PARAMETER IDENTIFICATION IN MODEL BASED NETWORKED CONTROL SYSTEMS USING KALMAN FILTERS

Technical Report of the ISIS Group

at the University of Notre Dame

ISIS-09-004

June, 2009

Eloy Garcia and Panos J. Antsaklis

Department of Electrical Engineering

University of Notre Dame

Notre Dame, IN 46556

Email: {egarcia7, antsaklis.1}@nd.edu

Interdisciplinary Studies in Intelligent Systems-Technical Report

Abstract.

The applicability of Model Based Networked Control Systems (MB-NCS) is often limited by the inexact knowledge of the dynamics of the system being controlled. On-line identification of system parameters is used in this paper to upgrade the model of the system, which is used to control the real system when feedback information is unavailable. Background material is offered on the topic of parameter identification with emphasis on the Recursive Least Squares algorithm. The Extended Kalman Filter (EKF) is analyzed in detail in the context of parameter identification and implemented in the Model Based Networked Control Systems (MB-NCS) framework. Simulations are included that show the efficiency of these tools.

1. Introduction.

In Networked Control Systems (NCS), dynamical systems are controlled by using feedback over a communication network. Advantages of NCS are well known, and some of them are: NCS reduce wiring, increase reliability, and improve reconfigurability of control systems [15]. At the same time, different undesired situations are encountered due to communication channel effects such as packet dropouts, time delays, and bandwidth restrictions [6], [17]. A type of NCS called Model Based Networked Control Systems (MB-NCS) aims to reduce communication over the network by incorporating an explicit model of the system to be controlled. The state of this model is used for control when no feedback is available (open loop). When the loop is closed, the state of the model is updated with new information, namely, the state of the real system. The MB-NCS framework is able to reduce network communication; consequently, the network is available for other uses, reducing time delays and bandwidth limitations.

For periodic updates of MB-NCS Montestruque and Antsaklis [12], [14] provide necessary and sufficient conditions for stability; the amount of reduction in network communication that we are able to achieve, i.e. the longer we can wait for a new update without compromising stability is directly related to the accuracy of the model; indeed, this is one of the most important limitations of this framework, namely, the absence of a sufficiently accurate model of the system. Even when an accurate model is initially available, in many applications the parameters of a system may change slowly over time due to the use and age of the physical plant or of its components.

In this paper we focus on applying identification algorithms in the MB-NCS context. Correct knowledge of the plant dynamics will provide an improvement in the control action over the network, i.e. we can achieve longer periods of time without need for feedback. At the same time, we overcome another important

limitation on MB-NCS; the usual assumption in the MB-NCS literature is that the controller is designed to stabilize the real system; this may be unrealistic since our knowledge of the plant dynamics is limited. As we will see, the identification process allows us to update not only the model but the controller itself so it can better respond to the dynamics of the real plant being controlled.

The rest of the paper is organized as follows: in section 2 background on system identification is provided along with detailed discussion of the Recursive Least Squares (RLS) algorithm, section 3 introduces the Kalman Filter for identification of parameters, in section 4 the Extended Kalman Filter (EKF) is presented. The main results in this paper are presented in section 5 and 6; the use of Kalman filters on Model Based Networked Control Systems (MB-NCS) for parameter identification is first discussed in section 5 and different implementations are shown in section 6. Finally, some conclusions are offered at the end of the paper.

2. Background material and RLS.

This section is intended to provide a very brief introduction to the broad topic of system identification; for an extended treatment see for example [11]. The recursive least squares estimation method has been chosen to receive more attention because of its simplicity and wide range application and its implementation on the Model Based Networked Control Systems framework is shown with an example in appendix A.

There exist two typical approaches for system identification namely, parametric and nonparametric [4]. A parametric model may take different forms, the most common ones are transfer functions (expressed in polynomial or poles and zeros form), state space representations, and differential equations. In these forms there exist coefficients (parameters) that specify completely the model. Nonparametric

models result generally from the data obtained from frequency response methods. In these cases the system is subject to a wide range of inputs in order to find a characteristic curve. A frequency response is difficult to obtain while the system is in normal operation, limiting the use of nonparametric approaches for on-line identification.

The focus of this paper is to identify the system parameters on-line in order to detect any changes on these parameters (abrupt changes or slow variations due to aging of the components of the physical plant). The identified parameters are used to update an explicit model of the plant and the state of this model is used for control when no feedback information from the real plant is available. An explicit model means that a parametric model is needed; in order to achieve the described goals we will follow the parametric approach in what follows.

One type of common parametric methods for identification are the gradient methods [7]; in general, gradient algorithms use a model of the form $z = \hat{\theta}\varphi$ where $\hat{\theta}$ is the estimate of θ (the unknown parameters) and z and φ are signals available for measurement. Some appropriate functional $J(\theta)$ has to be defined and minimized. Different gradient algorithms exist as the consequence of the choice of $J(\theta)$.

Model Reference Adaptive techniques are usually used for adaptive control but Landau used this approach for identification of single input-single output and multivariable systems in [10].

Recursive Least Squares algorithm.

Least Squares Estimation was initially used to estimate a constant based on a set of noisy measurements.

Let $\mathbf{x} \in \mathbb{R}^n$ be a vector of constants and $\mathbf{y} \in \mathbb{R}^m$ the vector of measurements defined by:

$$\mathbf{y} = H\mathbf{x} + \mathbf{v} \tag{1}$$

where H is a matrix of appropriate dimensions and $\mathbf{v} \in \mathbb{R}^m$ is some measurement noise. Least squares aims to solve the quadratic minimization problem: $\min \|\mathbf{y} - H\hat{\mathbf{x}}\|^2$.

