ONLINE LEARNING AND OPTIMIZATION OF MARKOV JUMP LINEAR MODELS

Sevi Baltaoglu⋆ *Lang Tong*⋆ *Qing Zhao*⋆

⋆ School of Electrical and Computer Engineering, Cornell University, Ithaca, NY, USA, 14850
Email: {msb372,lt35,qz16}@cornell.edu

ABSTRACT

The problem of online learning and optimization of unknown Markov jump linear models is considered. A new online learning algorithm, referred to as Markovian simultaneous perturbations stochastic approximation (MSPSA), is proposed. It is shown that MSPSA achieves the minimax regret order of $\Theta(\sqrt{T})$. Using the Van Trees inequality (stochastic Cramér-Rao bound), it is shown that $\Theta(\sqrt{T})$ is the lowest regret order achievable. Simulation results show scenarios that MSPSA offers significant gain over the greedy certainty equivalent approaches.

Index Terms— Online learning, stochastic approximation, stochastic Cramér-Rao bound, continuum multi-armed bandit, sequential decision making

1. INTRODUCTION

Markov jump models are widely used in signal processing, communications, and control. Some of the applications in signal processing are highlighted in [1, 2]. See also [3] for an extensive coverage and relevant applications.

In this paper, we consider the problem of *online learning and optimization* of Markov jump linear models with unknown parameters. By online learning and optimization we mean that the input of the unknown model is chosen sequentially to minimize the expected total cost or maximize the expected accumulative reward.

The model considered here is a linear (affine) model, modulated by an exogenous finite state Markov chain $(\mathcal{S}, P)$ where $\mathcal{S} = \{1, \cdots, K\}$ is the state space and $P = [p_{i,j}]$ the transition probability matrix. We assume that the state space $\mathcal{S}$ is known but the transition matrix P is unknown.

Each state $k \in \mathcal{S}$ of the Markov chain is associated with a linear model whose model parameters are denoted by $\theta_k = (A_k, b_k)$ where $A_k \in \Re^{m\times n}$ has full column rank and $b_k \in \Re^m$. All system parameters $\theta = \{\theta_k\}$ are assumed deterministic and unknown. At time t, the input-output relation of the system is given by

$$y_t = A_{s_t} x_t + b_{s_t} + w_t^{(s_t)}, \tag{1}$$

where $x_t \in \Re^n$ is the input, $y_t \in \Re^m$ the observable output, $s_t \in \mathcal{S}$ the state of the system, and $w_t^{(s_t)}$ the random noise which is independent over t with a possibly state dependent distribution.

A policy μ is defined as a sequence of decision rules, $i.e.$, $\mu = (\mu_0, \mu_1, \cdots)$, such that, at time $t-1$, μ_{t-1} maps input vector $X^{t-1} = (x_0, \cdots, x_{t-1})$, output vector Y^{t-1}, and state vector S^{t-1} to the system input x_t at time t.The problem of *online learning and optimization* is to construct a policy that achieves a given objective.

This work was supported in part by the National Science Foundation under Grants CNS-1135844 and 1549989 and by the Army Research Office under Grant W911NF-12-1-0271.

We measure the performance of a policy μ by the accumulative cost incurred at each stage. Because x_t is determined before s_t is realized, the stage cost $l(s_{t-1}, x_t)$ is a function of x_t and s_{t-1}. The objective of *online learning* is to minimize the expected accumulated cost

$$\min_{\mu} \mathbb{E}\bigg(\sum_{t=1}^{T} l(s_{t-1}, x_t) | s_0 = s \bigg),$$

where T is the learning and optimization horizon. Note that the above quantity is a function of system parameters θ that are deterministic. In characterizing online learning policies in such settings, it is standard to use the minimax formulation that considers the performance of the best policy under the worst system parameters.

In this paper, we focus on the quadratic cost that arises naturally from the tracking or the regulation problems. In particular, the stage cost at time t is given by

$$l(s_{t-1}, x_t) \triangleq \mathbb{E}\bigg(||y^* - y_t||^2 \bigg| s_{t-1}, x_t \bigg), \tag{2}$$

where $y^* \in \Re^m$ is the target value for output.

