Carnegie Mellon University
Research Showcase @ CMU

Robotics Institute | School of Computer Science

11-2007

Square Root Iterated Kalman Filter for Bearing-Only SLAM

Hyungpil Moon
Carnegie Mellon University

George Kantor
Carnegie Mellon University

Howie Choset
Carnegie Mellon University

Stephen Tully
Carnegie Mellon University

Follow this and additional works at: http://repository.cmu.edu/robotics

Part of the Robotics Commons

Published In
Proceedings of 4th International Conference on Ubiquitous Robots and Ambient Intelligence (URAI 2007).

This Conference Proceeding is brought to you for free and open access by the School of Computer Science at Research Showcase @ CMU. It has been accepted for inclusion in Robotics Institute by an authorized administrator of Research Showcase @ CMU. For more information, please contact research-showcase@andrew.cmu.edu.

Square Root Iterated Kalman Filter for Bearing-Only SLAM

Hyungpil Moon, George Kantor, Howie Choset
The Robotics Institute, Carnegie Mellon University, USA
hyungpil,kantor@cmu.edu, choset@cs.cmu.edu

Stephen Tully
ECE, Carnegie Mellon University, USA
stully@ece.cmu.edu

Abstract - This paper presents an undelayed solution to the bearing-only simultaneous localization and mapping problem (SLAM). We employ a square-root iterated Kalman filter for nonlinear state estimation. The proposed technique incorporates a modified Kalman update that is equivalent to a variable-step iterative Gauss-Newton method, and is numerically stable because it maintains a square-root decomposition of the covariance matrix. Although many existing bearing-only algorithms focus on proper initialization of landmark locations, our method allows for arbitrary initialization along the initial measurement ray without sacrificing map accuracy. This is desirable because we require only one filter and the state dimension of that filter need not include numerous temporary hypotheses. For this reason, the proposed algorithm is more computationally efficient than other methods. We demonstrate the feasibility of this approach in simulation and with experiments on mobile robots.

Keywords - Bearing-only SLAM, Iterated Kalman Filter, Square root filter

1. Introduction

Simultaneous localization and mapping (SLAM) has drawn extensive attention in the past decade (see the recent tutorial series [1], [2]). In part because of the availability of cost efficient monocular vision, bearing-only SLAM has received increased attention recently [3]–[8]. Many of these previous attempts employ an extended Kalman filter (EKF) framework and focus on proper initialization of landmark locations after obtaining several bearing measurements. Some methods choose to delay until a proper initialization is determined, which means information about the map is not represented in the state until initialization occurs [3], [4], [9]. Other methods require either a bank of filters or an expanded state dimension to develop an undelayed multi-hypothesis solution to the problem [5], [6]. An undelayed approach with one filter that does not expand the size of the state would be preferred. We approach this goal by using an iterated form of the Kalman filter that incorporates arbitrary initialization of landmarks and is numerically stable.

In [10], we have shown the problem of EKF based bearing-only SLAM. In this paper, we consider the perfect measurement and control example in [10] again to motivate the use of an iterated estimation technique. Suppose that a mobile robot moves with perfect odometry and there is a single landmark in the map (see Fig. 1). If the bearing measurements and the robot motion are perfect, but initially no information is available for the landmark, then two measurements of bearing at two distinctive known positions should be enough to localize the exact location of the landmark. It can be computed that even under this ideal case, with perfect measurements and motions, EKF based bearing only SLAM fails to exactly localize the landmark if its initialized state mean is not the same as the true location.

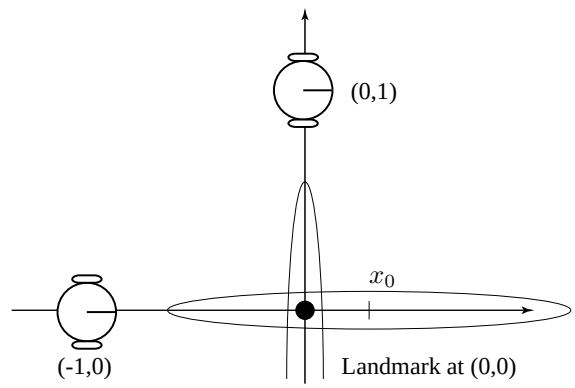

Fig. 1. An example of perfect measurement and motions. The uncertainty ellipses are drawn for illustration only (their thickness is supposed to be arbitrarily small).

When deriving the EKF update rule for the situation of perfect measurements and perfect robot control (Fig. 1 depicts the situation), we end up with

$$x_1 = x_0 - (x_0^2 + 1)\arctan(x_0) \tag{1}$$

where x_0 is the initialization of the landmark location. Notice that if $x_0 \neq 0$, x_1 will not equal zero (the correct location).

The measurement update rule, for the same situation, that results from the iterated Kalman filter can be expressed as follows.

$$x_{i+1} = x_i - (x_i^2 + 1)\arctan(x_i) \tag{2}$$

Note that (1) is just one iteration of (2). Apparently, the proper method is to iterate the update, which will converge upon the desired solution. We will show later that the iterated Kalman filter is an application of the Gauss-Newton method for approximating a maximum likelihood estimate [11], thus one should not iterate the Newton update step just once as EKF does. This is why we adopt the iterated Kalman filter (IKF) for bearing-only SLAM problem.

