The Unscented Kalman Filter

SCJ Robertson

14th July, 2016

Introduction

The purpose of this document is to derive the Unscented Kalman Filter using both Belief Update and Belief Propagation in a clique tree. Two different tree topologies are explored which support the different propagation techniques, a standard tree and a simpler chain.

1. Representation

This section is concerned with establishing the ground rules necessary to derive the Unscented Kalman Filter using a clique tree. We will make use of both of Belief Update Propagation and the now very familiar Belief Propagation, using two different tree topologies. In both cases we no longer assign pre-defined distributions to cliques, rather we will dynamically create the required joint distributions using the first available information.

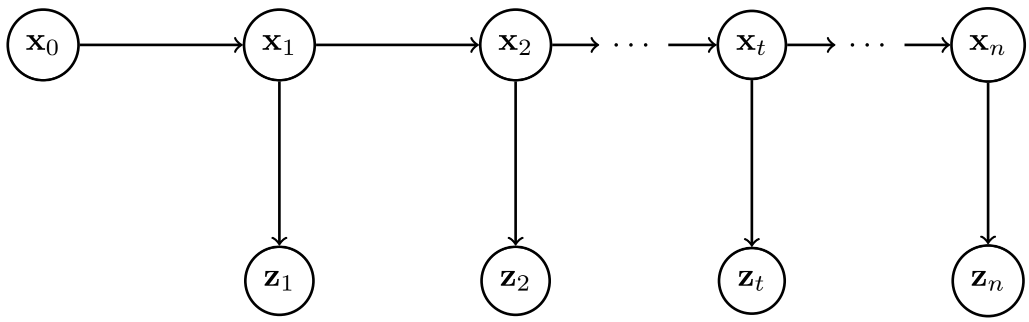


Figure 1.1: The Bayes Net for a Kalman Filter. The control vector, $\boldsymbol{u}_t$, has been omitted as it is deterministic.

1.1 The Kalman Filter for Non-linear Systems

We now extend the Kalman Filter, relaxing the assumptions of linearity, allowing the distributions to be governed by arbitrary non-linear functions.

1. The state transition probability $p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1})$ is now determine by some function, with added Gaussian noise.

$$\boldsymbol{x}_t = \boldsymbol{g}(\boldsymbol{x}_{t-1}, \boldsymbol{u}_t) + \boldsymbol{\epsilon}_t \tag{1.1}$$

   $\boldsymbol{g}(\cdot)$ is arbitrarily complex, but returns an $n \times 1$ vector. $\boldsymbol{\epsilon}_t$ is a Gaussian random vector which accounts for the uncertainty introduced during the state transition, it is zero mean with a covariance matrix R.

2. The measurement probability $p(\boldsymbol{z}_t|\boldsymbol{x}_t)$ is also governed by some arbitrary function, with added Gaussian noise:

$$\boldsymbol{z}_t = \boldsymbol{h}(\boldsymbol{x}_t) + \boldsymbol{\delta}_t \tag{1.2}$$

   $\boldsymbol{h}(\cdot)$ returns a $k \times 1$ vector. $\boldsymbol{\delta}_t$ is also zero mean Gaussian noise with a covariance matrix Q.

3. The belief about the initial state, $p(\boldsymbol{x}_0)$ is normally distributed:

$$p(\boldsymbol{x}_0) = \mathcal{N}(\boldsymbol{x}_0|\boldsymbol{\mu}_0, \Sigma_0) \tag{1.3}$$

Equations 1.1 and 1.2 assume a clean transition through the process and observation models, after which the results are then sprinkled with noise. This is less general than augmenting the state vector, $\boldsymbol{x}_t$, with the noise and pushing it through the transformation, but it is consistent with [13, 14], works, and is quite a bit easier.

1.2 PGM Representation

We can transform Figure 1.1 into a clique tree by applying the Junction Tree Algorithm. The algorithm determines the tree's topology by finding a maximal spanning tree weighted on the cardinality of legal sepsets [3, 4]. The solution is not unique and multiple topologies exist. We will choose the configuration given in Figure 1.2 as it directly corresponds with the Bayes Net of Figure 1.1 and is therefore the most intuitive. It does not support Kalman Filter using standard Belief Propagation (BP) and we must use Belief Update Propagation (BUP) as it allows for a more flexible message schedule.

Section 3 provides an alternate topology, in the form of a chain, which supports left-to-right BP.

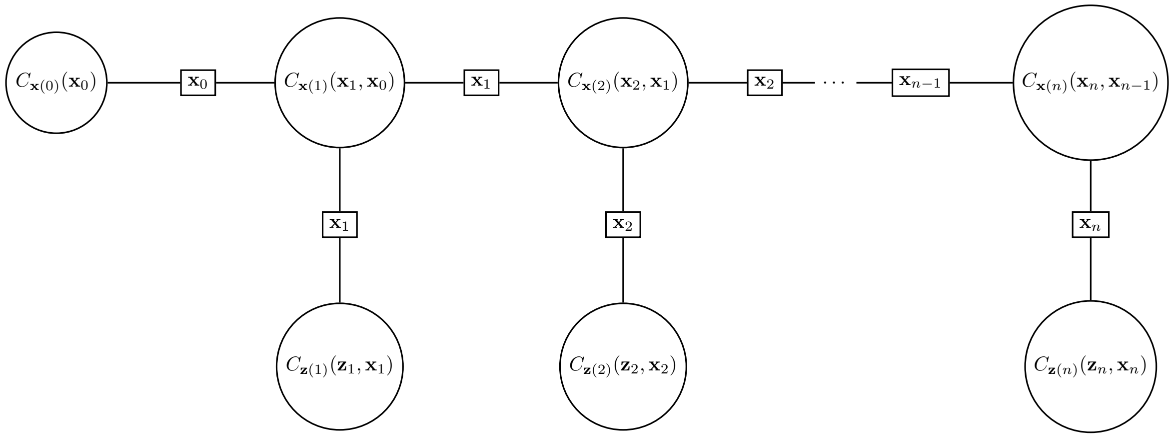


Figure 1.2: A clique tree for a Kalman Filter. The transition clique $C_{\boldsymbol{x}(t)}(\boldsymbol{x}_t, \boldsymbol{x}_{t-1})$ has been assigned $p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1})$, the measurement clique has been assigned $p(\boldsymbol{z}_t|\boldsymbol{x}_t)$.

1.2.1 Belief Update Propagation

The topology of the clique tree in Figure 1.2 and our choice of initial clique distributions (Section 1.2.2) do not support BP. We will use BUP as an alternate message passing scheme. It is also an exact inference technique in clique trees and under most conditions equivalent to BP, but it allows for incremental updates using a more flexible message passing schedule [9, 10].

Belief Update (BU) is typically invoked when answering queries on a clique, when we wish to have full knowledge about variables in the clique's scope. The clique belief is the belief held over a clique i after receiving all possible information from the outside system (Figure 1.3), it is represented as a product of the incoming messages and the initial distribution:

$$\beta_i = \Psi_i \prod_k \delta_{k \to i} \tag{1.4}$$

After completing BU, the clique cannot pass any messages to its neighbours without repeating known information back to them. It is necessary to remove the received information, $\delta_{j \to i}$, before passing the message $\delta_{i \to j}$ to a clique j. The information is removed by dividing out the received message:

$$\begin{aligned}
\tilde{\delta}_{i \to j} &= \frac{\delta_{i \to j}}{\delta_{j \to i}} \\
&= \frac{1}{\delta_{j \to i}} \int \Psi_i \prod_k \delta_{k \to i} d\{\boldsymbol{C}_i - \boldsymbol{S}_{i,j}\} \\
&= \int \Psi_i \prod_{k \neq j} \delta_{k \to i} d\{\boldsymbol{C}_i - \boldsymbol{S}_{i,j}\}
\end{aligned} \tag{1.5}$$

As we primarily work with canonical Gaussians, division is always well defined [11]. Equation 1.5 is equivalent to standard BP, showing that BU delivers the same information in clique trees [9].

Algorithm 1 provides the message passing algorithm used for the remainder of this document. As it can be seen BU is first invoked on every clique in the tree, after which we resort to incremental updates between neighbouring cliques. New information replaces old information by swapping out the existing information's corresponding factor. This allows us to use an incredibly flexible schedule, passing messages only when new information becomes available. As all cliques always hold a full clique belief based on the latest available information, multiple queries can be answered at once [9].

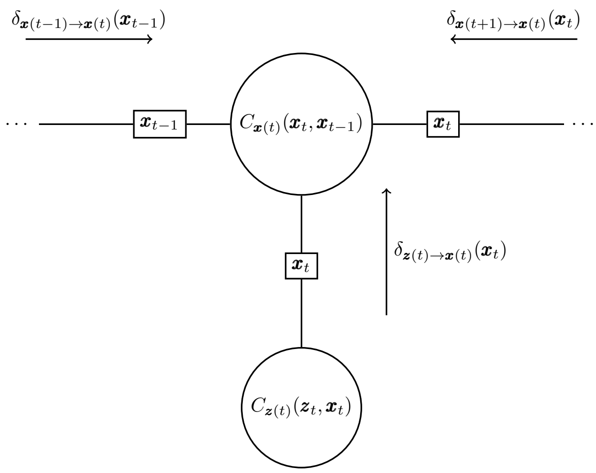


Figure 1.3: Belief Update is invoked on a transition clique in Kalman Filter. The clique belief is the belief held over the clique after it has received all possible information from the outside system.

Algorithm 1 Belief Update Propagation in a clique tree.

$\beta_i = \Psi_i \prod_k \delta_{k\to i}$ The clique belief over the transmitting clique i.
$\beta_j = \Psi_j \prod_k \delta_{k\to j}$ The clique belief over the receiving clique j.

function BELIEF-UPDATE(β_i, β_j)

$\tilde{\sigma}_{i\to j} = \int \beta_i d\{\boldsymbol{C}_i - \boldsymbol{S}_{i,j}\}$ ▷ Marginalise clique i onto the sepset.
$\tilde{\delta}_{i\to j} = \frac{\tilde{\sigma}_{i\to j}}{\delta_{j\to i}}$ ▷ Remove repeated information.
$\beta_j \leftarrow \beta_j \frac{\tilde{\delta}_{i\to j}}{\delta_{j\to i}}$ ▷ Update β_j, replacing $\delta_{j\to i}$ with $\tilde{\delta}_{j\to i}$.

return β_j

1.2.2 Initial Clique Distributions

The Junction Tree Algorithm assigns an initial product of distributions to a clique which satisfy its scope of variables. As the clique tree of Figure 1.2 corresponds directly to its Bayes Net, we assigned the transition distribution to $C_{\boldsymbol{x}(t)}(\boldsymbol{x}_t, \boldsymbol{x}_{t-1})$ and the measurement distribution to $C_{\boldsymbol{x}(t)}(\boldsymbol{x}_t, \boldsymbol{x}_{t-1})$. In the non-linear case, these are token assignments and each clique is given a vacuous distribution,

$$\Psi_j(\boldsymbol{y}, \boldsymbol{x}) = \mathcal{C}(\boldsymbol{y}, \boldsymbol{x}; 0, \mathbf{0}, 0) \tag{1.6}$$

satisfying only its scope requirements. We no longer have a predefined linear conditional Guassian which can be rewritten as joint distribution, we accept that we initially cannot represent the distribution and we know nothing.

As an aside, a vacuous canonical form is mathematically well defined and is in a sense similar to a uniform distribution. It is continuous everywhere with a unity amplitude, does not have a finite area and therefore cannot represent a probability density function. It does not have legal marginals, but intuitively, if the joint distribution contains no information any outgoing message must also be vacuous.

