Event-Triggered Output Feedback Control of Finite Horizon Discrete-time Multi-dimensional Linear Processes

Lichun Li, Michael Lemmon

***Abstract*— Event-triggered control systems are systems in which the control signal is recomputed when the plant's output signal leaves a *triggering-set*. There has been recent interest in event-triggered control systems as a means of reducing the communication load in control systems. This paper re-examines a problem [1] whose solution characterizes triggering-sets that minimize a quadratic control cost over a finite horizon subject to a hard constraint on the number of times the feedback control is computed. Computational complexity confined prior solutions of this problem to *scalar* linear systems. This paper presents an approximate solution that is suitable for multi-dimensional linear systems. This approximate solution uses families of quadratic forms to bound the value functions generated in solving the probelm. This approach has a computational complexity that is polynomial in state-space dimension and horizon length. This paper's results may therefore provide a basis for developing practical methods for the event-triggered output control of multi-dimensional discrete-time linear systems.**

I. Introduction

There has been recent interest in *event-triggered* control systems. Event-triggered controllers adapt the real-time system's task period directly in response to the application's performance [2]. Under event-triggering, the control task is only executed when the application's error signal leaves a specified *triggering-set*. Ostensibly, this error provides a measure of how valuable the current sensor data is in maintaining the overall system's closed-loop performance. Since the system state is always changing, this approach generates an *aperiodic* sequence of controller invocations. In general, the hope is that the average rate of this aperiodic task set will be much lower than the rate of a periodic task set with comparable performance levels.

There is experimental evidence to support the assertion that event-triggered feedback can maintain performance levels while reducing feedback information. Results from [3] consider a controlled scalar diffusion process where control updates are triggered when the absolute value of the system state exceeds a specified constant threshold. These results show that the event-triggered system has better performance (in the sense of a lower steady state state variance) than a comparable system with periodically triggered control updates. Such results have helped stimulate interest in using event-triggering as a means of minimizing the feedback information used in achieving control objectives.

Recent work in [4], [5] has quantified the feedback rate in state-dependent event-triggered systems. The feedback rate is quantified by the *intersampling interval*; the time between consecutive samples. These papers analytically determine the minimum intersampling interval for event-triggered systems enforcing a specified stability concept such as input-to-state stability [4] or $\mathcal{L}_2$ stability [5]. The determination of the associated intersampling interval, however, is done as an afterthought. In other words, these results first enforce the desired stability concept and then determine what the resulting intersampling interval will be. For some systems, this approach leads to a more efficient use of the feedback channel. It is also relatively easy, however, to use these methods to obtain event-triggered systems that exhibit Zeno-sampling [6]. In such cases the system generates an infinite sequence of intersampling intervals that asymptotically go to zero over a finite time interval. This leads to infinitely fast sampling of the system state, which clearly does not minimize the information rate in the feedback channel.

What is really needed is an analysis that treats control system performance and computational resource utilization within the same analytical framework. One well used framework treats the design of event-triggers as a constrained optimization problem that optimizes control system performance subject to a constraint on the feedback rate. This framework was used in [7] where the mean square control cost was minimized over an infinite horizon subject to a feedback rate constraint. Related finite horizon versions of this problem were considered in [1] and [8]. In general, these problems were all solved using dynamic programming methodologies that optimize system performance with respect to the event triggering-sets.

In practice, however, this framework was of limited utility for it was quickly recognized that the complexity of computing the optimal triggering-sets was impractical for multi-dimensional systems. As a result nearly all of the recent works [9], [10], [11] based on this framework have confined their attention to scalar linear systems. This restriction to scalar systems is of limited use in developing real-life applications of event-triggered systems. A major challenge to be addressed by the research community therefore lies in finding practical ways of extending this analytical framework to multi-dimensional systems.

First efforts at such multi-dimensional extensions were suggested in [12], [13] with respect to the infinite horizon problem posed in [7]. The approach in [12] used a single quadratic form to approximate the value function used in determining the optimal triggering-sets. This quadratic approximation was well suited to the infinite horizon problems in [7], but it was less effective in approximating the value

Lichun Li and Michael Lemmon are with the department of Electrical Engineering, University of Notre Dame, Notre Dame, IN 46556, USA. (e-mail:lli3,lemmon@nd.edu). The authors gratefully acknowledge the partial financial support of the National Science Foundation NSF-CNS-0925229.

functions for the finite horizon problem in [1]. This is because the value functions for the finite-horizon problem may not be convex. Recent work [14] suggested that this problem could be addressed by using families of quadratic forms. The results presented below show how the suggestion in [14] might be used for event-triggered output feedback control over a finite horizon.

II. Problem Statement

Consider a linear discrete-time process over a finite horizon of length $M+1$, during which only $\bar{b} \in \{0, 1, \cdots, M+1\}$ transmissions are allowed. A block diagram of the closed-loop system is shown in figure 1. This closed-loop system consists of a discrete-time linear *plant* which generates measurement sequence, a sensor subsystem which processes the measurement sequence and decide when to transmit the processed data and an actuator subsystem which uses the information sent by sensor subsystem to compute the control signal.

