Partially Observed Markov Decision Processes - From Filtering to Stochastic Control

Prof. Vikram Krishnamurthy
Dept. Electrical & Computer Engineering
Univ of British Columbia
email vikramk@ece.ubc.ca

These lecture notes contain all the transparancies that will be used during the lectures.
Logistics

- Two 1.5 hr lectures per week.
- 4 assignments. (important to do them)


OUTLINE

1. **Stochastic State Space models and Stochastic Simulation** [4 hours]
   - Stochastic Dynamical Systems
   - Markov Models, Perron Frobenius Theorem, Geometric ergodicity
   - Linear Gaussian Models
   - Jump Markov Linear Systems and Target Tracking
• Stochastic Simulation: Acceptance Rejection, Composition method. Simulation-based optimal predictors

Useful books in stochastic simulation:
1. Sheldon Ross, Simulation

2. **Bayesian State Estimation** [6 hours]

• Review of Regression Analysis and RLS.
• The Stochastic Filtering Problem
• Hidden Markov Model Filter
• Kalman Filter
• Particle Filters and Sequential MCMC
• Reference Probability Method for Filtering
• Filtering with non-standard information patterns: Non-universal filters, Social learning

**Reference Books**

• Ristic, Arulampalam, Gordon, Beyond The Kalman Filter
• Anderson and Moore, Optimal Filtering
• Jazwinski, Stochastic Processes and Filtering
3. Stochastic Control: Full Observed case [2 hours]
   - Dynamic Programming
   - Structural Results

4. Structural Results for POMDPs [6 hours]
   - Stochastic Orders
   - Stochastic Dominance of Filters
   - Lattice Programming
   - Example 1: Quickest Detection with Optimal Sampling
   - Example 2: Optimized Social Learning
   - Example 3: Global Games
   - Multi-armed bandits.
Applications

Sensor Adaptive Signal Processing

**Key idea:** Feedback and Reconfigurability leading to a smart sensor – “active sensing”

**Key Issue:** Dynamic Decision making under uncertainty.
**Example 1:** TDMA Cognitive Radio System where users access spectrum hole. Each user is equipped with a decentralized scheduler and rate adaptor for transmission control.

![Diagram](image)

- QoS – Signal to Interference Ratio (SIR) of receiver

**Aim:** Minimize blocking probability of user subject to constraints on

(i) QoS (Estimated SIR)

(ii) Waiting time in Data Buffer
PART 1: Basic Setup

Partially observed stochastic dynamical system.

Known input $u_k$  Observation $y_k$

Parameter $\theta$  state $x_k$

Stochastic System

$\begin{aligned}
x_{k+1} &= f(x_k, u_k, w_k; \theta) \\
y_k &= h(x_k; \theta) + v_k
\end{aligned}$

Noisy Sensor

State Estimator

$\begin{aligned}
\hat{x}_k &= E\{x_k|y_1, \ldots, y_k\} \\
\hat{\theta}_k
\end{aligned}$

Parameter Estimator

Q1: State Estimation  Q2: Parameter Estimation
Overview:
1. **State** $x_k$ evolves rapidly with time (stochastic process).
2. **Parameter** $\theta$ – constant or varies slowly.
3. Bayesian setting: states and parameters are identical.
4. An optimal **filter** is a recursive algorithm for optimal state estimation. Assumes a known dynamical model. It seeks to compute the MMSE state estimate

\[
\hat{x}_k \triangleq \mathbb{E}\{x_k | y_1, \ldots, y_k\}
\]

Examples: Kalman filter, Hidden Markov Model filter
5. In off-line applications often one seeks to compute **maximum likelihood estimate** (MLE) of parameter $\theta$.

\[
\hat{\theta}_{\text{MLE}} \triangleq \max_{\theta} p(y_1, \ldots, y_T; \theta)
\]

Example: Expectation Maximization algorithm
6. In recursive parameter estimation, $\theta$ is assumed to vary slowly with time with unknown dynamics. Recursive Prediction Error (RPE) and Recursive Maximum Likelihood (RML) methods are widely used for tracking $\theta$. They are essentially **adaptive filtering**
(also known as *Stochastic Approximation* algorithms (e.g. Least Mean Squares (LMS), Recursive Least Squares (RLS)) cross coupled with the optimal filter. Analysis of recursive parameter estimation algorithms is difficult (and not considered here).

