Chapter 1

Derivation of the EKF and of the UKF from the affine LSQ estimator

In the case of non linear systems, $p(x_k|u^{k-1}, y^{k-1})$ and $p(x_k|u^{k-1}, y^k)$ are no longer Gaussian and several approaches can be used to approximate them, ranging from methods relying on a Gaussian fitting of these pdf, like the EKF and the UKF (often called *parametric methods*[1]), to others (often denoted as non parametric filters) like the Histogram filter (HF) or the Particle filter (PF). While the first approaches only provide an approximation of the expected value of x_k and of its covariance matrix, HF and PF determine a numerical approximation of the complete pdf. Notice that the knowledge of only the expected value and the covariance matrix can be misleading in some cases where, e.g. the pdf is bimodal (see Fig. 1.1). However, the parametric approximations are usually much more efficient from a computational point of view, and so it is of interest to derive them.

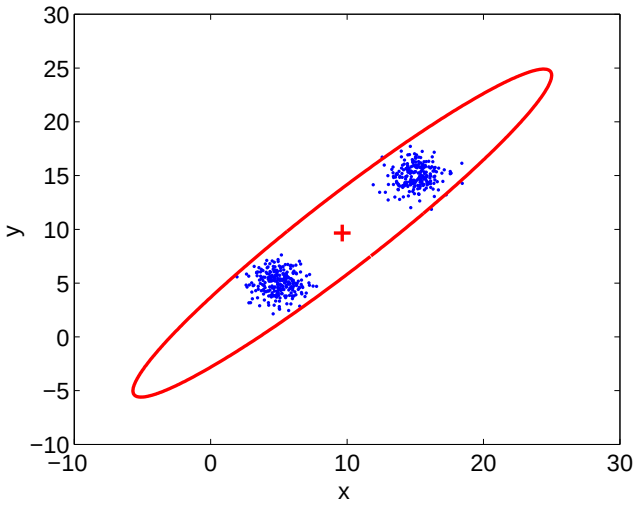

Figure 1.1: Approximating a generic pdf $p(x, y)$ with its mean and covariance is not always a good choice, especially if the pdf is not unimodal ('+' = mean of the pdf, red ellipse = covariance, blue dots = approximation of the pdf through a PF)

At this regard, let's give a look to the steps necessary to obtain the optimal estimator in the LSQ (Least SQuares) sense in this case. As we will see, we will simplify the problem by adopting in

[1]Actually, the EKF and the UKF do not make any explicit Gaussian assumption on the pdf but only offer a tool to compute the mean and the covariance of the state of a system along its evolution. For simplicity, since mean and covariance completely characterize a Gaussian pdf, we will say that these approaches perform a Gaussian fitting of the true pdf.

the correction step an estimation affine in the measurements (while the LSQ estimator is in general non linear). Under this restriction, it is possible to derive the recursive equations of the affine LSQ estimator (which is an estimator which manipulates linearly the measurements).

The practical implementation of these equations can be accomplished in different ways: (i) using a linearization of the non linearities, and we obtain the Extended Kalman Filter (EKF); (ii) using a different approximation called the Unscented Transformation, and we get the Unscented Kalman Filter (UKF).

In this section we will first derive the recursive equations of the affine LSQ estimator and then present its two approximate implementations: the EKF and the UKF[2].

1.1 The affine LSQ estimator for general systems

Consider a system

$$\begin{aligned} x_k &= f(x_{k-1}, u_{k-1}, w_{k-1}) \\ y_k &= h(x_k) + v_k \end{aligned}$$

where w_k and v_k are noise sequences, assumed w.l.o.g. 0 mean, and we have considered a simple dependence of the measurement on the noise v_k, being very common in practical settings. Let Q_k and R_k be the covariance matrices associated with w_k and v_k respectively and assume also independence of these sequences as in the linear case: i.e. we have i.i.d. sequences, independent of each other and independent of the initial state x_0, assumed a random variable with expected value m_0 and covariance matrix P_0 (the Gaussianity of the noises and of the initial state however is no longer required here).

Let $x \in \Re^n$ and $y \in \Re^q$. Then the affine LSQ estimator is the following recursive filter (initialized with $\hat{x}_0 = m_0$ and $P_0 = E[(x_0 - m_0)(x_0 - m_0)']$):

$$\begin{aligned} prediction: \quad \hat{x}_k^- &= E[f(x_{k-1}, u_{k-1}, w_{k-1})] && (1.1) \\ P_k^- &= E[(x_k - \hat{x}_k^-)(x_k - \hat{x}_k^-)'] && (1.2) \\ correction: \quad \hat{x}_k &= \hat{x}_k^- + \Sigma_{xy,k}\Sigma_{y,k}^{-1}(y_k - E[h(x_k)]) && (1.3) \\ P_k &= P_k^- - \Sigma_{xy,k}\Sigma_{y,k}^{-1}\Sigma_{yx,k} && (1.4) \end{aligned}$$

where $\Sigma_{y,k} = E[(y_k - E[y_k])(y_k - E[y_k])']$ and $\Sigma'_{yx,k} = \Sigma_{xy,k} = E[(x_k - E[x_k])(y_k - E[y_k])']$, and all the expectations are taken w.r.t. the random variables in the arguments, with $y_k = h(x_k) + v_k$ (see below for more details).

