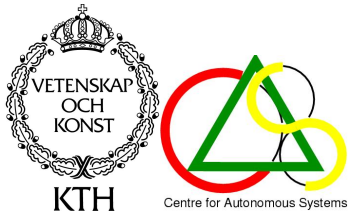
EL2320 - EKF

John Folkesson

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



Extended Kalman Filter - Matching

$$L_i = G(\mathbf{y}_{t,i}, 0, S_{t,i}), \quad S_{t,i} = H_{t,i}\bar{\Sigma}_t H_{t,i}^T + R_t, \quad m_i = \mathbf{y}_{t,i}^T S_{t,i}^{-1} \mathbf{y}_{t,i}$$

We have a robot in 2D, (x, y, θ), with a range sensor. The predict step produced this posteriori: $\bar{bel}(\mathbf{x}) = N(\bar{\mu}, \bar{\Sigma})$ where $\bar{\mu} = (0, 0, 0)^T$ and

$$\bar{\Sigma} = \begin{pmatrix} .25 & 0 & 0 \\ 0 & .25 & 0 \\ 0 & 0 & .01 \end{pmatrix}$$

There are known features at $xy = (1, 0)$ and $(2, 0)$

The range sensor gives us a value $r = 1.5$ with Gaussian noise with $R = .5$.

What is our measurement function $\mathbf{h}(\mathbf{x}|feature_i) =$?

What is the jacobian $H(\mathbf{x}|i) =$?

What are the mahalanobis distances?

What is the 'likelihood' of the range being from each feature?

$p(r|\mathbf{x} = \bar{\mu}, feature_i)$

Extended Kalman Filter - Matching

$$L_i = G(\mathbf{y}_{t,i}, 0, S_{t,i}), \quad S_{t,i} = H_{t,i}\bar{\Sigma}_t H_{t,i}^T + R_t, \quad m_i = \mathbf{y}_{t,i}^T S_{t,i}^{-1} \mathbf{y}_{t,i}$$

$h(\mathbf{x}|feature = 1) = \sqrt{(x-1)^2 + y^2}$
$h(\mathbf{x}|feature = 2) = \sqrt{(x-2)^2 + y^2}$
$H(\mathbf{x}|1) = \frac{(x-1,y,0)}{\sqrt{(x-1)^2+y^2}}$ and $H(\mathbf{x}|2) = \frac{(x-2,y,0)}{\sqrt{(x-2)^2+y^2}}$
$H(\mathbf{x}|i)|_{\mathbf{x}=\bar{\mu}} = (-1, 0, 0)$ and $(-1, 0, 0)$
$S_1 = S_2 = .25 + .5 = .75$
$m_1 = m_2 = .5^2/.75 = 1/3$
$L_1 = L_2 = \frac{1}{\sqrt{1.5\pi}} e^{-.5/3} = .39$

Optimal Estimators Cramer-Rao bound

For unbiased estimates, $\hat{x}$ of some scalar value x made by using measurements z with $p(z|x)$ given. The Cramer-Rao bound on the variance of $\hat{x}$ is:

$$E_z[(x - \hat{x})^2] \geq I(x)^{-1}$$

$$I(x) = E_z[(\frac{\partial}{\partial x}(ln(p(z|x)))^2]$$

$$= E_z[\frac{\partial^2}{\partial x^2}(-ln(p(z|x))]$$

$I(x)$ is called the Fischer Information.

Optimality implies that the variance equals the bound. We call the estimator a minimum mean square error MMSE or a minimum-variance unbiased estimator MVUE.

The regularity conditions for using this bound are:

$\frac{\partial}{\partial x}(ln(p(z|x))$ exists and is finite and

$$\frac{\partial}{\partial x} \int \hat{x}(z)p(z|x)dz = \int \hat{x}(z)\frac{\partial}{\partial x}(p(z|x))dz$$

Optimal Estimators Cramer-Rao bound

There are results that prove Cramer-Rao lower bound propagates according to the EKF equations under certain conditions including linearizing around the true (ie unknown) state. (Taylor 1979)

As the true state is unknown we simply use our estimates μ_t and $\bar{\mu}_t$

We might also linearize around a point that gives:

