1 Basics

$$\begin{aligned} x(k) &= q_{k-1}(x(k-1), u(k-1), v(k-1)) \\ z(k) &= h_k(x(k), w(k)) \end{aligned}$$

where $x(0)$, $\{v(\cdot)\}$, $\{w(\cdot)\}$ have a probabilistic description.

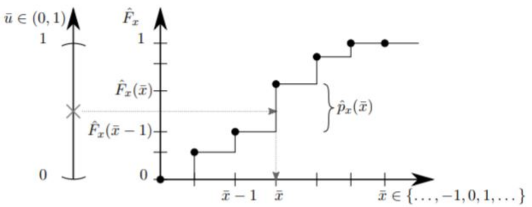
1.1 Sampling a Distribution

1.1.1 A1 Single Variable

Given a desired PDF $\hat{p}_x$ for a DRV x, samples $\bar{x}$ can be generated via the following procedure:

1. Calculate the CDF: $\hat{F}(\bar{x}) := \sum_{i=-\infty}^{\infty} \hat{p}_x(i)$

   Note that the range of $\hat{F}(\bar{x})$ is $(0, 1)$.
2. Sample $\bar{u}$ from a uniform distribution in $(0, 1)$.
3. Solve $\hat{F}_x(\bar{x}-1) < \bar{u}$ and $\bar{u} \leq \hat{F}_x(\bar{x})$ for $\bar{x}$.
4. The resulting $\bar{x}$ is sampled from the DRV x with pdf $\hat{p}_x$.



1.1.2 A2 Multiple DRVs

Option 1: For finite $\mathcal{X}$ and $\mathcal{Y}$

1. Let N_x and N_y be the number of elements in $\mathcal{X}$ and $\mathcal{Y}$. Define $\mathcal{Z} = \{1, 2, \dots, N_x N_y\}$.
2. Define

   $$\hat{p}_z(1) = \hat{p}_{xy}(1,1),\ \hat{p}_z(2) = \hat{p}_{xy}(1,2), \dots, \hat{p}_z(N_x N_y) = \hat{p}_{xy}(N_x, N_y)$$
3. Apply A1.

Option 2: For infinite numbers of elements

1. Decompose $\hat{p}_{xy}(\bar{x}, \bar{y}) = \hat{p}_{x|y}(\bar{x}|\bar{y})\hat{p}_y(\bar{y}$
2. Apply A1 to first get $\bar{y}$ via $\hat{p}_y(\bar{y})$.
3. Apply A1 to get $\bar{x}$ via $\hat{p}_{x|y}(\bar{x}|\bar{y}$ with $\bar{y}$ now fixed.

   Note that the independence of the uniform number generator between successive calls is key.

1.1.3 A3 One continuous random variable

1. Calculate the CDF $\hat{F}_x(\bar{x}) := \int_{-\infty}^{\bar{x}} \hat{p}_x(\lambda) d\lambda$
2. Let u be uniform on $(0, 1)$.
3. Let $\bar{x}$ be any solution to $\bar{u} = \hat{F}_x(\bar{x})$.
4. x has PDF $\hat{p}_x$.

1.1.4 A4 Multiple CRVs

Analog to option 2 of A2.

1.2 Change of Variables

1.2.1 DRV

Let p_y be given, $x = g(y)$. The goal is to calculate p_x.

$$\boxed{p_x(\bar{x}) = \sum_{\bar{y} \in \mathcal{Y}: g(\bar{x} = \bar{x}} p_y(\bar{y})}$$

1.2.2 CRV

Let $x = g(y)$, $g(y)$ strictly monotonic and continuously differentiable and p_y continuous.

$$\boxed{p_x(\bar{x}) = \frac{p_y(\bar{y}}{\frac{dg}{dy}(\bar{y})}}$$

This result in an alternative way of sampling:
Given a desired PDF $\hat{p}_x$ and a method for sampling p_y find a function $x = g(y)$ such that

$$\frac{dg}{dy}(\bar{y}) = \frac{p_y(\bar{y}}{\hat{p}_x(g(\bar{y}))}$$

Equivalently solve $\frac{d\bar{x}}{d\bar{y}} = \frac{p_y(\bar{y})}{\hat{p}_x(\bar{x})}$

