

Polynomial Approximation of Optimal Event Triggers for State Estimation Problems Using SOSTOOLS

Lichun ‘Luncinda’ Li¹ and Michael Lemmon¹

Abstract—This paper uses polynomials to approximate the optimal event triggers in state estimation problems, and efficiently computes the polynomial approximation with SOSTOOLS. From the examples under study, the polynomial approximation provides a tight lower bound on the optimal cost, and a tight upper bound on the suboptimal cost for unstable systems. The cost generated by the polynomial suboptimal event trigger is very close to the lower bound on the optimal cost. We also apply this polynomial suboptimal event trigger to an 8 dimensional 3DOF helicopter to demonstrate that one can efficiently compute the polynomial suboptimal event triggers for highly nonlinear high dimensional systems. To our best knowledge, this is the first time the suboptimal trigger has been applied to a system whose dimension is greater than 2.

I. INTRODUCTION

Event triggering is a sampling or transmission method different from periodic sampling. With event triggering, sensors or controllers only transmit information when some event occurs. In particular, information is transmitted when a measure of data ‘novelty’ exceeds a specified threshold. Previous work has demonstrated that event triggering maintains system performance while using fewer communication resources than periodic transmission [1], [2], [3]. The reason for this greater efficiency is that event triggering makes use of on-line information in making transmission decisions. This method, therefore, can adapt its usage of the communication channel to the importance of the data it must transmit.

Much of the prior work on event-triggered state estimation has formulated the problem as an optimization problem, which seeks to minimize the mean square state estimation error given limited communication. [4], [5] studied how to schedule a fixed number of transmissions to minimize the sum mean square state estimation error over a finite horizon. This is a sum cost optimization problem, and Bellman’s principle (dynamic programming) was used to solve the problem. Event triggered transmission was shown to be the optimal transmission rule in the presence of process noises. Computing the optimal event trigger, however, is not trivial, and much of the prior work has been confined to scalar systems. In [6], quadratic functions were used to approximate the optimal event trigger over a finite horizon and extended the prior work to vector cases. For infinite horizon problems, [7] introduced a communication price which is the cost paid for every transmission, and studied the optimal transmission rule that minimizes the average mean square state estimation error discounted by the communication price. This is an

average cost optimization problem, and the average cost optimality equation [8] was used to solve the problem. It was shown that event triggered transmission is the optimal transmission rule. Even for this case, the computation of the optimal event trigger was still proved to be difficult. Thus, [9] used quadratic functions to approximate the optimal event trigger for stable systems and proved that the corresponding suboptimal cost is within 6 times of the optimal cost. For unstable systems, [10], [11] obtained quadratic approximation of the optimal event triggers and explicit upper bounds on the corresponding suboptimal costs, but they could not characterize how far away the suboptimal costs were from the optimal costs.

All of this prior work was based on quadratic approximations of the optimal event trigger. This paper uses higher order polynomials to approximate the optimal event trigger. Unlike the prior work, this paper shows that SOSTOOLS provide a computationally efficient way of calculating the polynomial approximation, an upper bound on the cost of the polynomial approximation (suboptimal cost), and a lower bound on the optimal cost. The difference between the upper bound on the cost of polynomial approximation and the lower bound on the optimal cost provides an upper bound on how far away the suboptimal cost is from the optimal cost. For the examples under study, with polynomial approximation and SOSTOOLS, we obtain a tighter lower bound on the optimal cost than the one provided by [9], achieve a cost close to the lower bound on the optimal cost (and hence the optimal cost), and give a tighter upper bound on the suboptimal cost than the ones given by [10] and [11] for unstable systems.

This paper applies the polynomial event trigger to a highly nonlinear 8 dimensional 3DOF (degree of freedom) helicopter system. To our best knowledge, this is the first time the suboptimal trigger has been applied to a system whose dimension is greater than 2.

II. BACKGROUND ON AVERAGE OPTIMALITY FOR MARKOV CONTROL PROCESSES

This section presents the existing work on *average optimality for Markov control processes*, since this existing work is the basis for the main results in this paper.

Given a Markov control process and a policy π , the expected long-run **average cost** incurred by π is given by

$$V(\pi, x) = \limsup_{N \rightarrow \infty} E \left[\frac{1}{N} \sum_{t=0}^{N-1} c(x(t), a(t)) \right],$$

¹The authors are with the Department of Electrical Engineering, University of Notre Dame, Notre Dame, IN 46556, USA. (email: lli3,lemmon@nd.edu; web: www.nd.edu/~lli3,lemmon)

where $a(t)$ is the action taken at step t , and the average optimal value function $V^*(x)$ is as follows:

$$V^*(x) = \inf_{\pi \in \Pi} V(\pi, x).$$

Let \mathbf{X} be the state space. We first give the conditions for the average optimality of a Markov control process.

Assumption 2.1 (Lyapunov-like condition): 1) There exist constants $b > 0$ and $\beta \in (0, 1)$, and a (measurable) function $\omega(x) \geq 1$ for all $x \in \mathbf{X}$ such that for all $x \in \mathbf{X}$ and for all feasible actions a when the system is in state x ,

$$\int_{\mathbf{X}} \omega(y)Q(dy|x, a) \leq \beta\omega(x) + b,$$

where $Q(D|x(t), a(t)) = P(x(t+1) \in D|x(t), a(t))$

2) There exists a constant $M > 0$, such that $|c(x, a)| \leq M\omega(x)$ for all $x \in \mathbf{X}$ and for all feasible actions a when the system is in state x .

Assumption 2.2 (Finiteness condition): There are finite number of feasible actions for all $x \in \mathbf{X}$.

Assumption 2.3 (Uniform condition w.r.t. α): There exist two functions ν_1 and ν_2 , and some state $x_0 \in \mathbf{X}$, such that

$$\nu_1(x) \leq h_\alpha(x) \leq \nu_2(x), \forall x \in \mathbf{X}, \forall \alpha \in (0, 1),$$

where $h_\alpha(x) = V_\alpha^*(x) - V_\alpha^*(x_0)$, and

$$V_\alpha^*(x) = \min_{a \in A(x)} \left\{ c(x, a) + \alpha \int_{\mathbf{X}} V_\alpha^*(y)Q(dy|x, a) \right\}, \quad (1)$$

for all $x \in \mathbf{X}$.

We now give the main results about average optimality. Please check Theorem 4.1 of [12] for the proofs of Lemma 2.4 and Corollary 2.5 and 2.6.

Lemma 2.4 (Average optimality): Under Assumption 2.1, 2.2, and 2.3, the following assertions hold.

