M. Xia, V. Gupta, and P.J. Antsaklis, State Estimation of Multiple Plants over a Shared Communication Network, ISIS Technical Report ISIS-2013-001, March 2013.

State Estimation of Multiple Plants over a Shared Communication Network

Technical Report of the ISIS Group

at the University of Notre Dame

ISIS-13-001

March 2013

M. Xia and V. Gupta and P. J. Antsaklis

Department of Electrical Engineering

University of Notre Dame

Notre Dame, IN 46556

Interdisciplinary Studies in Intelligent Systems

State Estimation of Multiple Plants over a Shared Communication Network

M. Xia and V. Gupta and P. J. Antsaklis
Department of Electrical Engineering
University of Notre Dame
Notre Dame, IN 46556
e-mail: { mxia, vgupta2, antsaklis.1 }@nd.edu

Abstract

This paper considers state estimation for multiple plants over a shared communication network. Each linear time-invariant plant transmits information through the common network according to either a time-triggered or an event-triggered rule. For an event-triggered algorithm with CSMA (carrier sense multiple access), each plant is assumed to access the network based on a priority mechanism. For a time-triggered algorithm combined with TDMA (time division multiple access), each plant uses the network according to an off-line scheduling. Performance in terms of the communication frequency and the estimation error covariance is analytically characterized for some special cases. The main result is that event-triggered schemes may perform worse than time-triggered schemes when considering the effect of communication network.

I. Introduction

A. Background and Motivating Problem

In networked control systems, periodic sampling is normally used for control and estimation purposes. Standard sampled-data control theory can be used for analysis of periodic sampling [1], [2]. In recent literature, event-based techniques have been proposed for more efficient utilization of network resources [1], [3], [4]. The basic idea of event trigger is to update the control input only when something significant occurs. It has been shown that event-triggered schemes are superior to time-triggered methods in the sense that comparable performance can be achieved at the expense of vastly reduced communication rate [5].

An important stream of work in event triggered schemes is analytically characterizing the performance achieved through such algorithms. Works such as Tabuada [3] and Lemmon [5] focus on stabilizing control tasks via a state-dependent event triggered scheduler. Rabi and Baras [6] consider a scalar system driven by a weiner process and for a constant threshold event triggering policy, provide analytic expressions for the variance of the state. Li and Lemmon [7] extend this work to vector linear process and propose a way to calculate sub-optimal event triggering thresholds. The optimal policies are derived in [8] as solutions to an average cost optimization problem using dynamic programming. A related problem is studied in Imer and Basar [9] where the estimation error is minimized under the constraint of limited observer actions. Shi and Qiu [10] propose a joint time-based and event-based schedule to tradeoff the estimation error and the computation complexity permitting limited communication resources. The works discussed above concentrate on a single loop in the network and evaluate the system performance (such as stability or error covariance), but they do not provide analytical results for the communication rate through event triggered algorithms.

As for a single loop in the system, event trigger may outperform periodic sampling by reducing utilization of the communication resources as shown in the previous results [1], [3], [5]. When multiple loops share a common network, however, the interaction between control architecture and the communication strategies become more complicated. For instance, with multiple plants over a shared medium, a multiple access method is needed to multiplex the data streams, such as TDMA (time division multiple access) and CSMA (carrier sense multiple access) [11], [12]. Obviously, the methods to access the network have a significant impact on the performance of the system [13]. Intuitively, TDMA is suitable for periodic sampling and CSMA is suitable for event trigger since the arrival of events for different loops are generally unpredictable. For event trigger associated with CSMA and time trigger with TDMA, will event triggered schemes still perform better than time triggered schemes? This is the problem we are interested in and for the best of our knowledge, this problem has not been fully investigated.

B. Relevant Work

Works such as [14], [15], [2] start to look at the problem of multiple loops in the network. Cervin and Henningsson [2] consider a NCS with a number of independent control loops over a shared network and use numerical methods to compute the control performance under various multiple access schemes such as TDMA, FDMA and CSMA. The interference between event triggered control and shared medium is also studied in Cervin [15] where the stationary state distribution is calculated based on a simple model. Rabi and Johansson [14] consider packet loss due to contention of different loops (using event triggered control) for the shared medium and characterize the control performance based on the packet loss rate. However, they assume the losses for different loops are independent, which is not true in general [11].

The analytical results of [15][14] are grounded on either a simplified system model or an unrealistic assumption on the correlation of different loops.

The common communication protocols have been studied in [2], but their results are mainly derived from simulation. To analytically characterize the interaction between control and communication, a suitable model of the communication systems is needed. Most recently, a simple ALOHA protocol is used for modeling the communication networks in Blind [16]. Similar to [14], each loop is modeled by noisy integrator dynamics. The correlation of different loops is removed through a particular triggering rule. In such networks, packets are transmitted whenever an event is generated for the plant. However, the packets will be lost if more than two plants transmit information simultaneously since no backup strategy for collision resolution is considered. CSMA protocols for event based systems are considered in Ramesh and Johansson [11]. A Markov based model is introduced to characterize the probability of successful transmission for each plant in steady state. The key assumption is that the conditional probability of a busy channel for the attempting node to transmit is independent for each node as in [17]. Although the correlations between various loops and the need for joint analysis between event trigger and CRM are addressed, no performance analysis of the NCS is provided in this work.

C. *Summary of this Paper*

This paper is an extension of [18] where performance expressions for a single loop are characterized. In this paper, we consider multiple plants transmitting information through a common network according to either a time triggered rule or an event triggered rule. To avoid collision, we use CSMA for event trigger based on a priority mechanism as in [2] and TDMA for time trigger. Performance in terms of the communication rate and the estimation error covariance is analytically characterized under various medium access schemes for some special cases. Our results demonstrate time trigger can outperform event trigger when multiple loops share access to the network.

The rest of the paper is organized as follows. Section II describes medium access schemes and presents an illustrating example to show time trigger may perform better than event trigger. Section III presents the problem formulation. The analysis for event triggered estimation of a single plant is provided in Section IV and extended to NCS with multiple plants in Section V. Numerical illustration is provided in Section VII. This paper concludes with some avenues for future work in Section VIII.

Notation: The n-dimensional real space is denoted by $\mathbb{R}^n$. Denote the vector of all zeros by $\underline{0}$ and the vector of all ones by $\underline{1}$. The vector is denoted by $\underline{z}$ or simply z when causing no confusion. The infinity norm of a vector x is denoted by $|x|$. For a matrix M, the (i, j)-th element is denoted by $M(i, j)$. The variable y is less than but close to a real number b is denoted by $y \lesssim b$. For a m-dimensional multivariate Gaussian random variable X with mean vector μ and covariance R, we denote the generalization of the cumulative distribution function F function as $Pr(|X| \leq x) \triangleq F(m, \mu, R, x)$, where the inequality is interpreted element-wise. Also for the truncated multivariate Gaussian random variable obtained by truncating X between the vectors t_1 and t_2, define the variance by $\Sigma(X, t_1, t_2)$. As with the standard F functions and truncated Gaussian distributions, evaluation of these generalizations is done through Gaussian integrals (see, e.g., [19, Equation (16)] for formulas for the variance of truncated Gaussian distributions) and is a standard feature in most statistics packages.

II. Preliminary Work

In this section, we first introduce medium access (MA) schemes and then present a simple example to demonstrate time trigger may perform better than event trigger with associated MA schemes.

A. Medium Access Schemes

For event triggered algorithms, a contention based multi access method (such as CSMA) is introduced to multiplex the data flows. More specifically, when two or more plants intend to use the network simultaneously, the network grants the plant with the highest priority to access the network and the other packets will be discarded. The priority orders of the plants can be decided according to one of the following collision resolution mechanisms (CRM) [2], [11], [13].

- Static priority [13], [12]: The priority orders of the plants are decided in advance and remain fixed during system operation. This scheme is typically implemented by polling or token ring and used in Control Area Network (CAN). It is reasonable to assume the plant who is more sensitive to delays has higher priority.
- Random priority [2], [11]: In wireless networks (such as Ethernet or WLAN), random back off strategies are normally used and a random plant is allowed to access the network maybe after some delay. When plant information is not available to the network, it is reasonable to randomly assign priorities to the competing nodes.
- Dynamic priority [13], [2]: The priorities are adapt to dynamically changing progress during the system operation. The objective is to use the the network more efficiently. It makes sense to assume the plant with maximum error to access the network first (MEF). The dynamic scheduler can also be used in CAN.

For time triggered algorithms, TDMA is used to multiplex the date where a cyclic access schedule is decided in advance. For a NCS consisting of finite number of plants, an optimal schedule can be found by evaluating the cost for every possible schedule [2]. When there is no cost for using the network, it is reasonable to assume the network is used at every time step. Applications of TDMA include mobile communications and WirelessHART [20].

B. One Illustrating Example

Consider the following example where a NCS consists of two plants over a shared communication medium.

Example 1: The plant $\mathcal{S}_i$ is described as follows.

$$\mathcal{S}_i : x_i(k+1) = A_i x_i(k) + w_i(k),$$
$$y_i(k) = x_i(k),$$

with $A_1 = 1, A_2 = 0.9$. The process noise $w_i(k)$ is white, zero mean, Gaussian with covariance unity. The initial condition of $x_i(0)$ is a normal Gaussian random variable. The process noise and initial conditions are assumed to be mutually independent.