Since we do not assign anything useful to the initial distributions, we will create the joint distribution on the fly using the first meaningful information the uninformed clique receives. In the BUP framework, we assume that belief update is initially invoked on every clique, resulting in a clique belief:

$$\begin{aligned} \beta_j &= \Psi_j(\boldsymbol{y}, \boldsymbol{x}) \prod_k \delta_{k \to j} \\ &= \Psi_j(\boldsymbol{y}, \boldsymbol{x}) \delta_{i \to j}(\boldsymbol{x}) \end{aligned} \tag{1.7}$$

During filtering only the past events propagate any information and the initial clique belief is simply:

$$\delta_{i \to j}(\boldsymbol{x}) = \mathcal{C}(\boldsymbol{x}; \Sigma^{-1}_{\boldsymbol{x},\boldsymbol{x}}, \boldsymbol{h}_{\boldsymbol{x}}, g_{\boldsymbol{x}}) \tag{1.8}$$

We use the only available information to create a joint Gaussian in $\boldsymbol{x}$ and $\boldsymbol{y}$. $\boldsymbol{y}$ is result of $\boldsymbol{x}$ being advanced through the system, therefore $\boldsymbol{y}$ is function of $\boldsymbol{x}$. $\boldsymbol{x}$ is ensured Guassian, but $\boldsymbol{f}(\cdot)$ is non-linear and $\boldsymbol{f}(\boldsymbol{x})$ will be arbitrarily complex. We will approximate the resulting distribution with a Guassian, matching the distributions first two moments. First we determine the mean of $\boldsymbol{y}$:

$$\boldsymbol{\mu}_{\boldsymbol{y}} = \mathbb{E}\left[\boldsymbol{f}(\boldsymbol{x})\right] \tag{1.9}$$

After which we determine its covariance:

$$\Sigma_{\boldsymbol{y},\boldsymbol{y}} = \mathbb{E}\left[(\boldsymbol{f}(\boldsymbol{x}) - \boldsymbol{\mu}_{\boldsymbol{y}})(\boldsymbol{f}(\boldsymbol{x}) - \boldsymbol{\mu}_{\boldsymbol{y}})^T\right] + \Sigma_{\boldsymbol{n},\boldsymbol{n}} \tag{1.10}$$

Here $\Sigma_{\boldsymbol{n},\boldsymbol{n}}$ is the noise covariance matrix used as perturbation to ensure $\Sigma_{\boldsymbol{y},\boldsymbol{y}}$ is non-singular. We can make no guarantees on $\boldsymbol{f}(\cdot)$'s behaviour, in the worst case it could map all of $\boldsymbol{x}$ to a single point[1].

Most importantly, we must determine the cross-covariance of $\boldsymbol{x}$ and $\boldsymbol{y}$. It is critical to fully describe the variables' dependencies, without it our operations, such as marginalisation, will yield incorrect results.

$$\Sigma_{\boldsymbol{y},\boldsymbol{x}} = \mathbb{E}\left[(\boldsymbol{f}(\boldsymbol{x}) - \boldsymbol{\mu}_{\boldsymbol{y}})(\boldsymbol{x} - \boldsymbol{\mu}_{\boldsymbol{x}})^T\right] \tag{1.11}$$

$$\Sigma_{\boldsymbol{x},\boldsymbol{y}} = \Sigma^T_{\boldsymbol{y},\boldsymbol{x}} \tag{1.12}$$

[1] Which is pretty damn singular.

We can now construct a well-defined joint Gaussian. Σ is guaranteed non-singular and our standard operations are now well defined. The clique is now capable of propagating information.

$$\boldsymbol{C}_{\boldsymbol{y}}(\boldsymbol{y}, \boldsymbol{x}) = \mathcal{C}(\boldsymbol{y}, \boldsymbol{x}; \Sigma^{-1}, \Sigma^{-1}\boldsymbol{\mu}, g) \tag{1.13}$$

$$\Sigma = \begin{bmatrix} \Sigma_{\boldsymbol{y},\boldsymbol{y}} & \Sigma_{\boldsymbol{y},\boldsymbol{x}} \\ \Sigma_{\boldsymbol{x},\boldsymbol{y}} & \Sigma_{\boldsymbol{x},\boldsymbol{x}} \end{bmatrix} \tag{1.14}$$

$$\boldsymbol{\mu} = \begin{bmatrix} \boldsymbol{\mu}_{\boldsymbol{y}} \\ \boldsymbol{\mu}_{\boldsymbol{x}} \end{bmatrix} \tag{1.15}$$

1.3 The Unscented Transform

In the previous section we described a general approach to dynamically constructing joint distributions. The crux of this operation is determining the expected value of a function of a random variable, namely:

$$\mathbb{E}\left[\boldsymbol{g}\left(\boldsymbol{x}\right)\right] = \int_{\boldsymbol{x}} \boldsymbol{g}\left(\boldsymbol{x}\right) f\left(\boldsymbol{x}\right) d\boldsymbol{x} \tag{1.16}$$

For linear function the integral above will have well defined solutions and slapping a Gaussian on the resulting distribution will be exact. However, for most non-linear functions we will have to resort to numerical integration. Equation 1.16 has a form which allows it to be approximated by Gaussian quadrature as follows [5, 12, 15]:

$$\int_{-\infty}^{\infty} \boldsymbol{g}\left(\boldsymbol{x}\right) f\left(\boldsymbol{x}\right) d\boldsymbol{x} \approx \int_{-a}^{a} \boldsymbol{g}\left(\boldsymbol{x}\right) f\left(\boldsymbol{x}\right) d\boldsymbol{x} \approx \sum_{i=1}^{N} \mathcal{W}_i \boldsymbol{g}(\mathcal{X}_i) \tag{1.17}$$

Here $\mathcal{W}_i$ are predefined weights and $\mathcal{X}_i$ are points, or knots, specially chosen from $f(\boldsymbol{x})$ [12]. The Unscented Transform provides an accurate method of approximating the integral, drawing the knots using a specific, deterministic algorithm.

The Unscented Transform samples a set of $2n + 1$ Sigma-Points from $f(\boldsymbol{x})$'s covariance matrix, Σ, and passes them through $\boldsymbol{g}(\cdot)$. The Sigma-Points are given as:

$$\boldsymbol{\mathcal{X}}^0 = \boldsymbol{\mu} \tag{1.18}$$

$$\boldsymbol{\mathcal{X}}^i = \boldsymbol{\mu} + \left(\sqrt{(n+\lambda)\Sigma}\right)_i \quad \text{for } i = 1, \ldots, n \tag{1.19}$$

$$\boldsymbol{\mathcal{X}}^i = \boldsymbol{\mu} - \left(\sqrt{(n+\lambda)\Sigma}\right)_{i-n} \quad \text{for } i = n+1, \ldots, 2n \tag{1.20}$$

These are scaled points, symmetrically distributed on and around the mean of $f(\boldsymbol{x})$. The choice of matrix square root is arbitrary, but we will use the following definition as it consistent with [6–8]:

$$\Sigma = \sqrt{\Sigma}\sqrt{\Sigma}^T \tag{1.21}$$

All matrix square roots consistent with this definition will merely be rotations of one another [8]. From this definition, the Sigma-Points will be drawn as vectors from the rows of $\sqrt{\Sigma}$. It is typical to use Cholesky decomposition as it is numerically stable and efficient, however this will not be given any further consideration here. The following notation, consistent with [14], will be used to show the selection of Sigma-Points from a distribution:

$$\boldsymbol{\mathcal{X}} = \left(\boldsymbol{\mu}, \boldsymbol{\mu} + \gamma\sqrt{\Sigma}, \boldsymbol{\mu} - \gamma\sqrt{\Sigma}\right) \tag{1.22}$$

We forgo the scaling parameters and just use:

$$\gamma = \sqrt{n + \lambda} \tag{1.23}$$

The selection of scaling parameters is probably an art unto itself and won't be discussed here[2]; however, it usually suggested that $n + \lambda > 0$ and that $n + \lambda = 3$ is particularly useful for multidimensional distributions [6–8, 14, 15].

The weights, $\mathcal{W}_i$, are defined as follows:

$$\mathcal{W}_0 = \frac{\lambda}{n+\lambda} \tag{1.24}$$

$$\mathcal{W}_i = \frac{1}{2(n+\lambda)} \tag{1.25}$$

$$\sum_{i=0}^{2n} \mathcal{W}_i = 1 \tag{1.26}$$

Some sources include additional scaling for the mean's weight [14], but we will once again neglect them. In general, the weights and Sigma-Points are chosen such that they fully capture the original distribution's mean and covariance. Therefore the Sigma-Points can be viewed as the observed events of the some underlying discrete distribution and the weights their probability of occurrence. The mean of this new distribution must be equivalent to the original distribution:

$$\begin{aligned}
\hat{\boldsymbol{\mu}} &= \sum_{i=0}^{2n} \mathcal{W}_i \boldsymbol{\mathcal{X}}^i \\
&= \frac{\lambda}{n+\lambda}\boldsymbol{\mu} + \frac{1}{2(n+\lambda)} \sum_{i=1}^{2n} \boldsymbol{\mathcal{X}}^i \\
&= \frac{\lambda}{n+\lambda}\boldsymbol{\mu} + \frac{1}{2(n+\lambda)} \left[\sum_{i=1}^{n} \left(\boldsymbol{\mu} + \left(\sqrt{(n+\lambda)\Sigma} \right)_i \right) + \sum_{i=1}^{n} \left(\boldsymbol{\mu} - \left(\sqrt{(n+\lambda)\Sigma} \right)_i \right) \right] \\
&= \frac{\lambda}{n+\lambda}\boldsymbol{\mu} + \frac{1}{2(n+\lambda)} \left[2 \sum_{i=1}^{n} \boldsymbol{\mu} \right] \\
&= \frac{\lambda}{n+\lambda}\boldsymbol{\mu} + \frac{n}{n+\lambda}\boldsymbol{\mu} \\
&= \boldsymbol{\mu}
\end{aligned}$$

The covariance must also be preserved:

$$\begin{aligned}
\hat{\Sigma} &= \sum_{i=0}^{2n} \mathcal{W}_i \left[\boldsymbol{\mathcal{X}}^i - \boldsymbol{\mu} \right] \left[\boldsymbol{\mathcal{X}}^i - \boldsymbol{\mu} \right]^T \\
&= \frac{1}{n+\lambda} \sum_{i=1}^{n} \left[\left(\sqrt{(n+\lambda)\Sigma} \right)_i \right] \left[\left(\sqrt{(n+\lambda)\Sigma} \right)_i \right]^T \\
&= \sum_{i=0}^{n} \sqrt{\Sigma_i} \sqrt{\Sigma_i}^T \\
&= \Sigma
\end{aligned}$$

The Sigma-Points are now passed through $\boldsymbol{g}(\cdot)$ and the mean and covariance reassembled

[2] For obvious reasons; I don't know anything about it.

on the other side, we have completed our goal of numerically approximating Equation 1.16.

$$\overline{\boldsymbol{\mathcal{X}}}^i = \boldsymbol{g}(\boldsymbol{\mathcal{X}}_i) \tag{1.27}$$

$$\overline{\boldsymbol{\mu}} = \sum_{i=0}^{2n} \mathcal{W}_i \overline{\boldsymbol{\mathcal{X}}}^i \tag{1.28}$$

$$\overline{\Sigma} = \sum_{i=0}^{2n} \mathcal{W}_i \left[\overline{\boldsymbol{\mathcal{X}}}^i - \overline{\boldsymbol{\mu}}\right] \left[\overline{\boldsymbol{\mathcal{X}}}^i - \overline{\boldsymbol{\mu}}\right]^T \tag{1.29}$$

The set of vectors $\boldsymbol{\mathcal{X}}_i$ are linearly independent; however, they may not retain this independence once passed through $\boldsymbol{g}(\cdot)$. Therefore Σ is not guaranteed to have full rank and we can see why it is necessary to perturb it with the noise covariance matrix.