1) There exist a unique constant J^* , two functions h_1, h_2 , and a deterministic stationary rule f^* , such that for all $x \in \mathbf{X}$ the two average cost optimality inequalities hold.

$$J^* + h_1(x) \leq \min_a \left\{ c(x, a) + \int_{\mathbf{X}} h_1(y)Q(dy|x, a) \right\} \quad (2)$$

$$J^* + h_2(x) \geq \min_a \left\{ c(x, a) + \int_{\mathbf{X}} h_2(y)Q(dy|x, a) \right\} \quad (3)$$

$$= c(x, f^*(x)) + \int_{\mathbf{X}} h_2(x)Q(dy|x, f^*(x)) \quad (4)$$

2) $J^* = V^*(x)$ for all $x \in \mathbf{X}$.
3) Any deterministic stationary rule f realizing the minimum of (3) is average optimal; thus, f^* in (4) is a deterministic stationary policy for the average cost problem.

Corollary 2.5 (Lower bound on the optimal cost): Under Assumption 2.1, 2.2, and 2.3, the following assertions hold.

1) There exist a constant \underline{J} and a function h_1 , such that for all $x \in \mathbf{X}$, equation (2) holds.

$$\underline{J} + h_1(x) \leq \min_{a \in A(x)} \left\{ c(x, a) + \int_{\mathbf{X}} h_1(y)Q(dy|x, a) \right\}.$$

2) $J^* \geq \underline{J}$.

Corollary 2.6 (Upper bounds on suboptimal costs): Under Assumption 2.1, 2.2, and 2.3, the following assertions hold.

1) There exist a constant \bar{J} , a function h_2 , and a deterministic stationary rule f , such that for all $x \in \mathbf{X}$ equation (3) and (4) holds.

$$\begin{aligned} \bar{J} + h_2(x) &\geq \min_{a \in A(x)} \left\{ c(x, a) + \int_{\mathbf{X}} h_2(y)Q(dy|x, a) \right\} \\ &= c(x, f(x)) + \int_{\mathbf{X}} h_2(y)Q(dy|x, f(x)). \end{aligned}$$

2) $V(f, x) \leq \bar{J}$.

Remark 2.7: Similar results to that of Corollary 2.5 and 2.6 can also be found in [13], [14].

III. BACKGROUND ON SOSTOOLS

SOSTOOLS can efficiently solve linear polynomial inequalities using SOS technique, and hence is used to compute the polynomial approximation of the optimal event trigger in this paper. This section introduces the basic problems that SOSTOOLS solves.

The basic feasibility problem in SOS programming is formulated as finding polynomials $p_i(x)$ for $i = 1, 2, \dots, N$, such that

$$a_{0,j}(x) + \sum_{i=1}^N p_i(x)a_{i,j}(x) = 0, \text{ for } j = 1, 2, \dots, \hat{J} \quad (5)$$

$$a_{0,j}(x) + \sum_{i=1}^N p_i(x)a_{i,j}(x) \geq 0, \text{ for } j = \hat{J} + 1, \dots, J, \quad (6)$$

where $a_{i,j}(x)$ are given scalar constant coefficient polynomials.

SOS programming also solves the problem of optimizing of an objective function which is linear in the coefficients of $p_i(x)$'s. This optimization problem is formulated as searching for $p_i(x)$ for $i = 1, 2, \dots, N$ that

$$\min_c w^T c \quad (7)$$

subject to: equation (5) and (6),

where w is a given weight vector, and c is a vector consisting of the coefficients of $p_i(x)$'s.

To define and solve an SOS programming using SOSTOOLS, please check Chapter 2 of [15].

IV. EVENT TRIGGERED STATE ESTIMATION PROBLEM

A block diagram of the event triggered state estimation problem is shown in Figure 1. This system consists of three components: a *plant subsystem*, a *sensor subsystem*, and a *remote observer*.

The *plant subsystem* consists of two parts: a plant and a sensor, satisfying the following difference equation

$$\begin{aligned} x(k) &= Ax(k-1) + w(k-1), \\ y(k) &= Cx(k) + v(k), \end{aligned}$$

for $k = 1, 2, \dots, \infty$. $x : \mathbb{Z}^+ \rightarrow \mathbb{R}^n$ is the system state with initial state $x(0)$ being a Gaussian random variable with

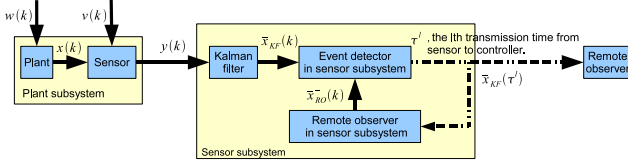


Fig. 1. Structure of the event triggered state estimation systems

mean μ_0 and variance Π_0 . w is a zero mean white Gaussian noise process with variance W . v is another zero mean white Gaussian noise process with variance V . The initial state $x(0)$, w and v are independent. The pair (A, C) is observable. $y : \mathbb{Z}^+ \rightarrow \mathbb{R}^p$ is the measurement of the plant which is fed into the sensor subsystem.

The *sensor subsystem* uses sensor measurements to decide when to transmit information to the remote observer. The sensor subsystem consists of a *Kalman filter*, a *remote observer in sensor subsystem* and an *event detector*.

The *Kalman filter* generates a filtered state $\bar{x}_{KF} : \mathbb{Z}^+ \rightarrow \mathbb{R}^n$ that minimizes the weighted mean square estimation error (MSEE), i.e.

$$\bar{x}_{KF}(k) = \min_{\bar{x}_{KF}(k)} E [\|x(k) - \bar{x}_{KF}(k)\|_Z^2 | \{y(0), \dots, y(k)\}]$$

where $Z \geq 0$ is a symmetric weighting matrix, and $\|\theta\|_Z^2 = \theta^T Z \theta$. For the process under study the filter equation is

$$\bar{x}_{KF}(k+1) = A\bar{x}_{KF}(k) + L[y(k) - CA\bar{x}_{KF}(k)], \quad (8)$$

where L is the steady Kalman gain. The steady state estimation error $e_{KF}(k) = x(k) - \bar{x}_{KF}(k)$ is a Gaussian random variable with zero mean and variance Q .

While the Kalman filter generates the most ‘knowledgable’ state estimate, the *remote observer in the sensor subsystem* duplicates the remote state estimate. With these two state estimates, the event detector knows how far away the remote state estimate is from the most ‘knowledgable’ state estimate \bar{x}_{KF} . If the remote state estimate is too far away from \bar{x}_{KF} , then \bar{x}_{KF} should be transmitted. Now, let us see how the remote observer in the sensor subsystem works. At step k , before the event detector in the sensor subsystem decides whether to transmit or not, the remote observer in the sensor subsystem produces an *a priori* remote state estimate $\bar{x}_{RO}(k)$ which will be described in detail when we introduce the remote observer. The *a priori* remote state estimate $\bar{x}_{RO}(k)$ together with the filtered state $\bar{x}_{KF}(k)$ is then handed to the event detector in the sensor subsystem to decide whether or not to transmit the filtered state $\bar{x}_{KF}(k)$ at step k .