1.1.1 The prediction step

We now derive the equations (1.1)-(1.2) of the Prediction step. To have a LSQ estimator, we must take:

$$\hat{x}_k^- = \arg\min_{\xi} E[(\xi - x_k)'(\xi - x_k)] = \arg\min_{\xi} E[(\xi - f(x_{k-1}, u_{k-1}, w_{k-1}))'(\xi - f(x_{k-1}, u_{k-1}, w_{k-1}))],$$

where the expectation is taken w.r.t. the independent random variables x_{k-1} and w_{k-1} (i.e. according to the pdf $p(x_{k-1}|u^{k-2}, y^{k-1})$ of x_{k-1} and to $p(w_{k-1})$). We have already shown that the solution of the previous equation is $\xi = E[f(x_{k-1}, u_{k-1}, w_{k-1})]$. In fact, let for brevity $f = f(x_{k-1}, u_{k-1}, w_{k-1})$. Then, any other $\xi = E[f] + g$, where g is a deterministic vector (function of u^{k-1} and of y^{k-1}), would give

$$E[(\xi - f)'(\xi - f)] = E[(f - E[f])'(f - E[f])] + g'g$$

[2]It is worth mentioning that in the case of linear systems with *non* Gaussian noises, the Kalman Filter remains optimal in the LSQ sense among affine estimators: its equations in fact result from the application of the LSQ estimator presented in Section 1.1 to the linear case. This will be also mentioned at the beginning of Section 1.2.

which is minimized by $g = 0$ ($E[(f - E[f])'(f - E[f])]$ is a quantity which only depends on the process and can not be influenced by the choice of $\hat{x}$). So, we have:

$$\hat{x}_k^- = E[f(x_{k-1}, u_{k-1}, w_{k-1})]$$

The covariance matrix is then given by:

$$P_k^- = E[(x - \hat{x}_k^-)(x - \hat{x}_k^-)'] = E[(f - E[f])(f - E[f])'].$$

1.1.2 The correction step

Before deriving the equations of the correction step of the affine LSQ estimator, let's consider the case we have a random vector x, with pdf $p(x)$, mean m_x and covariance matrix P_x. Assume we are given a measurement $y = h(x) + v$ of x, where v is a 0-mean noise, independent of x.

We know that the optimal estimation in the LSQ sense of x given y is $\bar{x}_y = E[x|y]$. Usually it is not possible to derive a closed form expression for this quantity and so we only look for an affine estimator $\hat{x} = Ay + b$, for some optimal A and b. So we look for an estimator:

$$\hat{x} = A^* y + b^*$$

where $A^* \in \Re^{n \times q}$ and $b^* \in \Re^n$ are the values of A and b minimizing:

$$J_{LSQ} = E[(x - \hat{x})'(x - \hat{x})] = E[tr\{(x - \hat{x})(x - \hat{x})'\}] = tr\{E[(x - \hat{x})(x - \hat{x})']\}$$

$$= tr\{E[(x - Ay - b)(x - Ay - b)']\} \tag{1.5}$$

The expectation is taken w.r.t. the random variables x and v (v is the noise in the measurement). In fact, since we are assuming that it is too difficult to compute the expectation of x given y, we determine A and b which minimize the expected square error w.r.t. all possible y that may occur. This may be expressed in closed form and corresponds to take the expectation w.r.t. all possible x and v (being $y = h(x) + v$). Notice that if the LSQ estimator was really affine, i.e. if it was

$$\bar{x}_y = E[x|y] = \int x p(x|y) dx = Ay + b$$

for all y, then our procedure would determine the correct A^* and b^*. In fact, in that case, the optimal A_y and b_y

$$[A_y, b_y] = \arg\min_{A,b} J_{LSQ,y} = \arg\min_{A,b} E[(x - Ay - b)'(x - Ay - b)|y] \equiv [A^*, b^*]$$

would be always the same for all y. Now, if we are given a parameterized function $g_y(r)$, such that $\arg\min_r g_y(r) = \bar{r}$ is the same for all y, clearly also $\arg\min_r \int g_y(r) p(y) dy = \bar{r}$ for any (non negative) weighting function $p(y)$. For this reason, in the case the LSQ estimator of x given y is really affine, A^* and b^* can also be computed as the values of A and b minimizing

$$J_{LSQ} = E[J_{LSQ,y}] = \int J_{LSQ,y} p(y) dy = \int E[(x - Ay - b)'(x - Ay - b)|y] p(y) dy$$

$$= \int\int (x - Ay - b)'(x - Ay - b) p(x|y) p(y) dx dy = \int\int (x - Ay - b)'(x - Ay - b) p(y|x) p(x) dy dx$$

