CSE 483: Mobile Robotics

Lecture by: Prof. K. Madhava Krishna
Scribe:Enna Sachdeva, Gourav Kumar

Module # 09
Date: 23rd September, 2016 (Friday)

Square-Root SAM(smoothing and mapping) - THEORY

This document discusses the theory of square root SAM along with the brief description of the associated topics. This is written as a subsequent module to EKF SLAM and and a brief idea of makes references and comparisons to the concept .

1 What is SAM

$SLAM$(Simultaneous localization and mapping), is the process in which a robot locates itself in an unknown environment while simultaneously mapping the environment. Landmark based SLAM (Landmarks' location can be uniquely identified by the sensors: laser range finders or cameras) methods are mostly based on the Extended Kalman filter(EKF), which is a linear approximation of a nonlinear system model, and recursively estimates the Gaussian density over the current pose of the robot. However, as the number of landmarks being observed by the robot increases, H matrix becomes dense, thereby increasing the computational complexity of EKF. Moreover, filtering itself has been reported to be inconsistent when applied to the inherently non-linear $SLAM$ problem. To address these issues, SAM based SLAM is used. SAM(smoothing and mapping) is a framework, wherein the SLAM problem is solved using a smoothing approach rather than the filtering one. This framework has been referred to as Square root SAM or $\sqrt{SAM}$, as it involves factorizing information matrix(I), or measurement Jacobian(A), into a square root form. The square root form of the matrix(I) results in a more stable and accurate algorithm.

1.1 EKF Based SLAM versus SAM based SLAM

Smoothing approach to SLAM involves not just the most current robot location but the entire robot trajectory up to the current time. Smoothing techniques possess following advantages over the filtering one:-

1. In smoothing, the covariance or information matrix(I) stays sparse, and the optimization problem can be solved using large least squares problem.

2. The sparse information matrix (I) or Jacobian (A) can be efficiently factorized using Cholesky or QR factorization, that speeds up the process to obtain the optimal trajectory and map.

3. It deals with the nonlinear process and measurement models in a much better way than EKF.

4. It is exact, rather than approximate as the Jacobians are calculated at the true state, unlike in EKF, where they are evaluated at estimated state.

2 Graphical models of SLAM

A graphical model is a prababilistic model, which encodes the conditional independencies between random variables.Graphs are an intuitive way of representing and visualising the relationships between variables(for example,we can answer questions like "*is A dependent on B given that we know the value of C?*", just by looking at the graph). Interpretation of an algorithm in terms of graphical models, yields good insight into the working of 'black box' algorithms such as QR or Cholesky factorization(which will be discussed later in the document).

2.1 SLAM as a Belief net

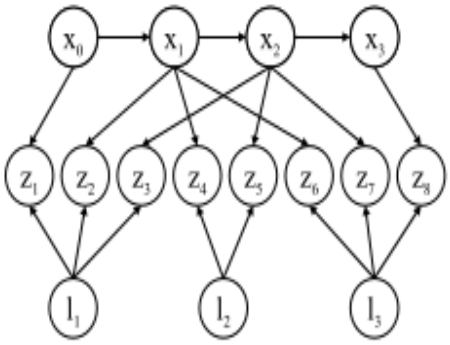

Figure 1:
Bayesian Belief Net for a SLAM problem.

A belief net is a directed acyclic graph(DAG) that encodes the conditional independent structure of a set of variables, where each variable only directly depends on its predecessors in the graph. The nodes of the graph represents variables and link represents the conditional dependence.

