Optimal Estimation

Bryan Chiang and Jeannette Bohg

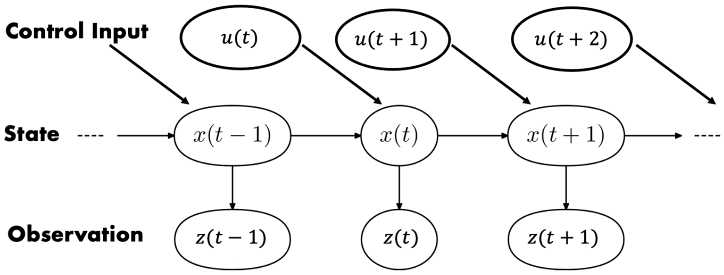
February 23, 2022

1 State estimation

Knowing the current state of the environment is crucial for artificial agents to reliably perform real-time decision-making. In **state estimation**, we aim to infer the latent state of the system at the current point in time by continuously combining measurements from different sensor sources (multi-modal perception).

We first examine **discrete-time dynamical systems** from a probabilistic perspective. Figure 1 visualizes a **partially observable Markov decision process** (POMDP) in form of a graphical model.

Figure 1: Graphical POMDP model.



The directed edges indicate conditional dependence relations. $x_t \in \mathbb{R}^n$ is the state at time t and it only depends on the previous state x_{t-1}. $z_t \in \mathbb{R}^k$ is a sensor observation that depends on the state at time t. $u_t \in \mathbb{R}^m$ represents the control input applied at time t. For instance, the state of an artificial agent situated in some environment could be position, orientation, linear and angular velocity, or any combination of the above. Similarly, measurements derived from the state could be camera images or Lidar measurements of the environment that depend on the robot position and movement.

The measurements and control inputs are known, but the hidden state history $x_{0:t}$ is unknown. We assume that the system evolves in a stochastic manner and that observations are stochastic. Therefore, we model the state X_t and observations Z_t as random variables that can take on x_t and z_t as possible values. To simplify notation, we write $p(X_t = x_t)$ as $p(x_t)$ and similarly for Z_t.

The goal of a **state estimator** is to compute or approximate the posterior probability distribution of the state x_t given the available **data** (histories of measurements and control inputs) and known **models** for state transitions and observations. Specifically, we want to know the posterior distribution of the state $p(x_t|z_{1:t}, u_{1:t})$, which assigns a probability to every possible value that the state can take on given the sequence of measurements and control inputs. The posterior is also called the **belief** about the state's value at time t, represented by $\text{bel}(t)$. The probabilistic formulation does not give an exact value for the state, but enables us to quantify our uncertainty about what the state may be.

By representing a dynamical system using the graphical model in Figure 1, we assume that the **state is complete**. This assumptions leads to two key properties. First, we assume that the system is Markovian, i.e. the current state x_t only depends on the previous state x_{t-1} and previous control input u_{t-1}, as opposed to the entire history $z_{1:t-1}$. This can be expressed as

$$p(x_t|x_{0:t-1}, z_{1:t-1}, u_{1:t}) = p(x_t|x_{t-1}, u_t) \tag{1}$$

which is also called the **transition model**, representing how likely the system transitions to state x_t when currently in x_{t+1} and given control input u_t.

Second, we assume that that current measurement z_t only depends on the current state x_t, i.e. z_t is conditionally independent of all previous states $x_{0:t-1}$, measurements $z_{1:t-1}$ and control inputs $u_{1:t}$. This can be expressed as

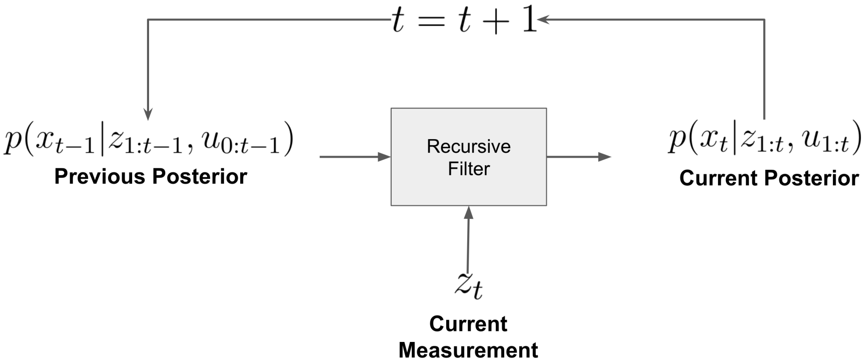
$$p(z_t|x_{0:t}, z_{1:t-1}, u_{1:t}) = p(z_t|x_t) \tag{2}$$

This is also referred to as the **measurement model**, which represents how likely a measurement z_t is given the state x_t.