The best estimate given by the Least Squares criterion is given by:

$$\hat{\mathbf{x}} = (H^T H)^{-1} H^T \mathbf{y} \tag{2}$$

It is assumed that the system (1) is overdetermined, *i.e.* there exist more measurements than unknowns; the dimensions of $H \in \mathbb{R}^{m\,x\,n}$ follow $m > n$.

This estimate (2) involves matrix inversion, which is numerically sensitive and computationally expensive when the number of measurements grows. An alternative is the Recursive Least Squares which makes use of the well known matrix inversion lemma where no matrix inversion is performed and it is recursive in nature. The next set of equations defines the RLS algorithm and its complete derivation can be found in [5].

Consider the Auto Regressive model:

$$x_k = \sum_{i=1}^{n} a_i x_{k-i} + w_k \tag{3}$$

where a_i *i=1…n* describe the unknown parameters, x_k is the *k-th* measurement and w_k is inaccessible white noise. The estimate of the unknown parameter vector is:

$$\hat{a}_r = \hat{a}_{r-1} + \mathrm{P_r}\,\mathbf{x}_r (x_r - \mathbf{x}_r{}^T \hat{a}_{r-1}) \tag{4}$$

The above equation yields the vector estimate $\hat{a}_r$ in terms of $\hat{a}_{r-1}$, i.e. in terms of the previous estimate, and in terms of a correction term which is a function of the prediction error $x_r - \mathbf{x}_r^T \hat{a}_{r-1}$ that uses the previous estimate.

The matrix $\mathrm{P_r}$ is given by:

$$\mathrm{P_r} = \mathrm{P_{r-1}} - \frac{\mathrm{P_{r-1}}\,\mathbf{x}_r \mathbf{x}_r^T\,\mathrm{P_{r-1}}}{1 + \mathbf{x}_r^T\,\mathrm{P_{r-1}}\,\mathbf{x}_r} \tag{5}$$

An example of the RLS algorithm implemented over MB-NCS is offered in appendix A.

Limitations of the Recursive Least Squares algorithm.

Least squares algorithm is an estimation method based on the input and output measurements of the system. Two of the limitations of this scheme are: First, the input to the system needs to excite all its dynamics, then, some input signals may not be useful for identification using least squares including zero-signal, for this purpose we need a persistently exciting signal [4], [7]. Second, least squares estimation is able to identify all parameters that uniquely characterize a system, for example the coefficients of the system transfer function, on the other hand, a state space realization is non-unique and it typically involves a larger number of parameters than the transfer function. Least squares may work in a state space context assuming we know some of the parameters. A canonical form is a typical representation that is suitable for least squares application as in eq. (6) where we know the value of the parameters in *n-1* rows of matrix *A*.

$$\mathbf{x}_{k+1} = A\mathbf{x}_k + Bu_k \tag{6}$$

$$A = \begin{bmatrix} a_1\ a_2 \ldots a_n \\ I \quad \mathbf{0} \end{bmatrix};\quad B = \begin{bmatrix} b1 \\ \mathbf{0} \end{bmatrix}$$

3. Estimation of System Parameters using a Kalman Filter.

Given the large number of applications in which it is necessary to stabilize a system from its initial conditions, i.e. under zero input, and the necessity of being able to identify a system with general state space representation, not necessarily in canonical form as discussed in the last section, we present in this section a derivation of the Kalman filter appropriate to identify the parameters of the dynamical system in state space form. More details may be found in [18].

The Kalman filter is generally used to estimate the states of a system using the measured input and output. In this case we will use the Kalman filter to estimate a set of unknown parameters *p*. Consider a discrete time system model in which the system matrices depend on the unknown parameter vector *p*.

$$\mathbf{x}_{k+1} = F_k(p)\mathbf{x}_k + G_k(p)u_k + L_k(p)w_k \tag{7}$$

$$y_k = H_k\mathbf{x}_k + v_k$$

The noise processes w_k and v_k are white, zero-mean, uncorrelated, and have known covariance matrices Q_k and R_k, respectively.

We do not really care about estimating the state, but we are interested in estimating *p*. This is the case, for example, in the aircraft engine health estimation problem [9]. In that paper it was assumed that we want to estimate aircraft engine health (for the purpose of maintenance scheduling), but we do not really care about estimating the states of the engine.

In order to estimate the parameter *p*, we first augment the state with the parameter vector to obtain an augmented state vector:

$$\tilde{x}_k = \begin{bmatrix} x_k \\ p_k \end{bmatrix}$$

$$\tilde{\mathbf{x}}_{k+1} = \begin{bmatrix} F_k(p_k)\mathbf{x}_k + G_k(p_k)u_k + L_k(p_k)w_k \\ p_k + w_{pk} \end{bmatrix} = f(\tilde{\mathbf{x}}_k, u_k, w_k, w_{pk})$$

$$y_k = \begin{bmatrix} H_k & 0 \end{bmatrix} \begin{bmatrix} \mathbf{x}_k \\ p_k \end{bmatrix} + v_k \tag{8}$$

Where w_{pk} is a small artificial noise. Note that $f(\bullet)$ is a nonlinear function of the augmented state $\tilde{\mathbf{x}}_k$; therefore, we can use a nonlinear filter to estimate $\tilde{\mathbf{x}}_k$.

4. Extended Kalman Filter.

The Extended Kalman Filter is a type of linearized Kalman filter used for estimating the states of a nonlinear system; originally, it was proposed by S. Schmidt [2]. The derivation here follows [18].

