2011 50th IEEE Conference on Decision and Control and
European Control Conference (CDC-ECC)
Orlando, FL, USA, December 12-15, 2011

Model-Based Event-Triggered Control with Time-Varying Network Delays.

Eloy Garcia and Panos J. Antsaklis, *Fellow, IEEE*

Abstract—**In this paper two approaches for reducing communication traffic in a control network, namely, Model-Based Networked Control Systems (MB-NCS) and event-triggered control, are unified under a single framework. The use of a model of the plant in the controller node not only generalizes the Zero-Order-Hold (ZOH) implementation in traditional event-triggered control schemes but it also provides stability thresholds that are robust to model uncertainties. With respect to MB-NCS, the stability conditions presented here do not need explicit knowledge of the plant parameters as in previous work but are given only in terms of the parameters of the nominal model and some bounds in the model uncertainties. The resulting framework is capable of increasing the update time intervals compared with the individual approaches considered in this paper.**

I. Introduction

In Networked Control Systems (NCS) a digital communication network is used to transfer information among the components of a control system. NCS can also help to improve efficiency, flexibility, and reliability of the network interconnected system reducing reconfiguration and maintenance costs [1]. In contrast, the protocols used to establish an ordered communication between nodes and the number of control systems and different applications that share the communication network introduce time delays and loss of information. These situations force us to revaluate the tools that are commonly used in control design in order to account for limited feedback information in the analysis and design of NCS compared to traditional point-to-point control systems. Extensive research has been done in the area of NCS as described in [2] and references therein. Reducing the amount of communication between sensor and controller nodes without compromising the stability of the control system has been a topic of many papers. In particular, Walsh, *et al.* [3] introduced a network control protocol Try-Once-Discard (TOD) to allocate network resources to the different nodes in a Networked Control System, all of them may access the network at any time assuming each access occurs before the Maximum Allowable Transfer Time (MATI). The work in [4]-[5] uses more efficiently the packet structure, that is, reduction on communication is obtained by sending packets of information using all data bits available (excluding overhead) in the structure of the packet.

Both authors are with the Department of Electrical Engineering, University of Notre Dame, Notre Dame, IN 46556 USA (e-mail: egarcia7@nd.edu). Financial support by the National Science Foundation under Grants CCF-0819865 and CNS-1035655 is gratefully acknowledged.

In event-triggered broadcasting [6]-[11] a subsystem sends its local state to the network to reduce communication rate only when it is necessary, that is, when a measure of the local subsystem state error is above a specified threshold. Event-triggered control schemes offer a new point of view, with respect to conventional time-driven strategies, on how information could be sampled for control purposes. Tabuada [6] showed the stabilizing properties of the event-triggered control strategy; he presented a triggering condition based on the norms of the state and the state error $e = x(t_i) - x(t)$, that is, the last measured state minus the current state of the system. This means that the measurement received in the controller node is held constant until a new measurement arrives; when this happens, the error is set equal to zero and starts growing until it triggers a new execution or measurement update. Wang and Lemmon [7] presented a new method to design stabilizing controllers based on the event-triggered control strategy by noting that the closed loop system Lyapunov function V needs not to be monotone decreasing for all time but an appropriate subsequence of V needs to be.

The paper is organized as follows: section II provides brief background on the MB-NCS framework. Sections III and IV provide the main results of the paper. Conditions for stability of MB-NCS using event-triggered control are presented in section III, and the network induced delay case is studied in section IV. An example is given in section V and conclusions are presented in section VI.

II. Background on MB-NCS

Model-Based Networked Control Systems (MB-NCS) were introduced by Montestruque and Antsaklis [12]-[13]; this configuration makes use of an explicit model of the plant which is added to the controller node to compute the control input based on the state of the model rather than on the plant state. This approach aims at reducing the rate at which feedback information is sent from sensor to controller. Fig. 1 shows a basic MB-NCS configuration where the network channel is implemented only in the sensor-controller side while the controller is connected directly to the plant. For the system in Fig.1 the dynamics of the plant and the model can be described respectively by:

$$\dot{x} = Ax + Bu \qquad (1)$$

$$\dot{\hat{x}} = \hat{A}\hat{x} + \hat{B}u \qquad (2)$$

where $x, \hat{x} \in \mathbb{R}^n$, $u = K\hat{x}$, and the matrices $\hat{A}, \hat{B}$ represent the available model of the system matrices A,B. The plant may be unstable i.e. not all eigenvalues of A have negative real parts. The same authors provided necessary and sufficient conditions for stability when the updates from the system are periodic (every h time units) [14]. In this paper we discard the periodicity assumption for updating the model, instead we embrace a nonperiodic approach that is based on events; we use the estimate of the state given by the model of the plant to compare it with the actual state, and then the sensor transmits the state of the plant if the error is above some predefined tolerance. In this way the update time will be variable instead of fixed.

