Computational Synthesis of Event-triggers for MMSE State Estimators with Communication Constraints

Lichun 'Lucinda' Li and Michael Lemmon

Abstract

Event-triggered transmission has been shown to minimize an estimator's mean square error discounted by the communication cost. This optimal trigger, however, is difficult to compute, which motivated the use of sub-optimal quadratic event triggers that are easier to synthesize. This paper introduces algorithms for computing a larger class of polynomial event-triggers for MMSE state estimators. These algorithms pose the synthesis problem as a semi-definite program (SDP) that is efficiently solved using computational tools such as SOSTOOLS. The paper derives upper bounds and lower bounds on the performance achieved by these polynomial event-triggers and simulation results show that these polynomial triggers always out-perform the quadratic triggers proposed in earlier works. Finally, the paper applies these methods to an 8 dimensional nonlinear three degree-of-freedom (DOF) helicopter. The simulation results show that our event-triggered estimator uses fewer communication resources than periodically triggered estimator while achieving similar performance levels. To our best knowledge, this is the first time this particular approach to event-trigger synthesis has been applied to systems with dimension greater than 2.

I. Introduction

Wireless networked control systems are invaluable in many civil and military applications, such as environmental monitoring, traffic control, smart grid, manufacturing, collecting information on the battlefield, and so on. In most cases, the sensors are battery driven and the wireless channel has very low bandwidth, which limits the transmission frequency. Meanwhile, to assure accurate monitoring of the system, one wants the sensors to transmit information as often as possible. This tradeoff between communication usage and the monitoring quality (performance) can be formulated as an optimization problem that seeks to minimize the average mean square estimation error discounted by the cost of communication with respect to the transmission rule.

This optimization problem was first studied in [1] where it was shown that the optimal transmission rule was event triggered transmission. Event triggered transmission is a transmission method in which sensors or controllers only transmit information when some event occurs. In particular, information is transmitted when a measure of

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

data 'novelty' exceeds a specified threshold. As early as 1999, it was claimed that event triggering could make more efficient use of communication resources than periodic transmission schemes [2]. This claim was further demonstrated experimentally by the fact that event triggering maintains comparable system performance while using fewer communication resources than periodic transmission [3]–[9]. These experimental results coincide with the theoretic result given in [1] which said that the optimal transmission rule that minimizes the average mean square estimation error discounted by the communication cost is event triggered transmission. This optimal event trigger, however, is difficult to compute, and the computational complexity for finding the optimal event trigger grows exponentially with respect to the state dimension.

Based on the sub-optimality bounds in [10], people began to consider suboptimal event triggers, and tried to use the difference between an upper bound on the suboptimal cost and a lower bound on the minimum cost to characterize how well the suboptimal event trigger performs. [11] provided a suboptimal event trigger, and guaranteed that the suboptimal cost is always less than 6 times of the minimum cost. This result, however, only holds for stable systems. Later, [12] and [13] gave suboptimal event triggers for unstable systems and the corresponding upper bounds on the suboptimal costs, but didn't provide any measure of how good these suboptimal event triggers were.

This paper proposes an algorithm to compute a polynomial suboptimal event trigger and an upper bound on the suboptimal cost, and an algorithm to compute a lower bound on the minimum cost. Both algorithms are based on semi-definite programs, and hence computationally effective. Both algorithms work for both stable and unstable systems. In our simulation, when we increase the order of the polynomial event trigger to 10, the ratio of the upper bounds on the suboptimal costs to the lower bounds on the minimum costs are less than 1.4 for both stable and unstable systems, while [11] only guarantees that this ratio is less than 6 for stable systems, and [12], [13] didn't provide any result on this ratio.

Later, we apply the polynomial event triggers to an 8 dimensional 3 degree-of-freedom (3DOF) helicopter. To our best knowledge, this is the first time the suboptimal event trigger has been applied to a system whose dimension is greater than 2. Our simulation results show that with polynomial event triggers, the 3DOF helicopter tracked the reference signal with small overshoot and no steady error. This event triggered helicopter used fewer communication resources than a periodically triggered helicopter with comparable performance, and tolerated the same amount of transmission delay as the periodically triggered helicopter while maintaining the system performance.

II. 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

$$x(k) = Ax(k-1) + w(k-1),$$
$$y(k) = Cx(k) + v(k),$$

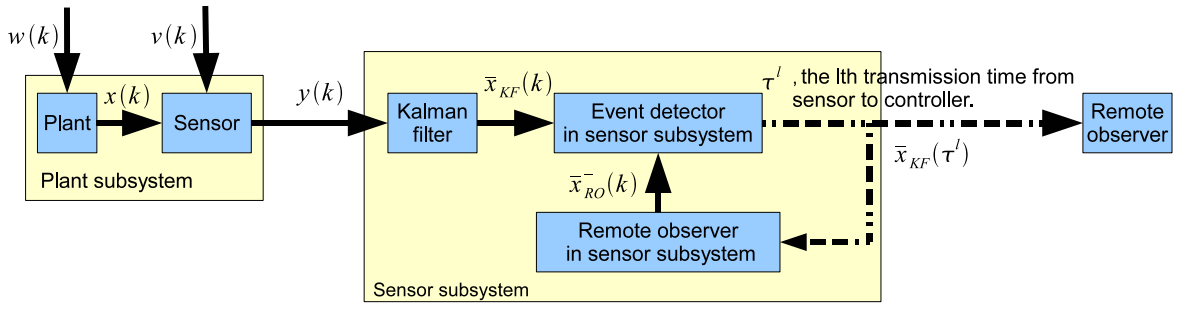

Fig. 1: Structure of the event triggered state estimation systems

for $k = 1, 2, \ldots, \infty$. Let $\mathbb{R}^n$ denote the n dimensional real space, and $\mathbb{N}_0$ denote the set of all non-negative integers. In the difference equation,

- $x : \mathbb{N}_0 \to \mathbb{R}^n$ is the system state with initial state $x(0)$ being a Gaussian random variable with 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 statistically independent.
- The pair (A, C) is observable.
- $y : \mathbb{N}_0 \to \mathbb{R}^p$ 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 $\overline{x}_{KF} : \mathbb{N}_0 \to \mathbb{R}^n$ that minimizes the weighted mean square estimation error (MSEE), i.e.

$$\overline{x}_{KF}(k) = \min_{\overline{x}_{KF}(k)} E\left[\|x(k) - \overline{x}_{KF}(k)\|_Z^2 \mid \{y(0), y(1), \cdots, y(k)\}\right]$$

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

$$\overline{x}_{KF}(k) = A\overline{x}_{KF}(k-1) + L\left[y(k) - CA\overline{x}_{KF}(k-1)\right], \tag{1}$$

where $L = AXC^T(CXC^T + V)^{-1}$, and X satisfies the discrete linear Riccati equation

$$AXA^T - X - AXC^T(CXC^T + V)^{-1}CXA^T + W = 0. \tag{2}$$

The steady state estimation error $e_{KF}(k) = x(k) - \overline{x}_{KF}(k)$ is a Gaussian random variable with zero mean and variance

$$E(e_{KF}e_{KF}^T) = Q = (I - LC)X.$$

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

The *event detector in the sensor subsystem* uses the *a priori gap*

$$e_{KF,RO}^{-}(k) = \overline{x}_{KF}(k) - \overline{x}_{RO}^{-}(k), \tag{3}$$

to decide whether or not to transmit $\overline{x}_{KF}(k)$ 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

$$\theta \gg \lambda > 0, \tag{4}$$

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

The *remote observer* generates the remote state estimate $\overline{x}_{RO}(k)$ to minimize the MSEE based on all of the remote observer's information up to step k. Let $l(k) = \max\left\{l : \tau^l \leq k\right\}$ 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

$$\mathbf{H}_{RO}(k) = \left\{\overline{x}_{KF}(\tau^1), \overline{x}_{KF}(\tau^2), \cdots, \overline{x}_{KF}(\tau^{l(k)}), a(0), a(1), \ldots, a(k)\right\},$$

for $k = 0, 1, \ldots, \infty$ with $\mathbf{H}_{RO}(-1) = \emptyset$. To minimize the MSEE, the *a posteriori remote state estimate* $\overline{x}_{RO}(k)$ satisfies (see section 3.2.4 in [14])

$$\overline{x}_{RO}(k) = E(x(k)|\mathbf{H}_{RO}(k)),$$

and the *a priori remote state estimate* $\overline{x}_{RO}^{-}(k)$ satisfy

$$\overline{x}_{RO}^{-}(k) = E(x(k)|\mathbf{H}_{RO}(k-1)).$$

It was shown in [15] that the remote state estimate takes the form of

$$\overline{x}_{RO}^{-}(k) = A\overline{x}_{RO}(k-1), \text{ with } \overline{x}^{-}(0) = \mu_0 \tag{5}$$

$$\overline{x}_{RO}(k) = a(k)\overline{x}_{KF}(k) + (1 - a(k))\overline{x}_{RO}^{-}(k). \tag{6}$$

Remark 2.1: If the controller is assumed to know the triggering event in the sensor subsystem (which is not the case in this paper), the history information of the remote observer should include the event triggering information in the sensor subsystem, and the minimum MSEE won't be realized by the form given by equation (5) and (6). An approximated minimum MSEE assuming that the controller knows the triggering event in sensor subsystem was presented in section 2 of [16].