2 Bayes filter

A recursive filter continuously ingests new measurements to estimate the state posterior as illustrated in Figure (2). At each time step t, we compute our new posterior $p(x_t|z_{1:t}, u_{0:t})$ using only our old posterior $p(x_{t-1}|z_{1:t-1}, u_{0:t-1})$, new control input u_{t-1} and new measurement z_t. Thus, the complexity of a recursive filter is constant with respect to time; it does not depend on the size of the history and is suitable for real-time inference.

Figure 2: Diagram of a general recursive filter.



2.1 Conditional probability review

We can factorize a joint probability distribution as $p(A, B) = p(A|B)p(B)$, where A and B are random variables. If the joint distribution is conditioned on another random variable C, we can just pass the conditioning along as $p(A, B|C) = p(A|B, C)p(B|C)$. We can marginalize a distribution as $p(A) = \int_B p(A, B)dB$. Similar to before, we have $p(A|C) = \int_B p(A, B|C)dB$. We can factorize a joint distribution in two separate ways, resulting in **Bayes rule**, which is $p(A|B)p(B) = p(B|A)p(A)$.

2.2 Derivation

We now derive the **Bayes filter**, the simplest recursive filter. Our goal is to derive a recursive expression for the state posterior that only depends on previous posterior, the current measurement z_t, control input u_t, and the transition and measurement models. The models are assumed to be known and tractable to compute $p(x_t|x_{t-1}, u_t)$ and $p(z_t|x_t)$. Bayes rule and conditional independence of measurements gives us:

$$
\begin{aligned}
p(x_t|z_{1:t}, u_{1:t}) &= \frac{p(z_t|x_t, z_{1:t-1}, u_{1:t})p(x_t|z_{1:t-1}, u_{1:t})}{p(z_t|z_{1:t-1}, u_{1:t})} && (3) \\
&= \frac{p(z_t|x_t)p(x_t|z_{1:t-1}, u_{1:t})}{p(z_t|z_{1:t-1}, u_{1:t})} && (4)
\end{aligned}
$$

We then simplify the two terms on the right hand side that still depend on the entire history of observations. Let us first look at the denominator. Marginalizing over x_t and using the conditional independence of measurements gives:

$$
\begin{aligned}
p(z_t|z_{1:t-1}, u_{1:t}) &= \int_{x_t} p(z_t, x_t|z_{1:t-1}, u_{1:t})dx_t \\
&= \int_{x_t} p(z_t|x_t, z_{1:t-1}, u_{1:t})p(x_t|z_{1:t-1}, u_{1:t})dx_t \\
&= \int_{x_t} p(z_t|x_t)p(x_t|z_{1:t-1}, u_{1:t})dx_t
\end{aligned}
$$

Next, we can then simplify $p(x_t|z_{1:t-1}, u_{1:t})$, which now appears twice in Equation 4. Marginalizing over x_{t-1} and applying the Markov assumption gives:

$$
\begin{aligned}
p(x_t|z_{1:t-1}, u_{1:t}) &= \int_{x_{t-1}} p(x_t, x_{t-1}|z_{1:t-1}, u_{1:t})dx_{t-1} \\
&= \int_{x_{t-1}} p(x_t|x_{t-1}, z_{1:t-1}, u_{1:t})p(x_{t-1}|z_{1:t-1}, u_{1:t})dx_{t-1} \\
&= \int_{x_{t-1}} p(x_t|x_{t-1}, u_t)p(x_{t-1}|z_{1:t-1}, u_{1:t})dx_{t-1}
\end{aligned}
$$

We can now put together the full form of the Bayes filter, illustrated in Figures (3) and (4).

Figure 3: Diagram of the Bayes filter.

