Part 5: Partially Observed Markov Decision Processes

Aim: This part covers discrete time stochastic optimization of systems whose state is observed via noisy measurements. The key idea is to convert the partially observed problem to a fully observed problem – the resulting fully observed problem is in terms of the information state. Then *stochastic dynamic programming* (SDP) can be used to solve the problem. The resulting SDP is usually infinite dimensional apart from the HMM case (POMDP) and Linear Quadratic Gaussian (LQG) case.

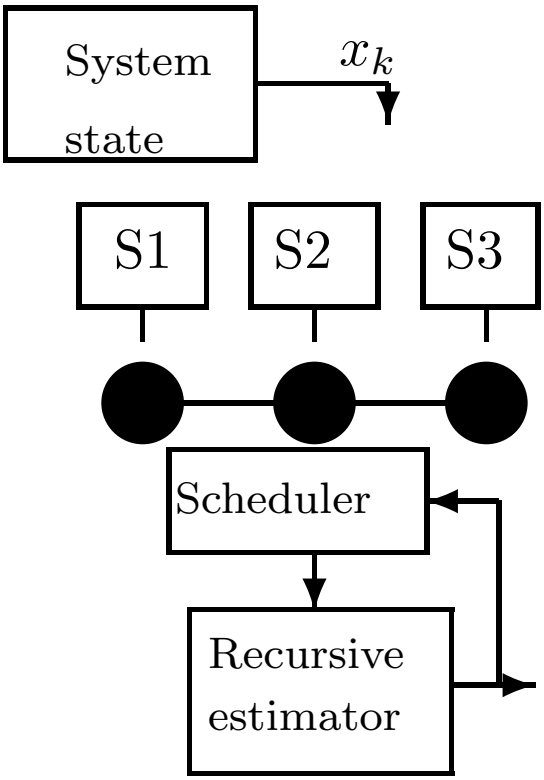
Example 1: Sensor Adaptive Signal Processing

"Active Sensing"

$$x_{k+1} = A(x_k) + w_k$$
$$y_k = C(x_k, u_k) + v_k$$

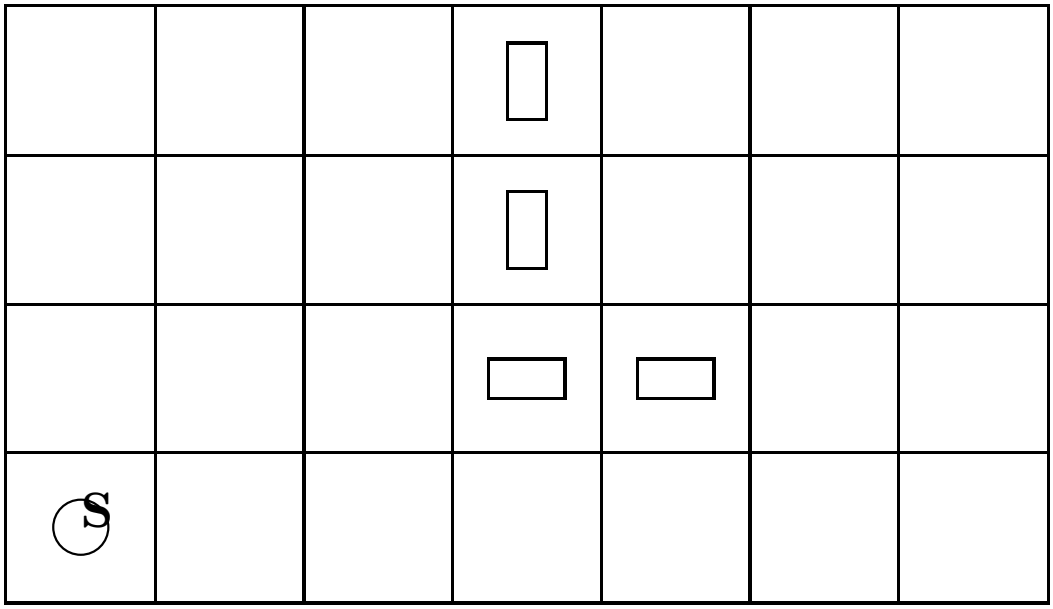
$u_k \in \{1, 2, 3\}$ denotes choice of sensor.