The *event detector in the sensor subsystem* uses the *a priori gap* $e_{KF,RO}(k) = \bar{x}_{KF}(k) - \bar{x}_{RO}(k)$ to decide whether or not to transmit $\bar{x}_{KF}(k)$ to the remote observer. Let $a(k) \in \{0, 1\}$ be the action the event detector takes at step k . We say that

$$a(k) = \begin{cases} 1, & \text{if the event detector decides to transmit;} \\ 0, & \text{otherwise.} \end{cases}$$

Here, we define a large positive constant θ such that $\theta \gg \lambda > 0$, where λ is the communication price paid for

one transmission. Once $\|e_{KF,RO}(k)\|_Z^2$ is greater than θ , $\bar{x}_{KF}(k)$ has to be transmitted. Otherwise, the event detector in the sensor subsystem can choose either to transmit or not to transmit. The l th transmission time from the sensor subsystem to the remote observer is denoted as τ^l .

The *remote observer* generates the remote state estimate $\bar{x}_{RO}(k)$ to minimize the MSEE based on all the history information of the remote observer up to step k . Let $l(k) = \max \{l : \tau^l \leq k\}$ indicate the latest transmission time instants from the sensor subsystem to the remote observer. The history information $\mathbf{H}_{RO}(k)$ of the remote observer at step k is

$$\mathbf{H}_{RO}(k) = \left\{ \bar{x}_{KF}(\tau^1), \dots, \bar{x}_{KF}(\tau^{l(k)}), a(0), \dots, a(k) \right\},$$

for $k = 0, 1, \dots$ with $\mathbf{H}_{RO}(-1) = \emptyset$. To minimize the MSEE, the *a posteriori remote state estimate* $\bar{x}_{RO}(k)$ satisfies $\bar{x}_{RO}(k) = E(x(k) | \mathbf{H}_{RO}(k))$, and the *a priori remote state estimate* $\bar{x}_{RO}^-(k)$ satisfies $\bar{x}_{RO}^-(k) = E(x(k) | \mathbf{H}_{RO}(k-1))$. It was shown in [16] that the remote state estimate takes the form of

$$\bar{x}_{RO}(k) = A\bar{x}_{RO}(k-1), \text{ with } \bar{x}_{RO}^-(0) = \mu_0 \quad (9)$$

$$\bar{x}_{RO}(k) = a(k)\bar{x}_{KF}(k) + (1 - a(k))\bar{x}_{RO}^-(k). \quad (10)$$

Now, let us define the remote state estimation error $e_{RO}(k)$ as $e_{RO}(k) = x(k) - \bar{x}_{RO}(k)$. The average cost in this event triggered state estimation problem is

$$J(\{a(k)\}_{k=0}^{\infty}) = \lim_{N \rightarrow \infty} \frac{1}{N} \sum_{k=0}^{N-1} E(c(e_{RO}(k), a(k))),$$

where the cost function

$$c(e_{RO}(k), a(k)) = \|e_{RO}(k)\|_Z^2 + a(k)\lambda. \quad (11)$$

Our objective is to find a transmission rule to minimize the average cost $J(\{a(k)\}_{k=0}^{\infty})$, i.e.

$$J^* = \min_{\{a(k)\}_{k=0}^{\infty}} J(\{a(k)\}_{k=0}^{\infty}). \quad (12)$$

V. THE OPTIMAL EVENT TRIGGER

Before talking about the optimal strategy of the original problem in (12), let us simplify it first. Let $e_{KF,RO}(k) = \bar{x}_{KF}(k) - \bar{x}_{RO}(k)$ be the *posteriori gap* between the filtered state and the remote state estimate. We find that this gap $e_{KF,RO}(k)$ is orthogonal to the filtered state error $e_{KF}(k)$. This is stated in the following lemma. Please check Lemma 2 in [16] for the detailed proof.

Lemma 5.1: The filtered state error, $\bar{e}_{KF}(k)$ is orthogonal to $e_{KF,RO}(k)$, the gap between filtered state and the remote state estimate.

Since $e_{RO} = e_{KF} + e_{KF,RO}$, according to Lemma 5.1, the expected value of the cost function $E(c)$ satisfies

$$\begin{aligned} E(c(e_{RO}(k), a(k))) &= \text{trace}(QZ) + E(\|e_{KF,RO}(k)\|_Z^2) \\ &= \text{trace}(QZ) + E(c_s(e_{KF,RO}(k), a(k))) \end{aligned}$$

where

$$c_s(s, a(k)) = a(k)\lambda + (1 - a(k))\|s\|_Z^2. \quad (13)$$

Thus, our optimal problem turns to be

$$J^* = \min_{\{a(k)\}_{k=0}^{\infty}} \lim_{N \rightarrow \infty} \frac{1}{N} \sum_{k=0}^{N-1} E \left(c_s(e_{KF,RO}^-(k), a(k)) \right) + \text{trace}(QZ). \quad (14)$$

This is an average cost optimal problem in a Markov control process with the state to be $e_{KF,RO}^-(k)$. Let us analyze the dynamic behavior of $e_{KF,RO}^-(k)$. From equation (8), (9) and (10), we have

$$\begin{aligned} e_{KF,RO}^-(k+1) &= Ae_{KF,RO}^-(k) + L\tilde{y}(k+1) \\ &= (1-a(k))Ae_{KF,RO}^-(k) + L\tilde{y}(k+1) \end{aligned}$$

where $\tilde{y}(k+1) = y(k+1) - CA\bar{x}_{KF}(k) = CAe_{KF}(k) + Cw(k) + v(k+1)$. It is easy to see that $\tilde{y}(k+1)$ is a zero mean Gaussian random variable with variance $Y = CAQA^T C^T + CWC^T + V$. Let $E_h(\mu) = E(h(s))$, where s is a Gaussian random variable with mean μ and covariance LYL^T . It is easy to show that

$$\begin{aligned} &E \left(h(e_{KF,RO}^-(k+1)) | e_{KF,RO}^-(k), a(k) \right) \\ &= (1-a(k))E_h \left(Ae_{KF,RO}^-(k) \right) + a(k)E_h(0). \end{aligned}$$

According to Lemma 2.4, we have the following theorem about the optimal transmission rule and the optimal cost for the optimal problem described in (14).