$$= \int\int (x - Ay - b)'(x - Ay - b) p_v(y - h(x)) p(x) dy dx$$

$$= \int\int (x - A(h(x) + v) - b)'(x - A(h(x) + v) - b) p_v(v) p(x) dx dv$$

This fact, together with the possibility of obtaining a closed form expression for the estimator, motivates the choice of selecting A and b in order to minimize the expected (square) difference between x and $\hat{x} = Ay + b$ taking the expectation w.r.t. the random variables x and v, i.e. before knowing the value of y. If the LSQ estimator is not affine, this choice is clearly an approximation, which corresponds to select A and b providing the best LSQ behavior in the average w.r.t. all the y that can be observed. So, from (1.5):

$$[A^*, b^*] = \arg\min_{A,b} J_{LSQ} = \arg\min_{A,b} tr\{E[(x - Ay - b)(x - Ay - b)']\}$$

$$= \arg\min_{A,b} tr\{-E[xy']A' - E[x]b' - AE[yx'] + AE[yy']A' + AE[y]b' - bE[x]' + bE[y]'A' + bb'\}$$

where all the expectations are taken, as mentioned, w.r.t. x and v. By introducing the notation $m_x = E[x]$, $m_y = E[y] = E[h(x) + v]$ and observing that $E[yy'] = \Sigma_y + m_y m_y'$ (since $\Sigma_y = E[(y - m_y)(y - m_y)'] = E[yy'] - m_y m_y'$), and similarly $E[xy'] = \Sigma_{xy} + m_x m_y'$, we have:

$$[A^*, b^*] = \arg\min_{A,b} J_{LSQ} =$$

$$= \arg\min_{A,b} tr\{-(\Sigma_{xy}+m_x m_y')A' - m_x b' - A(\Sigma_{yx}+m_y m_x') + A(\Sigma_y+m_y m_y')A' + Am_y b' - bm_x' + bm_y' A' + bb'\}$$

The covariance matrices Σ_y and Σ_{xy} do not depend on the estimator but are intrinsic to the process. In fact $\Sigma_y = E[(y - E[h(x)])(y - E[h(x)])']$ only depends on $p(x)$ and on the noise v (the same holds for Σ_{xy}).

Taking the derivatives w.r.t. A and b we get[3]:

$$\frac{\partial J_{LSQ}}{\partial b} = -2m_x' + 2m_y' A' + 2b' \tag{1.6}$$

$$\frac{\partial J_{LSQ}}{\partial A} = -2(\Sigma_{yx} + m_y m_x') + 2(\Sigma_y + m_y m_y')A' + 2m_y b' \tag{1.7}$$

Putting them at 0 we get:

$$\begin{aligned} -m_x' + m_y' A' + b' &= 0 \\ -(\Sigma_{yx} + m_y m_x') + (\Sigma_y + m_y m_y')A' + m_y b' &= 0 \end{aligned}$$

From equation (1.6):

$$b' = m_x' - m_y' A'$$

which, substituted in (1.7), gives:

$$-(\Sigma_{yx} + m_y m_x') + (\Sigma_y + m_y m_y')A' + m_y(m_x' - m_y' A') = -\Sigma_{yx} + \Sigma_y A' = 0$$

i.e.

$$A' = \Sigma_y^{-1}\Sigma_{yx}$$

which finally gives:

$$\begin{aligned} A^* &= \Sigma_{xy}\Sigma_y^{-1} \\ b^* &= m_x - A^* m_y \end{aligned}$$

[3]The way to compute the derivative of the trace of a matrix with respect to vectors and matrices is not straightforward: try first with a scalar case to get familiarity with the given equations. For the multi-dimensional case, we proceed as follows. First, we use as derivative of a scalar function ϕ w.r.t. a vector the usual gradient of ϕ, while, in a similar way, the derivative of ϕ w.r.t. a matrix A is a matrix which element (i, j) is the derivative of ϕ w.r.t. A_{ji}. Using this definition it is possible to see that the derivative of $tr\{AB\}$ w.r.t. B is A. In fact, assume A is $n \times q$ and B is $q \times n$, as it occurs in our derivations. Then $\phi = trace\{AB\} = \sum_{i=1}^{n}\sum_{j=1}^{q} A_{ij}B_{ji}$ with $\frac{\partial \phi}{\partial B_{ij}} = A_{ji}$. So, defining the derivative w.r.t. a matrix as mentioned above, we get the assessed property $\frac{\partial \phi}{\partial B} = A$. Finally, if a is a vector and we consider $\phi = tr\{aa'\}$ we have: $\phi = \sum_{i=1}^{n} a_i^2$ with the gradient of ϕ given in fact by $\frac{\partial \phi}{\partial a} = 2a'$. Similarly, if $\phi = tr(ab')$, $\frac{\partial \phi}{\partial a} = b'$.

with the optimal affine LSQ estimator given by:

$$\hat{x} = A^* y + b^* = A^* y + m_x - A^* m_y = m_x + A^*(y - m_y) = m_x + \Sigma_{xy}\Sigma_y^{-1}(y - m_y)$$