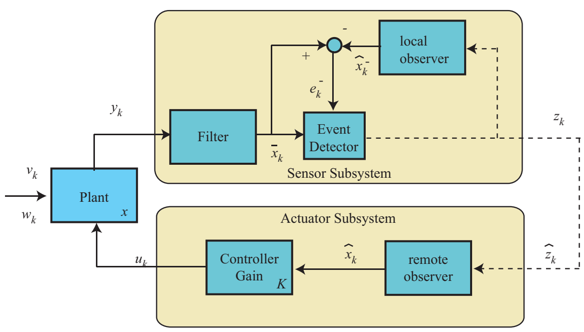

Fig. 1. Event-Triggered Control System

The plant satisfies the difference equation below

$$\begin{aligned} x_{k+1} &= Ax_k + Bu_k + w_k, \\ y_k &= Cx_k + v_k \end{aligned}$$

for $k \in [0, 1, \ldots, M]$ where A is a $n \times n$ real matrix, B is a $n \times m$ real matrix, u is the control input, and $w : [0, 1, \ldots, M] \to \mathbb{R}^n$ is a zero mean white noise process with covariance matrix Q. The initial state, x_0, is a Gaussian random variable with mean μ_0 and variance Π_0. y_k is the sensor measurement at time k. C is a real $p \times n$ matrix and $v : [0, 1, \ldots, M] \to \mathbb{R}^m$ is another zero mean white noise process with variance R. w,v and x_0 are uncorrelated with each other. We assume that (A, B, C) is controllable and observable. The sensor outputs are fed into a sensor-subsystem that decides when to transmit information to the actuator-subsystem

The sensor-subsystem consists of three components:, a *filter*, a *local observer*, and an *event detector*. Let $\mathbb{Y}_k = \{y_0, y_1, \ldots, y_k\}$ denote the sensor information available at time k. The *filter* generates a state estimate $\overline{x}$ that minimizes the mean square estimation error(MSEE) $E\left[(x_k - \overline{x}_k)^2 \,|\, \mathbb{Y}_k\right]$ at each time step conditioned on all of the sensor information received up to and including time k. These estimates are computed using a Kalman filter. The filter equations for the system are,

$$\begin{aligned} \overline{x}_k &= E\left[x_k \,|\, \mathbb{Y}_k\right] = \overline{x}_k^- + L_k(y_k - C\overline{x}_k^-) \\ \overline{P}_k &= E\left[(x_k - \overline{x}_k)^2 \,|\, \mathbb{Y}_k\right] \\ &= A\overline{P}_{k-1}A^T + Q - L_k C(A\overline{P}_{k-1}A^T + Q) \end{aligned}$$

for $k = 1, 2, \ldots, M$ where L_k is the Kalman filter gain and $\overline{x}_k^- = A\overline{x}_{k-1} + Bu_{k-1}$. The initial condition $\overline{x}_0$ is the first a posteriori update based on y_0 and $\overline{P}_0$ is the covariance of this initial estimate.

Because the sensor subsystem has access to the information received by actuator subsystem, the *local observer* can duplicate the state estimate, $\hat{x}$, made by the remote observer in the actuator subsystem. The behavior of local and remote observers will be explained later.

The *event detector* observes the filtered state, $\overline{x}_k$ and the gap between filtered state and the remote estimated state, $e_k^- = \overline{x}_k = \hat{x}_k^-$. If the vector $\begin{bmatrix} \overline{x}_k \\ e_k^- \end{bmatrix}$ lies outside the specified triggering set S_k^b, where b is the remaining transmission times, the filtered state $\overline{x}_k$ will be transmitted to the actuator subsystem. Given a set of transmission times $\{\tau^\ell\}_{\ell=1}^{\overline{b}}$, let $\overline{\mathbb{X}}_k = \{\overline{x}_{\tau^1}, \overline{x}_{\tau^2}, \ldots, \overline{x}_{\tau^{\ell(k)}}\}$ denote the filter estimates that were transmitted to the remote observer by time k where $\ell(k) = \max\{\ell : \tau^\ell \leq k\}$. This is the information set available to the remote observer at time k.

The actuator-subsystem consists of two components; a *remote observer* and the *controller gain*. The remote observer uses the received information to compute an a posteriori estimate $\hat{x}$ of the process state that minimizes the MSEE, $E\left[(x_k - \hat{x}_k)^2 \,|\, \overline{\mathbb{X}}_k\right]$, at time k conditioned on the information received up to and including time k. The a priori estimate of the remote observer, $\hat{x}^- : [0, 1, \ldots, M] \to \mathbb{R}^n$, minimizes $E\left[(x_k - \hat{x}_k)^2 \,|\, \overline{\mathbb{X}}_{k-1}\right]$, the MSEE at time k conditioned on the information received up to and including time $k-1$. These estimates take the form

$$\begin{aligned} \hat{x}_k^- &= E\left[x_k \,|\, \overline{\mathbb{X}}_{k-1}\right] = A\hat{x}_{k-1} + Bu_{k-1} \\ \hat{x}_k &= E\left[x_k \,|\, \overline{\mathbb{X}}_k\right] = \begin{cases} \hat{x}_k^- & \text{don't transmit at step } k \\ \overline{x}_k & \text{transmit at step } k \end{cases} \end{aligned}$$

where $\hat{x}_0^- = \mu_0$. This estimate is then used to compute the control, $u_k = K\hat{x}_k$, for $k = 0, 1, \ldots, M$ where K is some real $m \times n$ matrix.

