Following the sparsification idea discussed in general terms in the previous section, we now replace $p(x_t \mid m^0, m^+, m^- = 0)$ by $p(x_t \mid m^+, m^- = 0)$ and thereby drop the dependence on m^0.

(12.24)
$$\begin{aligned}&\tilde{p}(x_t, m \mid z_{1:t}, u_{1:t}, c_{1:t})\\ &\quad = \; p(x_t \mid m^+, m^- = 0, z_{1:t}, u_{1:t}, c_{1:t})\; p(m^0, m^+, m^- \mid z_{1:t}, u_{1:t}, c_{1:t})\end{aligned}$$

This approximation is obviously equivalent to the following expression:

(12.25)
$$\begin{aligned}&\tilde{p}(x_t, m \mid z_{1:t}, u_{1:t}, c_{1:t})\\ &\quad = \; \frac{p(x_t, m^+ \mid m^- = 0, z_{1:t}, u_{1:t}, c_{1:t})}{p(m^+ \mid m^- = 0, z_{1:t}, u_{1:t}, c_{1:t})}\; p(m^0, m^+, m^- \mid z_{1:t}, u_{1:t}, c_{1:t})\end{aligned}$$

12.5.3 Mathematical Derivation of the Sparsification

In the remainder of this section, we show that the algorithm **SEIF_sparsification** in Table 12.4 implements this probabilistic calculation, and that it does so in constant time. We begin by calculating the information matrix for the distribution $p(x_t, m^0, m^+ \mid m^- = 0)$ of all variables but m^-, and conditioned on $m^- = 0$. This is obtained by extracting the sub-matrix of all state variables but m^-:

(12.26)
$$\Omega_t^0 \;=\; F_{x,m^+,m^0}\; F_{x,m^+,m^0}^T\; \Omega_t\; F_{x,m^+,m^0}\; F_{x,m^+,m^0}^T$$

With that, the matrix inversion lemma (Table 3.2 on page 50) leads to the following information matrices for the terms $p(x_t, m^+ \mid m^- = 0, z_{1:t}, u_{1:t}, c_{1:t})$ and $p(m^+ \mid m^- = 0, z_{1:t}, u_{1:t}, c_{1:t})$, denoted Ω_t^1 and Ω_t^2, respectively:

(12.27)
$$\Omega_t^1 \;=\; \Omega_t^0 - \Omega_t^0\, F_{m_0}\, (F_{m_0}^T\, \Omega_t^0\, F_{m_0})^{-1}\, F_{m_0}^T\, \Omega_t^0$$

(12.28)
$$\Omega_t^2 \;=\; \Omega_t^0 - \Omega_t^0\, F_{x,m_0}\, (F_{x,m_0}^T\, \Omega_t^0\, F_{x,m_0})^{-1}\, F_{x,m_0}^T\, \Omega_t^0$$

Here the various F-matrices are projection matrices that project the full state y_t into the appropriate sub-state containing only a subset of all variables—in analogy to the matrix F_x used in various previous algorithms. The final term in our approximation (12.25), $p(m^0, m^+, m^- \mid z_{1:t}, u_{1:t}, c_{1:t})$, possesses the following information matrix:

(12.29)
$$\Omega_t^3 \;=\; \Omega_t - \Omega_t F_x (F_x^T \Omega_t F_x)^{-1} F_x^T \Omega_t$$

Putting these expressions together according to Equation (12.25) yields the following information matrix, in which the feature m^0 is now indeed deactivated:

(12.30)
$$\tilde{\Omega}_t \;=\; \Omega_t^1 - \Omega_t^2 + \Omega_t^3$$

$$\begin{aligned} &= \Omega_t - \Omega_t^0\, F_{m_0}\, (F_{m_0}^T\, \Omega_t^0\, F_{m_0})^{-1}\, F_{m_0}^T\, \Omega_t^0 \\ &\quad + \Omega_t^0\, F_{x,m_0}\, (F_{x,m_0}^T\, \Omega_t^0\, F_{x,m_0})^{-1}\, F_{x,m_0}^T\, \Omega_t^0 \\ &\quad - \Omega_t\, F_x\, (F_x^T\, \Omega_t\, F_x)^{-1}\, F_x^T\, \Omega_t \end{aligned}$$

The resulting information vector is now obtained by the following simple consideration:

(12.31)
$$\begin{aligned} \tilde{\xi}_t &= \tilde{\Omega}_t\, \mu_t \\ &= (\Omega_t\ -\ \Omega_t\ +\ \tilde{\Omega}_t)\, \mu_t \\ &= \Omega_t\, \mu_t\ +\ (\tilde{\Omega}_t\ -\ \Omega_t)\, \mu_t \\ &= \xi_t\ +\ (\tilde{\Omega}_t\ -\ \Omega_t)\, \mu_t \end{aligned}$$

This completes the derivation of lines 3 to 5 in Table 12.4.