Furthermore, IKF with conventional covariance update equation tends to cause numerical instability more than EKF because of its double loop feedback structure. The conventional covariance update equation updates the covariance matrix by subtracting a positive definite matrix from a positive definite matrix. When the covariance matrix converges, all the landmarks become correlated through the robot state, thus becomes singular. It seems that, in this case, the conventional covariance update generates numerically non-positive definite matrix which is theoretically impossible. Thus, in this paper, we propose a square root form of iterated Kalman filter to improve the numerical stability.

For bearing only SLAM with vision sensors, data association is dealt with separately, using vision-based approaches. Specifically, we intend to use a monocular omni-directional camera for bearing detection and use SIFT features to associate measurements of landmarks as in [12], [13]. Therefore, throughout this paper, we assume that data association is solved and focus on the description of bearing-only SLAM using a modified IKF that employs variable-step Gauss-Newton optimization in the form of a square root filter using singular value decomposition.

We briefly review various approaches to bearing-only SLAM problem in Section 2. We introduce the conventional IKF and present a modified version of IKF for bearing-only SLAM as the Gauss-Newton method to nonlinear least square problem in Section 4. In Section 5, a numerically stable square root IKF algorithm is presented. In Section 6, we evaluate the proposed algorithm with simulations and present off-line results based on data from our own experiments. We close with a discussion in Section 7.

2. Related Work

Bearing-only SLAM is attractive, in part, because of the technical availability of low cost monocular vision. This is also an inherently more difficult problem than range-bearing SLAM because a bearing sensor does not provide sufficient information to estimate the full state of landmarks from a single observation. Bearing-only SLAM requires multiple observations from multiple poses, which exacerbates the typical SLAM challenge, inter-dependence between mapping and localization.

Delayed initialization of the landmark location is a popular method in bearing only SLAM. A batch update with all of the stored observations is demonstrated in [9]. In [3], initialization is postponed until a pair of measurements are distinguishable enough and the probability density of the corresponding landmark becomes sufficiently Gaussian. Once initialized, a single batch update is performed to refine and correct the map using remaining accumulated measurements. In [4], the persistence of landmark pose estimation is tracked without prior knowledge of data association. To incorporate the positioning and sensing uncertainties, the authors project measurements from the sensor space to the plane by approximating Gaussian distributions with bivariate ellipse representations.

Another popular method related to the extended Kalman filter (EKF) is the Gaussian sum filter (GSF). The GSF approximates arbitrary probability density functions by a weighted combination of many multivariate Gaussians. Since the GSF, which is introduced in [14], [15], requires retaining a large set of EKFs, its computational complexity is not desirable to SLAM problems, especially when a large number of landmarks are initialized simultaneously (typical for vision-based methods). In [5], a Sequential Probability Ratio Test (SPRT) is employed to reduce the number of members in the Gaussian sum by pruning highly improbable hypothesis, but this method still relies on a large number of filters during initialization. An approximated Gaussian sum method is proposed in [6] where a GSF is approximated by a set of parametrized cascaded Gaussian distributions and a single covariance matrix for all the Gaussians is managed and updated by Federated Information Sharing (FIS) which processes the information in a decentralized way. Even in this approximated form, a much larger state will be required.

Particle Filters (PFs), which incorporate non-Gaussian distributions, are widely used in SLAM research. In [7], particle filters are adopted for bearing measurements by associating hypothesized pseudo-ranges with each bearing measurement and by implementing a re-sampling procedure to eliminate improbable particles. In [16], a set of particles are maintained along the viewing ray of a landmark and landmark initialization is delayed until the range distribution is roughly Gaussian. In [17], a FastSLAM particle filter is used for single-camera SLAM with a partial initialization strategy which estimates the inverse depth of new landmarks rather than their depth for adding new landmarks to the map. In [8], authors compare the efficiency and robustness of EKFs, GSFs, and PFs for bearing-only SLAM. They implement a modified FastSLAM where the EKFs of the original FastSLAM in [18] are replaced by GSFs. Particle filter methods often require a very large number of particles for these bearing-only methods to work properly, making the technique computationally difficult.

3. Review of the Iterated Kalman Filter

The iterated Kalman filter (IKF) is an extension of the extended Kalman filter (EKF), a common tool for nonlinear state estimation. The primary difference between the two variants is that the state estimation of the IKF is repeatedly updated until it converges between measurement intervals. Thus, IKF has a double feedback loop structure for the whole process as oppose to a single loop for EKF. In this section, we briefly review the IKF as a Gauss-Newton method [11] and begin to develop motivation for using an iterated algorithm to solve the nonlinear problem of bearing-only SLAM.

