Chair for Computer Aided Medical Procedures & Augmented Reality
Lehrstuhl für Informatikanwendungen in der Medizin & Augmented Reality

Sensor Fusion using the Kalman Filter

Overview

Goal of this project is to do a fusion of magnetic and optic sensor data via Extended and Federated Kalman Filters. The given data consists of positional data (x,y,z) and orientation data given as quaternions **r** =(r1,r2,r3,r4).

For more details on Quaternions and Kalman Filters we refer to the literature below.

Approach

We start with a simple approach using only positional data for tracking and regarding only one filter. Then step by step this sytem is extended. Rotation is included and the linear model is replaced by a general one. Next two filters are combined and finally the federated Kalman Filter is included.

Linear Kalman Filter for position tracking only

- State Vector: The state vector x is a 9x1 vector containing the positional data (x,y,z) with its first and second derivatives (velocity and acceleration)

$$\mathbf{x} = (x, y, z, \dot{x}, \dot{y}, \dot{z}, \ddot{x}, \ddot{y}, \ddot{z})^T$$

- Process Model: The process model relates the state at a previous time k-1 with the current state at time k. A is called state transition matrix and w represents the normal distributed process noise with covariance Q. The optinal control input Bu is discarded in the second equation.

$$\mathbf{x}_k = A\mathbf{x}_{k-1} + B\mathbf{u}_k + \mathbf{w}_{k-1}$$

$$\begin{pmatrix} x_k \\ y_k \\ z_k \\ \dot{x}_k \\ \dot{y}_k \\ \dot{z}_k \\ \ddot{x}_k \\ \ddot{y}_k \\ \ddot{z}_k \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & \Delta t & 0 & 0 & \frac{1}{2}(\Delta t)^2 & 0 & 0 \\ 0 & 1 & 0 & 0 & \Delta t & 0 & 0 & \frac{1}{2}(\Delta t)^2 & 0 \\ 0 & 0 & 1 & 0 & 0 & \Delta t & 0 & 0 & \frac{1}{2}(\Delta t)^2 \\ 0 & 0 & 0 & 1 & 0 & 0 & \Delta t & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & \Delta t & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & \Delta t \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_{k-1} \\ y_{k-1} \\ z_{k-1} \\ \dot{x}_{k-1} \\ \dot{y}_{k-1} \\ \dot{z}_{k-1} \\ \ddot{x}_{k-1} \\ \ddot{y}_{k-1} \\ \ddot{z}_{k-1} \end{pmatrix} + \mathbf{w}_{k-1}$$

- Measurement Model: The measurement model relates the current state to the measurement z with the matrix H. v is the normal distributed measurement noise with covariance R.

$$\mathbf{z}_k = H\mathbf{x}_k + \mathbf{v}_k$$

$$\begin{pmatrix} x_k \\ y_k \\ z_k \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} x_k \\ y_k \\ z_k \\ \dot{x}_k \\ \dot{y}_k \\ \dot{z}_k \\ \ddot{x}_k \\ \ddot{y}_k \\ \ddot{z}_k \end{pmatrix} + \mathbf{v}_k$$

