Markov Processes :

Ex: $P(X_n = S_n \mid X_{0:n-1} = S_{0:n-1}) = P[X_n = S_n \mid X_{n-1} = S_{n-1}] \rightarrow$ Def of markov chain.

$X_n = f_n(X_{n-1}, U_n) \rightarrow \{U_n\} \rightarrow$ independent sequence

time, lat, last incoming noise

Ind. of $X_0, U_1, \ldots U_{n-1}$

$\{X_n\}$ is markov; It obeys

$X_1 = f_1(X_0, U_1)$

$X_{n-1} = f_{n-1}(X_{n-1}, U_{n-1})$

summing over all conditions

$$P[X_n = S_n \mid X_{0:n-1} = S_{0:n-1}] = \sum_{U: S_n = f_n(S_{n-1}, U)} P[U_n = U \mid X_{0:n-1} = S_{0:n-1}]$$

Since U_n is independent of X variables

$$= \sum_{U: S_n = f_n(S_{n-1}, U)} P[U_n = U]. \rightarrow \text{depends only on } S_{n-1} \text{ only.}$$

Properties of Markov chains

i) Algorithmic Properties of M.C.

→ to compute efficiently filters (Estimators).

ii) Asymptotic Properties (Ergodic properties) → in pdf

→ Simulation & optimization using markov chains.

1. Dynamic Programming – used in Signal processing & process control

You have:

Signal X_n : markov

W_n : noise.

$Y_n = h_n(X_n, W_n) \rightarrow W_N \rightarrow$ independent sequence → white noise.

$h_n \rightarrow$ channel function

n, k, N: 0 : N

0 ↔ N

$S_{0:N} \rightarrow$ state of signal by markov (encoder) chain

Joint probability

$$J_N(S_{0:N}) = P(Y_{0:N}, S_{0:N}) \rightarrow \text{Joint probability by maximizing } S_{0:N} \text{ (dummy)}$$

$$= \underbrace{P(Y_{0:N} \mid X_{0:N} = S_{0:N})}_{\prod_{k=0}^{N} \underbrace{P(Y_k \mid X_k = S_k)}_{B_k(S_k)}} \times \underbrace{P(S_{0:N})}_{P_0(S_0)\,P(S_1|S_0)\,P(S_2|S_1)\ldots P(S_N|S_{N-1})}$$

Rearranging them in order of time '

$$= P(s_0) \cdots g_n(s_{n-1}, s_n) \cdots g_N(s_{N-1}, s_N)$$ $g_n \to$ function.

Problem : find the sequence $\hat{S}_{0:N} = \underset{S_{0:N}}{\text{Arg max}} J_N(S_{0:N})$ → estimated signal

We cannot maximize J_N because S_N is overlapping in g_n and $|\$|^N$ is thousands of bits.

Hence we resort to dynamic programming tool.

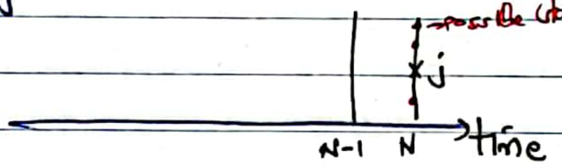
→ let $\hat{J}_N$ = Maximum value of all sequences

$$\hat{J}_N = \max_{S_{0:N}} J_N(S_{0:N}) \qquad S_k \in \$ = \{1, \dots r\}$$ → possible states

$$= \max_j \bar{J}_N(j) = \max_j \max_{S_{0:N-1}} J_N(S_{0:N-1}, S_N = j)$$

$\bar{J}_N(j)$ → matrix

max all sub-sequences ending at j



→ possible states (ending of possible seq. that can end at time N).