As an application, the above formulation is particularly relevant in dynamic pricing problems where the pricing signal x_t is used to induce demand y_t. The idea is to set price x_t sequentially to match a certain contracted demand [4, 5, 6, 7]. Here the online learning problem is one of exploration and exploitation, in which the price x_t must be sufficiently diverse to learn the system model and also set to exploit the learned system parameters in some optimal fashion.

1.1. Related work and main contributions

Without Markovian jump as part of the model, *i.e.,* $|\mathcal{S}| = 1$, the problem considered here is the classical problem of control in experiment design studied by Anderson and Taylor [8]. It is also a special case of self-tuning regulation introduced by Aström and Wittenmark [9]. Anderson and Taylor proposed a certainty equivalence rule where the input x_t is determined by using the maximum likelihood estimates of system parameters as if they were the true parameters.

Despite its intuitive appeal, the Anderson-Taylor rule was shown to be suboptimal by Lai and Robbins [10]. In fact, there is a non-zero probability that the Anderson-Taylor rule does not produce consistent estimates of system parameters. For the scalar model, Lai and Robbins proposed a stochastic approximation approach that achieves the optimal regret order [11]. In particular, it was shown that their technique achieves the lowest possible accumulative regret order of $\Theta(\log T)$. The result was further generalized by Lai [12] to the $m \times m$ multivariate linear dynamic systems. Lai showed that the best achievable cumulative regret remains to be $\Theta(\log T)$ when the system matrix $A \in \Re^{m\times m}$ is invertible.

The main contribution of this paper is the generalization of online learning of time invariant linear models to that of Markov jump

linear models. Using a multidimensional version of the Kiefer-Wolfowitz method [13], we propose an online learning policy, referred to as Markovian simultaneous perturbations stochastic approximation (MSPSA). We show that MSPSA achieves the lowest regret order of $\Theta(\sqrt{T})$. The notion of regret is made precise in Sec 2.

A key implication of our results is that, in comparing with the learning problem of a linear time invariant model studied in [12], modulating a linear model by a Markov jump process introduces substantial learning complexity; the regret order increases from $\Theta(\log T)$ to $\Theta(\sqrt{T})$. As a special case, we also show that, even in the absence of Markovian jump, when the system matrix A is not invertible, the best regret order is also $\Theta(\sqrt{T})$. It worths noting that adding just one row to a square and invertible A can change the worst case accumulative regret from $\Theta(\log T)$ to $\Theta(\sqrt{T})$. This can be interpreted as the consequence of the minimum of the cost function is not a root anymore as in the case of single state with square and invertible A and decision maker can't understand how close it is to the minimum by looking at its observations.

The results presented here are obtained using several techniques developed in different contexts. The learning policy proposed in this paper is a generalization of Spall's multivariate stochastic approximation for the Markov jump linear models. To show the optimality of the proposed learning policy, we use the van Trees inequality [14] to lower bound the estimation error, which is a familiar technique in the signal processing community and used to analyze online learning by Keskin and Zeevi in [7].

The literature on online learning and optimization of time varying models, to which this work belongs, is limited. A relevant work by Yin, Ion, and Krishnamurthy considered the problem of stochastic optimization when the system parameters have Markov jump dynamics [15]. Their analysis deals with the infinite horizon problem and does not provide a characterization of regret order.

Besbes, Gur, and Zeevi [16] considered a more general notion time varying objective function where the total temporal change over the time horizon is restricted to a "variation budget". Since the temporal changes are assumed to be deterministic in their formulation, the regret is defined as the difference between the cumulative cost obtained and the cumulative cost of a clairvoyant who knows all the temporal changes exactly and chooses the optimal input. Hence, their characterization of the minimax regret is too pessimistic for the Markov jump model considered here.

2. ONLINE LEARNING AND REGRET

To measure the performance of an online learning algorithm, we use the regret as a proxy for optimization. In particular, the cumulative regret $R_T^{\mu}(\theta, P)$ of a learning algorithm μ, defined in (6), is measured by the difference between the expected accumulated cost of the decision maker and that of a clairvoyant who knows the system parameters and sets the system input optimally.

