**CSE 483: Mobile Robotics**

**Lecture by**: Prof. K. Madhava Krishna — **Lecture** # 04
**Scribe**: Lincy Pattanaik, Tanmay Chaudhari, Manan Bhandari — **Date**: 6th Sep, 2016

**EKF - Localization( Batch vs Incremental)**

# 1 The Extended Kalman Filter

The assumptions of linear state transitions and linear measurements made in Kalman filter rarely holds in practice. EKF overcomes this wherein the next state probability and measurement probabilities are governed by non-linear functions $g$ and $h$ respectively.
$\mathbf{x_{t+1}} = g(\mathbf{u_{t+1}}, \mathbf{x_t})$ and $\mathbf{z_{t+1}} = h(\mathbf{x_{t+1}})$
It then linearises about an estimate of the current mean and covariance.

# 2 EKF Localization

EKF localization is a special case of Markov localization Here the robots state is characterized by multivariate Gaussian distribution i.e Bel($\mathbf{x_t}$) used in Markov localization is represented by mean vector and a covariance matrix.

# 3 EKF Algorithm

```
Algorithm 1: Algorithm Extended Kalman Filter
   Input  : $(\mu_\mathbf{t}, \mathbf{\Sigma_t}, \mathbf{u_{t+1}}, \mathbf{z_{t+1}})$
   Output: $(\mu_{t+1}, \Sigma_{t+1})$
 1 $\hat{\mu_{\mathbf{t+1}}} = g(\mathbf{u_{t+1}}, \mu_\mathbf{t})$
 2 $\mathbf{\hat{\Sigma}_{t+1}} = \mathbf{F \Sigma_t F^T} + \mathbf{G \Sigma_{u,t+1} G^T}$
 3 $\mathbf{K} = \mathbf{\hat{\Sigma}_{t+1} H^T} (\mathbf{H \hat{\Sigma}_{t+1} H^T} + \mathbf{Q})^{-1}$
 4 $\mu_\mathbf{t+1} = \hat{\mu}_\mathbf{t+1} + \mathbf{K}(\mathbf{z_{t+1}} - h(\hat{\mu}_\mathbf{t+1}))$
 5 $\mathbf{\Sigma_{t+1}} = (\mathbf{I} - \mathbf{KH})\mathbf{\hat{\Sigma}_{t+1}}$
```

The Extended Kalman Filter can be implemented in two modes: **incremental**, where update takes place after each observation and **batch**, which is a one shot update taking all observations at same time. Both these algorithms differ only in updating step of EKF algorithm (steps 3, 4 and 5).

Consider a single time step with current state $\mu_\mathbf{t} = [\mu_{x,t}, \mu_{y,t}, \mu_{\theta,t}]^T$ and given control $\mathbf{u_{t+1}} = [T, \phi]^T$ with state noise $\mathbf{\Sigma_t}$ and control noise $\mathbf{\Sigma_{u,t+1}}$. After prediction (step 1 and 2) we obtain $\hat{\mu}_\mathbf{t+1}$ and $\mathbf{\hat{\Sigma}_{t+1}}$. Then sensors are fired and N measurements(each consisting of range $r$ and bearing $\psi$) are obtained. For updating we use one of the following algorithms.

### 3.1 Incremental mode

Here the associated updates are handled one-by-one. The measurement vector is of size 2 (consisting of a range and a bearing) and hence the measurement covariance matrix $\mathbf{Q_t}$ is of size 2x2.

**Algorithm 2:** Pseudo Code: Incremental mode

```
for i=1:N
    $\mathbf{\hat{z}_{2x1}} = h(\hat{\mu}_{\mathbf{t+1}})$
    $\mathbf{H_{2x3}} = \frac{\partial \mathbf{\hat{z}_{t+1}}}{\partial \hat{\mu}_{\mathbf{t+1}}}$
    Innovation Covariance: $\mathbf{S_{t+1_{2x2}}} = \mathbf{H}(\mathbf{\hat{\Sigma}_{t+1}})\mathbf{H^T} + \mathbf{Q}$
    $\mathbf{K_{3x2}} = (\mathbf{\hat{\Sigma}_{t+1}})\mathbf{H^T}\mathbf{S}^{-1}$
    $\mu_{\mathbf{t+1}} = (\hat{\mu}_{\mathbf{t+1}}) + \mathbf{K}(\mathbf{z} - \mathbf{\hat{z}})$
    $\mathbf{\Sigma_{t+1}} = (\mathbf{I} - \mathbf{KH})(\mathbf{\hat{\Sigma}_{t+1}})$
end for
```

### 3.2 Batch mode

The associated measurements are handled together within a joint matrix in the batch update process. The measurement vector $\mathbf{z}$ is of size 2Nx1 where the number of measurements is N. Similarly the measurement covariance matrix is of 2Nx2N.

**Algorithm 3:** Pseudo Code: Batch mode

```
for i=1:N
    $\mathbf{\hat{z}_{2Nx1}} = h(\hat{\mu}_{\mathbf{t+1}})$
    $\mathbf{H_{2Nx3}}(\mathbf{2i} - \mathbf{1} : \mathbf{2i}, :) = \frac{\partial \mathbf{\hat{z}_{i_{t+1}}}}{\partial \hat{\mu}_{\mathbf{t+1}}}$
    $\mathbf{QQ_{2Nx2N}}(\mathbf{2i} - \mathbf{1} : \mathbf{2i}, \mathbf{2i} - \mathbf{1} : \mathbf{2i}) = \mathbf{Q}$
    Innovation Covariance: $\mathbf{S_{t+1_{2Nx2N}}} = \mathbf{H}(\mathbf{\hat{\Sigma}_{t+1}})\mathbf{H^T} + \mathbf{QQ}$
    $\mathbf{K_{3x2N}} = (\mathbf{\hat{\Sigma}_{t+1}})\mathbf{H^T}\mathbf{S}^{-1}$
    $\mu_{\mathbf{t+1}} = (\hat{\mu}_{\mathbf{t+1}}) + \mathbf{K}(\mathbf{z} - \mathbf{\hat{z}})$
    $\mathbf{\Sigma_{t+1}} = (\mathbf{I} - \mathbf{KH})(\mathbf{\hat{\Sigma}_{t+1}})$
end for
```

## 4 Analysis between the two modes

The Kalman gain, $\mathbf{K}$, is calculated by using the inverse of the measurement uncertainty matrix, S. To calculate the inverse of $\mathbf{S}$, Cholesky decomposition can be used. The complexity of the inversion of a 2X2 matrix is $O(2^{2.4})$, therefore the computational complexity of the $\mathbf{S}$ inversion operation is $O(N * 2^{2.4})$in the incremental update for all observations, where N is the number of the associated measurements. The optimal complexity of the inverse calculation of a 2N X 2N matrix is $O((2N)^{2.4})$, hence the computational complexity of $\mathbf{S}$ inversion operation is $O(N^{2.4} * 2^{2.4})$ in the batch update for whole observations.