Theorem 5.2: 1) There exist a unique constant ρ^* , two functions h_1^*, h_2^* , such that for all $s \in \mathbb{R}^n$ the following two average cost optimality inequalities hold.

$$\rho^* + h_1^*(s) \leq \min \left\{ \|s\|_Z^2 + E_{h_1^*}(As), \lambda + E_{h_1^*}(0) \right\}, \quad (15)$$

$$\rho^* + h_2^*(s) \geq \min \left\{ \|s\|_Z^2 + E_{h_2^*}(As), \lambda + E_{h_2^*}(0) \right\}. \quad (16)$$

- 2) The optimal cost $J^* = \rho^* + \text{trace}(QZ)$.
3) There exists a deterministic stationary optimal transition rule, i.e. *the optimal event trigger*, which is

$$a(k) = \begin{cases} 1, & \text{if } \varepsilon^*(e_{KF,RO}^-(k)) > 0, \\ 0, & \text{otherwise,} \end{cases}$$

where $\varepsilon^*(s) = \max\{\|s\|_Z^2 + E_{h_2^*}(As) - \lambda - E_{h_2^*}(0), \|s\|_Z^2 - \theta\}$.

Proof: According to Lemma 2.4, if we show that Assumption 2.1, 2.2 and 2.3 hold, then this theorem is true.

To check Assumption 2.1, we choose $\omega(e_{KF,RO}^-(k)) = \theta$, $\beta = 0.5$, $b = \theta$, and $M = 1$. It is easy to see that for any $(x, a) \in \mathbf{K}$,

$$\int \omega(y)Q(dy|x, a) = \theta \leq 0.5\theta + \theta = \beta\omega(x) + b,$$

which demonstrates part 1) of Assumption 2.1. For part 2) of Assumption 2.1, if $\|e_{KF,RO}^-(k)\|_Z^2 \leq \theta$, the feasible action set is $\{0, 1\}$, and

$$|c_s(e_{KF,RO}^-(k), a(k))| \leq \theta = M\omega(e_{KF,RO}^-(k)), \forall a \in \{0, 1\}.$$

Otherwise, the feasible action set is $\{1\}$, and it is easy to see that $c_s(e_{KF,RO}^-(k), 1) = \lambda < \theta = M\omega(e_{KF,RO}^-(k))$.

It is obvious that there are only finite actions for all $e_{KF,RO}^-(k) \in \mathbb{R}^n$. So Assumption 2.2 holds.

Now, let us look at assumption 2.3. Note that $V_\alpha^*(x)$ satisfies equation (1). In our case, equation (1) takes the form of $V_\alpha^*(s) = \min \left\{ \|s\|_Z^2 + \alpha E_{V_\alpha^*}(As), \lambda + \alpha E_{V_\alpha^*}(0) \right\}$. Let $h_\alpha(s) = V_\alpha^*(s) - V_\alpha^*(0)$. we have

$$\begin{aligned} h_\alpha(s) &\leq \lambda + \alpha E_{V_\alpha^*}(0) - \min \left\{ \alpha E_{V_\alpha^*}(0), \lambda + \alpha E_{V_\alpha^*}(0) \right\} \\ &= \lambda. \end{aligned}$$

Meanwhile, $h_\alpha(s) > -\alpha E_{V_\alpha^*}(0)$. Thus, assumption 2.3 also holds, and theorem 5.2 is true. \blacksquare

Remark 5.3: 1) The average cost optimality inequalities (2) and (3) are more general sufficient conditions than the average cost optimality equation derived in [7] for the optimal event trigger, since the average cost optimality equation in [7] is a special case of the average cost optimality inequalities (2) and (3) when $h_1^* = h_2^*$. Besides, by using the existing results on average optimality, this paper provides clearer and more straightforward proof than [7].

2) It is hard to find an analytic solution to equation (15) and (16). Although we can iteratively compute the solution of equation (15) and (16) by value iteration [17] or policy iteration [18], the computational complexity increases dramatically with respect to the state dimension.

VI. POLYNOMIAL APPROXIMATION OF THE OPTIMAL EVENT TRIGGER

Since it is intractable to compute the optimal event trigger for multi-dimensional systems, an alternative is to use an approximation of the optimal event trigger. This approximation should be easy to compute, and the corresponding cost should be close to the optimal cost. This section uses a polynomial function to approximate the optimal event trigger, and gives an explicit SOSTOOLS algorithm to compute the polynomial approximation and an upper bound on the cost of this polynomial approximation. To characterize how good the polynomial approximation is, this section also provides a lower bound on the optimal cost and the corresponding SOSTOOLS algorithm to compute the lower bound.

A. Computing a polynomial approximation and an upper bound on its cost

In the proof of Theorem 5.2, we have shown that assumption (2.1), (2.2) and (2.3) all hold. According to Corollary 2.6, there always exist an approximation of the optimal event trigger, and an upper bound on the cost of the approximation.

Theorem 6.1: 1) There exist a constant ρ_2 and a function h_2 , such that for all $s \in \mathbb{R}^n$,

$$\rho_2 + h_2(s) \geq \min \left\{ \|s\|_Z^2 + E_{h_2}(As), \lambda + E_{h_2}(0) \right\}. \quad (17)$$

2) The following deterministic stationary transmission rule realizing the minimum of equation (17).

$$a(k) = \begin{cases} 1, & \text{if } \varepsilon(e_{KF,RO}^-(k)) \geq 0; \\ 0, & \text{otherwise.} \end{cases} \quad (18)$$

where $\varepsilon(s) = \max\{\|s\|_Z^2 + E_{h_2}(As) - \lambda - E_{h_2}(0), \|s\|_Z^2 - \theta\}$.

3) With the strategy $a(k)$ defined in equation (18), the average cost $J(\{a(k)\}_{k=0}^{\infty})$ satisfies

$$J(\{a(k)\}_{k=0}^{\infty}) < \rho_2 + \text{trace}(QZ).$$

Now, we need to find a way to solve the inequality (17). The basic idea of inequality (17) is that when s is in a neighborhood of the origin, the first term in the minimum equation dominates. Otherwise, the second term in the minimum equation dominates. Based on this idea, using the indicator function 1_{Ω} , we have

$$\begin{aligned} & \rho_2 + h_2(s) \\ & \geq \|s\|_Z^2 + E_{h_2}(As) - (\|s\|_Z^2 + E_{h_2}(As)) \left(1 - 1_{\|s\|_Z^2 \leq d}\right), \\ & \rho_2 + h_2(s) \geq \lambda + E_{h_2}(0) - (\lambda + E_{h_2}(0)) 1_{\|s\|_Z^2 \leq d}. \end{aligned}$$

We know that for some big enough positive constant ϕ ,

$$1_{\|s\|_Z^2 \leq d} \approx \frac{1}{1 + \left(\frac{\|s\|_Z^2}{d}\right)^{\phi}}.$$

Meanwhile, to make sure equation (17) holds, we have

$$\rho_2 + h_2(s) \geq \|s\|_Z^2 + E_{h_2}(As) \quad (19)$$

$$- (\|s\|_Z^2 + E_{h_2}(As)) \left(1 - \frac{1}{1 + \left(\frac{\|s\|_Z^2}{d}\right)^{\phi}}\right) (\|s\|_Z^2 - d),$$

$$\rho_2 + h_2(s) \geq \lambda + E_{h_2}(0) \quad (20)$$

$$- (\lambda + E_{h_2}(0)) \frac{1}{1 + \left(\frac{\|s\|_Z^2}{d}\right)^{\phi}} (d - \|s\|_Z^2).$$

It is easy to verify that equation (19) and (20) imply equation (17).