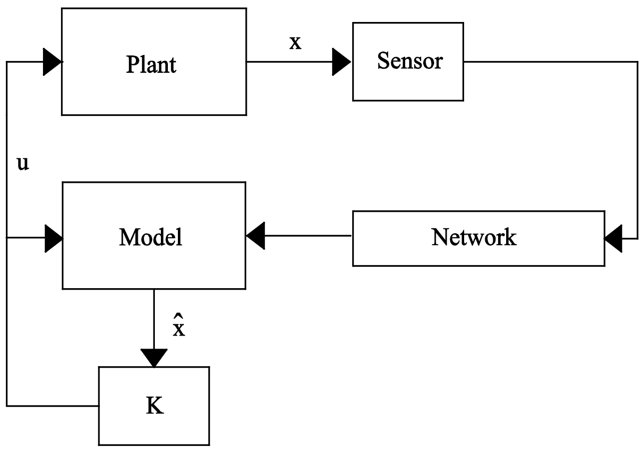

Fig. 1. Representation of a Model-Based Networked Control System.

This approach increases the time intervals that we use to update the model with respect to MB-NCS with periodic sampling by selecting the stabilizing threshold. By increasing the update interval (reducing communication rate) we release the network for other uses. In case we have several control systems implemented over the network, by reducing network traffic, we are also reducing the size of time delays and reducing the probability of packets being lost. In addition, the conditions to select a stabilizing threshold are given in terms of the nominal model parameters and bounds on the model uncertainties, assuming the dimension of the system is known.

With respect to previous work in event-triggered control, the implementation of this strategy using MB-NCS accounts for the unavoidable existence of model uncertainties in the stability analysis. This problem has not been dealt with previously within the event-triggered approach and as it is shown it affects directly the estimated threshold values that aim to ensure stability of the system. Additionally, we use the combined model-based event-triggered control method for stabilization of systems with network delays. The work in [15] presents a configuration that stabilizes a NCS with large constant delays using passivity and the scattering transformation. The works in [16] and [17] derive general models of NCSs that consider time-varying sampling intervals and delays. Although the admissible delays may be greater than the ones derived here, the authors of those papers do not consider model uncertainties which account for a conservative allowable delay in our work. In contrast, we are able to provide robustness to parameter uncertainties in the presence of time-varying delays.

III. Stabilizing Model-Based Event-Triggered Strategies

A. A fixed threshold strategy.

The work on event-triggered control presented in [6]-[11] considers only a zero-order-hold as a model and the main purpose here is to generalize this work using MB-NCS. We will assume in this section that the communication delay is negligible and the initial conditions of the plant are nonzero but finite.

In this scheme the sensor has different functions to perform; the sensor contains a copy of the model and the controller gain so it can have access to the model state. It continuously measures the actual state and computes the state error, defined by $e = \hat{x} - x$, it compares the norm of the error to a predefined threshold α, and it broadcasts the plant state to update the model state if the error is greater than the threshold.

It is clear that while $|e| \le \alpha$ the plant is running open loop based on the input generated by the model state $\hat{x}$. After substituting the input $u = K\hat{x}$ in (1) and using the definition of the error we can write:

$$\dot{x} = (A + BK)x + BKe \tag{3}$$

In the case of the model, after substituting the input u we have a state space system of the form:

$$\dot{\hat{x}} = (\hat{A} + \hat{B}K)\hat{x} \qquad for \qquad t_i \le t < t_{i+1} \tag{4}$$

At the update times t_i, $i=0,1,2...$ the state of the model is updated with the measurement obtained from the plant; the update times are non-periodic in general and are triggered by the size of the state error.

Theorem 1. For $|x(0)| \le \beta_1$, $0 < \beta_1 < \infty$ the system described by (3) with state feedback based on error events is bounded-input bounded-state stable with respect to the measurement error if the eigenvalues of $A+BK$ are in the left hand side of the complex plane.