The covariance is given by:

$$P = E[(x - \hat{x})(x - \hat{x})'] = E[(x - m_x - \Sigma_{xy}\Sigma_y^{-1}(y - m_y))(x - m_x - \Sigma_{xy}\Sigma_y^{-1}(y - m_y))']$$

$$= E[(x - m_x)(x - m_x)'] - E[(x - m_x)(y - m_y)']\Sigma_y^{-1}\Sigma_{yx} - \Sigma_{xy}\Sigma_y^{-1}E[(y - m_y)(x - m_x)']$$

$$+\Sigma_{xy}\Sigma_y^{-1}E[(y - m_y)(y - m_y)']\Sigma_y^{-1}\Sigma_{yx} = P_x - \Sigma_{xy}\Sigma_y^{-1}\Sigma_{yx}$$

So, coming back to the notation of the filter and observing that $m_x = E[f(x_{k-1}, u_{k-1}, w_{k-1})] = \hat{x}_k^-$ (i.e. the expected value of x_k before taking the measurement y_k), and $m_y = E[y_k] = E[h(x_k) + v_k] = E[h(x_k)]$ (being $E[v_k] = 0$) and $P_x = P_k^-$, we have:

$$\hat{x}_k = \hat{x}_k^- + \Sigma_{xy,k}\Sigma_{y,k}^{-1}(y_k - E[h(x_k)]) \tag{1.8}$$

and

$$P_k = P_k^- - \Sigma_{xy,k}\Sigma_{y,k}^{-1}\Sigma_{yx,k} \tag{1.9}$$

where $\Sigma_{y,k} = E[(y_k - E[y_k])(y_k - E[y_k])']$ and $\Sigma'_{yx,k} = \Sigma_{xy,k} = E[(x_k - E[x_k])(y_k - E[y_k])']$, and all the expectations are taken w.r.t. the random variables $x_k|y^{k-1}, u^{k-1}$ (i.e. using the pdf of x_k before integrating the last measurement y_k) and the noise v_k.

1.2 Implementation through linearization: the Extended Kalman Filter (EKF)

If the dynamic function f and the measurement function h are linear, the equations (1.1)-(1.4) of the optimal affine LSQ estimator result in the equations of the KF. Remember however that in the linear case, if the noise is also Gaussian, the KF is optimal in the LSQ sense not only among linear estimators but among *all* estimators (and is actually the *exact* solution to the filtering problem, as mentioned previously).

In the non linear case, a first possibility to provide an approximate implementation of the equations (1.1)-(1.4) of the optimal affine LSQ estimator is based on a linearization of f and h around the estimates. This is the EKF.

1.2.1 Prediction step of the EKF

We report here for convenience the equations (1.1)-(1.2) of the prediction step of the optimal affine LSQ estimator:

$$\hat{x}_k^- = E[f(x_{k-1}, u_{k-1}, w_{k-1})] \tag{1.10}$$

$$P_k^- = E[(x_k - \hat{x}_k^-)(x_k - \hat{x}_k^-)'] \tag{1.11}$$

and introduce the linearization of f. So we can write

$$f(x_{k-1}, u_{k-1}, w_{k-1}) \approx f(\hat{x}_{k-1}, u_{k-1}, 0) + \left.\frac{\partial f}{\partial x}\right|_{(\hat{x}_{k-1}, u_{k-1}, 0)} (x_{k-1} - \hat{x}_{k-1})$$

$$+ \left.\frac{\partial f}{\partial w}\right|_{(\hat{x}_{k-1}, u_{k-1}, 0)} w_{k-1} \tag{1.12}$$

having neglected higher order terms. Substituting this linear approximation in (1.10) gives:

$$E[f(x_{k-1},u_{k-1},w_{k-1})] \approx f(\hat{x}_{k-1},u_{k-1},0) + \left.\frac{\partial f}{\partial x}\right|_{(\hat{x}_{k-1},u_{k-1},0)} E[x_{k-1}-\hat{x}_{k-1}]$$

$$+ \left.\frac{\partial f}{\partial w}\right|_{(\hat{x}_{k-1},u_{k-1},0)} E[w_{k-1}] \approx f(\hat{x}_{k-1},u_{k-1},0)$$

being $\hat{x}_{k-1} \approx E[x_{k-1}]$ in the limit of the linear approximation adopted[4] and $E[w_{k-1}]=0$. So we take:

$$\hat{x}_k^- = f(\hat{x}_{k-1},u_{k-1},0).$$

Then, substituting this linear approximation in (1.11) gives:

$$E[(x_k-\hat{x}_k^-)(x_k-\hat{x}_k^-)']$$

$$= E\left[[f(x_{k-1},u_{k-1},w_{k-1})-f(\hat{x}_{k-1},u_{k-1},0)][f(x_{k-1},u_{k-1},w_{k-1})-f(\hat{x}_{k-1},u_{k-1},0)]'\right]$$

