2003 American Control Conference
http://www.cds.caltech.edu/~murray/papers/2002i_gmh03-acc.html

On the Control of Jump Linear Markov Systems with Markov State Estimation

Vijay Gupta, Richard M. Murray and Babak Hassibi
Division of Engineering and Applied Science
California Institute of Technology
{gupta,murray,hassibi}@caltech.edu

Abstract

We analyze a jump linear Markov system being stabilized using a linear controller. We consider the case when the Markov state is associated with the probability distribution of a measured variable. We assume that the Markov state is not known, but rather is being estimated based on the observations of the variable. We present conditions for the stability of such a system and also solve the optimal LQR control problem for the case when the state estimate update uses only the last observation value. In particular we consider a suboptimal version of the causal Viterbi estimation algorithm and show that a separation property does not hold between the optimal control and the Markov state estimate. Some simple examples are also presented.

1 Introduction and Motivation

Jump linear Markov systems can be used to model a wide variety of dynamic systems. Fault-prone dynamic systems may experience abrupt changes in their structure and parameters, caused by phenomena such as component failures or repairs, changing subsystem interconnections, and abrupt environmental disturbances. Such systems can be modeled as operating in different forms [3], where each form corresponds to some combinations of these events. Another area where jump Markov systems are proving useful is in the tracking community, where the standard problem is that of state and mode estimation. Ideas like “controlling” the transmitted waveforms from radars to minimize errors in tracking [9] rely on jump Markov systems to tackle settings in which maneuvers are allowed. Recently, these systems have also gained attention in their ability to model effects of communication networks and/or channels present between remote sensors, actuators and processors. Random time-delays and error exponents introduced by imperfect communication links can be modeled as a Markov chain. The questions of stability and optimal control of the plants utilizing such links, thus arise naturally.

As an example of how jump linear Markov systems might be useful to model systems being controlled over a network or a communication channel, consider the system shown in Figure 1. The figure represents a system in which the sensor and the controller communicate over a medium which introduces random delays. The medium can be a bus shared with other devices, or a network where routing protocols introduce random delays, or a wireless channel in which protocols like Bluetooth introduce random latency delays before succesful transmission. If the delays being introduced can be modeled by a Markov chain [7], analysis techniques for jump linear Markov systems apply immediately.

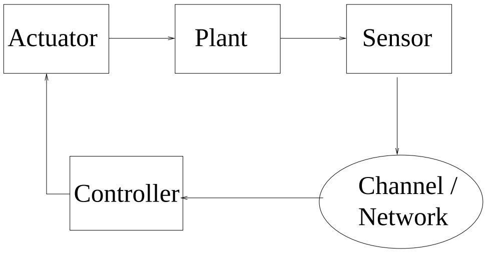

Figure 1: General system in which the sensor and the controller utilize an imperfect communication channel or a network to communicate

As another example, consider the same system with the communication channel modeled as a wireless channel in which an Automatice Repeat Request (ARQ) protocol like that of Bluetooth is not used. Then we can model the channel as a Markov model in which the various states correspond to different SNR values [10]. Corresponding to each SNR value, we will have a different probability of error for any given modulation scheme. The system can again be analyzed using the results of jump linear Markov system analysis.

Because of their importance, jump linear Markov systems have been studied and analyzed extensively. Discrete-time versions of the jump-linear quadratic (JLQ) optimal control problem were solved for finite-time horizons in Blair and Sworder [2]. Ji and Chizeck [5] studied the problem in detail for the case where system parameters were determined by a Markov chain. Concepts like stability and controllability were defined for a Markovian jump linear system and the JLQ problem was solved. Luck and Ray [6] considered a system in which delays were governed by a Markov chain and presented a simple scheme to reduce the system to a time-invariant case by introducing buffers of the same length as the worst case delays. Nilsson and Bernhardsson [8] generalized the results of Ji and Chizeck to the case where the Markov chain determines the probability density function of the variables rather than the values of the variables themselves.