- we are picking ending at N at state j

Matrix $\bar{J}_N$

Recursion For $\bar{J}_N(j)$

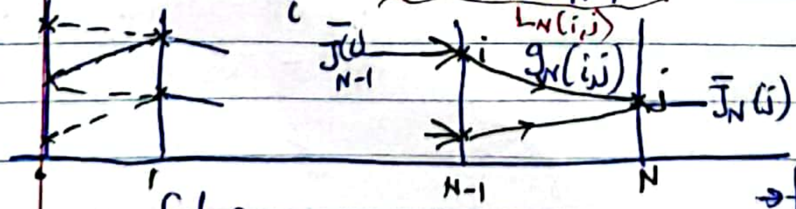
$$\bar{J}_N(j) = \max_i \max_{S_{0:N-2}} \left(J_N(S_{0:N-2}, S_{N-1} = i, S_N = j) \right)$$ ← $g_0(x_0) \, g_{N-1}$

i (S_{N-2}, S_{N-1})

j $g_N(S_{N-1}, S_N)$

$$= \max_i \underbrace{g_N(i,j) \bar{J}_{N-1}(i)}_{L_N(i,j)}$$

each state one arrow



i → any possible past value i

→ Considering all contribution of $\bar{J}(i)$

→ take the best incoming & outgoing $(\bar{J}(j))$

$g_N(i,j)$ = fxn of ~~from~~ noise, and encode S_N

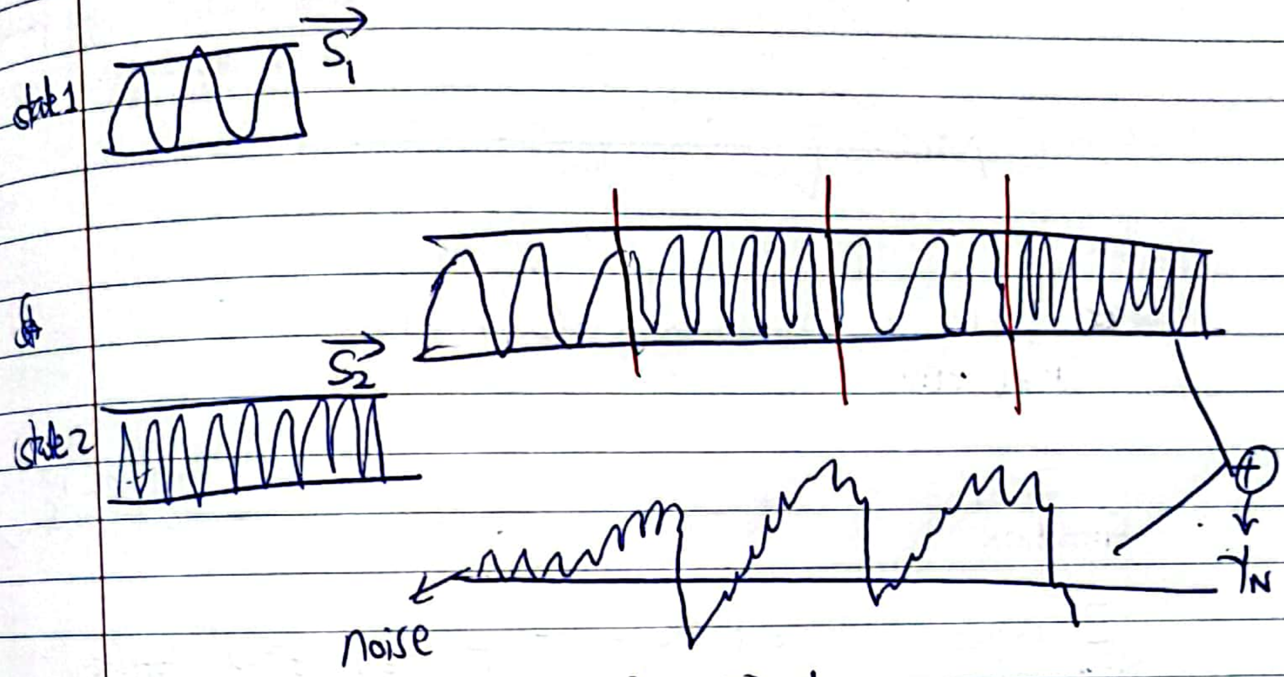
→ At the end one arrow on each possible state - hence $|\$|^N \to N$ Converted from $\$^N$ to just N