Substituting the expression of the linearized f given in (1.12), simplifying and adopting the positions:

$$F_{x,k-1} = \left.\frac{\partial f}{\partial x}\right|_{(\hat{x}_{k-1},u_{k-1},0)}$$

$$F_{w,k-1} = \left.\frac{\partial f}{\partial w}\right|_{(\hat{x}_{k-1},u_{k-1},0)},$$

we obtain:

$$E[(x_k-\hat{x}_k^-)(x_k-\hat{x}_k^-)']$$

$$\approx E\left[(F_{x,k-1}(x_{k-1}-\hat{x}_{k-1})+F_{w,k-1}w_{k-1})\,(F_{x,k-1}(x_{k-1}-\hat{x}_{k-1})+F_{w,k-1}w_{k-1})'\right],$$

which, exploiting the independence of w_{k-1} from previous noises and the noise on x_0, becomes:

$$E[(x_k-\hat{x}_k^-)(x_k-\hat{x}_k^-)'] \approx F_{x,k-1}E[(x_{k-1}-\hat{x}_{k-1})(x_{k-1}-\hat{x}_{k-1})']F_{x,k-1}' + F_{w,k-1}E[w_{k-1}w_{k-1}']F_{w,k-1}'$$

$$\approx F_{x,k-1}P_{k-1}F_{x,k-1}' + F_{w,k-1}Q_{k-1}F_{w,k-1}'.$$

So also in this case we take the approximate expression

$$P_k^- = F_{x,k-1}P_{k-1}F_{x,k-1}' + F_{w,k-1}Q_{k-1}F_{w,k-1}'$$

To conclude, the prediction step of the EKF is given by:

$$\hat{x}_k^- = f(\hat{x}_{k-1},u_{k-1},0) \tag{1.13}$$

$$P_k^- = F_{x,k-1}P_{k-1}F_{x,k-1}' + F_{w,k-1}Q_{k-1}F_{w,k-1}' \tag{1.14}$$

Notice how these expressions look very similar to the ones in the prediction step of the KF, if the matrices A and $B_w = I$ are replaced by the Jacobians of f.

[4] Actually, already at the first step, even if $\hat{x}_0$ is exactly taken as the expected value $E[x_0]$, the position $\hat{x}_1^- = f(\hat{x}_0,u_0,0) = f(E[x_0],u_0,0)$ is different from the true expected value $E[f(x_0,u_0,w_0)]$ being coincident only if we neglect higher order terms in the Taylor expansion of $f(x_0,u_0,w_0)$. Another source of error is the fact we are considering only an *affine* LSQ estimator rather than a *true* LSQ estimator.

1.2.2 Correction step of the EKF

We report here for convenience the equations (1.3)-(1.4) of the correction step of the optimal affine LSQ estimator:

$$\hat{x}_k = \hat{x}_k^- + \Sigma_{xy,k}\Sigma_{y,k}^{-1}(y_k - E[h(x_k)]) \tag{1.15}$$
$$P_k = P_k^- - \Sigma_{xy,k}\Sigma_{y,k}^{-1}\Sigma_{yx,k} \tag{1.16}$$

Now we introduce the linearization of h:

$$h(x_k) \approx h(\hat{x}_k^-) + \left.\frac{\partial h}{\partial x}\right|_{\hat{x}_k^-}(x_k - \hat{x}_k^-) \tag{1.17}$$

Using this expression, and introducing the position

$$H_{x,k} = \left.\frac{\partial h}{\partial x}\right|_{\hat{x}_k^-}$$

we get

$$E[y_k] = E[h(x_k)] \approx h(\hat{x}_k^-)$$

being $\hat{x}_k^- \approx E[x_k]$ (see e.g. (1.10) where $f(x_{k-1}, u_{k-1}, w_{k-1})$ is actually x_k).

Then we have:

$$\Sigma_{y,k} = E[(y_k - E[y_k])(y_k - E[y_k])'] = E[(h(x_k) + v_k - E[y_k])(h(x_k) + v_k - E[y_k])']$$

which, exploiting the linearization of h in (1.17) and the independence of v_k from previous noises and from w_k becomes:

$$\Sigma_{y,k} \approx E[(H_{x,k}(x_k - \hat{x}_k^-) + v_k)(H_{x,k}(x_k - \hat{x}_k^-) + v_k)'] \approx H_{x,k}E[(x_k - \hat{x}_k^-)(x_k - \hat{x}_k^-)']H_{x,k}'$$
$$+E[v_k v_k'] \approx H_{x,k}P_k^- H_{x,k}' + R_k$$

Similarly,

$$\Sigma_{xy,k} = E[(x_k - E[x_k])(y_k - E[y_k])'] \approx E[(x_k - \hat{x}_k^-)(H_{x,k}(x_k - \hat{x}_k^-) + v_k)']$$
$$= E[(x_k - \hat{x}_k^-)(x_k - \hat{x}_k^-)']H_{x,k}' \approx P_k^- H_{x,k}'$$

Substituting all these expressions in (1.15)-(1.16) gives:

$$\hat{x}_k = \hat{x}_k^- + P_k^- H_{x,k}'(H_{x,k}P_k^- H_{x,k}' + R_k)^{-1}(y_k - h(\hat{x}_k^-))$$
$$P_k = P_k^- - P_k^- H_{x,k}'(H_{x,k}P_k^- H_{x,k}' + R_k)^{-1}H_{x,k}P_k^-.$$

Introducing the Kalman Gain $K_k = P_k^- H_{x,k}'(H_{x,k}P_k^- H_{x,k}' + R_k)^{-1}$ we have for the covariance matrix:

$$P_k = P_k^- - K_k H_{x,k}P_k^- = (I - K_k H_{x,k})P_k^-.$$

In conclusion the correction step of the EKF is given by:

$$\hat{x}_k = \hat{x}_k^- + K_k(y_k - h(\hat{x}_k^-)) \tag{1.18}$$
$$P_k = (I - K_k H_{x,k})P_k^- \tag{1.19}$$
$$K_k = P_k^- H_{x,k}'(H_{x,k}P_k^- H_{x,k}' + R_k)^{-1} \tag{1.20}$$

Notice again how these expressions look very similar to the ones in the correction step of the KF, if the matrix C is replaced by the Jacobian of h.