2 Bayes' Theorem

$$\boxed{p(x|z) = \frac{p(y|z)p(x)}{p(z)}} \quad \text{Bayes' Theorem}$$

2.1 Generalization to multiple observations

- N observations: $z_1, \dots, z_n$
- Often conditional independence is assumed

$$p(z_1, \dots, z_N|x) = p(z_1|x) \cdots p(z_n|x)$$

$$\boxed{\underbrace{p(x|z_1, \dots, z_N)}_{\text{posterior}} = \frac{\overbrace{p(x)}^{\text{prior}} \overbrace{\prod_i p(z_i|x)}^{\text{observation likelihood}}}{\underbrace{p(z_1, \dots, z_N)}_{\text{normalization}}}}$$

$$p(z_1, \dots, z_N) = \sum_{x \in \mathcal{X}} p(x) \prod_i p(z_i|x) \quad \text{Normalization}$$

3 Bayesian Tracking

$$\begin{aligned} x(k) &\quad q_{k-1}(x(k-1), v(k-1)), \quad k = 1, 2, \dots \\ z(k) &\quad h_k(x(k), w(k)) \end{aligned}$$

Assume $p(x(k-1)|z(1:k-1))$ is known. For $k = 1$: $p(x(0))$.

1. **Prior update:** Forward prediction of the state estimate using the process model.

   $$\boxed{p(x(k)|z(1:k-1)) = \sum_{x(k-1) \in \mathcal{X}} \overbrace{p(x(k)|x(k-1))}^{\text{process model}} \overbrace{p(x(k-1)|z(1:k-1))}^{\text{previous iteration}}}$$
2. **Measurement update:** Combination of the prior with observations/measurements.

   $$\boxed{p_{x(k)|z(1:k)}(\bar{x}(k)|\bar{z}(1:k)) = \frac{\overbrace{p_{z(k)|x(k)}(\bar{z}(k)|\bar{x}(k))}^{\text{measurement model}} \overbrace{p_{x(k)|z(1:k-1)}(\bar{x}(k)|\bar{z}(1:k-1))}^{\text{prior}}}{\underbrace{\sum_{i \in \mathcal{X}} p_{z(k)|x(k)} p_{z(k)|x(k)}(\bar{z}(k)|i) p_{x(k)|z(1:k-1)}(i|\bar{z}(1:k-1))}_{\text{normalization}}}}$$

3.1 Computer Implementation

- Enumerate the state $\mathcal{X} = \{0, 1, \dots, N-1\}$
- Define $\underline{a}^i_{k|k} := p_{x(k)|z(1:k)}(i|\bar{z}(1:k))$, $i = 0, \dots, N-1$, an array with N elements that we use to store the posterior PDF at time k.
- $\underline{a}^i_{k|k-1} := p_{x(k)|z(1:k-1)}(i|\bar{z}(1:k-1))$, $i = 0, \dots, N-1$ to store the prior PDF at time k
- Algorithm
  1. Initialization, $k = 0$

     $$\underline{a}^i_{0|0} = p_{x(0)}(i), \ i = 0, \dots, N-1$$
  2. Recursion, $k > 0$

     $$\begin{aligned} \underline{a}^i_{k|k-1} &= \sum_{j=0}^{N-1} p_{x(k)|x(k-1)}(i|j)\underline{a}^j_{k-1|k-1}, \ i = 0, \dots, N-1 \\ \underline{a}^i_{k|k} &= \frac{p_{z(k)|x(k)}(\bar{z}(k)|i)\underline{a}^i_{k|k-1}}{\sum_{j=0}^{N-1} p_{z(k)|x(k)}(\bar{z}(k)|j)\underline{a}^j_{k|k-1}} \ i = 0, \dots, N-1 \end{aligned}$$
- Note that $p_{x(k)|x(k-1)}(i|j)$ can be calculated from $x(k) = q_{k-1}(x(k-1), v(k-1))$ and $p_{v(k-1)}(\bar{v}(k-1))$. Similarly, $p_{z(k)|x(k)}(\bar{z}(k)|i)$ can be calculated from $z(k) = h_k(x(k), w(k))$ and $p_{w(k)}(\bar{w}(k))$.

4 Extracting Estimates from Probability Distributions

4.1 Maximum Likelihood (ML)

$$\hat{x}^{ML} := \underset{\bar{x} \in \mathcal{X}}{\arg\max}\, p_{z|x}(\bar{z}|\bar{x}$$

4.1.1 Generalization

$$z = Hx + w, \quad H = \begin{bmatrix} H_1 \\ H_2 \\ \vdots \\ H_m \end{bmatrix} \in \mathbb{R}^{m \times n}, \quad H_i = \begin{bmatrix} h_{i1} & \cdots & h_{in} \end{bmatrix}, \ h_{ij} \in \mathbb{R}$$

Let g be a function mapping $w \in \mathbb{R}^m$ to $z \in \mathbb{R}^m$, $z = g(w)$ and we assume that:

$$\det\left(\frac{\partial g}{\partial w}(w)\right) \neq 0 \ \forall w$$

Also we assume that $z = g(w)$ has a unique solution for w in terms of z:

$$p_z(\bar{z}) = p_w(h(\bar{z})) \left|\det\left(\frac{\partial g}{\partial w}(h(\bar{z}))\right)\right|^{-1}$$