Now, let us define the remote state estimation error $e_{RO}(k)$ as

$$e_{RO}(k) = x(k) - \overline{x}_{RO}(k).$$

The average cost in this event triggered state estimation problem is

$$J(\{a(k)\}_{k=0}^{\infty}) = \lim_{N\to\infty} \frac{1}{N} \sum_{k=0}^{N-1} E\left(c(e_{RO}(k), a(k))\right), \tag{7}$$

where the cost function

$$c(e_{RO}(k), a(k)) = \|e_{RO}(k)\|_Z^2 + a(k)\lambda, \tag{8}$$

and λ is the communication price for one transmission.

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

$$J^* = \min_{\{a(k)\}_{k=0}^{\infty}} J(\{a(k)\}_{k=0}^{\infty}). \tag{9}$$

III. The Optimal Event Trigger

This section first transforms the optimal problem described in (9) to a new optimal problem whose cost function relies on $e_{KF,RO}^{-}$ instead of e_{RO}, and then provide the optimal event trigger for the new optimal problem.

Let $e_{KF,RO}(k) = \overline{x}_{KF}(k) - \overline{x}_{RO}(k)$ be the *a 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 see Lemma 2 in [15] for the detailed proof.

Lemma 3.1: The filtered state error, $\overline{e}_{KF}(k)$ 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 3.1, the expected value of the cost function $E(c)$ satisfies

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

where

$$c_n(e_{KF,RO}^{-}(k), a(k)) = a(k)\lambda + (1 - a(k))\|e_{KF,RO}^{-}(k)\|_Z^2. \tag{10}$$

Thus, the average cost is rewritten as

$$J(\{a(k)\}_{k=0}^{\infty}) = trace(QZ) + J_n(\{a(k)\}_{k=0}^{\infty}), \tag{11}$$

where

$$J_n(\{a(k)\}_{k=0}^{\infty}) = \lim_{N\to\infty} \frac{1}{N} \sum_{k=0}^{N-1} E\left(c_n(e^-_{KF,RO}(k), a(k))\right), \tag{12}$$

and the minimum cost J^* satisfies

$$J^* = trace(QZ) + J_n^*, \tag{13}$$

where

$$J_n^* = \min_{\{a(k)\}_{k=0}^{\infty}} J_n(\{a(k)\}_{k=0}^{\infty}) \tag{14}$$

This is an average cost optimal problem in a discrete time Markov control process (MCP) [17], [18]. In this MCP, $e^-_{KF,RO}(k) \in \mathbb{R}^n$ is the state, $a(k) \in \{0,1\}$ is the action with feasible action set to be $\{0,1\}$ if $\|e^-_{KF,RO}(k)\|_Z^2 \le \theta$, $\{1\}$ otherwise, and c_n is the cost function. Since $e^-_{KF,RO}(k)$ is the state in the MCP, we are interested in the dynamic behavior of $e^-_{KF,RO}(k)$. From equation (1), (5) and (6), we have

$$\begin{aligned} e^-_{KF,RO}(k+1) &= Ae_{KF,RO}(k) + L\tilde{y}(k+1) \\ &= (1-a(k))Ae^-_{KF,RO}(k) + L\tilde{y}(k+1) \end{aligned} \tag{15}$$

where $\tilde{y}(k+1) = y(k+1) - CA\overline{x}_{KF}(k) = CAe_{KF}(k) + Cw(k) + v(k+1)$. It is easy to see that $\tilde{y}(k+1)$ is a zero mean Gaussian random variable with variance $Y = CAQA^TC^T + CWC^T + V$. Thus, the transition probability of $e^-_{KF,RO}(k+1)$ conditioned on $e^-_{KF,RO}(k)$ and $a(k)$ is

$$p\left(e^-_{KF,RO}(k+1)|e^-_{KF,RO}(k), a(k)\right) = \mathcal{N}\left((1-a(k))Ae^-_{KF,RO}(k), LYL^T\right), \tag{16}$$

where $\mathcal{N}(\mu, \Pi)$ is the Gaussian probability density function with mean and variance to be μ and Π, respectively. Let

$$E_h(\mu) = E\left(h(s)\right),$$

where s is a random variable whose probability density function satisfies $p(s) = \mathcal{N}(\mu, LYL^T)$. It is easy to show that

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

With the Markov control process characterized above, with the same technique used in Theorem 1 of [1], we have the following theorem about the optimal transmission rule and the optimal cost for the optimal problem described in (14).

Theorem 3.2: 1) There exist a unique constant ρ^* and a function h_1^*, such that for all $e^-_{KF,RO}(k) \in \mathbb{R}^n$

$$\rho^* + h_1^*(e^-_{KF,RO}(k)) = \min\left\{\|e^-_{KF,RO}(k)\|_Z^2 + E_{h_1^*}\left(Ae^-_{KF,RO}(k)\right), \lambda + E_{h_1^*}(0)\right\}, \tag{17}$$

2) The optimal cost $J^* = \rho^* + trace(QZ)$.

3) There exists a deterministic stationary optimal transition rule, i.e. *the optimal event trigger*, which is

$$a(k)=\begin{cases}1, & \text{if } \|e^-_{KF,RO}(k)\|_Z^2+E_{h_2^*}\left(Ae^-_{KF,RO}(k)\right)\geq\lambda+E_{h_2^*}(0) \text{ or } \|e_{KF,RO}\|_Z^2>\theta,\\ 0, & \text{otherwise.}\end{cases}$$

Remark 3.3: It is hard to find an analytic solution to equation (17). Although we can iteratively compute the solution of equation (17) by value iteration [19] or policy iteration [20], [21], the computational complexity increases exponentially with respect to the state dimension.

IV. A Suboptimal Event Trigger and An Upper Bound on The Suboptimal Cost

Because of the difficulty of computing the optimal event trigger, much of the prior work has only solved this problem for scalar systems. Higher dimensional systems can only been addressed using quadratic approximations, but even in these cases the manual computation of these quadratic approximations can be tedious and the bounds may be poor. This section provides a semi-definite program based algorithm to compute a polynomial suboptimal event trigger and an upper bound on the suboptimal cost.

The suboptimal event trigger and the upper bound on the corresponding suboptimal cost is computed based on Theorem 1 in [10]. Here, we state this lemma without proof. The proof can be found in Theorem 1 in [10] or Theorem 4.1 of [22].

Lemma 4.1: Suppose in a Markov control process, $s(k)\in\mathbf{S}$ is the state where $\mathbf{S}$ is the state space, $a(k)\in\mathbf{A}$ is the action where $\mathbf{A}$ is the action set, c is the cost function, and Q is the transition probability of $s(k+1)$ conditioned on $s(k)$ and $a(k)$. If there exist a constant $\overline{J}$, a function h_2 bounded from below, and a deterministic stationary rule f, such that for all $s\in\mathbf{S}$

$$\begin{aligned}\overline{J}+h_2(s(k))\geq&\min_{a\in\mathbf{A}}\left\{c(s(k),a(k))+\int_{\mathbf{S}}h_2(s(k+1))Q(s(k+1)|s(k),a(k))\right\}\\ =&c(s(k),f(s(k)))+\int_{\mathbf{S}}h_2(s(k+1))Q(s(k+1)|s(k),f(s(k))),\end{aligned}$$

then the average cost $J(\{a(k)\}_{k=0}^{\infty})=J(f)=\lim_{N\to\infty}\frac{1}{N}\sum_{k=0}^{N-1}E\left(c(s(k),a(k))\right)=\lim_{N\to\infty}\frac{1}{N}\sum_{k=0}^{N-1}E\left(c(s(k),f)\right)$ satisfies

$$J(f)\leq\overline{J}.$$

According to Lemma 4.1, we have the following theorem.

Theorem 4.2: If there exist a constant ρ_2 and a function h_2, such that for all $e^-_{KF,RO}(k)\in\mathbb{R}^n$,

$$\rho_2+h_2(e^-_{KF,RO}(k))\geq\min\left\{\|e^-_{KF,RO}(k)\|_Z^2+E_{h_2}\left(Ae^-_{KF,RO}(k)\right),\lambda+E_{h_2}(0)\right\},\tag{18}$$

and

$$a(k)=\begin{cases}1, & \text{if } \|e^-_{KF,RO}(k)\|_Z^2+E_{h_2}\left(Ae^-_{KF,RO}(k)\right)\geq\lambda+E_{h_2}(0) \text{ or } \|e^-_{KF,RO}(k)\|_Z^2>\theta;\\ 0, & \text{otherwise,}\end{cases}\tag{19}$$

then the average cost $J(\{a(k)\}_{k=0}^{\infty})$ defined in equation (7) satisfies

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

Proof: Consider the average cost J_n described in (12). The corresponding Markov process involves the state $e^-_{KF,RO}$, the action a, the cost function c_n, and the transition probability p described in (16). According to Lemma 4.1, if there exists a constant ρ_2, a function h_2, and a deterministic stationary rule f such that for all $e^-_{KF,RO}(k) \in \mathbb{R}^n$,

$$\rho_2 + h_2(e^-_{KF,RO}(k)) \geq \min_{a\in\{0,1\}} \left\{ c_n(e^-_{KF,RO}(k), a(k)) + E\left(h_2(e^-_{KF,RO}(k+1)|e^-_{KF,RO}(k), a(k)) \right) \right\} \tag{20}$$

$$= c_n(e^-_{KF,RO}(k), f) + E\left(h_2(e^-_{KF,RO}(k+1)|e^-_{KF,RO}(k), f) \right), \tag{21}$$

then $J_n(f) \leq \rho_2$, and from equation (11), we have $J(f) \leq \rho_2 + trace(QZ)$.