However, all the above approaches assumed the Markov state to be known. This assumption can severely limit the utility of the whole Markov model. In a more practical scenario, a Markov state estimation algorithm would typically be used. In this paper, we extend the above results to this case. We have been able to prove results only for the cases where the state estimate update depends only on the latest observation value. In particular, we consider a suboptimal version of causal Viterbi algorithm and show that a separation property need not hold between the control law and the state estimation algorithm. Work is being done to extend the results to more general state estimation algorithms.

In Section 2, we formulate the problem and state the assumptions. Stability results are presented in Section 3. The absence of the separation property is shown and a few examples are presented. Section 4 deals with the design of the optimal LQR controller. Section 5 presents the conclusions and outlines the future work that needs to be done.

2 Problem Formulation

For the sake of notational convenience, we shall use the notations $x[k]$ and x_k interchangeably. Consider the system described by

$$\begin{aligned} z[k+1] &= \phi_{r_k} z[k] + \gamma_{r_k} u[k] + \Gamma(r_k,\hat{r}_k) v[k] \\ y[k] &= C_{r_k} z[k] \\ u[k] &= F_{\hat{r}_k} y[k], \end{aligned}$$

where $z[k] \in \mathbf{R}^n$ is assumed to be the process state, $u[k] \in \mathbf{R}^m$ is the controlled input which is a linear function of the state observation variable $y[k]$, $v[k] \in \mathbf{R}^n$ is white noise with zero mean and covariance R_v. Let Q denote the transition probability matrix of the Markov chain whose states are represented by $r_i \in \{1,2,...,s\}$. The states of the Markov chain define the probability distribution of the various variables in the system equation. For simplicity of presentation, in this paper we assume that there is only one variable being affected by the Markov chain. We denote the observed value of this variable at time step k by o_k and its probability density function given that the Markov state at time k is j by $f_{o_k|r_k=j}(t|r_k=j)$. Also we denote the expectation of a random variable X taken over another random variable Y by $E_Y[X]$. Such a system might arise, e.g., from discretization of an underlying continuous-time system, which can possibly have time-delays less than the sampling time period present in it. If the controller has the knowledge of the Markov state, it can vary the control law matrix F according to the Markov state r_k. However, if the Markov state is not known, it has to depend on the estimate of the Markov state $\hat{r}_k$. Thus F becomes a function of $\hat{r}_k$.

Thus we see that the system can be written as

$$z[k+1] = \Phi(r_k,\hat{r}_k) z[k] + \Gamma(r_k,\hat{r}_k) v[k] \qquad (1)$$

where

$$\Phi(r_k,\hat{r}_k) = \phi_{r_k} + \gamma_{r_k} F_{\hat{r}_k} C_{r_k}.$$

One central assumption we make is that the variable value being affected by the Markov chain is measurable and the value taken by the variable in every time step is available accurately to the controller. Thus, if the variable is time delay, we use measures such as timestamping of the packets; if it is the system matrix γ — which can be used to model, say, the effect of varying SNR in a communication channel over which $z[k]$ is passed to the controller — then pilot tone measurements are used, and so on.

We are interested in deriving the conditions under which the system in equation (1) is stable. Results exist in the literature about the stability conditions for the case of known Markov state. We extend these results to the case when an estimation algorithm is being used to estimate the Markov state. In this paper, we treat only the case where the state estimation update is based only on the latest measurement and not on the previous measurements as well. In particular, a causal version of the Viterbi algorithm is used as an example. Results also exist for the design of an optimal control law when the Markov state is known at the controller. We present a design incorporating the fact that the state is being estimated.

2.1 Hidden Markov Models and the Viterbi Algorithm

Consider a Markov chain with a finite number of states and given transition probability matrix $Q = \{q_{ij}\}$. Suppose that when a transition occurs, we cannot observe the states corresponding to the transition directly. Rather we obtain an observation related to the transition. We are given the probability $p(o,i)$ of the observed value of the variable being affected by the Markov state being o when the Markov state is i. Such Markov chains are called hidden Markov models and the problem of estimating the state from the observation sequence is called the state estimation problem.