For convenience, we let

$$\mathbb{S}_r^b(k) = \left\{ S_k^{\max\{0, b-k+r\}}, \ldots, S_k^{\min\{b, M+1-k\}} \right\}$$

denote the triggering sets to be used at step k with b transmissions remaining. We let $\mathbb{S}_r^b = \{\mathbb{S}_r^b(r), \ldots, \mathbb{S}_r^b(M)\}$ be the collection of all triggering-sets that will be used by the sensor-subsystem after and including time step r.

We are now in a position to formally state the problem being addressed. Consider the cost function

$$J_M(\mathbb{S}_0^{\overline{b}}) = E\left[\sum_{k=0}^{M} p_k^T Z p_k\right]$$

where $Z = \begin{bmatrix} Z_{11} & Z_{12} \\ Z_{21} & Z_{22} \end{bmatrix}$ is a symmetric and positive semi-definite $2n$ by $2n$ matrix and $p_k = \begin{bmatrix} x_k \\ \hat{e}_k \end{bmatrix}$ is the system state at time k, where $\hat{e}_k = x_k - \hat{x}_k$, is the remote state estimation error. The objective is to find the collection, $\mathbb{S}_0^{\overline{b}}$, of triggering-sets that minimizes the cost function. The optimal cost then becomes

$$J_M^* = \min_{\mathbb{S}_0^{\overline{b}}} J_M(\mathbb{S}_0^{\overline{b}})$$

III. Main Results

The problem is an optimal control problem whose controls are the triggering-sets in $\mathbb{S}_0^{\overline{b}}$. The solution may be characterized using dynamic programming techniques. Define the problem's value function as

$$V(\theta, b; r) = \min_{\mathbb{S}_r^b} \left(\sum_{k=r}^{M} p_k^T Z p_k \,|\, I_r^- = (\theta, b) \right).$$

For convenience, indicate $\begin{bmatrix} \overline{x}_r \\ e_r^- \end{bmatrix}$ by q_r^- and $\begin{bmatrix} \overline{x}_r \\ e_r \end{bmatrix}$ by q_r. I_r^- is the a priori information set at time step r consisting of an ordered pair (q_r^-, b) with b remaining transmissions. The value function is defined as the minimum cost conditioned on $q_r^- = \theta = \begin{bmatrix} \eta \\ \zeta \end{bmatrix}$ with b remaining transmissions.

Theorem 3.1: The value function satisfies

$$V(\theta, b; r) = \min \{V_{nt}(\theta, b, r), V_t(\theta, b, r)\} \quad (1)$$

where V_{nt} is the cost function without transmitting at step r and V_t is the cost function if transmitting at step r. Both of them are defined as

$$\begin{aligned} V_{nt}(\theta, b, r) &= E\left[V(q_{r+1}^-, b; r+1) \,|\, I_r = (\theta, b)\right] \\ &\quad + \theta^T Z \theta + \beta_r \end{aligned} \quad (2)$$

$$\begin{aligned} V_t(\theta, b, r) &= E\left[V(q_{r+1}^-, b-1; r+1) \,|\, I_r = (\theta_0, b-1)\right] \\ &\quad + \theta_0^T Z \theta_0 + \beta_r \end{aligned} \quad (3)$$

where I_r is the posteriori information set with ordered pair (q_r, b). $\theta = \begin{bmatrix} \eta \\ \zeta \end{bmatrix}$ and $\theta_0 = \begin{bmatrix} \eta \\ 0 \end{bmatrix}$ are the actual values of a posteriori random variable, q_r. The scalar β_r equals $\mathrm{tr}(\overline{P}_r(Z_{11} + Z_{12} + Z_{21} + Z_{22}))$.
This theorem indicates that the value function choose the smaller one between these two cost functions.

The preceding theorem shows that $V(\theta, b, r)$ can be computed through a recursion that ranges over the indices b (number of remaining transmissions) and r (current time). The initial conditions for this recursion occur when $b = 0$ or $b = M + 1 - r$ for all values of r. For the first case ($b = 0$), this corresponds to the cost of never transmitting after time step r. The other case ($b = M+1-r$) corresponds to transmitting at every single remaining time step. In both cases, the value function can be computed in closed form, and the expressions are given in appendix.