Finally the maximum likelihood estimator becomes

$$\boxed{\bar{x} = (H^T H)^{-1} H^T \bar{z}}$$

4.2 Maximum a Posteriori (MAP)

$$p_{x|z}(\bar{x}|\bar{z}) = \frac{p_{z|x}(\bar{z}|\bar{x})p_x(\bar{x})}{p_z(\bar{z})}$$

$$\boxed{\bar{x}^{MAP} := \underset{\bar{x} \in \mathcal{X}}{\arg\max}\, p_{z|x}(\bar{z}|\bar{x})p_x(\bar{x})}$$

4.3 Minimum Mean Squared Error (MMSE)

$$\begin{aligned} \hat{x}^{MMSE} &:= \underset{\hat{x}}{\arg\min} \underset{x|z}{E}\left[(\hat{x}-x)^T(\hat{x}-x)|\bar{z}\right] \\ &= \underset{\bar{x}}{\arg\min}\left(\hat{x}^T\hat{x} - 2\hat{x}^T \underset{x|z}{E}[x|\bar{z}] + \underset{x|z}{E}\left[x^T x|\bar{z}\right]\right) \\ &= \underset{x|z}{E}[x|\bar{z}] \end{aligned}$$

4.4 Recursive Least Squares (RLS)

$$z(k) = H(k) + w(k) \qquad z(k), w(k) \in \mathbb{R}^m,\ x \in \mathbb{R}^n$$

- Prior knowledge: mean and variance of x, $\hat{x}_0 := \mathbb{E}[x]$ and $P_x := \mathbb{E}\left[(x-\hat{x}_0)(x-\hat{x}_0)^T\right] = \mathrm{Var}(x)$ are given.
- Measurement noise: zero-mean with known variance. $\mathbb{E}[w(k)] = 0,\ R(k) := \mathrm{Var}(w(k))$
- Typically, $n > m$, fewer equations than unknowns at a particular time.

1. **Initialization** $\hat{x}(0) = \hat{x}_0,\ P(0) = P_x = \mathrm{Var}(x)$
2. **Recursion**
   - Observe $\bar{z}(k)$
   - Update

$$\begin{aligned} K(k) &= P(k-1)H^T(k)(H(k)P(k-1)H^T(k) + R(k))^{-1} \\ \hat{x}(k) &= \hat{x}(k-1) + K(k)(\bar{z}(k) - H(k)\hat{x}(k-1)) \\ P(k) &= (I - K(k)H(k))P(k-1)(I - K(k)H(k))^T + K(k)R(k)K^T(k) \end{aligned}$$

The matrices $K(k)$ and $P(k)$ can be pre-computed from the problem data P_x, $\{H(\cdot)\}$ and $\{R(\cdot)\}$.

5 The Kalman Filter

5.1 Model

$$\begin{aligned} x(x) &= A(k-1)x(k-1) + u(k-1) + v(k-1) \\ z(k) &= H(k)x(k) + w(k) \end{aligned}$$

5.2 Gaussian Random Variable (GRV)

$$p(y) = \frac{1}{(2\pi)^{D/2}\det(\Sigma)^{1/2}} \exp\left(-\tfrac{1}{2}(x-\mu)^T\Sigma^{-1}(y-\mu)\right) \qquad y \sim \mathcal{N}(\mu, \Sigma)$$

If Σ is diagonal we can say:

$$p(y) = \prod_{i=1,\dots,D} \frac{1}{\sqrt{2\pi\sigma_i^2}} \exp\left(-\frac{(y_i-\mu_i)^2}{2\sigma_i^2}\right)$$

5.2.1 Jointly Gaussian Random Variables

If the joint random variable (x, y) is a GRV, then x and y are said to be jointly Gaussian Random Variables. If they are additionally independent, then,

$$p(x,y) = p(x)p(y) = \exp\left(-\tfrac{1}{2}\left[(x-\mu_s)^T(y-\mu_y)\right]^T \begin{bmatrix} \sigma_x^{-1} & 0 \\ 0 & \Sigma_y^{-1} \end{bmatrix} \begin{bmatrix} x-\mu_x \\ y-\mu_y \end{bmatrix}\right)$$

5.2.2 Properties of GRV

- An affine transformation of a GRV is a GRV
- A linear combination of two jointly GRVs is a GRV

5.3 Problem Formulation

Application of bayesian tracking to CRVs.