From the dynamic behavior of $e^-_{KF,RO}$ described in (15) and the new cost function c_n in (10), the right hand side of (20) is rewritten as

$$\begin{aligned} &\min\left\{ \|e^-_{KF,RO}(k)\|^2_Z + E_{h_2}\left(Ae^-_{KF,RO}(k) \right), \lambda + E_{h_2}(0) \right\} \\ =&c_n(e^-_{KF,RO}(k), f) + E\left(h_2(e^-_{KF,RO}(k+1)|e^-_{KF,RO}(k), f) \right). \end{aligned}$$

The equality holds if

$$a(k) = f(e^-_{KF,RO}(k)) = \begin{cases} 1, & \text{if } \|e^-_{KF,RO}(k)\|^2_Z + E_{h_2}\left(Ae^-_{KF,RO}(k) \right) \geq \lambda + E_{h_2}(0) \text{ or } \|e^-_{KF,RO}(k)\|^2_Z > \theta; \\ 0, & \text{otherwise.} \end{cases}$$

■

Next, we want to effectively search for ρ_2 and h_2 such that equation (18) is satisfied. The basic idea behind equation (18) is that $\rho_2 + h_2(e^-_{KF,RO}(k))$ is greater than $\|e^-_{KF,RO}(k)\|^2_Z + E_{h_2}\left(Ae^-_{KF,RO}(k) \right)$ if $e^-_{KF,RO}(k)$ is in a neighborhood of the origin, $\lambda + E_{h_2}(0)$ otherwise. Based on this idea, we have the following corollary.

Corollary 4.3: Given positive constants d, d_1, d_2 and positive integers ϕ, $\delta_1 \leq \phi$ and δ_2. If there exist a constant ρ_2, and a positive definite polynomial function h_2 such that the following inequalities hold for all $e^-_{KF,RO}(k) \in \mathbb{R}^n$,

$$\begin{aligned} \rho_2 + h_2(e^-_{KF,RO}(k)) \geq& \|e^-_{KF,RO}(k)\|^2_Z + E_{h_2}\left(Ae^-_{KF,RO}(k) \right) \\ &- \left(\|e^-_{KF,RO}(k)\|^2_Z + E_{h_2}\left(Ae^-_{KF,RO}(k) \right) \right) \frac{\|e^-_{KF,RO}(k)^{\phi}\|^2_Z - d}{d_1 + \|e^-_{KF,RO}(k)^{\phi-\delta_1}\|^2_Z}, \end{aligned} \tag{22}$$

$$\rho_2 + h_2(e^-_{KF,RO}(k)) \geq \lambda + E_{h_2}(0) + (\lambda + E_{h_2}(0)) \frac{\|e^-_{KF,RO}(k)^{\phi}\|^2_Z - d}{d_2 + \|e^-_{KF,RO}(k)^{\phi+\delta_2}\|^2_Z}. \tag{23}$$

where $s^{\phi} = [s_1^{\phi}\ s_2^{\phi}\ \dots\ s_n^{\phi}]^T$, and $a(k)$ satisfies equation (19), then

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

Proof: If $\|e^-_{KF,RO}(k)^\phi\|_Z^2 - d \leq 0$, from equation (22), we have

$$\begin{aligned}\rho_2 + h_2(e^-_{KF,RO}(k)) \geq & \|e^-_{KF,RO}(k)\|_Z^2 + E_{h_2}\left(Ae^-_{KF,RO}(k)\right) \\ & \geq \min\left\{\|e^-_{KF,RO}(k)\|_Z^2 + E_{h_2}\left(Ae^-_{KF,RO}(k)\right), \lambda + E_{h_2}(0)\right\}.\end{aligned}$$

If $\|e^-_{KF,RO}(k)^\phi\|_Z^2 - d > 0$, from equation (23), we have

$$\begin{aligned}\rho_2 + h_2(e^-_{KF,RO}(k)) \geq & \lambda + E_{h_2}(0) \\ & \geq \min\left\{\|e^-_{KF,RO}(k)\|_Z^2 + E_{h_2}\left(Ae^-_{KF,RO}(k)\right), \lambda + E_{h_2}(0)\right\}.\end{aligned}$$

Therefore, equation (22) and (23) guarantee equation (18) for all $e^-_{KF,RO}(k) \in \mathbb{R}^n$. According to Theorem 4.2, we have $J(\{a(k)\}_{k=0}^\infty) \leq \overline{J} = \rho_2 + trace(QZ)$. ■

Remark 4.4: Equation (22) and (23) can be transformed to a semi-definite program, and hence efficiently solved by the SDP solvers (SeDumi or SDPT3).

Equation (22) and (23) can be transformed to the following two inequalities (which will be shown to be polynomial).

$$\begin{aligned}&\left(\rho_2 + h_2(e^-_{KF,RO}(k)) - \|e^-_{KF,RO}(k)\|_Z^2 - E_{h_2}\left(Ae^-_{KF,RO}(k)\right)\right)\left(d_2 + \|e^-_{KF,RO}(k)^{\phi-1}\|_Z^2\right) \\ &\quad + \left(\|e^-_{KF,RO}(k)\|_Z^2 + E_{h_2}\left(Ae^-_{KF,RO}(k)\right)\right)\left(\|e^-_{KF,RO}(k)^\phi\|_Z^2 - d_1\right) \geq 0 \qquad (24) \\ &\left(\rho_2 + h_2(e^-_{KF,RO}(k)) - \lambda - E_{h_2}(0)\right)\left(d_3 + \|e^-_{KF,RO}(k)^{\phi+1}\|_Z^2\right) \\ &\quad + (\lambda + E_{h_2}(0))\left(d_1 - \|e^-_{KF,RO}(k)^\phi\|_Z^2\right) \geq 0, \qquad (25)\end{aligned}$$

Since h_2 is a polynomial, according to the Isserlis' theorem [1] [23], $E_{h_2}\left(Ae^-_{KF,RO}(k)\right)$ is also a polynomial, and the corresponding coefficients are linear combinations of the coefficients of h_2. So, the two inequalities (24) and (25) are polynomial inequalities whose coefficients are linear combinations of the coefficients of h_2.

With the sum of squares (SOS) decomposition of multi-variable polynomials, the two polynomial inequalities (24) and (25) can be transformed to a semi-definite program. Let $h_2(\mathbf{s}) = \sum_{i=1}^N a_i \mathbf{s}^{\alpha_i}$, where $a_i \in \mathbb{R}$, $\mathbf{s} \in \mathbb{R}^n$, $\alpha_i \in \mathbb{N}^n$ and $\mathbf{s}^{\alpha_i} = \mathbf{s}_1^{\alpha_{i,1}} \cdot \ldots \cdot \mathbf{s}_n^{\alpha_{i,n}}$. Both (24) and (25) can be expressed as

$$g(\mathbf{s}) = \sum_{i=1}^M c_i \mathbf{s}^{\beta_i} \geq 0,$$

where c_i is a linear combination of $\{a_i\}_{i=1}^N$, and $\beta_i \in \mathbb{N}^n$. $g(\mathbf{s}) \geq 0$ is guaranteed, if there exist polynomials $f_1(\mathbf{s}), \ldots, f_m(\mathbf{s})$ such that

$$g(\mathbf{s}) = \sum_{i=1}^m f_i^2(\mathbf{s}).$$

[1][Isserlis' theorem:] If $(x_1, \ldots, x_{2n})$ is a zero mean multivariate normal random vector, then

$$\begin{aligned}E(x_1 x_2 \cdots, x_{2n}) =& \sum \prod E(x_i x_j), \\ E(x_1 x_2 \cdots, x_{2n-1}) =& 0,\end{aligned}$$

where the notation $\sum \prod$ means summing over all distinct ways of partitioning $x_1, \ldots, x_{2n}$ into pairs.

This SOS condition is equivalent to the existence of a positive semi-definite matrix G such that

$$g(\mathbf{s}) = \overline{\mathbf{s}}^T G \overline{\mathbf{s}},$$

where $\overline{\mathbf{s}}$ is some properly chosen vector of monomials with $\overline{\mathbf{s}}_i = \mathbf{s}^{\gamma_i}$ where $\gamma_i \in \mathbb{N}^n$ [24]–[26]. At this point, the SOS decomposition of the polynomial $g(\mathbf{s})$ is the same as finding a symmetric matrix G such that

$$\sum_{\gamma_i+\gamma_j=\beta_k} G_{i,j} = c_k, \text{for all } k = 1, \ldots, M,$$

$$G \geq 0.$$

This is a semi-definite programm, and can be efficiently solved by SeDumi or SDPT3. SOSTOOLS provides a convenient tool which automatically converts polynomial inequalities to SDP, calls the SDP solver (SeDumi or SDPT3), and converts the SDP solution back to the solution of the original polynomial inequalities. For detail about SOSTOOLS, please check [27].