Consider the observation sequence $O_N = \{o_1, o_2, \ldots, o_N\}$. We wish to estimate the state transition sequence $\hat{R}_N = \{\hat{r}_1, \hat{r}_2, \ldots, \hat{r}_N\}$ that maximizes the conditional probability $p(R_N|O_N)$ over all $R_N = \{r_1, r_2, \ldots, r_N\}$. It is well-established, see e.g. [1], that the optimal estimate is given by the Viterbi algorithm, which is a solution to the problem of minimizing $-\ln(\pi_{r_0}) - \sum_{k=1}^{N} \ln(p(o_k, r_k|r_{k-1}))$ over all possible sequences $\{r_1, r_2, \ldots, r_N\}$. The probability distribution of the original states is denoted by π_{r_0}.

However, the Viterbi algorithm is non-causal in that it requires all the observations before estimating the state sequence. In practice, a causal version of the algorithm is used which is based on forward dynamic programming. If we know the smallest costs $D_k(r_k)$ from the beginning to all the states r_k on the basis of the observation sequence O_k, we compute the smallest costs $D_{k+1}(r_{k+1})$ by using the recursion

$$D_{k+1}(r_{k+1}) = \min[D_k(r_k) - \ln(p(o_{k+1}, r_{k+1}|r_k))], \qquad (2)$$

where the minimization is taken over all r_k such that $q_{r_k r_{k+1}}$ is non-zero. An advantage of this procedure is that it can be executed in real time, as soon as each new observation is obtained. Another simplification can be made if the state estimate update is made only on the basis of the current measurement. This amounts to ignoring the first term on the right hand side of equation (2) and minimizing the second term over all transitions from the current state estimate $\hat{r}_k$. This algorithm is refered to as the one-step Viterbi algorithm.

3 Stability Results for Unknown Markov State

3.1 Definition of Stability

Following [8], we consider the stability of the covariance of the system in equation (1). We define the conditional covariance as

$$\begin{aligned} P_{jn}[k+1] =& E_{Y_k}[z_{k+1} z_{k+1}^T | r_{k+1} = j, \hat{r}_{k+1} = n] \qquad (3) \\ =& E_{Y_k}[\Phi(r_k, \hat{r}_k) z_k z_k^T \Phi^T(r_k, \hat{r}_k) | r_{k+1} = j, \hat{r}_{k+1} = n]. \end{aligned}$$

Y_k denotes all the uncertainity in the system at time step k, which is due to the initial state, noise and the Markov states at previous times. At time step t, the Markov state is denoted by r_k while the estimate of the Markov state by $\hat{r}_k$. We also define

$$\tilde{P}_{jn}[k+1] = P_{jn}[k+1] \times \text{Prob}(r_{k+1} = j, \hat{r}_{k+1} = n). \qquad (4)$$

The state covariance $P[k]$ is then given by

$$P[k] = \sum_{i=1}^{s} \sum_{j=1}^{s} \tilde{P}_{ij}(k). \qquad (5)$$

Let $Q = \{q_{ij}\}$ denote the transition probability matrix of the Markov chain. Also, let R_{ij} denote the matrix $\{r_{mn|ij}\}$ where $r_{mn|ij}$ refers to the probability that $(\hat{r}_{k+1} = n | \hat{r}_k = m, r_k = i, r_{k+1} = j)$. Finally let $A_{i,m} = E[\Phi(r_k, \hat{r}_k) \otimes \Phi(r_k, \hat{r}_k) | r_k = i, \hat{r}_k = m]$ where $\otimes$ denotes the Kronecker product. Define the matrix diag$(A_{i,m})$ as a block diagonal matrix with the blocks $A_{1,1}, A_{1,2}, \cdots, A_{1,s}, A_{2,1}, A_{2,2}, \cdots, A_{2,s}, \cdots, A_{s,1}, \cdots, A_{s,s}$ along the diagonal and zeros elsewhere. Denote by Σ the matrix having the following structure

$$\Sigma = \begin{bmatrix} q_{11}R_{11} & q_{21}R_{21} & \cdots & q_{s1}R_{s1} \\ q_{12}R_{12} & q_{22}R_{22} & \cdots & q_{s2}R_{s2} \\ \vdots & \vdots & \vdots & \vdots \\ q_{1s}R_{1s} & q_{2s}R_{2s} & \cdots & q_{ss}R_{ss} \end{bmatrix}.$$

We have the following theorem.