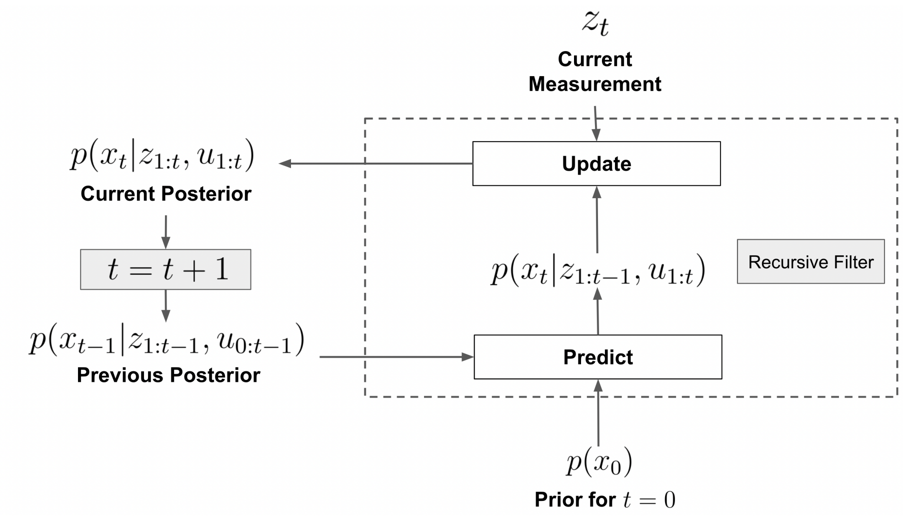
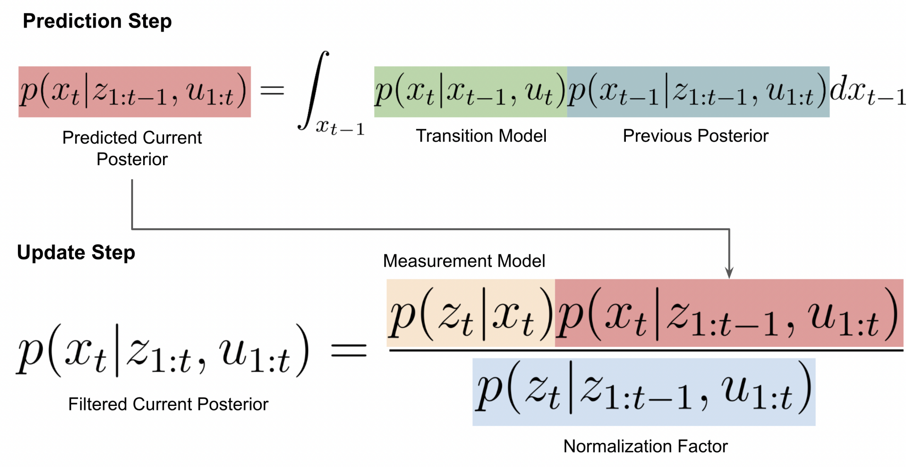


Figure 4: Predict and update step equations in the Bayes filter.


Prediction Step
$p(x_t|z_{1:t-1}, u_{1:t}) = \int_{x_{t-1}} p(x_t|x_{t-1}, u_t)p(x_{t-1}|z_{1:t-1}, u_{1:t})dx_{t-1}$
Predicted Current Posterior
Transition Model
Previous Posterior
Update Step
Measurement Model
$p(x_t|z_{1:t}, u_{1:t}) = \frac{p(z_t|x_t)p(x_t|z_{1:t-1}, u_{1:t})}{p(z_t|z_{1:t-1}, u_{1:t})}$
Filtered Current Posterior
Normalization Factor

There are two steps. In the **prediction step**, we use our transition model $p(x_t|x_{t-1}, u_t)$ and previous posterior to predict the state posterior at the current time step $p(x_t|z_{1:t-1}, u_{1:t})$. The prediction is represented as $\hat{\mathrm{bel}}(t-1)$, and it is our best guess for the distribution of x_t without any new information. When $t = 1$, the previous posterior does not exist yet, so we use a prior $p(x_0)$ that represents our initial belief about how x is distributed at the start $t = 0$. In the **update step**, we refine the predicted belief by incorporating the new measurement z_t. The **measurement likelihood**, $p(z_t|x_t)$, indicates how likely it would be to observe z_t given the predicted x_t. Thus, the numerator in the update step assigns a probability to a specified x_t based on both 1) how well x_t explains our measurement z_t and 2) how likely our dynamical system transitions from the previous posterior estimate x_{t-1} to x_t. The denominator is simply a **normalization constant** to ensure that our updated posterior distribution sums to 1. Once we predict and update our belief, we can repeat the process at $t+1$ as long as we have a new measurement z_{t+1}.