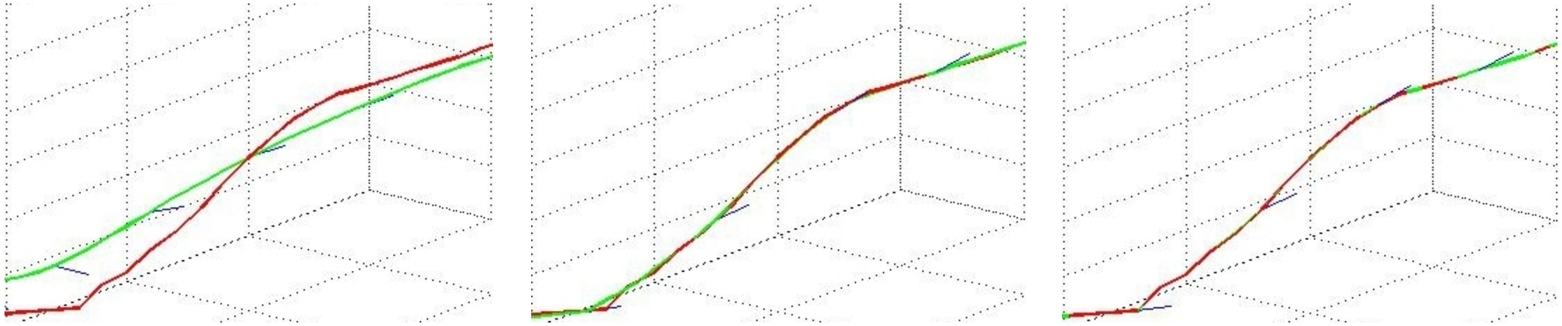
- Kalman Filter Cycle: The filter equations can be divided in a prediction and a correction step. The prediction step projects forward the current state and covariance to obtain an *a priori* estimate. After that the correction step uses the incorporates a new measurement to get an improved *a posteriori* estimate.

Time update ('Prediction')	Measurement update ('Correction')
(1) Project the state ahead: $\hat{x}_k^- = A\hat{x}_{k-1} + Bu_k$ (2) Project the covariance ahead: $P_k^- = AP_{k-1}A^T + Q$	(1) Compute the Kalman gain: $K_k = P_k^- H^T (HP_k^- H^T + R)^{-1}$ (2) Update state estimate: $\hat{x}_k = \hat{x}_k^- + K_k(z_k - H\hat{x}_k^-)$ (3) Update the covariance: $P_k = (I - K_k H)P_k^-$

- Influence of the covariance matrices Q and R
  - Q, the process noise covariance, contributes to the overall uncertainty. When Q is large, the Kalman Filter tracks large changes in the data more closely than for smaller Q.
  - R, the measurement noise covariance, determines how much information from the measurement is used. If R is high, the Kalman Filter considers the measurements as not very accurate. For smaller R it will follow the measurements more closely.

The three images below visualize the positional data. The red lines represent the measurement data, the green lines are the estimated states.

Q small, R large	**Q and R equal**	**Q large, R small**



Linear Kalman Filter for position and orientation tracking

Including orientation as euler angles allows us to keep the linear model from above. The angles can be derived from the quaternion 4-vector by the following formulas.

$$\psi = atan(2(q_y q_z + q_w q_x), q_w^2 - q_x^2 - q_y^2 + q_z^2)$$

$$\theta = asin(-2(q_x q_z - q_w q_y))$$

$$\phi = atan(2(q_x q_y + q_w q_z), q_w^2 + q_x^2 - q_y^2 - q_z^2)$$

- State Vector: The state vector x now is expanded. Rotation is added in form of three euler angles (roll, pitch, jaw) together with their first and second derivatives (angular velocity and acceleration).

  $$\mathbf{x} = (x, y, z, \dot{x}, \dot{y}, \dot{z}, \ddot{x}, \ddot{y}, \ddot{z}, \psi, \theta, \phi, \dot{\psi}, \dot{\theta}, \dot{\phi}, \ddot{\psi}, \ddot{\theta}, \ddot{\phi})^T$$

- Process Model: The process model stays linear and connects position and rotation information of the previous and the next state.

  $$\mathbf{x}_k = \begin{pmatrix} A_T & \mathbf{0} \\ \mathbf{0} & A_T \end{pmatrix} \mathbf{x}_{k-1} + \mathbf{w}_{k-1}$$

  $$A_T = \begin{pmatrix} 1 & 0 & 0 & \Delta t & 0 & 0 & \frac{1}{2}(\Delta t)^2 & 0 & 0 \\ 0 & 1 & 0 & 0 & \Delta t & 0 & 0 & \frac{1}{2}(\Delta t)^2 & 0 \\ 0 & 0 & 1 & 0 & 0 & \Delta t & 0 & 0 & \frac{1}{2}(\Delta t)^2 \\ 0 & 0 & 0 & 1 & 0 & 0 & \Delta t & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & \Delta t & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & \Delta t \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