2. Derivation

We have finally reached the point where we can derive the UKF. First we will derive the UKF for a general system. We will then derive the linear Kalman Filter using the same approach, proving it is just a special case of the UKF. Both filters are given in their full glory in Algorithms 2 and 3, the notation used in both derivations will be consistent with [13, 14].

The approach will make use of the following BU message schedule:

1. Prediction (Figure 2.1): The transition clique is populated using the incoming message's information.

2. Measurement Clique Construction (Figure 2.2a): The process model's prediction distribution is used to create the measurement clique by passing it a message.

3. Measurement Update (Figure 2.2b): The observation evidence is introduced, the measurement clique collapses and is absorbed into the transition clique.

4. Forward Propagation (Figure 2.3): The evidence is incorporated into the prediction and the message is propagated outwards to the next transition clique.

The last two steps are lumped together for reasons which will become clear in Section 3.

2.1 Non-linear Systems

This section just handles the general non-linear case we have assumed so far in this document. When fully assembled the clique tree operates exactly as Algorithm 2 does, with a few additional steps.

2.1.1 Prediction: Transition Clique Construction

As seen in Figure 2.1, we begin with some incoming Gaussian message represented in canonical form:

$$\tilde{\delta}_{\boldsymbol{x}(t-1)\to\boldsymbol{x}(t)}(\boldsymbol{x}_{t-1}) = \mathcal{C}(\boldsymbol{x}_{t-1}; \Sigma_{t-1}^{-1}, \boldsymbol{h}_{t-1}, g_{t-1}) \tag{2.1}$$

Since we are using BUP, the transition clique already holds the following belief and we merely incorporate the new information:

$$\begin{aligned}\beta_{\boldsymbol{x}(t)}(\boldsymbol{x}_t, \boldsymbol{x}_{t-1}) &= \frac{\tilde{\delta}_{\boldsymbol{x}(t-1)\to\boldsymbol{x}(t)}(\boldsymbol{x}_{t-1})}{\delta_{\boldsymbol{x}(t-1)\to\boldsymbol{x}(t)}(\boldsymbol{x}_{t-1})}\delta_{\boldsymbol{x}(t-1)\to\boldsymbol{x}(t)}(\boldsymbol{x}_{t-1})\delta_{\boldsymbol{x}(t+1)\to\boldsymbol{x}(t)}(\boldsymbol{x}_t)\delta_{\boldsymbol{z}(t)\to\boldsymbol{x}(t)}(\boldsymbol{x}_t)\Psi_{\boldsymbol{x}(t)}(\boldsymbol{x}_t, \boldsymbol{x}_{t-1}) \\ &= \tilde{\delta}_{\boldsymbol{x}(t-1)\to\boldsymbol{x}(t)}(\boldsymbol{x}_{t-1})\Psi_{\boldsymbol{x}(t)}(\boldsymbol{x}_t, \boldsymbol{x}_{t-1})\end{aligned} \tag{2.2}$$

Algorithm 2 The Unscented Kalman Filter.

function UNSCENTED-KALMAN-FILTER($\boldsymbol{\mu}_{t-1}$, Σ_{t-1}, $\boldsymbol{u}_t$, $\boldsymbol{z}_t^e$)

$$\boldsymbol{\mathcal{X}}_{t-1} = \left(\boldsymbol{\mu}_{t-1}, \boldsymbol{\mu}_{t-1} + \gamma\sqrt{\Sigma_{t-1}}, \boldsymbol{\mu}_{t-1} - \gamma\sqrt{\Sigma_{t-1}}\right)$$

$$\overline{\boldsymbol{\mathcal{X}}}_{t-1} = \boldsymbol{g}\left(\boldsymbol{\mathcal{X}}_{t-1}, \boldsymbol{u}_t\right)$$

$$\overline{\boldsymbol{\mu}}_{t-1} = \sum_{i=0}^{2n} \mathcal{W}_i \overline{\boldsymbol{\mathcal{X}}}_{t-1}^i$$

$$\overline{\Sigma}_t = \sum_{i=0}^{2n} \mathcal{W}_i \left[\overline{\boldsymbol{\mathcal{X}}}_{t-1}^i - \overline{\boldsymbol{\mu}}_{t-1}\right] \left[\overline{\boldsymbol{\mathcal{X}}}_{t-1}^i - \overline{\boldsymbol{\mu}}_{t-1}\right]^T + R$$

$$\hat{\boldsymbol{\mathcal{X}}}_t = \left(\overline{\boldsymbol{\mu}}_t, \overline{\boldsymbol{\mu}}_t + \gamma\sqrt{\overline{\Sigma}_t}, \overline{\boldsymbol{\mu}}_t - \gamma\sqrt{\overline{\Sigma}_t}\right)$$

$$\overline{\boldsymbol{\mathcal{Z}}}_t = \boldsymbol{h}\left(\hat{\boldsymbol{\mathcal{X}}}_t\right)$$

$$\overline{\boldsymbol{z}}_t = \sum_{i=0}^{2n} \mathcal{W}_i \overline{\boldsymbol{\mathcal{Z}}}_t^i$$

$$\overline{\mathcal{S}}_t = \sum_{i=0}^{2n} \mathcal{W}_i \left[\overline{\boldsymbol{\mathcal{Z}}}_t^i - \overline{\boldsymbol{z}}_t\right] \left[\overline{\boldsymbol{\mathcal{Z}}}_t^i - \overline{\boldsymbol{z}}_t\right]^T + Q$$

$$\overline{\Sigma}_{\boldsymbol{x},\boldsymbol{z}} = \sum_{i=0}^{2n} \mathcal{W}_i \left[\hat{\boldsymbol{\mathcal{X}}}_t^i - \overline{\boldsymbol{\mu}}_t\right] \left[\overline{\boldsymbol{\mathcal{Z}}}_t^i - \overline{\boldsymbol{z}}_t\right]^T$$

$$K_t = \overline{\Sigma}_{\boldsymbol{x},\boldsymbol{z}} \mathcal{S}_t^{-1}$$

$$\boldsymbol{\mu}_t = \overline{\boldsymbol{\mu}}_t + K_t\left(\boldsymbol{z}_t^e - \overline{\boldsymbol{z}}_t\right)$$

$$\Sigma_t = \overline{\Sigma}_t - K_t \mathcal{S}_t K_t^T$$

return $\boldsymbol{\mu}_t$, Σ_t

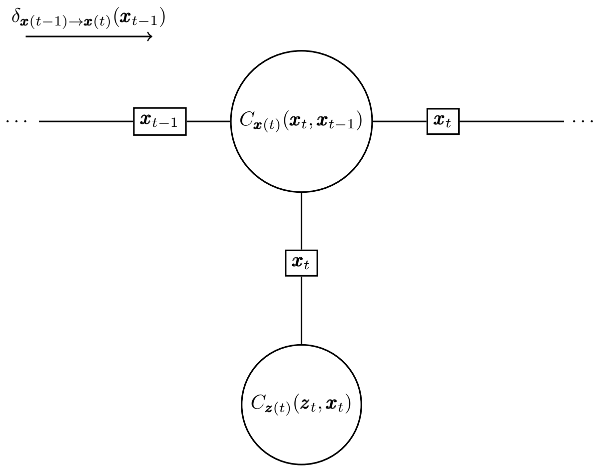


Figure 2.1: The incoming message to the transition clique. This information is used to populate the currently vacuous clique by creating a joint Gaussian, as described in Section 1.2.2.

The belief is reduced as only the incoming message contains any information, current and future events are naturally vacuous.

We now begin the arduous journey of populating the clique by creating the joint Gaussian density. First we draw the Sigma-Points from the incoming distribution and then we pass them through the process model:

$$\boldsymbol{\mathcal{X}}_{t-1} = \left(\boldsymbol{\mu}_{t-1}, \boldsymbol{\mu}_{t-1} + \gamma\sqrt{\Sigma_{t-1}}, \boldsymbol{\mu}_{t-1} - \gamma\sqrt{\Sigma_{t-1}}\right) \tag{2.3}$$

$$\overline{\boldsymbol{\mathcal{X}}}_{t-1} = \boldsymbol{g}\left(\boldsymbol{\mathcal{X}}_{t-1}, \boldsymbol{u}_t\right) \tag{2.4}$$

Using the transformed Sigma-Points, we can determine the mean of the predicted state $\boldsymbol{x}_t$:

$$\overline{\boldsymbol{\mu}}_t = \sum_{i=0}^{2n} \mathcal{W}_i \overline{\boldsymbol{\mathcal{X}}}_{t-1}^i \tag{2.5}$$

$\boldsymbol{x}_t$'s covariance matrix can now be constructed, perturbed with the process noise's covariance matrix R:

$$\begin{aligned}\overline{\Sigma}_t &= \mathbb{E}\left[(\boldsymbol{g}(\boldsymbol{x}_{t-1}, \boldsymbol{u}_t) - \overline{\boldsymbol{\mu}}_t)(\boldsymbol{g}(\boldsymbol{x}_{t-1}, \boldsymbol{u}_t) - \overline{\boldsymbol{\mu}}_t)^T\right] + R \\ &= \sum_{i=0}^{2n} \mathcal{W}_i \left[\overline{\boldsymbol{\mathcal{X}}}_{t-1}^i - \overline{\boldsymbol{\mu}}_{t-1}\right]\left[\overline{\boldsymbol{\mathcal{X}}}_{t-1}^i - \overline{\boldsymbol{\mu}}_{t-1}\right]^T + R\end{aligned} \tag{2.6}$$

We also determine the cross-covariance of $\boldsymbol{x}_{t-1}$ and $\boldsymbol{x}_t$.

$$\begin{aligned}\overline{\Sigma}_{\boldsymbol{x}_{t-1},\boldsymbol{x}_t} &= \mathbb{E}\left[(\boldsymbol{x}_{t-1} - \boldsymbol{\mu}_{t-1})(\boldsymbol{g}(\boldsymbol{x}_{t-1}, \boldsymbol{u}_t) - \overline{\boldsymbol{\mu}}_t)^T\right] \\ &= \sum_{i=0}^{2n} \mathcal{W}_i \left[\boldsymbol{\mathcal{X}}_t^i - \boldsymbol{\mu}_t\right]\left[\overline{\boldsymbol{\mathcal{X}}}_t^i - \overline{\boldsymbol{\mu}}_t\right]^T\end{aligned} \tag{2.7}$$

This will not be used directly in this section, but it is absolutely required as we will be representing the clique belief as a canonical Gaussian and require a full precision matrix.

$$\beta_{\boldsymbol{x}(t)}(\boldsymbol{x}_t, \boldsymbol{x}_{t-1}) = \mathcal{C}(\boldsymbol{x}_t, \boldsymbol{x}_{t-1}; \Sigma_{\boldsymbol{x}(t)}^{-1}, \Sigma_{\boldsymbol{x}(t)}^{-1}\boldsymbol{\mu}_{\boldsymbol{x}(t)}, g_{\boldsymbol{x}(t)}) \tag{2.8}$$

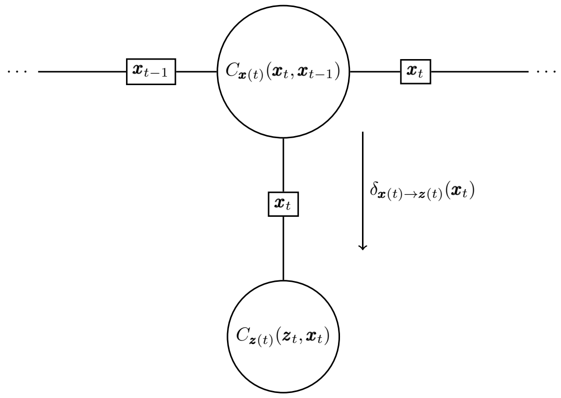
Here the joint covariance and mean are arranged as in Section 1.2.2.