The Bayes filter is simple and generalizes to any probability distribution. In practice, however, it is limited since it is often intractable to compute the integral for the normalization constant. We need to know the entire probability distribution (e.g., in a tabular form for every value on the domain), which is often only possible when dealing with discrete random variables or when assuming a particular functional form of the posterior distribution, e.g. Gaussian. Thus, it is more useful to think of the Bayes filter as a "probabilistic template" that we can adapt to different sets of systems with specific needs and assumptions.

2.3 Discrete Bayes filter

Let's look at a practical implementation for the discrete case, where X_t and Z_t have a finite number of outcomes. Assume X_t is a discrete random variable with N possible values; the domain of X_t is $x_t \in \{1, 2, \dots, N\}$. Similarly, Z_t is a discrete random variable with M possible outcomes; the domain of Z_t is $z_t \in \{1, 2, \dots, M\}$. The predict and update equations for the **discrete Bayes filter** are the same as in the continuous case, except the integrals are replaced with summations.

Equation (5) is the belief vector representing the entire state posterior distribution at time t.

$$\mathrm{bel}(t) = \begin{pmatrix} p(X_t = 1 | z_{1:t}, u_{1:t}) \\ p(X_t = 2 | z_{1:t}, u_{1:t}) \\ \vdots \\ p(X_t = N | z_{1:t}, u_{1:t}) \end{pmatrix} \tag{5}$$

The predict step can then be implemented as a matrix multiplication:

$$\hat{\mathrm{bel}}(t-1) = T(u_t)\mathrm{bel}(t-1) \tag{6}$$

$$= \begin{pmatrix} p(X_t = 1 | z_{1:t-1}, u_{1:t}) \\ p(X_t = 2 | z_{1:t-1}, u_{1:t}) \\ \vdots \\ p(X_t = N | z_{1:t-1}, u_{1:t}) \end{pmatrix} \tag{7}$$

with $T(u_t)$ representing the transition probabilities:

$$T(u_t) = \begin{pmatrix} p(X_t = 1|X_{t-1} = 1, u_t) & \dots & p(X_t = 1|X_{t-1} = N, u_t) \\ \vdots & \ddots & \vdots \\ p(X_t = N|X_{t-1} = 1, u_t) & \dots & p(X_t = N|X_{t-1} = N, u_t) \end{pmatrix}. \tag{8}$$

Similarly, the update step can be implemented as a matrix multiplication:

$$\text{bel}(t) = \frac{M(z_t)\hat{\text{bel}}(t-1)}{\mathbf{1}^T(M(z_t)\hat{\text{bel}}(t-1))} \tag{9}$$

where M represents the measurement model probabilities:

$$M(z_t) = \begin{pmatrix} p(Z_t = 1|X_t = 1) & \dots & p(Z_t = 1|X_t = N) \\ \vdots & \ddots & \vdots \\ p(Z_t = M|X_t = 1) & \dots & p(Z_t = M|X_t = N) \end{pmatrix} \tag{10}$$

In the denominator of Equation 5, $\mathbf{1}^T$ is a vector where each entry is 1 and is used to sum the unnormalized probabilities.

We can also apply this approach to continuous random variables by splitting up the infinitely large state space into a finite number of regions with a single probability value representing the cumulative posterior. Discretizing a continuous state space results in the **histogram filter**.

3 Kalman filter

Earlier, we saw the state transition model $p(x_t|x_{t-1}, u_t)$ and measurement model $p(z_t|x_t)$ represented as probability distributions. This is the Bayesian perspective. The transition and measurement models can also be viewed from a dynamical systems perspective where Equation (11) represent the dynamics model and Equation (12) the measurement model.

$$x_t = f(x_{t-1}, u_t) + w_{t-1} \tag{11}$$

$$z_t = h(x_t) + v_t \tag{12}$$

$f(\cdot)$ is the **dynamics** of the system that encapsulates how the state evolves over time from x_t to x_{t+1}. Similarly, $h(\cdot)$ is a function that maps the state x_t to the corresponding observation. w_t and v_t are the **process noise** (random disturbances in the system) and **measurement noise** (within the sensor), respectively, that arise naturally in dynamical systems. We assume that the dynamics and measurement model as well as the noise statistics D_w, D_v (such that $w \sim D_w, v \sim D_v$) are known. The dynamical system equations are consistent with the graphical model in Figure 1 and our assumption that the state is complete: x_t only depends on x_{t-1} and u_t, and z_t only depends on x_t.