Consider a nonlinear system described by:

$$\mathbf{x}_k = f_{k-1}(\mathbf{x}_{k-1}, u_{k-1}, w_{k-1})$$

$$y_k = h_k(\mathbf{x}_k, v_k) \tag{9}$$

The noise processes w_k and v_k are white, zero-mean, uncorrelated, and have known covariance matrices Q_k and R_k, respectively. A Taylor series expansion is performed on the state equation around $\mathbf{x}_{k-1} = \hat{\mathbf{x}}_{k-1}^{+}$ and $w_{k-1} = 0$

$$\mathbf{x}_k = f_{k-1}(\hat{\mathbf{x}}_{k-1}^{+}, u_{k-1}, 0) + \left.\frac{\partial f_{k-1}}{\partial \mathbf{x}}\right|_{\hat{\mathbf{x}}_{k-1}^{+}} (\mathbf{x}_{k-1} - \hat{\mathbf{x}}_{k-1}^{+}) + \left.\frac{\partial f_{k-1}}{\partial w}\right|_{\hat{\mathbf{x}}_{k-1}^{+}} w_{k-1}$$

$$= F_{k-1}\mathbf{x}_{k-1} + \tilde{u}_{k-1} + \tilde{w}_{k-1} \tag{10}$$

Where:

$$\tilde{u}_k = f_k(\hat{\mathbf{x}}_k^+, u_k, 0) - F_k \hat{\mathbf{x}}_k^+$$

$$\tilde{w}_k \sim (0, L_k Q_k L_k^T)$$

$$F_{k-1} = \frac{\partial f_{k-1}}{\partial \mathbf{x}}\bigg|_{\hat{\mathbf{x}}_{k-1}^+}$$

$$L_{k-1} = \frac{\partial f_{k-1}}{\partial w}\bigg|_{\hat{\mathbf{x}}_{k-1}^+}$$

Similarly, linearize the measurement equation around $\mathbf{x}_k = \hat{\mathbf{x}}_k^-$ and $v_k = 0$, the mentioned linearization yields:

$$y_k = H_k \mathbf{x}_k + z_k + \tilde{v}_k \tag{11}$$

Where:

$$z_k = h_k(\hat{\mathbf{x}}_k^-, 0) - H_k \hat{\mathbf{x}}_k^-$$

$$\tilde{v}_k \sim (0, M_k R_k M_k^T)$$

$$H_k = \frac{\partial h_k}{\partial \mathbf{x}}\bigg|_{\hat{\mathbf{x}}_k^-}$$

$$M_k = \frac{\partial h_k}{\partial v}\bigg|_{\hat{\mathbf{x}}_k^-}$$

The Kalman filter equations for the new linearized model are given by:

$$P_k^- = F_{k-1} P_{k-1}^+ F_{k-1}^T + L_{k-1} Q_{k-1} L_{k-1}^T$$

$$K_k = P_k^- H_k^T (H_k P_k^- H_k^T + M_k R_k M_k^T)^{-1}$$

$$\hat{\mathbf{x}}_k^- = f_{k-1}(\hat{\mathbf{x}}_{k-1}^+, u_{k-1}, 0) \tag{12}$$

$$\hat{\mathbf{x}}_k^+ = \hat{\mathbf{x}}_k^- + K_k(y_k - h_k(\hat{\mathbf{x}}_k^-, 0))$$

$$P_k^+ = (I - K_k H_k)P_k^-$$

Where $\hat{\mathbf{x}}_k^-$ and $\hat{\mathbf{x}}_k^+$ are the a priori and a posteriori estimates of the state $\mathbf{x}$ at time k, P_k^- and P_k^+ are the a priori and a posteriori covariance of the estimation error and K_k is the Kalman filter gain.

The filter is initialized as follows:

$$\hat{\mathbf{x}}_0^+ = E[\mathbf{x}_0] \tag{13}$$

$$P_0^+ = E[(\mathbf{x}_0 - \hat{\mathbf{x}}_0^+)(\mathbf{x}_0 - \hat{\mathbf{x}}_0^+)^T]$$

A simple example.

Consider the discrete-time linear system:

$$x_{k+1} = Ax_k + Bu_k + w_k \tag{14}$$

$$y_k = x_k + v_k$$

With:

$$A = \begin{bmatrix} a_{10} + \delta_1 & a_2 \\ a_3 & a_{40} + \delta_4 \end{bmatrix}; \quad B = \begin{bmatrix} 1 \\ 1 \end{bmatrix} \tag{15}$$

$a_{10} = 0, a_2 = 0.3, a_3 = -0.5, a_{40} = 0$, the unknown parameter disturbances satisfy $|\delta_1| \le 0.5, |\delta_4| \le 0.7$, we wish to estimate the real value of the parameters $a_1 = a_{10} + \delta_1$ and $a_4 = a_{40} + \delta_4$. Initial conditions for the system are random with uniform distribution with support on [-1,1].