$z_t = h_t(\tilde{\mu}_{\mathbf{z}})$

Extended Kalman Filter - Divergence

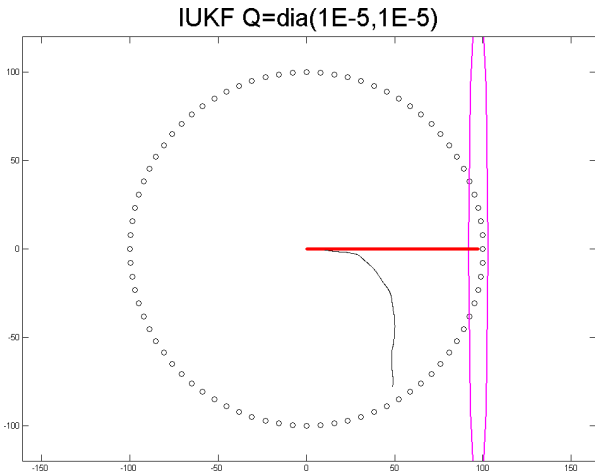
Divergence can be caused by:

- Numerical problems with the Matricies
- Model Errors
- Coorelations between Measurements ie. Bias
- Disturbances

Several methods might help:

- Inject noise: increase modeled uncertianty
- Do updates 'one row at a time' rather than all measurements at time t. This works if each row is independent of the others. The benefit is in computing the inverse in K_t. (This may however cause other types of divergence.)
- Change the update formula to:
$$\Sigma_t = \bar{\Sigma}_t - K_t(H_t\bar{\Sigma}_t H_t^T + R_t)K_t^T \text{ or}$$
$$\Sigma_t = (I - K_t H_t)\bar{\Sigma}_t(I - K_t H_t)^T + K_t R_t K_t^T$$
- Use the squareroot filter
- Abondon iterative/recursive estimates and use batch methods.

Extended Kalman Filter - Divergence



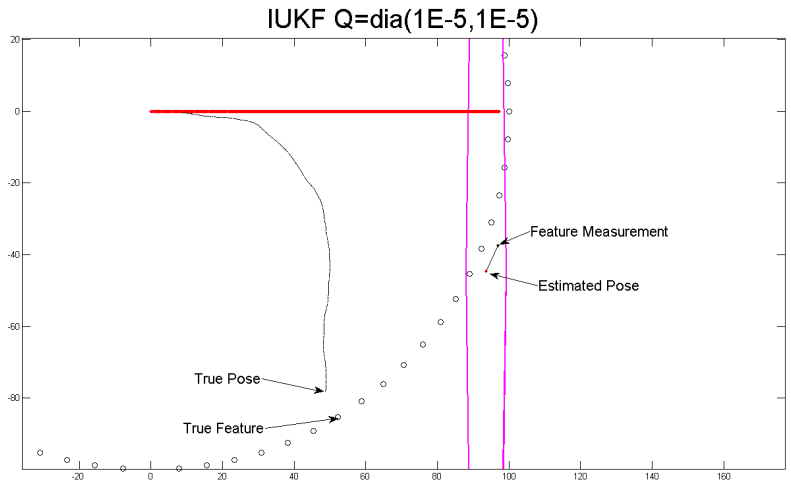


Figure : So here the linearization error has been reduces by using the UKF but a Gaussian is not able to consistently represent the posterior distribution. The updated pose is consistent with the inconsistent Gaussian.

Extended Kalman Filter - Divergence

A major problem is making relatively accurate measurements when the state is relatively uncertain and poorly estimated.

So EKF update is a sort of weighted average between the previous estimate and an estimate implied by linearizing $h(x)$.
$\mu_t = (I - W)\bar{\mu}_{\mathbf{t}} + W\mu_{\mathbf{z}}$

Problem is:

$h(\mu_{\mathbf{z}}) \neq z_t$ due to linearization far from μ_z.

So the measurement does not really imply μ_z which is garbage. Now if W is 'big' that is R_t 'small' then we will move the state to this garbage point and think we know where we are.