To conclude the above discussion, we have the following corollary.

Corollary 6.2: Given a positive constant d and a positive integer ϕ . If there exist a constant ρ_2 , and a polynomial function h_2 such that equation (19) and (20) hold for all $s \in \mathbb{R}^n$, then assertion 2) and 3) in Theorem 6.1 are true.

Equation (19) and (20) can be solved efficiently by SOS-TOOLS. Here, we give an explicit SOS algorithm to compute ρ_2 and h_2 , and hence the polynomial event trigger and the upper bound on the cost of the polynomial event trigger.

Algorithm 6.3 (Compute polynomial approximation I): Find a polynomial h_2 such that ρ_2 is minimized subject to

$$\begin{aligned} & (\rho_2 + h_2(s) - \|s\|_Z^2 - E_{h_2}(As)) \left(1 + \left(\frac{\|s\|_Z^2}{d}\right)^{\phi}\right) \\ & + (\|s\|_Z^2 + E_{h_2}(As)) \left(\frac{\|s\|_Z^2}{d}\right)^{\phi} (\|s\|_Z^2 - d) \geq 0 \\ & (\rho_2 + h_2(s) - \lambda - E_{h_2}(0)) \left(1 + \left(\frac{\|s\|_Z^2}{d}\right)^{\phi}\right) \\ & + (\lambda + E_{h_2}(0)) (d - \|s\|_Z^2) \geq 0 \end{aligned}$$

Remark 6.4: We need adjust ϕ and d such that SOS-TOOLS can provide as small as possible ρ_2 . Larger ϕ

provides smaller ρ_2 , but consumes more computation effort. So, ϕ can be chosen to be large enough such that the computation time is not too long. d can be adjusted using bisection method. According to our experience, there exists a d^* such that SOSTOOLS provides the smallest ρ_2 . We can use bisection method to find d^* very quickly.

Notice that $\left(\frac{\|s\|_Z^2}{d}\right)^{\phi}$ in equation (19) and (20) adds extra terms in the polynomial inequalities, and hence increases the computational complexity. Especially when the state dimension is higher than 4, the computation time increases dramatically. So, we need another set of inequalities to compute the suboptimal strategy for high dimensional system. This set of inequalities should have fewer terms, and hence lower computational complexity than inequality (19) and (20). Thus, we have the following corollary.

Corollary 6.5: Given positive constants d_1, d_2, d_3 and positive integers ϕ and δ , where $\delta < 2\phi$. If there exist a constant ρ_2 , and a polynomial function h_2 such that

$$\rho_2 + h_2(s) \geq \|s\|_Z^2 + E_{h_2}(As) \quad (21)$$

$$\begin{aligned} & + (\|s\|_Z^2 + E_{h_2}(As)) \frac{0.5 - \frac{\sum_{i=1}^n s_i^{2\phi}}{d_3}}{1 + \frac{\sum_{i=1}^n s_i^{2\phi-\delta}}{d_1}}, \\ & \rho_2 + h_2(s) \geq \lambda + E_{h_2}(0) + (\lambda + E_{h_2}(0)) \frac{\frac{\sum_{i=1}^n s_i^{2\phi}}{d_3} - 0.5}{1 + \frac{\sum_{i=1}^n s_i^{2\phi+\delta}}{d_1}}, \end{aligned} \quad (22)$$

hold for all $s \in \mathbb{R}^n$, then assertion 2) and 3) in Theorem 6.1 are true.

Equation (21) and (22) can be solved using the following SOS algorithm.

Algorithm 6.6 (Compute polynomial approximation II): Find a polynomial h_2 such that ρ_2 is minimized subject to

$$\begin{aligned} & (\rho_2 + h_2(s) - \|s\|_Z^2 - E_{h_2}(As)) \left(1 + \frac{\sum_{i=1}^n s_i^{2\phi-\delta}}{d_1}\right) \\ & + (\|s\|_Z^2 + E_{h_2}(As)) \left(0.5 - \frac{\sum_{i=1}^n s_i^{2\phi}}{d_3}\right) \geq 0 \\ & (\rho_2 + h_2(s) - \lambda - E_{h_2}(0)) \left(1 + \frac{\sum_{i=1}^n s_i^{2\phi+\delta}}{d_1}\right) \\ & + (\lambda + E_{h_2}(0)) \left(\frac{\sum_{i=1}^n s_i^{2\phi}}{d_3} - 0.5\right) \geq 0 \end{aligned}$$

This is an optimization problem in SOSTOOLS described in (7), and we can use SOSTOOLS to solve it.

Now, we know how to calculate the suboptimal strategy and the associated upper bound on the average cost. We would like to know how good this suboptimal strategy is. We can use the difference between the upper bound on the suboptimal cost and the lower bound on the optimal cost to characterize how good the suboptimal strategy is. The next subsection will talk about how to use SOSTOOLS to compute the lower bound on the optimal cost.

B. Computing a lower bound on the optimal cost

Since Assumption 2.1, 2.2 and 2.3 are all true (shown in the proof of Theorem 5.2), according to corollary 2.5, we have the following theorem.

Theorem 6.7: 1) There exists a constant ρ_1 and a polynomial function h_1 , such that for all $s \in \mathbb{R}^n$

$$\rho_1 + h_1(s) \leq \min \{ \|s\|_Z^2 + E_{h_1}(As), \lambda + E_{h_1}(0) \}.$$

2) $J^* \geq \rho_1 + \text{trace}(QZ)$.

The lower bound on the optimal cost can be computed using SOSTOOLS, and the SOSTOOLS algorithm is given below.