Denote the estimate for state $x_i(k)$ by $\hat{x}_i^{dec}(k)$. At the ith estimator, we have

$$\hat{x}_i^{dec}(k) = \begin{cases} x_i(k), & \text{if } x_i(k) \text{ received at } k, \\ A_i \hat{x}_i^{dec}(k-1), & \text{otherwise,} \end{cases}$$

Define the estimation error

$$e_i^{dec}(k) = x_i(k) - \hat{x}_i^{dec}(k),$$

and the quality of estimate for the NCS is measured by

$$J = \sum_{i=1}^{2} \lim_{t\to\infty} \frac{1}{t} \sum_{k=0}^{t} \mathbb{E}\left[e_i^{dec}(k)\right]^2.$$

Event trigger: Information transmission for $\mathcal{S}_i$ occurs if the local event occurs, i.e. when

$$\mid x_i(k) - A_i \hat{x}_i^{dec}(k-1) \mid > \varepsilon_i,$$

where ε_i is a given constant. CRM described in Section II-A is applied when two local events occur simultaneously. We assume the network allows each plant to transmit at least once for every T_e time steps. This assumption is to guarantee fairness and guard against the practical concern of maximum delay that each plant can tolerate.

Time trigger: Each plant uses the network periodically. To avoid collision, we assume the two plants use the network asynchronously. Since there is no cost associated with using the network, we consider the communication rate $P \lesssim 1$. As an example, $P = 0.98$ is the result by transmitting $\mathcal{S}_1$ at odd time steps in every 50 time steps and transmitting $\mathcal{S}_2$ at even time steps except for the multiples of 50.

The simulation results are provided in Figure 1 for event trigger with different CRMs by conducting $10,000$ Monte Carlo experiments and setting $T_e = 10$ and $\varepsilon_1 = \varepsilon_2$. It can be seen that the communication rate is similar for various CRMs (in the top plot), but the system performance (in the bottom plot) is quite different for small triggering levels. The communication rate converging to 0.2 is because for large ε (no local events are generated), the network transmits information for each plant every T_e time steps.

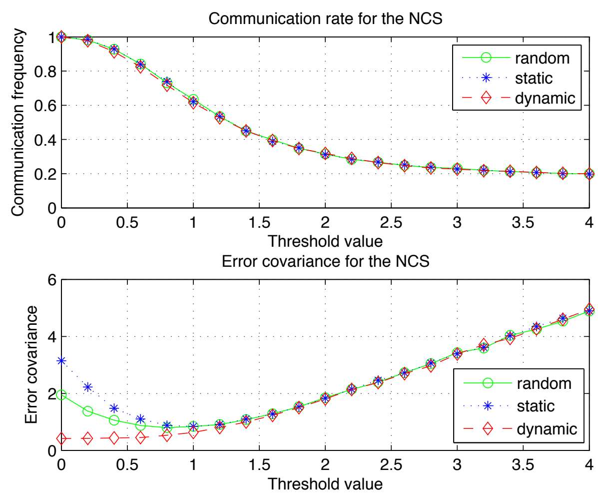

Fig. 1. The communication frequency and the error covariance for Example 1 using event trigger with various CRMs.

The comparison between time trigger and event trigger are summarized in Table I. We see from Table I that event trigger with static and random schedulers have larger estimation error covariance than time trigger with TDMA, under the same communication rate $P = 0.98$. For event trigger with dynamic scheduler, the estimation error covariance can be larger than time trigger (with cost $J = 1$) when the triggering level $\varepsilon \geq 1$, as shown in the bottom plot of Figure 1.

In the next section, we are going to formalize the problem statement of state estimation for NCS under time-triggered and event-triggered schemes.

TABLE I

PERFORMANCE COMPARISON OF TIME TRIGGER AND EVENT TRIGGER UNDER THE SAME COMMUNICATION RATE

Scheme	Scheduler	Performance	Threshold	Comm Rate
Time-trigger	TDMA	$J = 1.036$	-	98%
	Static	$J = 2.618$	0.215	98%
Event-trigger	Random	$J = 1.348$	0.221	98%
	Dynamic	$J = 0.427$	0.176	98%

III. PROBLEM FORMULATION

Consider the problem setup as shown in Figure 2 where N plants transmit information over a shared network.

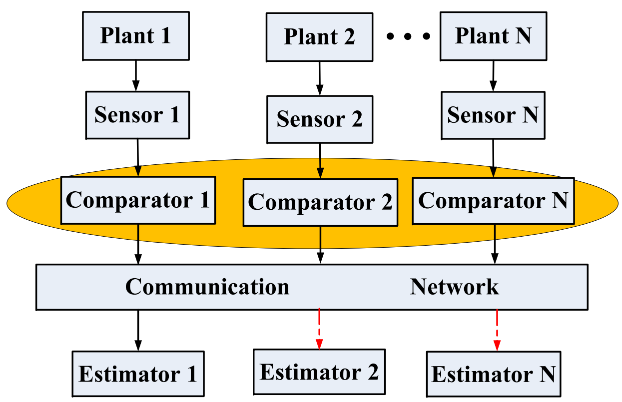

Fig. 2. Problem setup considered in this paper.

Plant and Sensor: The ith plant is described by the following discrete linear time-invariant evolution:

$$x_i(k+1) = A_i x_i(k) + w_i(k),$$
$$y_i(k) = C_i x_i(k) + v_i(k), \tag{1}$$

where $x_i(k) \in \mathbb{R}^n$ denotes the state vector, $y_i(k) \in \mathbb{R}^m$ is the output vector, $w_i(k)$ is the process noise assumed to be white Gaussian with zero mean and covariance $R_{w_i} > 0$, and $v_i(k)$ is the measurement noise assumed to be white Gaussian with zero mean and covariance $R_{v_i} > 0$. For the analytical results in the paper, we will consider $n = m = 1$, although the arguments can be easily generalized at the expense of more notation. The initial condition of the process $x_i(0)$ is assumed to be a Gaussian random vector with zero mean and covariance $R_i(0)$. The process noise $\{w_i(k)\}$, the measurement noise $\{v_i(k)\}$, and the initial condition $x_i(0)$ are assumed to be mutually independent. A_i and C_i are real matrices and the pair (A_i, C_i) is assumed to be observable.

Estimator: At every time k, the ith estimator generates a minimum mean squared error (MMSE) estimate for the state $x_i(k)$ based on whatever information is available to it. In a time-triggered architecture, this information is the set of measurements $\{y_0, \cdots, y_k\}$ that received from the network in a periodic manner. In an event-triggered architecture, this information is any information transmitted by the comparator, and the time steps at which information transmission occurs. Denote the estimate for state $x_i(k)$ held by the ith estimator as $\hat{x}_i^{dec}(k)$. At the ith estimator, we have

$$\hat{x}_i^{dec}(k) = \begin{cases} x_i(k), & \text{if the } i\text{th packet received,} \\ A_i \hat{x}_i^{dec}(k-1), & \text{otherwise.} \end{cases}$$

where $A_i \hat{x}_i^{dec}(k-1)$ is the optimal estimate at the estimator if the estimator did not receive any information at time k. Thus, the estimation error evolves as

$$e_i^{dec}(k) = \begin{cases} 0, & \text{if the } i\text{th packet received,} \\ A_i e_i^{dec}(k-1) + w_i(k-1), & \text{otherwise.} \end{cases}$$

Comparator: The event-triggered algorithm is implemented at the comparator. We consider a level based scheme. Specifically, we consider two cases. In the first simpler case, we assume that the measurement noise $v_i(k)$ is identically zero, and the matrix C_i is identity. Thus, the ith sensor observes the state $x_i(k)$ at every time k. The local event is defined as

$$\mid e_i^{comp}(k) \mid > \varepsilon_i, \tag{2}$$

where $e_i^{comp}(k) \triangleq x_i(k) - A_i \hat{x}_i^{dec}(k-1)$, the threshold ε_i is a given constant. The second case we consider is when the measurement noise is not zero. In this case, we assume that the comparator calculates a local estimate $\hat{x}_k^{enc}$ of the state x_k based on all measurements $\{y_0, \cdots, y_k\}$. However, in this case, $e_i^{comp}(k) \triangleq \hat{x}_i^{enc}(k) - A_i \hat{x}_i^{dec}(k-1)$. Calculation of $\hat{x}_i^{enc}(k)$ admittedly requires more computational resources at the comparator; however, this scheme can transmit much more information than simply transmitting the latest measurement $y(k)$ (c.f. [21]).

Communication Network: The communication network is modeled by satisfying the following assumptions.

- A1: The network does not permit simultaneous transmissions. The transmission time is less than one time step [11], [16].
- A2: The plant sends information according to an off-line scheduling (for time-triggered schemes) or whenever an event is generated (for event-triggered schemes).
- A3: When two or more plants send information simultaneously, the network will transmit the packet received from the plant with highest priority based on CRMs in Section II-A and the rest packets will be discarded.
- A4: The network allows each plant to transmit at least once for every T time steps to guard against the practical concern of maximum tolerable delay.

We are interested in the problem of state estimation for the NCS, particularly the following two metrics: (1) The communication rate P, which is defined as the average probability for the network to transmit information at each time step. (2) The quality of estimate for the NCS, which is measured by the aggregate error covariance,

$$J = \sum_{i=1}^{N} \lim_{t\to\infty} \frac{1}{t} \sum_{k=0}^{t} \mathbb{E}\left\{ e_i^{dec}(k)[e_i^{dec}(k)]^T \right\},$$

with $e_i^{dec}(k) \triangleq x_i(k) - \hat{x}_i^{dec}(k)$ as the estimation error for $\mathcal{S}_i$.

IV. Single Plant Results

In this section, we present results for event triggered estimation of a single plant. The analysis is extended to NCS with multiple event triggered loops in the next section. We will focus on the case when the state is observed. When the process state is not observed by the sensor, the development will be similar by using, e.g., a Kalman filter [18].

In the following analysis, we drop the subscript i in the previous section. The information can be successfully transmitted through the network whenever $|e^{comp}(k)| > \varepsilon$ since there is no contention to access the network. As shown in Figure 3, we can define a discrete-time discrete-state Markov chain $\mathcal{M}$ with $T+1$ modes, the state $\{X(k)\}_{k\geq 0}$ and the transition probabilities

$$p_{ij} = Pr(X(k+1) = j \big| X(k) = i),$$

such that $X(k) = j$ implies that at time k, the last transmission occurred at time $k-j$.