2.0.1. Clairvoyant policy

We begin by deriving the expected cumulative cost when the system parameters are known. To this end, the clairvoyant's objective is to minimize the expected cumulative cost

$$\min_{\{x_t\}_{t=1}^T} \mathbb{E}\Big(\sum_{t=1}^{T} l(s_{t-1}, x_t) \Big| s_0 \Big). \tag{3}$$

Since the Markov process is independent of the decision policy, the above optimization decouples to choosing the system input x_t separately for each decision stage with stage cost

$$l(s_{t-1}=i, x_t) = \sum_j p_{i,j} \left(\|y^* - A_j x_t - b_j\|_2^2 + \mathrm{Tr}(\Sigma_w^{(j)}) \right), \tag{4}$$

where $\Sigma_w^{(j)}$ is the covariance matrix of $w_t^{(j)}$. The optimal system input, when (θ, P) are known, is then given by

$$x_i^*(\theta, P) = \Big(\sum_j p_{i,j} A_j^{\mathrm{T}} A_j \Big)^{-1} \Big(\sum_j p_{i,j} A_j^{\mathrm{T}} (y^* - b_j) \Big). \tag{5}$$

Thus, the optimal input x_t^* of a clairvoyant at any time t depends only on the system parameter θ, transition matrix P, and the previous state $s_{t-1} = i$. In the sequel, we use x_i^* to represent the optimal set-point of the input when the system state is i, dropping the explicit parameter dependency in the notation.

2.0.2. Regret

We are now in the position to introduce the notion of regret. The instantaneous regret at stage t is the expected difference of the stage cost obtained by policy μ and the stage cost of the optimal input $x_{s_{t-1}}^*$, *i.e.*,

$$\begin{aligned} r_t^{\mu}(\theta, P) &= \mathbb{E}\left(l(s_{t-1}, x_t^{\mu}) - l(s_{t-1}, x_{s_{t-1}}^*) \right) \\ &= \mathbb{E}\left(\|A_{s_t}(x_t^{\mu} - x_{s_{t-1}}^*)\|_2^2 \right), \end{aligned}$$

where we used the first order optimality condition for $x_{s_{t-1}}^*$. The cumulative regret can then be expressed as

$$R_T^{\mu}(\theta, P) = \mathbb{E}\Big(\sum_{t=1}^{T} \|A_{s_t}(x_t^{\mu} - x_{s_{t-1}}^*)\|_2^2 \Big). \tag{6}$$

Since the regret defined above is a function of system parameters, we characterize the performance of μ by the worst case regret

$$\bar{R}_T^{\mu} \triangleq \sup_{\theta, P} R_T^{\mu}(\theta, P).$$

Note that $\bar{R}_T^{\mu}$ grows monotonically with T. We are interested in the learning rule that has the slowest regret growth.

3. AN ORDER OPTIMAL ALGORITHM AND PERFORMANCE ANALYSIS

3.1. MSPSA: An online learning algorithm

In this section, we develop an online learning policy that achieves the slowest growth rate of regret. Referred to as MSPSA, the algorithm is an extension of the simultaneous perturbation stochastic approximation (SPSA) algorithm proposed by Spall [13] to Markov jump linear models. Our algorithm applies to cases where the decision maker knows a convex compact set $\Pi_i \subset \Re^n$, *e.g.,* $\Pi_i = [l_i, u_i]^n$, containing the optimal solution x_i^* for each state $i \in \mathcal{S}$.

SPSA is a Kiefer-Wolfowitz type algorithm that updates the estimates of the optimal solution by a stochastic approximation of the objective gradient. The key step is to generate two observations corresponding to two randomly perturbed inputs and use them to construct gradient estimates. In applying this idea to Markov jump linear models, a complication arises due to the uncertainty associated with

the system state at the time when the system input is determined; consecutive observations may correspond to different system states.

To circumvent this complication, the key idea of MSPSA is to keep track of the estimate $\hat{x}_i$ of the optimal system input x_i^* . When state i is realized, a randomly perturbed $\hat{x}_i$ is used as input for the next stage. And the $\hat{x}_i$ is updated only when we obtain two observations of the system output under state i.