Algorithm 6.8 (Lower bound on optimal cost): Find a polynomial h_1 to minimize the constant $-\rho_1$ subject to

$$\begin{aligned} -\rho_1 - h_1(s) + \|s\|_Z^2 + E_{h_1}(As) &\geq 0 \\ -\rho_1 - h_1(s) + \lambda + E_{h_1}(0) &\geq 0 \end{aligned}$$

VII. MATHEMATICAL EXAMPLES

This section uses two examples to test how SOSTOOLS works for obtaining the suboptimal strategy described in Corollary 6.2. We are interested in the elapsed real time for computing the suboptimal strategy T , the upper bound on the cost of the suboptimal strategy \bar{J} , the actual cost of the suboptimal triggering set J , and the lower bound on the optimal cost \underline{J} . We would also like to compare these results with the existing literature. The two examples were run on a Microsoft Windows XP system with 2.99 GHz CPU and 3.37GB of RAM.

Consider a marginally stable system as below

$$\begin{aligned} x(k+1) &= \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} x(k) + w(k) \\ y(k) &= x(k), \end{aligned}$$

with covariance matrix $W = \begin{bmatrix} 0.03 & -0.02 \\ -0.02 & 0.04 \end{bmatrix}$, the weight matrix $Z = \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix}$, and the communication price $\lambda = 20$. This is the same example used in [9], and we would like to compare the results in this paper with the results in [9] and [10].

Let $\phi = 10$ and $d = 2.9$. We first produce a symmetric polynomial $h_2(x)$ which contains all possible monomials whose degrees are even and no greater than a positive integer D_2 , and then calculate $E_{h_2}(Ax)$ and $E_{h_2}(0)$. With $h_2(x)$, $E_{h_2}(Ax)$ and $E_{h_2}(0)$, we use SOSTOOLS to solve the problem described in Algorithm 6.3, and obtain the polynomial approximation and the upper bound \bar{J} on the cost. The elapsed real time for computing the polynomial approximation is indicated by T_2 . After obtaining the polynomial approximation, we apply the polynomial event trigger to the state estimation system, and run the system for 3000 step to get the actual cost J . Finally, we generate a polynomial $h_1(x)$ which contains all possible monomials whose degrees are no greater than a given positive integer D_1 , and use Algorithm 6.8 to compute the lower bound \underline{J} on the optimal cost. The elapsed real time for computing the lower bound on the optimal cost is indicated by T_1 . The results are given

	D_2	\bar{J}	T_2	J	D_1	\underline{J}	T_1
SOS	2	4.52	3.48s	1.37	3	0	0.14s
SOS	4	4.52	5.34s	1.38	5	1.35	0.094s
SOS	6	×	×	×	8	1.35	0.375s
[10]	2	3.78	×	1.38	×	×	×
[9]	2	2.74	×	1.53	2	0.46	×

TABLE I
RESULTS OF THE STABLE SYSTEM.

in Table I, where \times indicates no feasible solution or the information is not available in the prior work.

From Table I, we can see that as the highest degree in h_2 increases, the upper bound \bar{J} on the cost of the polynomial approximation remains the same. Compared with the methods in [10] and [9], the upper bound \bar{J} on the cost calculated from SOSTOOLS is the highest one, i.e. the least tightest one, and [9] provides the tightest upper bound. The actual costs J of the polynomial approximations with different highest degree D_2 are similar, and are similar to the actual cost of the quadratic approximation in [10]. The quadratic approximation in [9] gives the highest actual cost. The SOSTOOLS algorithm obtains the highest lower bound on the optimal cost when the highest degree in h_1 is larger than 3. This lower bound calculated by SOSTOOLS is close to the actual cost, which means that the polynomial approximation is close to the optimal cost and the lower bound on the optimal cost is tight.

Next, we consider an unstable system.

$$\begin{aligned} x(k+1) &= \begin{bmatrix} 0.95 & 1 \\ 0 & 1.01 \end{bmatrix} x(k) + w(k) \\ y(k) &= \begin{bmatrix} 0.1 & 1 \end{bmatrix} x(k) + v, \end{aligned}$$

with the covariance matrix $W = \begin{bmatrix} 0.2 & 0 \\ 0 & 0.2 \end{bmatrix}$ and $V = 0.3$, the weight matrix $Z = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$, and the communication price $\lambda = 5$. We would like to compare the results in this paper with the results in [10] and [11].

Let $\phi = 10$ and $d = 1.25$. The polynomial approximation, its upper bound and the lower bound on the optimal cost can be calculated using SOSTOOLS following the same steps provided for the marginally stable system. We then apply the polynomial approximation in the state estimation system, and obtain the actual cost of the polynomial approximation. Table II provides the relative results. \times indicates no feasible solution or the information is not available in the prior work.

From Table II, we find that with SOSTOOLS algorithm, as we increase the highest degree of h_2 , the upper bound on the cost of the polynomial approximation decreases, and the elapsed real time for computing the upper bound increases. When D_2 is 12, the upper bound on the cost of the polynomial approximation decreases to 3.49, and the elapsed real time for computing the polynomial approximation is 25.7s. Compared with the work in [10] and [11], the SOSTOOLS algorithm proposed in this paper provides the tightest upper bound. The actual costs of the polynomial approximations

	D_2	\bar{J}	T_2	J	D_1	\underline{J}	T_1
SOS	2	5.76	2.7s	3.31	3	1.82	0.17s
SOS	4	5.74	4.5s	3.28	5	2.7	0.39s
SOS	6	3.83	7.7s	3.36	7	2.72	0.56s
SOS	8	3.74	12.7s	3.5	9	×	0.75s
SOS	10	3.51	20.5s	3.5	11	×	0.56s
SOS	12	3.49	25.7s	3.37	13	3.23	0.78s
[10]	2	5.57	×	3.32	×	×	×
[11]	2	7.69	×	4.39	×	×	×

TABLE II
RESULTS OF THE UNSTABLE SYSTEM.

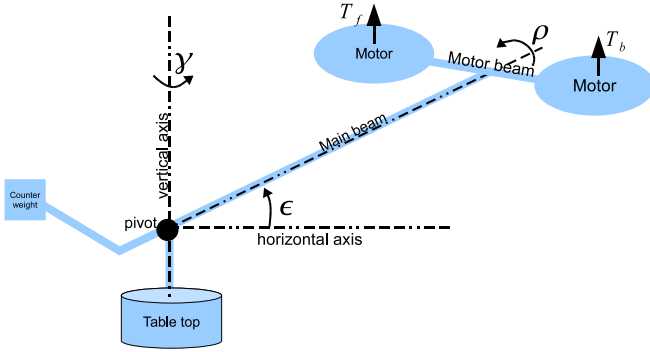


Fig. 2. Schematic of the 3DOF helicopter. ϵ : elevation, ρ : pitch, γ : travel.

with different highest degree are similar, and are similar to the actual cost given by [10]. [11] gives the highest actual cost. For unstable systems, only this paper talks about how to calculate a lower bound on the optimal cost. The lower bound on the optimal cost calculated using SOSTOOLS algorithm increases as the highest degree in h_1 increases. When D_1 is 13, the lower bound on the optimal cost is 3.23 which is close to the actual cost 3.37 and the upper bound on the cost 3.49 when $D_2 = 12$. This fact means that the polynomial approximation is close to the optimal event trigger, and both the upper bound on the actual cost and the lower bound on the optimal cost are tight.