Proof. Note that by considering $y=x$, then (y,e) is BIBO stable when $A+BK$ is asymptotically stable. If A,B is controllable then the relation is if and only if. Then we need to ensure that the error is bounded by updating the model when $|e| \le \alpha$ is not satisfied. An extended form of the proof is obtained by directly finding a bound on the norm of the response of the system but it is omitted here for brevity. ■

B. A relative threshold strategy.

In many different applications it is desirable to asymptotically stabilize a system. It is intuitive that by varying the magnitude of the threshold value we can obtain longer update intervals or a smaller output size. The idea of reducing the threshold value as we approach the equilibrium point of the system is logical; the work in [6] follows this approach by comparing the norm of the state error to a function of the norm of the state of the plant; in this way, the threshold value is not fixed anymore, and, in particular, it can be reduced as we approach the equilibrium point of the system, assuming that the zero state is the equilibrium of the system. Previous work on event-triggered control dealt with systems controlled by static gains that generate piecewise constant inputs due to the fact that the update is held constant in the controller. The main difference in this section is that we use a Model-Based controller i.e. a model of the system and a static gain; the model provides an estimate of the state between updates and the model/gain controller provides an input for the plant that does not remain constant between measurement updates.

Consider again the plant and model described by (1) and (2) and by using the control input $u = K\hat{x}$ we obtain the description (3) for the plant. Assume that the control input *u* renders the system (3) Input-to-state stable (ISS) with respect to the measurement error *e*. For the definition of ISS we use the next [11]:

Definition 2. A smooth function $V:\mathbb{R}^n \to \mathbb{R}_0^+$ is said to be an ISS Lyapunov function for the dynamical system $\dot{x} = f(x,u)$, $x(t)\in\mathbb{R}^n, u(t)\in\mathbb{R}^m, t\in\mathbb{R}_0^+$ if there exist class $\mathcal{K}_\infty$ functions $\underline{\alpha},\overline{\alpha},\alpha$ and γ satisfying:

$$\underline{\alpha}(|x|) \le V(x) \le \overline{\alpha}(|x|) \tag{5}$$

$$\frac{\partial V}{\partial x} f(x,u) \le -\alpha(|x|) + \gamma(|u|) \tag{6}$$

The system $\dot{x} = f(x,u)$ is said to be ISS with respect to the input *u* if and only if there exists an ISS Lyapunov function for that system. ■

In our particular case, we choose a control law $u = K\hat{x}$ that renders the closed loop *model* (4) globally asymptotically stable. Any such *K* also renders the closed loop model Input-to-State Stable with respect to the measurement errors. We proceed to choose a quadratic ISS-Lyapunov function, $V = x^T P x$ where *P* is symmetric positive definite and is the solution of the closed loop model Lyapunov function:

$$(\hat{A} + \hat{B}K)^T P + P(\hat{A} + \hat{B}K) = -Q \tag{7}$$

where *Q* is a symmetric positive definite matrix.

Let us first analyze the case when $\hat{B} = B$ for simplicity and define the uncertainty $\tilde{A} = A - \hat{A}$, also assume that the next bound on the uncertainty $\left|\tilde{A}^T P + P\tilde{A}\right| \le \Delta < q$ holds where $q = \underline{\sigma}(Q)$, the smallest singular value of *Q* in the Lyapunov equation (7). This bound can be seen as a measure of how close *A* and $\hat{A}$ should be. It can be seen from (7) that the solution *P* depends on the choice of *Q*. One way to obtain a small *P* and large *q* is to make $Q = -(\hat{A} + \hat{B}K)$ and design *K* such this closed loop model matrix is very stable. Unfortunately, the predefined location of the eigenvalues of $\hat{A} + \hat{B}K$ does not ensure, in general, a particular selection of the singular values. A particular case when this can be easily achieved is when the number of inputs is equal or greater than the number of states. In such a case, we can obtain a closed loop model matrix that is diagonal with desired eigenvalues, and with the previous choice of *Q*, the solution of (7) is always $P = 0.5 * I_{n\times n}$. Since $\hat{A} + \hat{B}K$ is diagonal its singular values are equal to the absolute value of its eigenvalues. Therefore, we can easily manipulate *q* while *P* remains the same.