$$\Sigma_{\boldsymbol{x}(t)} = \begin{bmatrix} \overline{\Sigma}_t & \overline{\Sigma}_{\boldsymbol{x}_{t-1},\boldsymbol{x}_t}^T \\ \overline{\Sigma}_{\boldsymbol{x}_{t-1},\boldsymbol{x}_t} & \Sigma_{t-1} \end{bmatrix} \tag{2.9}$$

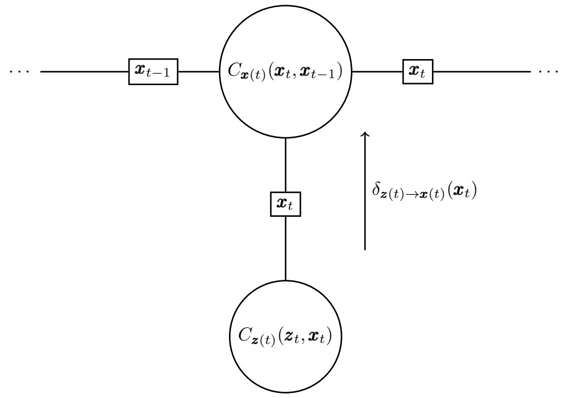
$$\boldsymbol{\mu}_{\boldsymbol{x}(t)} = \begin{bmatrix} \overline{\boldsymbol{\mu}}_t \\ \boldsymbol{\mu}_{t-1} \end{bmatrix} \tag{2.10}$$

We now must determine the outgoing message, to be used to construct the measurement clique. This is immensely simple as we just drop everything we don't need from the joint covariance and mean. The division here is also only included to be formal, the measurement clique is currently vacuous and offered no initial information.

$$\begin{aligned}\tilde{\delta}_{\boldsymbol{x}(t)\to\boldsymbol{z}(t)}(\boldsymbol{x}_t) &= \frac{1}{\delta_{\boldsymbol{z}(t)\to\boldsymbol{x}(t)}(\boldsymbol{x}_t)} \int \beta_{\boldsymbol{x}(t)}(\boldsymbol{x}_t, \boldsymbol{x}_{t-1}) d\boldsymbol{x}_{t-1} \\ &= \int \beta_{\boldsymbol{x}(t)}(\boldsymbol{x}_t, \boldsymbol{x}_{t-1}) d\boldsymbol{x}_{t-1} \\ &= \mathcal{C}(\boldsymbol{x}_t; \overline{\Sigma}_t^{-1}, \overline{\Sigma}_t^{-1}\overline{\boldsymbol{\mu}}_t, g_{\boldsymbol{x}(t)})\end{aligned} \tag{2.11}$$



(a) The predicted state vector's distribution is used to populate the vacuous joint distribution.



(b) The evidence is then introduced, the clique collapses and is absorbed into the transition clique.

Figure 2.2: The creation and subsequent destruction of the measurement clique. After the introduction of evidence the clique no longer has a unique scope and has no need to exist independently.

2.1.2 Measurement Clique Construction

As seen in Figure 2.2a, the incoming message of Equation 2.11 arrives and is incorporated into the current clique belief:

$$\begin{aligned}\beta_{\boldsymbol{z}(t)}(\boldsymbol{x}_t, \boldsymbol{z}_t) &= \frac{\tilde{\delta}_{\boldsymbol{x}(t)\to\boldsymbol{z}(t)}(\boldsymbol{x}_t)}{\delta_{\boldsymbol{x}(t)\to\boldsymbol{z}(t)}} \delta_{\boldsymbol{x}(t)\to\boldsymbol{z}(t)}(\boldsymbol{x}_t)\Psi_{\boldsymbol{z}(t)}(\boldsymbol{x}_t, \boldsymbol{z}_t) \\ &= \tilde{\delta}_{\boldsymbol{x}(t)\to\boldsymbol{z}(t)}(\boldsymbol{x}_t)\Psi_{\boldsymbol{z}(t)}(\boldsymbol{x}_t, \boldsymbol{z}_t) \end{aligned} \tag{2.12}$$

The incoming message again contains the only information, $\Psi_{\boldsymbol{z}(t)}(\boldsymbol{x}_t, \boldsymbol{z}_t)$ is kept just to satisfy the required scope.

We set about creating the joint Gaussian much as before, advancing $\boldsymbol{x}_t$ through the measurement model. We first draw the Sigma-Points from the message distribution and pass them through $\boldsymbol{h}(\cdot)$.

$$\hat{\boldsymbol{\mathcal{X}}}_t = \left(\overline{\boldsymbol{\mu}}_t, \overline{\boldsymbol{\mu}}_t + \gamma\sqrt{\overline{\Sigma}_t}, \overline{\boldsymbol{\mu}}_t - \gamma\sqrt{\overline{\Sigma}_t}\right) \tag{2.13}$$

$$\overline{\boldsymbol{\mathcal{Z}}}_t = \boldsymbol{h}\left(\hat{\boldsymbol{\mathcal{X}}}_t\right) \tag{2.14}$$

We now determine the mean of the measurement variable, $\boldsymbol{z}_t$:

$$\begin{aligned}\overline{\boldsymbol{z}}_t &= \mathbb{E}\left[\boldsymbol{h}(\boldsymbol{x}_{t-1})\right] \\ &= \sum_{i=0}^{2n} \mathcal{W}_i \overline{\boldsymbol{\mathcal{Z}}}_t^i \end{aligned} \tag{2.15}$$

Then we calculate its covariance, perturbed again by the measurement noise's covariance matrix Q:

$$\begin{aligned}\mathcal{S}_t &= \mathbb{E}\left[(\boldsymbol{h}(\boldsymbol{x}_{t-1}) - \overline{\boldsymbol{z}}_t)(\boldsymbol{h}(\boldsymbol{x}_{t-1}) - \overline{\boldsymbol{z}}_t)^T\right] + Q \\ &= \sum_{i=0}^{2n} \mathcal{W}_i \left[\overline{\boldsymbol{\mathcal{Z}}}_t^i - \overline{\boldsymbol{z}}_t\right]\left[\overline{\boldsymbol{\mathcal{Z}}}_t^i - \overline{\boldsymbol{z}}_t\right]^T + Q \end{aligned} \tag{2.16}$$

The variables' cross-covariance is now determined:

$$\begin{aligned}\overline{\Sigma}_{\boldsymbol{x},\boldsymbol{z}} &= \mathbb{E}\left[(\boldsymbol{x}_t - \overline{\boldsymbol{\mu}}_t)(\boldsymbol{h}(\boldsymbol{x}_{t-1}) - \overline{\boldsymbol{z}}_t)^T\right] \\ &= \sum_{i=0}^{2n} \mathcal{W}_i \left[\hat{\boldsymbol{\mathcal{X}}}_t^i - \overline{\boldsymbol{\mu}}_t\right]\left[\overline{\boldsymbol{\mathcal{Z}}}_t^i - \overline{\boldsymbol{z}}_t\right]^T \end{aligned} \tag{2.17}$$

We can now piece together the joint mean and covariance, marvel in their glory:

$$\Sigma_{\boldsymbol{z}(t)} = \begin{bmatrix} \overline{\Sigma}_t & \overline{\Sigma}_{\boldsymbol{x},\boldsymbol{z}} \\ \overline{\Sigma}_{\boldsymbol{x},\boldsymbol{z}}^T & \mathcal{S}_t \end{bmatrix} \tag{2.18}$$

$$\boldsymbol{\mu}_{\boldsymbol{z}(t)} = \begin{bmatrix} C\overline{\boldsymbol{\mu}}_t \\ \overline{\boldsymbol{\mu}}_t \end{bmatrix} \tag{2.19}$$

Unfortunately, we must work directly with the canonical form as it is still the most convenient method of introducing evidence. The current clique belief is:

$$\beta_{\boldsymbol{z}(t)}(\boldsymbol{x}_t, \boldsymbol{z}_t) = \mathcal{C}(\boldsymbol{x}_t, \boldsymbol{z}_t; \mathcal{P}_{\boldsymbol{z}(t)}, \boldsymbol{h}_{\boldsymbol{z}(t)}, g_{\boldsymbol{z}(t)}) \tag{2.20}$$

This makes use of the incredibly unwieldy precision matrix, $P_{\boldsymbol{z}(t)}$, and information vector, $\boldsymbol{h}_{\boldsymbol{z}(t)}$:

$$\begin{aligned}\mathcal{P}_{\boldsymbol{z}(t)} &= \Sigma_{\boldsymbol{z}(t)}^{-1} \\ &= \begin{bmatrix} \left(\overline{\Sigma}_t - \overline{\Sigma}_{\boldsymbol{x},\boldsymbol{z}}\mathcal{S}_t^{-1}\overline{\Sigma}_{\boldsymbol{x},\boldsymbol{z}}^T\right)^{-1} & -\left(\overline{\Sigma}_t - \overline{\Sigma}_{\boldsymbol{x},\boldsymbol{z}}\mathcal{S}_t^{-1}\overline{\Sigma}_{\boldsymbol{x},\boldsymbol{z}}^T\right)^{-1}\overline{\Sigma}_{\boldsymbol{x},\boldsymbol{z}}\mathcal{S}_t^{-1} \\ -\mathcal{S}_t^{-1}\overline{\Sigma}_{\boldsymbol{x},\boldsymbol{z}}^T\left(\overline{\Sigma}_t - \overline{\Sigma}_{\boldsymbol{x},\boldsymbol{z}}\mathcal{S}_t^{-1}\overline{\Sigma}_{\boldsymbol{x},\boldsymbol{z}}^T\right)^{-1} & \mathcal{S}_t^{-1} + \mathcal{S}_t^{-1}\overline{\Sigma}_{\boldsymbol{x},\boldsymbol{z}}^T\left(\overline{\Sigma}_t - \overline{\Sigma}_{\boldsymbol{x},\boldsymbol{z}}\mathcal{S}_t^{-1}\overline{\Sigma}_{\boldsymbol{x},\boldsymbol{z}}^T\right)^{-1}\overline{\Sigma}_{\boldsymbol{x},\boldsymbol{z}}\mathcal{S}_t^{-1} \end{bmatrix}\end{aligned} \tag{2.21}$$

$$\begin{aligned}\boldsymbol{h}_{\boldsymbol{z}(t)} &= \mathcal{P}_{\boldsymbol{z}(t)}\boldsymbol{\mu}_{\boldsymbol{z}(t)} \\ &= \begin{bmatrix} \left(\overline{\Sigma}_t - \overline{\Sigma}_{\boldsymbol{x},\boldsymbol{z}}\mathcal{S}_t^{-1}\overline{\Sigma}_{\boldsymbol{x},\boldsymbol{z}}^T\right)^{-1}(\overline{\boldsymbol{\mu}}_t - \overline{\Sigma}_{\boldsymbol{x},\boldsymbol{z}}\mathcal{S}_t^{-1}\overline{\boldsymbol{z}}_t) \\ \mathcal{S}_t^{-1}\overline{\boldsymbol{z}}_t - \mathcal{S}_t^{-1}\overline{\Sigma}_{\boldsymbol{x},\boldsymbol{z}}^T\left(\overline{\Sigma}_t - \overline{\Sigma}_{\boldsymbol{x},\boldsymbol{z}}\mathcal{S}_t^{-1}\overline{\Sigma}_{\boldsymbol{x},\boldsymbol{z}}^T\right)^{-1}(\overline{\boldsymbol{\mu}}_t - \overline{\Sigma}_{\boldsymbol{x},\boldsymbol{z}}\mathcal{S}_t^{-1}\overline{\boldsymbol{z}}_t) \end{bmatrix}\end{aligned} \tag{2.22}$$

