then optimized using different numerical methods. Derivation with more details is provided in appendix A.

Using a Gaussian representation with the latent extension, the sensor model, the process model and measurement equation are defined as

$$\begin{aligned} \boldsymbol{x}_i &= f_i(\boldsymbol{x}_{i-1}, \boldsymbol{u}_i) + \boldsymbol{\eta}_i \\ \boldsymbol{z}_k &= h_k(\boldsymbol{x}_{i_k}, \boldsymbol{l}_{j_k}) + \boldsymbol{\theta}_k \end{aligned} \tag{3}$$

where $\boldsymbol{\eta}_i$ and $\boldsymbol{\theta}_k$ are noise terms which follow zero-mean Gaussian distribution with covariance matrices $\boldsymbol{\Gamma}_i$ and $\boldsymbol{\Sigma}_k$. With this formulation, the second part of the joint probability 1 is augmented with the mobility indicator. In addition we also apply a robust kernel v_k to the observation term, then the conditional probability of $\boldsymbol{z}_k$ is defined as an augmented Gaussian distribution:

$$\begin{aligned} P(\boldsymbol{z}_k|\boldsymbol{x}_{i_k}, \boldsymbol{l}_{j_k}, w_{j_k}) \propto \exp(-w_{j_k}\tilde{\boldsymbol{\mu}}_k^T \boldsymbol{\Sigma}_k^{-1} \tilde{\boldsymbol{\mu}}_k), \\ \tilde{\boldsymbol{\mu}}_k = v_k(h_k(\boldsymbol{x}_{i_k}, \boldsymbol{l}_{j_k}) - \boldsymbol{z}_k) \end{aligned} \tag{4}$$

where w_{j_k} represents the likelihood of being static for landmark $\boldsymbol{l}_{j_k}$ associated with measurement $\boldsymbol{z}_k$ at time k, and v_k is the robust scaling factor associated with each landmark observation representing whether the measurement is an inlier. When w_{j_k} or v_k approaches zero, the effect is equivalent to making the covariance of the Gaussian very large, effectively rendering the distribution uniform and the constraint represented by the distribution of no impact on the graph optimization process. Note that w_{j_k} can be negative because the formulation given here is proportional to a normalizing constant.

IV. ALGORITHMS

Following the notation in section III, we introduce the Expectation Maximization (EM) algorithms for estimating w with robustified objective function to learn the mobility of landmarks and estimate the robot trajectory and the map.

A. Estimation of Latent Parameters

As described in [11], in Equation 4, the hidden variables w_k must be estimated from multiple observations of each landmark. In the **M step**, we select the optimal $\boldsymbol{w}$ which maximizes the joint likelihood. However there is a trivial solution to the likelihood maximization which sets $w_k = 0$ for all k, we penalize the log-likelihood objective in Equation 5 with Lagrange multiplier which act as priors of the latent landmark mobility variables:

$$\begin{aligned} &\mathrm{Obj}(Z, X, L, W) = \\ &\quad \sum_k \left[-w_{j_k}(\tilde{\boldsymbol{\mu}}_k^T \boldsymbol{\Sigma}_k^{-1} \tilde{\boldsymbol{\mu}}_k)\right] - \frac{1}{2}\lambda(1-\boldsymbol{w})^T(1-\boldsymbol{w}) \end{aligned} \tag{5}$$

where $\tilde{\boldsymbol{\mu}}_k = v_k(h_k(\boldsymbol{x}_{i_k}, \boldsymbol{l}_{j_k}) - \boldsymbol{z}_k)$ is the robustified prediction error of observation k obtained from the E step, and w_{j_k} is the weight to estimate for landmark $l = j_k$ associated with the k-th observation. We equate the derivative of the objective function with respect to w_l to zero, and maximize the log likelihood, then for each w_l we get

$$w_l = 1 - \frac{1}{\lambda} \sum_{k \in K_l} \tilde{\boldsymbol{\mu}}_k^T \boldsymbol{\Sigma}_k^{-1} \tilde{\boldsymbol{\mu}}_k \tag{6}$$

where K_l is the set of measurements of landmark l, and λ is an assigned constant parameter to trade off the penalty. Larger λ will penalize more against the number of landmarks being mobile $w_k = 0$ for any k.

B. Graph SLAM Optimization

In the **E step**, we use the estimated latent parameters to obtain minimum variance results with graph optimization methods. We employ the standard approach of graph optimization as in [5], with our augmentation to the objective by introducing landmark mobility indicators w_{j_k} and observation robust kernel v_k. Specifically, we want to compute the maximum likelihood estimate of the robust objective

$$\begin{aligned} X^*, L^* &= \underset{X,L}{\arg\max}\{-\log p(X, L, U, Z, W)\} \\ &= \underset{X,L}{\arg\max}\{\sum_{i=1}^{M} ||f_i(\boldsymbol{x}_{i-1}, \boldsymbol{u}_i) - \boldsymbol{x}_i||^2_{\boldsymbol{\Gamma}_i} \\ &\quad + \sum_{k=1}^{K} w_{j_k} ||v_k(h_k(\boldsymbol{x}_{i_k}, \boldsymbol{l}_{j_k}) - \boldsymbol{z}_k)||^2_{\boldsymbol{\Sigma}_k} \\ &\quad + \sum_{k=1}^{K} ||1 - v_k||^2_{\boldsymbol{\Xi}_k}\} \end{aligned} \tag{7}$$

where the $||.||_{\boldsymbol{\Sigma}}$ term denotes the Mahalanobis distance with covariance $\boldsymbol{\Sigma}$. The penalty term $||1 - v_k||^2_{\boldsymbol{\Xi}_k}$ here is the switching prior introduced in [12]. Note that in this step, $\boldsymbol{w}$ is fixed and the penalty term with respect to w_k in the objective for estimating $\boldsymbol{w}$ becomes a constant, thus it does not appear in the objective function.

Here we apply the Dynamic Covariance Scaling[1] kernel v_k to each individual observation k. The factor v_k for observation k is given as:

$$v_k = \min\left\{1, \frac{2\Phi}{\Phi + w_{j_k}\boldsymbol{\mu}_k^T \boldsymbol{\Sigma}_k^{-1} \boldsymbol{\mu}_k}\right\} \tag{8}$$

where $\boldsymbol{\mu}_k = h_k(\boldsymbol{x}_{i_k}, \boldsymbol{l}_{j_k}) - \boldsymbol{z}_k$ is the original prediction error of observation k. The DCS kernel is shown to result in an upper bound specified by Φ for the second part of the objective function. By specifying the upper bound Φ for all robust edges, v dynamically scales the information matrices of all edges each iteration within the graph optimization process, and replaces the covariances of outliers with large ones making it near uniform. Φ is chosen as 1 in our implementation as suggested by its author.

With some derivation (Appendix section), the optimization problem can be formed as a standard least-squares problem:

$$\boldsymbol{\delta}^* = \underset{\boldsymbol{\delta}}{\arg\max}\, ||\boldsymbol{A}\boldsymbol{\delta} - \boldsymbol{b}||^2 \tag{9}$$

The algorithm, while reliable as shown in experiments in [3], is possible to have variants modified for real-time