$g_n(i,j) = P(Y_N \mid S_N = j)\, P(S_N = j \mid S_{N-1} = i)$

prob. of flipping from i to j

→ Most probable sequence is selecting j having $J_N(j)$

$\hat{S}_{N-1}(j) = \arg\max_i L_N(i,j)$



$\vec{Y_N} = h_n(\vec{X_n}, \vec{W_N})$. h_n → summing

otherwise h_n → linear operator (matrix).

Viterbi algo:

→ Red transition of codes: = high state - high f
= low - low f

→ Red down → received signal (noise + signal).

Summary: $(X_{1:N}) \rightarrow Y(1:N) \rightarrow \max_S p(Y, S)$ → minimum block error (→ Viterbi (Dynamic)).

↓

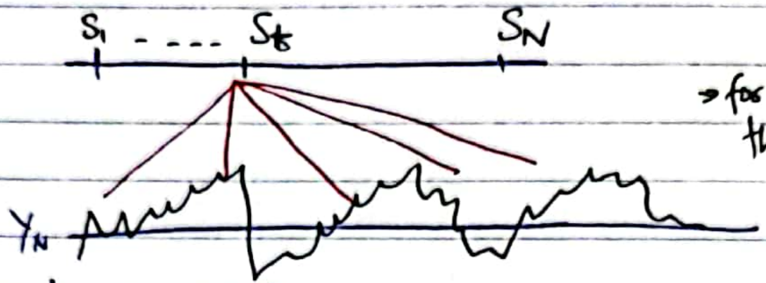
$\left\{ \max P(S_t \mid Y_{1:N}) \right.$ → min Prob. symbol error.
$t = 1:N$ → spectral processing.

F-B algo

$P(A|B) = \frac{P(A \cap B)}{P(B)}$

* **Forward - Backward Algorithm (BCJR)** forget (no interest)

$P(S_t = j \mid Y_{1:N}) = P(S_t = j ; Y_{1:N}) / \cancel{P(Y_{1:N})}$ → select j having max Prob.



→ for each t you process the all block.

So proportional to

$\propto \sum_{S_{1:t+1}, t+1:N} P(S_{1:t}, S_t = j, S_{t+1:N} ; Y_{1:N})$ → all blocks

$\sum_{S_{1:t+1}, t+1:N} P_1(S_1) \dots P(Y_t | S_t) P(\underset{S_t = j}{S_t} | S_{t-1}) \dots P(Y_N | S_N) P(S_N | S_{N-1})$

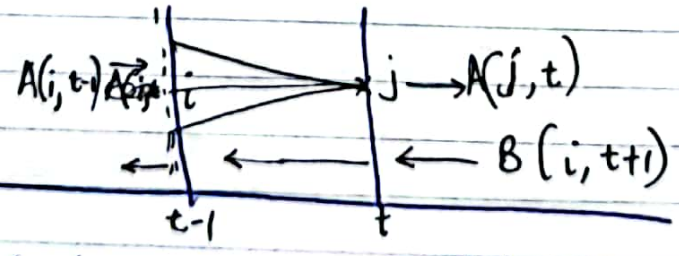
$\underbrace{\sum_{S_{1:t+1}} P_1(S_1) \dots P(Y_t|S_t) P(S_t = j | S_{t-1})}_{A(j,t)} \underbrace{\sum_{S_{t+1:N}} P(S_{t+1} | S_t = j) P(Y_{t+1} | S_{t+1}) \dots P(Y_N|S_N) P(S_N | S_{N-1})}_{B(j,t)}$.