To avoid dealing with this, we quickly introduce the evidence $\boldsymbol{z}_t^e$. This sets $\boldsymbol{z}_t$ in place and reduces the clique's scope to $\boldsymbol{x}_t$. This is represented in canonical form as:

$$\beta_{\boldsymbol{z}(t)}(\boldsymbol{x}_t) = \mathcal{C}(\boldsymbol{x}_t; \overline{\mathcal{S}}_t, \overline{\boldsymbol{h}}_t, g_{\boldsymbol{z}(t)}) \tag{2.23}$$

Where,

$$\begin{aligned}\overline{\boldsymbol{h}}_t &= \left(\overline{\Sigma}_t - \overline{\Sigma}_{\boldsymbol{x},\boldsymbol{z}}\mathcal{S}_t^{-1}\overline{\Sigma}_{\boldsymbol{x},\boldsymbol{z}}^T\right)^{-1}(\overline{\boldsymbol{\mu}}_t + \overline{\Sigma}_{\boldsymbol{x},\boldsymbol{z}}\mathcal{S}_t^{-1}(\boldsymbol{z}_t^e - \overline{\boldsymbol{z}}_t)) \\ &= \left(\overline{\Sigma}_t^{-1} - \overline{\Sigma}_t^{-1}\left(\overline{\Sigma}_{\boldsymbol{x},\boldsymbol{z}}^T\overline{\Sigma}_t^{-1}\overline{\Sigma}_{\boldsymbol{x},\boldsymbol{z}} - \mathcal{S}_t\right)^{-1}\overline{\Sigma}_t^{-1}\right)(\overline{\boldsymbol{\mu}}_t + \overline{\Sigma}_{\boldsymbol{x},\boldsymbol{z}}\mathcal{S}_t^{-1}(\boldsymbol{z}_t^e - \overline{\boldsymbol{z}}_t))\end{aligned} \tag{2.24}$$

$$\begin{aligned}\overline{\mathcal{S}}_t^{-1} &= \left(\overline{\Sigma}_t - \overline{\Sigma}_{\boldsymbol{x},\boldsymbol{z}}\mathcal{S}_t^{-1}\overline{\Sigma}_{\boldsymbol{x},\boldsymbol{z}}^T\right)^{-1} \\ &= \overline{\Sigma}_t^{-1} - \overline{\Sigma}_t^{-1}\left(\overline{\Sigma}_{\boldsymbol{x},\boldsymbol{z}}^T\overline{\Sigma}_t^{-1}\overline{\Sigma}_{\boldsymbol{x},\boldsymbol{z}} - \mathcal{S}_t\right)^{-1}\overline{\Sigma}_t^{-1}\end{aligned} \tag{2.25}$$

The clique no longer has a unique scope and there is no need for it to exist independently [3, 4, 9]. The clique can be absorbed into transition clique (Figure 2.2b), therefore the outgoing message is the entirety of the remaining distribution. We must now legitimately apply the BUP algorithm, dividing out the received message so that the information is not double-counted:

$$\begin{aligned}\tilde{\delta}_{\boldsymbol{z}(t)\to\boldsymbol{x}(t)}(\boldsymbol{x}_t) &= \frac{\beta_{\boldsymbol{z}(t)}(\boldsymbol{x}_t)}{\delta_{\boldsymbol{x}(t)\to\boldsymbol{z}(t)}(\boldsymbol{x}_t)} \\ &= \mathcal{C}(\boldsymbol{x}_t; \overline{\mathcal{S}}_t^{-1} - \overline{\Sigma}_t^{-1}, \overline{\boldsymbol{h}}_t - \overline{\Sigma}_t^{-1}\overline{\boldsymbol{\mu}}_t, g_{\boldsymbol{z}(t)} - g_{\boldsymbol{x}(t)}) \\ &= \mathcal{C}(\boldsymbol{x}_t; \hat{\mathcal{S}}_t^{-1}, \hat{\boldsymbol{h}}_t, \hat{\boldsymbol{g}}_{\boldsymbol{z}(t)})\end{aligned} \tag{2.26}$$

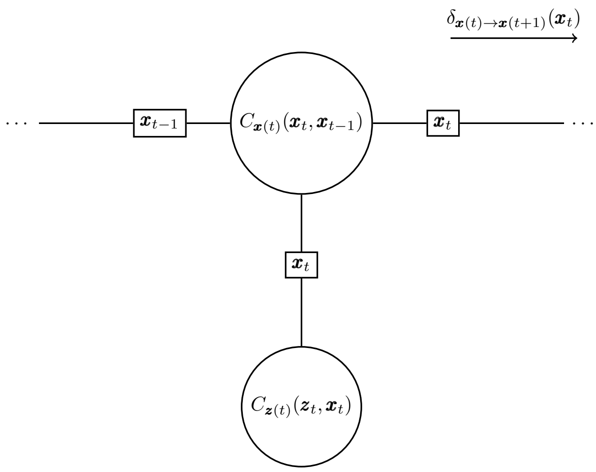


Figure 2.3: Forward Message Propagation. The final outgoing message fuses the predicted state and the observation, much as it should.

Where,

$$\hat{\boldsymbol{h}}_t = \overline{\Sigma}_t^{-1}\overline{\Sigma}_{\boldsymbol{x},\boldsymbol{z}}\mathcal{S}_t^{-1}(\boldsymbol{z}_t^e - \overline{\boldsymbol{z}}_t)) - \overline{\Sigma}_t^{-1}\left(\overline{\Sigma}_{\boldsymbol{x},\boldsymbol{z}}^T\overline{\Sigma}_t^{-1}\overline{\Sigma}_{\boldsymbol{x},\boldsymbol{z}} - \mathcal{S}_t\right)^{-1}\overline{\Sigma}_t^{-1}(\overline{\boldsymbol{\mu}}_t + \overline{\Sigma}_{\boldsymbol{x},\boldsymbol{z}}\mathcal{S}_t^{-1}(\boldsymbol{z}_t^e - \overline{\boldsymbol{z}}_t)) \tag{2.27}$$

$$\hat{\mathcal{S}}_t^{-1} = -\overline{\Sigma}_t^{-1}\left(\overline{\Sigma}_{\boldsymbol{x},\boldsymbol{z}}^T\overline{\Sigma}_t^{-1}\overline{\Sigma}_{\boldsymbol{x},\boldsymbol{z}} - \mathcal{S}_t\right)^{-1}\overline{\Sigma}_t^{-1} \tag{2.28}$$

2.1.3 Measurement Update

The incoming message of Figure 2.2b is incorporated into the existing clique belief:

$$\begin{aligned}\beta_{\boldsymbol{x}(t)}(\boldsymbol{x}_t, \boldsymbol{x}_{t-1}) &= \frac{\tilde{\delta}_{\boldsymbol{z}(t)\rightarrow\boldsymbol{x}(t)}(\boldsymbol{x}_t)}{\delta_{\boldsymbol{z}(t)\rightarrow\boldsymbol{x}(t)}(\boldsymbol{x}_t)}\mathcal{C}(\boldsymbol{x}_t, \boldsymbol{x}_{t-1}; \mathcal{P}_{\boldsymbol{x}(t)}, \boldsymbol{h}_{\boldsymbol{x}(t)}, g_{\boldsymbol{x}(t)}) \\ &= \tilde{\delta}_{\boldsymbol{z}(t)\rightarrow\boldsymbol{x}(t)}(\boldsymbol{x}_t)\mathcal{C}(\boldsymbol{x}_t, \boldsymbol{x}_{t-1}; \mathcal{P}_{\boldsymbol{x}(t)}, \boldsymbol{h}_{\boldsymbol{x}(t)}, g_{\boldsymbol{x}(t)}) \end{aligned} \tag{2.29}$$

We can finally propagate the message forward to the next transition clique, this is obviously done by marginalising onto the sepset:

$$\begin{aligned}\tilde{\delta}_{\boldsymbol{x}(t)\rightarrow\boldsymbol{x}(t+1)}(\boldsymbol{x}_t) &= \int \beta_{\boldsymbol{x}(t)}(\boldsymbol{x}_t, \boldsymbol{x}_{t-1})d\boldsymbol{x}_{t-1} \\ &= \tilde{\delta}_{\boldsymbol{z}(t)\rightarrow\boldsymbol{x}(t)}(\boldsymbol{x}_t)\int \mathcal{C}(\boldsymbol{x}_t, \boldsymbol{x}_{t-1}; \mathcal{P}_{\boldsymbol{x}(t)}, \boldsymbol{h}_{\boldsymbol{x}(t)}, g_{\boldsymbol{x}(t)})d\boldsymbol{x}_{t-1} \\ &= \tilde{\delta}_{\boldsymbol{z}(t)\rightarrow\boldsymbol{x}(t)}(\boldsymbol{x}_t)\tilde{\delta}_{\boldsymbol{x}(t)\rightarrow\boldsymbol{z}(t)}(\boldsymbol{x}_t) \\ &= \mathcal{C}(\boldsymbol{x}_t; \Sigma_t^{-1}, \boldsymbol{h}_t, g_t) \end{aligned} \tag{2.30}$$

Where,

$$\Sigma_t = \overline{\Sigma}_t - \overline{\Sigma}_{\boldsymbol{x},\boldsymbol{z}}\mathcal{S}_t^{-1}\overline{\Sigma}_{\boldsymbol{x},\boldsymbol{z}}^T \tag{2.31}$$

To be consistent with [7, 14], we will make use of the stock definition of Kalman gain:

$$K_t = \overline{\Sigma}_{\boldsymbol{x},\boldsymbol{z}}\mathcal{S}_t^{-1} \tag{2.32}$$

$$\therefore \Sigma_t = \overline{\Sigma}_t - K_t\mathcal{S}_tK_t^T \tag{2.33}$$

Using the information vector we now determine and simplify the mean:

$$\boldsymbol{h}_t = \left(\overline{\Sigma}_t - \overline{\Sigma}_{\boldsymbol{x},\boldsymbol{z}}\mathcal{S}_t^{-1}\overline{\Sigma}_{\boldsymbol{x},\boldsymbol{z}}^T\right)^{-1}(\overline{\boldsymbol{\mu}}_t + \overline{\Sigma}_{\boldsymbol{x},\boldsymbol{z}}\mathcal{S}_t^{-1}(\boldsymbol{z}_t^e - \overline{\boldsymbol{z}}_t)) \tag{2.34}$$

$$\begin{aligned}\boldsymbol{\mu}_t &= \Sigma_t \boldsymbol{h}_t \\ &= \overline{\boldsymbol{\mu}}_t + \overline{\Sigma}_{\boldsymbol{x},\boldsymbol{z}}\mathcal{S}_t^{-1}(\boldsymbol{z}_t^e - \overline{\boldsymbol{z}}_t) \\ &= \overline{\boldsymbol{\mu}}_t + K_t(\boldsymbol{z}_t^e - \overline{\boldsymbol{z}}_t)\end{aligned} \tag{2.35}$$

Equations 2.24, 2.25, 2.34 and 2.31 are equivalent. Equation 2.26 removes the prediction information that the transition clique initially sent, while Equation 2.30 adds it back in. We could have just propagated the final message outwards from the measurement clique, avoided the unnecessary step and saved ourselves some time. This implies standard BP and provides the motivation for the simpler chain structure explored in Section 3.