The next theorem provides conditions on the error and its threshold value so the networked system is asymptotic stable. The error threshold is defined as a function of the norm of the state and Δ which is a bound on the uncertainty in the state matrix *A*. Similarly, the occurrence of an error event leads the sensor to send the current measurement of the state of the plant that is used in the controller to update the state of the model.

Theorem 3. System (1) with $u = K\hat{x}$ and feedback based on error events generated when the relation:

$$|e| \le \frac{\sigma(q-\Delta)}{b}|x| \tag{8}$$

is not satisfied, is globally asymptotically stable, where $b = \left|K^T \hat{B}^T P + P\hat{B}K\right|$ and $0 < \sigma < 1$.

Proof. In order to prove this theorem we will set a bound on the derivative of $V = x^T P x$ along the trajectories of the system (3) which is equal to (1) when the input $u = K\hat{x}$ has already been substituted and expressed in terms of the state error, then we can easily show that this bound can be appropriately tuned by the choice of the threshold on the error.

$$\begin{aligned}\dot{V} &= x^T[(A+BK)^T P + P(A+BK)]x + e^T K^T B^T P x + x^T PBKe \\ &= x^T[(\hat{A}+\tilde{A}+\hat{B}K)^T P + P(\hat{A}+\tilde{A}+\hat{B}K)]x + e^T K^T \hat{B}^T P x + x^T P\hat{B}Ke \\ &= -x^T Q x + x^T(\tilde{A}^T P + P\tilde{A})x + e^T K^T \hat{B}^T P x + x^T P\hat{B}Ke\end{aligned}$$

We have just expressed $\dot{V}$ in terms of the model parameters and the uncertainty of the state matrix *A*. We now proceed to evaluate the contributions of each, the model, the uncertainty, and the error.

$$\begin{aligned}\dot{V} &\le -q|x|^2 + \left|\tilde{A}^T P + P\tilde{A}\right||x|^2 + \left|K^T \hat{B}^T P + P\hat{B}K\right||e||x| \\ &\le (-q+\Delta)|x|^2 + b|e||x|\end{aligned}$$

Now, by restricting the error to satisfy (8) we can finally write:

$$\dot{V} \le (\sigma - 1)(q-\Delta)|x|^2 \tag{9}$$

Then *V* is guaranteed to decrease for any σ such $0 < \sigma < 1$ and updating the state of the model every time the error does not satisfy the condition imposed in (8). ■

Remark 1. In comparison to usual strategies in MB-NCS, an important advantage of this approach is that we define the controller in terms of the model and some bound on the uncertainty, quantities that we specifically know.

The extension to consider the case of $\hat{A} \ne A$ and $\hat{B} \ne B$ is straightforward by assuming that the next bounds on the uncertainty matrices hold:

$$\left|(\tilde{A} + \tilde{B}K)^T P + P(\tilde{A} + \tilde{B}K)\right| \le \Delta < q \tag{10}$$

$$\left|\tilde{B}\right| \le \beta \tag{11}$$

where $\tilde{B} = B - \hat{B}$. In order to obtain the bound (9) the error is set to satisfy (triggering an update otherwise):

$$|e| \le \frac{\sigma(q-\Delta)}{\bar{b}} |x| \tag{12}$$

where $\bar{b} = b + 2\beta |K||P|$.

IV. Systems with Time-Varying Delays

Although MB-NCS may help to reduce network induced delays we should be prepared for situations in which given peak conditions on the network considerable time delays are present when transmitting information. The solutions provided in the previous section assumed negligible time delays but it has been shown that the sole event-triggered control strategy is able to compensate for delays in a natural way: if some delay characteristics are known (a bound or even the exact time delay when using time-stamped messages) the next update should be scheduled before the regular one (the update when no delay is present) in such a way that stability is never compromised. In this section we take this approach along with the model dynamics in order to determine the best time to update in the presence of time delays. Two advantages are obtained by including the MB-NCS framework with respect to only using an event-triggered controller. The first one is the known property of generating an estimate of the state when operating in open loop mode to get longer update intervals. The second advantage is that the model is able to produce almost instantaneously an estimate of the current plant state based on the delayed measurement. We can use this estimate instead of using the delayed measurement to update the model in the controller.