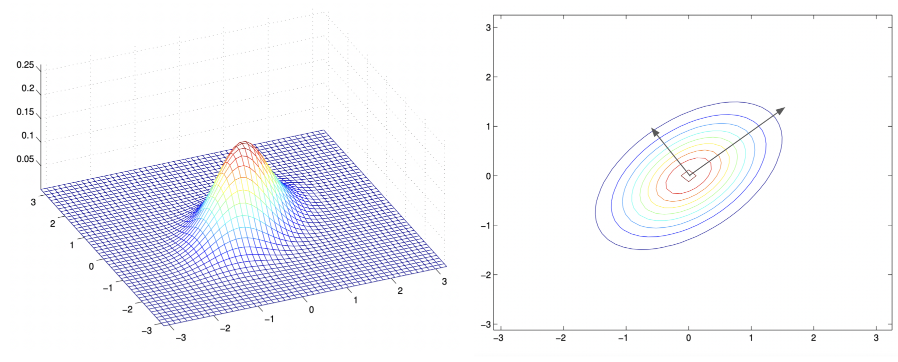
3.1 Gaussian distributions

If $X \in \mathbb{R}^N$ is a random vector sampled from a **multivariate Gaussian distribution** (also known as a normal distribution), we say that $X \sim \mathcal{N}(\mu, \Sigma)$, where $\mu \in \mathbb{R}^N$ is the **mean vector** and $\Sigma \in \mathbb{R}^{N \times N}$ is the positive semi-definite **covariance matrix** (analogous to the variance in the univariate case). The corresponding *probability density function* (PDF) is:

$$p(X) = p(\mu, \Sigma) = \frac{1}{\sqrt{(2\pi^N|\Sigma|}} \exp(-\frac{1}{2}(x-\mu)^T \Sigma^{-1}(x-\mu)) \qquad (13)$$

Figure (5) illustrates both the physical shape of the distribution and level sets, where the probability is constant along each ellipse.

Figure 5: The multivariate Gaussian shape on the left [2], and the contour ellipses on the right.



We see that Σ indicates the "spread" of the distribution in each direction; the eigenvectors of Σ are the axis directions and the scaling is based on the eigenvalues. Large eigenvalues indicate a high degree of uncertainty in the direction of the corresponding eigenvector, and vice versa for small eigenvalues. Not only does the covariance matrix represent the distribution confidence, it also indirectly captures the correlation between different components of the state, which is useful for estimation. For more details on multivariate

Gaussians, please refer to the CS 229 notes [2, 1] and Berkeley CS 189 notes [3].

The **Kalman filter** is the adaptation of the Bayes filter to **linear-Gaussian systems**. As the name implies, linear-Gaussian systems make two key assumptions. First, we assume that all random variables involved (e.g. state, measurements, posteriors, noise) are multivariate Gaussians. Second, we assume that all variables are linear in their parent variables; $f(\cdot)$ and $h(\cdot)$ are linear (in the form $y = Ax$). Equations (14) and (15) illustrate the linear-Gaussian system model.

$$x_t = A_t x_{t-1} + B_t u_t + w_t \tag{14}$$

$$z_t = C_t x_t + v_t \tag{15}$$

We assume the process noise and measurement noise are Gaussian white noise with zero mean, that is $w_t \sim \mathcal{N}(0, Q_t)$ and $v_t \sim \mathcal{N}(0, R_t)$, and for times t, τ such that $t \neq \tau$, $\text{Cov}(w_t, w_\tau) = 0$ and $\text{Cov}(v_t, v_\tau) = 0$. Our initial condition, or the prior, is represented as $x_0 \sim \mathcal{N}(\mu_{0|0}, \Sigma_{0|0})$. We further assume that $\text{Cov}(x_0, v_t) = 0, \text{Cov}(x_0, w_t) = 0$ for all t, and $\text{Cov}(w_t, v_\tau) = 0$ for all t, τ.

While the Kalman filter is limited to linear-Gaussian systems, it enables us to efficiently deal with continuous random variables and vectors (infinitely many outcomes). This is because as long as we have μ and Σ, we have the entire probability distribution. Thus, instead of predicting and updating the entire distribution $p(x)$ (for all possible x) as we did for the Bayes filter, we simply need to predict and update our mean vector and covariance matrix at each time step, which we can do with closed-form equations. Here, we only lay the Kalman filter equations and interpret them to explain the underlying intuition. There are numerous derivations online from the probabilistic perspective by applying the expectation definitions for mean and covariance, or from a optimization perspective as the best linear unbiased estimator in terms of mean-squared error.