$$\left.\begin{array}{ll} \textbf{Init:} & x_m(0) := x(0) \\ \textbf{S1:} & x_p(k) := A(k-1)x_m(k-1) + u(k-1) + v(k-1) \\ \textbf{S2:} & z_m(k) := H(k)x_p(k) + w(k) \\ & x_m(k) \text{ defined via its PDF} \\ & p_{x_m(k)}(\zeta) := p_{x_p(k)|z_m(k)}(\zeta|\bar{z}(k))\ \forall\zeta \end{array}\right\} \quad k = 1, 2, \dots$$

$$\begin{aligned} \hat{x}_p(k) &= \mathbb{E}[x_p(k)] \qquad & P_p(k) &:= \mathrm{Var}(x_p(k)) \\ \hat{x}_m(k) &= \mathbb{E}[x_m(k)] \qquad & P_m(k) &:= \mathrm{Var}(x_m(k)) \end{aligned}$$

- It can be shown that all auxiliary variables (x_p, x_m) are GRVs.
- Using this the mean and the variance can be calculated and lead to the **Kalman Filter equations** below

5.4 Kalman Filter Equations

$$\begin{array}{ll} \textbf{Init:} & x_m(0) := x(0),\ P_m(0) = P_0 \\ \textbf{S1:} & x_p(k) := A(k-1)\hat{x}_m(k-1) + u(k-1) \\ & P_p(k) = A(k-1)P_m(k-1)A^T(k-1) + Q(k-1) \\ \textbf{S2:} & P_m(k) = (P_p^{-1}(k) + H^T(k)R^{-1}(k)H(k))^{-1} \\ & \hat{x}_m(k) = \hat{x}_p(k) + P_m(k)H^T(k)R^{-1}(k)(\bar{z}(k) - H(k)\hat{x}_p(k)) \end{array}$$

5.4.1 Alternative Formulation

$$\begin{aligned} K(k) &= P_p(k)H^T(k)(H(k)P_p(k)H^T(k) + R(k))^{-1} \\ \hat{x}_m(k) &= \hat{x}_p(k) + K(k)(\bar{z}(k) - H(k)\hat{x}_p(k)) \\ P_m(k) &= (I - K(k)H(k))P_p(k) \\ &= (I - K(k)H(k))P_p(k)(I - K(k))^T + K(k)R(k)K^T(k) \end{aligned}$$

6 Kalman Filter as State Observer

- Assumption: Time invariant system
  $$A(k) = A,\ H(k) = H,\ Q(k) = Q,\ R(k) = R$$

6.1 Asymptotic Properties of the Kalman Filter

For constant A, H, Q, and R the KF is still time varying:

$$\begin{aligned} P_p(k) &= AP_m(k-1)A^T + Q \\ K(k) &= P_p(k)H^T(HP_p(k)H^T) + R)^{-1} \\ P_m(k) &= (I - K(k)H)P_p(k) \end{aligned}$$

Thus

$$P_p(k+1) = AP_p(k)A^T + Q - AP_p(k)H^t(HP_p(k)H^T + R)^{-1}HP_p(k)A^T$$

- The Kalman filter's variance might converge or diverge depending on the choice of A, H, Q, R and P_0.
- In the scalar case the dynamics reduce to:
  $$P_p(k+1) = \underbrace{\frac{a^2rP_p(k)}{h^2P_p(k) + r} + q}_{:=f(P_p(k))}$$

6.1.1 Steady State Behaviour

Steady state is reached when $P_\infty = f(P_\infty)$.
The Kalman filter converges to the unique steady-state solution provided that either $|a| < 1$ of if $|a| \geq 1,\ h \neq 0,\ q > 0$.

6.2 Detectability

The pair (A, H) is detectable
⇔ For a deterministic LTI system $(x(k) = Ax(k-1),\ z(k) = Hx(k))$, $\lim_{k\to\infty} z(k) = 0 \Rightarrow \lim_{k\to\infty} x(k) = 0,\ \forall x_0 \in \mathbb{R}^n$.
⇔ $\begin{bmatrix} A - \lambda I \\ H \end{bmatrix}$ is full rank for all $\lambda \in \mathbb{C}$ with $|\lambda| \geq 1$ (PBH-Test).
⇔ The eigenvalues of $A - LH$ (or equivalently (I-LH)A) can be placed within the unit circle by a suitable choice of the matrix $L \in \mathbb{R}^{n\times m}$.