Extended Kalman Filter - Divergence

$K_t = \bar{\Sigma}_t H_t^T (H_t \bar{\Sigma}_t H_t^T + R_t)^{-1}$ and $\Sigma_t = \bar{\Sigma}_t - K_t H_t \bar{\Sigma}_t$
$\mu_t = \bar{\mu}_\mathbf{t} + K_t \mathbf{y}_t$
$\mu_t = \bar{\mu}_\mathbf{t} + \Sigma_t H_t^T R_t^{-1} (z_t - h_t(\bar{\mu}_t))$
$\mu_t = \bar{\mu}_\mathbf{t} + \Sigma_t H_t^T R_t^{-1} (z_t - h_t(\bar{\mu}_t) + H_t \bar{\mu}_\mathbf{t} - H_t \bar{\mu}_\mathbf{t}))$
$\mu_t = (I - \Sigma_t H_t^T R_t^{-1} H_t) \bar{\mu}_\mathbf{t} + \Sigma_t H_t^T R_t^{-1} (z_t - h_t(\bar{\mu}_t) + H_t \bar{\mu}_\mathbf{t})$

$\mu_t = (I - W) \bar{\mu}_\mathbf{t} + W \mu_\mathbf{z}$
$W = \Sigma_t H_t^T R_t^{-1} H_t$
where $H_t \mu_\mathbf{z} = (z_t - h_t(\bar{\mu}_t) + H_t \bar{\mu}_\mathbf{t})$ if this can be solved, e.g.

$\mu_\mathbf{z} = H^T (H H^T)^{-1} [z_t - h_t(\bar{\mu}_t) + H_t \bar{\mu}_\mathbf{t}]$ if $H H^T$ is full rank. Even
$\mu_\mathbf{z} = W^{-1} \Sigma_t H_t^T R_t^{-1} (z_t - h_t(\bar{\mu}_t) + H_t \bar{\mu}_\mathbf{t})$ works if W is full rank.

Extended Kalman Filter - Divergence

Is $\Sigma_t H_t^T R_t^{-1} = K_t$?

$$[\bar{\Sigma}_t - \bar{\Sigma}_t H_t^T (H_t \bar{\Sigma}_t H_t^T + R_t)^{-1} H_t \bar{\Sigma}_t] H_t^T R_t^{-1} = \bar{\Sigma}_t H_t^T (H_t \bar{\Sigma}_t H_t^T + R_t)^{-1}$$

$$H_t^T R_t^{-1} - H_t^T (H_t \bar{\Sigma}_t H_t^T + R_t)^{-1} H_t \bar{\Sigma}_t H_t^T R_t^{-1} = H_t^T (H_t \bar{\Sigma}_t H_t^T + R_t)^{-1}$$

We are correct if

$$R_t^{-1} - (H_t \bar{\Sigma}_t H_t^T + R_t)^{-1} H_t \bar{\Sigma}_t H_t^T R_t^{-1} = (H_t \bar{\Sigma}_t H_t^T + R_t)^{-1}$$

$$(H_t \bar{\Sigma}_t H_t^T + R_t) R_t^{-1} - H_t \bar{\Sigma}_t H_t^T R_t^{-1} = I$$

Why yes it is.

Extended Kalman Filter - Divergence

$$\mu_t = (I - W)\bar{\mu}_{\mathbf{t}} + W\mu_{\mathbf{z}}$$
$$W = \Sigma_t H_t^T R_t^{-1} H_t$$
$$H_t\mu_{\mathbf{z}} = (z_t - h_t(\bar{\mu}_t) + H_t\bar{\mu}_{\mathbf{t}})$$

As the true state is unknown we might linearize around a point that gives:
$z_t = h_t(\tilde{\mu}_{\mathbf{z}})$
One solution replace μ_z in the above form of the EKF update with a state $\tilde{\mu}$ that minimizes:

$$|\mathbf{h}(\tilde{\mu}) - z_t|^2 \qquad \text{or even} \qquad (\mathbf{h}(\tilde{\mu}) - \mathbf{z}_t)^T R_t(\mathbf{h}(\tilde{\mu}) - \mathbf{z}_t)$$

But this would not have unique solutions in general and even if it did the solution might not be 'optimal' for the full non-linear problem.

Extended Kalman Filter - Detecting Divergence

It can be good to be able to automatically detect when the EKF solution starts to diverge. Besides testing the innovations process (which can be slow) we can simple check if:

$$\frac{(\mathbf{h}(\mu_t)-\mathbf{z}_t)^T R_t (\mathbf{h}(\mu_t)-\mathbf{z}_t)}{(\mathbf{h}(\bar{\mu}_t)-\mathbf{z}_t)^T R_t (\mathbf{h}(\bar{\mu}_t)-\mathbf{z}_t)} < 1$$

When this does not hold you have a problem as the updated state is explaining the measurement worse that the pre-updated state. One can also compare the Mahalonobis distance (defined soon) before and after the update but the test above focuses only on the mean and is not effected by inconsistent covariance estimates (which do effect the Mahalonobis distance).