The communication frequency and the estimation error covariance are characterized by this Markov chain. To this end, define the random variables

$$Z_i(k) = \sum_{j=0}^{i} A^j w(k+i-j), \qquad 0 \leq i \leq T. \tag{3}$$

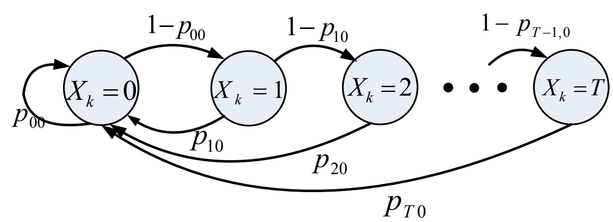

Fig. 3. Transition graph of the Markov Chain defined for a single plant.

Since the noise $w(k)$ is white, the probability density function of the variables $Z_i(k)$ is independent of k. In the sequel, we will simply write Z_i to denote the random variables. Clearly, for any i, the vector random variable $M_i = \begin{bmatrix} Z_0^T, Z_1^T, \cdots, Z_i^T \end{bmatrix}^T$ has a multi-variate normal distribution with mean 0 and covariance matrix R_i as

$$\begin{bmatrix} R_w & R_wA^T & \cdots & R_w(A^T)^i \\ AR_w & AR_wA^T + R_w & AR_w(A^T)^2 + R_wA^T & \cdots \\ \vdots & & \ddots & \\ A^iR_w & \cdots & & \end{bmatrix}.$$

Now for $1 \leq i \leq T$, define the events

$$N_i = (| Z_0 |< \varepsilon) \cap (| Z_1 |< \varepsilon) \cap \cdots \cap (| Z_{i-1} |< \varepsilon), \quad (4)$$

with the convention that N_0 is the sure event. The following result is immediate:

$$Pr(N_i) = F(ni, 0, R_i, \varepsilon\underline{1}), \quad (5)$$

with $Pr(N_0) = 1$.

Lemma 1: Consider the Markov chain $\mathcal{M}$ as defined above. The transition probabilities p_{ij} are given by

$$p_{ij} = \begin{cases} 1 - \frac{F(n(i+1),0,R_{i+1},\varepsilon\underline{1})}{F(ni,0,R_i,\varepsilon\underline{1})} & 0 \leq i \leq T-1, j = 0 \\ 1 & i = T, j = 0 \\ 1 - p_{i0} & 0 \leq i \leq T-1, j = i+1 \\ 0 & \text{otherwise} \end{cases} \quad (6)$$

Proof: We concentrate on the case when $0 \leq i \leq T-1, j = 0$ since the other expressions are obvious from the structure of the Markov chain shown in Figure 3. Consider the transition probability

p_{00}. Since $X(k)=0$ is equivalent to $e^{dec}(k)=0$, we have

$$\begin{aligned}p_{00} &= Pr(X(k+1)=0\big|X(k)=0)\\ &= Pr(|w(k)|>\varepsilon\big|e^{dec}(k)=0)\\ &\overset{(a)}{=} Pr(|w(k)|>\varepsilon) = Pr(\mid Z_0 \mid > \varepsilon),\end{aligned}$$

where (a) holds because $e^{dec}(k)$ is independent of the process noise at time step k. Similarly, for any i such that $0 \le i \le T-1$, the probability

$$\begin{aligned}p_{i0} =& Pr(X(k+1)=0\big|X(k)=i)\\ \overset{(b)}{=}& Pr\left(\mid Z_i \mid > \varepsilon\big|N_i, e^{dec}(k-i)=0\right)\\ \overset{(c)}{=}& Pr\left(\mid Z_i \mid > \varepsilon\big| \mid Z_{i-1} \mid < \varepsilon, \cdots, \mid Z_0 \mid < \varepsilon\right)\\ =& \frac{Pr(\mid Z_i \mid > \varepsilon, N_i)}{Pr(N_i)} = 1 - \frac{Pr(N_{i+1})}{Pr(N_i)},\end{aligned}$$

where (b) follows the Markovian property and the definiations in (3), and (c) holds because $e^{dec}(k-i)$ is independent of the process noise after time step $k-i$ and in particular, Z_i. Now the result follows from (5), which can be evaluated using Gaussian integrals and the fact that $p_{T0}=1$. ∎

Theorem 2: The average communication rate for the event triggered algorithm described above is given by $\frac{1}{1+\sum_{j=1}^{T}\prod_{i=0}^{j-1}(1-p_{i0})}$, which can be calculated using (6).

Proof: The average communication rate for the system is given by $\lim_{k\to\infty} Pr(X(k)=0)$. From the fact that p_{i0}'s are time-invariant and using the structure of the Markov chain from Fig. 3, the probability for each mode j ($j \ge 1$) can be computed as

$$\begin{aligned}Pr(X(k)=j) &= (1-p_{j-1,0})Pr(X(k)=j-1)\\ &= \prod_{i=0}^{j-1}(1-p_{i0})Pr(X(k)=0).\end{aligned} \tag{7}$$

Thus, the balance equation for the Markov chain yields

$$\begin{aligned}1 &= \sum_{j=0}^{T} Pr(X(k)=j)\\ &= Pr(X(k)=0) + \sum_{j=1}^{T}\prod_{i=0}^{j-1}(1-p_{i0})Pr(X(k)=0)\\ &= \left(1+\sum_{j=1}^{T}\prod_{i=0}^{j-1}(1-p_{i0})\right)Pr(X(k)=0).\end{aligned}$$

The required probability $Pr(X(k)=0)$ can now be calculated as

$$Pr(X(k)=0) = \frac{1}{1+\sum_{j=1}^{T}\prod_{i=0}^{j-1}(1-p_{i0})}.$$

■

The other performance metric is the covariance of estimation error $\Pi(k) = \mathbb{E}[e^{dec}(k)(e^{dec}(k))^T]$ which is given by the following relation.

Theorem 3: The steady state average error covariance $\Pi = \lim_{k\to\infty}\Pi(k)$ for the event triggered algorithm described above is given by

$$\Pi = \sum_{j=1}^{T}\prod_{t=0}^{j-1}(1-p_{t0})Pr(X(k)=0)\Sigma_{M,j}(j,j), \tag{8}$$

where $\Sigma_{M,j} = \Sigma(M_j, -\varepsilon\underline{1}, \varepsilon\underline{1})$.

Proof: We use the relation

$$\Pi(k) = \sum_{j=0}^{T} Pr(X(k)=j)\mathbb{E}[e^{dec}(k)(e^{dec}(k))^T \mid X(k)=j].$$

For $j=0$, since the estimation error $e^{dec}(k)=0$, we obtain

$$\mathbb{E}[e^{dec}(k)(e^{dec}(k))^T \mid X(k)=j] = 0.$$

For $j>0$, we use the fact that the error covariance $e^{dec}(k)$ under the event $X(k)=j$ is simply $\sum_{i=0}^{j} A^i w(k-i)$. However, since the process noise $w(j)$ is white and has a time-invariant probability distribution function, we can alternatively write

$$\mathbb{E}[e^{dec}(k)(e^{dec}(k))^T \mid X(k)=j] = \text{var}[Z_{j-1} \mid N_j],$$

where var(X) is the variance of the random variable X and N_i was defined in (4). The variance of Z_{j-1} is given by the (j,j)-th element of the variance matrix of M_j; however, as calculated under the truncation imposed by N_j, i.e., all the elements $Z_0, \cdots, Z_{j-1}$ being bounded between $-\varepsilon\underline{1}$ and $\varepsilon\underline{1}$. This variance is given by $\Sigma_{M,j}(j,j)$. Together with (7), this yields the desired expression. ■

Together, these two results provide analytic expressions for the communication frequency and average error covariance given any level ε.

V. Main Results

In this section, we present the main results of this paper. We are going to analyze performance of event triggered algorithms with various CSMA schedulers.

A. Markov Model for Multiple Plants

We focus on the case when plant states can be observed. For a NCS with $N \geq 2$ plants over a common network, we can also define a discrete-time discrete-state Markov chain $\mathcal{M}$ with $N_s = (T_e + 1)T_e \cdots (T_e - N + 2)$ states $\{X(k)\}_{k \geq 0} \in \mathbb{R}^N$ and the transition probabilities

$$Pr[X(k+1) = \underline{\mathrm{m}} \big| X(k) = \underline{\mathrm{n}}] \triangleq p(m_1, \cdots, m_N \big| n_1, \cdots, n_N),$$

such that $X(k) = \underline{\mathrm{m}}$ implies that at time k, the last transmission for the ith plant occurred at time $k - m_i$. Note that $m_i \neq m_j$ for all $i \neq j$ since the network does not permit simultaneous transmissions. Performance of event triggered algorithms can be characterized by this Markov chain. In the following analysis, we concentrate on the case when $N = 2$ and the arguments can be easily generalized to $N > 2$.

When two plants share a network, at every time step, there are 3 possibilities of information transmission:

- The network transmits information from $\mathcal{S}_1$.
- The network transmits information from $\mathcal{S}_2$.
- The network does not transmit any information.