- Measurement Model:

$$\begin{pmatrix} x_k \\ y_k \\ z_k \\ \psi_k \\ \theta_k \\ \phi_k \end{pmatrix} = \begin{pmatrix} I & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & I & \mathbf{0} & \mathbf{0} \end{pmatrix} \mathbf{x}_k + \mathbf{v}_k$$

Extended Kalman Filter for position and orientation tracking

As the rotational data is given in form of quaternions the upper representation is unfavourable. But updating a state vector which contains quaternions requires a non-linear model. Heren the extended Kalman Filter can be applied which linearizes about the current mean and covariance.

- State Vector: The rotation entries of the state vector are modified. x now contains the 9 elements for position, 4 elements of the quaternion and the 6 elements for angular velocity and acceleration as before.

$$\mathbf{x} = (x, y, z, \dot{x}, \dot{y}, \dot{z}, \ddot{x}, \ddot{y}, \ddot{z}, r_x, r_y, r_z, r_w, \omega_1, \omega_2, \omega_3, \dot{\omega_1}, \dot{\omega_2}, \dot{\omega_3})^T$$

- General Process and Measurement Models: The models for the extended Kalman filter are given by the functions f and h.

$$\mathbf{x}_k = f(\mathbf{x}_{k-1}, \mathbf{u}_{k-1}, \mathbf{w}_{k-1})$$

$$\mathbf{z}_k = h(\mathbf{x}_k, \mathbf{v}_k)$$

- Detailed Process and Measurement Models: The state vector is given as

$$\mathbf{x} = (\mathbf{p}^T, \dot{\mathbf{p}}^T, \ddot{\mathbf{p}}^T, \mathbf{r}^T, \omega^T, \dot{\omega}^T)^T$$

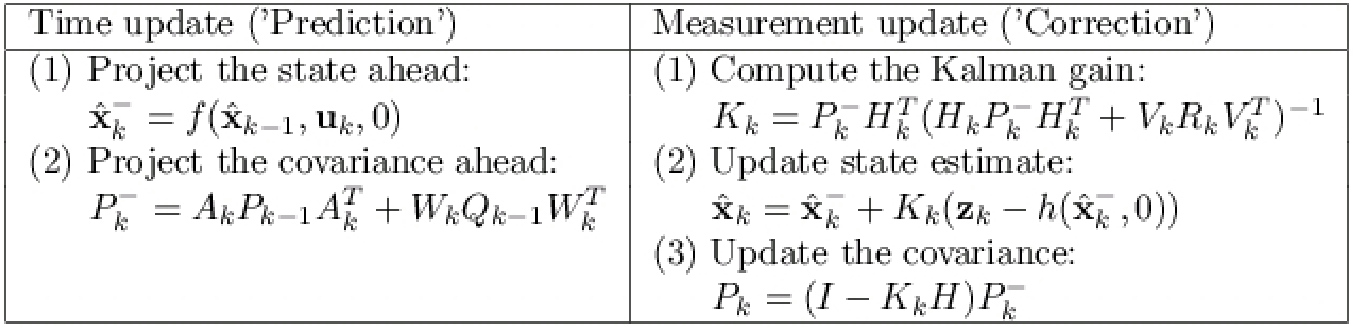
The function f handles the positional data in the linear way as described above. The quaternion **r** is updated through a quaternion multiplication with **d** which represents the difference caused by angular velocity. The angular velocity and acceleration can again be computed linearly.
The function h has to extract the position (x,y,z) and the quaternion data **r** from the state vector. For the position this can be done directly, the rotation quaternion needs an additional normalization.