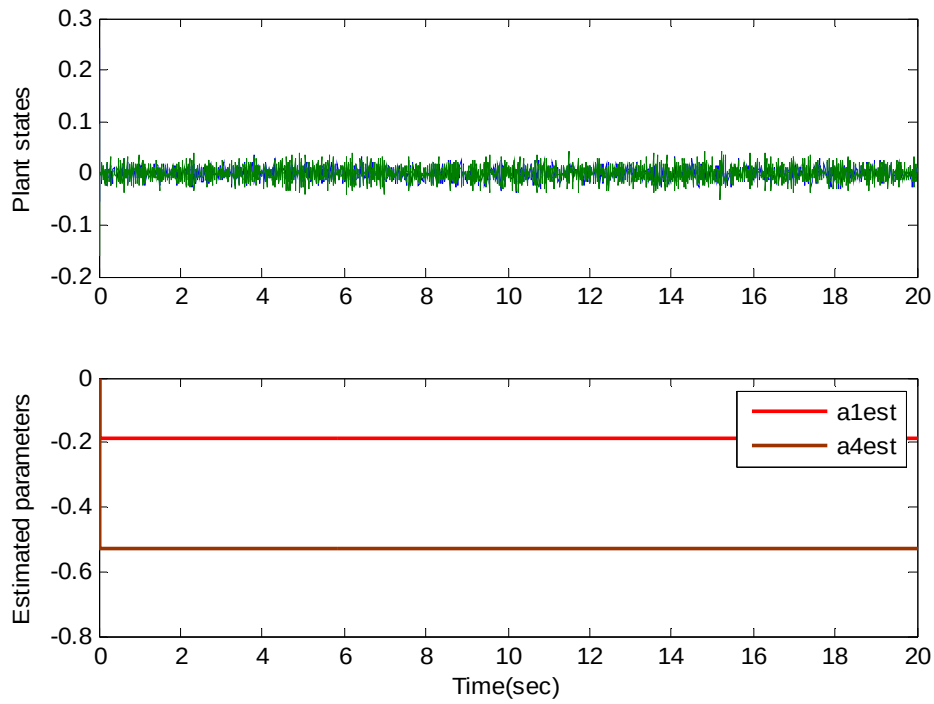

Fig. 1. Kalman filter for system identification with continuous feedback

The real values of the parameters in this simulation were:

$a1 = -0.1853, a2 = -0.5299$

5. Model Based Networked Control Systems (MB-NCS).

As it was mentioned earlier, MB-NCS make 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. Fig. (2) shows a basic MB-NCS configuration, where the network exists only on the sensor-controller side while the controller is connected directly to the actuator and the plant.

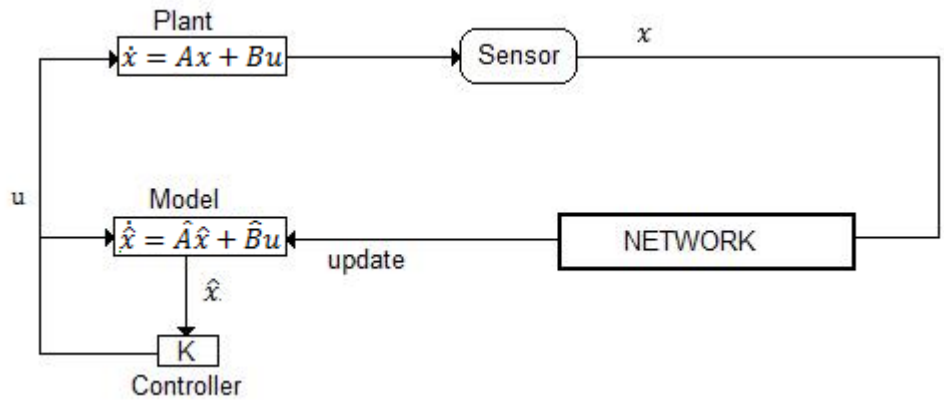

Fig. 2. State feedback Networked Control System.

The dynamics of the plant and the model are given respectively by:

$$\dot{\mathbf{x}} = A\mathbf{x} + Bu \tag{16}$$

$$\dot{\hat{\mathbf{x}}} = \hat{A}\hat{\mathbf{x}} + \hat{B}u \tag{17}$$

Where $u = K\hat{\mathbf{x}}$, and the matrices $\hat{A}, \hat{B}$ represent the available model of the system matrices *A,B.*

Since it is almost impossible to have a model that resembles exactly the real system we are trying to control, the model state will not be exactly equal to the plant state, thus, generating an error:

$$e(t) = \hat{\mathbf{x}}(t) - \mathbf{x}(t) \tag{18}$$

We are required to update the state of the model with the real one to ensure stability. Necessary and sufficient conditions for stability are provided in [12], [14] for periodic instantaneous updates i.e. for a single update of the state at time t_k and for $t_k - t_{k-1} = h$ where h is constant and for both continuous and discrete time systems. Estrada *et. al.* [3] showed that an improved performance on MB-NCS can be reached by using intermittent feedback, in this situation the system

works in closed loop for a finite period of time not just a single update for every cycle.

Let us work under this situation, intermittent feedback, and let the filter be implemented in the controller node of the MB-NCS configuration. As the feedback path from sensor to model-controller (and filter) is closed, the filter will use the set of received measurements to estimate the parameters. The data in this case consist of noisy measurements of the plant state, where the state and measurement equations are given by eq. (14). Once the estimates of the filter converge, the parameters of the model will be updated with the estimates of the filter and the state of the model will be updated with the last received measurement. That is we use intermittent feedback for parameter identification and instantaneous feedback for control.

MB-NCS with periodic updates example.

Consider a stable discrete-time-varying linear system described as in eq. (14), with:

$$A = \begin{bmatrix} a_1 & a_2 \\ a_3 & a_4 \end{bmatrix} \; B = \begin{bmatrix} 1 \\ 1 \end{bmatrix} \tag{19}$$

where $a_1 = -1, a_2 = 0.3, a_3 = -0.5$ are constant and known, and the last parameter, a_4, is unknown and is time-varying governed by $a_4 = 0.5\sin(0.1t)$.