This corresponds to the structure of the Markov chain. In particular, for any mode $\{i_1, i_2\}$ when $i_1, i_2 < T_e$, it can go to the following modes correspondingly,

$$\begin{bmatrix} 0 \\ i_2 + 1 \end{bmatrix}, \begin{bmatrix} i_1 + 1 \\ 0 \end{bmatrix}, \begin{bmatrix} i_1 + 1 \\ i_2 + 1 \end{bmatrix},$$

whose transition probabilities are determined by the scheduling policies. For the modes with $i_1 = T_e$ or $i_2 = T_e$, the network transits information for $\mathcal{S}_1$ or $\mathcal{S}_2$, respectively. Thus, for any scheduling policy, we have transitions as

$$\begin{bmatrix} T_e \\ i_2 \end{bmatrix} \to \begin{bmatrix} 0 \\ i_2 + 1 \end{bmatrix}, \begin{bmatrix} i_1 \\ T_e \end{bmatrix} \to \begin{bmatrix} i_1 + 1 \\ 0 \end{bmatrix}$$

with probability 1. To clarify this, let us see an example.

Example 4: Consider a NCS with $N = 2$ plants over a shared medium. Assume the maximum delay that each plant can tolerate is $T_e = 2$. We can define a Markov chain with the following $N_s = 6$ modes as shown in Fig. 4. The communication rate for $\mathcal{S}_1$ and $\mathcal{S}_2$ are given as

$$P_1 = Pr(\begin{bmatrix} 0 \\ 1 \end{bmatrix}) + Pr(\begin{bmatrix} 0 \\ 2 \end{bmatrix}),$$

$$P_2 = Pr(\begin{bmatrix} 1 \\ 0 \end{bmatrix}) + Pr(\begin{bmatrix} 2 \\ 0 \end{bmatrix}),$$

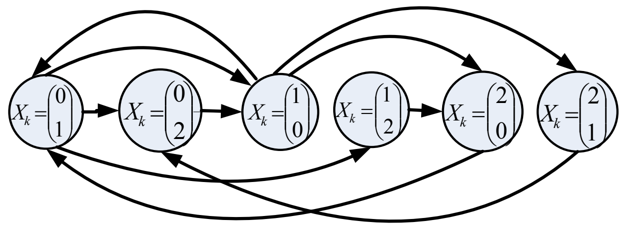

Fig. 4. Illustrating example for the Markov model with $T_e = 2$.

respectively. The communication rate for the network is then given by $P_0 = P_1 + P_2$. From the mode $\{1,0\}$ and $\{0,1\}$, there are three possible transitions and the following transitions are with probability 1.

$$\begin{bmatrix}0\\2\end{bmatrix} \to \begin{bmatrix}1\\0\end{bmatrix}, \begin{bmatrix}1\\2\end{bmatrix} \to \begin{bmatrix}2\\0\end{bmatrix},$$
$$\begin{bmatrix}2\\0\end{bmatrix} \to \begin{bmatrix}0\\1\end{bmatrix}, \begin{bmatrix}2\\1\end{bmatrix} \to \begin{bmatrix}0\\2\end{bmatrix}.$$

To characterize the system performance, we need to calculate the probability of each Markov mode. To this end, define $\underline{\mathrm{P}} \in \mathbb{R}^{N_s}$ as the vector for probability of each mode and define $\underline{\mathrm{b}} = [1, 0, \cdots] \in \mathbb{R}^{N_s}$. The relations of the modes are given through the following equation

$$\Delta \underline{\mathrm{P}} = \underline{\mathrm{b}}, \tag{9}$$

where $\Delta \in \mathbb{R}^{N_s \times N_s}$ with the first row $[1, 1, \cdots, 1]$ given by the balance equation and the rest elements can be determined from the structure of the Markov model. One can verify that the matrix Δ is always of full rank. This guarantees the above equation (9) has a unique solution.

Remark 1: The matrix Δ may not be unique since the relations between the Markov modes can be expressed in various manners, however, all these Δs will give the same probability of each mode in the end.

In the next, we are going to characterize this matrix Δ and further evaluate the performance of event triggered algorithms with static, random and dynamic schedulers through the Markov model defined above.

B. Event Trigger with Static Scheduler

We assume the ith plant has the ith priority without loss of generality. The NCS will execute Algorithm 1. This procedure guarantees that $\mathcal{S}_1$ wins the arbitration to access the network whenever contends with

$\mathcal{S}_2$.

Algorithm 1 Event Trigger with Static Scheduler

if $t_1 \geq T_e$ **then**
  transmit the packet from $\mathcal{S}_1$
else
  if $t_2 \geq T_e$ **then**
    transmit the packet from $\mathcal{S}_2$
  else
    if $\mid e_1^{comp}(k) \mid > \varepsilon_1$ **then**
      transmit the packet from $\mathcal{S}_1$
    else
      if $\mid e_2^{comp}(k) \mid > \varepsilon_2$ **then**
        transmitthe packet from $\mathcal{S}_2$
      else
        no transmission
      end if
    end if
  end if
end if

Lemma 2: By using static scheduler, for any $0 \leq i < T_e$,

$$p(0, i+1 \big| T_e, i) = 1; p(i+1, 0 \big| i, T_e) = 1. \tag{10}$$

Furthermore, for $0 < i < T_e$, we have

$$p(0, i+1 \big| 0, i) = p_{0,0}^{(1)}, \tag{11}$$

where $p_{0,0}^{(1)}$ can be calculated through (6) using $\{A_1, w_1\}$.

Proof: The equality (10) holds since the network transmits information for each plant at least once every T_e time steps. (11) holds because $\mathcal{S}_1$ has higher priority and thus information transmission is delayed for $\mathcal{S}_2$ when local event for $\mathcal{S}_1$ is generated and $i < T_e$, i.e.

$$\begin{aligned} p(0, i+1 \big| 0, i) &= Pr\left(\begin{bmatrix} 0 \\ i+1 \end{bmatrix} \mid \begin{bmatrix} 0 \\ i \end{bmatrix}\right) \\ &= Pr(X_1(k+1) = 0 \mid X_1(k) = 0) \\ &= Pr(|w_1(k)| > \varepsilon) \triangleq p_{0,0}^{(1)}. \end{aligned}$$

■

Let us consider Example 4. We have the following relation

$$Pr\left(\begin{bmatrix} 1 \\ 0 \end{bmatrix}\right) = Pr\left(\begin{bmatrix} 0 \\ 1 \end{bmatrix}\right) p_{01}^{(1)} \bar{p}_{10}^{(2)} + Pr\left(\begin{bmatrix} 0 \\ 2 \end{bmatrix}\right), \tag{12}$$

where $p_{01}^{(1)} = 1 - p_{00}^{(1)}$ and $\bar{p}_{10}^{(2)}$ given by

$$\bar{p}_{10}^{(2)} = Pr(|A_2 w_2(k-1) + w_2(k)| > \varepsilon).$$

One step further, we have the following transition,

$$\begin{bmatrix} 0 \\ 1 \end{bmatrix} \to \begin{bmatrix} 1 \\ 0 \end{bmatrix} \to \begin{bmatrix} 2 \\ 1 \end{bmatrix}, \begin{bmatrix} 0 \\ 2 \end{bmatrix} \to \begin{bmatrix} 1 \\ 0 \end{bmatrix} \to \begin{bmatrix} 2 \\ 1 \end{bmatrix},$$

and from these transitions we have

$$\begin{aligned} Pr\left(\begin{bmatrix} 2 \\ 1 \end{bmatrix}\right) &= Pr\left(\begin{bmatrix} 0 \\ 1 \end{bmatrix}\right) p_{01}^{(1)} \bar{p}_{10}^{(2)} p_{12}^{(1)} p_{01}^{(2)} \\ &+ Pr\left(\begin{bmatrix} 0 \\ 2 \end{bmatrix}\right) \bar{p}_{12}^{(1)} p_{01}^{(2)}, \end{aligned} \tag{13}$$

where $p_{01}^{(2)} = 1 - p_{00}^{(2)}$ and

$$p_{12}^{(1)} = Pr(|A_1 w_1(k-1) + w_1(k)| < \varepsilon \big| |w_1(k-1)| < \varepsilon)$$

can be calculated through (6) by using $\{A_2, w_2\}$ and respectively $\{A_1, w_1\}$. $\bar{p}_{12}^{(1)}$ is given by

$$\bar{p}_{12}^{(1)} = Pr(|A_1 w_1(k-1) + w_1(k)| < \varepsilon).$$

We can obtain the following relations in a similar manner,

$$Pr\left(\begin{bmatrix} 1 \\ 2 \end{bmatrix}\right) = Pr\left(\begin{bmatrix} 0 \\ 1 \end{bmatrix}\right) p_{01}^{(1)}\bar{p}_{12}^{(2)}, \tag{14}$$

$$\begin{aligned} Pr\left(\begin{bmatrix} 2 \\ 0 \end{bmatrix}\right) &= Pr\left(\begin{bmatrix} 0 \\ 1 \end{bmatrix}\right) p_{01}^{(1)}\bar{p}_{10}^{(2)}p_{12}^{(1)}p_{00}^{(2)} \\ &+ Pr\left(\begin{bmatrix} 0 \\ 1 \end{bmatrix}\right) p_{01}^{(1)}\bar{p}_{12}^{(2)} + Pr\left(\begin{bmatrix} 0 \\ 2 \end{bmatrix}\right) \bar{p}_{12}^{(1)}p_{00}^{(2)}, \end{aligned} \tag{15}$$

$$\begin{aligned} Pr\left(\begin{bmatrix} 0 \\ 2 \end{bmatrix}\right) &= Pr\left(\begin{bmatrix} 0 \\ 1 \end{bmatrix}\right) p_{01}^{(1)}\bar{p}_{10}^{(2)}p_{12}^{(1)}p_{01}^{(2)} \\ &+ Pr\left(\begin{bmatrix} 0 \\ 1 \end{bmatrix}\right) p_{00}^{(1)} + Pr\left(\begin{bmatrix} 0 \\ 2 \end{bmatrix}\right) \bar{p}_{12}^{(1)}p_{01}^{(2)}, \end{aligned} \tag{16}$$

where $\bar{p}_{12}^{(2)} = 1 - \bar{p}_{10}^{(2)}$.