Details of this implementation is given in Algorithm 1. First, MSPSA policy assigns an arbitrary predetermined optimal input estimate $\hat{x}_{i,1} \in \Re^n$ for each state. At the beginning of each period t, MSPSA checks the previous state s_{t-1} (line 3 in Algorithm 1), and checks if any observation is taken using the most recent optimal input estimate related to state s_{t-1}, *i.e.,* $\hat{x}_{s_{t-1},t_{s_{t-1}}}$ (line 4). If the first observation is not taken yet, MSPSA sets the input as a randomly perturbed estimate $\hat{x}_{s_{t-1},t_{s_{t-1}}}$ (line 5) otherwise it sets the input by perturbing the estimate in the opposite direction as the first one (line 9). In line 5, a simple choice for the random perturbation $\Delta_{t_i,j}$ is a Bernoulli(0.5) distribution with values $+1$ and -1, and the gain sequence c_{t_i} should be chosen larger in the high noise setting for an accurate gradient estimate [17]. Then, in line 12, it updates the optimal solution estimate by a stochastic approximation using the stage costs calculated from both observations (line 6 and 10) and by projecting it to a convex compact set $\Pi_{s_{t-1}}$ containing $x^*_{s_{t-1}}$. The choice of the sequence a_{t_i}, that is used in line 12 for update, determines the step size.

Algorithm 1 MSPSA

1: **Initialize:**
For every $i \in \mathcal{S}$, $t_i \leftarrow 1$, $e_i \leftarrow 0$, $\hat{x}_{i,1} \in \Re^n$ be an arbitrary predetermined input, and Π_i be a convex compact set contained in $\Re^n$.
2: **for** t = 1 to T **do**
3: **if** $s_{t-1} = i$ is observed **then**
4: **if** $e_i = 0$ **then**
5: $x_t \leftarrow \hat{x}_{i,t_i} + c_{t_i}\Delta_{t_i}$
where $\Delta_{t_i} = [\Delta_{t_i,1}, ..., \Delta_{t_i,n}]^{\mathrm{T}}$ and $\Delta_{t_i,j}$'s are drawn from an independent and identical distribution that is symmetrical around zero, bounded, and satisfying $\mathbb{E}[\frac{1}{\Delta^2_{t_i,j}}] < \infty$.
6: $d^+_{i,t_i} \leftarrow \|y_t - y^*\|_2^2$
7: $e_i \leftarrow 1$
8: **else**
9: $x_t \leftarrow \hat{x}_{i,t_i} - c_{t_i}\Delta_{t_i}$
10: $d^-_{i,t_i} \leftarrow \|y_t - y^*\|_2^2$
11: $e_i \leftarrow 0$
12: Update:

$$\hat{x}_{i,t_i+1} \leftarrow \mathcal{P}_{\Pi_i}\left(\hat{x}_{i,t_i} - a_{t_i}\Big(\frac{d^+_{i,t_i} - d^-_{i,t_i}}{c_{t_i}}\Big)\bar{\Delta}_{t_i}\right), \quad (7)$$

where $\mathcal{P}_{\Pi_i}(.)$ denotes the euclidean projection operator onto Π_i, and $\bar{\Delta}_{t_i} = \big[\frac{1}{\Delta_{t_i,1}}, ..., \frac{1}{\Delta_{t_i,n}}\big]^{\mathrm{T}}$.
13: $t_i \leftarrow t_i + 1$
14: **end if**
15: **end if**
16: **end for**

In the next section, we present the regret growth rate analysis of MSPSA and the sufficient conditions under which these results hold.

3.2. Regret analysis for MSPSA algorithm

Let $\lambda_{min,i}$ and $\lambda_{max,i}$ be the minimum and maximum eigenvalue of $\sum_j p_{i,j}A_j^{\mathrm{T}}A_j$, and t_i be the number of times the optimal input estimate $\hat{x}_{i,t_i}$ has been updated up to time t by MSPSA algorithm. Here, we show that the MSPSA achieves the regret growth rate of $O(\sqrt{T})$ under the following conditions on the selection of the compact set Π_i, and the algorithm parameters a_{t_i}, and c_{t_i}:

(C1) For every $i \in \mathcal{S}$, $x_i^* \in \Pi_i$.