Remark 4.5: There are 6 parameters to be chosen in equation (22) and (23). According to our experience, Larger ϕ, δ_1 and δ_2 provide smaller ρ_2, but consumes more computation effort. So, ϕ, δ_1 and δ_2 can be chosen to be large enough such that the computation time is not too long. d_1 and d_2 can be chosen such that $d_1 = 500d$ and $d_2 = d$. At this point, the only undetermined parameter is d, and we will use DIRECT optimization algorithm [28] to searches for the d which provides the smallest ρ_2.

The DIRECT optimization algorithm is an algorithm based on Lipschitzian optimization without knowing the Lipschitz constant [29]. The DIRECT optimization algorithm can solve global optimization problems with bound constraints and a real valued objective function without the knowledge of the objective function gradient. The accompanying MATLAB program, directc.m, can be found in [28].

Next, we provide an algorithm to compute the suboptimal event trigger and the upper bound on the suboptimal cost.

*Algorithm 4.6 (**Compute the suboptimal event trigger and the upper bound on the suboptimal cost**):*

1) Initialization
   a) Provide system parameters: A, C, W, V, L, Q, Z, and λ.
   b) Provide parameters in equation (22) and (23): ϕ, δ_1, and δ_2.
   c) Initialize an SOS program as *prog* using toolbox 'SOSTOOLS'.
      - Declare scalar decision variables: ρ_2, s_1, …, s_n ($s = [s_1, \ \ldots, \ s_n]^T$ denotes $e^-_{KF,RO}(k)$ in equation (22) and (23).
      - Declare a polynomial variable: h_2.
      - Compute $E_{h_2}(0)$ and $E_{h_2}(As)$ using the Isserlis' theorem.
2) Use DIRECT optimization algorithm to search for d^* which provides the smallest ρ_2.
   a) Fix the initial searching interval to be $[0, 10\lambda]$. Let $bound = 10\lambda$.

b) Call function *solve_poly_inequality* to compute ρ_2 and h_2 at the point $d = 10\lambda$. If there is no feasible solution, then use the bisection method to find the largest $d \in [0, 10\lambda]$ which provides a feasible solution. Let $bound$ be this largest d providing a feasible solution.

c) Search for d^* using *direct* function (given in [28]) with the function handle to be *solve_poly_inequality* and the searching interval to be $[0, bound]$.

3) Call function *solve_poly_inequality* to compute ρ_2 and h_2 at the point d^*. Compute the suboptimal event trigger: $s^T Z s + E_{h_2}(As) - \lambda - E_{h_2}(0) > 0$, and the upper bound on the suboptimal cost: $\rho_2 + trace(QZ)$.

4) Return.

solve_poly_inequality function:

$[\bar{\rho}_2, \bar{h}_2]$ = function *solve_poly_inequality*$(d, \lambda, \phi, \delta_1, \delta_2, prog, \rho_2, s, h_2, E_{h_2}(0), E_{h_2}(As))$

1) Initialize parameters in equation (22) and (23): $d_1 = 500d$ and $d_2 = d$.

2) Solve the following problem using the toolbox 'SOSTOOLS'.

$$\min \rho_2 \tag{26}$$

subject to: (24) and (25).

3) Get solutions from the SOS program *prog*. If problem (26) is feasible, then assign the value of ρ_2 and h_2 to $\bar{\rho}_2$ and $\bar{h}_2$, respectively. Otherwise, Let $\bar{\rho}_2 = \lambda$ and $\bar{h}_2 = 0$.

4) Return $[\bar{\rho}_2, \bar{h}_2]$.

At this point, we can effectively compute the suboptimal event trigger and the upper bound on the suboptimal cost. A natural question is how good this suboptimal event trigger is. This can be characterized by the difference between the upper bound on the suboptimal cost and the lower bound on the optimal cost. The next section will provide an algorithm to compute a lower bound on the optimal cost.

V. Compute a lower bound on the minimum cost

This section first states an existing result about a lower bound on the minimum cost for a general Markov control process, then provides the main theorem about the lower bound on the minimum cost for the state estimation problem described in Section II, and finally gives an algorithm to compute the lower bound on the minimum cost.

Lemma 3 of [10] provides a lower bound on the minimum cost for a general Markov control process, here we state this lemma without proof.

Lemma 5.1: Suppose in a Markov control process, $s(k) \in \mathbf{S}$ is the state where $\mathbf{S}$ is the state space, $a(k) \in \mathbf{A}$ is a static state-feedback action where $\mathbf{A}$ is the action set, c is the cost function, and Q is the transition probability of $s(k+1)$ conditioned on $s(k)$ and $a(k)$. If there exists a constant $\underline{J}$ and a function h_1, such that for all $s(k) \in \mathbf{S}$

$$\underline{J} + h_1(s(k)) \leq \min_{a(k)\in\mathbf{A}} \left\{ c(s(k), a(k)) + \int_{\mathbf{S}} h_1(s(k+1)) Q(s(k+1)|s(k), a(k)) ds(k+1) \right\},$$

then the minimum cost $J^* = \min_{\{a(k)\}_{k=0}^{\infty}} \lim_{N\to\infty} \frac{1}{N} \sum_{k=0}^{N-1} E\left(c(s(k), a(k))\right)$ satisfies

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

Based on Lemma 5.1, we have the following theorem.

Theorem 5.2: If there exists a constant ρ_1 and a polynomial function h_1, such that for all $e^-_{KF,RO}(k) \in \mathbb{R}^n$

$$\rho_1 + h_1(e^-_{KF,RO}(k)) \leq \|e^-_{KF,RO}(k)\|^2_Z + E_{h_1}\left(Ae^-_{KF,RO}(k)\right), \tag{27}$$

$$\rho_1 + h_1(e^-_{KF,RO}(k)) \leq \lambda + E_{h_1}(0), \tag{28}$$

then $J^* \geq \underline{J}^* = \rho_1 + trace(QZ)$.

Proof: According to Lemma 5.1, for the optimal problem described in equation (14), if there exists a constant ρ_1 and a function h_1, such that for all $e^-_{KF,RO}(k) \in \mathbb{R}^n$

$$\rho_1 + h_1(e^-_{KF,RO}(k)) \leq \min_{a(k)\in\{0,1\}} \left\{c_n(e^-_{KF,RO}(k), a(k)) + E(h_1(e^-_{KF,RO}(k+1))|e^-_{KF,RO}(k), a(k))\right\},$$

then $J^*_n \geq \rho_1$. From equation (13), we have $J^* \geq \rho_1 + trace(QZ)$.

From the dynamic behavior of $e^-_{KF,RO}$ described in (15) and the new cost function c_n in (10), the right hand side of (20) is rewritten as

$$\rho_1 + h_1(e^-_{KF,RO}(k)) \leq \min\left\{\|e^-_{KF,RO}(k)\|^2_Z + E_{h_2}\left(Ae^-_{KF,RO}(k)\right), \lambda + E_{h_2}(0)\right\}$$

$$\Leftrightarrow \begin{cases} \rho_1 + h_1(e^-_{KF,RO}(k)) \leq & \|e^-_{KF,RO}(k)\|^2_Z + E_{h_1}\left(Ae^-_{KF,RO}(k)\right), \\ \rho_1 + h_1(e^-_{KF,RO}(k)) \leq & \lambda + E_{h_1}(0). \end{cases}$$

So, we have Theorem 5.2. ■

Remark 5.3: With the same argument in Remark 4.4, we can see that equation (27) and (28) can be transformed to a semi-definite program, and hence efficiently solved. Next, we will give an algorithm to search for ρ_1 and h_1.

*Algorithm 5.4 (**Compute a lower bound on the minimum cost**):*

1) Initialization
   a) Provide system parameters: A, C, W, V, L, Q, Z, and λ.
   b) Initialize an SOS program as *prog* using toolbox 'SOSTOOLS'.
      - Declare scalar decision variables: ρ_1, s_1, …, s_n ($s = [s_1, \ldots, s_n]^T$ denotes $e^-_{KF,RO}(k)$ in equation (27) and (28).
      - Declare a polynomial variable: h_1.
      - Compute $E_{h_1}(0)$ and $E_{h_1}(As)$ using the Isserlis' theorem.
2) Solve the following problem using the toolbox 'SOSTOOLS'.

$$\min -\rho_1$$

subject to: (27) and (28)

3) Get solutions of ρ_1 and h_1 from the SOS program *prog*, and compute the lower bound $\underline{J}$ on the minimum cost as $\underline{J}^* = \rho_1 + traceQZ$.
4) Return.

computation method	$\overline{degree}_1$	elapsed time	lower bound $\underline{J}^*$
Algorithm 5.4	3	0.14s	0
	5	0.094s	1.35
	8	0.375s	1.35
[11]	2	$\times$	0.46
[13]	2	$\times$	$\times$

TABLE I: The lower bounds on the optimal cost of the stable system. $\times$ indicates no feasible solution or not available in the prior work.

VI. Mathematical Examples

This section uses Algorithm 4.6 and 5.4 to search for a suboptimal event trigger, an upper bound on the corresponding suboptimal cost, and a lower bound on the optimal cost for two linear time invariant (LTI) systems. The first LTI system is a marginally stable system, and the second LTI system is an unstable system. We would like to compare the upper bound on the suboptimal cost and the lower bound on the optimal cost computed from our algorithms with those computed based on the prior work [11]–[13]. The two examples were run on a Microsoft Windows XP system with 2.99 GHz CPU and 3.37GB of RAM. All of the source files are available at www.nd.edu/~lli3/projects.html.