Ideally, when we perform the Kalman filter update step, we would like to replace the state mean $\hat{x}^+_{k+1}$ with the maximum likelihood state estimate given the current measurement z_{k+1} and the predicted state estimate $\hat{x}^-_{k+1}$, P^-_{k+1}.

$$\hat{x}^+_{k+1} = \arg\max_x \; prob(x \mid z_{k+1}, \hat{x}^-_{k+1}, P^-_{k+1}) \qquad (3)$$

The solution to (3) is the minimization of the following nonlinear least-squares cost function:

$$c(x) = \begin{bmatrix} z_{k+1} - h(x) \\ x - \hat{x}_{k+1}^- \end{bmatrix}^T \begin{bmatrix} R & 0 \\ 0 & P_{k+1}^- \end{bmatrix}^{-1} \begin{bmatrix} z_{k+1} - h(x) \\ x - \hat{x}_{k+1}^- \end{bmatrix} \quad (4)$$

where $h(x)$ is the measurement model and R is the observation covariance matrix.

For a system with a linear measurement model, a direct solution is obtained via the traditional Kalman update equation, but for a system with a nonlinear measurement model, such as the bearing-only SLAM problem presented here, we must turn to numerical optimization methods. The Gauss-Newton algorithm iteratively solves the nonlinear least-squares minimization problem with the following recursive equation.

$$x_{i+1} = x_i - \gamma_i (\nabla^2 c(x_i))^{-1} \nabla c(x_i) \quad (5)$$

where γ_i is a parameter to vary the step-size. For traditional derivations of the IKF, $\gamma_i = 1$.

By substituting the appropriate Jacobian $\nabla c(x)$ into (5) along with the Hessian $\nabla^2 c(x_i)$, we arrive at the following update rule for the iterated Kalman filter, where H is the Jacobian of the measurement function $h(x)$. We refer the reader to [11] for details of this derivation.

$$\begin{aligned} x_0 &= \hat{x}_{k+1}^-, \; P_0 = P_{k+1}^- \\ K_i &= P_0 H_i^T (H_i P_0 H_i^T + R)^{-1} \\ P_{i+1} &= (I - K_i H_i) P_0 \quad (6) \\ x_{i+1} &= x_i + \gamma_i (H_i^T R^{-1} H_i + P_0^{-1})^{-1} \\ &\quad \cdot (H_i^T R^{-1} (z - h(x_i)) + P_0^{-1}(x_0 - x_i)) \quad (7) \\ &= x_0 + K_i (z_{k+1} - h(x_i) - H_i (x_0 - x_i)) \quad (8) \end{aligned}$$

In the equations above, the subscript k refers to the time-index and the subscript i refers to the iteration index. For a single update step at time k there may be many iterations of i before the state x_i converges. Once the IKF does converge, the estimate is overwritten with the result, $\hat{x}_{k+1}^+ = x_i$ and $P_{k+1}^+ = P_i$. In order to obtain the conventional IKF update equation (8), the variable-step parameter γ_i in (7) must equal one.

It is important to recognize that the EKF update rule is equivalent to performing just one iteration of the Gauss-Newton algorithm: taking (8) and setting $i = 0$ produces the well known EKF update equations below.

$$\begin{aligned} K_0 &= P_0 H_0^T (H_0 P_0 H_0^T + R)^{-1} \\ \hat{x}_{k+1}^+ &= x_1 = x_0 + K_0 (z_{k+1} - h(x_0)) \\ P_{k+1}^+ &= P_1 = (I - K_0 H_0) P_0, \quad (9) \end{aligned}$$

There are two points we would like to emphasize by formulating the IKF this way. The first is that an iterative numerical optimization method is required to compute the maximum likelihood estimate when the given application has a nonlinear measurement model. The second point is that the extended Kalman filter is just one step of this minimization process. To begin a numerical descent method and then stop only part way is often naive and in most cases will not provide the minimum of the objective function. Many applications do not suffer from this approximation. For example, range-bearing SLAM solutions have been successful and reliable when formulated as an EKF because the range and the bearing measurements provide fairly good estimations of the landmark locations. Unfortunately this is not the case for bearing-only SLAM, for which iteration is crucial. We later justify this claim by revisiting the analytical example introduced in Section 1.

4. Bearing-Only SLAM using an Iterated Kalman Filter

We will now formulate the IKF based bearing-only SLAM solution and focus on modifications that are specific to this application. The state will be defined as the robot pose $[x_R, y_R, \theta_R]$ appended by the locations of all N observed landmarks.

$$x_k = [x_R, y_R, \theta_R, x_{L_1}, y_{L_1}, x_{L_2}, y_{L_2}, ..., x_{L_N}, y_{L_N}]^T$$

For simulations and experiments, we will assume a unicycle-model for the mobile robot. The motion input $u_k = [v_k, \omega_k]^T$ contains the linear and rotational velocities of the robot at the corresponding time-step k and the state evolves according to the process model $f(x_k, u_k)$.

$$f(x_k, u_k) = x_k + \begin{bmatrix} I_{3x3} \\ 0_{2Nx3} \end{bmatrix} \begin{bmatrix} v_k \cos\theta_k \Delta t \\ v_k \sin\theta_k \Delta t \\ \omega_k \Delta t \end{bmatrix} \quad (10)$$

$$F_k = \frac{\partial f}{\partial x_k} = \begin{bmatrix} I_{3x3} \\ 0_{2Nx3} \end{bmatrix} \begin{bmatrix} 1 & 0 & -v_k \sin\theta_k \Delta t \\ 0 & 1 & v_k \cos\theta_k \Delta t \\ 0 & 0 & 1 \end{bmatrix} \quad (11)$$

$$W_k = \frac{\partial f}{\partial u_k} = \begin{bmatrix} I_{3x3} \\ 0_{2Nx3} \end{bmatrix} \begin{bmatrix} \cos\theta_k \Delta t & 0 \\ \sin\theta_k \Delta t & 0 \\ 0 & \Delta t \end{bmatrix} \quad (12)$$