(C2) For every $i \in \mathcal{S}$, $a_{t_i} = \frac{\gamma_i}{N_i + t_i}$ where constant $\gamma_i \geq \frac{1}{8\lambda_{min,i}}$, and $N_i \geq 0$ an integer.

(C3) For every $i \in \mathcal{S}$, $c_{t_i} = \frac{\gamma_i'}{(N_i' + t_i)^{0.25}}$ where constant $\gamma_i' \in \Re^+$, and $N_i' \leq N_i$ an integer.

Hence, the decision maker, who follows MSPSA, needs to have some information about a set Π_i containing the optimal solution, and a lower bound on the minimum eigenvalue $\lambda_{min,i}$ for every state $i \in \mathcal{S}$ to satisfy **(C1)** and **(C2)**. These assumptions are not restrictive since a decision maker, who is uncertain about the underlying system, can take the compact set containing the optimal solution or γ_i as large as he wants to ensure **(C1)** and **(C2)**. Especially for dynamic pricing applications, these assumptions are reasonable and common, $e.g.$, see [6, 7]. If the optimal input estimate update fluctuates between the borders of the compact set Π_i at the beginning of the algorithm, N_i can be taken greater than zero to prevent this fluctuation.

Let's define optimal input estimate mean squared error (MSE) of the MSPSA policy as $e_{i,t_i} = \mathbb{E}\left(\|\hat{x}_{i,t_i} - x_i^*\|_2^2\right)$. The following lemma provides a bound for e_{i,t_i+1} in terms of e_{i,t_i}.

Lemma 1 *If **(C1)** holds, and $w_t^{(i)}$ has a finite fourth-order order moment for all $i \in \mathcal{S}$, and the MSPSA algorithm parameters $a_{t_i}, c_{t_i} > 0$ are decreasing in t_i, then for any θ, there exists some constants $C_i^1, C_i^2 > 0$ satisfying*

$$e_{i,t_i+1} \leq (1 - a_{t_i}8\lambda_{min,i} + a_{t_i}^2 8\lambda^2_{max_i}C_i^1)e_{i,t_i} + a_{t_i}^2\frac{C_i^2}{c_{t_i}^2}$$

for any $i \in \mathcal{S}$.

Due to space limitations, the detailed proofs are omitted.

Define the accumulative input MSE as

$$E_T^{\mu}(\theta, P) = \mathbb{E}\left(\sum_{t=1}^{T}\|x_t^{\mu} - x^*_{s_{t-1}}\|_2^2\right).$$

Using lemma 1, we provide a bound for the decreasing rate of e_{i,t_i} and hence the growth rate of accumulative input MSE and accumulative regret for appropriate choices of a_{t_i} and c_{t_i}. The following theorem shows that the MSPSA algorithm achieves the optimal growth rate of accumulative regret.

Theorem 1 *If conditions of Lemma 1 hold, and the MSPSA algorithm parameters satisfy **(C2)** and **(C3)**, then for any system parameter set θ and P, there exists a constant $C > 0$ such that*

$$E_T^{MSPSA}(\theta, P) \leq C\sqrt{T}. \quad (8)$$

We thus have

$$R_T^{MSPSA}(\theta, P) \leq \lambda_M C\sqrt{T}, \quad (9)$$

where $\lambda_M = \max_j \lambda_{max}(A_j^T A_j)$.

3.3. A Lower bound on the growth rate of regret

We now show that MSPSA in fact provides the slowest regret growth. To this end, we provide a lower bound of regret growth for all online learning algorithms.

Theorem 2 *For any value of $K > 1$ and $m \geq n$, there exists some constants $C', C > 0$ such that, for any policy μ and for all $T \geq 2$,*

$$\bar{E}_T^{\mu} \geq C'\sqrt{T}, \tag{10}$$

and

$$\bar{R}_T^{\mu} \geq C\sqrt{T} \tag{11}$$

where $\bar{E}_T^{\mu}$ and $\bar{R}_T^{\mu}$ denote, respectively, the worst case cumulative input MSE and the worst case cumulative regret.

