## Q1

SAM as a Nonlinear Least-Squares Problem Formulation (i) (Equation 5)

A nonlinear least-squares problem called SAM (Smoothing and Mapping) is used to estimate both the environment map and a robot's trajectory at the same time. An example of a nonlinear least-squares problem is as follows:

min $x \sum i \| bi\ (x) - z\ i \|$
Two minutes x ∑ i // f i (x) −z i // 2

When SLAM is used, the robot's pose and landmark positions are represented by x, the anticipated measurement based on the robot's pose and landmarks is represented by y, and the actual measurements are represented by z i. Because of the generally nonlinear interaction between the poses, landmarks, and measurements, the problem is nonlinear.
Assumption -:

Gaussian Noise: The problem is made simpler by assuming that the measurement noise is Gaussian, which results in a quadratic form for the error terms.
Static Environment: For many applications, such indoor mapping, it makes sense to assume that the environment is static, meaning that landmarks remain stationary.
Since Gaussian noise is frequently assumed in sensor readings and many SLAM systems run in comparatively static surroundings, these assumptions are realistic.

## Q2) Finding the Linear Least-Square Solutions for Nonlinear Least-Squares (Eq. 9)

Using a technique similar to Gauss-Newton or Levenberg-Marquardt, we linearize the nonlinear least-squares issue around a first estimate in order to solve it. A linear least-squares issue of the following kind is produced by this linearization:

$A\ \delta\ x$ =
A´x=b

where dx is the update step for the state variables, b is the residual vector, and A is the Jacobian matrix of partial derivatives of the measurement functions with respect to the state variables.

Composition of Matrix A:

Dimensions: The number of measurements is represented by m, while the number of state variables is represented by n. These are the dimensions of UTMAZ.
Sparsity: $A$ A is sparse as, usually, each measurement depends only on a limited number of the state variables (e.g., the attitude of a robot at a specific moment). Utilising this sparsity allows for computational efficiency.

## Q3) Using QR Decomposition to Solve the Linear Least-Squares Problem (Eq. 13)

With QR decomposition, the linear least-squares problem Aδx=b can be solved:

$A = Q\ R$ A=QR

where R is an upper triangular matrix and Q is an orthogonal matrix. As a result, the issue becomes:

$R \delta x = Q T b$
R\x=Q T b

There are an endless number of possible solutions if A has linearly dependent columns, which occurs when A is not full column rank. Because it offers a least-squares solution, the QR technique is still effective; however, extra regularisation or constraints may be required to guarantee a unique answer.

## **Q4)** Applying EKF to Sequential SLAM Solving

With each new measurement, the Extended Kalman Filter (EKF) updates a state vector and its covariance matrix in order to solve the SLAM issue sequentially. The crucial actions are:

Prediction: Apply the motion model to estimate the new state.
Update: Adjust the state estimate with new measurements.
The way information is processed in EKF-SLAM varies from incremental square-root SAM.

EKF-SLAM: Updates the state successively; with time, linearization mistakes may occur, resulting in inconsistent results.
Square-root SAM: Uses sparsity in the problem to speed up computation while optimising the whole trajectory and map in a batch operation. Consistency is maintained.
These techniques take various approaches to the SLAM problem; EKF is better suited for real-time applications whereas square-root