The pair (A, H) is observable.
⇔ For a deterministic LTI system $(x(k) = Ax(k-1) + u(k-1),\ z(k) = Hx(k))$ knowledge of $z(0:n-1)$ and $u(0:n-1)$ suffices to determine $x(0)$.
⇔ $\operatorname{rank}\begin{pmatrix} H \\ HA \\ \vdots \\ HA^{-n1} \end{pmatrix} = n$
⇔ $\begin{bmatrix} A - \lambda I \\ H \end{bmatrix}$ is full rank for all $\lambda \in \mathbb{C}$ (PBH-Test)
⇔ The eigenvalues of $A - LH$ can be placed arbitrarily by a suitable choice of the matrix $L \in \mathbb{R}^{n\times m}$
Furthermore if (A, H) is detectable but not observable then there exists a state transformation such that

$$TA^{-1} = \begin{bmatrix} A_{11} & 0 \\ A_{21} & A_{22} \end{bmatrix}, \quad HT^{-1} = [H_1 0], \quad \text{and } (A_{11}, H_1) \text{ observable}$$

6.2.1 Stabilizability

Stabilizability is the dual of detectability, that is (A, B) is stabilizable iff (A^T, B^T) is detectable. Controllability is the dual of observability, that is (A, B) is controllable iff (A^T, B^T) is observable.

6.3 Steady-State Kalman Filter

+ Less computational effort.
+ Simpler implementation.
- Only works for convergent variances.

$$\boxed{P_\infty = AP_\infty A^T + Q - AP_\infty H^T(HP_\infty H^T + R)^{-1}HP_\infty A^T} \text{ DARE}$$

$$\boxed{K_\infty = P_\infty H^T(HP_\infty H^T + R)^{-1}}$$

$$\begin{aligned} \hat{x}(k) &= (I - K_\infty H)A\hat{x}(k-1) + (I - K_\infty H)u(k-1) + K_\infty\bar{z}(k) \\ &= \hat{A}\hat{x}(k-1) + \hat{B}u(k-1) + K_\infty\bar{z}(k), \quad \hat{x}(0) = x_0 \end{aligned}$$

The error dynamics can be described as

$$e(k) = \underbrace{(I - K_\infty H)A)}_{\text{stability important!}} Ae(k-1) + (I - K_\infty H)v(k-1) - K_\infty w(k)$$

$$E[e(k)] = (I - K_\infty H)AE[e(k-1)]$$

- If the filter is initialized with $E[e(0)] = 0$ then the expected value of the error remains zero for all time. Otherwise it will be nonzero for all time, but still the filter is stable.

- $P_p(k)$ might not converge.
- $P_p(k)$ does not converge to the same solution for different $P_p(1)$.
- $(I - K_\infty H)A$ might be unstable.

To address the issues above, assume $R > 0$ and $Q = GG^T \geq 0$, then the following statements are equivalent:

1. (A, H) is detectable and (A, G) is stabilizable.
2. The DARE has a unique positive semidefinite solution $P_\infty \geq 0$, the resulting $(I - K_\infty H)A$ is stable and
   $$\lim_{k\to\infty} P_p(k) = P_\infty \text{ for any initial } P_p(1) \geq 0 \text{ (and hence, any } P_m(0) = P_0 \geq 0)$$

- If $Q > 0$ then (A, G) is always stabilizable.

7 Extended Kalman Filter

Nonlinear discrete-time system:

$$x(k) = q_{k-1}(x(k-1), v(k-1)) \qquad E[x(0)] = x_0,\ \mathrm{Var}[x(0)] = P_0$$
$$E[v(k-1)] = 0,\ \mathrm{Var}[v(k-1)] = Q(k-1)$$
$$z(k) = h_k(x(k), w(k)) \qquad E[w(k)] = 0,\ \mathrm{Var}[w(k)] = R(k)$$

- $x(0), \{v(\cdot)\}$ and $\{w(\cdot)\}$ mutually independent.
- q_{k-1} continuously differentiable w.r.t. $x(k-1)$ and $v(k-1)$.
- h_k continuously differentiable w.r.t. $x(k)$
- The known input can be included in the above description by absorbing it in the explicit time dependency of $q_{k-1}(\cdot)$.
- **Initialization** $\hat{x}_m(0) = x_0,\ P_m(0) = P_0$
- **Prior update / Prediction step**

$$\hat{x}_p(k) = q_{k-1}(\hat{x}_m(k-1), 0)$$
$$P_p(k) = A(k-1)A^T(k-1) + L(k-1)Q(k-1)L^T(k-1)$$

where

$$A(k-1) := \frac{\partial q_{k-1}(\hat{x}_m(k-1), 0)}{\partial x} \text{ and } L(k-1) := \frac{\partial q_{k-1}(\hat{x}_m(k-1), 0)}{\partial v}$$

