likelihood of the measurement computed in line 21 of the EKF localization algorithm for known data associations (Table7.2). Thus, we can incrementally calculate the mixture weights for each new component. The only downside of this algorithm is the fact that the number of mixture components, or *tracks*, grows exponentially over time.

The MHT algorithm approximates this algorithm by keeping the number of mixture components small. This process is called *pruning*. Pruning terminates every component whose relative mixture weight

PRUNING

$$\frac{\psi_{t,l}}{\sum_m \psi_{t,m}} \tag{7.27}$$

is smaller than a threshold $\psi_{\min}$. It is easy to see that the number of mixture components is always at most $\psi_{\min}^{-1}$. Thus, the MHT maintains a compact posterior that can be updated efficiently. It is approximate in that it maintains a very small number of Gaussians, but in practice the number of plausible robot locations is usually very small.

We omit a formal description of the MHT algorithm at this point, and instead refer the reader to a large number of related algorithms in this book. We note than when implementing an MHT, it is useful to devise strategies for identifying low-likelihood tracks before instantiating them.

7.7 UKF Localization

UKF localization is a feature-based robot localization algorithm using the unscented Kalman filter. As described in Chapter 3.4, the UKF uses the unscented transform to linearize the motion and measurement models. Instead of computing derivatives of these models, the unscented transform represents Gaussians by sigma points and passes these through the models. Table 7.4 summarizes the UKF algorithm for landmark based robot localization. It assumes that only one landmark detection is contained in the observation z_t and that the identity of the landmark is known.

7.7.1 Mathematical Derivation of UKF Localization

The main difference between the localization version and the general UKF given in Table 3.4 is in the handling of prediction and measurement noise. Recall that the UKF in Table 3.4 is based on the assumption that prediction and measurement noise are additive. This made it possible to consider the

Algorithm UKF_localization($\mu_{t-1}, \Sigma_{t-1}, u_t, z_t, m$):

Generate augmented mean and covariance

$$M_t = \begin{pmatrix} \alpha_1 v_t^2 + \alpha_2 \omega_t^2 & 0 \\ 0 & \alpha_3 v_t^2 + \alpha_4 \omega_t^2 \end{pmatrix}$$

$$Q_t = \begin{pmatrix} \sigma_r^2 & 0 \\ 0 & \sigma_\phi^2 \end{pmatrix}$$

$$\mu_{t-1}^a = (\mu_{t-1}^T \quad (0\ 0)^T \quad (0\ 0)^T)^T$$

$$\Sigma_{t-1}^a = \begin{pmatrix} \Sigma_{t-1} & 0 & 0 \\ 0 & M_t & 0 \\ 0 & 0 & Q_t \end{pmatrix}$$

Generate sigma points

$$\mathcal{X}_{t-1}^a = (\mu_{t-1}^a \quad \mu_{t-1}^a + \gamma\sqrt{\Sigma_{t-1}^a} \quad \mu_{t-1}^a - \gamma\sqrt{\Sigma_{t-1}^a})$$

Pass sigma points through motion model and compute Gaussian statistics

$$\bar{\mathcal{X}}_t^x = g(u_t + \mathcal{X}_t^u, \mathcal{X}_{t-1}^x)$$

$$\bar{\mu}_t = \sum_{i=0}^{2L} w_i^{(m)} \bar{\mathcal{X}}_{i,t}^x$$

$$\bar{\Sigma}_t = \sum_{i=0}^{2L} w_i^{(c)} (\bar{\mathcal{X}}_{i,t}^x - \bar{\mu}_t)(\bar{\mathcal{X}}_{i,t}^x - \bar{\mu}_t)^T$$

Predict observations at sigma points and compute Gaussian statistics

$$\bar{\mathcal{Z}}_t = h(\bar{\mathcal{X}}_t^x) + \mathcal{X}_t^z$$

$$\hat{z}_t = \sum_{i=0}^{2L} w_i^{(m)} \bar{\mathcal{Z}}_{i,t}$$

$$S_t = \sum_{i=0}^{2L} w_i^{(c)} (\bar{\mathcal{Z}}_{i,t} - \hat{z}_t)(\bar{\mathcal{Z}}_{i,t} - \hat{z}_t)^T$$

$$\Sigma_t^{x,z} = \sum_{i=0}^{2L} w_i^{(c)} (\bar{\mathcal{X}}_{i,t}^x - \bar{\mu}_t)(\bar{\mathcal{Z}}_{i,t} - \hat{z}_t)^T$$

Update mean and covariance

$$K_t = \Sigma_t^{x,z}\ S_t^{-1}$$

$$\mu_t = \bar{\mu}_t + K_t(z_t - \hat{z}_t)$$

$$\Sigma_t = \bar{\Sigma}_t - K_t\ S_t\ K_t^T$$

$$p_{z_t} = \det(2\pi S_t)^{-\frac{1}{2}} \exp\left\{-\tfrac{1}{2}(z_t - \hat{z}_t)^T S_t^{-1} (z_t - \hat{z}_t)\right\}$$

return μ_t, Σ_t, p_{z_t}

Table 7.4 The unscented Kalman filter (UKF) localization algorithm, formulated here for a feature-based map and a robot equipped with sensors for measuring range and bearing. This version handles single feature observations only and assumes knowledge of the exact correspondence. L is the dimensionality of the augmented state vector, given by the sum of state, control, and measurement dimensions.

noise terms by simply adding their covariances R_t and Q_t to the predicted state and measurement uncertainty, respectively (lines 5 and 9 in Table 3.4).