From the two examples, we have the following conclusion. For both stable and unstable systems, the SOSTOOLS algorithm provides a tight lower bound on the optimal cost when the highest degree of each term in h_1 is large enough. For both stable and unstable systems, the polynomial approximation calculated by the SOSTOOLS algorithm is close to the optimal strategy. For unstable systems, the upper bound on the suboptimal cost calculated using SOS program is tight when the highest degree of h_2 is large enough. But for stable systems, [14] gives a tighter upper bound on the suboptimal cost than the SOSTOOLS algorithm proposed in this paper.

VIII. APPLICATION IN A 3DOF HELICOPTER

Besides the mathematical examples in section VII, we are also interested in applying the polynomial approximation of optimal event trigger proposed in this paper to a non-trivial system, and take the 3 DOF helicopter as a study case. This experiment is done on a computer with 2.99 GHz CPU and 3.37GB of RAM.

l_a	0.67 m	ϵ_0	-0.136 rad
l_h	0.177 m	c_ϵ	0.18 $kg.m^2/s$
l_w	0.48 m	c_ρ	0.003 $kg.m^2/s$
d	0.04 m	c_γ	0.25 $kg.m^2/s$
M	1.4611 kg	$c_{\gamma\rho}$	0.003 $kg.m^2/s$
m	2 kg	J_ϵ	3.5 $kg.m^2$
M_{bf}	0.29 kg	J_ρ	0.01 $kg.m^2$
g	9.8 m/s^2	J_γ	4 $kg.m^2$

TABLE III
3DOF HELICOPTER PARAMETER VALUES

Figure 2 gives the basic schematic of the 3DOF helicopter. The 3DOF helicopter consists of three subsystems: elevation (ϵ), pitch (ρ) and travel (γ). Elevation is the angle between the main beam and the horizontal axis, pitch is the angle that the motor beam moves around the main beam, and travel is the angle that the main beam moves around the vertical axis. T_f and T_b are the front and back thrust generated by the DC motors. Our objective is to control the 3DOF helicopter to follow a commanded elevation and a commanded travel rate.

The system dynamic is described by the following equations

$$\begin{aligned}
 J_\epsilon \ddot{\epsilon}_m &= -\sqrt{((ml_w - Ml_a)g)^2 + ((m + M)gd)^2} \sin(\epsilon_m) \\
 &\quad + T_{col} \cos(\rho)(l_a + d \tan(\epsilon_m + \epsilon_0)) - c_\epsilon \dot{\epsilon}_m, \\
 J_\rho \ddot{\rho} &= T_{cyc} l_h - M_{bf} g d \sin(\rho) - c_\rho \dot{\rho} + c_{\gamma\rho} \dot{\gamma}, \\
 J_\gamma \ddot{\gamma} &= -l_a T_{col} \sin \rho \cos \epsilon - c_\gamma \dot{\gamma},
 \end{aligned}$$

The parameter values are given in Table III.

Neglecting the non-dominant terms and under the assumption that $\sin(\rho) \approx \rho$ and $\sin(\epsilon_m) \approx \epsilon_m$, the model of 3DOF helicopter can be simplified as

$$\begin{aligned}
 J_\epsilon \ddot{\epsilon}_m &= -\sqrt{((ml_w - Ml_a)g)^2 + ((m + M)gd)^2} \epsilon_m \\
 &\quad + c_\epsilon \dot{\epsilon}_m + l_a u_\epsilon
 \end{aligned} \tag{23}$$

$$J_\rho \ddot{\rho} = -M_{bf} g d \rho - c_\rho \dot{\rho} - c_{\gamma\rho} \dot{\gamma} + l_h u_\rho \tag{24}$$

$$J_\gamma \ddot{\gamma} = -c_\gamma \dot{\gamma} - l_a u_\gamma, \tag{25}$$

where $u_\epsilon = T_{col} \cos(\rho)$, $u_\rho = T_{cyc}$, and $u_\gamma = T_{col} \sin(\rho) \cos(\epsilon)$. ϵ_m , ρ and γ are measurements with measurement noise variances to be $1.857e - 6$, $1.857e - 6$, and $1.857e - 8$, respectively. u_ϵ and u_ρ are PID controllers with the PID gains to be $[44 \ 7 \ 68]$ and $[30 \ 3 \ 11]$, respectively. u_γ is a PI controller with the PI gain to be $[22 \ 14]$. All the controllers only use remote state estimates to calculate the control inputs.

From equation (23), (24) and (25), we can see that the helicopter system is decomposed into 2 decoupled subsystems: elevation subsystem and pitch-travel subsystems. We will compute a polynomial event trigger for each subsystem.

The elevation subsystem is a 3 dimensional system with its state $x_\epsilon(t) = [\int_{s=0}^t \epsilon_m(s) ds \ \epsilon_m(t) \ \dot{\epsilon}_m(t)]^T$. We assume that both $\int_{s=0}^t \epsilon_m(s) ds$ and $\epsilon_m(t)$ are measurable. The communication price λ for elevation subsystem is 0.1. Let $d = 0.05$, $\phi = 2$, weight matrix $Z = \text{diag}([2 \ 7 \ 1])$, and the highest degree of h_2 be 6. Using the Algorithm 6.3, we obtain a polynomial approximation within 5 seconds.