To sketch the proof, we consider a hypothetical case in which the decision maker receives additional observations at each period t. It is assumed that the additional observations provided to the decision maker are the observation values corresponding to input x_t^{μ} from the states that didn't occur at t. Since such observations can't increase the growth rate of regret of the optimal policy, we establish a lower bound for this case by showing that it is equivalent to single state case with $m > n$ and using the multivariate Van Trees inequality [18] in a similar way as in [7]. In particular, the following theorem states that there exists some system parameter θ, transition matrix P for which the growth rate of the cumulative regret and input MSE cannot be lower than $\sqrt{T}$ for any policy μ.

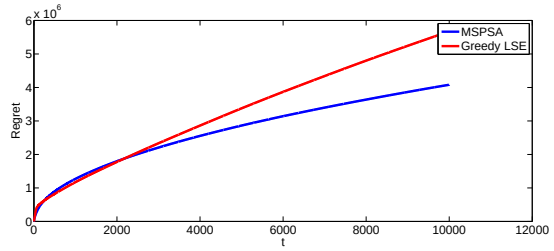
4. SIMULATION

We present a few simulation results on the performance of MSPSA. Note that these simulation results illustrate the performance for "typical" parameters, in contrast to the theoretical characterization of the worst case performance in Theorem 2.

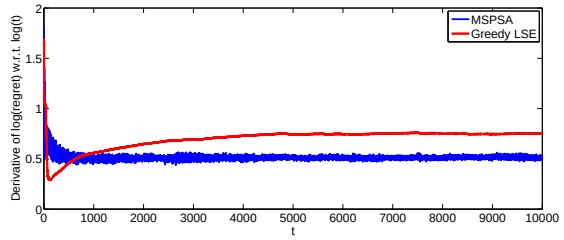
For a benchmark comparison, we consider the greedy LSE method in [8]. At each t, greedy LSE determines the input by using the least square estimates of system parameters as if they were the true parameters. If the estimates are computationally intractable then it takes the most recent estimate of the input and adds a small random dithering to improve the learning rate as showed in [4]. As a last step, it projects the calculated input to the predetermined input range which is assumed to contain the optimal input value. Although, in general, greedy LSE performed well numerically [8], it was shown that it can lead to incomplete learning [10].

Fig. 1 shows the average performance of MSPSA and greedy LSE under a high noise setting, *i.e.*, $\sigma = 20$. Parameters a_{t_i} and c_{t_i} were set to be $\frac{0.3/\lambda_{min,i}}{t_i}$ and $\frac{7}{t_i^{0.25}}$, respectively.

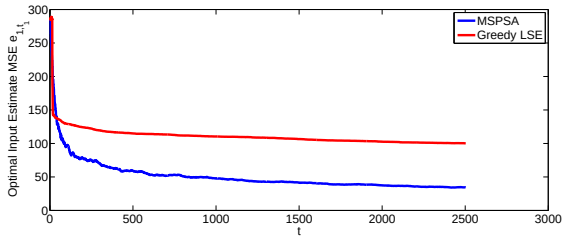
In many cases, greedy LSE performed quite well. However, in a high noise setting, we observed that greedy LSE's performance decreased significantly and its regret grew almost linearly whereas MSPSA preserved its performance which can be seen in Fig. 1(a). In Fig. 1(b), we plot the derivative of the logarithm of the average regret with respect to $log(t)$, and it can be seen that the value for MSPSA is around 0.5 which is consistent with the theoretical result of $O(\sqrt{T})$. On the other hand, the value for greedy LSE is close to 0.8 at T which is a significantly worse growth rate, and it has an increasing trend as T grows. Fig. 1(c) shows the MSE between the optimal input estimate under two policies and the optimal value. We see that greedy LSE performed poorly due to insufficient learning whereas MSPSA converged fast.



(a) Regret



(b) Derivative of log(regret) with respect to log(t)



(c) Optimal Input Estimate MSE

Fig. 1: Average performance comparison of MSPSA and the Greedy LSE. 500 Monte Carlo runs were used to calculate the average performance for T=10000 periods. The system with $K = 4$ states, and with dimensions $m = 5$, and $n = 5$ is used. The transition probability to any other state was set to be 0.25. Observation noise was taken as i.i.d. normal with covariance matrix $\sigma^2 I_m$ and y^* was taken as vector of all 10.

