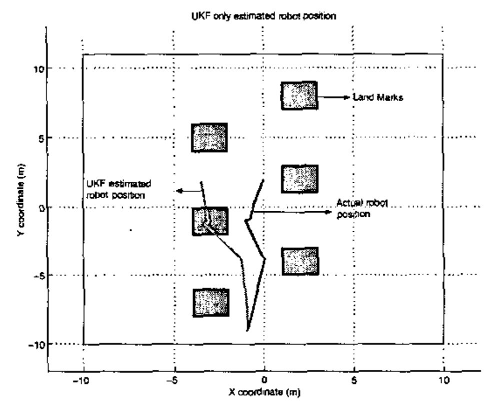

Fig. 7. UKF only estimated robot position

The algorithm described above has been successfully implemented in simulation using C++ and MATLAB software. *Figure* 7 shows the robot position estimate using UKF only, which is affected by sensor bias, drift etc. The *Figure* 8 shows the robot position after using the adaptive UKF robot position estimate using the SIVIA interval robot position for adaptation. Similarly the *Figure* 9 shows the UKF robot position estimation using the IMAGESP interval algorithm for the adaptive mechanism. In both cases it can be observed that the UKF estimate with the interval adaptive mechanism is much better than the UKF position estimate alone. It should be noted however that the SIVIA interval position uncertainty is greater than that of the IMAGESP algorithm. The reduction in uncertainty of the estimated interval robot position using the IMAGESP algorithm is attained at the cost of more computation time when compared with the SIVIA algorithm. This is because, in the IMAGESP algorithm, the whole initial subpaving is divided into many boxes of identical width less than or

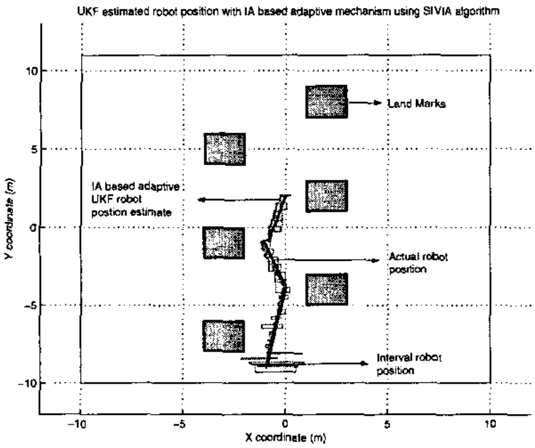

Fig. 8. Fused robot position using SIVIA algorithm as adaptive mechanism in UKF

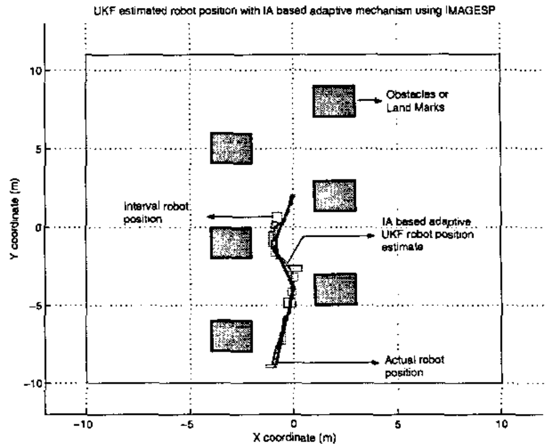

Fig. 9. Fused robot position using IMAGESP algorithm as adaptive mechanism in UKF

equal to ϵ (0.01 in our case) and the global test $t(p)$ is performed on all of these boxes.

In the SIVIA algorithm the global test $t(p)$ is performed on the initial subpaving and if the test result indicates that the actual position of the robot may be present inside the initial subpaving, the subpaving is bisected into two child subpavings and the global test is performed on each one of them. This process is repeated until the width of the subdivided boxes is less than or equal to ϵ and the resulting interval robot position is obtained.