Similar to the Bayes filter, the Kalman filter consists of a predict and update step.

In the predict step, we compute the mean and covariance of the predicted posterior $p(x_t|z_{1:t-1}, u_{1:t}) \sim \mathcal{N}(\mu_{t|t-1}, \Sigma_{t|t-1})$ from our previous posterior and knowledge of the system's process model:

$$\mu_{t|t-1} = A_t \mu_{t-1|t-1} + B_t u_t \tag{16}$$

$$\Sigma_{t|t-1} = A_{t-1} \Sigma_{t-1|t-1} A_{t-1}^T + Q_{t-1} \tag{17}$$

The previous posterior is $p(x_{t-1}|z_{1:t-1}, u_{1:t-1}) \sim \mathcal{N}(\mu_{t-1|t-1}, \Sigma_{t-1|t-1})$. To predict the mean $\mu_{t|t-1}$, we plug $\mu_{t-1|t-1}$ into Equation 14 to get Equation 16. The noise term w_t in Equation 14 has zero mean so it does not affect $\mu_{t|t-1}$. To derive the covariance prediction in Equation 17, we first make use of the fact that given a random variable x with covariance Σ, $\mathrm{Cov}(Ax) = A\Sigma A^T$. Then, to account for any possible disturbances as the system evolves over time, we add in the covariance Q of the process noise. Since all covariance matrices are positive definite, this summation leads to larger predicted covariance values representing increased uncertainty. Because we are making a prediction about the state at time t with data only up to $t-1$, we are now less confident about the state distribution.

In the update step, we refine the predicted mean and covariance with the new measurement z_t. The mean and covariance update equations are:

$$\mu_{t|t} = \mu_{t|t-1} + K_t(z_t - C_t\mu_{t|t-1}) \tag{18}$$

$$\Sigma_{t|t} = \Sigma_{t|t-1} - K_t C_t \Sigma_{t|t-1} \tag{19}$$

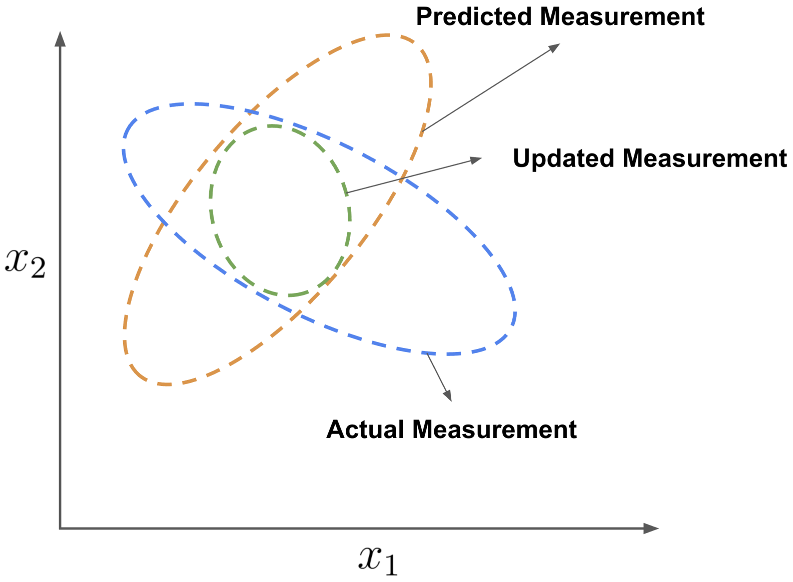
$$K_t = \Sigma_{t|t-1} C_t^T (C_t \Sigma_{t|t-1} C_t^T + R_t)^{-1} \tag{20}$$

We can see that the update step is trickier than the predict step from the equations alone. Starting with the mean update, z_t is our sensor measurement and $C_t\mu_{t|t-1}$ is our predicted measurement (if the state was what we predicted, this is the measurement we would have). Thus, the measurement residual, $z_t - C_t\mu_{t|t-1}$, also known as the **innovation**, represents how much our predicted measurement differs from the actual observation we obtained through sensors. The mean update therefore attempts to refine and reconcile our predicted mean $\mu_{t|t-1}$ with this measurement residual. This is done through the **Kalman Gain** K_t, which can be viewed as a "weighting factor" that informs us of how much we need to revise our estimate. Let's rewrite K_t as:

$$K_t = C_t^{-1} \frac{C_t \Sigma_{t|t-1} C_t^T}{C_t \Sigma_{t|t-1} C_t^T + R_t} \tag{21}$$