When referring to the execution rule described in theorem 3 it is important to guarantee that the inter-execution update times never become too close to each other and generating an execution in order to update the model in the controller when the previous execution has not been finished due to time delays or even resulting in accumulation points. It is not an easy task to show that this will never occur since the execution time intervals are only implicitly defined by (8).

Theorem 4. Let (1) be a control system with control input based on the nominal model $u = K\hat{x}$ and assume that: there exists a symmetric positive definite solution P for the model Lyapunov equation (7), $B = \hat{B}$, and the next bounds are satisfied: $|\tilde{A}| \le \Delta_A$ and $|\tilde{A}^T P + P\tilde{A}| \le \Delta < q$; then there exists an $\varepsilon > 0$ such that for all network delays $\tau_N \in [0, \varepsilon]$ the system is asymptotically stable, furthermore, there exists a time τ such that for any initial condition the inter-execution times $\{t_{i+1} - t_i\}$ implicitly defined by (8) with $\sigma < 1$ are lower bounded by τ, i.e. $t_{i+1} - t_i \ge \tau \quad \forall i \in \mathbb{N}$.

Proof: In order to show asymptotic stability for nonzero network delay case and to bound the inter-execution times let us look at the dynamics of $|e|/|x|$:

$$\begin{aligned}
\frac{d}{dt}\frac{|e|}{|x|} &= \frac{d}{dt}\frac{(e^T e)^{1/2}}{(x^T x)^{1/2}} = \frac{(x^T x)^{1/2}(e^T e)^{-1/2} e^T \dot{e} - (e^T e)^{1/2}(x^T x)^{-1/2} x^T \dot{x}}{x^T x} \\
&= \frac{e^T \dot{e}}{(x^T x)^{1/2}(e^T e)^{1/2}} - \frac{(e^T e)^{1/2} x^T \dot{x}}{(x^T x)^{3/2}} \\
&= \frac{e^T(\hat{A}e - \tilde{A}x)}{|x||e|} - \frac{x^T[(\hat{A} + \tilde{A} + BK)x + BKe]}{|x||x|}\frac{|e|}{|x|} \\
&\le |\tilde{A}| + |\hat{A}|\frac{|e|}{|x|} + |\hat{A} + \tilde{A} + BK|\frac{|e|}{|x|} + |BK|\left(\frac{|e|}{|x|}\right)^2 \\
&\le \Delta_A + (\Delta_A + |2\hat{A} + BK|)\frac{|e|}{|x|} + |BK|\left(\frac{|e|}{|x|}\right)^2
\end{aligned} \tag{13}$$

Let us denote the term $|e|/|x|$ by θ so we have the estimate:

$$\begin{aligned}
\dot{\theta} &\le \Delta_A + (\Delta_A + |2\hat{A} + BK|)\theta + |BK|\theta^2 \\
&\le \Delta_A + |2\hat{A}| + (\Delta_A + |2\hat{A}| + |BK|)\theta + |BK|\theta^2
\end{aligned} \tag{14}$$

and consider the differential equation:

$$\dot{\phi} = \Delta_A + |2\hat{A}| + (\Delta_A + |2\hat{A}| + |BK|)\phi + |BK|\phi^2 \tag{15}$$

then we can conclude that $\theta(t) \le \phi(t, \phi_0)$, where $\phi(t, \phi_0)$ is the solution of (15) satisfying $\phi(0, \phi_0) = \phi_0$.

For the case when $\tau_N = 0$, the inter-execution times are bounded by the time it takes for ϕ to evolve from 0 to $\sigma(q-\Delta)/b$, i.e. the solution $\tau \in \mathbb{R}^+$ of $\phi(\tau, 0) = \sigma(q-\Delta)/b$. An estimate of that time can be obtained by solving (15), such solution is given by:

$$\phi(t,0) = \frac{-e^{dt(c-1)} + 1}{e^{dt(c-1)}/c - 1} \tag{16.a}$$

for $c \ne 1$ and let $y = \sigma(q-\Delta)/b = \phi(\tau, 0)$, then

$$\tau = (\ln(y+1) - \ln(\frac{y}{c}+1))\frac{1}{d(c-1)} \tag{16.b}$$

where $d = |BK|$ and $c = (\Delta_A + |2\hat{A}|)/d$. In the analysis if we have the case *c=1* we can easily avoid it by increasing the bound on the uncertainty by a very small amount. It can also be verified that $\tau > 0$ for any $y > 0$. Moreover, the range of values for τ for any positive value of the threshold y is given by $\tau \in [0, \tau_m)$, where:

$$\tau_m = \lim_{y \to \infty} \tau = \frac{\ln(c)}{d(c-1)} \tag{16.c}$$

For $\tau_N > 0$, we choose some σ' such the next is satisfied $\sigma < \sigma' < 1$, and let $0 < \varepsilon_1 < \tau_m$ satisfy the solution $\phi(\varepsilon_1, y) = y' = \sigma'(q-\Delta)/b$, such ε_1 always exists since ϕ is continuous in the range $\tau \in [0, \tau_m)$ that covers all positive thresholds $0 < y, y' < \infty$, also $\dot{\phi} > 0$ and $y < y'$ since $\sigma < \sigma'$. Then, by sending the state measurement at time t_i in order to update the model in the controller, this execution is released by the condition $|e| = y|x|$, we guarantee that for $t \in [t_i, t_i + \varepsilon_1]$ we have $|e| \le y'|x|$, and since $\sigma' < 1$ asymptotic stability is still guaranteed. The inter-execution times are now bounded by $\tau_N + \tau$, where τ is the time it takes ϕ to evolve from $|e(t_i + \tau_N)| / |x(t_i + \tau_N)| = |\hat{x}(t_i + \tau_N) - x(t_i + \tau_N)| / |x(t_i + \tau_N)|$ to y, then the admissible delays τ_N need to satisfy $|e(t_i + \tau_N)| / |x(t_i + \tau_N)| < y$ since $\dot{\phi} > 0$. From continuity of $|\hat{x}(t_i + \tau_N) - x(t_i + \tau_N)| / |x(t_i + \tau_N)|$ with respect of τ_N there exists an $\varepsilon_2 > 0$ such that for any $0 \le \tau_N \le \varepsilon_2$ we have $|\hat{x}(t_i + \tau_N) - x(t_i + \tau_N)| / |x(t_i + \tau_N)| < y$. The term $|\hat{x}(t_i + \tau_N) - x(t_i + \tau_N)| / |x(t_i + \tau_N)|$ is continuous due to the fact that $|x(t_i + \tau_N)|$ is never zero since the closed loop system is asymptotically stable and never reaches zero in finite time. We complete the proof by defining $\varepsilon = \min\{\varepsilon_1, \varepsilon_2\}$. ■

The importance of the results in this section relate to the fact that we can find a positive lower bound on the inter-execution times in the presence of both time delays and model uncertainties. The estimation of the admissible delays is conservative at the present time and the search for better delay estimation methods will be studied in the future. Note also that the estimation of ε is an upper bound for the admissible time-varying delays $\tau_N > 0$, that is, the results apply the same way for any $0 < \tau_N < \varepsilon$.

Updating the model state using delayed measurements. Since we need to implement the model of the plant in both nodes, the controller and the sensor node, in order to compute the control input and compute the state error respectively, we have to use wisely the delayed information received by the controller so a good estimate of the current plant state is obtained to update the model in the controller and compute a better control input for the plant. In the case that the network delays are constant then we can implement the next strategy: the sensor decides to send a feedback measurement to the controller at time t_i so it updates its own state but keeps using the old input, i.e. the input generated by the same model in the case that no update has taken place, similar to the plant being fed by the model/controller that has not been updated yet. Notice that the sensor knows the magnitude of the constant network delay then it will switch to closed loop mode at the end of the known delay. By using this strategy we need to implement a second model in the sensor node, but this is physically possible since we are talking about operations performed by a single processor, that is, if we are able to implement the computations needed to measure and compute the state error and threshold comparisons then, in general, we could be able to implement a second closed loop model that only works for short intervals $[t_i, t_i + \tau_N]$. When the controller receives the measurement $x(t_i)$ at time $t_i + \tau_N$ it uses this measurement to immediately estimate the state of the model in the sensor by computing the next:

$$\hat{x}_c(t_i + \tau_N) = e^{\hat{A}\tau_N} x(t_i) + \int_0^{\tau} e^{\hat{A}(\tau - s)} B u_c(s) ds \qquad (17)$$

which can be made arbitrarily accurate by storing the sequence of inputs over the previous delay interval, i.e. $[t_i, t_i + \tau_N]$ in the controller node and since the parameters in both models are exactly the same. The subscript c emphasizes the quantities belonging or available in the controller node. The result of the operation in (17) is used to update the state of the model in the controller.