Consider a belief net model in fig. 1 with,
x_i: state of the robot at i^{th} time step, $i \in 0..M$
l_j: a landmark, $j \in 0...N$
z_k: a measurement, $k \in 0...K$
$P(x_0)$: a prior on the initial stage, assuming that it is given and is a constant.
$P(x_i|x_{i-1}, u_i)$: state transition probability, given a prior x_{i-1} and a control u_i. It specifies how environmental state evolves over time as a function of robot control u_i. Since, robot environments are stochastic, $P(x_i|x_{i-1}, u_i)$ is a probability distribution, not a deterministic function.
$P(z_k|x_{ik}, l_{jk})$: probability of measurement z_k from state x_i and a landmark l_j , based on the measurement model.It is assumed that the state x_i is sufficient to predict the (potentially noisy) measurement z_{ik}. Knowledge of any other variable, such as past measurements, controls or even past states, is irrelevant if x_i is known.

For a given known $Z = (z_1, z_2, z_3.....z_k)$,the joint probability corresponding to this network is given by

$P(X, L|Z) = P(X|L, Z)P(L|Z, X) = \frac{P((X,L)\cap Z)}{P(Z)} \quad \alpha \quad P(X, L, Z)$

$$\mathbf{P(X, L, Z) = P(x_0)} \prod_{\mathbf{i=1}}^{\mathbf{M}} \mathbf{P(x}_i|\mathbf{x}_{i\text{-}1}, \mathbf{u}_i) \prod_{\mathbf{k=1}}^{\mathbf{K}} \mathbf{P(z}_k|\mathbf{x}_{ik}, \mathbf{l}_{jk}) \tag{1}$$

NOTE: It is assumed that the data association problem has been solved, i.e each landmark has a unique id associated with.

Assuming Gaussian process, the motion model is defined as,

$$\mathbf{x}_i = \mathbf{f}_i(\mathbf{x}_{i\text{-}1}, \mathbf{u}_i) + \mathbf{w}_i \quad \Rightarrow \quad \mathbf{P}(\mathbf{x}_i|\mathbf{x}_{i\text{-}1}, \mathbf{u}_i) \ \alpha \ \exp -\frac{\mathbf{1}}{\mathbf{2}}||\mathbf{f}_i(\mathbf{x}_{i\text{-}1}, \mathbf{x}_i) - \mathbf{x}_i||^{\mathbf{2}}_{\mathbf{\Lambda_i}}, \tag{2}$$

where,
$\mathbf{f}_i(.)$ is a process model,
$\mathbf{w}_i(.)$ is a normally distributed process noise with a covariance matrix $\mathbf{\Lambda_i}$
The observation model is defined as,

$$\mathbf{z}_k = \mathbf{h}_k(\mathbf{x}_{ik}, \mathbf{l}_{jk}) + \mathbf{v}_k \quad \Rightarrow \quad \mathbf{P}(\mathbf{z}_k|\mathbf{x}_{ik}, \mathbf{l}_{jk}) \ \alpha \ \exp -\frac{\mathbf{1}}{\mathbf{2}}||\mathbf{h}_k(\mathbf{x}_{ik}, \mathbf{l}_{jk}) - \mathbf{z}_k||^{\mathbf{2}}_{\mathbf{\Sigma_k}}, \tag{3}$$

where,
$\mathbf{h}_k(.)$ is a measurement equation,
$\mathbf{v}_k(.)$ is a normally distributed zero mean measurement noise with a covariance matrix $\mathbf{\Sigma_k}$

2.2 SAM as a Least Squares Problem

Given all measurements ($\mathbf{Z} \triangleq \{\mathbf{z_k}\}$) and control inputs($\mathbf{U} \triangleq \{\mathbf{u_i}\}$), we would obtain an optimal estimate for the set of unknowns,i.e the entire trajectory($\mathbf{X} \triangleq \{\mathbf{x_i}\}$) and the map($\mathbf{L} \triangleq \{\mathbf{l_j}\}$), by finding the maximum a posteriori (MAP) estimate.
By maximizing the joint probability (equation(1)), the MAP can be obtained. Let Θ be the vector consisting of unknown variables, X and L, i.e $\Theta \triangleq (X, L)$
Our objective is to find Θ^*,
$\Theta^* \triangleq \underset{\Theta}{\text{argmax}} \ P(X, L|Z) = \underset{\Theta}{\text{argmax}} \ P(X, L, Z)$
Since logarithmic is a monotonic function,
$\implies \underset{\Theta}{\text{argmax}} \ f(\Theta) = \underset{\Theta}{\text{argmax}} \log f(\Theta) = \underset{\Theta}{\text{argmin}} \ -\log f(\Theta).$