5. CONCLUSION

We present in this paper an online learning and optimization algorithm MSPSA for jump Markov linear model with unknown parameters. We establish that MSPSA achieves the lowest order of regret growth $\Theta(\sqrt{T})$. Our results highlight a change of the minimum order of regret growth from $\Theta(\log T)$ of the classical (non-Markovian) linear models to $\Theta(\sqrt{T})$ of the jump Markov linear models. Our simulation results verify that proposed method MSPSA can outperform the greedy LSE method.

6. REFERENCES

[1] A. Logothetis and V. Krishnamurthy, "Expectation maximization algorithms for map estimation of jump markov linear systems," *IEEE Transactions on Signal Processing*, vol. 47, no. 8, pp. 2139–2156, Aug 1999.

[2] Arnaud Doucet, N.J. Gordon, and V. Krishnamurthy, "Particle filters for state estimation of jump markov linear systems," *IEEE Transactions on Signal Processing*, vol. 49, no. 3, pp. 613–624, Mar 2001.

[3] O.L.V. Costa, M.D. Fragoso, and R.P. Marques, *Discrete-Time Markov Jump Linear Systems*, Probability and Its Applications. Springer, 2005.

[4] M. Lobo and S. Boyd, "Pricing and learning with uncertain demand," in *INFORMS Revenue Management Conference*, Columbia University, 2003.

[5] Dimitris Bertsimas and Georgia Perakis, "Dynamic pricing: A learning approach," in *Mathematical and Computational Models for Congestion Charging*, vol. 101, pp. 45–79. Springer US, 2006.

[6] Liyan Jia, Qing Zhao, and Lang Tong, "Retail pricing for stochastic demand with unknown parameters: An online machine learning approach," in *2013 51st Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, Oct 2013, pp. 1353–1358.

[7] N. Bora Keskin and Assaf Zeevi, "Dynamic pricing with an unknown demand model: Asymptotically optimal semi-myopic policies," *Operations Research*, vol. 62, no. 5, pp. 1142–1167, 2014.

[8] T.W. Anderson and J. Taylor, "Some experimental results on the statistical properties of least squares estimates in control problems," *Econometrica*, vol. 44, pp. 1289C1302, 1976.

[9] K. J. Aström and B. Wittenmark, "On self tuning regulators," *Automatica*, vol. 9, no. 2, pp. 185–199, Mar. 1973.

[10] T.L. Lai and Herbert Robbins, "Iterated least squares in multiperiod control," *Advances in Applied Mathematics*, vol. 3, no. 1, pp. 50 – 73, 1982.

[11] T. L. Lai and H. Robbins, "Adaptive design and stochastic approximation," *The Annals of Statistics*, vol. 7, no. 6, pp. 1196–1221, 1979.

[12] T.L Lai, "Asymptotically efficient adaptive control in stochastic regression models," *Advances in Applied Mathematics*, vol. 7, no. 1, pp. 23 – 45, 1986.

[13] J.C. Spall, "Multivariate stochastic approximation using a simultaneous perturbation gradient approximation," *IEEE Transactions on Automatic Control*, vol. 37, no. 3, pp. 332–341, Mar 1992.

[14] H. L. Van Trees, *Detection, Estimation, and Modulation Theory, Part I*, New York: Wiley, 1968.

[15] G. Yin, C. Ion, and V. Krishnamurthy, "How does a stochastic optimization/approximation algorithm adapt to a randomly evolving optimum/root with jump markov sample paths," *Mathematical Programming*, vol. 120, no. 1, pp. 67–99, 2009.

[16] O. Besbes, Y. Gur, and A. Zeevi, "Non-stationary Stochastic Optimization," *ArXiv e-prints*, July 2013.

[17] J.C. Spall, "Implementation of the simultaneous perturbation algorithm for stochastic optimization," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 34, no. 3, pp. 817–823, Jul 1998.

[18] Richard D. Gill and Boris Y. Levit, "Applications of the van trees inequality: A bayesian cramr-rao bound," *Bernoulli*, vol. 1, no. 1/2, pp. pp. 59–79, 1995.