M11 / Bar-Shalom Fusion

Full Step-by-Step Equation Workflow

Purpose. This workflow describes the two-branch M11 fusion procedure: each branch performs its own local EKF update, the cross-covariance between the two local estimators is propagated and updated, and then a final fusion gain combines the two correlated local posterior estimates into one fused estimate.

1. Notation

Let branch a and branch b denote the two local filters. At time step k:

- x_a, P_a: posterior state and covariance of branch a
- x_b, P_b: posterior state and covariance of branch b
- P_{ab}: cross-covariance between the estimation errors of the two branches
- F_k: linearized state transition matrix
- Q: process noise covariance
- z_a, z_b: measurements used by branch a and branch b
- $h_a(\cdot), h_b(\cdot)$: measurement models
- H_a, H_b: measurement Jacobians
- R_a, R_b: measurement noise covariances

2. Initialization

Start both local filters from the same initial posterior estimate:

$$x_a^{(0)} = x_b^{(0)} = x_0, \tag{1}$$

$$P_a^{(0)} = P_b^{(0)} = P_0. \tag{2}$$

If both branches begin from the same initial estimate, a consistent initial cross-covariance is

$$P_{ab}^{(0)} = P_0. \tag{3}$$

3. Full Step-by-Step Equation Workflow

For each time step $k = 1, 2, \dots,$ perform the following sequence.

3.1. Step 1: Prediction of Both Local Filters

Predict branch a:

$$x_{pa} = f(x_a, u_k), \tag{4}$$
$$P_{pa} = F_k P_a F_k^\top + Q. \tag{5}$$

Predict branch b:

$$x_{pb} = f(x_b, u_k), \tag{6}$$
$$P_{pb} = F_k P_b F_k^\top + Q. \tag{7}$$

Propagate the cross-covariance:

$$P_{ab}^- = F_k P_{ab} F_k^\top + Q. \tag{8}$$

3.2. Step 2: Local Update of Branch a

Form the innovation for branch a:

$$r_a = z_a - h_a(x_{pa}), \tag{9}$$
$$S_a = H_a P_{pa} H_a^\top + R_a. \tag{10}$$

Compute the local Kalman gain:

$$K_a = P_{pa} H_a^\top S_a^{-1}. \tag{11}$$

Update the local posterior state and covariance:

$$x_a = x_{pa} + K_a r_a, \tag{12}$$
$$P_a = (I - K_a H_a) P_{pa} (I - K_a H_a)^\top + K_a R_a K_a^\top. \tag{13}$$

Update the cross-covariance after branch a has assimilated its measurement:

$$P_{ab}^{(a)} = (I - K_a H_a) P_{ab}^-. \tag{14}$$

3.3. Step 3: Local Update of Branch b

Form the innovation for branch b:

$$r_b = z_b - h_b(x_{pb}), \tag{15}$$
$$S_b = H_b P_{pb} H_b^\top + R_b. \tag{16}$$

Compute the local Kalman gain:

$$K_b = P_{pb} H_b^\top S_b^{-1}. \tag{17}$$

Update the local posterior state and covariance:

$$x_b = x_{pb} + K_b r_b, \tag{18}$$
$$P_b = (I - K_b H_b) P_{pb} (I - K_b H_b)^\top + K_b R_b K_b^\top. \tag{19}$$

Complete the cross-covariance update after branch b has assimilated its measurement:

$$P_{ab} = P_{ab}^{(a)} (I - K_b H_b)^\top. \tag{20}$$

3.4. Step 4: Compute the Fusion Quantity D

Define the intermediate matrix

$$D = P_a + P_b - P_{ab} - P_{ab}^\top. \tag{21}$$

This matrix represents the uncertainty associated with the difference between the two correlated local estimates.

3.5. Step 5: Compute the Fusion Gain

Compute the M11 fusion gain:

$$K = (P_a - P_{ab})D^{-1}. \tag{22}$$

This K is **not** a measurement-update Kalman gain. It is the optimal linear fusion gain that determines how far the fused estimate should move from x_a toward x_b while accounting for the cross-correlation between the two local estimates.

3.6. Step 6: Fuse the Two Local Posterior Estimates

Form the fused posterior state:

$$x_f = x_a + K(x_b - x_a). \tag{23}$$

Equivalently,

$$x_f = (I - K)x_a + Kx_b. \tag{24}$$

So the fused estimate is a weighted combination of the two correlated local posteriors.

3.7. Step 7: Compute the Fused Covariance

The fused covariance is

$$P_f = (I - K)P_a(I - K)^\top + KP_bK^\top \tag{25}$$

$$+ (I - K)P_{ab}K^\top + KP_{ab}^\top(I - K)^\top. \tag{26}$$

This expression contains contributions from

- the uncertainty of branch a,
- the uncertainty of branch b,
- and the cross-correlation between their estimation errors.

3.8. Step 8: Reset for the Next Time Step

Store the fused posterior result:

$$x_k^+ = x_f, \tag{27}$$

$$P_k^+ = P_f. \tag{28}$$

A common implementation strategy is to restart both branches at the next step from the fused posterior:

$$x_a \leftarrow x_f, \qquad P_a \leftarrow P_f, \tag{29}$$
$$x_b \leftarrow x_f, \qquad P_b \leftarrow P_f. \tag{30}$$

If both branches are restarted from the same fused posterior, the cross-covariance is often reinitialized consistently as

$$P_{ab} \leftarrow P_f. \tag{31}$$

4. Compact Algorithm Summary

1. Initialize both local filters and the cross-covariance.
2. Predict branch a, predict branch b, and propagate P_{ab}.
3. Update branch a with its own measurement using K_a.
4. Update branch b with its own measurement using K_b.
5. Update the cross-covariance through the two local updates.
6. Compute D and then the fusion gain K.
7. Fuse x_a and x_b to obtain x_f.
8. Compute the fused covariance P_f.
9. Use the fused posterior as the common starting point for the next time step.

5. Key Interpretation

The M11 workflow contains **three distinct gains**:

$$K_a \quad : \text{ local measurement-update gain for branch } a, \tag{32}$$
$$K_b \quad : \text{ local measurement-update gain for branch } b, \tag{33}$$
$$K \quad : \text{ final fusion gain between two correlated local estimates.} \tag{34}$$

The algorithm therefore consists of

$$\boxed{\text{local filtering + cross-covariance tracking + correlated fusion}}$$

not merely a simple weighted average of two local states.