The following algorithm summarizes all the steps of the EKF.

Algorithm 1 Extended Kalman Filter (EKF)

Assume x_0 is a random vector with mean m_0 and covariance matrix P_0 and initialize the filter by $\hat{x}_0 = m_0$. At each time $k \geq 1$ we have the following recursive equations:

$$\begin{aligned} Prediction: \hat{x}_k^- &= f(\hat{x}_{k-1}, u_{k-1}, 0) \\ P_k^- &= F_{x,k-1} P_{k-1} F_{x,k-1}' + F_{w,k-1} Q_{k-1} F_{w,k-1}' \\ Correction: \hat{x}_k &= \hat{x}_k^- + K_k(y_k - h(\hat{x}_k^-)) \\ P_k &= (I - K_k H_{x,k}) P_k^- \\ K_k &= P_k^- H_{x,k}' (H_{x,k} P_k^- H_{x,k}' + R_k)^{-1} \end{aligned}$$

where

$$F_{x,k-1} = \left.\frac{\partial f}{\partial x}\right|_{(\hat{x}_{k-1}, u_{k-1}, 0)} \qquad F_{w,k-1} = \left.\frac{\partial f}{\partial w}\right|_{(\hat{x}_{k-1}, u_{k-1}, 0)} \qquad H_{x,k} = \left.\frac{\partial h}{\partial x}\right|_{\hat{x}_k^-}.$$

1.3 Implementation through Unscented Transformation: the Unscented Kalman Filter (UKF)

Another way to provide an approximate implementation of the optimal affine LSQ estimator (1.1)-(1.4) is based on the Unscented Transformation (UT), a trick to propagate mean and covariance of random variables (or vectors) through non linear transformations.

1.3.1 The Unscented Transformation

Given a random vector x with mean m_x and covariance matrix P_x, consider the random vector $y = f(x)$. We know that if f is linear (i.e. $y = Ax$), the expected value $m_y = E[y] = E[f(x)]$ of y and its covariance matrix $P_y = E[(y - m_y)(y - m_y')]$ are simply given by $m_y = Am_x$ and $P_y = AP_xA'$. In addition, if x is a Gaussian random vector, also y remains a Gaussian random vector. If f is non linear, it is not straightforward in general to compute m_y and P_y, since y has a generic pdf $p(y)$ even if x is Gaussian. To determine an approximation of m_y and of the covariance matrix P_y (actually, as mentioned, this corresponds to find a Gaussian pdf $\hat{p}(y)$ fitting the true pdf $p(y)$ of y) it is possible to proceed as illustrated in the EKF section, by taking $m_y \approx f(m_x)$ and $P_y \approx FP_xF'$, where $F = df/dx$ is the Jacobian of f w.r.t. x (this corresponds to linearize the non linear f around m_x).

A smarter approach is based on the Unscented Transformation, consisting of the following procedure.

Algorithm 2 The Unscented Transformation (UT)

- *Let x be an n dimensional random vector with mean m_x and covariance matrix P_x. Based on m_x and P_x, compute a set of $2n+1$* Sigma points *ξ_i and corresponding weights W_i (see Algorithm 3 below), such that*

$$\sum_{i=1}^{2n+1} W_i = 1 \qquad \sum_{i=1}^{2n+1} W_i \xi_i = m_x \qquad \sum_{i=1}^{2n+1} W_i (\xi_i - m_x)(\xi_i - m_x)' = P_x. \tag{1.21}$$

- *Determine the transformation $\mathcal{Y}_i = f(\xi_i)$ of each Sigma Point ξ_i, $i = 1, 2, \ldots, 2n+1$.*
- *The approximate mean and covariance of $y = f(x)$ are then computed by:*

$$\begin{aligned} \hat{m}_y &\approx \sum_{i=1}^{2n+1} W_i \mathcal{Y}_i \\ \hat{P}_y &\approx \sum_{i=1}^{2n+1} W_i (\mathcal{Y}_i - \hat{m}_y)(\mathcal{Y}_i - \hat{m}_y)' \end{aligned}$$

This usually provides a much better approximation of the mean and of the covariance of $y = f(x)$ w.r.t. the approximation based on Jacobians[5]. The selection of the Sigma Points (the first step in Algorithm 2) can be performed by considering a symmetric set of values, as in the following algorithm.