A more general situation in many networked systems is that the network induced delays are time-varying and bounded. In this case the sensor is unable to know the magnitude of the delay but by time-stamping the measurement sent over the network the controller node does know the size of the delay for every packet containing a feedback measurement. A simple strategy in this case is to let the model in the sensor to remain working in closed loop after measuring and updating its state. When the controller receives the delayed measurement it simply computes the following:

$$\hat{x}_c(t_i + \tau_N) = e^{(\hat{A} + BK)\tau_N} x(t_i) \qquad (18)$$

which is used to update the model in the controller node.

A slightly different strategy can also be implemented in this case that, in general, results in a better performance, i.e. longer broadcast intervals, by realizing that the states of both models do not need to be the same, as long as the model in the controller produces a smaller state error than the model in the sensor. This is basically a combination of the two strategies above. The sensor updates its state and continues working in closed loop mode but the controller uses the quantity obtained by (17) in order to obtain a better estimate of the current plant state not of the current sensor model state based on the delayed measurement.

V. Example

Consider the following networked system implemented as in Fig. 1, where the system to be controlled is unstable and is represented by the unknown parameters:

$$A = \begin{bmatrix} 0.55 & -0.4 \\ 0.3 & -0.7 \end{bmatrix} \quad B = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

the model represents an alteration of the physical parameters by 10%, and the controller can be found by using these model parameters:

$$\hat{A} = \begin{bmatrix} 0.495 & -0.360 \\ 0.270 & -0.630 \end{bmatrix} \quad \hat{B} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

$$K = \begin{bmatrix} -1.3268 & 0.4618 \end{bmatrix}$$

We assume that the next uncertainty bounds are given:

$\Delta = 1.05, \quad \Delta_A = 0.1$

By choosing the following parameters: q = 5 and σ = 0.5 we can find the threshold y = 0.1382, then, by using the results of previous section we get ε = 0.065 seconds.

Simulation results are shown in Fig. 2; it shows the response of the norm of the state of the plant and the norm of the error for a time-varying delay bounded by 0.06 seconds. The discrete variations on the error correspond to the events generated at the sensor node i.e. when the sensor decides to transmit the current measurement and updates its internal model, resetting the error as measured by the sensor. It can also be seen that we are able to asymptotically stabilize the system in the presence of delays and using feedback measurements sent through the network at very distant intervals of time, i.e. significantly reducing the traffic in the network.

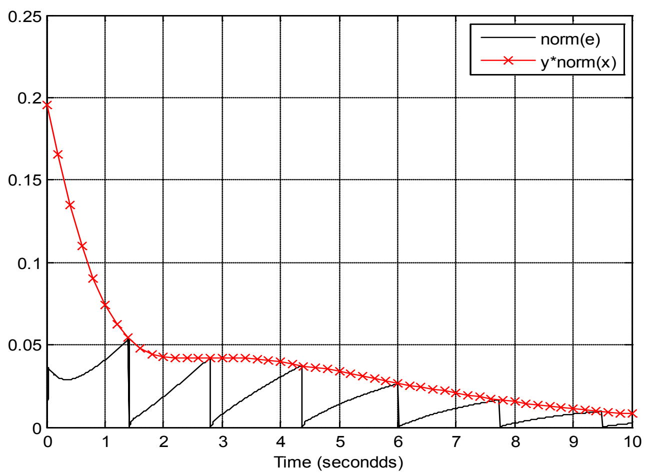

Fig. 2. Response of $\left|e(t)\right|$ and $y\left|x(t)\right|$ for time-varying bounded delays $\varepsilon = 0.06$ seconds.

In order to draw a comparison to the case when a ZOH model is used in the controller node, that is, the received measurement is held constant until a new measurement arrives we execute a similar simulation using the same parameters, controller gains, and time delays as before. Fig. 3 shows the simulation results and it can be seen that error events are triggered more frequently increasing the amount of information transmitted over the network.