A(j,t) → past values

B(j,t) → future values

$A(j,t) = \sum_i A(i, t-1) P(Y_t | S_t = j) P(S_t = j | S_{t-1} = i)$.

$B(j,t) = \sum_i (P(S_{t+1} = j | S_t = i) \; P(Y_{t+1} | S_{t+1} = j) B(i, t+1))$ → Backward.



- reduced to recursive computation to minimize errors in (symbol errors) communication to decode noisy speech.

(4) Gaussian Markov process - ~~channel~~ filtering
Kalman.

(A) Gaussian M. Process → 2^{nd} pdf.

$U_n \to$ independent.

Evolution equation → $\vec{X}_n = A_n \vec{X}_{n-1} + \vec{U}_n$ where $X_0, U_0 \dots U_n$ are jointly gaussian.

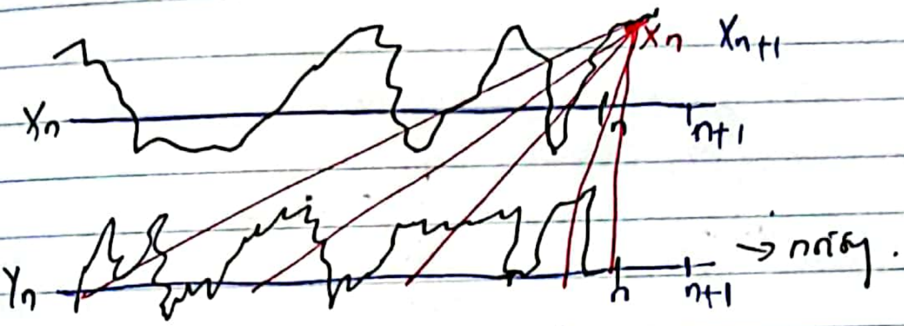
function of its past + Incoming independent r.v.e

To recover $\vec{X}_n$ we limit it to linear dependance.

→ Similarly: $\vec{Y}_n = C_n \vec{X}_n + \vec{W}_n$ → state estimation: Kalman filter (B).

observation eqn. →

$\vec{X}_n = f_n(\vec{X}_{n-1}, \vec{U}_n)$ → linearization → Extended ~~ckd~~ Kalman filter
$Y_n = h_n(X_n, W_n)$ or → use particle filters.



→ we use conditional Expectation.

$\hat{X}_{n|n} = \mathbb{E}[X_n | Y_{0:n}]$. ⇒ Best Mean Square Estimator of $X_n | Y_{0:n}$

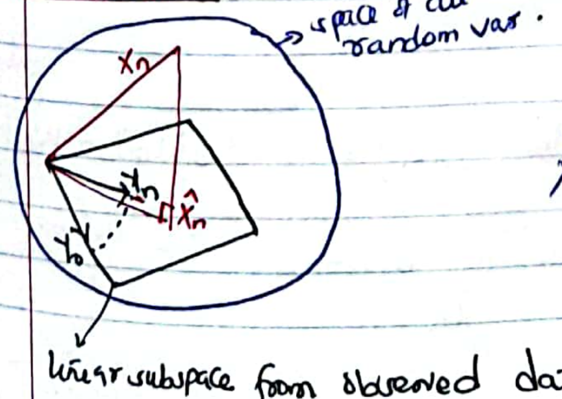
⇒ It minimizes $\underset{}{\text{Arg min}}\ \mathbb{E}[|X_n - \phi(Y_{0:n})|^2]$ → error

→ In Gaussian model:
when X_n & Y_n are jointly gaussian, $\mathbb{E}$ is linear.