In such a way, we represent the probabilities of all modes through the relations with mode $\{0, 1\}$ and $\{0, 2\}$ as in (12)-(16). Then from the balance equation that the sum of all probabilities equal to 1, we can solve for probability of each mode. More compactly, define

$$a = p_{01}^{(1)}\bar{p}_{10}^{(2)}p_{12}^{(1)}p_{00}^{(2)} + p_{01}^{(1)}\bar{p}_{12}^{(2)}, b = p_{01}^{(1)}\bar{p}_{10}^{(2)}p_{12}^{(1)}p_{01}^{(2)},$$

and we obtain the probability for every individual mode from equation (9) with Δ given as

$$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ p_{01}^{(1)}\bar{p}_{10}^{(2)} & 1 & -1 & 0 & 0 & 0 \\ p_{01}^{(1)}\bar{p}_{12}^{(2)} & 0 & 0 & -1 & 0 & 0 \\ a & \bar{p}_{12}^{(1)}p_{00}^{(2)} & 0 & 0 & -1 & 0 \\ b & \bar{p}_{12}^{(1)}p_{01}^{(2)} & 0 & 0 & 0 & -1 \\ p_{00}^{(1)} + c & -1 + \bar{p}_{12}^{(1)}p_{01}^{(2)} & 0 & 0 & 0 & 0 \end{bmatrix}. \tag{17}$$

Remark 2: Notice $\bar{p}_{12}^{(1)} \neq p_{12}^{(1)}$, since in transitions such as

$$\begin{bmatrix} 0 \\ 2 \end{bmatrix} \rightarrow \begin{bmatrix} 1 \\ 0 \end{bmatrix} \rightarrow \begin{bmatrix} 2 \\ 1 \end{bmatrix},$$

$X_1(k) = 1$ is caused by $t_2 = T_e$ independent of the error $|w_1(k-1)|$ which yields $\bar{p}_{12}^{(1)}$. However, in transitions such as

$$\begin{bmatrix} 0 \\ 1 \end{bmatrix} \rightarrow \begin{bmatrix} 1 \\ 0 \end{bmatrix} \rightarrow \begin{bmatrix} 2 \\ 1 \end{bmatrix},$$

$X_1(k) = 1$ is caused by $|w_1(k-1)| < \varepsilon$ and this yields $p_{12}^{(1)}$. Similarly, we have $\bar{p}_{10}^{(2)} \neq p_{10}^{(2)}$.

Remark 3: For single plant case, we can easily obtain the relations between the modes from the structure of the Markov model. Particularly, the matrix Δ for single plant is given as

$$\begin{bmatrix} 1 & 1 & & & & \cdots 1 \\ p_{01} & -1 & & & & \\ & p_{12} & -1 & & & \\ & & p_{23} & -1 & & \\ & & & \ddots & \ddots & \\ & & & & p_{T-1,T} & -1 \end{bmatrix},$$

and the transition probabilities are given in Lemma 1. For the multiple case, however, it is more complicated because of coupling of the two Markov states in one mode.

By solving (9), we obtain the probability of each Markov mode. The following result is immediate.

Theorem 5: For $T_e = 2$, the average communication rate for $\mathcal{S}_1$ under event triggered algorithm described above is given by $P_1 = Pr(\{0,1\}) + Pr(\{0,2\})$, and $P_2 = Pr(\{1,0\}) + Pr(\{2,0\})$ for $\mathcal{S}_2$ through $\underline{\mathrm{P}} = \Delta^{-1}\underline{\mathrm{b}}$ with Δ given in (17). Furthermore, the average communication rate for the network is given by $P_0 = P_1 + P_2$.

The other performance metric is the covariance of the estimation error $\Pi_i(k) = \mathbb{E}[e_i^{dec}(k)(e_i^{dec}(k))^T]$, which is given by the following result.

Theorem 6: For $T_e = 2$, the steady state average error covariance for the rth plant, $\Pi_r = \lim_{k\to\infty} \Pi_r(k)$, under the event triggered algorithm described above is given by $\Pi_r(k) = \sum_{j=1}^{N_s} \Pi_r(j)$ from (18-24). Furthermore, the average error covariance for the NCS is given by $\Pi = \Pi_1 + \Pi_2$.

Proof: To calculate Π_1, we use the relation $\Pi_1 = \sum_{j=0}^{N_s} \Pi_1(j)$, where $\Pi_1(j)$ corresponds to the error covariance under the Markov mode j as defined above. We have

$$\Pi_1(1) = 0, \Pi_1(2) = 0, \tag{18}$$

since the estimation error $e_1^{dec}(k) = 0$. Under the Markov mode $\{1,0\}$, we have

$$\begin{aligned} \Pi_1(3) = {} & Pr(\{0,1\})\Delta(1,3)\mathrm{var}\{w_1(k) \mid |w_1(k)| < \varepsilon\} \\ & + Pr(\{0,2\})\mathrm{var}\{w_1(k)\}. \end{aligned}$$

As for single plant case, $\mathrm{var}\{w_1(k) \mid |w_1(k)| < \varepsilon\}$ is given by $\Sigma_{M,1}^{(1)}(1,1)$. Thus, we have

$$\Pi_1(3) = Pr(\{0,1\})\Delta(1,3)\Sigma_{M,1}^{(1)}(1,1) + Pr(\{0,2\})R_{w_1}. \tag{19}$$

Under the Markov mode $\{1,2\}$, we have

$$\Pi_1(4) = Pr(\{0,1\})\Delta(1,4)\Sigma_{M,1}^{(1)}(1,1). \tag{20}$$

We can also obtain the error covariance under mode $\{2,0\}$,

$$\begin{aligned}\Pi_1(5) = {} & Pr(\{0,2\})\bar{p}_{12}^{(1)} p_{00}^{(2)} \Xi_1 \\ & + Pr(\{0,1\}) p_{01}^{(1)} \bar{p}_{10}^{(2)} p_{12}^{(1)} p_{00}^{(2)} \Xi_2 \\ & + Pr(\{0,1\}) p_{01}^{(1)} \bar{p}_{12}^{(2)} \Xi_3,\end{aligned} \tag{21}$$

where $\Xi_1 = \Sigma(A_1 w_1(k-1) + w_1(k), -\varepsilon, \varepsilon)$ can be evaluated through Gaussian integrals, $\Xi_2 = \Sigma_{M,2}^{(1)}(2,2)$, and

$$\begin{aligned}\Xi_3 &= \text{var}\{A_1 w_1(k-1) + w_1(k) \mid |w_1(k-1)| < \varepsilon\} \\ &= A_1 \Sigma_{M,1}^{(1)}(1,1) A_1^T + R_{w_1}.\end{aligned}$$

Also, the error covariance under the mode $\{2,1\}$ is given by

$$\begin{aligned}\Pi_1(6) = {} & Pr(\{0,2\})\bar{p}_{12}^{(1)} p_{01}^{(2)} \Xi_1 \\ & + Pr(\{0,1\}) p_{01}^{(1)} \bar{p}_{10}^{(2)} p_{12}^{(1)} p_{01}^{(2)} \Xi_2.\end{aligned} \tag{22}$$

To calculate Π_2, similar to calculation of Π_1, we use the relation $\Pi_2 = \sum_{j=0}^{N_s} \Pi_2(j)$ with

$$\Pi_2(3) = 0, \Pi_2(5) = 0, \tag{23}$$

since the estimation error $e_2^{dec}(k) = 0$. We can also have the following relations

$$\begin{aligned}\Pi_2(1) &= R_{w_2}, \\ \Pi_2(2) &= Pr(\{0,1\}) p_{00}^{(1)} \Xi_4 + Pr(\{2,1\}) \Xi_5, \\ \Pi_2(4) &= Pr(\{1,2\}) \Xi_4, \\ \Pi_2(6) &= Pr(\{2,1\}) \Sigma_{M,1}^{(2)}(1,1),\end{aligned} \tag{24}$$

where $\Xi_4 = \Sigma(A_2 w_2(k-1) + w_2(k), -\varepsilon, \varepsilon)$ and $\Xi_5 = A_2 \Sigma_{M,1}^{(2)}(1,1) A_2^T + R_{w_2}$. Together with the probabilities from the previous theorem, this yields the desired expressions. ■

Remark 4: For single plant case, $e^{dec}(k) = 0$ for $X(k) = 0$ and $X(k) = j > 0$ implies the estimation error in previous steps all less than ε. As a result, the error covariance under the mode $X(k) = j > 0$ is simply

$$\Pi(j) \triangleq Pr(X(k)=j)\mathbb{E}[e^{dec}(k)(e^{dec}(k))^T \mid X(k)=j]$$
$$= Pr(X(k)=j)\Sigma_{M,j}(j,j)$$

and the average estimation error covariance can be calculated as $\sum_{j=1}^{T}\Pi(j)$. For the multiple case, however, we have to identify how it comes to the current mode, caused by local events or network constraints, which yields different expressions for the error covariance.

Remark 5: For $T_e > 2$, a similar Markov chain can be defined by considering two more variables for each mode indicating how long has each plant signaled it wants to transmit and basically the issue is that to calculate the transition probabilities, one has to track the past states as well. This will result in too many Markov states and the problem is beyond the scope of this paper.

C. Event Trigger with Random Scheduler

With random scheduler, both plants have the chance to win the arbitration when contention occurs. Denote P_α as the probability for $\mathcal{S}_1$ to win, and $1-P_\alpha$ for $\mathcal{S}_2$. The access probability P_α is provided by the network [11]. When there is no contention, the plant can transmit information successfully whenever its local event is generated.