Theorem 1 *Consider the system given in equation(1) and the one-step Viterbi Markov state estimation algorithm. The system is stable, in the sense that its covariance is bounded, iff the matrix $(\Sigma \otimes I) diag(A_{im})$ has all its eigenvalues in the unit circle.*

Proof: We assume that the additive noise term in system of equation (1) is bounded in the mean square sense. Thus it would have no effect on stability and we only need to consider an equation of the form

$$z[k+1] = \Phi(r_k, \hat{r}_k) z[k]. \qquad (6)$$

We have by the definition of the conditional state variance

$$\begin{aligned} \tilde{P}_{jn}[k+1] =& E_{Y_k}[z_{k+1} z_{k+1}^T | r_{k+1} = j, \hat{r}_{k+1} = n] \\ & \times \text{Prob}(r_{k+1} = j, \hat{r}_{k+1} = n) \\ =& E_{Y_k}[\Phi(r_k, \hat{r}_k) z_k z_k^T \Phi^T(r_k, \hat{r}_k) | r_{k+1} = j, \hat{r}_{k+1} = n] \\ & \times \text{Prob}(r_{k+1} = j, \hat{r}_{k+1} = n). \end{aligned}$$

Because of the definition of Markov chain and the assumption that the state estimate update would involve only the observations obtained in time step $(k+1)$, we have that

$$\begin{aligned} \text{Prob}(r_{k+1} = j, \hat{r}_{k+1} = n | f(z_k, r_k, \hat{r}_k), r_k = i, \hat{r}_k = m) = \\ P(r_{k+1} = j, \hat{r}_{k+1} = n | r_k = i, \hat{r}_k = m), \end{aligned}$$

where $f(.)$ is any deterministic function of its arguments. Thus any function involving z_k, $\hat{r}_k$ and r_k only is independent of the event $(r_{k+1} = j, \hat{r}_{k+1} = n)$. Now we use the fact that if C and A are independent given B,

$$E[C|A] = \sum_B \frac{P(A|B)P(B)E[C|B]}{P(A)}. \qquad (7)$$

Thus, in particular, if we define

$$\begin{aligned} C &= z_{k+1} z_{k+1}^T \\ A &= (r_{k+1} = j, \hat{r}_{k+1} = n) \\ B &= (r_k = i, \hat{r}_k = m), \end{aligned}$$

we obtain

$$\begin{aligned} \tilde{P}_{jn}[k+1] = & \\ & \sum_{i=1}^{s} \sum_{m=1}^{s} p_{i \to j, m \to n} \times \text{Prob}(r_k = i, \hat{r}_k = m) \\ & \times E_{Y_k}[\phi(r_k, \hat{r}_k) z_k z_k^T \phi^T(r_k, \hat{r}_k) | r_k = i, \hat{r}_k = m]. \end{aligned} \qquad (8)$$

In the above, $p_{i \to j, m \to n}$ represents the probability that the true Markov state goes from i in step k to j in step $(k+1)$ and the estimated state goes from m to n at the same time. Now we observe that

$$\begin{aligned} E_{Y_k}[f(r_k, \hat{r}_k) g(z_k) | r_k = i, \hat{r}_k = m] = & \\ E_{Y_k}[f(r_k, \hat{r}_k) | r_k = i, \hat{r}_k = m] \times & \\ E_{Y_k}[g(z_k) | r_k = i, \hat{r}_k = m], & \end{aligned}$$

for any functions $f(.)$ and $g(.)$. This can be proved by considering the fact that the variable distribution depends only on the Markov state and from the equivalent condition