12.6 Amortized Approximate Map Recovery

The final update step in SEIFs is concerned with the computation of the mean μ. Throughout this section, we will drop the time index from our notation, since it plays no role in the techniques to be discussed. So we will write μ instead of μ_t.

Before deriving an algorithm for recovering the state estimate μ from the information form, let us briefly consider what parts of μ are needed in SEIFs, and when. SEIFs need the state estimate μ of the robot pose and the active features in the map. These estimates are needed at three different occasions:

1. The mean is used for the linearization of the motion model, which takes place in lines 3, 4, and 10 in Table 12.2.

2. It is also used for linearization of the measurement update, see lines 6, 8, 10, 13 in Table 12.3.

3. Finally, it is used in the sparsification step, specifically in line 4 in Table 12.4.

However, we never need the full vector μ. We only need an estimate of the robot pose, and an estimate of the locations of all active features. This is a small subset of all state variables in μ. Nevertheless, computing these estimates efficiently requires some additional mathematics, as the *exact* approach for recovering the mean via $\mu = \Omega^{-1}\, \xi$ requires matrix inversion or the use of some other optimization technique—even when recovering a subset of variables.

Once again, the key insight is derived from the sparseness of the matrix Ω. The sparseness enables us do define an iterative algorithm for recovering state variables online, as the data is being gathered and the estimates ξ and Ω are being constructed. To do so, it will prove convenient to reformulate $\mu = \Omega^{-1}\,\xi$ as an optimization problem. As we will show in just a minute, the state μ is the mode

$$\hat{\mu} \;=\; \underset{\mu}{\text{argmax}}\;p(\mu) \tag{12.32}$$

of the following Gaussian distribution, defined over the variable μ:

$$p(\mu) \;=\; \eta\;\exp\left\{-\tfrac{1}{2}\;\mu^T\;\Omega\;\mu+\xi^T\;\mu\right\} \tag{12.33}$$

Here μ is a vector of the same form and dimensionality as μ. To see that this is indeed the case, we note that the derivative of $p(\mu)$ vanishes at $\mu = \Omega^{-1}\,\xi$:

$$\frac{\partial p(\mu)}{\partial \mu} \;=\; \eta\,(-\Omega\;\mu+\xi)\;\exp\left\{-\tfrac{1}{2}\;\mu^T\;\Omega\;\mu+\xi^T\;\mu\right\} \;\stackrel{!}{=}\; 0 \tag{12.34}$$

which implies $\Omega\;\mu = \xi$ or, equivalently, $\mu = \Omega^{-1}\;\xi$.

This transformation suggests that recovering the state vector μ is equivalent to finding the mode of (12.33), which now has become an optimization problem. For this optimization problem, we will now describe an iterative hill climbing algorithm which, thanks to the sparseness of the information matrix.

COORDINATE DESCENT

Our approach is an instantiation of *coordinate descent*. For simplicity, we state it here for a single coordinate only; our implementation iterates a constant number K of such optimizations after each measurement update step. The mode $\hat{\mu}$ of (12.33) is attained at:

$$\begin{aligned}\hat{\mu} \;&=\; \underset{\mu}{\text{argmax}}\;\exp\left\{-\tfrac{1}{2}\;\mu^T\;\Omega\;\mu+\xi^T\;\mu\right\} \\ &=\; \underset{\mu}{\text{argmin}}\;\tfrac{1}{2}\;\mu^T\;\Omega\;\mu-\xi^T\;\mu\end{aligned} \tag{12.35}$$

We note that the argument of the min-operator in (12.35) can be written in a form that makes the individual coordinate variables μ_i (for the i-th coordinate of μ_t) explicit:

$$\tfrac{1}{2}\mu^T\;\Omega\;\mu-\xi^T\mu \;=\; \tfrac{1}{2}\sum_i\sum_j \mu_i^T\;\Omega_{i,j}\;\mu_j-\sum_i \xi_i^T\;\mu_i \tag{12.36}$$

where $\Omega_{i,j}$ is the element with coordinates (i,j) in the matrix Ω, and ξ_i if the i-th component of the vector ξ. Taking the derivative of this expression with