The prediction step for our iterated method is equivalent to that which is used for standard implementations of EKF SLAM. The equations used in the prediction step are shown below.

$$\begin{aligned} \hat{x}_{k+1}^- &= f(\hat{x}_k^+, u_k) \\ P_{k+1}^- &= F_k P_k^+ F_k^T + W_k U W_k^T \end{aligned}$$

where $\hat{x}_k^+$ and P_k^+ are the state mean and covariance matrix from the previous time step, U is the covariance matrix for the motion input, and F_k and W_k are the Jacobians of the process model $f(x_k, u_k)$.

The nonlinear measurement model for an observation of the i-th landmark is

$$h_i(x_k) = \arctan\left(\frac{y_R - y_{Li}}{x_R - x_{Li}}\right) - \theta_R.$$

When performing the update step, all bearing measurements (obtained during the current time step) are appended to form a column vector z_k. The expected measurement $\hat{z}_k$ is computed by appending each appropriate $h_i(x_k)$ as a column vector of equal size. The Jacobian of the expected measurement is represented by H and is always computed from the current

state estimate. Also, we assume that each measurement has been disrupted by white Gaussian noise with covariance R. The update step is performed iteratively according to (7) until convergence.

4.1. Landmark Initialization

It is well known that the location of a new landmark cannot be estimated by a single bearing measurement. For this reason, accurate landmark initialization has been a common focus of previous work on bearing-only SLAM. Many solutions either implement a multiple hypothesis filter [5], [6], [14], [15] or delay until more information has been obtained [3], [4]. We choose to simplify this task by performing a naive undelayed initialization of the landmark. A ray is cast from the robot that represents all locations that agree with the initial observation. Because all points that lie on this ray have equal likelihood given the measurement, we can initialize the landmark mean arbitrarily on this ray according to (13). The state covariance matrix is also modified by appending a large uncertainty α for each dimension of the new landmark.

$$\hat{x}_k = \begin{bmatrix} \hat{x}_k \\ x_R + r_I \cos\left(\hat{\theta}_R + z_I\right) \\ y_R + r_I \sin\left(\hat{\theta}_R + z_I\right) \end{bmatrix}, \; P_k = \left[\begin{array}{c|cc} P_k & \multicolumn{2}{c}{0} \\ \hline 0 & \alpha & 0 \\ & 0 & \alpha \end{array}\right] \quad (13)$$

There are two design parameters for this method, r_I and α. α is the variance that is used to define the prior distribution of the landmark location. This value should be very large because the actual distribution is uniform. The parameter r_I is essentially a "range guess" and, if desired, could be chosen arbitrarily although large values of r_I experimentally produce better results (see Fig. 3). The sensitivity of the filter to the parameter r_I will be discussed in Section 6.

After initialization, a standard EKF update is performed. It is not necessary to iterate on this update. Only when previously initialized landmarks are observed again will it become necessary to perform an iterated update.

4.2. A Variable-Step IKF

In the previous section, we briefly reviewed the IKF as a full-step Gauss-Newton method that minimizes a nonlinear quadratic cost function. Introducing the step-size γ_i in the Gauss-Newton method (5) is required for bearing only SLAM because, in general, the Gauss-Newton method does not converge everywhere. What is guaranteed by (5) when $\gamma_i = 1$ is that the objective cost function begins to decrease as we start to move in the Newton direction. By taking the full Newton step ($\gamma_i = 1$), however, we may move too far for the approximated equation to be valid. Therefore, the cost does not necessarily decrease.

We choose to use a backtracking line search, which is one of many techniques that can be found in [19], [20]. Because our goal is minimizing the objective cost function (4) as we iterate, we compute the value of the cost at each step and check its decrease. The idea of backtracking is to ensure the average rate of decrease of the cost to be at least some fraction α of the initial rate of decease in the gradient direction. We note that we can always decrease the objective cost function by moving a very small step along the Newton direction.

Using a variable step-size approach prevents us from using the conventional IKF update equation (8). Therefore, we revert back to (7) and perform all updates in a form that more closely resembles the Gauss-Newton recursive step.

Iterating the update step is more costly than performing the update step once (EKF). We claim that the added computation is manageable because the number of iterations required to reach convergence is usually minimal. This can be attributed to the quadratic convergence of Newton's method. Also, after the landmark estimates have converged, the number of required iterations significantly decreases (see Fig. 4). We analyze the increase in computation experimentally in Section 6.

4.3. An Analytical Example

We now revisit the analytical example which is presented in the introduction and depicted in Fig. 1 to see how the IKF behaves for an estimation problem with bearing-only measurements.