Therefore, our objective function becomes,

$$\implies \mathbf{\Theta}^* \triangleq \underset{\Theta}{\text{argmin}} - \log P(X, L, Z) \tag{4}$$

$$\mathbf{\Theta}^* \triangleq \underset{\Theta}{\text{argmin}} - \log(P(x_0) \prod_{i=1}^{M} P(x_i|x_{i\text{-}1}, u_i) \prod_{k=1}^{K} P(z_k|x_{ik}, l_{jk})) \tag{5}$$

Using equations 2, 3 in equation 5, we get the following nonlinear least squares problem,

$$\mathbf{\Theta}^* \triangleq \underset{\Theta}{\text{argmin}} \left(\sum_{i=1}^{M} ||f_i(x_{i-1}, u_i) - x_i||^2_{\Lambda_i} + \sum_{k=1}^{K} ||h_k(x_{ik}, l_{jk}) - z_k||^2_{\Sigma_k} \right)$$

NOTE: The mahalanobis distance associated with motion model ($f(.)$) involves process noise covariance Λ_i and that associated with measurement model ($h(.)$) involves measurement noise covariance, Σ_k

3 Linearization of motion and observation models

For most of the practical purposes, the process model f_i or/and the measurement equation h_k is/are nonlinear. The models can be linearized about a point, whereas if if a good linearization point is not available, nonlinear optimization methods such as Gauss Newton or Levenberg-Marquart algorithms, which solve the succession of linear approximations, until convergence. Here, we will assume that either a good linearization point is available, or we are working on 1 iteration of a nonlinear optimization method.
Using taylor's series expansion, $f_i(x_{i-1}, u_i) - x_i$ can be linearized about x^0_{i-1} and x^0_i as below

$$f_i(x_{i-1}, u_i) - x_i \approx \left(f_i(x^0_{i-1}, u_i) + F_i^{i-1}\delta x_{i-1}\right) - \left(x^0_i - G^i_i \delta x_i\right)$$

where,F_i^{i-1} and G^i_i are the Jacobian of $f_i(.) - x_i$ at the linearization point x^0_{i-1} and x^0_i, respectively

$F_i^{i-1} \triangleq \frac{\partial(f_i(x_{i-1},ui)-x_i)}{\partial x_{i-1}}\Big|_{x^0_{i-1}}$, $G^i_i \triangleq \frac{\partial(f_i(x_{i-1},ui)-x_i)}{\partial x_i}\Big|_{x^0_i}$
and $a_i = x^0_i - f_i(x^0_{i-1}, u_i)$, is the odometry prediction error, which is a result of odometry perturbation($x^0_i = f_i(x^0_{i-1}, u_i) + w_i$).

Similarly,

$$h_k(x_{ik}, l_{jk}) - z_k \approx \left(h_k(x^0_{ik}, l^0_{jk}) + H^{ik}_k \delta x_{ik} + J^{jk}_k \delta l_{jk})\right) - z_k = (H^{ik}_k \delta x_{ik} + J^{jk}_k \delta l_{jk}) - c_k$$

where,H^{ik}_k and J^{ik}_k are the Jacobian of $h_k(.)$ with respect to x_{ik} and l_{jk} respectively, at the linearization point (x^0_{ik}, l^0_{jk})defined by

$H^{ik}_k \triangleq \frac{\partial h_k(x_{ik},ljk)}{\partial x_{ik}}\Big|_{(x^0_{ik},l^0_{jk})}$, and $J^{jk}_k \triangleq \frac{\partial h_k(x_{ik},ljk)}{\partial l_{jk}}\Big|_{(x^0_{ik},l^0_{jk})}$ and

$c_k = z_k - h_k(x^0_{ik}, l^0_{jk})$ is the measurement prediction error.