Algorithm 3 Sigma Points generation: $[W, \Xi] = \{W_i, \xi_i\}_{i=1,\ldots,2n+1} = SigmaPoints(m_x, P_x)$

Let x be an n dimensional random vector with mean m_x and covariance matrix P_x. Let $\mathcal{K} \in \Re$ be an arbitrary quantity such that $\mathcal{K} + n > 0$. Then set:

$$\begin{aligned} \xi_{2n+1} &= m_x & W_{2n+1} &= \frac{\mathcal{K}}{n+\mathcal{K}} \\ \xi_i &= m_x + \sqrt{n+\mathcal{K}}\, L_i' & W_i &= \frac{1}{2(n+\mathcal{K})} \quad i = 1, \ldots, n \\ \xi_{i+n} &= m_x - \sqrt{n+\mathcal{K}}\, L_i' & W_{i+n} &= \frac{1}{2(n+\mathcal{K})} \quad i = 1, \ldots, n \end{aligned}$$

where $L_i = (P_x^{1/2})_i$ is the i-th row of the square root of the matrix P_x, in the sense that $L'L = P_x$ (in MATLAB, $L = chol(P_x)$ is the Choleski decomposition of P_x and is an efficient procedure to compute L such that $L'L = P_x$).

It is easy to verify that the set of vectors and weights derived in Algorithm 3 meets all the conditions in (1.21). The quantity $\mathcal{K}$ is a degree of freedom which can be used to improve the quality of the approximation: in our applications to mobile robot localization where $n = 3$ we have always selected $\mathcal{K} = 0$.

1.3.2 The Unscented Kalman Filter

Using the UT, it is quite straightforward to implement the optimal affine LSQ estimator (1.1)-(1.4).

Let's start with the prediction step and report for convenience the equations (1.1)-(1.2) of the optimal affine LSQ estimator:

$$\hat{x}_k^- = E[f(x_{k-1}, u_{k-1}, w_{k-1})] \tag{1.22}$$

$$P_k^- = E[(x_k - \hat{x}_k^-)(x_k - \hat{x}_k^-)'] \tag{1.23}$$

As in the EKF case, we have (in the limit of the approximation adopted) that $\hat{x}_{k-1} = E[x_{k-1}]$ and, consequently, $E[(x_{k-1} - E[x_{k-1}])(x_{k-1} - E[x_{k-1}])'] = E[(x_{k-1} - \hat{x}_{k-1})(x_{k-1} - \hat{x}_{k-1})'] = P_{k-1}$.

To compute $E[f(x_{k-1}, u_{k-1}, w_{k-1})]$ using the UT we apply the following procedure. We define an extended state $x_{a,k-1}$ comprising the previous state x_{k-1} and the noise w_{k-1}, with dimension n_a. In view of the independence of x_{k-1} and w_{k-1} this extended state $x_{a,k-1}$ is a random vector with a mean $\hat{x}_{a,k-1}$ and a block diagonal covariance matrix $P_{a,k-1}$ given by:

$$\hat{x}_{a,k-1} = \begin{bmatrix} \hat{x}_{k-1} \\ 0 \end{bmatrix} \quad P_{a,k-1} = \begin{bmatrix} P_{k-1} & 0 \\ 0 & Q_{k-1} \end{bmatrix}$$

According to $x_a = [x', w']'$, define also the following function:

$$f_a(x_a, u) = f(x, u, w)$$

The approximate determination of $\hat{x}_k^- = E[f(x_{k-1}, u_{k-1}, w_{k-1})] = E[f_a(x_{a,k-1}, u_{k-1})]$ can then been performed using the UT:

[5]It is possible to show that the unscented transformation calculates the mean and the covariance correctly to the second order while the linearized approach only considers a first order approximation

- Compute the set of $2n_a+1$ Sigma Points applying the procedure in Algorithm 3 with $m_x = \hat{x}_{a,k-1}$ and $P_x = P_{a,k-1}$:
$$[W, \Xi_{k-1}] = sigmaPoints(\hat{x}_{a,k-1}, P_{a,k-1});$$
- For each Sigma Point $\xi_{j,k-1}$, $j = 1, \ldots, 2n_a + 1$, apply the dynamics:
$$\xi^-_{j,k} = f_a[\xi_{j,k-1}, u_{k-1}]$$
- Get the a priori estimate and its covariance matrix:
$$\begin{aligned} \hat{x}^-_k &= \sum_{j=1}^{2n_a+1} W_j \xi^-_{j,k} \\ P^-_k &= \sum_{j=1}^{2n_a+1} W_j (\xi^-_{j,k} - \hat{x}^-_k)(\xi^-_{j,k} - \hat{x}^-_k)' \end{aligned}$$

As for the correction step, we have to implement (1.3)-(1.4) reported here for convenience:

$$\hat{x}_k = \hat{x}^-_k + \Sigma_{xy,k}\Sigma^{-1}_{y,k}(y_k - E[h(x_k)]) \quad (1.24)$$
$$P_k = P^-_k - \Sigma_{xy,k}\Sigma^{-1}_{y,k}\Sigma_{yx,k} \quad (1.25)$$