2.2 Linear Systems

This section simply re-derives the linear Kalman using the suggested approach. It is pretty much a word-for-word rehash of Section 2.1, showing that the linear Kalman is simply a special case of the UKF. Now for an an unnecessary recap of the linear Kalman Filter:

1. The state transition probability $p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}, \boldsymbol{u}_t)$ is a linear function in its arguments, with added Gaussian noise.

$$\boldsymbol{x}_t = A\boldsymbol{x}_{t-1} + B\boldsymbol{u}_t + \boldsymbol{\epsilon}_t \tag{2.36}$$

   A and B are $n \times n$ and $m \times n$ matrices respectively, where n is the dimension of the state vector. We can neatly capture the process model as:

$$\boldsymbol{g}(\boldsymbol{x}_{t-1}, \boldsymbol{u}_t) = A\boldsymbol{x}_{t-1} + B\boldsymbol{u}_t \tag{2.37}$$

   $\boldsymbol{\epsilon}_t$ is still the same zero mean Gaussian noise with a covariance matrix R.

2. The measurement probability $p(\boldsymbol{z}_t|\boldsymbol{x}_t)$ is also linear in its arguments, with added Gaussian noise.

$$\boldsymbol{z}_t = C\boldsymbol{x}_t + \boldsymbol{\delta}_t \tag{2.38}$$

   C is $k \times n$, $\boldsymbol{z}_t$ being k dimensional. We can neatly capture the measurement model as:

$$\boldsymbol{h}(\boldsymbol{x}_t) = C\boldsymbol{x}_t \tag{2.39}$$

   $\boldsymbol{\delta}_t$ is still zero mean Gaussian noise with a covariance matrix Q.

3. The belief about the initial state, $p(\boldsymbol{x}_0)$ is normally distributed:

$$p(\boldsymbol{x}_0) = \mathcal{N}(\boldsymbol{x}_0|\boldsymbol{\mu}_0, \Sigma_0) \tag{2.40}$$

While in practice we will always make use of the Unscented Transform, for this derivation we will allow the integral of Equation 1.16 its exact closed-form solution. This will allow us to arrive neatly at Algorithm 3, including a few extra steps.

Algorithm 3 The Linear Kalman Filter.

function KALMAN-FILTER($\boldsymbol{\mu}_{t-1}$, Σ_{t-1}, $\boldsymbol{u}_t$, $\boldsymbol{z}_t^e$)

$\overline{\boldsymbol{\mu}}_t = A\boldsymbol{\mu}_{t-1} + B\boldsymbol{u}_t$
$\Sigma_t = A\Sigma_{t-1}A^T + R$

$K_t = \overline{\Sigma}_t C^T \left(C\overline{\Sigma}_t C^T + Q\right)^{-1}$
$\boldsymbol{\mu}_t = \overline{\boldsymbol{\mu}}_t + K_t \left(\boldsymbol{z}_t^e - C\overline{\boldsymbol{\mu}}_t\right)$
$\Sigma_t = \left(I - K_t C\right)\overline{\Sigma}_t$

return $\boldsymbol{\mu}_t$, Σ_t

2.2.1 Prediction: Transition Clique Construction

As seen in Figure 2.1, we begin with some incoming Gaussian message represented in canonical form:

$$\tilde{\delta}_{\boldsymbol{x}(t-1)\to\boldsymbol{x}(t)}(\boldsymbol{x}_{t-1}) = \mathcal{C}(\boldsymbol{x}_{t-1}; \Sigma_{t-1}^{-1}, \boldsymbol{h}_{t-1}, g_{t-1}) \tag{2.41}$$

Since we are using BUP, the transition clique already holds the following belief and we merely incorporate the new information:

$$\begin{aligned}\beta_{\boldsymbol{x}(t)}(\boldsymbol{x}_t, \boldsymbol{x}_{t-1}) &= \frac{\tilde{\delta}_{\boldsymbol{x}(t-1)\to\boldsymbol{x}(t)}(\boldsymbol{x}_{t-1})}{\delta_{\boldsymbol{x}(t-1)\to\boldsymbol{x}(t)}(\boldsymbol{x}_{t-1})}\delta_{\boldsymbol{x}(t-1)\to\boldsymbol{x}(t)}(\boldsymbol{x}_{t-1})\delta_{\boldsymbol{x}(t+1)\to\boldsymbol{x}(t)}(\boldsymbol{x}_t)\delta_{\boldsymbol{z}(t)\to\boldsymbol{x}(t)}(\boldsymbol{x}_t)\Psi_{\boldsymbol{x}(t)}(\boldsymbol{x}_t, \boldsymbol{x}_{t-1}) \\ &= \tilde{\delta}_{\boldsymbol{x}(t-1)\to\boldsymbol{x}(t)}(\boldsymbol{x}_{t-1})\Psi_{\boldsymbol{x}(t)}(\boldsymbol{x}_t, \boldsymbol{x}_{t-1})\end{aligned} \tag{2.42}$$

The belief is reduced as only the incoming message contains any information, current and future events are naturally vacuous.

We now begin the arduous journey of populating the clique by creating the joint Gaussian density. We can determine the exact mean of the predicted state $\boldsymbol{x}_t$:

$$\begin{aligned}\overline{\boldsymbol{\mu}}_t &= \mathbb{E}\left[\boldsymbol{g}(\boldsymbol{x}_{t-1}, \boldsymbol{u}_t)\right] \\ &= \mathbb{E}\left[A\boldsymbol{x}_{t-1} + B\boldsymbol{u}_t\right] \\ &= A\boldsymbol{\mu}_{t-1} + B\boldsymbol{u}_t\end{aligned} \tag{2.43}$$

$\boldsymbol{x}_t$'s covariance matrix can now be constructed, perturbed with the process noise covariance R:

$$\begin{aligned}\overline{\Sigma}_t &= \mathbb{E}\left[(\boldsymbol{g}(\boldsymbol{x}_{t-1}, \boldsymbol{u}_t) - \overline{\boldsymbol{\mu}}_t)(\boldsymbol{g}(\boldsymbol{x}_{t-1}, \boldsymbol{u}_t) - \overline{\boldsymbol{\mu}}_t)^T\right] + R \\ &= \mathbb{E}\left[(A\boldsymbol{x}_{t-1} - A\boldsymbol{\mu}_{t-1})(A\boldsymbol{x}_{t-1} - A\boldsymbol{\mu}_{t-1})^T\right] + R \\ &= A\mathbb{E}\left[(\boldsymbol{x}_{t-1} - \boldsymbol{\mu}_{t-1})(\boldsymbol{x}_{t-1} - \boldsymbol{\mu}_{t-1})^T\right]A^T + R \\ &= A\Sigma_{t-1}A^T + R\end{aligned} \tag{2.44}$$

We will also determine the cross-covariance of $\boldsymbol{x}_{t-1}$ and $\boldsymbol{x}_t$.

$$\begin{aligned}\overline{\Sigma}_{\boldsymbol{x}_{t-1},\boldsymbol{x}_t} &= \mathbb{E}\left[(\boldsymbol{x}_{t-1} - \boldsymbol{\mu}_{t-1})(\boldsymbol{g}(\boldsymbol{x}_{t-1}, \boldsymbol{u}_t) - \overline{\boldsymbol{\mu}}_t)^T\right] \\ &= \mathbb{E}\left[(\boldsymbol{x}_{t-1} - \boldsymbol{\mu}_{t-1})(A\boldsymbol{x}_{t-1} - A\boldsymbol{\mu}_{t-1})^T\right] \\ &= \mathbb{E}\left[(\boldsymbol{x}_{t-1} - \boldsymbol{\mu}_{t-1})(\boldsymbol{x}_{t-1} - \boldsymbol{\mu}_{t-1})^T\right]A^T \\ &= \Sigma_{t-1}A^T\end{aligned} \tag{2.45}$$

This will not be used directly in this section, but it is absolutely required as we will be representing the clique belief as a canonical Gaussian and require a full precision matrix.

$$\beta_{\boldsymbol{x}(t)}(\boldsymbol{x}_t, \boldsymbol{x}_{t-1}) = \mathcal{C}(\boldsymbol{x}_t, \boldsymbol{x}_{t-1}; \mathcal{P}_{\boldsymbol{x}(t)}, \boldsymbol{h}_{\boldsymbol{x}(t)}, g_{\boldsymbol{x}(t)}) \tag{2.46}$$

Here the joint covariance and mean are determined as in Section 1.2.2.

$$\Sigma_{\boldsymbol{x}(t)} = \begin{bmatrix} A\Sigma_{t-1}A^T + R & A\Sigma_{t-1} \\ \Sigma_{t-1}A^T & \Sigma_{t-1} \end{bmatrix} \tag{2.47}$$

$$\boldsymbol{\mu}_{\boldsymbol{x}(t)} = \begin{bmatrix} A\boldsymbol{\mu}_{t-1} + B\boldsymbol{u}_t \\ \boldsymbol{\mu}_{t-1} \end{bmatrix} \tag{2.48}$$

For interests sake, we will determine the precision matrix and information vector, its not as general and therefore not as unwieldy. They are equivalent to the rewritten linear conditional Gaussian's after receiving the recursive belief update:

$$\mathcal{P}_{\boldsymbol{x}(t)} = \Sigma^{-1}_{\boldsymbol{x}(t)} = \begin{bmatrix} R^{-1} & -R^{-1}A \\ -A^T R^{-1} & \Sigma^{-1}_{t-1} + A^T R^{-1} A \end{bmatrix} \tag{2.49}$$

$$\boldsymbol{h}_{\boldsymbol{x}(t)} = \mathcal{P}_{\boldsymbol{x}(t)}\boldsymbol{\mu}_{\boldsymbol{x}(t)} = \begin{bmatrix} R^{-1}B\boldsymbol{u}_t \\ -A^T R^{-1} B\boldsymbol{u}_t + \Sigma^{-1}_{t-1}\boldsymbol{\mu}_{t-1} \end{bmatrix} \tag{2.50}$$

We now must determine the outgoing message, to be used to construct the measurement clique. This is immensely simple as we just drop everything we don't need from the joint covariance and mean. The division here is also only included to be formal, the measurement clique is currently vacuous and offered no initial information.

$$\begin{aligned} \tilde{\delta}_{\boldsymbol{x}(t)\to\boldsymbol{z}(t)}(\boldsymbol{x}_t) &= \frac{1}{\delta_{\boldsymbol{z}(t)\to\boldsymbol{x}(t)}(\boldsymbol{x}_t)} \int \beta_{\boldsymbol{x}(t)}(\boldsymbol{x}_t, \boldsymbol{x}_{t-1}) d\boldsymbol{x}_{t-1} \\ &= \int \beta_{\boldsymbol{x}(t)}(\boldsymbol{x}_t, \boldsymbol{x}_{t-1}) d\boldsymbol{x}_{t-1} \\ &= \mathcal{C}(\boldsymbol{x}_t; \overline{\Sigma}_t^{-1}, \overline{\Sigma}_t^{-1}\overline{\boldsymbol{\mu}}_t, g_{\boldsymbol{x}(t)}) \end{aligned} \tag{2.51}$$

2.2.2 Measurement Clique Construction

As seen in Figure 2.2a, the incoming message of Equation 2.11 arrives:

$$\begin{aligned} \beta_{\boldsymbol{z}(t)}(\boldsymbol{x}_t, \boldsymbol{z}_t) &= \frac{\tilde{\delta}_{\boldsymbol{x}(t)\to\boldsymbol{z}(t)}(\boldsymbol{x}_t)}{\delta_{\boldsymbol{x}(t)\to\boldsymbol{z}(t)}} \delta_{\boldsymbol{x}(t)\to\boldsymbol{z}(t)}(\boldsymbol{x}_t)\Psi_{\boldsymbol{z}(t)}(\boldsymbol{x}_t, \boldsymbol{z}_t) \\ &= \tilde{\delta}_{\boldsymbol{x}(t)\to\boldsymbol{z}(t)}(\boldsymbol{x}_t)\Psi_{\boldsymbol{z}(t)}(\boldsymbol{x}_t, \boldsymbol{z}_t) \end{aligned} \tag{2.52}$$

The incoming message again contains the only information, $\Psi_{\boldsymbol{z}(t)}(\boldsymbol{x}_t, \boldsymbol{z}_t)$ is kept just to satisfy the required scope.

