EL2320 - EKF

John Folkesson

Divergence Issues with Gaussian estimation, EKF, UKF...

VETENSKAP OCH KONST
KTH

Centre for Autonomous Systems

Complexity of the Kalman Filter

Typically the state consists of a dynamic part and a static part. Denote the dimensions of these by (n_d, n_s) with $n = n_d + n_s$ and assume we have arranged the state vector with the dynamic part first followed by the static part. For the predict:

$$\begin{aligned} \bar{\Sigma}_t &= Q_t + A_t \Sigma_{t-1} A_t^T && O(nn_d^2) \\ \bar{\mu}_{\mathbf{t}} &= A_t \mu_{t-1} + B_t \mathbf{u}_t && O(n_d^2) \end{aligned}$$

A will be a n x n matrix with the n_s rows and columns all zeros except for ones along the diagonal. And B will be zero for rows beyond n_d. If n_z is dimension of $\mathbf{z}$ and n_m are the number of states that have non-zero columns in C then the update:

$$\begin{aligned} K_t &= \bar{\Sigma}_t C_t^T (C_t \bar{\Sigma}_t C_t^T + R_t)^{-1} && O(nn_z n_m) \\ \mathbf{y}_t &= \mathbf{z}_t - \bar{\mathbf{z}}_t - C_t \bar{\mu}_t && O(n_z n_m) \\ \mu_t &= \bar{\mu}_{\mathbf{t}} + K_t \mathbf{y}_t && O(nn_z) \\ \Sigma_t &= \bar{\Sigma}_t - K_t C_t \bar{\Sigma}_t && O(n^2 n_z) \end{aligned}$$

So the overall leading term is $O(n^2 n_z + nn_d^2)$, often called $O(n^2)$.

Complexity of the Information Filter

For prediction we have a nasty inverse:

$$\bar{\Omega}_t = (Q_t + A_t \Omega_{t-1}^{-1} A_t^T)^{-1} = A^{-T}(\Omega_{t-1} A^{-1} Q_t A^{-T} + I)^{-1} \Omega_{t-1} A_t^{-1}$$
$$= A^T M^{-1} \Omega_{t-1} A^{-1} \qquad \text{where M =...}$$

Due to its sparseness the inverse can be computed in $O(nn_d^2)$ The various multiplications have leading term $O(n^2 n_d)$

$$\bar{\zeta}_t = \bar{\Omega}_t [A_t \Omega_{t-1}^{-1} \zeta_{\mathbf{t-1}} + B_t \mathbf{u}_t]$$
$$= M^{-1} \Omega_{t-1} A_t^{-1} A_t \Omega_{t-1}^{-1} \zeta_{\mathbf{t-1}} + \bar{\Omega}_t B_t \mathbf{u}_t$$
$$= M^{-1} \zeta_{\mathbf{t-1}} + \bar{\Omega}_t B_t \mathbf{u}_t$$

As most of this computation was done above there is no additional hard parts for predicting ζ. For the update:

$$\Omega_t = \bar{\Omega}_t + C_t^T R_t^{-1} C_t \qquad O(n_m^2 n_z)$$
$$\zeta_{\mathbf{t}} = \bar{\zeta}_{\mathbf{t}} + C_t^T R_t^{-1} \mathbf{y}_t. \qquad O(n_m n_z)$$

So the overall comlexity is $O(n^2 n_d)$, but it is hard to implement this without solving for the mean state each iteration:

$$\bar{\Omega}_t \bar{\mu}_{\mathbf{t}} = \bar{\zeta}_{\mathbf{t}} \qquad O(n^3) \text{ ouch!}$$

Unscented Kalman Filter (UKF)

The filter that doesn't stink.

This is an example of a wider class called Linear Regressive Kalman Filter LRKF.

The UKF idea similar to particle filters in that a set of points and weights are passed through the non-linear functions.

In the UKF the set of points and weights are chosen to give the 'correct' weighted mean and covariance.