Consider the Markov model shown in Fig. 4. As mentioned earlier, one has to track the past states to calculate the transition probabilities. As an example, consider the transition from mode $\{0,1\}$ to $\{1,0\}$. The transition probability for $X_2(k)=1\to 0$ is not given by $\bar{p}_{10}^{(2)}$ (as for static scheduler). The reason is that $X_2(k)=1$ in the mode $\{0,1\}$ depends on the error $w_2(k-1)$ in the previous step. Similarly, in transition $\{0,1\}\to\{1,0\}\to\{2,0\}$, the transition probability for $X_1(k)=1\to 2$ is not given by $\bar{p}_{12}^{(1)}$ since $X_1(k)=0\to 1$ might be caused by $|w_1(k-1)|>\varepsilon$ as well. However, the approximation of ignoring this past and calculating transition probability only with the current state is close, which is verified through simulations. Through such approximations, the matrix Δ for a random scheduler is given as

$$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ \Delta_{21} & -1 & 0 & 0 & 0 & 1 \\ \Delta_{31} & 1 & -1 & 0 & 0 & 0 \\ p_{01}^{(1)}(1-\bar{p}_{10}^{(2)}) & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & \Delta_{53} & 1 & -1 & 0 \\ 0 & 0 & \bar{p}_{12}^{(1)}(1-p_{00}^{(2)}) & 0 & 0 & -1 \end{bmatrix}, \tag{25}$$

where

$$\Delta_{31} = [p_{01}^{(1)} + p_{00}^{(1)}(1 - P_\alpha)]\bar{p}_{10}^{(2)},$$

$$\Delta_{53} = p_{00}^{(2)}[\bar{p}_{12}^{(1)} + (1 - \bar{p}_{12}^{(1)})(1 - P_\alpha)],$$

$$\Delta_{21} = (1 - p_{01}^{(1)})(\bar{p}_{10}^{(2)} P_\alpha + 1 - \bar{p}_{10}^{(2)}).$$

By solving equation (9) with Δ given in (25), we can get the probability for each mode. The approximate results calculated in this way match closely to the Monte Carlo simulations as demonstrated in Section VII. We can therefore characterize the communication rate and error covariance from this Markov model along the same lines as for static scheduler.

D. Event Trigger with Dynamic Scheduler

With dynamic scheduler, when two local events are generated simultaneously, the network grants the one with maximum error $|e_i^{comp}(k)|$ to access the network first. As a result, the network transmits information for $\mathcal{S}_1$ if it has a larger error when both local events are generated, i.e.

$$|e_1^{comp}(k)| > |e_2^{comp}(k)|, |e_1^{comp}(k)| > \varepsilon, |e_2^{comp}(k)| > \varepsilon,$$

or the following events occur

$$|e_1^{comp}(k)| > \varepsilon, |e_2^{comp}(k)| < \varepsilon.$$

Define the conditional probability P_d as follows,

$$P_d \triangleq Pr(|e_1^{comp}| > |e_2^{comp}| \big| |e_1^{comp}| > \varepsilon, |e_2^{comp}| > \varepsilon),$$

where the dependence of the errors on time k is omitted for notational convenience. It is worthwhile to point out that for random scheduler case, when both errors exceed the predefined threshold, the probability of the network to transmit information for $\mathcal{S}_1$ is actually

$$P_\alpha Pr(|e_1^{comp}| > \varepsilon, |e_2^{comp}| > \varepsilon).$$

For the dynamic case, unlike P_α defined above, P_d depends on the magnitudes of the errors of both plants and hence the interference between the plants and the shared medium becomes more complicated. P_d can be exactly evaluated through Gaussian integrals because the errors are Gaussian random variables as defined in (3). However, for simplicity, we can use

$$\lambda \triangleq Pr(|e_1^{comp}| > |e_2^{comp}|)$$

as an approximation of the conditional probability P_d. In fact, we have $\lambda = 1/2$ based on the following arguments. From the fact that

$$
\begin{aligned}
& Pr(|e_1^{comp}| > |e_2^{comp}|) = Pr(|e_1^{comp}|^2 > |e_2^{comp}|^2) \\
& = Pr(e_1^{comp} + e_2^{comp} > 0, e_1^{comp} - e_2^{comp} > 0) \\
& + Pr(e_1^{comp} + e_2^{comp} < 0, e_1^{comp} - e_2^{comp} < 0) \\
& \overset{(e)}{=} Pr(e_1^{comp} + e_2^{comp} > 0) Pr(e_1^{comp} - e_2^{comp} > 0) \\
& + Pr(e_1^{comp} + e_2^{comp} < 0) Pr(e_1^{comp} - e_2^{comp} < 0),
\end{aligned}
$$

and (e) holds because $e_1^{comp} + e_2^{comp}$ and $e_1^{comp} - e_2^{comp}$ are Gaussian random variables and mutually independent. Since e_1^{comp} and e_2^{comp} are zero mean, we have

$$
\begin{aligned}
& Pr(e_1^{comp} + e_2^{comp} < 0) = 1/2, \\
& Pr(e_1^{comp} - e_2^{comp} < 0) = 1/2.
\end{aligned}
$$

This yields the desired result. Therefore, the communication rate can be calculated as a special case of random access by setting $P_\alpha = \lambda = 1/2$. The results given by this approximation match the Monte Carlo experiments very closely as demonstrated in Section VII.

Remark 6: $\lambda \neq 1/2$ for $N \geq 2$, although λ can be evaluated through Gaussian integrals for the general case.

Remark 7: The error covariance is different from random scheduler case (with $P_\alpha = 1/2$) since for dynamic scheduler there exists additional condition on the magnitudes of e_1^{comp} and e_2^{comp}. However, the error covariance can be evaluated through Gaussian integrals as discussed in Section V-D.

VI. Discussions on Special Cases

In this section, we provide analytical results for special cases of event and time triggered algorithms. When $\varepsilon = 0$, the local event (2) for each plant is generated at every time step. This implies both plants intend to access the network simultaneously at each time step. The network is thus utilized at every time step and the decision to transmit packets from which plant is based on the scheduling policies. It is worthwhile to point out that the following analysis can be used as an approximation for $\varepsilon \gtrsim 0$.

A. Static scheduler for $\varepsilon = 0$

The network transmits information for $\mathcal{S}_1$ with higher priority until the *hard* constraint for maximum tolerable delay of $\mathcal{S}_2$ is triggered. The Markov model will be reduced to $T_e + 1$ states with all transition probabilities 1 as shown in Fig 5. The following result is immediate.

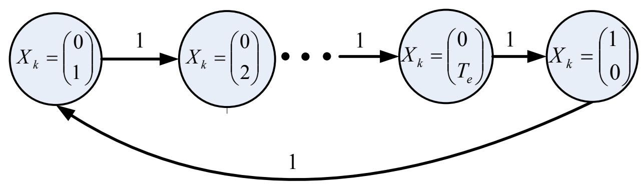

Fig. 5. Static Scheduler when $\varepsilon = 0$.

Lemma 3: Consider the event triggered algorithm with static scheduler for $\varepsilon = 0$. The average communication rate for $\mathcal{S}_1$ and $\mathcal{S}_2$ are given by

$$P_1 = \frac{T_e}{T_e + 1}, P_2 = \frac{1}{T_e + 1},$$

respectively. Furthermore, the steady state average error covariance for $\mathcal{S}_1$ and $\mathcal{S}_2$ are given by

$$\Pi_1 = \frac{1}{T_e + 1} R_{w_1}, \Pi_2 = \frac{1}{T_e + 1} \sum_{i=1}^{T_e} \sum_{j=0}^{i-1} {A_2}^j R_{w_2} (A_2^T)^j.$$

Proof: From the structure of the Markov chain shown in Fig 5, the probabilities of all modes are identical. From the balance equation that the sum of the probabilities of all modes is equal to 1, the probability for each mode is $1/(T_e + 1)$. Therefore, the communication rate for $\mathcal{S}_i$ is given by

$$P_1 = \sum_{j=1}^{T_e} Pr(X(k) = [0; j]) = \frac{T_e}{T_e + 1},$$
$$P_2 = 1 - P_1 = \frac{1}{T_e + 1}.$$

The estimation error for $\mathcal{S}_2$ under the mode $X(k) = [0, j]$ for $1 \leq j \leq T_e$ is given by Z_j. Thus the error covariance for $\mathcal{S}_2$ can be calculated as

$$\begin{aligned}\Pi_2 &= \sum_{j=1}^{T_e} Pr(X(k) = [0; j]) \mathbb{E}\{Z_{j-1} {Z_{j-1}}^T\} \\ &= \sum_{j=1}^{T_e} \frac{1}{T_e + 1} \sum_{i=0}^{j-1} A_2^i R_{w_2} {A_2^T}^i.\end{aligned}$$

The error covariance for $\mathcal{S}_1$ is simply $\frac{1}{T_e+1} R_{w_1}$. ■

B. Random scheduler for $\varepsilon = 0$

Since both local events are generated, the network transmits the packets from $\mathcal{S}_1$ with probability P_α and transmits the packets from $\mathcal{S}_2$ with probability $1 - P_\alpha$. The Markov model is presented in Fig. 6. The communication rate and error covariance can thus be obtained along the lines of Lemma 3 through the following result.

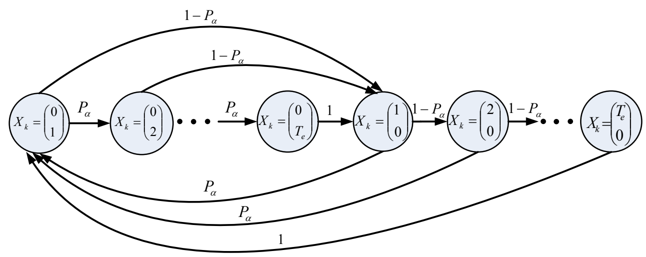

Fig. 6. Random Scheduler when $\varepsilon = 0$.