- **A posteriori update / Measurement update step**

$$K(k) = P_p(k)H^T(k)(H(k)P_p(k)H^T(k) + M(k)R(k)M^T(k))^{-1}$$
$$\hat{x}_m(k) = \hat{x}_p(k) + K(k)(\bar{z}(k) - h_k(\hat{x}_p(k), 0))$$
$$P_m(k) = (I - K(k)H(k))P_p(k)$$

where

$$H(k) := \frac{\partial h_k(\hat{x}_p(k), 0)}{\partial x} \text{ and } M(k) := \frac{\partial h_k(\hat{x}_p(k), 0)}{\partial w}$$

- $A(k-1), L(k-1), H(k), M(k)$ are obtained from linearization about the current state estimate, and can thus not be computed offline, even if model and noise distributions are known for all k.
- The above approximations are good as long as the actual state and noise values are close to the points we linearize about. This assumption might be bad, especially when considering gaussian noise, which is actually unbounded.
- $\hat{x}_p(k), \hat{x}_m(k), P_p(k), P_m(k)$ no longer capture the true conditional mean and variance of $x(k)$, especially if there are strong non-linearities (it holds for linear systems though).

7.1 Hybrid Extended Kalman Filter

$$E[v_d[k]] = 0 \text{ and } E[v_d[k]v_d^t[k+n]] = Q\delta_d[n]$$
discrete time white noise $v_d[k]$

where $\delta[n]$ is the discrete time Dirac pulse.

$$E[v(t)]) = 0 \text{ and } E[v(t)v^T(t+\tau)] = Q_c\delta(\tau)$$
continuous time white noise $v(t)$

where $\delta(\tau)$ is the continuous time Dirac pulse.

- **Initialization** $\hat{x}_m[0] = x_0,\ P_m[0] = P_0$
- **Prior update / Prediction step**
  Solve
  $$\dot{\hat{x}} = q(\hat{x}(t), 0, t), \text{ for } (k-1)T \le t \le kT \text{ and } \hat{x}((k-1)T) = \hat{x}_m(k-1)$$
  Then $\hat{x}_p[k] = \hat{x}(kT)$
  Solve
  $$\dot{P}(t) = A(t)P(t) + P(t)A^T(t)L(t)Q_cL^T(t) \text{ for } (k-1)T \le t \le kT \text{ and } P((k-1)T) = P_m(k-1)$$
  where
  $$A(t) = \tfrac{\partial q(\hat{x}(t),0,t)}{\partial x} \text{ and } L(t) = \tfrac{\partial q(\hat{x}(t),0,t)}{\partial v}$$
  Then $P_p[k] := P(kT)$
- **A posteriori update / Measurement update step**
  Identical to the discrete time EKF, see section 7

8 Particle Filter

$$x(k) = q_{k-1}(x(k-1), v(k-1))$$
$$z(k) = h_k(x(k), w(k))$$

where $x(0), \{v(\cdot)\}, \{w(\cdot)\}$ are mutually independent and can be discrete or continuous random variables.

8.1 Monte Carlo Sampling

8.1.1 Discrete Random Variables

$$s_i^n := \delta(i - y^n) = \begin{cases} 1 & \text{if } y^n = i \\ 0 & \text{otherwise} \end{cases}$$

$$E[s_i^n] = \sum_{\bar{y}^n=1}^{\bar{Y}} \delta(i - \bar{y}^n)p_y(\bar{y}^n) = p_y(i)$$

$$\boxed{s_i := \tfrac{1}{N}\sum_{n=1}^{N} s_i^n}$$

then

$$\boxed{\lim_{N\to\infty} s_i = E[s_i^n] = p_y(i) \text{ and } p_y \approx \tfrac{1}{N}\sum_{n=1}^{N} \bar{s}_i^n}$$

Change of variables

$$x = g(y) \text{ with } x \in \mathcal{X} := g(\mathcal{Y})$$

$$\boxed{p_x(j) = \tfrac{1}{N}\sum_{n=1}^{N} \delta(j - g(\bar{y}^n))}$$

Joint DRVs

$$\boxed{p_x(\zeta) \approx \tfrac{1}{N}\sum_{n=1}^{N} \delta(\zeta - \bar{x}^n)\ \forall\zeta}$$

where ζ and $\bar{x}^n$ are now vectors and $\delta(\cdot)$ refers to the vector version of the Kronecker delta, thus δ is one if all entries of its vector argument are zero.