Let us analyze the fraction on the right hand side for two extreme cases. First, the case that R_t approaches zero in the limit means that we believe there is little measurement noise. The fraction approaches 1 and K is simply C_t^{-1}, and therefore $\mu_{t|t} = C_t^{-1} z_t$. No noise means we are confident that our measurement z_t is highly accurate, and we can simply invert our measurement model to obtain the true state. Furthermore, the covariance is updated to be $K_t = \Sigma_{t|t-1} - \Sigma_{t|t-1} = 0$, indicating our high confidence about the value of the state. The second case is that Σ_t approaches zero in the limit, which means that there is little process noise. The fraction approaches 0, $K = 0$, the updated mean is $\mu_{t|t} = \mu_{t|t-1}$ and the updated covariance is $\Sigma_{t|t} = \Sigma_{t|t-1}$. This means that because there are no disturbances in the system, the process model alone is sufficient to propagate the state across time so we do not need to incorporate the measurement at all (assuming we initialize the filter with a good prior). In practice, neither of these two cases ever occur; the filter attempts to update the mean and covariance to use both the actual measurement and the predicted measurement, as illustrated in Figure (6) for the case of a 2D state.

Figure 6: Kalman filter update from the distributions perspective.



The axes in Figure 6 are the components of the state; we are looking at the distributions from top-down. The ellipse in orange represents the distribution for our predicted measurement, given by $\mathcal{N}(C_t\mu_{t|t-1}, C_t\Sigma_{t|t-1}C_t^T)$. The ellipse in blue represents the distribution for the actual measurement, given by $\mathcal{N}(z_t, R_t)$. The Kalman filter multiplies both of these distributions to find the overlapping region under which x_t is likely for both distributions (recall $p(x_1, x_2) = p(x_1)p(x_2)$. This turns out to be our updated distribution, which is another Gaussian represented by the green ellipse, given by $\mathcal{N}(\mu_{t|t}, \Sigma_{t|t})$. Since we're taking the overlap, notice how the updated distribution is smaller compared to the two parent distributions. While the predict step increases the covariance (uncertainty), the update step uses the new measurement z_t to reduce the covariance. Thus, the covariance alternates between increasing and decreasing after the predict and update step. Overall, the covariance decreases until it converges.

4 Extended Kalman filter (EKF)

We mentioned earlier the Kalman filter holds for linear-Gaussian systems. Linear process and measurement models ensure that the predicted and updated distributions are also Gaussian.

Figure 7: Pushing a Gaussian distribution through a linear mapping [4].