This requires the (approximate) determination of $E[h(x_k)]$, $\Sigma_{y,k}$ and $\Sigma_{xy,k}$. Now, x_k, based on the prediction step, is characterized (approximately) by a mean $\hat{x}^-_k$ and a covariance matrix P^-_k: the vectors $\xi^-_{j,k}$ are actually the Sigma Points of this distribution, with weights W_j. So, according to the UT, if we introduce the notation

$$\begin{aligned} \mathcal{Y}_{j,k} &= h(\xi^-_{j,k}) \\ y^-_k &= \sum_{j=1}^{2n_a+1} W_j \mathcal{Y}_{j,k}, \end{aligned}$$

we have the following approximate expressions:

$$\begin{aligned} E[h(x_k)] &\approx y^-_k \\ \Sigma_{y,k} &\approx E[(y_k - y^-_k)(y_k - y^-_k)'] \approx \sum_{j=1}^{2n_a+1} W_j (\mathcal{Y}_{j,k} - y^-_k)(\mathcal{Y}_{j,k} - y^-_k)' + R_k \\ \Sigma_{xy,k} &\approx \sum_{j=1}^{2n_a+1} W_j (\xi^-_{j,k} - \hat{x}^-_k)(\mathcal{Y}_{j,k} - y^-_k)' \end{aligned}$$

The reason for the presence of R_k in the expression of $\Sigma_{y,k}$ is the fact that $y_k = h(x_k) + v_k$. So, taking into account that $E[h(x_k)] = y^-_k$ we have:

$$\Sigma_{y,k} \approx E[(y_k - y^-_k)(y_k - y^-_k)'] \approx E[(h(x_k) + v_k - E[h(x_k)])(h(x_k) + v_k - E[h(x_k)])']$$
$$\approx \underbrace{E[(h(x_k) - E[h(x_k)])(h(x_k) - E[h(x_k)])']}_{\sum_{j=1}^{2n_a+1} W_j (\mathcal{Y}_{j,k} - y^-_k)(\mathcal{Y}_{j,k} - y^-_k)'} + \underbrace{E[v_k v_k']}_{R_k}$$

The quantities just determined must be substituted in (1.24)-(1.25) to get the correction step of the UKF. The complete UKF is sketched in the following algorithm.

Algorithm 4 The Unscented Kalman Filter (UKF)

Assume x_0 is a random vector with mean m_0 and covariance matrix P_0 and initialize the filter by $\hat{x}_0 = m_0$. At each time $k \geq 1$ we have the following recursive equations:

- *Define the n_a dimensional vector and the $n_a \times n_a$ matrix:*

$$\hat{x}_{a,k-1} = \begin{bmatrix} \hat{x}_{k-1} \\ 0 \end{bmatrix} \quad P_{a,k-1} = \begin{bmatrix} P_{k-1} & 0 \\ 0 & Q_{k-1} \end{bmatrix}$$

- *Compute the set of $2n_a + 1$ Sigma Points (see Algorithm 3):*

$$[W, \Xi_{k-1}] = sigmaPoints(\hat{x}_{a,k-1}, P_{a,k-1});$$

- *For each Sigma Point $\xi_{j,k-1}$, $j = 1, \ldots, 2n_a + 1$, apply the dynamics:*

$$\xi^-_{j,k} = f_a[\xi_{j,k-1}, u_{k-1}]$$

where $f_a(x_a, u) = f(x, u, w)$.

- *(PREDICTION) Get the a priori estimate and its covariance matrix:*

$$\begin{aligned} \hat{x}^-_k &= \sum_{j=1}^{2n_a+1} W_j \xi^-_{j,k} \\ P^-_k &= \sum_{j=1}^{2n_a+1} W_j (\xi^-_{j,k} - \hat{x}^-_k)(\xi^-_{j,k} - \hat{x}^-_k)' \end{aligned}$$

- *For each transformed Sigma Point $\xi^-_{j,k}$, $j = 1, \ldots, 2n_a + 1$, compute the measurement vector:*

$$\mathcal{Y}_{j,k} = h(\xi^-_{j,k})$$

- *Then, the a priori expected measurement vector is:*

$$y^-_k = \sum_{j=1}^{2n_a+1} W_j \mathcal{Y}_{j,k}$$

- *Compute the covariance matrices:*

$$\begin{aligned} P_y &= \sum_{j=1}^{2n_a+1} W_j (\mathcal{Y}_{j,k} - \hat{y}^-_k)(\mathcal{Y}_{j,k} - \hat{y}^-_k)' + R_k \\ P_{xy} &= \sum_{j=1}^{2n_a+1} W_j (\xi^-_{j,k} - \hat{x}^-_k)(\mathcal{Y}_{j,k} - \hat{y}^-_k)' \end{aligned}$$

- *Compute the Kalman gain:*

$$K_k = P_{xy} P_y^{-1}$$

- *(CORRECTION) Compute the estimate and update the covariance matrix*

$$\begin{aligned} \hat{x}_k &= \hat{x}^-_k + K_k \left(y_k - \hat{y}^-_k \right) \\ P_k &= P^-_k - K_k P_y K_k' \end{aligned}$$