Given these initial conditions, the value function at index (b, r) may be computed from the value function at indices $(b, r+1)$ and $(b-1, r+1)$. This computational dependence in the recursion is illustrated in figure 2. This figure shows the indices including the the triggering set collection $\mathbb{S}_1^2$. The indices for the initial value functions are filled in. The order of computation used to compute $\mathbb{S}_1^2$ is shown by the arrows.

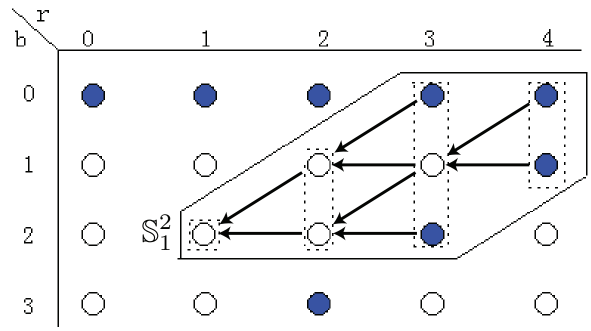

Fig. 2. Index Sets for Value Function Recursion

The selection in equation (1) defines the triggering-sets used by the event detector.

Corollary 3.2: The optimal triggering-set used at time step r with b transmissions remaining will be

$$S_r^{b*}(\eta) = \left\{ \theta = \begin{bmatrix} \eta \\ \zeta \end{bmatrix} \,|\, V_{nt}(\theta, b, r) \le V_t(\theta, b, r) \right\} \quad (4)$$

The initial triggering-sets are $S_r^{0*} = \mathbb{R}^{2n}$ and $S_r^{(M+1-r)*} = \emptyset$.

The recursion used in equations (2) and (3) may only be tractable for first order linear systems. In this case, the triggering sets are subsets of $\mathbb{R}^2$ and the bisection search from [14] may be employed to determine the triggering-sets S_r^{b*}. This is done for a specific example below. Extending this approach to multi-dimensional systems is impractical. The approach used in [14] involves computing the value function over a grid of points in the state space. Overall, there are $\overline{b}(M + 1 - \overline{b})$ triggering sets in the collection $\mathbb{S}_0^{\overline{b}*}$. If each value function is evaluated in a $2n$-dimensional space over a range of $[-c/2, c/2]$ with a granularity of ϵ, then there are a total of $\left(\frac{c}{\epsilon}\right)^{2n}$ points at which the value functions are computed. This means the computational effort required to compute $V(\theta, b; r)$ will be on the order of $O\left(\overline{b}(M + 1 - \overline{b}) \left(\frac{c}{\epsilon}\right)^{2n}\right)$. This is exponential in the state space dimension and generally $\frac{c}{\epsilon}$ will be very large. As a result this approach is impractical for all but scalar linear systems.

Since the computational complexity of the recursion in equations (2) and (3) will be prohibitively large, one must resort to approximation methods. One useful approximation [12] was developed for the infinite horizon problem considered in [7]. This approximation used a single quadratic form to over bound the value function. While this method works well for infinite horizon problems, it seems to be ill-suited for finite horizon problems. In particular, recent work [14] for the finite horizon estimation problem [15] shows that the value functions are non-convex and are therefore poorly approximated by a single quadratic form. The work in [14] suggested that a family of quadratic forms provide a much better way of approximating the value function for the estimation problem. This approach can also be adopted for the output feedback control problem considered in this paper.

The basic idea behind the approximations used in [14] is as follows. While the value function, V, is inherently non-convex due to the choice in equation 1, the functions V_t and V_{nt} may be well approximated by quadratic forms. This conjecture is based on two observations. First the initial value functions $V(\theta, b, r)$ for $b = 0$ and $b = M + 1 - r$ are quadratic and second that the recursion in equations (2) and (3) are nearly quadratic. It therefore seems possible that we can bound $V_{nt}(\theta, b, r)$ and $V_t(\theta, b, r)$ from above by a family of quadratic forms.

Proposition 3.3: There exist $\Lambda^b_{r,j} \in \mathbb{R}^{2n\times 2n}$, $\Psi^b_r \in \mathbb{R}^{n\times n}$, and scalars $c^b_{r,j}$, d^b_r for $r \in [0, 1, \ldots, M]$, $b \in [0, 1, \ldots, \overline{b}]$, and $j \in [1, 2, \ldots, \rho^b_r]$ such that

$$V_{nt}(\theta, b, r) \le \overline{V}_{nt}(\theta, b, r) = \min_{j\in[1,\ldots,\rho^b_r]} \left\{\theta^T \Lambda^b_{r,j}\theta + c^b_{r,j}\right\} \quad (5)$$

$$V_t(\theta, b, r) \le \overline{V}_t(\theta, b, r) = \eta^T \Psi^b_r \eta + d^b_r, \quad (6)$$

where ρ^b_r is a finite integer associated with step r and remaining transmissions b.