Y_n → all data 0:n

$\hat{X}_n = \mathbb{E}[X_n | \underbrace{Y_{0:n}}_{\underline{Y}_n}] = L_n \underline{Y}_n$ → linear operator to the all collected data ; $L_n = \underbrace{\mathbb{E}[X_n \cdot \underline{Y}_n^T] \cdot \mathbb{E}[\underline{Y}_n \cdot \underline{Y}_n^T]^{-1}}_{\text{matrix.}}$ → covariance

Geometrical interpret: Linear Projection



$\hat{X}_{n|n} = \text{Proj}(X_n | Y_{0:n})$ → minimizes euclidian Distance

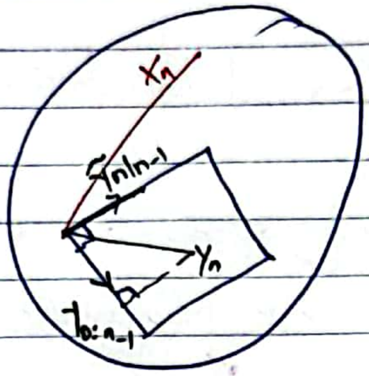
Linear subspace from observed data.

Innovation process.

Hence $\hat{X}_{n|n} = \text{Proj}(X_n | Y_{0:n}) = \text{Proj}(X_n | Y_{0:n-1}, \tilde{Y}_{n|n-1})$

$Y_n - \text{Proj}(Y_n | Y_{0:n-1})$

Proj Y_n on its past = Y_n - Predictable past.



$\tilde{Y}_n \to$ (innovation) $\to$ last data.

$$\underset{\text{Estimated}}{\hat{X}_{n|n}} = \underbrace{\text{Proj}(X_n | Y_{0:n-1})}_{\substack{\text{predicted} \\ \hat{X}_{n|n-1}}} + \underbrace{\text{Proj}(X_n | \tilde{Y}_{n|n-1})}_{K_n \cdot \tilde{Y}_{n|n-1}}$$

gain filter

Compute the two parts by recursion:

$A_n \to$ parameters of system (deterministic).

$\hat{X}_{n|n-1} = \text{Proj}(X_n | Y_{0:n-1})$

from evolution eqn:

$$= \text{Proj}(A_n X_n + U_n | Y_{0:n-1}) = A_n \overbrace{\text{Proj}(X_n | Y_{0:n-1})}^{\hat{X}_{n-1|n-1}} + \text{Proj}(U_n | Y_{0:n-1})$$

$\to$ independent $= 0$; 0 mean

Observation $\leftarrow Y_n = C_n X_n + W_n$

hence U_n is independent of $(U_0, \ldots U_{n-1})$ \$

$\hat{X}_{n|n} = \hat{X}_{n|n-1} + K_n \tilde{Y}_n$

(*) $\hat{X}_{n|n-1} = A_n \hat{X}_{n-1|n-1}$

orthogonal $\perp \to \text{Cov} = 0$

(*) $\tilde{Y}_n = Y_n - \text{Proj}(Y_n | Y_{0:n-1}) = Y_n - \text{Proj}(C_n Y_n + W_n | Y_{0:n-1})$

$$= Y_n - C_n \text{Proj}\underbrace{(X_n | Y_{0:n-1})}_{\hat{X}_{n|n-1}}.$$

$C_n \to$ Sensor response to respond linear

Developing $\tilde{Y}_n$

$$\tilde{Y}_n = C_n X_n + W_n - C_n \hat{X}_{n|n-1} = C_n \underbrace{(X_n - X_{n|n-1})}_{\text{state prediction error } \tilde{X}_{n|n-1}} + W_n \perp\!\!\!\perp$$

$\xrightarrow{\text{orthogonal}}$: uncorrelated $\perp$

N.B. $P_{proj}(W_n | Y_{0:n-1}) = E[W_n \cdot \underbrace{Y^T_{0:n-1}}_{\text{0 mean}}] \, E[Y_{0:n-1}, Y^T_{0:n-1}]^{-1}$

innovation

$$\circledast \quad K_n = \underbrace{E[X_n \cdot \tilde{Y}_n^T]}_{①} \, \underbrace{E[\tilde{Y}_n \cdot \tilde{Y}_n^T]^{-1}}_{②}$$

project X_n on the last term only not all

$\perp$ here Cov = 0

$$① \quad E[X_n \cdot (C_n \tilde{X}_{n|n-1} + W_n)^T] = \underbrace{E[\tilde{X}_{n|n-1} \cdot \tilde{X}^T_{n|n-1}]}_{P_{n|n-1}} C_n^T$$