A. Stable system

Consider a marginally stable system as below

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

$$y(k) = x(k),$$

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 [11], and we would like to compare the results in this paper with the results in [11] and [13].

Algorithm 5.4 was first used to compute a lower bound $\underline{J}$ on the minimum cost. In this algorithm, h_1 was set to be a polynomial which contained all possible monomials whose degrees were no greater than $\overline{degree}_1$. We varied $\overline{degree}_1$ to see how the lower bound $\underline{J}$ on the minimum cost changes with respect to the degree of h_1. The results are shown in Table I.

Algorithm 4.6 was then used to compute the suboptimal event trigger and the upper bound $\overline{J}$ on the suboptimal cost with $\phi = 6$ and $\delta_1 = \delta_2 = 4$. In this algorithm, h_2 was set to be a polynomial which contained all possible monomials whose degrees were even and no greater than $\overline{degree}_2$. We varied $\overline{degree}_2$ to see how the upper bound $\overline{J}$

computation method	$\overline{degree}_2$	elapsed time for operating *solve_poly_inequality* function	*solve_poly_inequality* function evaluation times	upper bound $\overline{J}$	difference between $\overline{J}$ and the largest $\underline{J}^*$	Actual cost
Algorithm 4.6	2	1.2s	13	6.75	5.40	≈ 5.1
	4	1.7s	13	6.07	4.72	≈ 1.86
	6	2.8s	13	2.14	0.79	≈ 1.70
	8	4.4s	13	1.85	0.5	≈ 1.56
	10	7.3s	15	1.79	0.44	≈ 1.56
[11]	2	$\times$	1	2.74	2.28	≈ 1.56
[13]	2	$\times$	1	3.78	$\times$	≈ 1.4

TABLE II: The upper bounds on the suboptimal cost of the stable system. $\times$ indicates no feasible solution or not available in the prior work.

on the suboptimal cost changes with respect to the degree of h_2. Moreover, to characterize how good the suboptimal event trigger is, we computed the difference between the upper bound $\overline{J}$ on the suboptimal cost and the largest lower bound $\underline{J}$ computed from Algorithm 5.4, and compared this difference with the prior work [11] and [13]. The results are given in Table II. From the fourth column of Table I, we see that as the degree of h_1 increases, the lower bound $\underline{J}^*$ on the minimum cost increases. When the degree of h_1 is 5, the lower bound $\underline{J}^*$ on the minimum cost increases to 1.35, which is about 3 times of the lower bound on the minimum cost given by [11].

From the fourth column of Table II, we see that as the degree of h_2 increases, the upper bound $\overline{J}$ on the suboptimal cost decreases. As the degree of h_2 increases to 10, the upper bound on the suboptimal cost decreases to 1.79, which is the smallest among all the upper bounds on suboptimal costs provided by this paper, [11] and [13]. When the degrees of h_1 and h_2 are 5 and 10, respectively, the difference between the upper bound on the suboptimal cost and the lower bound on the optimal cost is only 0.44 , which is about $1/5$ of the difference provided by [11].

After obtaining a suboptimal event trigger, we applied the event trigger to the marginal stable system, ran the system for 3000 steps, and computed the actual average cost J. The results are given in the last column of Table II. We find that as the degree of h_2 increases from 4 to 10, the actual cost only decreases by the amount of 0.3, but the computational effort to evaluate the suboptimal event trigger increases a lot. So, when applying a polynomial event trigger to a control system, we should use the polynomial event trigger with lowest degree while providing acceptable upper bound on the suboptimal cost.

computation method	$\overline{degree}_1$	elapsed time	lower bound $\underline{J}^*$
Algorithm 5.4	3	0.17s	1.82
	5	0.39s	2.7
	7	0.56s	2.72
	9	$\times$	$\times$
	13	0.78s	3.23
[13]	2	$\times$	$\times$
[12]	2	$\times$	$\times$

TABLE III: The lower bounds on the optimal cost of the unstable system. $\times$ indicates no feasible solution or not available in the prior work.

B. Unstable system

Next, we consider an unstable system.

$$x(k+1) = \begin{bmatrix} 0.95 & 1 \\ 0 & 1.01 \end{bmatrix} x(k) + w(k)$$

$$y(k) = \begin{bmatrix} 0.1 & 1 \end{bmatrix} x(k) + v,$$

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 $\lambda = 5$.

With the same steps as we did for the stable system, we computed the lower bound $\underline{J}^*$ on the minimum cost while varying the degree of h_1 (see Table III), the upper bound $\overline{J}$ on the suboptimal cost while varying the degree of h_2 (see Table IV), and the difference between $\overline{J}$ and the largest $\underline{J}^*$ (see Table IV). These results were compared with the results given by [12] and [13].

From Table III, we see that among all the existing work, only this paper provides a way to compute the lower bound on the minimum cost for unstable systems, and this lower bound $\underline{J}^*$ increases as the degree of h_1 increases. When the degree of h_1 increases to 13, the lower bound on the minimum cost increases to 3.23.

From Table IV, we see that the upper bound $\overline{J}$ on the suboptimal cost decreases as we increase the degree of h_2. When the degree of h_2 increases to 10, $\overline{J}$ decreases to 3.79 which is about 0.7 and 0.5 times of the $\overline{J}$ provided by [13] and [12], respectively. When the degrees of h_1 and h_2 are 13 and 10, respectively, the difference between the upper bound $\overline{J}$ on the suboptimal cost and the lower bound $\underline{J}^*$ on the minimum cost is 0.56 which is about 0.17 of $\underline{J}^*$.

After obtaining a suboptimal event trigger, we applied the event trigger to the unstable system, ran the system for 3000 steps, and computed the actual average cost J which is given in the last column of Table IV. We find that as the degree of h_2 increases from 4 to 8, the actual cost remains almost the same. When the degree of h_2

computation method	$\overline{degree_2}$	elapsed time for operating *solve_poly_inequality* function	*solve_poly_inequality* function evaluation times	upper bound $\overline{J}$	difference between $\overline{J}$ and the largest $\underline{J}^*$	actual cost
Algorithm 4.6	2	1.2s	15	6.83	3.60	≈ 3.73
	4	3.4s	13	6.70	3.47	≈ 3.30
	6	5.5s	13	4.40	1.17	≈ 3.30
	8	7.1s	11	4.34	1.11	≈ 3.29
	10	8.5s	15	3.79	0.56	≈ 3.47
[13]	2	$\times$	1	5.57	$\times$	≈ 3.24
[12]	2	$\times$	1	7.69	$\times$	≈ 5.4

TABLE IV: The upper bounds on the suboptimal cost of the unstable system. $\times$ indicates not available in the prior work.

increases from 8 to 10, the actual cost even increases by the amount of 0.18. So, when applying a polynomial event trigger to this control system, the polynomial event trigger should be chosen such that it has the lowest degree with acceptable upper bound on the suboptimal cost.

VII. Application in a $QUANSER^{©}$ 3DOF Helicopter

This section applies the suboptimal event trigger computed from Algorithm 4.6 to a 8 dimensional nonlinear 3DOF helicopter. Subsection VII-A introduces the nonlinear model of the 3DOF helicopter, linearization of this nonlinear model and the controllers we will use for the 3DOF helicopter. Subsection VII-B explains how we design the suboptimal event triggers for this 3DOF helicopter. The experiment results are given in subsection VII-C.

A. $QUANSER^{©}$ 3DOF Helicopter

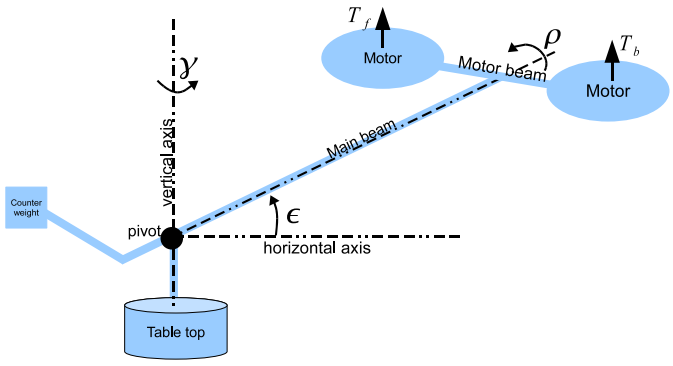

Fig. 2: Schematic of the 3DOF helicopter.

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 V: 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 thrusts generated by the DC motors. Our objective is to control the 3DOF helicopter to follow a commanded elevation ϵ_r and a commanded travel rate $\dot{\gamma}_r$.

The system dynamic is described by the following equations [30].

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

and the elevation, pitch and travel rate can be directly measured with the measurement noises to be white zero mean Gaussian, and the variances of the measurement noises are 1.857×10^{-6}, 1.857×10^{-6}, and 1.857×10^{-6}, respectively.

In this model, $\epsilon_m = \epsilon - \epsilon_0$, m is the gross counter weight at the tail, M is the gross weight at the head, $M_{bf} = m_b + m_f$ is the sum mass of the two motors, l_w is the length from the pivot to the tail while l_a is the length from the pivot to the head, d is some adjusted length with respect to the elevation, g is the gravity acceleration, $T_{col} = T_f + T_b$ and $T_{cyc} = T_b - T_f$ are the collective and cyclic thrusts, $c_\epsilon\dot{\epsilon}$, $c_\rho\dot{\rho}$, $c_{\gamma\rho}\dot{\gamma}$, $c_\gamma\dot{\gamma}$ are the drags generated by air due to the change of elevation, pitch and travel, and J_ϵ, J_ρ and J_γ are the inertia moments for elevation, pitch and travel respectively. The parameter values are given in Table V.