The objective function now becomes

$$\delta^* = \underset{\delta}{\operatorname{argmin}} \left(\sum_{i=1}^{M} ||F_i^{i-1}\delta x_{i-1} + G^i_i \delta x_i - a_i||^2_{\Lambda_i} + ||H^{ik}_k \delta x_{ik} + J^{ik}_k \delta l_{jk} - c_k||^2_{\Sigma_k} \right)$$

In the above equation, the Mahalanobis distance can be converted into a L-2 norm as follows:

$||e||^2_\Sigma \triangleq e^T\Sigma^{-1}e = (\Sigma^{-T/2}e)^T(\Sigma^{-T/2}e) = ||\Sigma^{-T/2}e||^2_2$,

Applying this converstion to equations, with Γ as a positive definite matrix having entries as $\Lambda_i^{-T/2}$ and $\Sigma_k^{-T/2}$, we can rewrite the objective function as

$$\delta^* = \underset{\delta}{\operatorname{argmin}} ||\Gamma^{-T/2}(A\delta - b)||_2^2$$

$$\delta^* = \underset{\delta}{\operatorname{argmin}} ||\Gamma^{-T/2}||_2^2 ||A\delta - b||_2^2$$

$$\delta^* = ||\Gamma^{-T/2}||_2^2 \underset{\delta}{\operatorname{argmin}} ||A\delta - b||_2^2$$

Since $||\Gamma^{-T/2}||_2^2$ is a constant, above equation is equivalent to

$$\delta^* = \underset{\delta}{\operatorname{argmin}} ||A\delta - b||_2^2$$

Hence, A matrix gets the following form.

$$A = \begin{bmatrix} G_1^1 & 0 & 0 & 0 & 0 \\ F_2^1 & G_2^2 & 0 & 0 & 0 \\ 0 & F_3^2 & G_3^3 & 0 & 0 \\ H_1^1 & 0 & 0 & J_1^1 & 0 \\ H_2^1 & 0 & 0 & 0 & J_2^2 \\ 0 & H_3^2 & 0 & J_3^1 & 0 \\ 0 & 0 & H_4^3 & 0 & J_4^2 \end{bmatrix}, b = \begin{bmatrix} a1 \\ a2 \\ a3 \\ c1 \\ c2 \\ c3 \\ c4 \end{bmatrix} \tag{6}$$

A can accomodate landmarks and measurements of different dimensions.

3.0.1 Dimensionality of A

$d_x \implies$ dimensions of state
$d_l \implies$ dimensions of landmarks
$d_z \implies$ dimensions of measurement
$M \implies$ number of states
$N \implies$ number of landmarks
$K \implies$ number of measurements obtained till current state x_n
then,
Dimensionality of A = $(Md_x + Kd_z).(Md_x + Nd_l)$

3.1 Formulating matrix A from the factor Graph

Factor graph is a biparte graph that represents the factorization of a function. A biparte graph is used to model relations between two different classes of objects, in our case these classes are states ($\mathbf{X} \triangleq \{\mathbf{x_i}\}$) and landmarks($\mathbf{L} \triangleq \{\mathbf{l_j}\}$). In a factor graph, there are nodes for the unknowns $(X, L)$and nodes, called *factor nodes*, for the probability factors(Z, U) defined on those variables, as shown in the figure 2.
The factor graphs at different time instants, as robot progresses in space, is shown in figure 3.