Fig (3) shows the estimated values of a_4; when we close the loop our estimate is updated by the arrival of a new set of measured states, the estimate of the parameter remains constant for the part of the cycle when the feedback loop is open. In this example the loop is closed every 2 seconds and remains closed for a finite period of time (until the estimation converges).

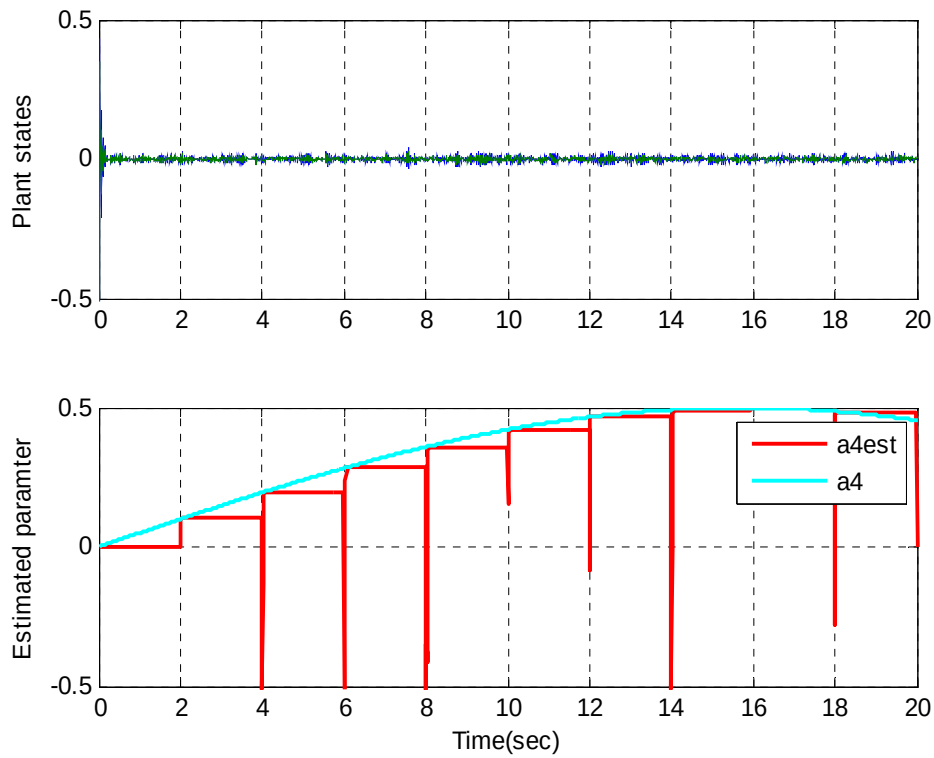

Fig. 3. Estimation of a time-varying parameter with intermittent feedback and periodic updates.

It is convenient in many applications to drop the periodic update implementation in favor of one based on events, for example, the event that the plant-model state error is equal to or greater than some predetermined threshold. Event-triggered control [19], [20] is a scheme that relies on the measurement of the state of the plant to guarantee stability. A sensor node within the network broadcasts its local state only when it is necessary, i.e. when a measure of the local subsystem state error is above some predetermined threshold, the error, in this case, is defined as the difference between the last measured state and the current value of the state of the plant:

$$e(t) = \mathbf{x}(t_i) - \mathbf{x}(t) \tag{20}$$

An extension in this area, self-triggered control [1], avoids the necessity of testing the error (20) frequently by computing the next deadline when the state should be broadcasted again in order to preserve stability.

For our purpose we will assume that the sensor measures the plant state frequently and computes the error. The error is defined as in (18), where the last measured state has been substituted by the state of the model. Another difference in this paper is the type of threshold to be used; in event-triggered the threshold is relative, that is, it depends on the current value of the state of the plant. In the examples shown in this paper we have assumed a fixed threshold for simplicity, but the relative threshold can be used as well. A consequence of using a fixed threshold is that we are only able to show bounded input-bounded output stability.

We can rewrite (16) as:

$$\dot{\mathbf{x}} = (A + BK)\mathbf{x} + BKe \tag{21}$$

where the error (18) has been used. The frequency response to an initial condition of (21) is:

$$\mathbf{X}(s) = (sI - (A + BK))^{-1} x_0 + (sI - (A + BK))^{-1} BKE(s) \tag{22}$$

where $x_0 = x(0) < \infty$ is the initial state of the plant which is unknown and assumed to be finite. From last equation we note that the state of the plant can be considered as the output of a system with initial condition x_0 and input *E(s)*.

Theorem 1. The system described by (22) is BIBO stable if the poles of the closed loop plant are in the left hand side of the complex plane.

Proof. In order to show boundedness of the plant state the L_∞ norm is used:

$$\|\mathbf{X}(s)\|_\infty = \left\|(sI-(A+BK))^{-1}x_0 + (sI-(A+BK))^{-1}BKE(s)\right\|_\infty \tag{23}$$

$$\leq \left\|(sI-(A+BK))^{-1}\right\|_\infty \|x_0\|_\infty + \left\|(sI-(A+BK))^{-1}\right\|_\infty \|BK\|_\infty \|E(s)\|_\infty$$

where $\|E(s)\|_\infty$ is bounded by the predefined fixed threshold. The conditions offered in this theorem relate to the usual assumptions on the MB-NCS literature; that is, a stabilizing and known controller exists for the original non-networked system. An accurate identification of the system parameters allow us to relax this assumption and design a controller based on the information available on the model. The boundedness of a discrete time plant can be shown in a similar way. □

Under the event triggered scenario it is convenient to revisit some implementation issues.