Kalman Filter - Notations

Σ_t is often denoted P_t
Q_t and R_t switch places in the book.
C_t becomes H_t and A_t becomes G_t for non-linear systems (EKF)
t often becomes k (I have more or less ignored the continuous case).
Many use two subscripts to separate predict and update values:
$\bar{\mu}_t \leftrightarrow \hat{\mathbf{x}}_{\mathbf{k}|\mathbf{k-1}}$
$\mu_t \leftrightarrow \hat{\mathbf{x}}_{\mathbf{k},\mathbf{k}}$
$\bar{\Sigma} \leftrightarrow P_{k|k-1}$
$\Sigma \leftrightarrow P_{k|k}$
For me Q_t and R_t are functions of the state estimate μ_t. Others show the jacobian explictly so
$Q_t \leftrightarrow G_t Q_t G_t^T$.
$R_t \leftrightarrow D_t R_t D_t^T$.

Kalman Filter - Growing the State

I cheated you by showing how to grow the state in the information form where it is easy. You have no initial infomation so the new Ω rows are just zero.
But what about the Kalman Filter? How do we initialize the rows of a new covariance?
For a prediction the assumption was that the dynamics would determine all the new state rows relative to the old state and the grow/shrink predition formula just happens.
If we are not doing a predict but rather adding to our world model we need to be sure the covariance is not infinity.

Kalman Filter - Growing the State

Two ways to add to the state, Way One:
The update formula is

$$K_t = \bar{\Sigma}_t C_t^T (C_t \bar{\Sigma}_t C_t^T + R_t)^{-1} \text{ and } \Sigma_t = \bar{\Sigma}_t - K_t C_t \bar{\Sigma}_t$$

So we need to modify $\bar{\Sigma}_t$ before we update with a C_t that depends on the new parts of the state.
We can simply add a big number along the diagonal for new rows. The update can then be done and the covariance will sink to a value about equal to the measurement uncertainty as long as big is big enough. All the correlations will be about correct as well.

Draw back is that we inject a small amount of absolute information which can cause some inconsistency. The bigger the big number the smaller this effect but too big a number might cause numerical problems with the update.
Advantage is that it is easy and the consistancy issue is often neglibly small compared to other sources of inconsistancy.

Kalman Filter - Growing the State

Way Two:
Assume we have sufficient information to initialize the new state $\mathbf{x}_{new}$ from the measurement and the old state:

$$\mathbf{x}_{new} = \mathbf{f}(\mathbf{x}_{old}, \mathbf{y}) \approx \mu_{new} + \mathbf{f}_x(\mathbf{x}_{old} - \mu_{old}) + \mathbf{f}_y(\mathbf{y})$$

$$\mathbf{x}_{new} - \mu_{new} \approx \mathbf{f}_x(\mathbf{x}_{old} - \mu_{old}) + \mathbf{f}_y(\mathbf{y})$$

Where $\mathbf{f}_x$ is the Jacobian wrt x. Remembering that the $\mathbf{y}$ are $N(0, R)$.

$$E[(\mathbf{x}_{new} - \mu_{new})(\mathbf{x}_{new} - \mu_{new})^T] = \mathbf{f}_x \bar{\Sigma}_{old} \mathbf{f}_x^T + \mathbf{f}_y R \mathbf{f}_y^T$$

$$E[(\mathbf{x}_{old} - \mu_{old})(\mathbf{x}_{new} - \mu_{new})^T] = \bar{\Sigma}_{old} \mathbf{f}_x^T$$

These can be plugged into the new rows and columns for Σ. They will be exactly correct up to the linearization errors. We then do not do any update step for these measurements.

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:

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

A will be a $n \times n$ 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:

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

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:
$\bar{\Omega}_t = (Q_t + A_t \Omega_{t-1}^{-1} A_t^T)^{-1}$
Due to its sparseness the inverse can be computed in $O(nn_d^2)$.
The various multiplications have leading term $O(n^2 n_d)$.
Then the rest of the predict:
$\bar{\zeta}_t = \bar{\Omega}_t [A_t \Omega_{t-1}^{-1} \zeta_{\mathbf{t-1}} + B_t \mathbf{u}_t]$
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

$$\hat{\chi}_t^i = g(\chi_{t-1}^i, \mathbf{u}_t)$$

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

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

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.