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 linearized as

$$J_\epsilon \ddot{\epsilon}_m = -\sqrt{((ml_w - Ml_a)g)^2 + ((m+M)gd)^2}\epsilon_m - c_\epsilon\dot{\epsilon}_m + l_a u_{\epsilon_m} \tag{29}$$

$$J_\rho \ddot{\rho} = -M_{bf} g d\rho - c_\rho\dot{\rho} - c_{\gamma\rho}\dot{\gamma} + l_h u_\rho \tag{30}$$

$$J_\gamma \ddot{\gamma} = -c_\gamma\dot{\gamma} - l_a u_\gamma, \tag{31}$$

where u_{ϵ_m}, u_ρ and u_γ are the control inputs for elevation, pitch and travel subsystems satisfying

$$u_{\epsilon_m} = T_{col}\cos(\rho), \tag{32}$$

$$u_\rho = T_{cyc}, \tag{33}$$

$$u_\gamma = T_{col}\rho\cos(\epsilon). \tag{34}$$

The control laws of these control inputs are

$$u_{\epsilon_m} = [7\ 44\ 68][\int_0^t \epsilon_m(s) - \epsilon_r(s)ds\ \ \epsilon_m(t) - \epsilon_r(t)\ \ \dot{\epsilon}_m(t)]^T, \tag{35}$$

$$u_\rho = [3.5\ 30\ 20][\int_0^t \rho(s) - \rho_r(s)ds\ \ \rho(t) - \rho_r(t)\ \ \dot{\rho}(t)]^T, \tag{36}$$

$$u_\gamma = [25\ 3][\gamma(t) - \gamma_r(t)\ \ \dot{\gamma}(t) - \dot{\gamma}_r(t)]^T, \tag{37}$$

where ρ_r is the reference pitch signal which will be explained later.

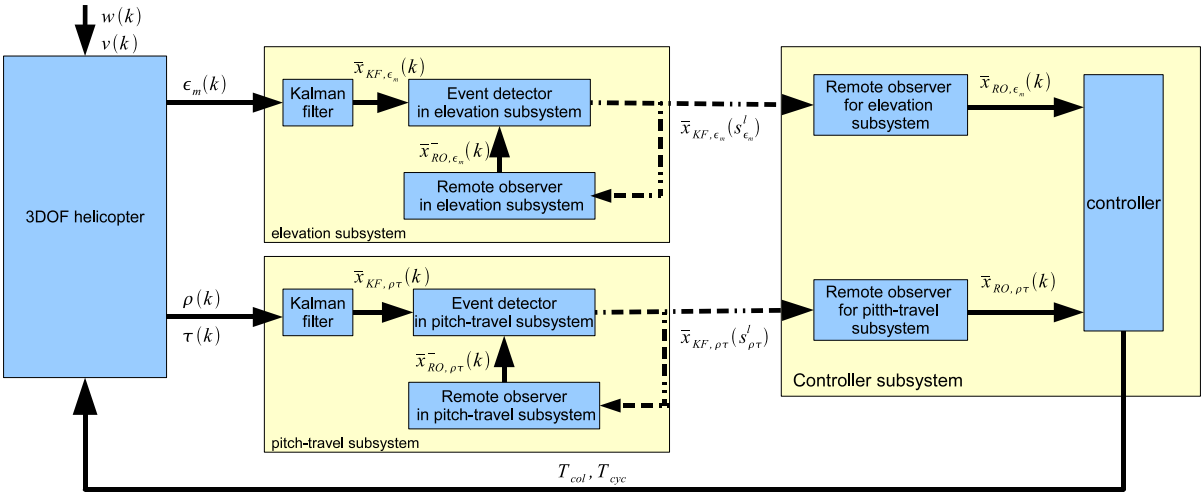

Fig. 3: Structure of event triggered 3DOF helicopter

To compute the collective and cyclic thrusts T_{col} and T_{cyc}, we first compute the control inputs u_{ϵ_m}, u_ρ and u_γ according to equation (35), (36) and (37) respectively, and then compute T_{col}, T_{cyc} and ρ_r based on the following equations which are derived from equation (32), (33 and (34).

$$\begin{aligned} T_{col} &= u_{\epsilon_m} / \cos(\rho), \\ T_{cyc} &= u_\rho, \\ \rho_r &= u_\gamma / (T_{col} \cos(\epsilon)). \end{aligned}$$

B. Design of the event triggered 3DOF helicopter.

From equation (29), (30) and (31), we can see that the helicopter system is decomposed into 2 decoupled subsystems: elevation subsystem and pitch-travel subsystems. For each subsystem, we will design a Kalman filter, a remote observer and a suboptimal event trigger. The structure of the whole closed loop system can be found in Figure 3.

1) Design of the elevation subsystem: The elevation subsystem is discretized with period $0.005s$, and the discrete model of the elevation subsystem is

$$x_{\epsilon_m}(k+1) = \begin{bmatrix} 1 & 0.005 & 1.25e-5 \\ 0 & 1 & 0.004999 \\ 0 & -0.001956 & 0.9997 \end{bmatrix} x_{\epsilon_m}(k) + \begin{bmatrix} 3.988e-9 \\ 2.393e-6 \\ 0.000957 \end{bmatrix} u_{\epsilon_m} + w_{\epsilon_m}$$

$$y_{\epsilon_m}(k) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} x_{\epsilon_m}(k) + v_{\epsilon_m},$$

subsystem	elapsed time for operating *solve_poly_inequality* function	*solve_poly_inequality* function evaluation times	upper bound $\overline{J}$	lower bound on minimum cost $\underline{J}^*$	$\frac{\overline{J}}{\underline{J}^*}$
elevation	$\approx$ 40s	21	0.20	0.076	2.65
pitch-travel	$\approx$ 630	21	0.67	0.16	4.18

TABLE VI: Upper bound on the suboptimal cost and lower bound on the minimum cost

where $x_{\epsilon_m}(k) = [\int_0^{0.005k} \epsilon_m(s)ds\ \epsilon_m(0.005k)\ \dot{\epsilon}_m(0.005k)]^T$, the variances of w_{ϵ_m} and v_{ϵ_m} are $diag([8e-3\ 1e-12\ 3.1e-4])$ and $diag([01.86e-6])$, respectively.

By solving the discrete linear Riccati equation (2), we have the Kalman filter gain as

$$L_{\epsilon_m} = \begin{bmatrix} 1 & 0.0016 \\ 0 & 0.3563 \\ 0 & 10.77 \end{bmatrix}.$$

Algorithm 4.6 was, then, used to compute a suboptimal event trigger. Let the weight matrix $Z = diag([1\ 6\ 1])$, $\lambda = 1$, $\phi = 2$, $\delta_1 = 1$, $\delta_2 = 1$. h_2 was chosen to be a polynomial which contains all possible monomials whose degree is even and no greater than 10.

A lower bound on the minimum cost was computed using Algorithm 5.4. In this Algorithm, h_1 was chosen to be a polynomial which contains all possible monomials whose degree is no greater than 10.

The related results about the upper bound on the suboptimal cost and lower bound on the minimum cost are given in Table VI. From the last column of this table, we find that the upper bound on the suboptimal cost is 2.65 times of the lower bound on the optimal cost, which is considered to be acceptable.

2) Design of the pitch-travel subsystem: With the period to be $0.005s$, the discrete model of the pitch-travel subsystem is given below.

$$x_{\rho\tau}(k+1) = \begin{bmatrix} 1 & 0.005 & 1.249e-5 & 0 & 6.247e-11 \\ 0 & 0.9999 & 0.004996 & 0 & 3.748e-8 \\ 0 & -0.05679 & 0.9984 & 0 & 1.499e-005 \\ 0 & 0 & 0 & 1 & 0.004999 \\ 0 & 0 & 0 & 0 & 0.9997 \end{bmatrix} x_{\rho\tau}(k) + \begin{bmatrix} 3.686e-7 & -1.308e-14 \\ 0.0002211 & -1.046e-11 \\ 0.08843 & -6.277e-9 \\ 0 & -2.094e-6 \\ 0 & -0.0008374 \end{bmatrix} \begin{bmatrix} u_\rho \\ u_\tau \end{bmatrix}$$
$$+ w_{\rho\tau}$$
$$y_{\rho\tau}(k) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} x_{\rho\tau}(k) + v_{\rho\tau},$$

where $x_{\rho\tau}(k) = [\int_{s=0}^{0.005k} \rho(s)ds\ \rho(0.005k)\ \dot{\rho}(0.005k)\ \tau(0.005k)\ \dot{\tau}(0.005k)]^T$. The variances of $w_{\rho\tau}$ and $v_{\rho\tau}$ are $diag([1e-3\ 1e-12\ 2e-4\ 1e-12\ 5.4e-4])$ and $diag([0\ 1.86e-6\ 1.86e-6])$, respectively.

sampling method	delay: s	elevation	pitch-travel	total
polynomial event trigger	0.005	1	53	54
periodic triggering		1	300	301
polynomial event trigger	0.01	1	48	49
periodic triggering		1	300	301
polynomial event trigger	0.02	1	2422	2423
periodic triggering		1	300	301

TABLE VII: Transmission times using event triggering and periodic triggering with different delays.