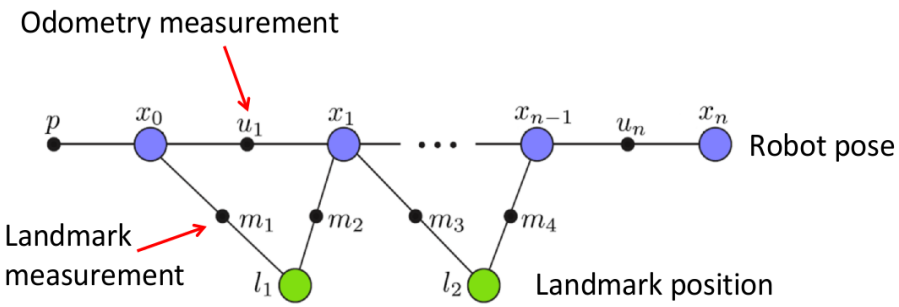

Figure 2:
Factor Graph representation for SLAM.

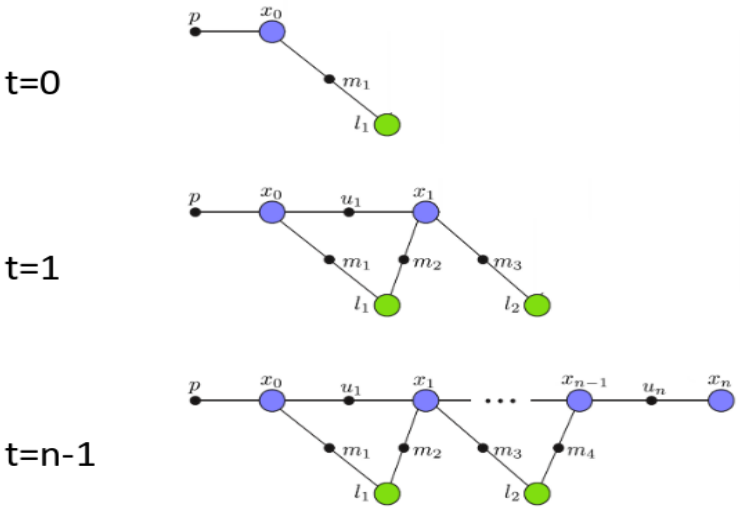

Figure 3:
Factor Graph at different time instants.

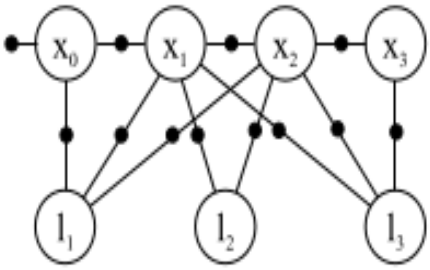

Figure 4:
Factor Graph.

Jacobian A is the measurement of the factor graphs associated with SLAM, hence can be used to construct A, directly. For instance, consider the factor graph in figure 4. to

construct its A matrix.

$$A = \begin{bmatrix} G_0^0 & 0 & 0 & 0 & 0 & 0 & 0 \\ F_1^0 & G_1^1 & 0 & 0 & 0 & 0 & 0 \\ 0 & F_2^1 & G_2^2 & 0 & 0 & 0 & 0 \\ 0 & 0 & F_3^2 & G_3^3 & 0 & 0 & 0 \\ H_1^0 & 0 & 0 & 0 & J_1^1 & 0 & 0 \\ 0 & H_2^1 & 0 & 0 & J_2^1 & 0 & 0 \\ 0 & H_3^1 & 0 & 0 & 0 & J_3^2 & 0 \\ 0 & H_4^1 & 0 & 0 & 0 & 0 & J_4^3 \\ 0 & 0 & H_5^2 & 0 & J_5^1 & 0 & 0 \\ 0 & 0 & H_6^2 & 0 & 0 & J_6^2 & 0 \\ 0 & 0 & H_7^2 & 0 & 0 & 0 & J_7^3 \\ 0 & 0 & 0 & H_8^3 & 0 & 0 & J_8^3 \end{bmatrix}, \delta = \begin{bmatrix} \delta_{x_0} \\ \delta_{x_1} \\ \delta_{x_2} \\ \delta_{x_3} \\ \delta_{l_1} \\ \delta_{l_2} \\ \delta_{l_3} \end{bmatrix}, b = \begin{bmatrix} a0 \\ a1 \\ a2 \\ a3 \\ c1 \\ c2 \\ c3 \\ c4 \\ c5 \\ c6 \\ c7 \end{bmatrix} \tag{7}$$