Which single sensor to pick at time k to optimize a cost function? e.g. $\mathbb{E}_\mu\{\sum_{k=1}^{N} \|x_k - \mathbb{E}\{x_k|Y_k, \mu\}\|^2\}$?



- Dynamically control the **type** and **flow** of information.
  Then estimate resulting signal.
- Motivation: Communication, energy or computational constraints

Example 2: Optimal Search for A Markovian Target



State: Markovian target $x_k \in \mathcal{X}$ – Trans prob P.
In addition termination state. So $\mathcal{X} = \{1, \ldots, X, t\}$.
Action: u_k: Which cell to search?
Obs: $y_k \in \{f, \bar{f}\}$. Observation probabilities: $p(y_k|x_k, u_k)$:
e.g $p(y_k = f|x_k = 1, u_k = 2) = 0$,
$p(y_k = f|x_k = 1, u_k = 1) = P_d$ (prob of detection).
Cost: Minimize $\mathbb{E}\{\sum_{k=1}^{N} c(x_k, u_k)\}$.
Examples: (a) Search Delay cost

$$c(x_k, u_k) = 1, \quad c(x_k = t, u_k) = 0$$

(b) Probability of Detection cost [Eagle 1984]:

$$c(i, u) = P(y_k = f|x_k = i, u_k = u), \quad c(x_k = t, u_k) = 0$$

1 The Problem

1. **Discrete-time dynamic system**

x_0 is random with pdf P_{x_0}.

$$\begin{aligned} \text{State:} \quad & x_{k+1} = A_k(x_k, u_k, w_k), \quad k = 0, 1, \ldots, N-1 \\ \text{Observations:} \quad & y_k = C_k(x_k, v_k, u_k), \quad y_0 = C_0(x_0, v_0) \end{aligned} \tag{1}$$

Process Noise w_k and measurement noise: v_k iid

Information vector: I_k denotes info at time k. $I_0 = y_0$

$$I_k = (y_0, y_1, \ldots, y_k, u_0, u_1, \ldots, u_{k-1}), \quad k = 1, \ldots, N-1$$

2. **Policy class**: Consider *admissible* policy

$$\pi = \{\mu_0, \ldots, \mu_{N-1}\} \text{ where } u_k = \mu_k(I_k) \in U_k$$

3. **Cost function**: additive cost function

$$J_\pi = \mathbb{E}\left\{ c_N(x_N) + \sum_{k=0}^{N-1} c_k(x_k, \mu_k(I_k)) \right\} \tag{2}$$

(Note expectation is wrt x_0, w_k, v_k, $k = 0, \ldots, N-1$)

Aim: Compute optimal policy $J^* = \min_{\pi \in \Pi} J_\pi$

Remarks:
1. In fully observed case, $u_k = \mu_k(x_k)$. Here u_k depends on all past observations and controls.
2. x_0 is random variable with pdf P_{x_0} (prior) unlike fully observed case.
Cost function involves expectation wrt x_0.

Timing:
(i) At time k, $x_k = A_{k-1}(x_{k-1}, u_{k-1}, w_{k-1})$ is generated. This is observed in noise as y_k.
(ii) Controller uses y_k to generate control signal u_k.
(iii) Set $k = k + 1$ and return to Step (i).

Key Idea: The key idea is to transform the partially observed problem (in state x_k) to a fully observed problem in a new state π_k.
π_k is called the "information state".
π_k is simply the filtered density of the state.