$$P(r_k,\hat{r}_k|z_k,r_k=i,\hat{r}_k=m)=P(r_k,\hat{r}_k|r_k=i,\hat{r}_k=m).$$

Thus we can vectorize equation(8) to obtain

$$\text{vec}(\tilde{P}_j[k+1])=\sum_{i,m}p_{i\to j,m\to n}A_{i,m}\text{vec}(\tilde{P}_i[k]),$$

where $A_{i.m}$ has been defined above. This in turn yields

$$\tilde{P}[k+1]=(\Sigma\otimes I)\text{diag}(A_{i,m})\tilde{P}[k], \tag{9}$$

with Σ and $\text{diag}(A_{i,m})$ already defined. It is apparent from the above equation that the stability of the system is given by the stability of the matrix $(\Sigma\otimes I)\text{diag}(A_{i,m})$. ■

To compute $r_{mn|ij}$, we condition it on the value of the underlying variable varying according to the Markov chain:

$$\begin{aligned} r_{mn|ij} = \int & \text{P}(\hat{r}_{k+1}=n|r_{k+1}=j,\hat{r}_k=m,o_{k+1}=t) \\ & \times f_{o_{k+1}|r_{k+1}=j}(t|r_{k+1}=j)dt. \end{aligned} \tag{10}$$

Both the terms in the above expression are computable. The second term is known since we know the distributions of the variable in each Markov state. The first term is computable for any particular estimation algorithm satisfying the assumption stated in the theorem. For the one-step Viterbi algorithm, it computes to the probability that the cost function for state n is least among all possible states. The cost function is given by

$$\begin{aligned} D_{k+1}(\hat{r}_{k+1}=n|\hat{r}_k=m,o_{k+1}=t) \\ = -\ln(q_{mn})-\ln(f_{o_{k+1|r_{k+1}=n}}(t|r_{k+1}=n)). \end{aligned}$$

3.2 Examples

We consider a few examples in this subsection to clarify the result.

1. Consider the case when the estimation algorithm always gives correct results, i.e., the measurement of the variable tells us the state of the Markov chain. Then, the matrix Σ has the form

$$\left[\ \sigma^1 \mid \sigma^2 \mid \cdots \mid \sigma^s\ \right],$$

   where the block σ^i has dimensions $s^2\times s$ and has the structure

$$\sigma^i=\left[\begin{array}{ccc} q_{i1} & q_{i1} & \cdots \\ 0 & 0 & \cdots \\ \vdots & s \text{ rows of zeroes} & \\ \hline q_{i2} & q_{i2} & \cdots \\ 0 & 0 & \cdots \\ \vdots & s \text{ rows of zeroes} & \\ \hline \vdots & & \\ \hline q_{is} & q_{is} & \cdots \end{array}\right].$$

   Thus, the final matrix $(\Sigma\otimes I)\text{diag}(A_{i,m})$ has the same structure, with each element in the above matrix replaced by a $n^2\times n^2$ block. We note that there are a total of s^2 row blocks in the matrix, each consisting of n^2 rows. However, only the blocks 1, $s+2$, $2s+3$, $\ldots$, s^2 are non-zero. Now a $m\times m$ matrix with ith row zero has the same eigenvalues as the $(m-1)\times(m-1)$ matrix formed by removing the ith row and ith coloumn from the original matrix, except for an additional zero eigenvalue. Also the matrix $A_{i,m}$ is the same as A_i defined in [8], when i and m are the same. Thus the eigenvalues of the matrix $(\Sigma\otimes I)\text{diag}(A_{i,m})$ are the same as the eigenvalues of the matrix $(Q^T\otimes I)\text{diag}(A_i)$ except for some additional zero eigenvalues. Thus our results reduce to the the results of [8] in this case, as they should.