With the upper bounds of the true value functions, $\overline{V}_{nt}$ and $\overline{V}_t$, we can construct a sub-optimal triggering-set S^{b+}_r of the form

$$S^{b+}_r = \left\{\theta \in \mathbb{R}^{2n} : \overline{V}_{nt}(\theta, b, r) \le \overline{V}_t(\theta, b, r)\right\} \quad (7)$$

which is an approximation of the optimal triggering-sets, S^{b*}_r, in equation (4).

We notice that (2) and (3) add a quadratic value to the expected minimum of V_t and V_{nt}. The approximation can be done by interchanging the expectation and minimization operators as $V_{nt} = \theta^T Z\theta + \beta + E\left[\min(V_t, V_{nt})\right] \le \theta^T Z\theta + \beta + \min\left\{E[V_t], E[V_{nt}]\right\}$, where the expected values can again be represented by a family of quadratic forms. Provided the variances of the noise processes are relatively small, this approximation can be made tight.

For convenience, we let $\overline{A} = \begin{bmatrix} A+BK & -BK \\ 0 & A \end{bmatrix}$, $\overline{L}_k = \begin{bmatrix} L_k \\ L_k \end{bmatrix}$, $\beta_k = \mathrm{tr}(\overline{P}_k(Z_{11} + Z_{12} + Z_{21} + Z_{22}))$ and $S_k = CA\overline{P}_k A^T C^T + CQC^T + R$. It can be easily shown by using mathematical induction and the fact that $E\left[\min(V_t, V_{nt})\right] \le \min\left\{E[V_t], E[V_{nt}]\right\}$ that

Lemma 3.4: Equation (5) and (6) hold, if for all $b \ge 1$ and all $\overline{b} - b \le r \le M - b$,

$$\Lambda^b_{r,j} = \begin{cases} Z + \overline{A}^T \Lambda^b_{r+1,j}\overline{A} & j = 1, \ldots, \rho^b_{r+1} \\ Z + \overline{A}^T \begin{bmatrix} \Psi^b_{r+1} & 0 \\ 0 & 0 \end{bmatrix} \overline{A} & j = \rho^b_r \end{cases} \quad (8)$$

$$c^b_{r,j} = \begin{cases} c^b_{r+1,j} + \beta_r + \mathrm{tr}(\overline{\Lambda}^b_{r+1,j}) & j = 1, \ldots, \rho^b_{r+1} \\ d^b_{r+1} + \beta_r + \mathrm{tr}(\overline{\Psi}^{b-1}_{r+1}) & j = \rho^b_r \end{cases} \quad (9)$$

$$\Psi^b_r = Z_{11} + (A + BK)^T \Psi^{b-1}_{r+1}(A + BK) \quad (10)$$

$$d^b_r = \min\{\hat{\Lambda}^{b-1}_{r+1}, \hat{\Psi}^{b-1}_{r+1}\} + \beta_r \quad (11)$$

where

$$\begin{aligned} \overline{\Lambda}^b_{r+1,j} &= S_r \overline{L}^T_{r+1}\Lambda^b_{r+1,j}\overline{L}_{r+1} \\ \overline{\Psi}^{b-1}_{r+1} &= S_r L^T_{r+1}\Psi^b_{r+1}L_{r+1} \\ \hat{\Lambda}^{b-1}_{r+1} &= \min_{j\in[1,\rho^{b-1}_{r+1}]} \left[\mathrm{tr}(\overline{\Lambda}^{b-1}_{r+1,j}) + c^{b-1}_{r+1,j}\right] \\ \hat{\Psi}^{b-1}_{r+1} &= \mathrm{tr}(\overline{\Psi}^{b-1}_{r+1}) + d^{b-1}_{r+1}. \end{aligned}$$

In this case, ρ^b_r equals $M + 1 - b - r$ for $b \ge 1$, and 1 for $b = 0$. The initial condition is the same as defined in theorem 3.1.

Because the recursion used above mimics the recursions used for the original value function, we expect these bounds to be relatively tight. Precisely how tight these bounds are is still being quantified.

Computing the suboptimal triggering-sets involves a $2n$ by $2n$ matrix-matrix multiplication with a computational complexity $O((2n)^3)$. The computation of $\overline{V}_{nt}$ dominates the effort since it has the most quadratic forms to compute. One can therefore show that the effort associated with computing the suboptimal triggering set S^{b+}_r will be $O(\overline{b}(M + 1 - \overline{b})(M+2-\overline{b})(2n)^3)$. This has a complexity that is polynomial in n and quadratic in M (the length of the horizon window). The complexity is much lower than that used in computing the value functions, so these approximations may represent a practical way of implementing optimal event-triggered controllers provided the approximations are tight. Preliminary simulation results are given below to experimentally evaluate how good the approximation really is.

IV. Preliminary Simulation Results

As stated above, we'd like to experimentally evaluate how closely the approximations in equations (5) and (6) approximate the value function computed using the equations (2) and (3). We'll do this for a specific example. Because we can only compute the exact value function for scalar systems, this example focuses only on the scalar system.