Then compute optimal control via dynamic programming. Similar approach for sensor-scheduling problems

2 Application Examples

2.1 Linear Quadratic Gaussian (LQG) Control

Partially observed problem: Assume $x_0 \sim N(0, S)$

$$x_{k+1} = A_k x_k + B_k u_k + w_k, \quad w_k \sim N(0, M_k)$$
$$y_k = C_k x_k + v_k, \quad v_k \sim N(0, N_k)$$

Cost function: $Q_k \geq 0$ and $R_k > 0$.

$$J_\pi = \mathbb{E}\left\{ x'_N \, Q_N \, x_N + \sum_{k=0}^{N-1} u'_k R_k u_k + x'_k Q_k x_k \right\}$$

Optimal solution has a separation principle:

$$\text{LQG control} = \text{state estimator} + \text{LQ control}$$

2.2 Partially Observed Markov Decision Process (POMDP)

x_k is a X state Markov chain. $P(x_0 = i) = \pi_0(i)$
Transition prob: $P(x_{k+1} = j | x_k = i, u_k = u) = P_{ij}(u)$
Observations: (discrete-valued) $y_k \in \{O_1, \ldots, O_M\}$
$b_i(O_m) = P(y_k = O_m | x_k = i)$

Applications: Target tracking, Sensor scheduling, etc.

Benchmark Example: Machine Replacement
State: $x_k \in \{0, 1\}$ – machine state
$x_k = 0$ operational; $x_k = 1$ failed.

Control: $u_k \in \{0, 1\}$.
$u_k = 0$ keep machine; $u_k = 1$ replace by new one

Trans prob matrices: Let $\theta = P(x_{k+1} = 1 \mid x_k = 0)$.

$$P(0) = \begin{bmatrix} 1-\theta & \theta \\ 0 & 1 \end{bmatrix}, \quad P(1) = \begin{bmatrix} 1 & 0 \\ 1 & 0 \end{bmatrix}$$

Observations: $P(y_k = i | x_k = i) = q$, $0.5 < q < 1$.
Cost: Minimize $\mathbb{E}\{\sum_{k=0}^{N-1} c(x_k, u_k)\}$

3 Transformation to Fully Observed Problem

Information vector satisfies: $I_0 = y_0$,
$I_{k+1} = (I_k, y_{k+1}, u_k), \quad k = 0, 1, \ldots, N-2$

$$\text{Now cost is } \mathbb{E}\left\{ c_N(x_N) + \sum_{k=0}^{N-1} c_k(x_k, u_k \right\}$$

$$= \mathbb{E}\left\{\mathbb{E}\{c_N(x_N)|I_N\}\right\} + \sum_{k=0}^{N-1} \mathbb{E}\left\{\mathbb{E}\{c_k(x_k, u_k) \mid I_k, u_k\}\right\}$$

$$= \mathbb{E}\{\bar{c}_N(I_N) + \sum_{k=0}^{N-1} \bar{c}_k(I_k, u_k)\}$$

$$\text{where } \bar{c}_k(I_k, u_k) = \int_{\mathbb{R}} c_k(x_k, u_k)\, p(x_k|I_k, u_k) dx_k$$

Information state: $\pi_k(x) = p(x_k|I_k, u_k)$.
It denotes the filtered density of the state.

$$\pi_{k+1} = T(\pi_k, y_{k+1}, u_k)$$

$$\pi_{k+1}(x) = \frac{\int_{\mathbb{R}} p_v(y_{k+1}|x) p_w(x|\zeta, u_k) \pi_k(\zeta) d\zeta}{\int_{\mathbb{R}} \int_{\mathbb{R}} p_v(y_{k+1}|x) p_w(x|\zeta, u_k) \pi_k(\zeta) d\zeta}$$