8.1.2 Continuous Random Variables

$$s_y^n := \int_a^{a+\Delta y} \delta(\zeta - y^n)d\zeta = \begin{cases} 1 & \text{if } a \le y^n < a + \Delta y \\ 0 & \text{otherwise} \end{cases}$$

$$\boxed{p_y(\zeta) \approx \tfrac{1}{N}\sum_{n=1}^{N} \delta(\zeta - \bar{y}^n),\ \forall\zeta}$$

Change of variables

$$x = g(y) \text{ with } x \in \mathcal{X} := g(\mathcal{Y})$$

$$\boxed{p_x(\zeta) \approx \tfrac{1}{N}\sum_{n=1}^{N} \delta(\zeta - g(\bar{y}^n)),\ \forall\zeta}$$

8.2 Particle Filter

- **Init:** $x_m(0) := x(0)$
- **S1:** $x_p(k) := q_{k-1}(x_m(k-1), v(k-1))$
- **S2:**

$$z_m(k) := h_k(x_p(k), w(k))$$
$x_m(k)$defined via its PDF
$$p_{x_m(k)}(\zeta) := p_{x_p(k)|z_m(k)}(\zeta|\bar{z}(k))\ \forall\zeta$$

8.2.1 Prior Update

$$p_{x_m(k-1)}(\zeta) \approx \tfrac{1}{N}\sum_{n=1}^{N} \delta(\zeta - \bar{x}_m^n(k-1)),\ \forall\zeta$$

Thus we can find

$$p_{x_p(k)} \approx \tfrac{1}{N}\sum_{n=1}^{N} \delta(\zeta - \bar{x}_p^n(k)),\ \forall\zeta$$

where

$$\bar{x}_p^n(k) := q_{k-1}(\bar{x}_m^n(k-1), \bar{v}^n(k-1)), \text{ for } n = 1, 2, \ldots, N$$

8.2.2 Measurement Update

$$\boxed{p_{x_m(k)}(\zeta) = p_{x_p(k)|z_m(k)}(\zeta|\bar{z}(k)) \approx \sum_{n=1}^{N} \beta_n\delta(\zeta - \bar{x}_p^n(k)),\ \forall\zeta}$$

$$\beta_n = \alpha p_{z_m(k)|x_p(k)}(\bar{z}(k)|\bar{x}_p^n(k)) \qquad \alpha = \left(\sum_{n=1}^{N} p_{z_m(k)|x_p(k)}(\bar{z}(k)|\bar{x}_p^n(k))\right)^{-1}$$

Resampling Repeat N times:

- Select a random number r uniformly on $(0, 1)$
- Pick particle $\bar{n}$ such that $\sum_{n=1}^{\bar{n}-1} \beta_n < r$ and $\sum_{n=1}^{\bar{n}} \beta_n \ge r$

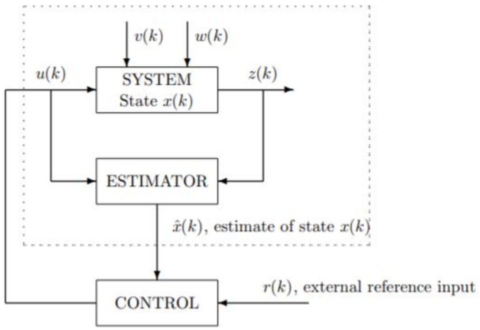
This gives N new particles $\bar{x}_m^n(k)$ which are a subset of the old particles, which now have equal weights:

$$\boxed{p_{x_m(k)}(\zeta) \approx \tfrac{1}{N}\sum_{n=1}^{N} \delta(\zeta - \bar{x}_m^n(k)),\ \forall\zeta}$$

8.2.3 Sample Impoverishment

- Through resampling we only retain a subset of the particles.
- This can lead particles to converge to the same one.
- A possible remedy is perturbing the particles after resampling
  $$\bar{x}_m^n(k) \leftarrow \bar{x}_m^n(k) + \Delta x^n(k)$$
  where $\Delta x^n(k)$ is drawn from a zero-mean, finite-variance distribution.
- One possible way of choosing the variance is
  $$\sigma_i = KE_iN^{-\frac{1}{d}}$$

9 Observer-Based Control



- The feedback control system resulting from the combination of a stable LTI observer with a stable static-gain controller is stable. If both designs are optimal, the combination of the two is optimal as well (in the sense of minimizing a quadratic cost).

$$\begin{aligned} x(k) &= Ax(k-1) + Bu(k-1) + v(k-1) \\ z(k) &= Hx(k) + w(k) \end{aligned}$$

where $v(k-1)$ and $w(k)$ are zero-mean CRVs representing noise.

- **Luenberger observer**

  $$\begin{aligned} \hat{x}(k) &= A\hat{x}(k-1) + Bu(k-1) + K(\bar{z}(k) - \hat{z}(k)) \\ \hat{z}(k) &= H(A\hat{x}(k-1) + Bu(k-1)) \end{aligned}$$

  where K is a static correction matrix that is to be designed and $A\hat{x}(k-1) + Bu(k-1)$ is what we predict the state should be according to the process mode and the current state estimate.