2. It is not necessary that if a process is stable when a controller based on known Markov state is used, it will be stable when the same controller is instead fed the states estimated by the one-step Viterbi algorithm. Consider the discrete time version of the system

$$\begin{aligned} \ddot{x}(t) &= u(t) \\ u(t) &= Fx(t-\tau). \end{aligned}$$

   Let the time step be h=0.1. Let the Markov states be characterized by different time delays τ in passing of the sensor signal to the controller, the state 1 having a time delay uniformly distributed between 0 and 0.7h, while the state 2 having a time delay uniformly distributed between 0.69h and 0.71h. Thus the equivalent discrete-time system is characterized by the equations

$$\begin{aligned} x[k+1] &= \phi x[k]+\gamma_0 u[k]+\gamma_1 u[k-1] \\ u[k] &= Fx[k]. \end{aligned}$$

   where

$$\phi=e^{Ah} \qquad A=\begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \qquad B=\begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

$$\gamma_0=\int_0^{h-\tau} e^{As}dsB \qquad \gamma_1=\int_{h-\tau}^{h} e^{As}dsB.$$

   Let the control laws in the two Markov states be given by $F=[-0.2-0.1]$ in the state 1 and by $F=[0.5-0.2]$ in state 2. Let the transition probability matrix be

$$Q=\begin{bmatrix} 0.6 & 0.4 \\ 0.8 & 0.2 \end{bmatrix}.$$

   Then, the matrix Σ is given by

$$\Sigma=\begin{bmatrix} 0.5914 & 0.5914 & 0.7886 & 0.7886 \\ 0.0086 & 0.0086 & 0.0114 & 0.0114 \\ 0 & 0 & 0 & 0 \\ 0.4 & 0.4 & 0.2 & 0.2 \end{bmatrix}.$$

When we obtain the eigenvalues of the 36×36 matrix $(\Sigma\otimes I)\mathrm{diag}(A_{i,m})$, we obtain an eigenvalue outside the unit circle. On the other hand, the eigenvalues of the 18×18 matrix $(Q^T\otimes I)\mathrm{diag}(A_i)$ are all inside the unit circle, with the highest absolute value being 0.9971. If we simulate the systems, we indeed find that the system is stable if Markov state is known. However it goes unstable if the same controller is used but the one-step Viterbi algorithm is used to estimate the state. Thus, a separation property does not hold between the Markov state estimate and stability of the system.

3.3 Comment

We have given necessary and sufficient conditions for stability of a jump linear Markov state when Markov state is being estimated. Stability considered is the asymptotic stability of the conditional covariances. However this might be too strong a condition. “Almost Sure stability” might provide a better estimate of stability; however the transient performance of the process might be unacceptable in this case. The relation between the various forms of stability is discussed in [4]. Also note that the result can easily be extended to the case of two or more independent Markov chains modeling many separate communication links.

4 Optimal Controller

We consider the cost function

$$J_N = z_N^T\Pi_N z_N + E\left(\sum_{k=0}^{N-1}\begin{bmatrix} z_k \\ u_k\end{bmatrix}^T \Pi \begin{bmatrix} z_k \\ u_k\end{bmatrix}\right). \qquad (11)$$

where Π is a symmetric, positive semidefinite matrix of the form

$$\Pi = \begin{bmatrix} \Pi_{11} & \Pi_{12} \\ \Pi_{12}^T & \Pi_{22}\end{bmatrix}.$$

Further, Π_{22} is positive definite.

4.1 Optimal State Feedback

We assume that the full state information about $z[k]$ is available to the controller. Then similar to [8], we have the following theorem.

Theorem 2 *Consider the problem of minimizing the cost function given by equation (11) for the system given by equation (1) with full state information about $z[k]$ available to the controller and when the one-step Viterbi algorithm is used. Assume that the Markov chain reaches a stationary state. Provided we have noise-free full state vector information, the control law that minimizes the cost function (11) is given by*