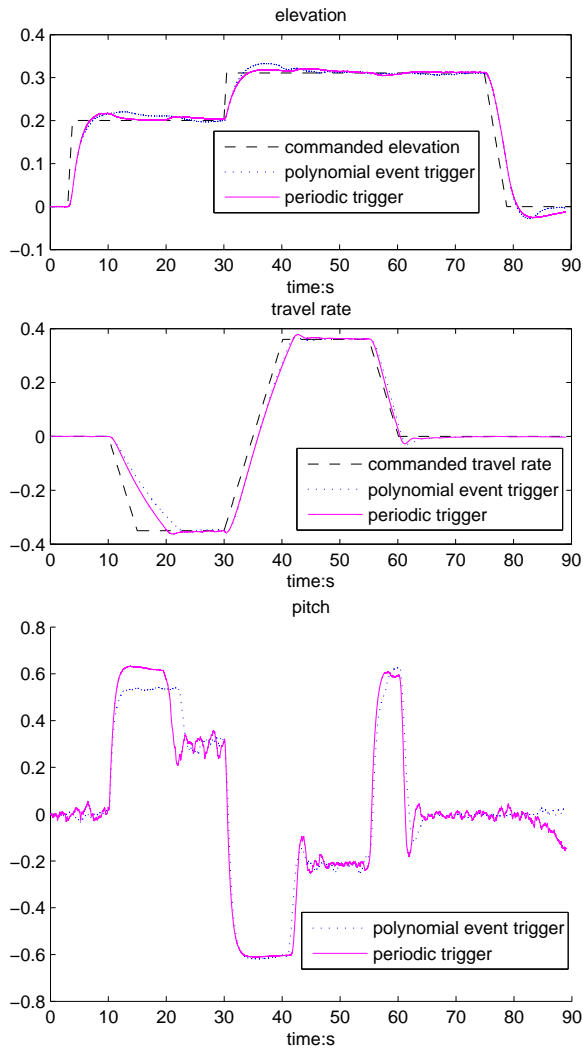


Fig. 3. The elevation, travel rate and pitch performance of the 3DOF helicopter with the remote state estimate system.

The pitch-travel subsystem is a 5 dimensional system with its state $x_{p\gamma}(t) = [\int_{s=0}^t \rho(s)ds \ \rho(t) \ \dot{\rho}(t) \ \gamma(t) \ \dot{\gamma}(t)]^T$, where $\int_{s=0}^t \rho(s)ds$, $\rho(t)$ and $\gamma(t)$ are measurable. The communication price λ for the pitch-travel subsystem is 0.1, and the weight matrix is $Z = \text{diag}([1 \ 40 \ 1 \ 1 \ 20])$. We first use Algorithm 6.3 to calculate the polynomial approximation, but it takes too long to obtain a feasible solution (more than 15 minutes). So we turn to using Algorithm 6.6 with $d_1 = 0.025$, $d_2 = 0.025$, $d_3 = 0.06$, $\phi = 1$, $\delta = 1$ and the highest degree in h_2 be 4. It takes 526 seconds to calculate a polynomial approximation.

We apply the remote state estimate system to the 3DOF helicopter, and run the system for 89 seconds. The elevation subsystem transmits 901 times and the pitch-travel subsystem transmits 1077 times. We, then, use the average period as the period for periodic transmission, and run the system again. The elevation, travel rate and pitch performances for both polynomial event trigger and periodic trigger are shown in Figure 3.

The top two plots of Figure 3 show the elevation and travel rate, respectively. The x -axis indicates the time, and the y -axis indicate the angle measured in rad. we can see that both the polynomial event trigger (dotted line) and the periodic trigger (solid line) track the commanded signals (dashed line) with small overshoot and 0 steady error, and provides similar performances. The bottom plot shows the pitch angle of the 3DOF helicopter with x -axis indicating time and y -axis indicating pitch angle. At time instants 22s, 44s, and 62s, polynomial event trigger provides smaller overshoot than the periodic trigger. From 20s to 30s, polynomial event trigger gives smaller oscillation than the periodic trigger. Thus, we say that polynomial event trigger provides better transient performance than the periodic trigger.

REFERENCES

- [1] K. Astrom and B. Bernhardsson, "Comparison of Riemann and Lebesgue sampling for first order stochastic systems," in *Decision and Control, 2002, Proceedings of the 41st IEEE Conference on*, vol. 2. IEEE, 2002, pp. 2011–2016.
- [2] P. Tabuada, "Event-triggered real-time scheduling of stabilizing control tasks," in *Automatic Control, IEEE Transactions on*, vol. 52, no. 9. IEEE, 2007, pp. 1680–1685.
- [3] X. Wang and M. Lemmon, "Self-Triggered Feedback Control Systems With Finite-Gain L_2 Stability," in *Automatic Control, IEEE Transactions on*, vol. 54, no. 3. IEEE, 2009, pp. 452–467.
- [4] O. Imer, "Optimal estimation and control under communication network constraints," Ph.D. dissertation, University of Illinois at Urbana-Champaign, 2006.
- [5] M. Rabi, K. Johansson, and M. Johansson, "Optimal stopping for event-triggered sensing and actuation," in *Decision and Control, 2008. CDC 2008. 47th IEEE Conference on*. IEEE, 2008, pp. 3607–3612.
- [6] L. Li, M. Lemmon, and X. Wang, "Event-triggered state estimation in vector linear processes," in *American Control Conference (ACC), 2010*. IEEE, 2010, pp. 2138–2143.
- [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] A. Arapostathis, V. Borkar, E. Fernández-Gaucherand, M. Ghosh, and S. Marcus, "Discrete-time controlled Markov processes with average cost criterion: a survey." ISR; TR 1991-109, 1991.
- [9] R. Cogill, S. Lall, and J. Hespanha, "A constant factor approximation algorithm for event-based sampling," in *American Control Conference, 2007. ACC'07*. IEEE, 2007, pp. 305–311.
- [10] L. Li and M. Lemmon, "Performance and average sampling period of sub-optimal triggering event in event triggered state estimation," in *conference of decision and control*. IEEE, 2011.
- [11] R. Cogill, "Event-based control using quadratic approximate value functions," in *Decision and Control, 2009 held jointly with the 2009 28th Chinese Control Conference. CDC/CCC 2009. Proceedings of the 48th IEEE Conference on*. IEEE, 2009, pp. 5883–5888.
- [12] X. Guo and Q. Zhu, "Average optimality for markov decision processes in borel spaces: a new condition and approach," *Journal of Applied Probability*, vol. 43, no. 2, pp. 318–334, 2006.
- [13] R. Cogill and S. Lall, "Suboptimality bounds in stochastic control: A queueing example," in *American Control Conference, 2006*. IEEE, 2006, pp. 1642–1647.
- [14] R. Cogill, S. Lall, and J. Hespanha, "A constant factor approximation algorithm for event-based sampling." Springer, 2010, pp. 51–60.
- [15] S. Prajna, A. Papachristodoulou, P. Seiler, and P. Parrilo, "Sostools: Sum of squares optimization toolbox for matlab," *Users guide*, 2004.
- [16] L. L. Li and M. Lemmon, "Weakly coupled transmissions in networked event triggered output feedback systems," *submitted to Discrete Event Dynamic systems*, 2012.
- [17] E. Gordienko and O. Hernández-Lerma, "Average cost markov control processes with weighted norms: value iteration," *Appl. Math*, vol. 23, pp. 219–237, 1995.
- [18] S. Meyn, "The policy iteration algorithm for average reward markov decision processes with general state space," *Automatic Control, IEEE Transactions on*, vol. 42, no. 12, pp. 1663–1680, 1997.