The system under study is a scalar system where $A, B, C, D = 1$, $Q, R = 1$, $\mu_0, \Pi_0 = 1$, $K = -.95$, $M = 4$ and $\overline{b} = 1$. We consider a control problem without a penalty on the control input, so that $Z = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}$. The value functions and their bounds were computed using the recursions described in the preceding section. The results from this comparison are shown below in figure 3.

The left column of the top plots in figure 3 shows the value functions and their upper bounds. While it may be difficult to see, both the value function and the upper bound are shown in these graphs. If one looks closely along the plane where $\eta = 0$, one may see a white line that marks the upper bound. For $k = 0$ and $k = 1$, these plots show a small

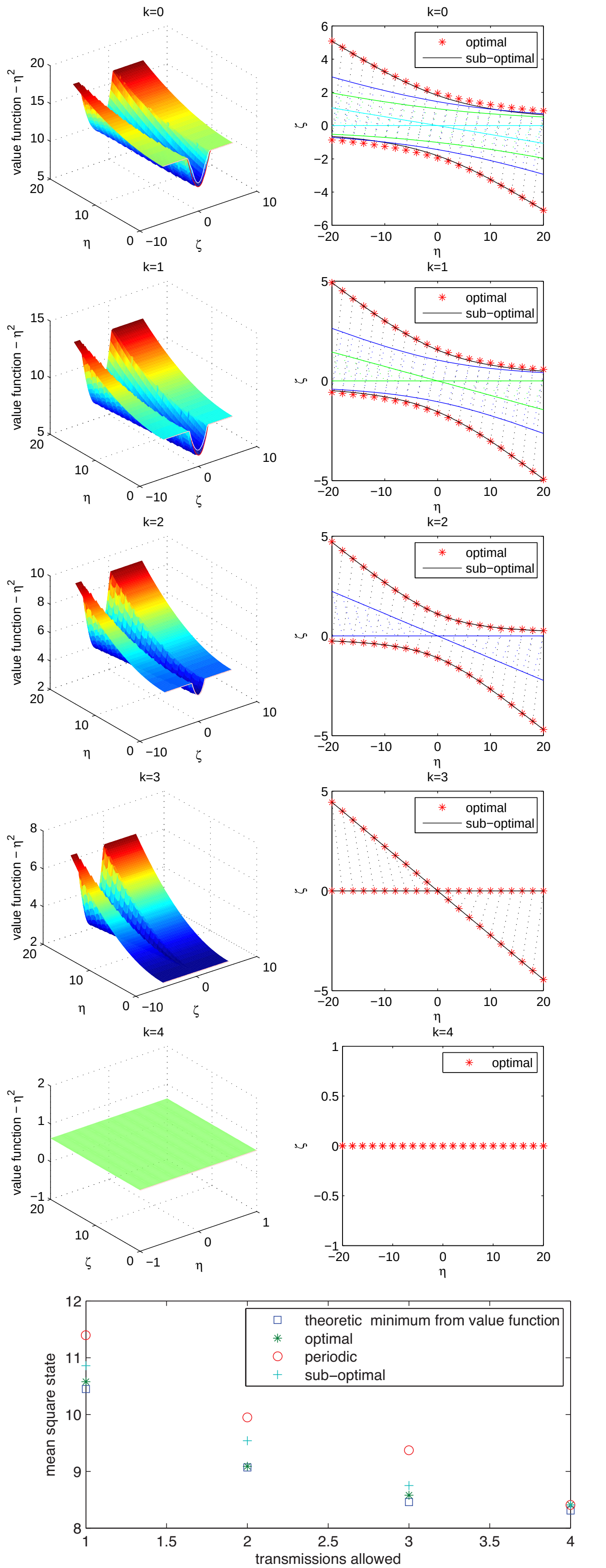

Fig. 3. Top plots show value functions and optimal/sub-optimal triggering sets. Bottom plot shows experiment results

difference between V and its bound appears. For the other values of k it is nearly impossible to see any difference. The triggering-sets are easily identified as the boundary of the deep values in these plots. These boundaries mark where V_t and V_{nt} are equal to each other. The triggering-sets are more clearly seen in the contour plots on the right column of figure 3. The boundary of the optimal triggering-set is marked by the asterisks. The boundary of the suboptimal triggering sets are marked by the solid lines. These figures show that the suboptimal and optimal triggering-sets are nearly identical with only small variations appearing for $k = 0$ and $k = 1$.

We can evaluate the performance of the system under periodic, optimal, and suboptimal event-triggering. In particular, let's vary the number of allowed transmissions, $\bar{b}$, between 1 and 4. For these values of $\bar{b}$, we computed the optimal and suboptimal triggering sets and then used these sets in a simulation of the system. The results of these simulations are shown in the bottom plot of figure 3. This figure plots the mean square state with respect to $\bar{b}$, when transmission is done using the optimal, sub-optimal and periodic triggering. One can see that the suboptimal event-triggers perform are only slightly worse than the optimal event triggering thresholds, and both of them have smaller mean square state errors than periodic triggering. Finally, we determine the actual mean square state that should have been achieved. This value matches what was achieved using the optimal event-triggers.