$$u_k^* = -L_k(o_k,\hat{r}_k)\begin{bmatrix} z_k \\ u_{k-1}^*\end{bmatrix}$$

where for $\hat{r}_k = i\ \ (i = 1,...,s)$, we have

$$\begin{aligned}
L_k(o_k,i) =& (\Pi_{22}+\tilde{S}_i^{22}(k+1))^{-1}\times \\
& [\Pi_{12}^T+\tilde{S}_i^{21}(k+1)\tilde{S}_i^{23}(k+1)] \\
\tilde{S}_i(k+1) =& G^T\sum_{j=1}^{s} r_{ij}S_j(k+1)G \\
G =& \begin{bmatrix} \Phi(o_k,r_k,\hat{r}_k) & \Gamma(o_k,r_k,\hat{r}_k) \\ 0 & I\end{bmatrix} \\
S_i(k) =& E_{o_k}\begin{bmatrix} F_2^T(\hat{r}_k)\tilde{S}_i(k+1)F_2(\hat{r}_k) \\ +F_1^T(\hat{r}_k)\Pi F_1(\hat{r}_k)|\ \hat{r}_k = i\end{bmatrix} \\
F_1(\hat{r}_k) =& \begin{bmatrix} I & 0 \\ -L_k(o_k,\hat{r}_k) & \end{bmatrix} \\
F_2(\hat{r}_k) =& \begin{bmatrix} I & 0 \\ -L_k(o_k,\hat{r}_k) & \end{bmatrix} \\
S_i(N) =& \begin{bmatrix} \Pi_N & 0 \\ 0 & 0\end{bmatrix}.
\end{aligned}$$

The elements r_{ij} refer to the probability of the state estimate changing from i in time step k to j in the time step $k+1$. Note that the elements r_{ij} are in general a function of o_k. $\tilde{S}_i^{ab}(k)$ is the block (a,b) of the symmetric matrix $\tilde{S}_i(k)$ and Π_{ab} is block (a,b) of Π.

Proof: The proof is similar to that of the problem treated in [8] and is omitted for the sake of brevity. ■

We now give a method to calculate the probability term r_{ij} for the case when one-step causal Viterbi algorithm is being used. We need to compute

$$P(\hat{r}_{k+1} = j|\hat{r}_k = i, o_k = t). \qquad (12)$$

Note that the information vector available at the time of making the decision includes the estimated states until that time step as well as all the time delays. Let us condition on the probability of the actual Markov state at time k being l. Thus we obtain

$$\begin{aligned}
&P(\hat{r}_{k+1} = j|\hat{r}_k = i, o_k = t) = \\
&\quad\sum_{l=1}^{s} P(\hat{r}_{k+1} = j|\hat{r}_k = i, o_k = t, r_k = l) \\
&\quad\times P(r_k = l|\hat{r}_k = i, o_k = t).
\end{aligned}$$

To calculate the first term on the right hand side, let us condition it on the probability of the next Markov state being m.

$$\begin{aligned}
&P(\hat{r}_{k+1} = j|\hat{r}_k = i, o_k = t, r_k = l) = \\
&\quad\sum_{m=1}^{s} P(\hat{r}_{k+1} = j|\hat{r}_k = i, o_k = t, r_k = l, r_{k+1} = m) \\
&\quad\times P(r_{k+1} = m|\hat{r}_k = i, o_k = t, r_k = l).
\end{aligned}$$

Now we can evaluate all the terms. The term $P(\hat{r}_{k+1} = j|\hat{r}_k = i, o_k = t, r_k = l, r_{k+1} = m)$ is simply $P(\hat{r}_{k+1} = j|\hat{r}_k =$

$i, r_k = l, r_{k+1} = m)$ which was evaluated in the stability proof. The term $P(r_{k+1} = m|\hat{r}_k = i, o_k = t, r_k = l)$ is simply $P(r_{k+1} = m|r_k = l)$ by the Markov property. To evaluate the term $P(r_k = l|\hat{r}_k = i, o_k = t)$, note that this is the same as $P(r_k = l|o_k = t)$. To prove this, consider the equivalent condition

$$f(o_k = t|r_k = l, \hat{r}_k = i) = f(o_k = t|r_k = l).$$

In the above equation, f refers to the probability distribution function of o_k. Now $P(r_k = l|o_k = t)$ can be evaluated by Baye's rule

$$P(r_k = l|o_k = t) = \frac{f(o_k = t|r_k = l)P(r_k = l)}{\sum_u f(o_k = t|r_k = u)P(r_k = u)}.$$

$P(r_k = u)$ can be evaluated from the stationary probabilities of the Markov chain. Thus we can evaluate the terms r_{ij}. From equation (12) and the probability distribution of o_k, the term r_{ij} can easily be calculated.