4 DP Solution of Partially Observed Control

Summary: Equivalent fully observed problem is:

$$\text{Info state: } \pi_{k+1} = T(\pi_k, y_{k+1}, u_k)$$

$$\text{Cost:} J = \mathbb{E}\left\{\sum_{k=0}^{N-1} \bar{c}_k(\pi_k, u_k) + \bar{c}_N(\pi_N)\right\}$$

$$\text{where } \bar{c}_k(\pi_k, u_k) = \int_{\mathbb{R}} c_k(x, u_k)\pi_k(x)dx$$

Dynamic Programming yields: $J_N(\pi) = \bar{c}_N(\pi)$,

$$J_k(\pi) = \min_{u_k} [\bar{c}_k(\pi, u_k) + \mathbb{E}\{J_{k+1}(\pi_{k+1})|\pi_k = \pi, u_k\}]$$

$$= \min_{u_k} \Bigg[\bar{c}(\pi_k, u_k) + \int_{\mathbb{R}} J_{k+1}(T(\pi_k, u_k, y_{k+1})) \, p(y_{k+1}|\pi_k = \pi, u_k)dy_{k+1}\Bigg]$$

Optimal policy $u_k^* = \mu_k^*(\pi_k)$, $\pi^* = \left\{\mu_0^*, \ldots, \mu_{N-1}^*\right\}$
In the above DP equation,

$$p(y_{k+1}|\pi_k, u_k) = \int_{\mathbb{R}} p(y_{k+1}|x_{k+1})p(x_{k+1}|\pi_k, u_k)\, dx_{k+1}$$

$$= \int_{\mathbb{R}}\int_{\mathbb{R}} p_v(y_{k+1}|x_{k+1})\, p_w(x_{k+1}|x_k, u_k)\, \pi_k(x_k)dx_{k+1}dx_k$$

i.e. it is the normalization term in the filtering eqn.

Remarks:

1: For a finite dimensional controller we need:
(i) A finite dimensional filter for $\pi_k(x)$ (finite dimensional sufficient statistic).
(ii) Explicit solution to DP

Only 2 cases are known to have finite dimensional controllers:
LQG: Kalman Filter + LQ controller
HMM: HMM Filter + finite dimensional soln to DP.

2: Information state for HMM is continuous valued. Sondik (1973) developed a finite dimensional HMM controller. See Lovejoy for state-of-the-art HMM control algorithms.

3: In general stochastic control, one needs stability for the expectations to exist. This is difficult to prove.

A more rigorous derivation of above DP equation is via a change of measure, see Elliott's book.

4: In continuous time, similar philosophy is used:
(i) Cost function is written as fully observed cost function in terms of information state.
(ii) Information state evolves according to Zakai eqn.

5 HMM Control Problem (POMDP)

Information state: For $j = 1, \ldots, S$

$$\pi_{k+1}(j) = \frac{\sum_{i=1}^{N} \pi_k(i) P_{ij}(u_k) b_j(y_{k+1})}{\sum_i \sum_j \pi_k(i) P_{ij}(u_k) b_j(y_{k+1})}$$

In matrix vector notation this reads

$$\pi_{k+1} = \frac{B(y_{k+1}) P'(u_k) \pi_k}{\mathbf{1}' B(y_{k+1}) P'(u_k) \pi_k}$$

$$\text{Cost is:} \qquad J = \mathbb{E}\left[\sum_{k=0}^{N-1} \bar{c}_k(\pi_k, u_k) + \bar{c}_N(\pi_N)\right]$$

$$\text{where } \bar{c}_k(\pi_k, u_k) = \sum_{i=1}^{N} c_k(i, u_k) \pi_k(i)$$

DP yields: $J_k(\pi)$

$$= \min_{u_k} \Bigg[\bar{c}_k(\pi_k, u_k) + \sum_{m=1}^{M} J_{k+1}\left(T(\pi_k, u_k, y_{k+1})\right) p(y_{k+1} = O_m | \pi_k, u_k) \Bigg]$$

$$= \min_{u_k} \Bigg[\bar{c}_k(\pi_k, u_k) + \sum_{m=1}^{M} J_{k+1}\left(\frac{B(O_m) P'(u_k) \pi_k}{\mathbf{1}' B(O_m) P'(u_k) \pi_k}\right) \mathbf{1}' B(O_m) P'(u_k) \pi_k \Bigg]$$