UKF_localization provides an alternative, more accurate approach to considering the impact of noise on the estimation process. The key "trick" is to augment the state with additional components representing control and measurement noise. The dimensionality L of the augmented state is given by the sum of the state, control, and measurement dimensions, which is $3+2+2=7$ in this case (the signature of feature measurements is ignored for simplicity). Since we assume zero-mean Gaussian noise, the mean μ^a_{t-1} of the augmented state estimate is given by the mean of the location estimate, μ_{t-1}, and zero vectors for the control and measurement noise (line 4). The covariance Σ^a_{t-1} of the augmented state estimate is given by combining the location covariance, Σ_{t-1}, the control noise covariance, M_t, and the measurement noise covariance, Q_t, as done in line 5.

The sigma point representation of the augmented state estimate is generated in line 6, using Equation (3.66) of the unscented transform. In this example, $\mathcal{X}^a_{t-1}$ contains $2L+1=15$ sigma points, each having components in state, control, and measurement space:

$$\mathcal{X}^a_{t-1} = \begin{pmatrix} \mathcal{X}^{x\,T}_{t-1} \\ \mathcal{X}^{u\,T}_{t} \\ \mathcal{X}^{z\,T}_{t} \end{pmatrix} \tag{7.28}$$

We choose mixed time indices to make clear that $\mathcal{X}^x_{t-1}$ refers to x_{t-1} and the control and measurement components refer to u_t and z_t, respectively.

The location components $\mathcal{X}^x_{t-1}$ of these sigma points are then passed through the velocity motion model g, defined in Equation (5.9). Line 7 performs this prediction step by applying the velocity motion model defined in Equation (5.13), using the control u_t with the added control noise component $\mathcal{X}^u_{i,t}$ of each sigma point:

$$\bar{\mathcal{X}}^x_{i,t} = \mathcal{X}^x_{i,t-1} + \begin{pmatrix} -\frac{v_{i,t}}{\omega_{i,t}}\sin\theta_{i,t-1} + \frac{v_{i,t}}{\omega_{i,t}}\sin(\theta_{i,t-1}+\omega_{i,t}\Delta t) \\ \frac{v_{i,t}}{\omega_{i,t}}\cos\theta_{i,t-1} - \frac{v_{i,t}}{\omega_{i,t}}\cos(\theta_{i,t-1}+\omega_{i,t}\Delta t) \\ \omega_{i,t}\Delta t \end{pmatrix} \tag{7.29}$$

where

$$v_{i,t} = v_t + \mathcal{X}^{u[v]}_{i,t} \tag{7.30}$$

$$\omega_{i,t} = \omega_t + \mathcal{X}^{u[\omega]}_{i,t} \tag{7.31}$$

$$\theta_{i,t-1} = \mathcal{X}^{x[\theta]}_{i,t-1} \tag{7.32}$$

are generated from the control $u_t = (v_t \ \omega_t)^T$ and the individual components of the sigma points. For example, $\mathcal{X}_{i,t}^{u[v]}$ represents the translational velocity v_t of the i-th sigma point. The predicted sigma points, $\bar{\mathcal{X}}_t^x$, are thus a set of robot locations, each resulting from a different combination of previous location and control.

Lines 8 and 9 compute the mean and covariance of the predicted robot location, using the unscented transform technique. Line 9 does not require the addition of a motion noise term, which was necessary in the algorithm described in Table 3.4. This is due to the state augmentation, which results in predicted sigma points that already incorporate the motion noise. This fact additionally makes the redrawing of sigma points from the predicted Gaussian obsolete (see line 6 in Table 3.4).

In line 10, the predicted sigma points are then used to generate measurement sigma points based on the measurement model defined in Equation (6.40) in Chapter 6.6:

$$\bar{\mathcal{Z}}_{i,t} = \begin{pmatrix} \sqrt{(m_x - \bar{\mathcal{X}}_{i,t}^{x[x]})^2 + (m_y - \bar{\mathcal{X}}_{i,t}^{x[y]})^2} \\ \operatorname{atan2}(m_y - \bar{\mathcal{X}}_{i,t}^{x[y]}, m_x - \bar{\mathcal{X}}_{i,t}^{x[y]}) - \bar{\mathcal{X}}_{i,t}^{x[\theta]}) \end{pmatrix} + \begin{pmatrix} \mathcal{X}_{i,t}^{z[r]} \\ \mathcal{X}_{i,t}^{z[\phi]} \end{pmatrix} \tag{7.33}$$

Observation noise is assumed to be additive in this case.

The remaining updated steps are identical to the general UKF algorithm stated in Table 3.4. Lines 11 and 12 compute the mean and covariance of the predicted measurement. The cross-covariance between robot location and observation is determined in line 13. Lines 14 through 16 update the location estimate. The likelihood of the measurement is computed from the innovation and the predicted measurement uncertainty, just like in the EKF localization algorithm given in Table 7.2.

7.7.2 Illustration

We now illustrate the UKF localization algorithm using the same examples as were used for the EKF localization algorithm. The reader is encouraged to compare the following figures to the ones shown in Chapter 7.4.4.

Prediction Step (Lines 2–9) Figure 7.12 illustrates the UKF prediction step for different motion noise parameters. The location components $\mathcal{X}_{t-1}^x$ of the sigma points generated from the previous belief are indicated by the cross marks located symmetrically around μ_{t-1}. The 15 sigma points have seven different robot locations, only five of which are visible in this x-y-projection.