It is obvious from the form of the optimal control law that no separation property holds between a controller implementing the optimal control law based on known Markov state and a Markov state estimation algorithm. In particular, even if we use a causal one-step Viterbi algorithm and feed the state derived from it into the controller which implements the optimal control law based on known Markov state, we would not obtain the lowest cost achievable with the one-step Viterbi state estimation algorithm.

4.2 Extensions

Note that the form of the optimal controller derived above is similar to the controller for the case of Markov state known, as derived in [8], except for the variables on which to condition while taking the expectation. Thus, we can go ahead and derive the optimal process state estimate and show that a separation property holds between the optimal controller and the optimal process state estimate in a manner similar to that given in the above reference.

5 Conclusions and Future Work

In this paper, we have analyzed jump linear Markov systems in which the Markov state is not known and is being estimated. We have presented stability conditions for such systems. We made the assumption that the state estimation algorithm takes only the latest measurements into account. We showed that even under the one-step Viterbi algorithm, a control law depending on the knowledge of the exact Markov state may no longer stabilize the system when we feed in the state estimate instead of the state itself. We have also presented an optimal control law by solving the LQR problem for the case when the Markov state is being estimated.

Work is being currently done to extend the results to the estimation algorithms which take the full history into account while updating the state estimate. Such algorithms, e.g, the causal Viterbi algorithm are optimal Markov state estimators but are more complicated to analyze. Initial results are promising and point to similar theorems as given in the paper.

Another possible direction for future work might be to jointly optimize the LQR problem with the estimation algorithm to see whether the Viterbi algorithm is indeed the best state estimation algorithm in this case. It would also be interesting to consider the possibility of sending data at variable rates over the network to cut down on the amount of communication costs, which will typically be high when the channel is in a Markov state corresponding to low SNR.

Acknowledgements
The authors would like to thank Abhishek Tiwari, Lun Li and Timothy Chung for discussions on this topic. Research supported in part by the AFOSR grant F49620-01-1-0460.

References

[1] D.P. Bertsekas. *Dynamic Programming and Optimal Control*. Athena Scientific, 2nd edition, 2000.

[2] W.P. Blair and D.D. Sworder. Feedback control of a class of linear discrete systems with jump parameters and quadratic cost criteria. *Int. J. Contr.*, 21(5):833–841, 1975.

[3] H. J. Chizeck, A. S. Willsky, and D. Castanon. Discrete-time Markovian-Jump Linear Quadratic optimal control. *Int. J. Contr.*, 43(1):213–231, 1986.

[4] X. Feng, K.A. Loparo, Y. Ji, and H.J. Chizeck. Stochastic stability properties of Jump Linear systems. *IEEE T. Automatic Control*, 37(1):38–53, 1992.

[5] Y. Ji and H.J. Chizeck. Controllability, stabilizability and continuous-time Markovian Jump Linear Quadratic control. *IEEE T. Automatic Control*, 35(7):777–788, 1990.

[6] R. Luck and A. Ray. An observer-based compensator for distributed delays. *Automatica*, 26(5):903–908, 1990.

[7] J. Nilsson. *Real-Time Control Systems with Delays*. PhD thesis, Department of Automatic Control, Lund Institute of Technology, 1998.

[8] J. Nilsson and B. Bernhardsson. LQG control over a Markov communication network. In *Proc. of the 36th IEEE Conf. on Decision and Control*, volume 5, pages 4586–4591, 1997.

[9] R. Niu, P. Willett, and Y. Bar-Shalom. Tracking considerations in selection of RADAR waveform for range and range-rate measurements. *IEEE T. Aero Elec Sys*, 38(2):467–487, 2002.

[10] Q. Zhang and S. A. Kassam. Finite-state Markov model for Rayleigh fading channels. *IEEE T. Communications*, 47(11):1688–1692, 1999.