We set about creating the joint Gaussian much as before, advancing $\boldsymbol{x}_t$ through the measurement model. We can determine the mean of the measurement variable, $\boldsymbol{z}_t$:

$$\begin{aligned} \overline{\boldsymbol{z}}_t &= \mathbb{E}\left[\boldsymbol{h}(\boldsymbol{x}_{t-1})\right] \\ &= \mathbb{E}\left[C\boldsymbol{x}_t\right] \\ &= C\overline{\boldsymbol{\mu}}_t \end{aligned} \tag{2.53}$$

After which we calculate its covariance, perturbed again by the measurement noise covariance Q:

$$\begin{aligned} \mathcal{S}_t &= \mathbb{E}\left[(\boldsymbol{h}(\boldsymbol{x}_{t-1}) - \overline{\boldsymbol{z}}_t)(\boldsymbol{h}(\boldsymbol{x}_{t-1}) - \overline{\boldsymbol{z}}_t)^T\right] + Q \\ &= \mathbb{E}\left[(C\boldsymbol{x}_t - C\overline{\boldsymbol{\mu}}_t)(C\boldsymbol{x}_t - C\overline{\boldsymbol{\mu}}_t)^T\right] + Q \\ &= C\mathbb{E}\left[(\boldsymbol{x}_t - \overline{\boldsymbol{\mu}}_t)(\boldsymbol{x}_t - \overline{\boldsymbol{\mu}}_t)^T\right] C^T + Q \\ &= C\overline{\Sigma}_t C^T + Q \end{aligned} \tag{2.54}$$

The variables' cross-covariance is now determined:

$$\begin{aligned}\overline{\Sigma}_{\boldsymbol{x},\boldsymbol{z}} &= \mathbb{E}\left[(\boldsymbol{x}_t - \overline{\boldsymbol{\mu}}_t)(\boldsymbol{h}(\boldsymbol{x}_{t-1}) - \overline{\boldsymbol{z}}_t)^T\right] \\ &= \mathbb{E}\left[(\boldsymbol{x}_t - \overline{\boldsymbol{\mu}}_t)(C\boldsymbol{x}_t - C\overline{\boldsymbol{\mu}}_t)^T\right] \\ &= \mathbb{E}\left[(\boldsymbol{x}_t - \overline{\boldsymbol{\mu}}_t)(\boldsymbol{x}_t - \overline{\boldsymbol{\mu}}_t)^T\right] C^T \\ &= \overline{\Sigma}_t C^T \end{aligned} \tag{2.55}$$

We can now piece together the joint mean and covariance, be incredibly unimpressed:

$$\Sigma_{\boldsymbol{z}(t)} = \begin{bmatrix} \overline{\Sigma}_t & \overline{\Sigma}_t C^T \\ C\overline{\Sigma}_t & C\overline{\Sigma}_t C^T + Q \end{bmatrix} \tag{2.56}$$

$$\boldsymbol{\mu}_{\boldsymbol{z}(t)} = \begin{bmatrix} C\overline{\boldsymbol{\mu}}_t \\ \overline{\boldsymbol{\mu}}_t \end{bmatrix} \tag{2.57}$$

We must now work directly with the canonical form as it is still the most convenient method of introducing evidence. The current clique belief is:

$$\beta_{\boldsymbol{z}(t)}(\boldsymbol{x}_t, \boldsymbol{z}_t) = \mathcal{C}(\boldsymbol{x}_t, \boldsymbol{z}_t; \mathcal{P}_{\boldsymbol{z}(t)}, \boldsymbol{h}_{\boldsymbol{z}(t)}, g_{\boldsymbol{z}(t)}) \tag{2.58}$$

Where,

$$\mathcal{P}_{\boldsymbol{z}(t)} = \Sigma^{-1}_{\boldsymbol{z}(t)} = \begin{bmatrix} \overline{\Sigma}_t + C^T Q^{-1} C & -C^T Q^{-1} \\ -Q^{-1} C & Q^{-1} \end{bmatrix} \tag{2.59}$$

$$\boldsymbol{h}_{\boldsymbol{z}(t)} = \mathcal{P}_{\boldsymbol{z}(t)} \boldsymbol{\mu}_{\boldsymbol{z}(t)} = \begin{bmatrix} \overline{\Sigma}^{-1}_{t-1} \overline{\boldsymbol{\mu}}_t \\ \mathbf{0} \end{bmatrix} \tag{2.60}$$

We introduce the evidence, $\boldsymbol{z}^e_t$, setting $\boldsymbol{z}_t$ in place and reducing the clique's scope to $\boldsymbol{x}_t$. This is represented in canonical form as:

$$\beta_{\boldsymbol{z}(t)}(\boldsymbol{x}_t) = \mathcal{C}(\boldsymbol{x}_t; \overline{\mathcal{S}}_t, \overline{\boldsymbol{h}}_t, g_{\boldsymbol{z}(t)}) \tag{2.61}$$

Where,

$$\overline{\boldsymbol{h}}_t = \overline{\Sigma}^{-1}_{t-1} \overline{\boldsymbol{\mu}}_t + C^T Q^{-1} \boldsymbol{z}^e_t \tag{2.62}$$

$$\overline{\mathcal{S}}^{-1}_t = \overline{\Sigma}_t + C^T Q^{-1} C \tag{2.63}$$

The clique no longer has a unique scope and has no need to exist independently [3, 4, 9]. The clique can be absorbed into transition clique (Figure 2.2b), therefore the outgoing message is the entirety of the remaining distribution. We now legitimately apply the BUP algorithm, dividing out the received message to ensure the information is not double-counted:

$$\begin{aligned}\tilde{\delta}_{\boldsymbol{z}(t)\to\boldsymbol{x}(t)}(\boldsymbol{x}_t) &= \frac{\beta_{\boldsymbol{z}(t)}(\boldsymbol{x}_t)}{\delta_{\boldsymbol{x}(t)\to\boldsymbol{z}(t)}(\boldsymbol{x}_t)} \\ &= \mathcal{C}(\boldsymbol{x}_t; \overline{\mathcal{S}}^{-1}_t - \overline{\Sigma}^{-1}_t, \overline{\boldsymbol{h}}_t - \overline{\Sigma}^{-1}_t \overline{\boldsymbol{\mu}}_t, g_{\boldsymbol{z}(t)} - g_{\boldsymbol{x}(t)}) \\ &= \mathcal{C}(\boldsymbol{x}_t; \hat{\mathcal{S}}^{-1}_t, \hat{\boldsymbol{h}}_t, \hat{\boldsymbol{g}}_{\boldsymbol{z}(t)}) \end{aligned} \tag{2.64}$$

Where,

$$\hat{\boldsymbol{h}}_t = C^T Q^{-1} \boldsymbol{z}^e_t \tag{2.65}$$

$$\hat{\mathcal{S}}^{-1}_t = C^T Q^{-1} C \tag{2.66}$$

2.2.3 Measurement Update

The incoming message of Figure 2.2b is incorporated into the existing clique belief:

$$\begin{aligned}\beta_{\boldsymbol{x}(t)}(\boldsymbol{x}_t, \boldsymbol{x}_{t-1}) &= \frac{\tilde{\delta}_{\boldsymbol{z}(t)\to\boldsymbol{x}(t)}(\boldsymbol{x}_t)}{\delta_{\boldsymbol{z}(t)\to\boldsymbol{x}(t)}(\boldsymbol{x}_t)}\mathcal{C}(\boldsymbol{x}_t, \boldsymbol{x}_{t-1}; \mathcal{P}_{\boldsymbol{x}(t)}, \boldsymbol{h}_{\boldsymbol{x}(t)}, g_{\boldsymbol{x}(t)}) \\ &= \tilde{\delta}_{\boldsymbol{z}(t)\to\boldsymbol{x}(t)}(\boldsymbol{x}_t)\mathcal{C}(\boldsymbol{x}_t, \boldsymbol{x}_{t-1}; \mathcal{P}_{\boldsymbol{x}(t)}, \boldsymbol{h}_{\boldsymbol{x}(t)}, g_{\boldsymbol{x}(t)}) \end{aligned} \tag{2.67}$$

We can finally propagate the message forward to the next transition clique, this is obviously done by marginalising onto the sepset:

$$\begin{aligned}\tilde{\delta}_{\boldsymbol{x}(t)\to\boldsymbol{x}(t+1)}(\boldsymbol{x}_t) &= \int \beta_{\boldsymbol{x}(t)}(\boldsymbol{x}_t, \boldsymbol{x}_{t-1}) d\boldsymbol{x}_{t-1} \\ &= \tilde{\delta}_{\boldsymbol{z}(t)\to\boldsymbol{x}(t)}(\boldsymbol{x}_t) \int \mathcal{C}(\boldsymbol{x}_t, \boldsymbol{x}_{t-1}; \mathcal{P}_{\boldsymbol{x}(t)}, \boldsymbol{h}_{\boldsymbol{x}(t)}, g_{\boldsymbol{x}(t)}) d\boldsymbol{x}_{t-1} \\ &= \tilde{\delta}_{\boldsymbol{z}(t)\to\boldsymbol{x}(t)}(\boldsymbol{x}_t)\tilde{\delta}_{\boldsymbol{x}(t)\to\boldsymbol{z}(t)}(\boldsymbol{x}_t) \\ &= \mathcal{C}(\boldsymbol{x}_t; \Sigma_t^{-1}, \boldsymbol{h}_t, g_t) \end{aligned} \tag{2.68}$$

Where,

$$\begin{aligned}\Sigma_t &= (\overline{\Sigma}_t^{-1} + C^T Q^{-1} C)^{-1} \\ &= \overline{\Sigma}_t - \overline{\Sigma}_t C^T (Q + C\overline{\Sigma}_t C^T)^{-1} C\overline{\Sigma}_t \end{aligned} \tag{2.69}$$

To be consistent with [14], we will make use of the stock definition of Kalman gain:

$$K_t = \Sigma_{\boldsymbol{x},\boldsymbol{z}}\mathcal{S}_t^{-1} = \overline{\Sigma}_t C^T (Q + C\overline{\Sigma}_t C^T)^{-1} \tag{2.70}$$

$$\therefore \Sigma_t = (I - K_t C_t)\overline{\Sigma}_t \tag{2.71}$$

Using the information vector, we now determine and simplify the mean:

$$\boldsymbol{h}_t = \overline{\Sigma}_t^{-1}\overline{\boldsymbol{\mu}}_t + C^T Q^{-1}\boldsymbol{z}_t^e \tag{2.72}$$

$$\begin{aligned}\boldsymbol{\mu}_t &= \Sigma_t \boldsymbol{h}_t \\ &= (I - K_t C_t)\overline{\Sigma}_t(\overline{\Sigma}_t^{-1}\overline{\boldsymbol{\mu}}_t + C^T Q^{-1}\boldsymbol{z}_t^e) \\ &= (I - K_t C)\overline{\boldsymbol{\mu}}_t + (\overline{\Sigma}_t C^T - K_t C\overline{\Sigma}_t C^T) Q^{-1}\boldsymbol{z}_t^e \\ &= (I - K_t C)\overline{\boldsymbol{\mu}}_t + (K_t Q) Q^{-1}\boldsymbol{z}_t^e \\ &= \overline{\boldsymbol{\mu}}_t + K_t(\overline{\boldsymbol{\mu}}_t - C\boldsymbol{z}_t^e) \end{aligned} \tag{2.73}$$

This may seem like a thoroughly pointless exercise, but the linear derivation guided the UKF derivation so it already existed and was worth including.