Lemma 4: Consider the event triggered algorithm with random scheduler for $\varepsilon = 0$. The average communication rate for $\mathcal{S}_1$ and $\mathcal{S}_2$ are given by

$$P_1 = \frac{(1 - P_\alpha^{T_e})}{1 - P_\alpha}\rho, P_2 = \frac{1 - (1 - P_\alpha)^{T_e}}{P_\alpha}\rho,$$

where $\rho = \frac{P_\alpha(1-P_\alpha)}{1-P_\alpha^{T_e+1}-(1-P_\alpha)^{T_e+1}}$. Furthermore, the steady state average error covariance for $\mathcal{S}_1$ and $\mathcal{S}_2$ are given by

$$\Pi_1 = \frac{\rho}{1 - P_\alpha}\sum_{i=1}^{T_e}(1 - P_\alpha)^i \sum_{j=0}^{i-1} A_1{}^j R_{w_1} (A_1^T)^j,$$

$$\Pi_2 = \frac{\rho}{P_\alpha}\sum_{i=1}^{T_e} P_\alpha^i \sum_{j=0}^{i-1} A_2{}^j R_{w_2} (A_2^T)^j.$$

C. Time triggered algorithm

In this section, we evaluate the performance by using time trigger with TDMA. Since we do not consider the cost of using the network, we assume the network transmits information at every time step.

For $N = 2$, there exist two possible schedules: $S_1 = \{1, 2, 1, 2 \cdots\}$ and $S_2 = \{2, 1, 2, 1 \cdots\}$. If $A_1 = A_2$ and $R_{w_1} = R_{w_2}$, we are going to show that the two round robin schedules S_1 and S_2 are both optimal. Otherwise, one can find an optimal schedule by evaluating the cost function for every possible schedule [2]. Assume $\hat{x}_i^{dec}(0) = 0$ for $i = 1, 2$. With schedule S_1, at the ith estimator, we have for any

$0 \leq j \in \mathbb{Z}$ and for any $k \geq 1$,

$$\hat{x}_i^{dec}(k) = \begin{cases} x_i(k), & \text{if } k = 2j + i, \\ A_i \hat{x}_i^{dec}(k-1), & \text{otherwise.} \end{cases}$$

It is easy to verify that for any $k \geq 2$,

$$\hat{x}_i^{dec}(k) = \begin{cases} x_i(k), & \text{if } k = 2j + i, \\ A_i x_i(k-1), & \text{otherwise,} \end{cases}$$

and the estimation error evolves as

$$e_i^{dec}(k) = \begin{cases} 0, & \text{if } k = 2j + i, \\ w_i(k-1), & \text{otherwise.} \end{cases}$$

Therefore, we can obtain for schedule S_1 and $k \geq 2$,

$$\sum_{i=1}^{2} \lim_{t \to \infty} \frac{1}{t} \sum_{k=2}^{t} \mathbb{E}[e_i^{dec}(k)(e_i^{dec}(k))^T] = \frac{1}{2}(R_{w_2} + R_{w_1}).$$

The system performance can be calculated as

$$\begin{aligned} J &= \lim_{t \to \infty} \frac{1}{t} \left(\sum_{i=1}^{2} \sum_{k=2}^{t} \mathbb{E}[e_i^{dec}(k)(e_i^{dec}(k))^T] + \nu_3 \right) \\ &= \frac{1}{2}(R_{w_2} + R_{w_1}), \end{aligned}$$

where $\nu_3 > 0$ is finite and depends on the values of A_2, R_2. For schedule S_2, we will obtain the same system performance along the above lines. Therefore, for $N = 2$, both schedules are optimal with the minimum cost given by

$$\Omega_T = \frac{1}{2}(R_{w_2} + R_{w_1}).$$

D. Comparison of special cases

Now, we are ready to state the main result in this section, which compares performance of time trigger and event trigger with various MA schemes. Note that the system performance is continuous with respect to the threshold ε and thus the previous results for $\varepsilon = 0$ can be considered as a good approximation for small positive ε.

Theorem 7: For $\varepsilon \gtrsim 0$, denote system performance as $\Omega_T, \Omega_S, \Omega_R$ for time trigger and event trigger with static scheduler and random scheduler, respectively. We have the following result: $\Omega_T < \Omega_S, \Omega_T < \Omega_R$.

Proof: When the network is fully utilized, for time trigger with TDMA, the system performance is given by

$$\Omega_T = \frac{1}{2}(R_{w_1} + R_{w_2}),$$

which is independent of the system dynamics A_1, A_2.

For event trigger with static scheduler, a lower bound for system performance Ω_S is given by

$$\Omega_S > \frac{1}{T_e + 1}\left[R_{w_1} + T_e R_{w_2}\right] + \nu_1.$$

The gap between two sides of the above inequality ($\nu_1 > 0$) depends on the dynamic of $\mathcal{S}_2$, or A_2. For instance, with $T_e = 2$, the gap is characterized by $\frac{A_2 R_{w_2} A_2^T}{(T_e+1)}$. For simplicity, we assume $R_{w_1} = R_{w_2}$, thus we obtain

$$\Omega_S > \Omega_T + \nu_1 > \Omega_T.$$

For event trigger with random scheduler, a lower bound of system performance Ω_R is given by

$$\Omega_R > \rho \sum_{i=1}^{T_e} \left[(1 - P_\alpha)^{i-1} R_{w_1} + P_\alpha^{i-1} R_{w_2}\right] + \nu_2,$$

The gap between two sides of the above inequality ($\nu_2 > 0$) depends on the the dynamics of both plants, or A_1, A_2. Again, for simplicity, we assume $R_{w_1} = R_{w_2}$, then we have

$$\begin{aligned} \Omega_R &> \sum_{i=1}^{T_e} \rho\left[(1 - P_\alpha)^{i-1} + P_\alpha^{i-1}\right] R_w + \nu_2, \\ &= \rho R_w \left(\frac{1 - (1 - P_\alpha)^{T_e}}{P_\alpha} + \frac{1 - P_\alpha^{T_e}}{1 - P_\alpha}\right) + \nu_2 \\ &= R_w + \nu_2 > \Omega_T, \end{aligned}$$

It has been shown that $\Omega_T < \Omega_S, \Omega_T < \Omega_R$. In other words, time trigger with TDMA performs better than event trigger with static and random schedulers. ■

Remark 8: For dynamic case, we can obtain similar results. However, the system performance is now evaluated through Gaussian integrals. As an example, for the transition from mode $[0; 1]$ to $[1; 0]$, we need the following formula,

$$\text{var}\{X \big| |X| < |Y|\} = \int_{-\infty}^{\infty} f(y)\mathrm{d}y \int_{-|y|}^{|y|} x^2 f(x)\mathrm{d}x$$

where $f(x) = N(0, R_1)$ and $f(y) = N(0, R_2)$ are normal distributions with mean 0 and variance $R_1 = R_{w_1}$ and $R_2 = (A_2^2 + 1)R_{w_2}$, respectively. This integral can be evaluated as

$$\begin{aligned} &2\int_0^\infty \int_{-y}^{y} \frac{x^2}{2\pi\sqrt{R_1 R_2}} \exp\left[-\frac{1}{2}(R_1^{-1}x^2 + R_2^{-1}y^2)\right] \mathrm{d}x\mathrm{d}y \\ &= 2\frac{R_1}{2\pi}\int_\beta^{\pi-\beta} \cos^2\theta \mathrm{d}\theta \int_0^\infty \exp\left(-\frac{1}{2}r^2\right) r^3 \mathrm{d}r \\ &= \frac{R_1}{\pi}[\pi - 2\beta - \sin(2\beta)], \end{aligned}$$

and $\beta \in [0, \pi/2]$ satisfies $r\cos\beta\sqrt{R_1} = r\sin\beta\sqrt{R_2}$. Together with the fact $Pr(|X| < |Y|) = 1/2$, this yields

$$\mathrm{var}\{X\big||X| < |Y|\} = 2\frac{R_1}{\pi}[\pi - 2\beta - \sin(2\beta)].$$

VII. Simulation Results

A. Static Scheduler

We consider the system model provided in Example 1 with $A_1 = 0.8$ and $A_2 = 0.5$. We set automatic transmission happens after $T_e = 2$ time steps. For simplicity, we set the triggering level $\varepsilon_1 = \varepsilon_2 = \varepsilon$. For various values of ε from 0 to 4, we evaluated system performance for static scheduler as predicted by Theorems 5 and 6. We compared the analytic results to Monte Carlo simulations of the system. For each value of ε, we conducted 20,000 simulations and obtained the mean communication rate and error covariance. The comparison is shown in Fig. 7 for the communication rate in the top plot and in the bottom one for the error covariance. It can be seen that the analytic results match the Monte Carlo simulations very closely.

From the bottom plot in Fig. 7, we can see that for $\varepsilon \in [0.2, 1.2]$, the error covariance for event trigger is less than time trigger; however, for other values of $\varepsilon \in [0, 4]$, time triggered algorithm performs better. This implies that there is a probability of 75% for event-triggered algorithm to perform worse than time-triggered algorithm if we choose the threshold randomly. Even for the same communication frequency, event-triggered algorithm may also perform worse as shown in section II-B.