It can be seen in *Figure* 9 that when the position estimate of UKF is fused with the IA it results to very small position error. That is because of the small interval of the position estimate obtained using IMAGESP. Hence even if IMAGESP interval estimate comes at the cost of more computation time when compared with the SIVIA algorithm it results in a much more accurate fused robot position estimate.

Moreover, in *Figure* 8, it can be seen that the interval position uncertainty increases as expected when there are no land marks visible to the robots ultrasonic sensors. This can be seen in the *Figure* 8 when the robot begins to move and also at instances in between landmarks, when the number of ultrasonic sensors for which the landmarks are visible is less.

VI. Conclusion

An Unscented Kalman filter (UKF) using an Interval Analysis (IA) based adaptive mechanism has been described. The UKF uses accelerometers, gyroscopes and encoders to measure the robots speed and heading angle, so that the robots position can be estimated. But the UKF robot position estimate is affected by errors in robot model, sensor bias, drift etc. The IA is a deterministic approach to estimating the robots position without using a model of the robot system, thereby minimizing errors due to inaccurate robot model. Additionally the previous work on robot localisation and navigation using IA has been extended, so as to incorporate sensor range limitation as well. Moreover instead of using dynamic interval model of the robot to predict the next step of the robot, the physical limitations of the robot are used to predict the next step of the robot in the IA algorithm.

The guaranteed IA robot position estimate is then used as an adaptive mechanism to correct the errors in the UKF robot position estimate.

References

[1] L. Jetto, S. Longhi, and G. Venturini, "Development and experimental validation of an adaptive extended kalman filter for the localization of mobile," *IEEE Transactions on Robotics and Automation*, vol. 15, no. 2, April 1999.

[2] S. Clark, G. Dissanayake, P. Newman, and H. Durrant-Whyte, "A solution to simultaneous localisation and map building (slam) problem," *IEEE Journal of Robotics and Automation*, vol. 17, no. 3, June 2001.

[3] J. Castellanos and J. Tardos, *Mobile Robot Localisation and Map Building: A Multisensor Fusion Approach.* Boston: Kluwer Ac. Pub., 1999.

[4] M. Kieffer, L. Jaulin, O. Didrit, and E. Walter, "Robust autonomous robot localization using interval analysis," *Reliable Computing*, vol. 6, no. 3, p. 337 to 362, 2000.

[5] L. Jaulin and E. Walter, "Set inversion via interval analysis for nonlinear bounded-error estimation," *Automatica*, vol. 29, no. 4, p. 1053 to 1064, 1993.

[6] M. Kieffer, L. Jaulin, and E. Walter, "Guaranteed recursive nonlinear state estimation using interval analysis," *Internal Report, Laboratoire des Signaux et Systemes*, 1998.

[7] L. Jaulin, M. Kieffer, O. Didrit, and E. Walter, *Applied Interval Analysis with examples in parameter and state estimation robust control and robotics.* London: Springer-Verlag, 2001.

[8] J. Alexander and J. Maddocks, "On the kinematics of wheeled mobile robots," *International Journal of robotics research*, vol. 8, no. 5, p. 15 to 27, Oct 1989.

[9] A. Alessandri, G. Bartolini, P. Pavanati, E. Punta, and A. Vinci, "An application of the extended kalman filter for integrated navigation in mobile robotics," *American Control Conference*, June 1997.

[10] G. Chen and C. Chui, *Kalman filtering with Real-Time Applications*, 2nd ed. New York: Springer-Verlag, 1991.

[11] S. Julier and J. Uhlmann, "A new extension of the kalman filter to nonlinear systems," *In Proc. of Aerosense: The 11thInt. Symp. on Aerospace/Sensing, Simulation and Controls*, 1997.

[12] E. Wan and R. V. der Merwe, *Kalman Filtering and Neural Networks, Chapter 7: The Unscented Kalman Filter.* Wiley Publishing, 2001.