The Kalman gain can be computed by solving the discrete linear Riccati equation (2). For the pitch-travel subsystem, the Kalman gain is

$$L_{\epsilon_m} = \begin{bmatrix} 1 & 0.0015 & 0 \\ 0 & 0.3177 & 0 \\ 0 & 8.68 & 0 \\ 0 & 0 & 0.4083 \\ 0 & 0 & 13.837 \end{bmatrix}.$$

We, then, used Algorithm 4.6 to compute a suboptimal event trigger. Let the weight matrix $Z = diag([1\ 6\ 1\ 1\ 6])$, $\lambda = 1$, $\phi = 2$, $\delta_1 = 1$, $\delta_2 = 1$. Here, we chose h_2 such that there was no cross terms between the pitch state and the travel state, and all possible monomials whose degree was even and no greater than 4 were included.

A lower bound on the minimum cost was also computed using Algorithm 5.4. In this algorithm, h_1 was chosen such that it contained all polynomials, except cross terms between the pitch state and the travel state, whose degree is no greater than 8. The related results about the upper bound on the suboptimal cost and the lower bound on the minimum cost is shown in the second row of Table VI. From the last column, we see that the upper bound on the suboptimal cost is 4.18 times of the lower bound on the minimum cost, which is considered to be acceptable.

C. Experimental results for the event triggered 3DOF helicopter and periodically triggered 3DOF helicopter

We first ran the event triggered 3DOF helicopter system for 90 seconds, and then ran a periodically triggered 3DOF helicopter system for 90 seconds. When we ran the periodically triggered 3DOF helicopter system, we adjusted the periods of the elevation subsystem and pitch-travel subsystem until the performance was similar to the performance of the event triggered 3DOF helicopter.

1) Transmission times and performance with $0.005s$ delay: The transmission times for both event triggered and periodically triggered helicopter are shown in Table VII. In this experiment, the transmission delay is set to be $0.005s$, from the last column, we can see that the total transmission times of event triggered helicopter is less than 0.2 times of the transmission times of the periodically triggered helicopter.

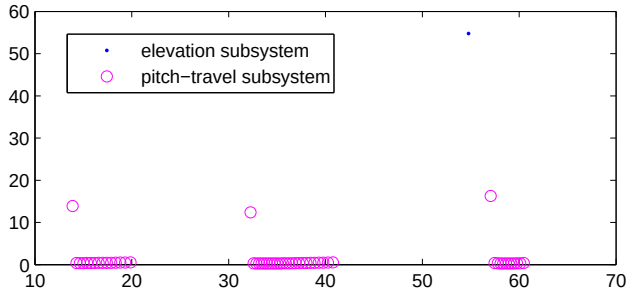

Fig. 4: inter-sampling intervals of event triggered 3DOF helicopter.

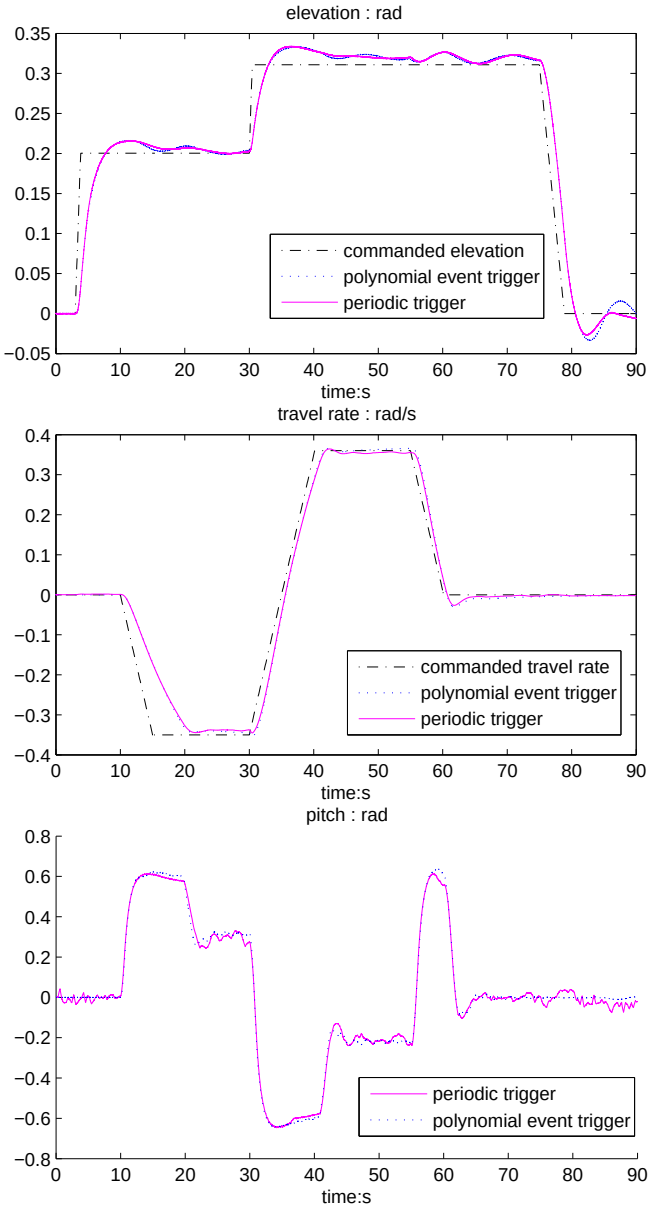

Fig. 5: The elevation, travel rate and pitch performance of the event triggered 3DOF helicopter and periodically triggered 3DOF helicopter.

Figure 4 shows the inter-sampling intervals of even triggered helicopter with x-axis indicating time and y-axis indicating inter-sampling intervals measured by second. Let us first look at the inter-sampling intervals (circles) of pitch-travel subsystem. The most frequent transmissions occurred during the intervals $[15s, 20s]$, $[32s, 42s]$ and $[55s, 60s]$. Compared with the middle plot of Figure 5, we see that these intervals are those when the travel subsystem was in transient processes. We, then, look at the inter-sampling intervals (dot) of elevation subsystem, and find that there was only one transmission in the elevation subsystem. Notice that the elevation subsystem is stable and only coupled with pitch subsystem (see equation (29)). So, if the remote state estimate of pitch is accurate enough, the remote state estimate of elevation will be accurate enough and there will be very few transmissions in the elevation subsystem.

The system performances are shown in Figure 5. The top plot is the system performance of elevation subsystem, with x-axis indicating time and y-axis indication elevation measured by rad. We can see that event triggered helicopter (dotted line) and periodically triggered helicopter (solid line) has similar elevation performance, and both of them track the commanded signal (dash dotted line) with small overshoot and no steady state error. The middle plot is the performance of travel rate measured by rad/s, with x-axis indicating time and y-axis indicating travel rate. From the middle plot, we see that event triggered helicopter (dotted line) and periodically triggered helicopter (solid line) has similar performance, and both of them track the commanded signal (dash dotted line) with small overshoot and no steady state error. The bottom plot is the performance of pitch measured by rad, with x-axis indication time and y-axis indication pitch angle. We see that the event triggered helicopter (dotted line) and the periodically triggered helicopter (solid line) have similar pitch performance, and both of them only have small oscillation. Overall, we can say that the event triggered helicopter and the periodically triggered helicopter have similar performance.

2) Transmission times and performances with $0.01s$ *and* $0.02s$ *delay:* In this experiment, we first set the transmission delay to be 0.01s, and then set the transmission delay to be 0.02s to see how the system performances of event triggered helicopter and periodically triggered helicopter decay with respect to transmission delays.

In the presence of 0.01s delay, the system performances of both event triggered 3DOF helicopter and periodically triggered 3DOF helicopter are shown in Figure 6. The top plot is the performance of the elevation subsystem, the middle plot is the performance of travel rate, and the bottom plot is the performance of pitch. In these three plots, solid lines are performances of periodically triggered helicopter, dotted lines are performances of event triggered helicopter, and dash dotted lines are commanded signals. From the three plots of Figure 6, we can see that in the presence of 0.01s delay, the performance of event triggered helicopter is similar to the performance of periodically triggered helicopter, and both of them tracked the commanded elevation and travel rate with small overshoot and no steady state error with small oscillation in the pitch angle. Now, let us look at the transmission times of both event triggered helicopter and periodically triggered helicopter. The transmission times are shown in the 3rd and 4th row of Table VII. We see that compared with the periodically triggered helicopter system, the total transmission times in the event triggered helicopter system is about 0.2 of the total transmission times in the periodically triggered helicopter system. Therefore, we conclude that when the transmission delay is 0.01s, the event triggered helicopter and the