6. A note on implementation.

In the last section we assumed that the filter was implemented in the model-controller node, as depicted in Fig. (4). Let us analyze the advantages and disadvantages of this choice so we can compare it later to a different choice of implementation.

In this configuration the filter in the controller receives a set of measurements (intermittent feedback) that uses for estimation of the parameters of interest. When the estimation algorithm converges, the model is updated with the new value of the parameter and the state of the model is updated using the last measurement available. This option is preferred when we work with periodic updates since no model of the plant is needed in the sensor node; we have only one model whose parameters we need to update. The filter updates directly the

model in the controller immediately after its estimates have converged since no network exists between filter and model.

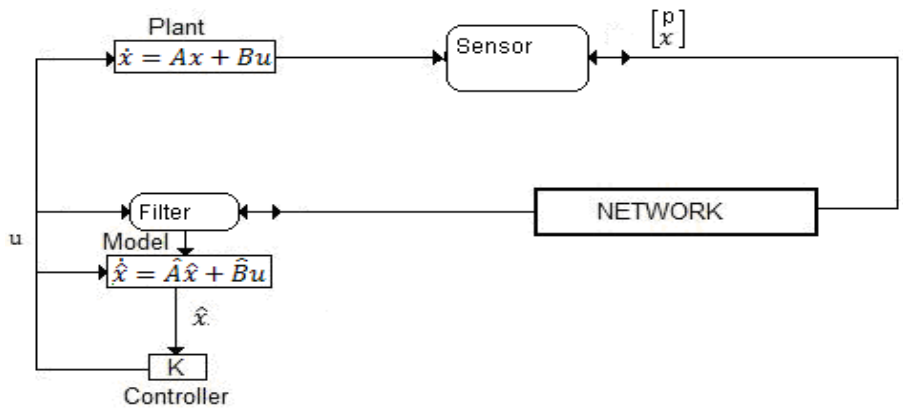

Fig 4. MB-NCS with filter implemented on the controller node.

On the other hand the filter does not have access to the measured state at all time; it only receives measurements when the feedback loop is closed. For the case when we send the measurements based on checking the error in the state (comparing the plant state with the model state) we need a copy of the model in the sensor node in order to generate the model's state. For this scenario we require the controller node to send back to the sensor node the new estimated parameter to update the model in the sensor as it does with the model in the controller. As we mentioned earlier we need intermittent feedback to be able to identify correctly the parameters, if the loop is closed instantaneously the filter does not receive sufficient measurements to perform an accurate estimation. Applications with instantaneous feedback are prevented to use this configuration. Convergence properties of the filter under this configuration are similar to those of a filter receiving continuous measurements, in this work it is assumed that no packets are lost when the loop closes and that time delays are negligible. In practice, however, we implement a limit on the iterations and if the limit is reached and the filter

does not pass the convergence test we choose to keep the last estimated values until new measurements arrive.

Example of first implementation with updates based on the state error.

Consider an unstable system described as in (14) with:

$$A = \begin{bmatrix} a_1 & a_2 \\ a_3 & a_4 \end{bmatrix} \quad B = \begin{bmatrix} 1 \\ 1 \end{bmatrix} \tag{24}$$

where $a_2 = 0.3, a_3 = -1.05$ are known parameters and a_1 and a_4 are unknown constants.

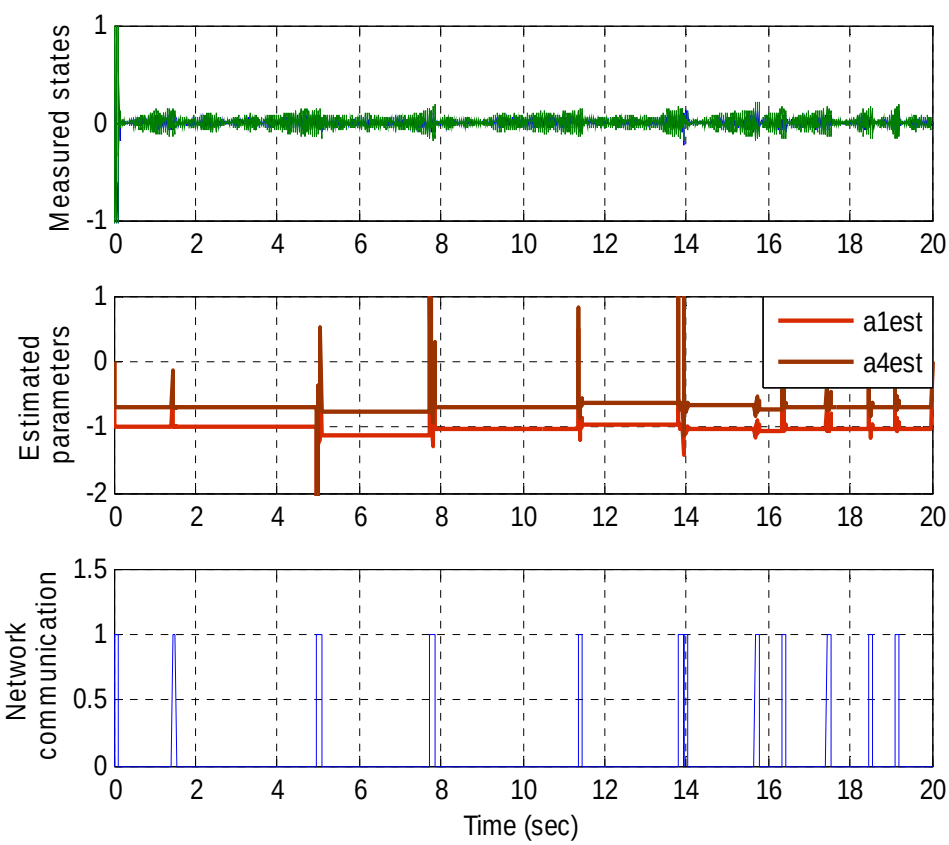

Fig. 5. Estimation of two parameters with intermittent feedback based on events.