Finite Dimensional HMM Controller

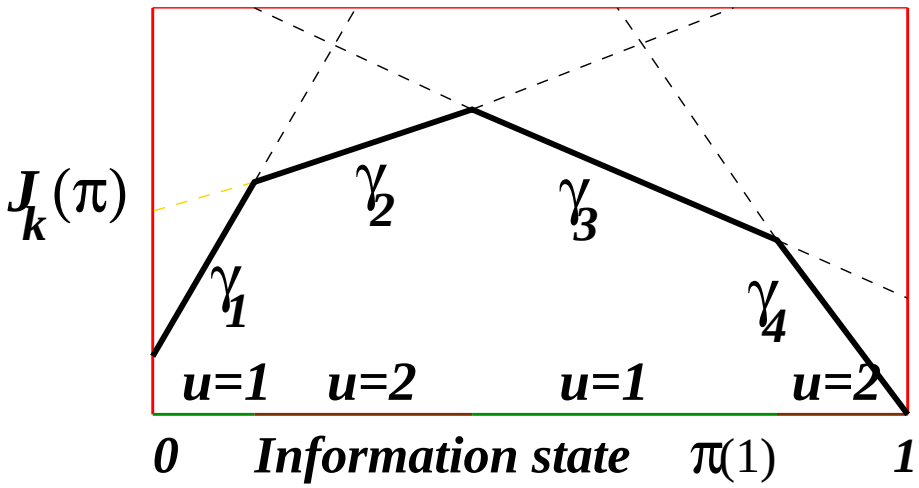
$J_k(\pi)$ is positively homogeneous, i.e. $cJ_k(\pi) = J_k(c\,\pi)$ for any $c > 0$. Therefore

$$J_k(\pi) = \min_{u_k} \left[\bar{c}_k(\pi_k, u_k) + \sum_{m=1}^{M} J_{k+1}\left(B(O_m)A'(u_k)\pi_k\right) \right]$$

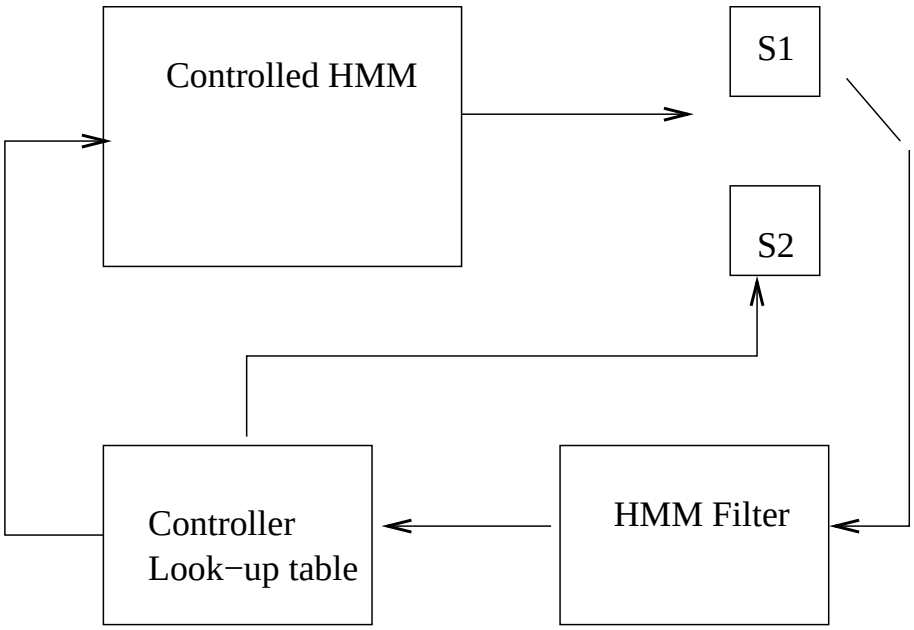
Sondik showed by induction J_k is piecewise linear and concave:

$$J_k(\pi_k) = \min\left[\gamma_k^1\pi_k, \gamma_k^2\pi_k, \ldots, \gamma_k^{n_k}\pi_k\right]$$

where $\gamma_k^1, \ldots, \gamma_k^{n_k}$ are S dimensional vectors and n_k is a finite number for any finite k.



Structure of HMM Scheduler/Controller



Remarks: (i) Solving POMDPS is PSPACE hard. Number of vectors grow exponentially.
(ii) Lovejoy proposed an ingenious sub-optimal approach.
(iii) In AI – numerous recent algorithms have been proposed.

6 LQG Control

$$x_{k+1} = A_k x_k + B_k u_k + w_k, \quad x_0 \sim N(0, S)$$
$$y_k = C_k x_k + v_k$$
$$J_\pi = \mathbb{E}\left\{ x_N' \, Q_N \, x_N + \sum_{k=0}^{N-1} u_k' R_k u_k + x_k' Q_k x_k \right\}$$

Assumptions: $Q_k \geq 0$ and $R_k > 0$, w_k, v_k white

Result: J_k is quadratic in x.

$$J_k(x) = x_k' K_k x_k + s_k \quad k = N-1, \ldots, 0$$

where K_k satisfies same backward Riccati as LQ case.

Optimal control law is $u_k^* = L_k x_{k|k}$ where

$$L_k = -\left(B_k' K_{k+1} B_k + R_k\right)^{-1} B_k' K_{k+1} A_k$$

Finally $x_{k|k}$ and $\Sigma_{k|k}$ are given by Kalman filter.

Separation Principle holds

Remarks on Optimal cost $J_0(x_{0|0})$:

1. Indpt of R_k and C_k.
2. Also K_k and L_k are indpt of R_k and C_k (certainty equivalence), see [AM89]

Discussion

1. Key idea: Reformulate partially observed problem as a fully observed problem in the information state.

2. LQG and HMM have finite dimensional controllers

3. For general stochastic control, solve DP recursion numerically.
(i) Discretize the state space x_k.
(ii) Discretize the observations y_k.
(iii) Discretize the information state π_k
Then we have a finite dimensional DP approximation.

Note in HMM case (i), (ii) already hold.

7 Sensor Scheduling Problem

7.1 General Problem

Given the system and measurement equation:

$$x_{k+1} = f_k(x_k, u_k^P, w_k), \quad k = 0, 1, \ldots, N$$
$$y_k = h_k(x_k, u_k^M, v_k)$$

Cost function:

$$J = \psi(u_0^M)$$
$$+ \mathbb{E}\{\sum_{k=0}^{N-1} l_k(x_k, u_k^P, u_{k+1}^M) + l_N(x_N, u_N^P) + \phi(x_{N+1})$$

Aim: Find controls $u_k^P(Y_k)$, $u_k^M(Y_k)$ to min J.

Discussion Above problem is a joint scheduling/control problem. The choice of sensor u_k^M, affects the control u_k^P and hence the state x_k.

Example: Sensor Scheduling Estimation Problem

$$x_{k+1} = f(x_k, w_k)$$

M noisy sensors $y_k^{(i)} = h(x_k, i) + v_k, \quad i = 1, 2, \ldots, M$

Aim: *Optimally pick one sensor at each time to minimize cost function.*

Let $u_k^M = \{e_1, \ldots, e_M\}$ where e_i is unit vector with 1 in i-th position.