By design, the IKF update equation for this specific example can be simplified directly from (7). The resulting iterative update expression is

$$x_{i+1} = x_i - \gamma_i (x_i^2 + 1) \arctan(x_i) \quad (14)$$

where x_0, as stated before, is the x-coordinate of the landmark when it is initialized.

When $\gamma_i = 1$ and x_0 is sufficiently small, the IKF converges upon the exact landmark location $(x_i, 0) = (0, 0)$ as iteration proceeds, which is obviously the only equilibrium. This is exactly what we hoped for. It is easy to see in this case, however, that there is a region of attraction; if $\gamma_i = 1$ and x_0 is sufficiently large, $\arctan(x_i)$ becomes near constant, the square term $-x_i^2$ dominates in the right hand side of (14), and thus the equilibrium of the iteration process becomes unstable.

Because the IKF is equivalent to a Gauss-Newton method, we can take a smaller step in the Newton direction to reduce the objective cost by selecting γ_i less than one. To guarantee the convergence of the iteration, we choose the step-size γ_i using backtracking method. Note that there always exists such a finite step $0 < \gamma_i \leq 1$ in the Newton direction that guarantees a reduction of the cost function as long as the Hessian is positive definite in (5).

Using an EKF update for this example, as reported in Section 1, produces the following final estimate for the landmark location $x_1 = x_0 - (x_0^2 + 1) \arctan(x_0)$. It can be seen that unless the initialization is perfect, this method will not produce the desired result. This example shows the importance of using an IKF instead of an EKF for bearing-only problems and also demonstrates the need for step-size control.

5. Square Root IKF

Unlike EKF SLAM, our variable step IKF algorithm, due to recursive iterations, creates an inner feedback loop between measurement intervals. The stability of this inner feedback

loop is guaranteed by the positive definiteness of the covariance matrix which is updated in the loop itself. The covariance matrix may become numerically non-positive definite (which is theoretically impossible) because of the numerically improper covariance update equation in the conventional IKF; the covariance matrix is updated by subtracting a positive definite matrix from the positive definite covariance in (6). We have seen in numerous simulations that it is indeed the case that the covariance matrix, when using the IKF for bearing-only SLAM, becomes non-positive definite due to numerical inaccuracies. The effect is divergence of previously converged landmark locations, resulting in a inaccurate map.

Thus, it is important for the iterated Kalman filter to have a more dependable covariance update rule. In this paper, an update rule based on singular value decomposition [21] is adapted to this application. To maintain the positive definite covariance matrix throughout the process, the covariance P_k^+ is maintained in the form of a singular value decomposition.

Suppose that P_k^+ is given as $P_{V\,k} P_{D\,k}^2 P_{V\,k}^T$. In the prediction step, a square root of the covariance matrix is obtained as follows:

$$\begin{aligned} P_{k+1}^- &= F_k P_k^+ F_k^T + W_k R W_k^T \\ &= F_k (P_{V\,k} P_{D\,k}^2 P_{V\,k}^T) F_k^T + W_k R W_k^T \\ &= \begin{bmatrix} P_{D\,k} P_{V\,k}^T F_k^T \\ \sqrt{R}^T W_k^T \end{bmatrix}^T \begin{bmatrix} P_{D\,k} P_{V\,k}^T F_k^T \\ \sqrt{R}^T W_k^T \end{bmatrix} \end{aligned}$$

A singular value decomposition is performed for the square root of P_{k+1}^-,

$$\text{SVD}\left(\sqrt{P_{k+1}^-}\right) = \text{SVD}\left(\begin{bmatrix} P_{D\,k} P_{V\,k}^T F_k^T \\ \sqrt{R}^T W_k^T \end{bmatrix}\right) = P_U \begin{bmatrix} P_D \\ \hline 0 \end{bmatrix} P_V^T \tag{15}$$

where P_U and P_V are orthonormal matrices and P_D is a diagonal matrix. Thus,

$$P_{k+1}^- = P_V^T \left[P_D^T \mid 0\right] P_U^T P_U \begin{bmatrix} P_D \\ \hline 0 \end{bmatrix} P_V^T = P_V P_D^2 P_V^T \tag{16}$$

Now in the iteration loop, the state estimate is updated by a variable step Gauss-Newton method using (7) as follows:

$$\begin{aligned} x_{i+1} = x_i + \gamma_i \quad & (H_i^T R^{-1} H_i + P_V P_D^{-2} P_V^T)^{-1} \\ & (H_i^T R^{-1} r + P_V P_D^{-2} P_V^T x_d) \end{aligned} \tag{17}$$

where $r = z - h(x_i)$, $x_d = x_0 - x_i$, and $x_0 = \hat{x}_{k+1}^-$. Here, we have that

$$\Delta x = x_{i+1} - x_i, \tag{18}$$

$$\nabla c = H_i^T R^{-1} r + P_V P_D^{-2} P_V^T x_d \tag{19}$$

After the iterations are complete (let's say at $i = N$), the measurement Jacobian H_N for the final iterated state x_N is obtained, then the covariance matrix P_{k+1}^+ is finally updated using (6) as follows:

$$\begin{aligned} P_{k+1}^+ &= P_{k+1}^- - P_{k+1}^- H_N^T (H_N P_{k+1}^- H_N^T + R)^{-1} H_N P_k^- && (20) \\ &= ((P_{k+1}^-)^{-1} + H_N^T R^{-1} H_N)^{-1} && (21) \\ &= (P_V {P_D}^{-2} {P_V}^T + H_N^T R^{-1} H_N)^{-1} \\ &= P_V \left({P_D}^{-2} + P_V^T H_N^T R^{-1} H_N P_V\right)^{-1} P_V^T && (22) \\ &= P_V \left(T^T T\right)^{-1} P_V^T \\ &= (P_V P_V^*) \, P_D^{*\,2} \, (P_V P_V^*)^T \\ &= P_{V\,k+1} P_{D\,k+1}^2 P_{V\,k+1}^T && (23) \end{aligned}$$

where $T = \begin{bmatrix} \sqrt{R^{-1}} H_N P_V \\ P_D^{-1} \end{bmatrix}$ and $P_U^* \begin{bmatrix} P_D^* \\ \hline 0 \end{bmatrix} {P_V^*}^T$ is the singular value decomposition of T. Thus, the final update is,

$$P_{V\,k+1} = P_V P_V^*, \quad P_{D\,k+1} = P_D^*.$$

Note that we choose the covariance update rule (21) which further becomes (22) over the numerically improper (20) to maintain the positive definiteness of the covariance. Its numerical positive definiteness is guaranteed by the term in middle of the quadratic equation, which is a sum of two positive definite matrices. Also, note that the covariance matrix is maintained in the singular value decomposition form.

To check the convergence of the iteration, we instead compute the cost function as,

$$c = x_d^T P_V P_D^{-2} P_V^T x_d + r^T R^{-1} r \tag{24}$$

Again, if the cost is not reduced after the iteration, the step-size variable γ_i is modified.

The algorithm for the proposed method, which summarizes the entire process, is shown in Algorithm 1.

6. Evaluation

In this section we discuss a simulation of a mobile robot that can create a map of a number of landmarks in a planar 2D world with bearing measurements. For each trial, the robot performs the proposed iterated bearing-only SLAM solution. The simulation is meaningful because it demonstrates how map accuracy is independent of the landmark initialization range. In addition, we discuss an experiment using a real robot with omnidirectional vision as well as an off-line experiment using our own experimental data.

6.1. Simulation Results

In our simulator, the mobile robot follows a unicycle model that has three degrees of freedom (the $x - y$ location of the vehicle and the vehicle heading). Two motion inputs (the translational velocity and rotational velocity) are used to evolve the state according to the process model. White Gaussian noise is added to each of the motion inputs but is hidden from the estimator for obvious reasons. In all cases, the only measurements used to update the Kalman filter are bearing measurements to landmarks, which are relative to the orientation of the robot and also include added white Gaussian noise. Table I shows the parameters that we used for each of the simulations.

Algorithm 1 Square root IKF

Require: $P_0 > 0, R > 0, \alpha$
  $P_{U_0} P_{D_0}^2 P_{V_0} = SVD(P_0)$
  loop
    /* Prediction */
    Update $x_0 = \hat{x}_{k+1}^-, F, W, P_{k+1}^-$ from (10), (11), (12), (15, 16)
    /* Measurement update*/
    Initialize all seen landmarks, x_1 and P_1 from (13)
    while $i <$ maximum iteration *and* not converged **do**
      /* IKF update */
      $x_s = x_i$
      $x_d = x_0 - x_i$
      Update $H, r = z - h(x_i), i$
      Compute the cost, c_{before} from (24)
      Update State from (17) and compute Δx from (18)
      Update $H, r = z - h(x_i)$
      Compute its gradient ∇c from (19)
      Compute the cost, c_{after} from (24)
      if $c_{before} + \gamma \nabla c^T \Delta x < c_{after}$ **then**
        $t = \beta t$
        $\gamma = \alpha t$ (choose $0 < \alpha < 0.5$)
        $x_i = x_s$
      else
        $t = 1, \gamma = 1$
        if $\|x_i - x_s\| < \epsilon$ **then**
          converged = true
        end if
      end if
    end while
    Update P_{k+1}^+ from (23)
  end loop

TABLE I
SIMULATION PARAMETERS

Measurement variance	σ_z^2	7.6x10^{-5} rad^2
Velocity variance	σ_v^2	10^{-4} m^2/s^2
Rotation variance	σ_ϕ^2	10^{-5} rad^2/s^2
Initialization variance	α	10^{10} m^2
Robot velocity	v	2.0 m/s
Robot rotational velocity	ω	0.314 rad/s

For the following discussion we will refer to the simulation result shown in Fig 2. The robot observes landmarks in the environment while performing a predefined circular trajectory. As stated before, only one measurement does not provide the information needed to initialize an estimate of the range to the observed landmark. Therefore, the algorithm must initialize each landmark with a predetermined range "guess". This can be chosen arbitrarily or may be chosen as a function of the sensor range. For this specific simulation shown in Fig. 2, the landmarks were all initialized at a distance of $5m$ when they were first observed. The uncertainty ellipses in Fig. 2 show that, as time elapses, the landmark estimates converge to the correct locations without divergence. With the same parameters, a simple EKF solution fails (diverges).