Fig. (5) shows results from the simulation under the mentioned implementation. The estimated parameters a_1 and a_4 converge to values around -0.985 and -0.7 respectively, which are the real parameters used in this simulation. Note that every time that the controller node receives data from the sensor it uses this data to estimate the unknown parameters, update the model, and redesign the controller computing a Linear Quadratic Regulator using the improved model. The loop needs to be closed for a finite period of time as shown at the bottom of fig. (5), which shows when and for how long the feedback loop is closed.

A second option for the implementation of a Kalman filter in the MB-NCS configuration is depicted in Fig. (6).

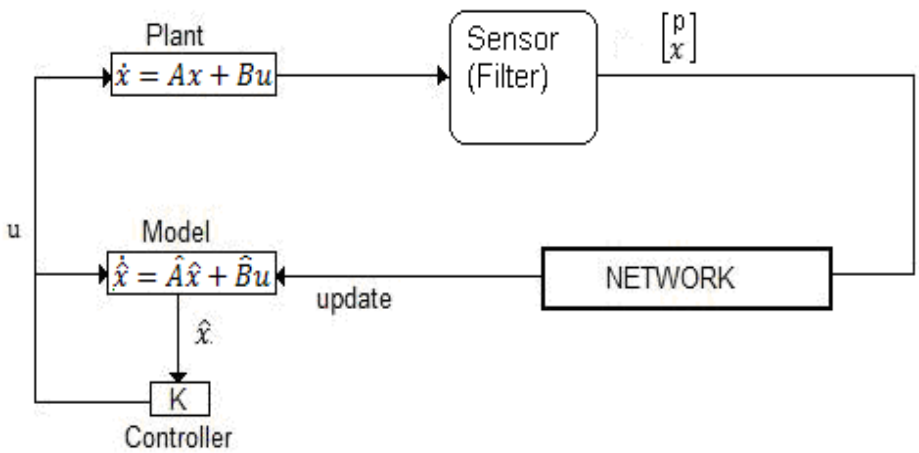

Fig. 6. MB-NCS with filter implemented on the sensor node.

In this configuration the filter is implemented in the sensor node. We assumed a copy of the model and controller are contained in the sensor to generate the state that is compared with the measured state, and the input that is needed by the filter. The sensor will transmit the measured state along with the new value of the estimated parameters.

This choice of implementation has some advantages over the previous one. The filter has access to the measured state at all time so it can generate better estimates. There is actually no network between the measurements and the filter so the convergence properties of the filter are preserved. In addition, we do not necessarily need intermittent feedback, i.e. a single transmitted packet may contain the estimated parameters and the measured state; another important and useful advantage in this situation is that the sensor can decide to send a smaller packet containing only the measured state if no change has been recorded in the value of the estimated parameters since the last update.

There are some disadvantages to consider as well. The sensor has to perform more functions: measure the state, run the model to generate the model state, compute the error and transmit if it is greater than some threshold, and now, it needs to run the filter to find estimates of the parameters, definitely, we are increasing the demands on our integrated sensor. Another possible disadvantage is found in the periodic updates implementation, if a transmitted packet is lost we do not only lose the update for the state but also the update for the estimated parameter and we need to wait again for the next cycle, in the case that no acknowledgement signals are used.

The choice of implementation has to be made primarily by considering what kind of updates are going to be used and the capacities of the sensor.

Example of second implementation with updates based on the state error.

Consider the same unstable system described in the last example, which is now implemented using the second configuration that was just described. The simulation is shown in fig. (7). It can be seen that the estimated parameters are more consistent and that we can use instantaneous feedback as shown at the

bottom of fig. (7). In this case when the loop closes, the sensor only needs to send a single packet of information.

In both cases the estimates are very close to the real values. The system contains process noise and measurement noise, and those perturbations account for the main reason in the difference between the real state and the plant state, when this difference or error is larger than the fixed threshold, the sensor sends packets of information to the controller node.