Equivalent observation

$$y_k(u_k^M) = \sum_{i=1}^{M} u_k^{M\prime} e_i y_k^{(i)} = h(x_k, u_k^M, v_k)$$

Remarks:

1. Easily generalized to picking $L < M$ sensors.
2. Note $v_k(u_k^M) = \sum_{i=1}^{M} u_k^{M\prime} e_i v_k^{(i)}$ is white. Obtained by pasting sample paths if white noise processes $v_k^{(1)}, v_k^{(2)}, \ldots, v_k^{(M)}$).

Example (cont)

Timing: Assume
$I_k = \{u_1^M, \ldots, u_k^M, y_1(u_1^M), \ldots, y_k(u_k^M)\}$
Scheduling sequence: $\mu = \{\mu_1, \ldots, \mu_n\}$
Estimator sequence $\epsilon = \{\epsilon_1, \ldots, \epsilon_N\}$

1. **Scheduling**: Based on I_k, generate $u_{k+1} = \mu_{k+1}(I_k)$.
2. **Observation**: $y_{k+1}(u_{k+1})$
3. **Estimation**: Using $y_{k+1}(u_{k+1})$ generate state estimate $e_{k+1} = \epsilon_{k+1}(I_{k+1})$.

Cost:

$$J_{\mu,\epsilon} = \mathbb{E}\left\{\sum_{k=1}^{N}(x_k - \epsilon_k(I_k))^2 + \sum_{k=0}^{N-1} c_k(x_k, \mu_{k+1}(I_k))\right\}$$

Note: Problem considerably simplified since no u_k^P. Choice of estimator ϵ_k does not affect future x_k of system. Hence estimator optimization can be independently done at each time.

Soln to Sensor Scheduling Problem

Define $u_k = [u_k^P \; u_{k+1}^M]$.
Information state: $\pi_k = p(x_k | Y_k, U_{k-1}, u_0^M)$
Updated as before: $\pi_{k+1}(x) = T(\pi_k(x), y_{k+1}, u_k)$

Fully observed problem in info state

$$J = \psi(u_0^M) + \mathbb{E}\left\{ \sum_{k=1}^{N-1} L_k(\pi_k, u_k) + \Phi(\pi_n, u_N^P) \right\}$$

$$L_k(\pi_k, u_k) = \int_{\mathbb{R}} l_k(x_k, u_k^P, u_{k+1}^M) \pi_k(x_k) dx_k$$

$$\Phi(\pi_n, u_N^P) = \int_{\mathbb{R}} \{ l_N(x_N, u_N^P) + \int_{\mathbb{R}} \phi[f_N(x_N, u_N^P, w_N)] \cdot p(w_N) dw_N \} \pi_N(x_N) dx_N$$

DP yields

$$J_N(\pi_n) = \min_{u_N^P} \Phi(\pi_N, N)$$

$$J_k(\pi_k) = \min_{u_k} \left[L_k(\pi_k, u_k) + \mathbb{E}_{z_{k+1}} \left\{ V_{k+1} \left[T(\pi_k, y_{k+1}, u_k) \right] \right\} \right]$$

7.2 LQG Sensor Scheduling

$$x_{k+1} = A_k x_k + B_k u_k^P + w_k$$
$$y_k = C_k(u_k^M) x_k + v_k$$

Cost

$$J = \mathbb{E}\left\{ \sum_{k=0}^{N} x_k' Q_k x_k + u_k^{P\prime} R_k u_k^P + l_k^M(u_k^M) \right\}$$

Solution: If U_k^M were specified, then $u_k^P = -L_k \hat{x}_{k|k}$ where $L_k = -\left(B_k' K_{k+1} B_k + R_k\right)^{-1} B_k' K_{k+1} A_k$

Recall gain L_k and backward Ricatti K_k are indept of Σ_v and C_k. Thus they are independent of u_k^M.

Thus, u_k^P can can be determined separately from u_k^M.