- This observer has the same structure as the steady-state Kalman Filter.
- The error $e(k)$ goes to zero as $k \to \infty$ and if there is no noise iff $(I - KH)A$ is stable.
- There exists such a K iff (A, HA) is detectable.
- (A, HA) is detectable iff (A, H) is detectable.
- Pole placement design: You can use the `place()` command in Matlab to find K that places the eigenvalues of the error dynamics corresponding to the observable modes at desired locations.

9.1 Static State-Feedback Control

$$\begin{aligned} x(k) &= Ax(k-1) + bu(k-1) \\ z(k) &= x(k) \end{aligned}$$

where we concentrate on the controller development by assuming perfect knowledge about the state.

$$u(k) = Fx(k) = Fz(k)$$

Thus the closed loop dynamics are:

$$x(k) = (A + BF)x(k-1)$$

Hence the system is stable iff $A + BF$ is stable. One can find such an F only if (A, B) is stabilizable.

- Pole placement design
- LQR

$$J_{LQR} = \sum_{k=0}^{\infty} x^T(k)\bar{Q}x(k) + u^T(k)\bar{R}u(k)$$

where $\bar{Q} = \bar{Q}^T \geq 0$ and $\bar{R} = \bar{R}^T > 0$ are weighting matrices.
Assuming (A, B) stabilizable and (A, G) detectable with $\bar{Q} = GG^T$ the optimal stablizing controller is:

$$F = -(B^T PB + \bar{R})^{-1} B^T PA$$

where $P = P^T \geq 0$ is the unique positive semidefinite solution to the DARE.

$$P = A^T PA + \bar{Q} - A^T PB(B^T PB + \bar{R})^{-1} B^T PA$$

9.2 Separation Principle

$$\begin{aligned} x(k) &= Ax(k-1) + Bu(k-1) \\ z(k) &= Hx(k) \end{aligned}$$

$$\begin{aligned} \hat{x}(k) &= A\hat{x}(k-1) + Bu(k-1) + K(z(k) - \hat{z}(k)) \\ \hat{z}(k) &= H(A\hat{x}(k-1) + Bu(k-1)) \\ u(k) &= F\hat{x}(k) \end{aligned}$$

Assume that $(I - KH)A$ and $(A + BF)$ are stable. Question: Is the overall system stable.
The error dynamics are:

$$e(k) = (I - KH)Ae(k-1)$$

And thus the state dynamics are:

$$x(k) = (A + BF)x(k-1) - BFe(k-1)$$

Therefore the closed loop dynamics can be described as

$$\begin{bmatrix} x(k) \\ e(k) \end{bmatrix} = \begin{bmatrix} A + BF & -BF \\ 0 & (I - KH)A \end{bmatrix} \begin{bmatrix} x(k-1) \\ e(k-1) \end{bmatrix}$$

- The eigenvalues of the closed-loop dynamics are given by the eigenvalues of $(I - KH)A$ and $(A + BF)$. Therefore the overall system is stable. This is called the **separation principle**.
- Including the noise in the analysis does not affect stability.
- Under mild conditions the above analysis generalizes to the time-varying case.
- In general, the separation principle does **not** hold for nonlinear systems.

9.3 Separation Theorem

$$\begin{aligned} x(k) &= Ax(k-1) + Bu(k-1) + v(k-1) \qquad & v(k-1) &\sim \mathcal{N}(0, Q) \\ z(k) &= Hx(k) + w(k) & w(k) &\sim \mathcal{N}(0, R) \end{aligned}$$

The control objective is to find the control policy that minimizes

$$J_{LQG} = \lim_{N\to\infty} E\left[\frac{1}{N}\sum_{k=0}^{N-1}(x^T(k)\bar{Q}x(k) + u^t(k)\bar{R}u(k))\right]$$

where $u(k)$ can depend on current and past measurements $z(1:k)$ (causal strategy).
Then the optimal strategy:

- Design a steady-state KF (The filter does not depend on $\bar{Q}$ and $\bar{R}$). The filter provides an estimate $\hat{x}(k)$ of $x(k)$.
- Design an optimal state-feedback strategy $u(k) = Fx(k)$ for the determinable LQR problem.

  $$x(k) = Ax(k-1) + Bu(k-1)$$

  that minimizes

  $$J_{LQR} = \sum_{k=0}^{\infty} x^T(k)\bar{Q}x(k) + u^T(k)\bar{R}u(k)$$

  The feedback gain does not dpeend on the noise statistics Q and R.

- Put both together.

This control design is called Linear Quadratic Gaussian (LQG) control.