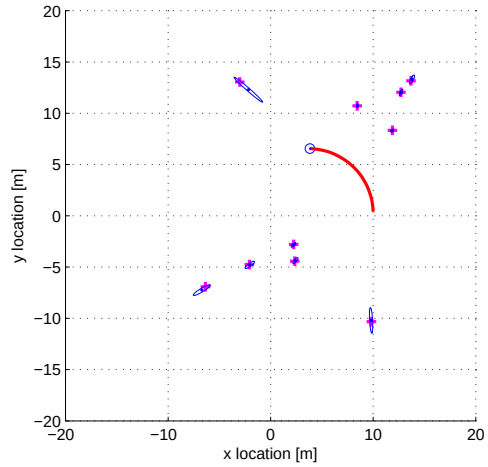

Fig. 2. This is a snapshot of a simulation: a mobile robot performs a circular trajectory while observing landmarks. The IKF is updated with bearing measurements and properly converges to the correct landmark locations.

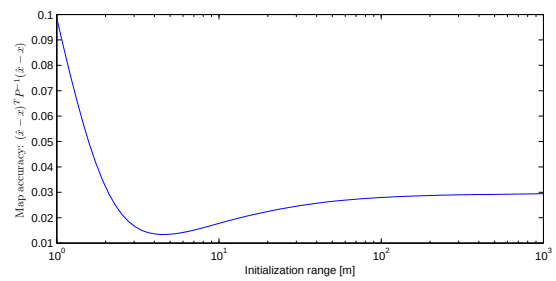

Fig. 3. This plot shows the map accuracy, over many trials, for the simulator in Fig. 2 versus the range value that was used when initializing each landmark.

We ran the same simulation again for many trials while varying the initialization range "guess" to test the sensitivity of the filter to this parameter. The accuracy of the map that is produced for each initialization scenario is plotted in Fig. 3. The graph is relatively flat as shown in Fig. 3, which demonstrates how the map accuracy is, for the most part, almost independent of landmark initialization. On the other hand, we note that, choosing an initialization distance that is too small may negatively impact the performance of the estimator. We attribute this to overconfidence in the landmark locations due to linearization.

The number of iterations is small enough as shown in Fig. 4, especially after the state estimation value converges. In this figure, the number of iterations include the backtracking steps. The real cost of the iteration steps comes from computing the inverse of the covariance matrix which is outside of the backtracking loop. We claim that the proposed IKF is not computationally exhaustive.

6.2. *Indoor Experiment with Visual Features*

An omnidirectional camera is used with our experimental platform for obtaining bearing measurements to landmarks in

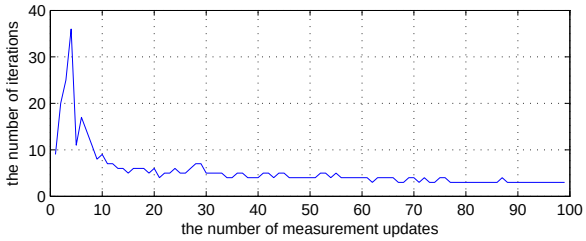

Fig. 4. The number of iterations per update significantly decreases as the state estimation converges.

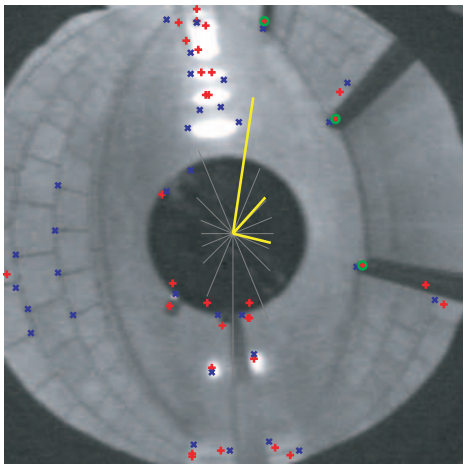

Fig. 5. A snap shot of the omnidirectional camera used to obtain bearing measurements to visual features in the environment.

the environment. The Scale Invariant Feature Transform, SIFT (see [22] for detail) is used for detecting natural landmarks such as doorways in an indoor environment. Fig. 5 shows a captured image where SIFT features at doorways are detected as landmarks and corresponding bearing measurements are inferred. Data association can be solved using the visual information available in the SIFT descriptor corresponding to the observed landmark. Fig. 6 shows a result obtained from the proposed bearing-only SLAM method with error ellipses overlapped with a ground truth floor plan. The map is accurate and demonstrates algorithm success for a typical indoor experiment.