**Ex 1: Basic Target Tracking**

Target in 2 dimensional space – e.g. ship or submarine.

\[ x[k + 1] = Ax[k] + Bu[k] + v[k] \]

where \( x[k] \triangleq [r_x[k], \dot{r}_x[k], r_y[k], \dot{r}_y[k]]' \) is target state vector at time \( kT \) (\( T \) is the sampling interval.) In target tracking the model used is more general

1. Exact observations of \( x[k] \) are not available.

Typically noisy observations are obtained at sensor (radar, sonar)

\[ z[k] = Cx[k] + v[k] \]

where \( v[k] \) denotes measurement noise.
If sensor measures position:

\[
C = \begin{bmatrix}
1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0
\end{bmatrix}
\]

More sophisticated sensors measure position and velocity, i.e., \( C = I_{4 \times 4} \)

Bearings-only target tracking: only measure angle \( \tan^{-1}(x_1[k]/x_3[k]) \), i.e. nonlinear measurement equation.

2. Clutter: Often radar records false targets.

**Aim:** Design a real time target-tracking algorithm – to estimate \( x[k] \) given measurements \( z[1], \ldots, z[k] \).
Ex 2: Bearings-only Tracking

Typical target model: \( x_k = [r^x_k, r^y_k, v^x_k, v^y_k]' \)

\[
x_k = A_k x_{k-1} + w_k, \quad w_k \sim N(0, \sigma^2_w)
\]

Measured data: noisy measurement of azimuth

\[
y_k = \arctan \left( \frac{r^x_k}{r^y_k} \right) + v_k
\]

\[
= f(x_k) + v_k, \quad v_k \sim N(0, \sigma^2_v)
\]

**Aim:** Estimate \( x_k \) given observation history \( y_1, \ldots, y_k \) in a recursive manner.

Target estimation is a filtering problem. The optimal state estimator – *filter* for this problem is not finite dimensional.
Ex 3: Maneuvering Targets

\[ x_{k+1} = A x_k + B s_k + v_k \]
\[ y_k = C x_k + w_k \]

\( s_k = (s_k^{(x)}, s_k^{(y)})' \) is maneuver command. \( s_k \) is modelled as a finite state Markov chain. This model is a generalized HMM or Jump Markov Linear System (JMLS).

Note: HMMs and Linear State Space models are special cases of JMLS. State estimation of JMLS is surprisingly difficult – computing the optimal state estimate is exponentially hard. Numerous suboptimal state estimation algorithms e.g., IMM, Particle filters, etc.

A more general version of a JMLS is

\[ x_k = A(s_k)x_{k-1} + B(s_k)u_k + G(s_k)v_k \]
\[ y_k = C(s_k)x_k + D(s_k)w_k + H(s_k)u_k \]

Here \( s_k \) is a finite state Markov chain, \( u_k \) denotes a known (exogeneous) input. \( x_k \) is the continuous valued state.
Part II: Partially Observed Stochastic Control

Ex 4: Optimal Observer Trajectory

Suppose observer can move. The model is

\[ x_k = A_k x_{k-1} + w_k \]
\[ y_k = f(x_k, u_k) + v_k \]

What is its optimal trajectory \( \{u_k\} \)?

Compute optimal trajectory \( \{u_k\} \) to minimize

\[ J = \sum_{k=1}^{N} \mathbb{E} \left\{ [x_k - \hat{x}_k(u_k)]^2 \right\} \]

This is a partially observed stochastic control problem called the sensor scheduling problem.
Intelligent Target Tracking

Suppose target is aware of observer.
Target maneuvers based on trajectory of observer.
The model is

\[ x_k = A_k x_{k-1} + w_k + u^P_k \]
\[ y_k = f(x_k, u^M_k) + v_k \]

What is optimal trajectory of observer \( \{u^M_k\} \) and optimal control for target \( \{u^P_k\} \)?

This is a “full blown” control-scheduling problem. In the third part we address such problems.

Such partially observed stochastic control problems require state estimation as an integral part of the solution.

There are exciting algorithms based on re-inforcement learning which can solve such problems.


Example 1: Smart (Cognitive) Radar

Which target should the radar look at?

Example 2: Optimal Search Problem

Outline
Example 3: Reconfigurable Nano-machines

Mobile gramicidin ion channel biosensor. Gramicidin dimer is broken when antibody bond (red) is broken by explosive molecule (green).

Tools
Bayesian Estimation + Feedback control (learning stochastic control)

Such problems suffer from the curse of dimensionality – exponential computational cost and memory (PSPACE hard).

We focus on structural results.
Supermodularity, lattice programming, Monotone Comparative Statics: see Topkis book [1998].

Under what conditions of $f$ does

$u^*(x) = \text{argmax}_u f(x,u) \uparrow x$?

![Monotone threshold policy](image)

Monotone threshold policy

Exciting area: Signal processing + Control + Economics: Social learning, game theory, etc.