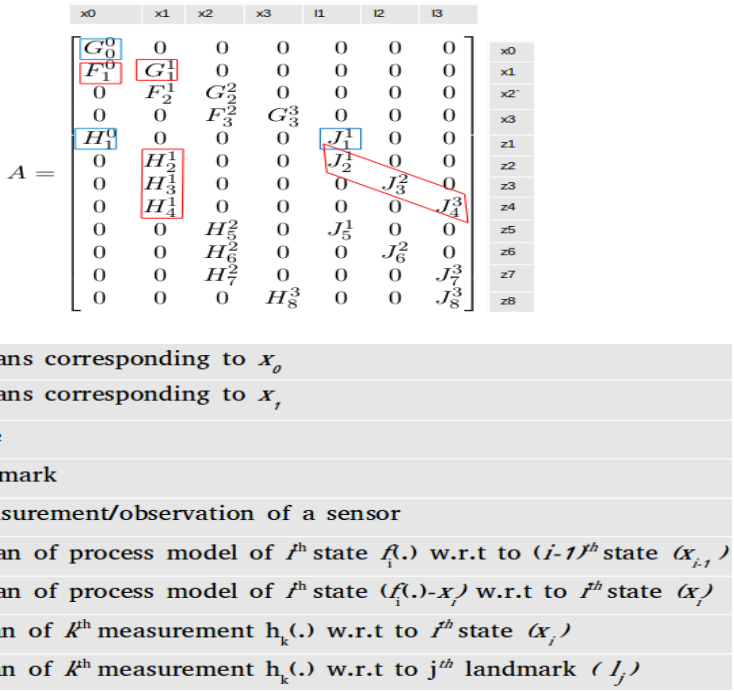

Figure 5:
A matrix

Here M=4, N=3, K=8 $\implies$ dimensions of $A = (4d_x + 8d_l) \times (3d_l + 4d_x)$. where

$d_x \implies$ dimensions of state
$d_l \implies$ dimensions of landmarks
$d_z \implies$ dimensions of measurement
$M \implies$ number of states
$N \implies$ number of landmarks
$K \implies$ number of measurements obtained till current state x_n
Every block of A corresponds to the least square criterion associated with either a landmark measurement or a odometry constraint.

3.2 Finding the solution of $A\delta = b$

For a full rank $m \times n$ matrix, the least square solution is found by solving $A^T A\delta^* = A^T b$, which is generally done by finding the Cholesky factorization of information matrix as, $I = A^T A = R^T R$, where,
R is the Cholesky traiangle $n \times n$ matrix, computed using cholesky factorization. Finding the δ^* using Cholesky factorization requires $(m + n/3)n^2$ flops.
When A matrix is ordered in canonical form(i.e, trajectory X first and then map L), the typical block structure is as below-

$$I = \begin{bmatrix} A_X^T A_X & I_X^L \\ I_X^{L^T} & A_L^T A_L \end{bmatrix} \tag{8}$$

Here, $I_X^L = A_X^T A_L$ encodes the correlation between robot state X and map L, and the diagonal blocks are band-diagonal.
An alternative approach to proceed the problem is QR factorization which has following advantages over cholesky factorization:

1. It is more accurate and numerically stable than cholesky factorization.
2. No need to compute I matrix, instead QR factorization of A if computed.