$$\begin{pmatrix} \mathbf{p}_k \\ \dot{\mathbf{p}}_k \\ \ddot{\mathbf{p}}_k \end{pmatrix} = \begin{pmatrix} I & \Delta t I & \frac{1}{2}(\Delta t)^2 I \\ \mathbf{0} & I & \Delta t I \\ \mathbf{0} & \mathbf{0} & I \end{pmatrix} \begin{pmatrix} \mathbf{p}_{k-1} \\ \dot{\mathbf{p}}_{k-1} \\ \ddot{\mathbf{p}}_{k-1} \end{pmatrix}$$

$$\mathbf{r}_k = \mathbf{r}_{k-1} \otimes d_{k-1} = \mathbf{r}_{k-1} \otimes e^{\Delta t \omega_{k-1} + \frac{1}{2}(\Delta t)^2 \dot{\omega}_{k-1}} = \mathbf{r}_{k-1} \otimes e^{\mathbf{v}} = (cos(\frac{1}{2}|\mathbf{v}|)\frac{\mathbf{v}}{|\mathbf{v}|}, cos(\frac{1}{2}|\mathbf{v}|))$$

$$z_k = h(\mathbf{x}) = \begin{pmatrix} \mathbf{p} \\ normalize(\mathbf{r}) \end{pmatrix} = \begin{pmatrix} \mathbf{p} \\ \frac{\mathbf{r}}{|\mathbf{r}|} \end{pmatrix}$$

- Kalman Filter Cycle: To take account of the non-linear models the equations for the filter cycle are slightly modified.



Time update ('Prediction')	Measurement update ('Correction')
(1) Project the state ahead:	(1) Compute the Kalman gain:
$\hat{\mathbf{x}}_k^- = f(\hat{\mathbf{x}}_{k-1}, \mathbf{u}_k, 0)$	$K_k = P_k^- H_k^T (H_k P_k^- H_k^T + V_k R_k V_k^T)^{-1}$
(2) Project the covariance ahead:	(2) Update state estimate:
$P_k^- = A_k P_{k-1} A_k^T + W_k Q_{k-1} W_k^T$	$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_k^- + K_k(\mathbf{z}_k - h(\hat{\mathbf{x}}_k^-, 0))$
	(3) Update the covariance:
	$P_k = (I - K_k H) P_k^-$

The matrices A, H, W, V are Jacobians with partial derivatives of the functions f and h.

$$A_k = \frac{\partial f}{\partial \mathbf{x}}(\hat{\mathbf{x}}_k, \mathbf{u}_k, 0)$$

$$W_k = \frac{\partial f}{\partial \mathbf{w}}(\hat{\mathbf{x}}_k, \mathbf{u}_k, 0)$$

$$H_k = \frac{\partial h}{\partial \mathbf{x}}(\hat{\mathbf{x}}_k^-, 0)$$

$$V_k = \frac{\partial h}{\partial \mathbf{v}}(\hat{\mathbf{x}}_k^-, 0)$$

Fusion of two 6DOF trackers using the Kalman Filter

- Required: HandEyeCalibration to align the two tracker coordinate systems

Fusion of two 6DOF trackers using the federated Kalman Filter

Literature

- R.E. Kalman, A New Approach to linear Filtering and Prediction Problems, 1960
- A. Gelb (editor), Applied Optimal Estimation
- G. Welch and G. Bishop. An Introduction to the Kalman Filter. Technical Report TR 95-041, University of North Carolina, Department of Computer Science, 1995.
- G. Welch: The Kalman Filter Page
- *Neal A Carlson*, Federated square root filter for decentralized parallel processors, IEEE Transactions on Aerospace and Electronic Systems, Vol. 26 (3) May 1990, pp. 517-525
- G. Welch: SCAAT: Incremental Tracking with Incomplete Information
- *David A. Forsyth, Jean Ponce*, Tracking with Linear Dynamic Models, Chapter 17 from "Computer Vision: A Modern Approach"
- D. Pustka: Handling Error in Ubiquitous Tracking Setups?, diploma thesis, TU Munich, 2004
- Kalman Filtering for Quaternions
- E. W. Weisstein: Quaternions, Mathworld

-- KatharinaPentenrieder - 08 Mar 2005