ALTERNATIVE FOR KALMAN FILTER – TWO DIMENSION SELF-LEARNING FILTER WITH MEMORY, IMPLEMENTATION IN STATE UNIVERSITY IN CHEŁM.

A. Fellner[1), P. Trómiński[1), J. Zając[1), K. Banaszek[2)
[1) The State University in Chełm, Poland
[2) Polish Air Navigation Services Agency

We propose new solution for idea Prof. Vanicek and Prof. Inzinga. This filter relies basically on the information contained in measurements on the vehicle: position fixes, velocities and their error statistics. The basic idea behind this new navigation filter is twofold:

1. **A cluster of the observed position fixes contains true kinematic information about the vehicle in motion,**
2. **A motion model of the vehicle associated with the error statistics of the position fixes should be able to get, to a large extent, the information out of the measurements for use.**

We base the filter on an analogy. We consider the statistical confidence region of every position fix as "source" tending to "attract" the undetermined trajectory to pass through this region. With these position fixes and their error statistics, a virtual potential field is constructed in which an imaginary mass particle moves. To make the filter flexible and responsive to a changing navigation environment, we leave some parameters free and let the filter determine their values, using a sequence of observations and the criterion of least squares of the observation errors. We show that the trajectory of the imaginary particle can well represent the real track of the vehicle.

Filter function:

$$\Phi_{r^0} = \frac{1}{K}\exp\left[-\frac{1}{2}(\mathbf{r}-\mathbf{r^0})^{\mathrm{T}}\mathbf{C}^{-1}(\mathbf{r}-\mathbf{r^0})\right]$$

$r = \begin{bmatrix} x \\ y \end{bmatrix}$ **position vector in actual time "t"**

$r^0 = \begin{bmatrix} x^0 \\ y^0 \end{bmatrix}$ **position vector in time "t_0"**

$$K = (2\pi)^{\frac{3}{2}}(detC)^{\frac{1}{2}}$$

Where C is a matrix of covariance:

$$C = \begin{bmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{bmatrix} = \begin{bmatrix} D^2X & cov(X,Y) \\ cov(Y,X) & D^2Y \end{bmatrix}$$

The basis for estimation position is potential U_i:

$$U_i(t) = G(\mathbf{r}-\mathbf{r}^0)^T C_i^{-1}(\mathbf{r}-\mathbf{r}^0)e^{-\alpha(t-t_i)}$$

Next step is conversion U_i when we know "n" position before time "t":

$$U = \sum_{i=1}^{n} U_i = Ge^{-\alpha t}\sum_{i=1}^{n}(\mathbf{r}-\mathbf{r}^0)^T C_i^{-1}(\mathbf{r}-\mathbf{r}^0)e^{\alpha t_i}$$

$$U(t) = \sum_{i=1}^{n} U_i\,(t) = Ge^{-\alpha t}\sum_{i=1}^{n}\left[\frac{(x-x_{0i})^2}{\sigma_{11i}^2}+\frac{(y-y_{0i})^2}{\sigma_{22i}^2}\right]e^{\alpha t_i}$$

where $r = \begin{bmatrix} x \\ y \end{bmatrix}$ $r_{0i} = \begin{bmatrix} x_{0i} \\ y_{0i} \end{bmatrix}$

In next step we have:

$$\ddot{r}(t) = e^{-\alpha t}G(Ar-B); \quad t \geq t_n$$

where

$$A = 2\sum_{i=1}^{n} e^{\alpha t_i}\, C_i^{-1} \quad \text{matrix } (2 \text{ x } 2)$$

$$B = 2\sum_{i=1}^{n} e^{\alpha t_i}\, C_i^{-1} r_{0i} \quad vector$$

$$C^{-1} = \begin{bmatrix} \frac{1}{\sigma_{11}^2} & 0 \\ 0 & \frac{1}{\sigma_{22}^2} \end{bmatrix} = \begin{bmatrix} p_{xi} & 0 \\ 0 & p_{yi} \end{bmatrix}$$