VI. Conclusion

The work presented in this paper combines two different approaches commonly used in NCS. This new control strategy generalizes the traditional event-triggered control scheme. It implements a nominal model of the system that is part of the controller node in order to generate an estimate of the state of the system between update intervals, which is an improvement compared to the ZOH that generates a constant input during the same interval. The event-triggered strategy provides a different way to update the model of the system in MB-NCS without compromising stability; it increases the time interval between updates depending on the working conditions of the plant. The resulting stability conditions can be easily checked compared to those in MB-NCS with periodic updates. Future work will lift the restriction that the controller should be adjacent to the plant, considering network channels in both sides of the control loop.

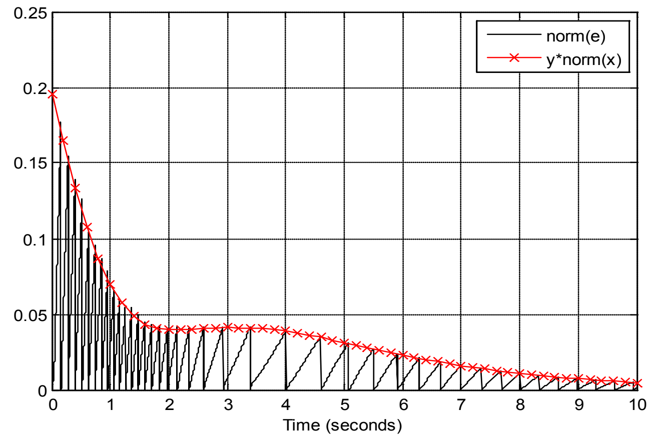

Fig. 3. Response of $\left|e(t)\right|$ and $y\left|x(t)\right|$ for the ZOH case.

References

[1] F. L. Lian, J. R. Moyne, D. M. Tilbury "Performance evaluation of control networks: Ethernet, Controlnet, and Devicenet," *IEEE Control Systems Magazine*, pp. 66 – 83, February 2001.

[2] P. J. Antsaklis and J. Baillieul, "Special issue on technology of networked control systems," *Proc. IEEE*, vol. 95, no. 1, Jan. 2007.

[3] G. C. 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.

[4] D. Georgiev, D. M. Tilbury "Packet-Based control" *Proceedings of the American Control Conference,* pp. 329 – 336, 2004.

[5] D. E. Quevedo, E.I. silva, and G.C. Goodwin, "Packetized predictive control over erasure channels," *Proc. of the American Control Conference*, 2007.

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

[7] X. Wang, M. D. Lemmon, "Event Design in Event-Triggered Feedback Control Systems," *Proceedings of the 47th IEEE Conference on Decision and Control*, 2008.

[8] P. Tabuada and X. Wang, "Preliminary results on state-triggered scheduling of stabilizing control tasks," *Proceedings of the 45th IEEE Conference on Decision and Control,* 2006.

[9] X. Wang and M. D. Lemmon, "Event triggered Broadcasting across distributed networked control systems," *Proceedings of the American Control Conference,* 2008.

[10] K. J. Astrom, B. M. Bernhardson "Comparison of Riemann and Lebesgue sampling for first order stochastic systems," *Proceedings of the 41st IEEE Conference on Decision and Control*, 2002.

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

[12] L. A. Montestruque and P. J. Antsaklis, "Model-based networked control systems: necessary and sufficient conditions for stability" *Proc. of the 10th Mediterranean Conference on Control and Automation*, 2002.

[13] L. A. Montestruque and P. J. Antsaklis. "State and output feedback control in model-based networked control systems," *Proceedings of the 41st IEEE Conference on Decision and Control*, 2002.

[14] L. A. Montestruque, P. J. Antsaklis "On the model-based control of networked systems," *Automatica*, vol. 39, no. 10, pp. 1837 - 1843, 2003.

[15] T. Matiakis, S. Hirche, and M. Buss, "Independent-of-delay stability of nonlinear networked control systems by scattering transformation," *in Proc. American Control Conference*, 2006, pp. 2801-2806.

[16] M. B. G. Cloosterman, L. Hetel, N. van de Wouw, W. P. M. H.Heemels, J. Daafouz, and H. Nijmeijer, "Controller synthesis for networked systems," *Automatica*, vol. 46, pp. 1584-1594, 2010.

[17] L. Hetel, J. Daafouz, and C. Iung, "Analysis and control of LTI and switched systems in digital loops via an even-based modeling," *International Journal of Control*, vol. 81, no.7, pp. 1125-1138, 2008.