Linear Regressive Kalman Filter LRKF

Kalman Filters for non-linear systems: a comparison of performance, Lefebvre2001.pdf

Gaussian Filters for Nonlinear Filtering Problems, ItoGFFNLP.pdf

Points, χ_t^i and weights w_i are chose so that:

$\mu_t = \sum_{i=1}^{M} w_i \chi_t^i$

$\Sigma_t = \sum_{i=1}^{M} w_i (\chi_t^i - \mu_t)(\chi_t^i - \mu_t)^T$

Here the χ_t^i are the so called regression points (vectors).

So this looks like a particle filter except the points are going to be chosen deterministically and few.

Linear Regressive Kalman Filter LRKF - Predict

$$\hat{\chi}_t^i = g(\chi_t^i, \mathbf{u}_t)$$

$$\bar{\mu}_t = \sum_{i=1}^M w_i \hat{\chi}_t^i$$

$$\bar{\Sigma}_t = \sum_{i=1}^M w_i (\hat{\chi}_t^i - \bar{\mu}_t)(\hat{\chi}_t^i - \bar{\mu}_t)^T + Q_t$$

Then new points and weights are chosen to give:

$$\bar{\mu}_t = \sum_{i=1}^M w_i \bar{\chi}_t^i$$

$$\bar{\Sigma}_t = \sum_{i=1}^M w_i (\bar{\chi}_t^i - \bar{\mu}_t)(\bar{\chi}_t^i - \bar{\mu}_t)^T$$

Linear Regressive Kalman Filter LRKF - Update

One then can replace the Kalman estimate of the predicted measurement error $S = (H_t \bar{\Sigma}_t H_t^T + R_t)$ with:

$\hat{\mathbf{z}}_\mathbf{t} = \sum_{i=1}^{M} w_i \mathbf{h}(\bar{\chi}_t^i)$

$S_t = \sum_{i=1}^{M} w_i (\mathbf{h}(\bar{\chi}_t^i) - \hat{\mathbf{z}}_\mathbf{t})(\mathbf{h}(\bar{\chi}_t^i) - \hat{\mathbf{z}}_\mathbf{t})^T + R_t$

And one then can replace the Kalman gain $\Sigma_t H_t^T S_t^{-1}$ with:

$K_t = \bar{\Sigma}_t^{x,z} S_t^{-1}$

$\bar{\Sigma}_t^{x,z} = \sum_{i=1}^{M} w_i (\bar{\chi}_t^i - \bar{\mu}_t)(\mathbf{h}(\bar{\chi}_t^i) - \hat{\mathbf{z}}_\mathbf{t})^T$

Choices for regression Points and Weights - UKF

The UKF makes a specific choice of 2n+1 regression points for an n dimensional state vector:

$$\{\chi\} = \{\mu, \mu \pm \sqrt{(n+\lambda)\Sigma}_i\}$$

Where $\Sigma = \sqrt{\Sigma}\sqrt{\Sigma}$. This can be computed as the squareroot of the eigen values D_i times the product of the eigen vectors $\mathbf{u}_i$:

$$\sqrt{\Sigma}_i = \mathbf{u}_i \sqrt{D_i} \mathbf{u}_i^T$$

The parameter $\lambda = \alpha^2(n+\kappa) - n$. These parameters (typically $1E-4 \leq \alpha \leq 1$ and $\kappa \geq 0$ is often 0) determine the spread of the regression points.

Choices for regression Points and Weights - UKF

The UKF differs slightly form the generic LRKF in that it uses a slightly different weight for the first regression point for the computations of the variances.

$w = \frac{\lambda}{n+\lambda}$

is used for the mean which then makes all the other weights equal to $w_i = \frac{1}{2(n+\lambda)}$ so that they add to 1.

For the covariance they add $(1 - \alpha^2 + \beta)$ to the first weight leaving the others unchanged. The parameter β encodes prior knowledge and 2 is optimal for true Gaussian distributions.