3. Belief Propagation using an Alternate Topology

As mentioned at the end of Section 2.1.3, BUP applied within the tree of Figure 1.2 contained an unnecessary step. This step can be removed by rearranging the tree as a simple chain, as seen in Figure 3.2. As the Junction Tree Algorithm provides multiple maximal spanning trees, multiple topologies exists for the Bayes Net of Figure 1.1. Operating on this topology BUP will become equivalent to standard BP, with just a single left-to-right pass for filtering.

This section won't go to great lengths to re-derive the system, it is largely equivalent to previous attempt, but the final message propagates out the measurement clique.

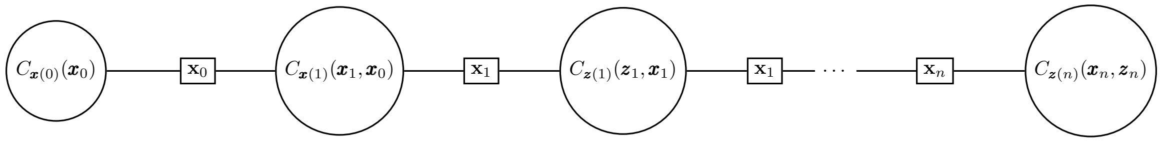


Figure 3.1: An alternate chain topology. This topology supports standard belief propagation, its initial clique assignments are identical to the previous attempt.

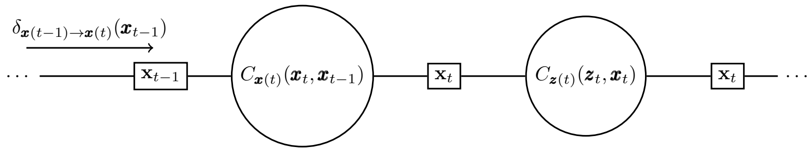


Figure 3.2: The incoming message to the transition clique. This information is used to populate the currently vacuous clique by creating a joint Gaussian, as described in Section 1.2.2.

3.1 Prediction

As seen in Figure 3.2, we begin with some incoming Gaussian message represented in canonical form:

$$\delta_{\boldsymbol{z}(t-1)\to\boldsymbol{x}(t)}(\boldsymbol{x}_{t-1}) = \mathcal{C}(\boldsymbol{x}_{t-1}; \Sigma_{t-1}^{-1}, \boldsymbol{h}_{t-1}, g_{t-1}) \tag{3.1}$$

This is the same incoming message as before and we create the same joint Gaussian as we did in Equation 2.8. The partial clique belief is given by:

$$\tilde{\beta}_{\boldsymbol{x}(t)}(\boldsymbol{x}_t, \boldsymbol{x}_{t-1}) = \mathcal{C}(\boldsymbol{x}_t, \boldsymbol{x}_{t-1}; \Sigma_{\boldsymbol{x}(t)}^{-1}, \Sigma_{\boldsymbol{x}(t)}^{-1}\boldsymbol{\mu}_{\boldsymbol{x}(t)}, g_{\boldsymbol{x}(t)}) \tag{3.2}$$

Where,

$$\Sigma_{\boldsymbol{x}(t)} = \begin{bmatrix} \overline{\Sigma}_t & \overline{\Sigma}_{\boldsymbol{x}_{t-1},\boldsymbol{x}_t}^T \\ \overline{\Sigma}_{\boldsymbol{x}_{t-1},\boldsymbol{x}_t} & \Sigma_{t-1} \end{bmatrix} \tag{3.3}$$

$$\boldsymbol{\mu}_{\boldsymbol{x}(t)} = \begin{bmatrix} \overline{\boldsymbol{\mu}}_t \\ \boldsymbol{\mu}_{t-1} \end{bmatrix} \tag{3.4}$$

We determine the outgoing message, again this equivalent to Equation 2.11:

$$\begin{aligned} \delta_{\boldsymbol{x}(t)\to\boldsymbol{z}(t)}(\boldsymbol{x}_t) &= \int \tilde{\beta}_{\boldsymbol{x}(t)}(\boldsymbol{x}_t, \boldsymbol{x}_{t-1}) d\boldsymbol{x}_{t-1} \\ &= \mathcal{C}(\boldsymbol{x}_t; \overline{\Sigma}_t^{-1}, \overline{\Sigma}_t^{-1}\overline{\boldsymbol{\mu}}_t, g_{\boldsymbol{x}(t)}) \end{aligned} \tag{3.5}$$

3.2 Measurement Update

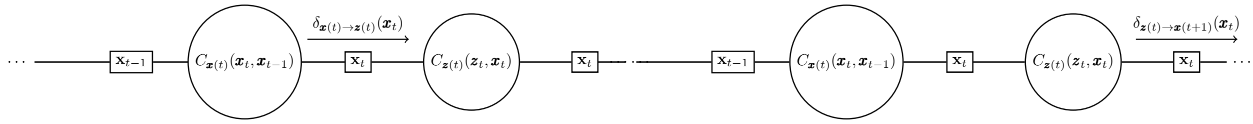
The incoming message of Figure 3.3 is used to construct the joint Gaussian. The partial clique belief is equivalent to Equation 2.20:

$$\begin{aligned} \tilde{\beta}_{\boldsymbol{z}(t)}(\boldsymbol{x}_t, \boldsymbol{z}_t) &= \delta_{\boldsymbol{x}(t)\to\boldsymbol{z}(t)}(\boldsymbol{x}_t)\Psi_{\boldsymbol{z}(t)}(\boldsymbol{x}_t, \boldsymbol{z}_t) \\ &= \mathcal{C}(\boldsymbol{x}_t, \boldsymbol{z}_t; \mathcal{P}_{\boldsymbol{z}(t)}, \boldsymbol{h}_{\boldsymbol{z}(t)}, g_{\boldsymbol{z}(t)}) \end{aligned} \tag{3.6}$$

Where,

$$\Sigma_{\boldsymbol{z}(t)} = \begin{bmatrix} \overline{\Sigma}_t & \overline{\Sigma}_{\boldsymbol{x},\boldsymbol{z}} \\ \overline{\Sigma}_{\boldsymbol{x},\boldsymbol{z}}^T & \mathcal{S}_t \end{bmatrix} \tag{3.7}$$

$$\boldsymbol{\mu}_{\boldsymbol{z}(t)} = \begin{bmatrix} C\overline{\boldsymbol{\mu}}_t \\ \overline{\boldsymbol{\mu}}_t \end{bmatrix} \tag{3.8}$$



(a) The predicted state vector's distribution is used to populate the vacuous joint distribution.

(b) The evidence is then introduced, the clique collapses and is absorbed into the next transition clique.

Figure 3.3: The creation and subsequent destruction of the measurement clique. After the introduction of evidence the clique no longer has a unique scope and no need to exist independently.

We once again introduce the evidence and the remaining distribution is absorbed into the next transition clique, see Figure 3.3b. The message is given by:

$$\delta_{\boldsymbol{z}(t)\to\boldsymbol{x}(t+1)}(\boldsymbol{x}_{t-1}) = \mathcal{C}(\boldsymbol{x}_t; \overline{\mathcal{S}}_t, \overline{\boldsymbol{h}}_t, g_{\boldsymbol{z}(t)}) \tag{3.9}$$

Where,

$$\overline{\boldsymbol{h}}_t = \left(\overline{\Sigma}_t - \overline{\Sigma}_{\boldsymbol{x},\boldsymbol{z}}\mathcal{S}_t^{-1}\overline{\Sigma}_{\boldsymbol{x},\boldsymbol{z}}^T\right)^{-1}(\overline{\boldsymbol{\mu}}_t + \overline{\Sigma}_{\boldsymbol{x},\boldsymbol{z}}\mathcal{S}_t^{-1}[\boldsymbol{z}_t^e - \overline{\boldsymbol{z}}_t]) \tag{3.10}$$

$$\overline{\mathcal{S}}_t^{-1} = \left(\overline{\Sigma}_t - \overline{\Sigma}_{\boldsymbol{x},\boldsymbol{z}}\mathcal{S}_t^{-1}\overline{\Sigma}_{\boldsymbol{x},\boldsymbol{z}}^T\right)^{-1} \tag{3.11}$$

Using the the stock definition of Kalman gain, we can simplify the covariance and mean.

$$K_t = \overline{\Sigma}_{\boldsymbol{x},\boldsymbol{z}}\mathcal{S}_t^{-1} \tag{3.12}$$

$$\therefore \boldsymbol{\mu}_t = \overline{\boldsymbol{\mu}}_t + K_t(\boldsymbol{z}_t^e - \overline{\boldsymbol{z}}_t) \tag{3.13}$$

$$\therefore \Sigma_t = \overline{\Sigma}_t - K_t\mathcal{S}_tK_t^T \tag{3.14}$$

4. Conclusion

This document is concluded.

References

[1] D. Barber. 'Belief Networks' in *Bayesian Reasoning and Machine Learning*, 1st ed., vol. 1. Cambridge, U.K: Cambridge University Press, 2012, pp. 29-58.

[2] D. Barber. 'Graphical Models' in *Bayesian Reasoning and Machine Learning*, 1st ed., vol. 1. Cambridge, U.K: Cambridge University Press, 2012, pp. 58-77.

[3] D. Barber. 'The Junction Tree Algorithm' in *Bayesian Reasoning and Machine Learning*, 1st ed., vol. 1. Cambridge, U.K: Cambridge University Press, 2012, pp. 102-127.

[4] J. Du Preez. 'Inference: The Junction-Tree Algorithm'. Internet: http://research.ee.sun.ac.za/pgms/instructions/week5_junctiontree.pdf, 2016 [Apr. 26, 2016].

[5] A. Gilat and V. Subramaniam. 'Numerical Integration' in *Numerical Methods: An Introduction with Applications using MATLAB*. M. McDonald, 2nd ed., Columbus, Ohio: John Wiley & Sons, 2011, pp. 263-269.

[6] S. Julier. "The Scaled Unscented Transformation", in *Proc. IEEE Cat. No.CH37301*, 2002, pp. 4555 - 4559.

[7] S. Julier and J. Ulhmann. "A New Extension of the Kalman Filter to Nonlinear Systems", in *Proc. SPIE 3068*, 1997.

[8] S. Julier and J. Ulhmann. "A General Method for Approximating Nonlinear Transformations of Probability Distributions", Technical report, 1996.

[9] D. Koller and N. Friedman. 'Exact Inference: Clique Trees' in *Probabilistic Graphical Models: Principles and Techniques*, 1st ed., vol. 1. Cambridge, Massachusetts: The MIT Press, 2009, pp. 345-376.

[10] D. Koller and N. Friedman. 'Inference as Optimization' in *Probabilistic Graphical Models: Principles and Techniques*, 1st ed., vol. 1. Cambridge, Massachusetts: The MIT Press, 2009, pp. 436-445.

[11] D. Koller and N. Friedman. 'Inference in Hybrid Networks' in *Probabilistic Graphical Models: Principles and Techniques*, 1st ed., vol. 1. Cambridge, Massachusetts: The MIT Press, 2009, pp. 609-611.

[12] T. Minka. 'A Family of Algorithms for Approximate Bayesian Inference'. PhD thesis, M.I.T, U.S.A, 2001.

[13] S. Thrun et al. 'Recursive State Estimation' in *Probabilistic Robotics*, 1st ed., vol. 1. Cambridge, Massachusetts: The MIT Press, 2009, pp. 14-35.

[14] S. Thrun et al. 'Gaussian Filters' in *Probabilistic Robotics*, 1st ed., vol. 1. Cambridge, Massachusetts: The MIT Press, 2009, pp. 39-53.

[15] Zoeter et al. 'Improved Unscented Kalman Smoothing for Stock Volatility Estimation', presented at the 2004 IEEE Workshop on Machine Learning for Signal Processing, IEEE, 2004.