7. Conclusion

We have presented bearing only SLAM using a square root iterated Kalman filter. By presenting the IKF as a Gauss-Newton optimization method, we were able to motivate the use of iteration in solving a nonlinear estimation problem. Careful study of an analytical example of bearing-only SLAM proved that iteration was especially important for this type of estimation problem. Several modifications were required, though, such as a reformulation of the update step as a square root filter to avoid numerical instability, and the alteration of the Gauss-Newton algorithm to incorporate a variable-step gain factor. The effectiveness of this approach was demonstrated with multiple experiments. Despite a naive approach to initialization

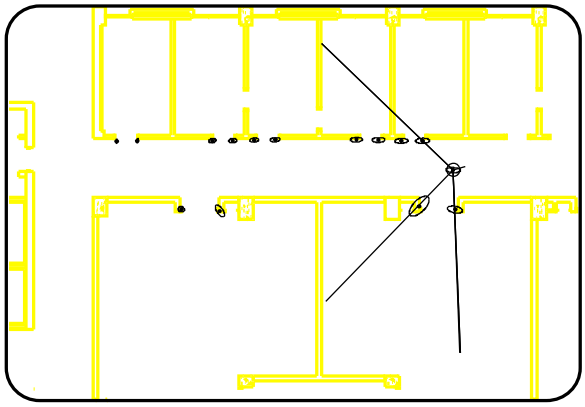

Fig. 6. The robot was configured to recognize doorway SIFT features. This figure displays, by means of comparison to a floorplan, the success of the IKF when estimating the doorway landmarks.

(which allows for a true undelayed algorithm), the filter properly converges upon landmark locations when creating a map. Our proposed solution takes the focus off of initialization when approaching bearing-only SLAM and makes up for the initial innaccuracy by using improved estimation techniques.

References

[1] H. Durran-Whyte and T. Bailey, "Simultaneous localization and mapping: Part i," *IEEE Robotics and Automaiton Magazine*, pp. 99–108, JUNE 2006.

[2] T. Bailey and H. Durran-Whyte, "Simultaneous localization and mapping: Part ii," *IEEE Robotics and Automaiton Magazine*, pp. 109–117, JUNE 2006.

[3] T. Bailey, "Constrained initialisation for bearing-only SLAM," in *Proceedings of the 2003 IEEE International Conference on Robotics and Automation*, September 2003.

[4] A. Costa, G. Kantor, and H. Choset, "Bearing-only landmark initialization with unknown data association," in *Proceedings of the 2004 IEEE International Conference on Robotics and Automation*, April 2004.

[5] N. M. Kwok, G. Dissanayake, and Q. P. Ha, "Bearing-only SLAM using a SPRT based gaussian sum filter," in *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, April 2005.

[6] J. Solà, A. Monin, M. Devy, and T. Lemaire, "Undelayed initialization in bearing only SLAM," in *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, April 2005.

[7] N. M. Kwok and A. B. Rad, "A modified particle filter for simultaneous localization and mapping," *Journal of Intelligent and Robotic Systems*, vol. 46, no. 4, pp. 365–382, 2006.

[8] K. E. Bekris, M. Glick, and L. Kavraki, "Evaluation of algorithms for bearing-only SLAM," in *Proceedings of the 2006 IEEE International Conference on Robotics and Automation*, May 2006.

[9] M. Deans and M. Hebert, "Experimental comparison of techniques for localization and mapping using a bearings only sensor," in *Proceedings of the 7th International Symposium on Experimental Robotics*, December 2000.

[10] S. Tully, H. Moon, G. Kantor, and H. Choset, "Iterated filters for bearing-only slam," in *IEEE International Conference on Robotics and Automation*, submitted.

[11] B. Bell and F. Cathey, "The iterated kalman filter update as a gauss-newton method," *IEEE Transactions on Automatic Control*, vol. 38, no. 2, pp. 294–297, 1993.

[12] L. Goncalves, E. D. Bernardo, D. Benson, M. Svedman, J. Ostrowski, N. Karlsson, and P. Pirjanian, "A visual front-end for simultaneous localization and mapping," in *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, April 2005.

[13] S. Se, D. Lowe, and J. Little, "Mobile robot localization and mapping with uncertainty using scale-invariant visual landmarks," *The International Journal of Robotics Research*, vol. 21, no. 8, pp. 735–758, 2002.

[14] H. W. Sorenson and D. L. Alspach, "Recursive bayesian estimation using gaussian sums," *Automatica*, vol. 7, pp. 465–479, 1971.

[15] D. L. Alspach and H. W. Sorenson, "Nonlinear bayesian estimation using gaussian sum approximations," *IEEE Transactions on Automatic Control*, vol. 17, no. 4, pp. 439–448, 1972.

[16] A. Davison, "Real time simultaneous localisation and mapping with a single camera," in *International Conference on Computer Vision*, July 2003.

[17] E. Eade and T. Drummond, "Scalable monocular slam," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, June 2006.

[18] M. Montemerlo, S. Thrun, D. Koller, and B. Wegbreit, "Fastslam 2.0: An improved particle filtering algorithm for simultaneous localization and mapping that provably converges," in *Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence (IJCAI)*, 2003.

[19] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical Recipes in C*, 2nd ed. Cambridge University Press, 1992.

[20] S. Boyd and L. Vandenberghe. Convex optimization. [Online]. Available: http://www.stanford.edu/~boyd/cvxbook/bv_cvxbook.pdf

[21] L. Wang, G. Libert, and P. Manneback, "Kalman filter algorithm based on singular value decomposition," in *Proceedings of the 31st Conference on Decision and Control*, December 1992.

[22] D. Lowe, "Distinctive image features from scale invariant features," *International Journal of Computer Vision*, vol. 60, pp. 91–110, 2004.