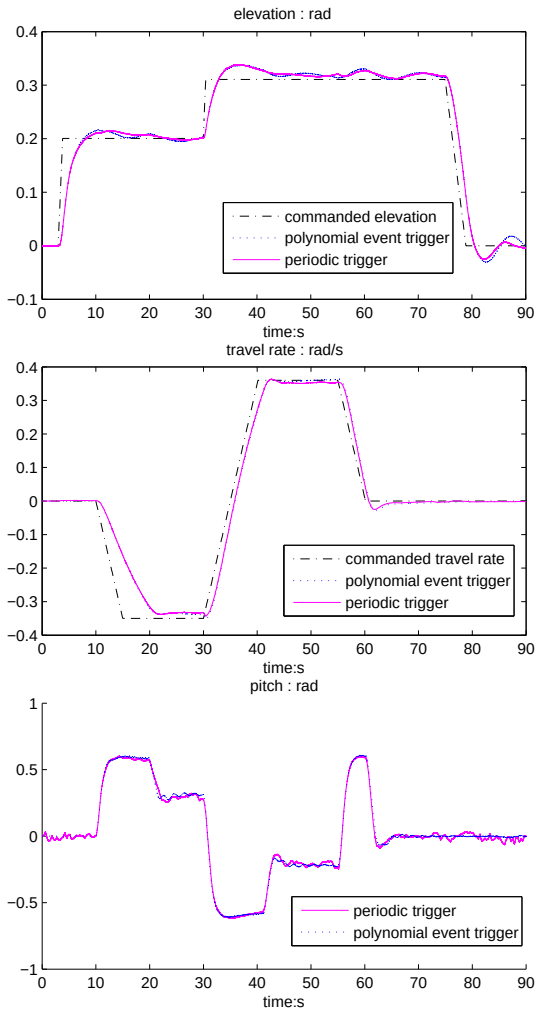

Fig. 6: System performance with 0.01s delay

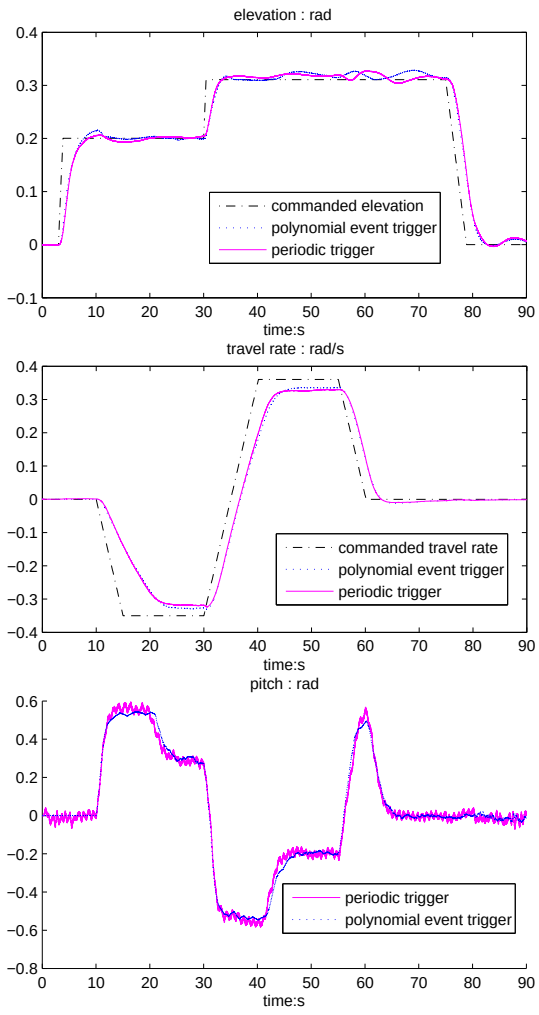

Fig. 7: System performance with 0.02s delay

periodically triggered helicopter achieved similar performances while the event triggered helicopter transmitted less than the periodically triggered helicopter.

When the transmission delay is 0.02s, the system performances are shown in Figure 7 with dotted lines indicating performances of the event triggered helicopter, solid line indicating performances of the periodically triggered helicopter, and dash dotted line indicating commanded signals. The top two plots are the performances of elevation and travel rate, respectively. We see that event triggered helicopter and periodically triggered helicopter had similar elevation and travel rate performance. While both event triggered system and periodically triggered system tracked the commanded elevation with small overshoot and no steady state error, there existed steady state errors in travel rate for both event triggered system and periodically triggered system. The bottom plot shows the performances of pitch. we can see that the event triggered helicopter had smaller oscillation in pitch than the periodically triggered helicopter. The transmission times of event triggered helicopter and the periodically triggered helicopter are given

in the 5th and 6th row of Table VII. We see that in event triggered system, while the transmission times of elevation subsystem remained at the same level, the transmission times of the pitch-travel subsystem increase to 2422 from 54 (0.005s delay) and 49 (0.01s delay). This is because the pitch angle kept oscillating, which means that the pitch subsystem was always in a transient process. So, the pitch-travel subsystem kept transmitting information during the whole running horizon. From Table VII, we see that event triggered helicopter transmitted more than the periodically triggered helicopter, and the total transmission times of event triggered helicopter is about 8 times of the transmission times of the periodically triggered helicopter. From Figure 7 and Table VII, we conclude that when the transmission delay is 0.02s, the event triggered helicopter had better performance than the periodically triggered helicopter, but consumed more communication resource than the periodically triggered helicopter.

ACKNOWLEDGMENT

The authors acknowledge the partial financial support of the National Science Foundation NSF-CNS-0931195 and NSF-ECCS-0925229.

REFERENCES

[1] 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.

[2] K. Arzen, "A simple event-based pid controller," in *Proc. 14th IFAC World Congress*, vol. 18, 1999, pp. 423–428.

[3] 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.

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

[5] 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.

[6] M. Mazo and P. Tabuada, "On event-triggered and self-triggered control over sensor/actuator networks," in *Decision and Control, 2008. CDC 2008. 47th IEEE Conference on*. IEEE, 2008, pp. 435–440.

[7] A. Anta and P. Tabuada, "To sample or not to sample: Self-triggered control for nonlinear systems," in *Automatic Control, IEEE Transactions on*, vol. 55, no. 9. IEEE, 2010, pp. 2030–2042.

[8] W. Heemels, J. Sandee, and P. Van Den Bosch, "Analysis of event-driven controllers for linear systems," in *International Journal of Control*, vol. 81, no. 4. Taylor & Francis, 2008, pp. 571–590.

[9] D. Dimarogonas and K. Johansson, "Event-triggered cooperative control," in *European Control Conference*, 2009, pp. 3015–3020.

[10] R. Cogill and S. Lall, "Suboptimality bounds in stochastic control: A queueing example," in *American Control Conference, 2006*. IEEE, 2006, pp. 1642–1647.

[11] 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.

[12] 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.

[13] L. Li and M. Lemmon, "Performance and average sampling period of sub-optimal triggering event in event triggered state estimation," in *submittted to conference of decision and control*. IEEE, 2011.

[14] A. Willsky, G. Wornell, and J. Shapiro, "Stochastic processes, detection and estimation," *Course notes for MIT*, vol. 6, 1995.

[15] L. L. Li and M. Lemmon, "Weakly coupled transmissions in networked event triggered output feedback systems," *submitted to Discrete Event Dynamic systems*, 2012.

[16] J. Sijs, “State estimation in networked systems,” Ph.D. dissertation, Ph. D. dissertation, Eindhoven University of Technology, The Netherlands, 2012.

[17] A. Arapostathis, V. Borkar, E. Fernández-Gaucherand, M. Ghosh, and S. Marcus, “Discrete-time controlled markov processes with average cost criterion: a survey,” *SIAM Journal on Control and Optimization*, vol. 31, no. 2, pp. 282–344, 1993.

[18] O. Hernández-Lerma and J. Lasserre, *Discrete-time Markov control processes: basic optimality criteria*. Springer New York, 1996.

[19] E. Gordienko and O. Hernández-Lerma, “Average cost markov control processes with weighted norms: value iteration,” *Appl. Math*, vol. 23, pp. 219–237, 1995.

[20] 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.

[21] O. Hernandez-Lerma and J. Lasserre, “Policy iteration for average cost markov control processes on borel spaces,” *Acta Applicandae Mathematicae*, vol. 47, no. 2, pp. 125–154, 1997.

[22] 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.

[23] L. Isserlis, “On a formula for the product-moment coefficient of any order of a normal frequency distribution in any number of variables,” *Biometrika*, vol. 12, no. 1/2, pp. 134–139, 1918.

[24] M. Choi, T. Lam, and B. Reznick, “Sums of squares of real polynomials,” in *Proceedings of Symposia in Pure Mathematics*, vol. 58. American Mathematical Society, 1995, pp. 103–126.

[25] V. Powers and T. Wörmann, “An algorithm for sums of squares of real polynomials,” *Journal of pure and applied algebra*, vol. 127, no. 1, pp. 99–104, 1998.

[26] P. Parrilo, “Semidefinite programming relaxations for semialgebraic problems,” *Mathematical Programming*, vol. 96, no. 2, pp. 293–320, 2003.

[27] S. Prajna, A. Papachristodoulou, P. Seiler, and P. Parrilo, “Sostools: Sum of squares optimization toolbox for matlab,” *Users guide*, 2004.

[28] D. Finkel, “Direct optimization algorithm user guide,” *Center for Research in Scientific Computation, North Carolina State University*, vol. 2, 2003.

[29] D. Jones, C. Perttunen, and B. Stuckman, “Lipschitzian optimization without the lipschitz constant,” *Journal of Optimization Theory and Applications*, vol. 79, no. 1, pp. 157–181, 1993.

[30] S. Bayraktar, “Aggressive landing maneuvers for unmanned aerial vehicles,” Ph.D. dissertation, Massachusetts Institute of Technology, 2006.