B. Random and Dynamic Schedulers

For event trigger with random and dynamic schedulers, we verified our results as predicted in Section V-C and Section V-D. The results for communication rates for each plant are shown in Fig. 8 for random scheduler by setting $P_\alpha = 0.7$ and for dynamic scheduler in Fig. 9 by setting $P_\alpha = 0.5$, respectively. For

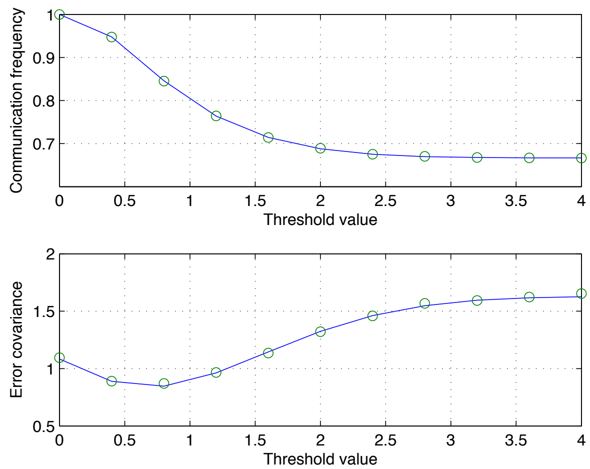

Fig. 7. Performance metrics for the NCS obtained from derived analytic expressions and Monte Carlo simulations.

the random case, we verified the results for other values of P_α as well. It can be seen that the results obtained from approximate models for both cases match the Monte Carlo simulations very closely.

The system performance by using approximate models for random scheduler in terms of the communication rate and the error covariance is provided in Fig. 10. Although the communicate rates match the simulations very closely, there is a small gap between the approximate error covariance and the simulation results. This basically tells us ignoring the past states of the Markov modes can provide us very good approximations but cannot yield the exact results.

VIII. Final Remarks

This paper studies state estimation for a NCS with multiple plants over a shared communication network. Each plant transmits information through the common network according to a time-triggered or an event-triggered rule. For a time-triggered algorithm combined with TDMA, each plant uses the network according to an off-line scheduling. For an event-triggered algorithm with CSMA, each plant is assumed to access the network based on one of the following scheduling strategies: static, random or dynamic schedulers. Performance in terms of the communication rate and estimation error covariance is analytically characterized for some special cases. Our results demonstrate that event-triggered schemes may preform worse than time-triggered schemes when considering the effect of communication strategies.

This work examines the interaction between the control world and the communication world. We consider a general system model which is not restricted to scalar integrator dynamics. Moreover, we use

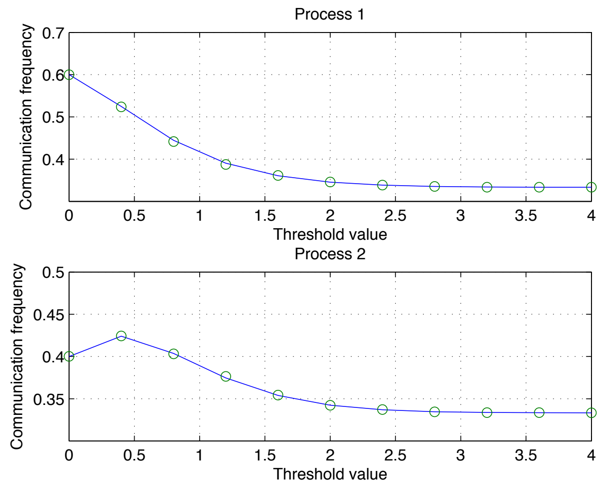

Fig. 8. Communication rates for each plant using random scheduler.

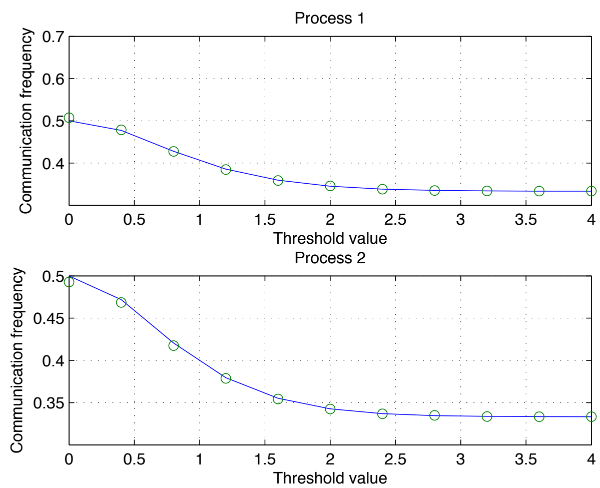

Fig. 9. Communication rates for each plant using dynamic scheduler.

CSMA protocols to model the communication network, which provides a collision resolution mechanism. We investigate the system performance more thoroughly, in terms of both the communication rate and the estimation error covariance. Different to the previous results, we show time trigger may outperform event trigger through both numerical examples and analytical results for some special cases. For future works, we need a more accurate model to analyze general cases (such as for $T_e > 2$). It is also interesting to find an optimal triggering level for various scheduling policies. Another extension is to consider a control

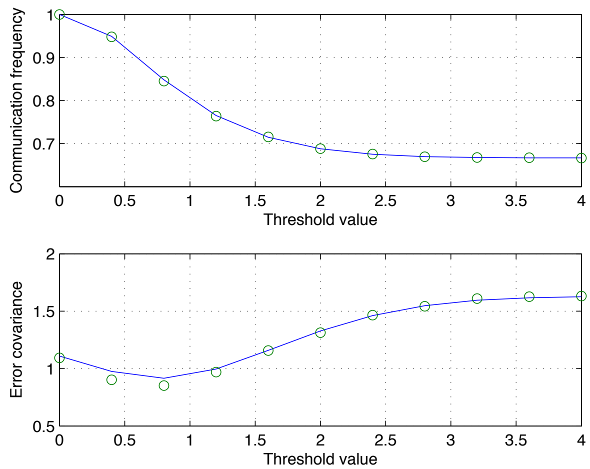

Fig. 10. Performance metrics for the NCS using random scheduler obtained from approximate expressions and Monte Carlo simulations.

setting where the control input is updated using an event triggered rule and consider other performance metrics (such as LQG). From a design point of view, it is also interesting to design an optimal scheduling strategy for given control tasks.

IX. Acknowledgment

The support of the National Science Foundation under Grant No. CNS-1035655 and 0846331 is gratefully acknowledged.

References

[1] K. J. Astrom and B. M. Bernhardsson, "Comparison of riemann and lebesgue sampling for first order stochastic systems." Proceedings of the IEEE Conference on Decision and Control, 2002.

[2] A. Cervin and T. Henningsson, "Scheduling of event-triggered controllers on a shared network," in *47th IEEE Conference on Decision and Control*, Dec. 2008, pp. 3601 –3606.

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

[4] H. Yu and P. Antsaklis, "Event-triggered real-time scheduling for stabilization of passive and output feedback passive systems," in *American Control Conference (ACC)*, 2011, pp. 1674 –1679.

[5] M. Lemmon, "Event-triggered feedback in control, esitmation, and optimization in networked control systems," *Volume 405 Lecture Notes in Control and Information Sciences*, 2010.

[6] M. Rabi, G. Moustakides, and J. Baras, "Multiple sampling for estimation on a finite horizon," in *45th IEEE Conference on Decision and Control*, Dec. 2006, pp. 1351 –1357.

[7] L. Li, M. Lemmon, and X. Wang, "Event-triggered state estimation in vector linear processes," in *American Control Conference (ACC)*, July 2010, pp. 2138 –2143.

[8] Y. Xu and J. Hespanha, "Optimal communication logics in networked control systems," in *43rd IEEE Conference on Decision and Control*, Dec. 2004, pp. 3527 – 3532.

[9] O. Imer and T. Basar, "Optimal estimation with limited measurements," in *44th IEEE Conference on Decision and Control and European Control Conference*, Dec 2005, pp. 1029 – 1034.

[10] S. Ling, K. H. Johansson, and Q. Li, "Time and event-based sensor shceduling for networks with limited communication resources," in *18th IFAC World Congress, Milan, Italy*, 2011.

[11] C. Ramesh, H. Sandberg, and K. H. Johansson, "Steady state performance analysis of multiple state-based schedulers with csma," in *50th IEEE Conference on Decision and Control (CDC)*, Dec. 2011.

[12] F. Gebali, *Analysis of computer and communication networks*. Springer- Computers, 2008.

[13] G. Walsh, H. Ye, and L. Bushnell, "Stability analysis of networked control systems," *IEEE Transactions on Control Systems Technology*, vol. 10, no. 3, pp. 438 –446, May 2002.

[14] M. Rabi and K. Johansson, "Scheduling packets for event-triggered control," in *Proceedings of the European Control Conference*, Aug. 2009, pp. 3779–3784.

[15] T. Henningsson and A. Cervin, "A simple model for the interference between event-based control loops using a shared medium," in *49th IEEE Conference on Decision and Control (CDC)*, dec. 2010, pp. 3240 –3245.

[16] R. Blind and A. Frank, "Analysis of networked event-based control with a shared communication medium: Part 1 - pure aloha," in *Proceedings of the 18th World Congress*, Aug. 2011, pp. 10 092 –10 097.

[17] G. Bianchi, "Performance analysis of the ieee 802.11 distributed coordination function," *IEEE Journal on Selected Areas in Communications*, vol. 18, no. 3, pp. 535 –547, Mar. 2000.

[18] M. Xia, V. Gupta, and P. Antsaklis, "Performance expressions for event-triggerd state estimation." *submitted to* Proceedings of the IEEE Conference on Decision and Control, 2012.

[19] B. Manjunath and S. Wilhelm, "Moments calculation for the double truncated multivariate normal density," *SSRN*, 2009.

[20] J. Song, S. Han, A. Mok, D. Chen, M. Lucas, and M. Nixon, "Wirelesshart: Applying wireless technology in real-time industrial process control," in *Real-Time and Embedded Technology and Applications Symposium, RTAS IEEE*, April 2008, pp. 377 –386.

[21] V. Gupta, A. Dana, J. Hespanha, R. Murray, and B. Hassibi, "Data transmission over networks for estimation and control," *IEEE Transactions on Automatic Control*, vol. 54, no. 8, pp. 1807 –1819, Aug. 2009.