Choice of u_k^M only affects Kalman covariance and l_k^M.

u_k^M is determined from a nonlinear deterministic control problem (Kalman Ricatti eqn is state). (see [MPD67] for details).

Hence triple separation principle.

Summary

- Note in LQG sensor scheduling, sensor schedule is indpt of observations. Hence past decisions do not affect future decisions on which sensor to pick.
- In all other scheduling problems, e.g. HMM sensor scheduling past decisions affect future ones.
- For risk sensitive LQG scheduling, no triple separation [LK98].

Recent Practical Examples

1. Markov Decision Processes:
   1. Optimal Search theory for sequential paging in cellular networks. [RM97]
   2. Admission control in Broadband networks [Ros95]
   3. Multi-arm bandit problem for optimal scheduling in multiclass priority queueing systems, Klimov's problem, etc [BN96]

2. LQG Control: See [AM89] - resonance suppression in aircraft

3. HMM control (POMDPs): Autonomous robot navigation [SK95], Optimal sensor maneuvering for bearings only tracking.

Concluding Remarks

We covered 3 broad philosophies:

1. Recursive Bayesian State Estimation of stochastic dynamical systems: Optimal Filtering e.g. HMMs, KF and suboptimal approximations such as particle filters.

2. Parameter Estimation: maximum likelihood and adaptive filtering

3. Stochastic Optimization: Stochastic Dynamic programming.

The methodologies of optimal filtering, adaptive filtering and SDP and their analysis form a major part of the areas of statistical signal processing, wireless communication networks (admission control, power control, etc).

The interplay of stochastic processes, sequential decision making in solving discrete optimization problems is a fascinating area. Problems such as multiarmed bandits fall in this category.

POMDP Stopping Time Problems

Action space $\mathcal{U} = \{1 \text{ (stop)}, 2 \text{ (continue) }\}$.
If $u_k = 1$, then problem terminates with cost $c(x, 1)$.
If $u_k = 2$, then accrue cost of $c(x, 2)$ and $x_k \sim P$.
Partially observed state $p(y_k|x_k)$.
Aim: Detect state $x = 1$ with minimum cost.

$$\min_{\mu} J_\mu = \mathbb{E}_\mu\{\sum_{k=0}^{\tau-1} c(x_k, u_k) + c(x_\tau, u_\tau)\}$$
$$\tau = \inf\{k : x_k = 1\}$$

Dynamic Programming:

$$V(\pi) = \min\{c_1'\pi, c_2'\pi + \sum_y V(T(\pi, y, 2))\sigma(\pi, y, 2)\}$$

$$\mu^*(\pi) = \text{argmin}\{c_1'\pi, c_2'\pi + \sum_y V(T(\pi, y, 2))\sigma(\pi, y, 2)\}$$

Aim: Characterize stopping set $\mathcal{S} = \{\pi : \mu^*(\pi) = 1\}$.

Theorem 1: [Lovejoy 1987]: $\mathcal{S}$ is convex subset of Π.
$\mu^*(\pi)$ is MLR increasing in π under foll. conditions:

(A1) $c(i,u) \downarrow i$.
(A2) P and $B_{xy} = P(y|x)$ are TP2.
(A3) $c(i,u)$ is submodular.

Monotone likelihood ratio (MLR) order: $p \geq_r q$ if $\frac{p_i}{q_i} \uparrow i$.

Result: (i) $p \geq_r q \implies p \geq_s q$.
(ii) $\geq_r$ is Closed under conditioning, i.e.,
$X \geq_r Y \implies E\{X|Z\} \geq_r \mathbb{E}\{Y|Z\}$.

Proof: Under (A1), (A2), $Q(\pi,u)$ and $V(\pi) \downarrow \pi$ with respect to MLR order.

$$Q(\pi,2) - Q(\pi,1) = (c_2' - c_1')\pi + \sum_y V(T(\pi,y,2))\sigma(\pi,y,2)$$