In this example, the complexity associated with computing and using the optimal triggering-sets is a thousand times greater than the complexity of the suboptimal triggering-sets. In particular, the optimal triggering-sets were characterized over a range of $[-20, 20]$ with a quantization level of 0.2. This requires 4×10^4 points per value function. Since there are $M + 1 - \bar{b}$ value functions, computing the thresholds requires us to store 1.6×10^5 points. These points are then used in a bisection search to determine the thresholds. This search requires $\lceil 2 \log_2(40/0.2) \rceil = 16$ steps to achieve an accuracy consistent with the quantization level of 0.2, so a total of 25×10^5 computations to determine the triggering-set thresholds. For this example there are a total of $\left(\frac{40}{0.2}\right) 2 \left(M + 1 - \bar{b}\right) \bar{b} = 1600$ thresholds to be used and checking whether a given θ lies in the triggering set or not requires $(40/0.2)2 = 400$ comparisons.

In contrast, we only need $\frac{1}{2}(M+1-\bar{b})(M+2-\bar{b}) = 10$ matrices to characterize the bounds on the value functions. Determining these matrices requires matrix-matrix multiplications on the order of $(2n)^3$ multiplies, so the total computational cost required to determine the upper bounds is $10(2n)^3 = 80$ multiplies. Evaluating the event-triggering bounds, requires all 10 matrices with a computational cost of $(2n)^2(M + 1 - r - b)$ multiplies if the current event index is (r, b). The second term represents the number of quadratic forms used in evaluating $\overline{V}_{nt}$. The worst-case occurs when $r = b = 0$, so the worst-case computational cost is $(2n)^2(M + 1) = 20$ multiplies.

From the preceding discussion it is clear that the total space-complexity of the optimal approach is on the order of

25×10^5 whereas the space-complexity of the suboptimal approach is $10(2n)^2 = 40$. The cost of evaluating an event-trigger for the optimal case is 400 whereas the suboptimal case only requires 20 multiplies. For this example, the proposed suboptimal method clearly has a much smaller computational cost than the optimal method. Moreover, the suboptimal thresholds work nearly as well as the optimal ones as indicated in the bottom plot of figure 3.

V. Summary

This paper presents a computationally tractable approach for determining suboptimal event-triggers in finite-horizon output-feedback problems. The approach relies on using a family of quadratic forms to characterize the value functions in the problem's optimal dynamic program. Our example shows that this sub-optimal sets is much more computational effective and have the similar performance as the optimal triggering sets.

VI. Appendix

Lemma 6.1: $\{I_k^-, I_k\}_{k=0}^M$ is Markov.

Proof: $I_k = \left(\begin{bmatrix} \overline{x}_k \\ e_k^- 1_{q_k^- \notin S_k^{b_k}} \end{bmatrix}, b_k - 1_{q_k^- \notin S_k^{b_k}} \right) = f(I_k^-)$. So it's easy to see that $p(I_k | I_k^-, \cdots, I_0^-) = p(I_k | I_k^-)$.

$I_{k+1}^- = \left(\overline{A} q_k, b_{k+1}\right) + \left(\overline{L}_{k+1} \bar{u}_k, 0\right)$, where $\bar{u}_k = CA\bar{e}_k + Cw_{k+1} + v_{k+1}$, $\bar{e}_k = x_k - \overline{x}_k$ is the local state estimation error. Because $\bar{u}_k$ is independent with $I_k, I_k^-, \cdots, I_0^-$. So $p(I_{k+1}^- | I_k, \cdots, I_0^-) = p(I_{k+1}^- | I_k)$. ∎

Proof of theorem 3.1

Proof: By the definition of the value function, we have

$$V(\theta, b; r) = \min_{S_r^b} (V_{nt}(\theta, b, r) 1_{\theta \in S_r^b}, V_t(\theta, b, r) 1_{\theta \notin S_r^b}),$$

where

$$V_{nt}(\theta, b, r) = \min_{\mathcal{S}_r^b - S_r^b} E(\sum_{k=r}^{M} p_k^T Z p_k | I_r^- = (\theta, b)) 1_{\theta \in S_r^b},$$

$$V_t(\theta, b, r) = \min_{\mathcal{S}_r^b - S_r^b} E(\sum_{k=r}^{M} p_k^T Z p_k | I_r^- = (\theta, b)) 1_{\theta \notin S_r^b}.$$

Consider V_{nt} first,

$$V_{nt}(\theta, b, r) = \min_{\mathcal{S}_r^b - S_r^b} E(\sum_{k=r}^{M} p_k^T Z p_k | I_r^- = (\theta, b), \theta \in S_r^b).$$

The condition is equivalent with $I_r^- = I_r = (\theta, b)$, because $\theta \in S_r^b$ means no transmission at step r. With lemma 6.1, we can derive that

$$V_{nt}(\theta, b, r) = \min_{\mathcal{S}_r^b - S_r^b} E(\sum_{k=r}^{M} p_k^T Z p_k | I_r = (\theta, b)).$$

We can pull the cost at the rth step out of the minimum, and the remaining costs are only related with $\mathcal{S}_{r+1}^b$, so

$$V_{nt}(\theta, b, r) = \theta^T Z \theta + \beta_r + \min_{\mathcal{S}_{r+1}^b} E(\sum_{k=r+1}^{M} p_k^T Z p_k | I_r = (\theta, b))$$

With lemma 6.1 and some mathematical deduction, we are able to show equation(2) holds.