$X_n = \hat{X}_{n|n-1} + \tilde{X}_{n|n-1}$

$\rightarrow$ sum of cov = cov of sum

$$② \quad Cov(\tilde{Y}_n) = Cov(C_n \tilde{X}_{n|n-1} + W_n) = C_n \underbrace{Cov(\tilde{X}_{n|n-1})}_{P_{n|n-1}} C_n^T + Cov(W_n) \rightarrow R_w(n).$$

$\perp$

$$* \quad P_{n|n-1} = Cov(\tilde{X}_{n|n-1}) = Cov(X_n - \hat{X}_{n|n-1}) = Cov(A_n X_{n-1} + U_n - A_n \hat{X}_{n-1|n-1})$$

$$= Cov(A_n \underbrace{[X_{n-1} - \hat{X}_{n-1|n-1})}_{\text{estimating error | filter}} + U_n).$$

$\tilde{X}_{n-1|n-1}$ $\perp$

$$= Cov(A_n \tilde{X}_{n-1|n-1}) = A_n \underbrace{Cov(\tilde{X}_{n-1|n-1})}_{P_{n-1|n-1} \rightarrow \text{Cov matrix of estimator}} A_n^T$$

$$P_{n|n} = Cov(X_n - \hat{X}_{n|n}) = Cov(X_n - [\hat{X}_{n|n-1} + K_n \tilde{Y}_{n-1}])$$

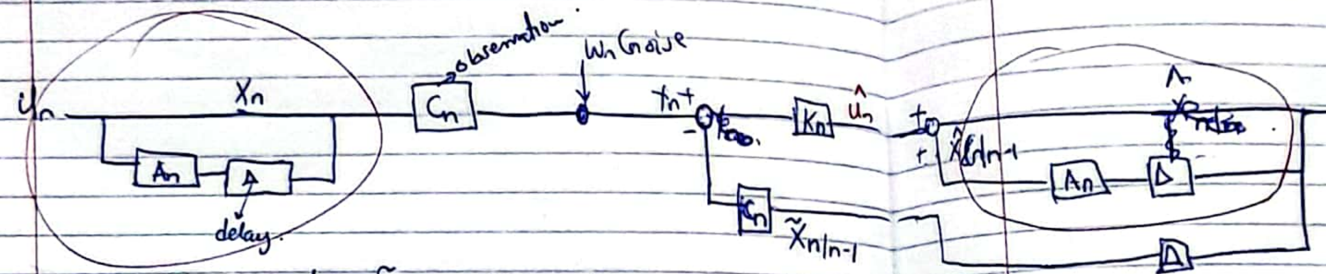
$\tilde{Y}_{n-1} = C_n \tilde{X}_{n|n-1} + W_n$

$$= Cov[(I - K_n C_n) \tilde{X}_{n|n-1} - K_n W_n]$$

$\perp \rightarrow$ orthogonal ; Cov = 0

to state.

$$= (I - K_n C_n) \underbrace{Cov(\tilde{X}_{n|n-1})}_{P_{n|n-1}} (I - K_n C_n)^T + K_n \underbrace{Cov(W_n)}_{R_W \rightarrow \text{Cov of noise.}} K_n^T$$



$\tilde{X}_{n|n-1} = X_n - \hat{X}_{n|n-1}$

$= X$

- Compute innovation $\tilde{Y}_n = Y_n - C_n \tilde{X}_{n|n-1}$

$\hat{U}_n = K_n \times$ innovation

- filter acts as model of the signal
- filter computes which input was sent
- Gives the best estimation of the state.