$$\ddot{x}(t) = -G(A_x x - B_x)e^{-\alpha t}; \quad t \geq t_n$$

$$\ddot{y}(t) = -G(A_y y - B_y)e^{-\alpha t}; \quad t \geq t_n$$

$$Ar = (A_x x, A_y y) \; and \; B = (B_x, B_y)$$

$$A_x = 2\sum_{i=1}^{n} e^{\alpha(t_i-t_n)}p_{xi}$$

$$A_y = 2\sum_{i=1}^{n} e^{\alpha(t_i-t_n)}p_{yi}$$

$$B_x = 2\sum_{i=1}^{n} e^{\alpha(t_i-t_n)}p_{xi}\, x_{0i}$$

$$B_y = 2\sum_{i=1}^{n} e^{\alpha(t_i-t_n)}p_{yi}\, x_{0i}$$

Final solution is:

$$\begin{cases} x(t) = \dfrac{B_x}{A_x} + a_1 J_0\left(\dfrac{2}{\alpha} e^{-\frac{\alpha}{2}t}\sqrt{GA_x}\right) + a_2 N_0\left(\dfrac{2}{\alpha} e^{-\frac{\alpha}{2}t}\sqrt{GA_x}\right) \\ y(t) = \dfrac{B_y}{A_y} + b_1 J_0\left(\dfrac{2}{\alpha} e^{-\frac{\alpha}{2}t}\sqrt{GA_y}\right) + b_2 N_0\left(\dfrac{2}{\alpha} e^{-\frac{\alpha}{2}t}\sqrt{GA_y}\right) \end{cases}$$

$$a_1 = -\frac{\pi}{\sigma}\left[\left(x_n - \frac{B_x}{A_x}\right)\sqrt{GA_x} N_1\left(\frac{2}{\alpha}\sqrt{GA_x}\right) - \dot{x}_n N_0\left(\frac{2}{\alpha}\sqrt{GA_x}\right)\right]$$

$$a_2 = -\frac{\pi}{\sigma}\left[\left(x_n - \frac{B_x}{A_x}\right)\sqrt{GA_x} J_1\left(\frac{2}{\alpha}\sqrt{GA_x}\right) - \dot{x}_n J_0\left(\frac{2}{\alpha}\sqrt{GA_x}\right)\right]$$

$$b_1 = -\frac{\pi}{\sigma}\left[\left(y_n - \frac{B_y}{A_y}\right)\sqrt{GA_y} N_1\left(\frac{2}{\alpha}\sqrt{GA_y}\right) - \dot{y}_n N_0\left(\frac{2}{\alpha}\sqrt{GA_y}\right)\right]$$

$$b_2 = -\frac{\pi}{\sigma}\left[\left(y_n - \frac{B_y}{A_y}\right)\sqrt{GA_y} J_1\left(\frac{2}{\alpha}\sqrt{GA_y}\right) - \dot{y}_n J_0\left(\frac{2}{\alpha}\sqrt{GA_y}\right)\right]$$

Our purpose is best estimation α and G from this equation:

$$f(\alpha, G) = \sum_{i=1}^{n}\{[x(t_i) - x_{0i}]^2 + [y(t_i) - y_{0i}]^2\} = minimum$$