Follow the same steps, (3) can be shown.

Initial conditions are given for two cases: $b = 0$ and $b + r = M + 1$. For the first case, $V_{nt}(\theta, 0, r) = \theta^T \Lambda_{r,1}^0 \theta + c_{r,1}^0$ and $V_t(\theta, 0, r) = \eta^T \Psi_r^0 \eta + d_r^0$ where $\Lambda_{r,1}^0 = \sum_{k=r}^{M} (\overline{A}^T)^{k-r} Z \overline{A}^{k-r}$, $c_{r,1}^0 = \sum_{k=1}^{M} (\beta_k + \sum_{j=r}^{k-1} tr(S_j \overline{L}_{j+1}^T (\overline{A}^T)^{k-j-1} Z \overline{A}^{k-j-1} \overline{L}_{j+1}))$ and $\Psi_r^0 = 0$, $d_r^0 = \infty$.

For the second case, $V_t(\theta, M+1-r, r) = \eta^T \Psi_r^{M+1-r} \eta + d_r^{M+1-r}$, where $\Psi_r^{M+1-r} = \sum_{k=r}^{M} ((A+BK)^T)^{k-r} Z_{11} (A+BK)^{k-r}$ and $d_r^{M+1-r} = \sum_{k=r}^{M} (\beta_k + \sum_{j=r}^{k-1} tr(S_j L_{j+1}^T ((A+BK)^T)^{k-j-1} Z_{11} (A+BK)^{k-j-1} L_{j+1}))$. ∎

References

[1] O. Imer and T. Basar, "Optimal control with limited controls," in *American Control Conference*, 2006.

[2] K.-E. Arzen, "A simple event-based PID controller," in *Procedingsof the 14th World Congress of the International Federation of Automatic Control (IFAC)*, Beijing, P.R. China, 1999.

[3] K. Astrom and B. Bernhardsson, "Comparison of Riemann and Lebesgue sampling for first order stochastic systems," in *Proceedings of the 41st IEEE Conference on Decision and Control*, vol. 2, Las Vegas, Nevada, USA, December 10-13 2002, pp. 2011–2016.

[4] P. Tabuada, "Event-triggered real-time scheduling of stabilizing control tasks," *IEEE Transactions on Automatic Control*, vol. 52, no. 9, pp. 1680–1685, September 2007.

[5] X. Wang and M. Lemmon, "Self-triggered feedback control systems with finite-gain l2 stability," *IEEE Transactions on Automatic Control*, vol. 54, no. 3, pp. 452–467, March 2009.

[6] M. Lemmon, "Event-triggered feedback in control, estimation, and optimization," to appear in book containing lectures on the 3rd WIDE summer school on networked control systems, 2010. [Online]. Available: http://www.nd.edu/ lemmon

[7] Y. Xu and J. Hespanha, "Optimal communication logics in networked control systems," in *Proceedings of the IEEE Conference on Decision and Control*, vol. 4, Nassau, Bahamas, 2004, pp. 3527–3532.

[8] M. Rabi and J. Baras, "Level-triggered control of a scalar linear system," in *Proceedings of the 16th Mediterranean Conference on Control and Automation*, Athens, Greece, July 2007.

[9] T. Henningsson, E. Johannesson, and A. Cervin, "Sporadic event-based control of first-order linear stochastic systems," *Automatica*, vol. 44, no. 11, pp. 2890–2895, November 2008.

[10] M. Rabi, K. Johansson, and M. Johansson, "Optimal stopping for event-triggered sensing and actuation," in *Proceedings of the 47th IEEE Conference on Decision and Control*, Cancun, Mexico, December 2008.

[11] L. Bao, M. Skoglund, and K. Johansson, "Encoder-decoder design for event-triggered feedback control over bandlimited channels," in *American Control Conference*, Minneapolis, Minnesota, USA, 2006.

[12] R. Cogill, S. Lall, and J. Hespanha, "A constant factor approximation algorithm for event-based sampling," in *Proceedings of the American Control Conference*, New York City, USA, July 2007.

[13] R. Cogill, "Event-based control using quadratic approximate value functions," in *IEEE Conference on Decision and Control*, Shanghai, China, 2009.

[14] L. Li and M. Lemmon, "Optimal event triggered transmission of information in distributed state estimation problems," in *American Control Conference*, Baltimore, MD, USA, 2010.

[15] O. Imer and T. Basar, "Optimal estimation with limited measurements," in *Proceedings of the IEEE Conference on Decision and Control*, Seville, Spain, 2005.