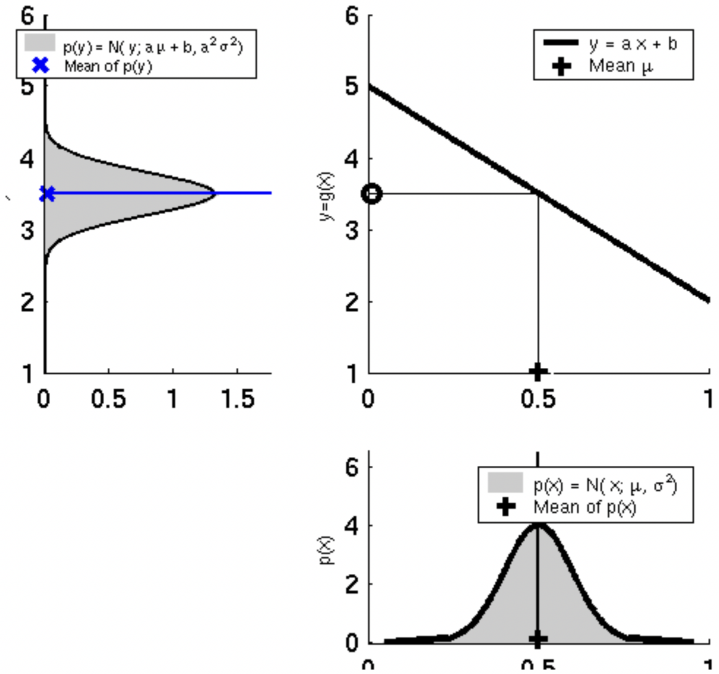
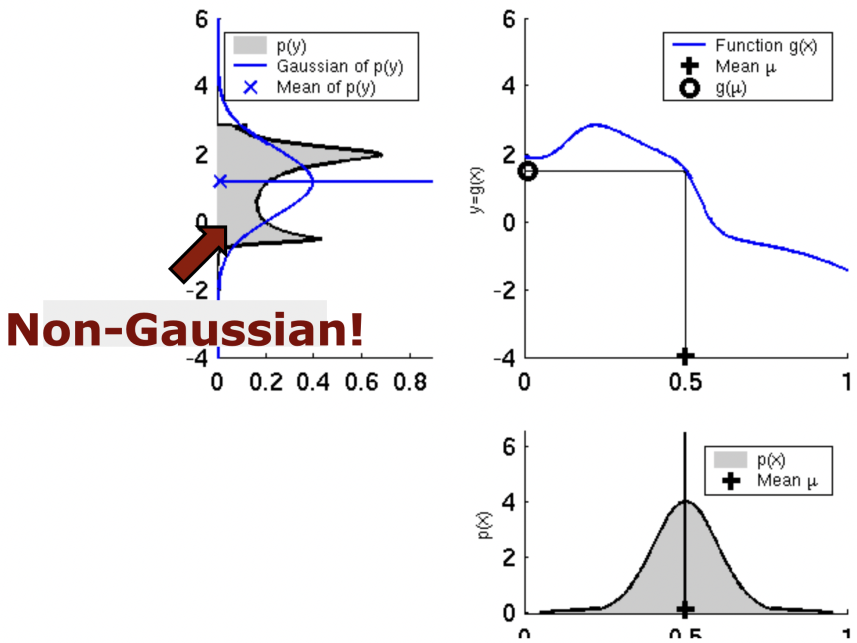


Figure (7) shows how applying a linear mapping to a random variable x still results in a Gaussian distribution after the transformation. On the other hand, Figure (8) shows how applying a non-linear mapping results in a non-Gaussian distribution and the previous predict and update equations cannot be applied anymore.

Figure 8: Pushing a Gaussian distribution through a non-linear mapping.



The **extended Kalman filter** handles non-linear dynamics and measurement functions by linearizing them about the input at each time t.

We do this with a first-order Taylor series expansion around $f(\mu_{t-1|t-1}, u_t)$ and $g(\mu_{t|t-1})$.

The predict step then becomes:

$$\mu_{t|t-1} = f(\mu_{t-1|t-1}, u_t) \tag{22}$$

$$\Sigma_{t|t-1} = A_{t-1}\Sigma_{t-1|t-1}A_{t-1}^T + Q_{t-1} \tag{23}$$

The update step becomes:

$$\mu_{t|t} = \mu_{t|t-1} + K_t(z_t - g(\mu_{t|t-1})) \tag{24}$$

$$\Sigma_{t|t} = \Sigma_{t|t-1} - K_t C_t \Sigma_{t|t-1} \tag{25}$$

$$K_t = \Sigma_{t|t-1}C_t^T(C_t\Sigma_{t|t-1}C_t^T + R_t)^{-1} \tag{26}$$

A_t and C_t are the Jacobians of the nonlinear dynamics and measurement model with respect to x:

$$A_t(\mu_{t-1|t-1}, u_t) = \frac{\partial f(x_t, u_t)}{\partial x_t}|_{x_t=\mu_{t-1|t-1}} \tag{27}$$

$$C_t(\mu_{t|t-1}, u_t) = \frac{\partial g(x_t, u_t)}{\partial x_t}|_{x_t=\mu_{t|t-1}} \tag{28}$$

5 Acknowledgements

References

[1] Chuong B Do. More on multivariate gaussians. *URL http://cs229. stanford. edu/section/more_on_gaussians. pdf.[Online]*, 2008.

[2] Chuong B Do. The multivariate gaussian distribution. *Section Notes, Lecture on Machine Learning, CS*, 229, 2008.

[3] Jonathan R. Shewchuk. Eigenvectors and the anisotropic multivariate normal distribution. *Lecture Notes, CS*, 189, 2019.

[4] Cyrill Stachniss. Extended kalman filter. `http://ais.informatik.uni-freiburg.de/teaching/ws13/mapping/pdf/slam04-ekf.pdf`, 2013. Accessed: 2022-01-31.