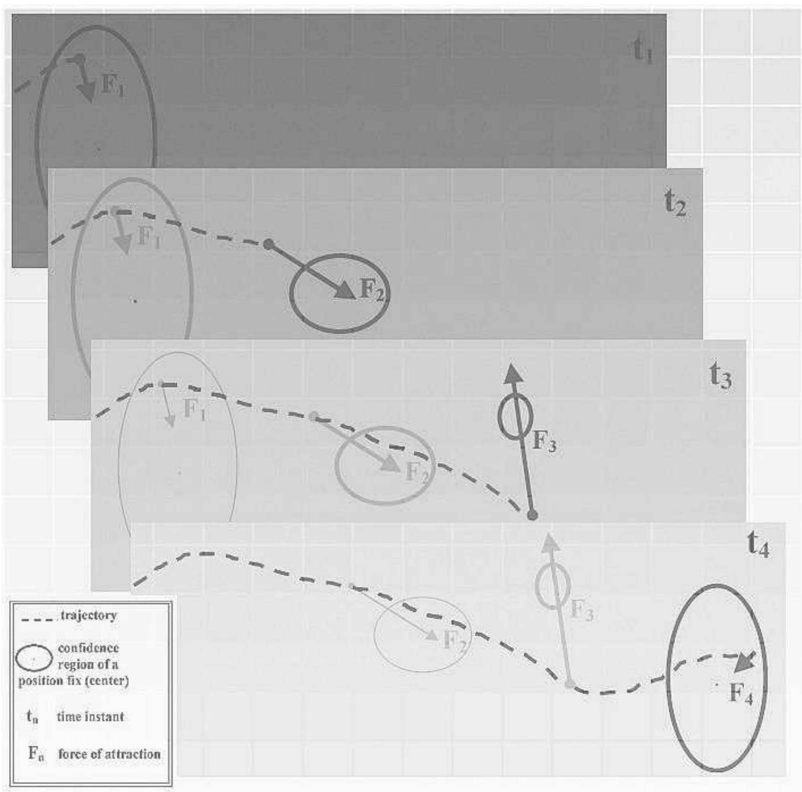

Fig. 1. Basic idea for New filter.

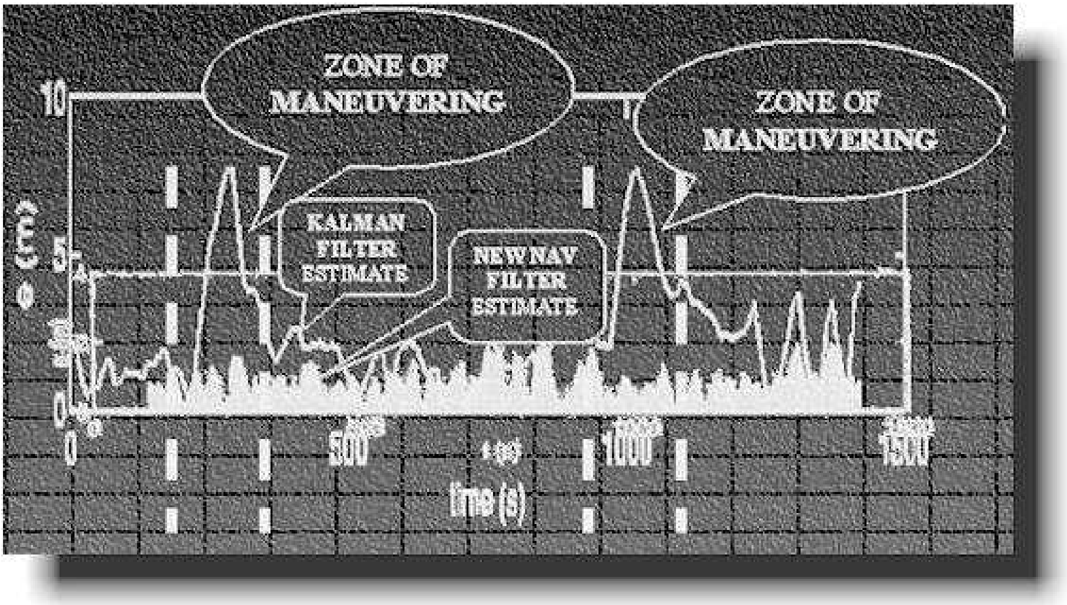

Fig. 2. Visualization estimation made witch Kalman filter and New filter.

BIBLIOGRAPHY:

1. **Benlin Xu. „A new navigation filter" 1996r.**
2. **Petr Vanìcek, Mensur Omerbašic (1999). „Does a navigation algorithm have to use Kalman filter?" Department of Geodesy and Geomatics Engineering University of New Brunswick.**
3. **Mathematical background Vanícek, P. and M. Omerbashich http://gge.unb.ca.**