$A = QR$, where Q is an $m \times m$ orthogonal matrix and R is the upper triangular cholesky triangle. Note that A is a full column rank matrix($m > n$), $\implies Q^T Q = I$

$$Q^T A = \begin{bmatrix} R \\ 0 \end{bmatrix}, Q^T b = \begin{bmatrix} d \\ e \end{bmatrix} \tag{9}$$

Using orthogonality property of Q, the least square problem can be formulated as :
$||A\delta - b||_2^2 = ||Q^T A\delta - Q^T b||_2^2 = ||R\delta - d||_2^2 + ||e||_2^2$
Since,$||e||_2^2$ is the least-squares residual, the LS solution δ^* can be obtained by solving the square system $R\delta = d$. R is an upper triangular matrix, and the problem can be solved using back substitution, easily.

4 Square Root SAM

In this section we will give a brief description about variants of Square-root SAM.

4.1 Batch $\sqrt{\text{SAM}}$

The SAM problem in batch mode follows simple and straightforward method of solving least-squares problem with sparse matrices.The graph-theoretic algorithm underlying the solution of least-squares is *variable-elimination*, hence the order in which the variables are eliminated has huge impact on running time of matrix factorization algorithm (both *Cholesky* and *QR-decomposition*).As the Jacobian matrix A is nothing but adjacency matrix corresponding to the factor graph, it is interesting to note that column reordering heuristics also automatically exploits the locality inherent in geographic nature of slam problem i.e. the Jacobians corresponding to the factor nodes and hence landmarks/poses corresponding to those factor nodes lying close to each other after column reordering of the A matrix. The algorithm for the square-root SAM in batch mode is :-

Algorithm 1 Batch Square-root SAM

1: Build Measurement Jacobian matrix $\mathbf{A}$ and odometry/measurement prediction error vector $\mathbf{b}$ as explained in section 3.

2: Find optimal column odering of $\mathbf{A}(\mathrm{A}\xrightarrow{T}A_t)$.▷ T is the transformation corresponding to optimal column ordering

3: Solve for

$$\delta_{\mathbf{t}}^{*} = \underset{\delta}{\operatorname{argmin}}||A_t\delta_t - b||_2^2$$

using either Cholesky or QR factorization method.

4: Recover δ^* from δ_t^* by:

$$\delta_{\mathbf{t}}^{*} \xrightarrow{T^{-1}} \delta^{*}$$

▷ where T^{-1} is the inverse transformation

Although this algorithm assumes linear motion/observation models or models where a good linearization point exists for optimization, the same will work for nonlinear models where nonlinear optimizers(e.g. *Levenberg Marquardt* algorithm) are used. The only difference is that the algorithm(except the ordering steps) will be called multiple time until convergence by a non-linear optimizer.

4.2 Incremental $\sqrt{\mathbf{SAM}}$

As the batch mode computes the whole Jacobian matrix for every step for each update, it turns out to be computationally expensive hence an incremental version of above algorithm is derived for both linear as well as non linear cases

4.2.1 Linear Incremental $\sqrt{\mathbf{SAM}}$

If motion/observation models are linear or if a good linearization point is available we can update the factorizations incrementally by rank 1 Cholesky update which compute R' corresponding to I':

$$I' = I + aa^T \tag{10}$$

where I is the information matrix and $I{=}QR$ is its *QR-decomposition* and a^T is the new Jacobian row corresponding to the new measurement at that step.This algorithm of Cholesky update is typically used for dense matrices, but in our case we can go for more efficient method taking into account the sparse storage scheme that we use. One such method which is easy to implement is to use a series of Givens rotations to eliminate non-zeros in the new measurement rows one by one (for reference take a look at lecture9.pdf uploaded on course portal).

4.2.2 Non-linear Incremental $\sqrt{\mathbf{SAM}}$

In case the motion model or/and the observation model is/are non-linear functions and there is no method by which we can figure out a good linearization point, we have to resort to non-linear iterative algorithms such as Gauss-Newton or the Levenberg-Marquardt algorithm, which solve a succession of linear approximations to approach the minimum. The only drawback of these algorithms over linear ones is that they find linear approximations of the

models in each iteration before the convergence point is reached(and this is done for each update step).