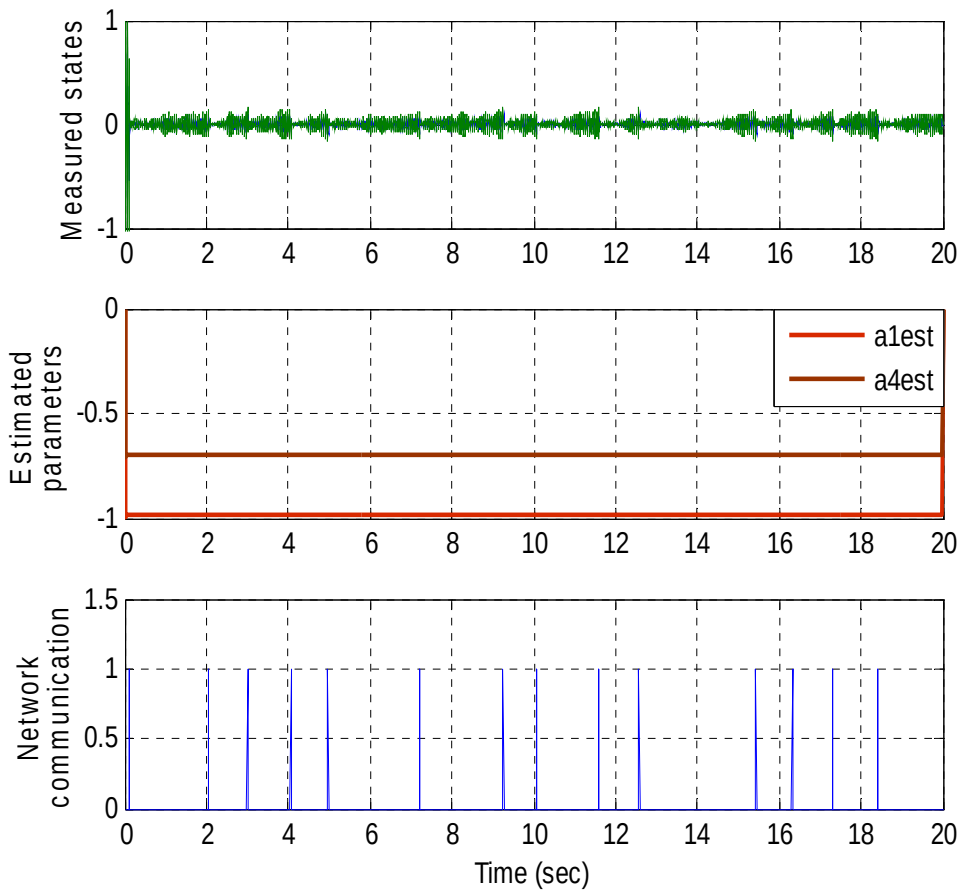

Fig. 7. Estimation of two parameters using a Kalman filter on the sensor node.

As it was mentioned earlier, under the last implementation, the sensor sends a packet that contains the estimated parameters and the measured state of the plant, but it can also choose to send a smaller packet containing only the state if the estimated parameters have not changed significantly; this can be seen in the next example. Here, the unknown parameters of the plant experience discrete changes in their values, and, for the purpose of illustration, we construct a signal that may take on any of the next three values:

$$r(k) = \begin{cases} 0 \text{ if no packet is sent} \\ 1 \text{ if only the state is sent} \\ 2 \text{ if both, parameters and state are sent} \end{cases} \tag{25}$$

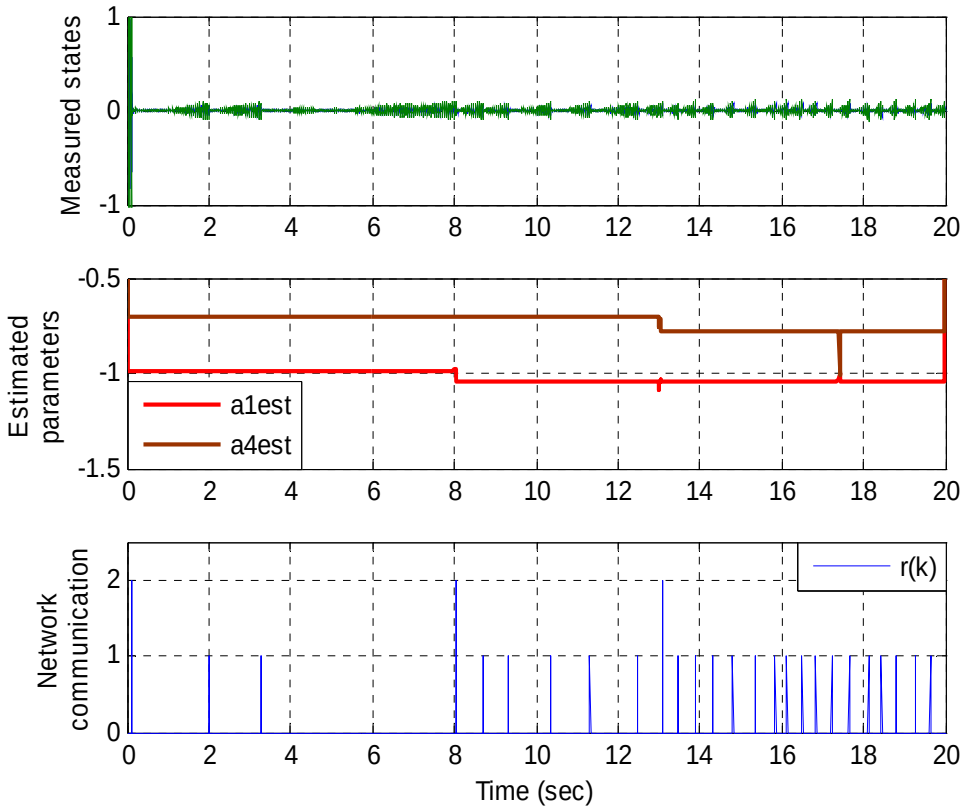

Fig. 8. Estimation and broadcast of time-varying parameters

As it can be seen from the bottom of fig. (8), the sensor node sent a complete package (state and parameters) in the first transmission, when a_1 changed its value from -0.985 to -1.04, and when a_4 changed from -0.7 to -0.78. As the parameters changed the original system became more unstable and the state error exceeds the threshold more frequently.

Remark 1. The use of the Kalman filter to identify parameters of a system in state space representation requires a partial knowledge of the plant dynamics, but, in contrast to the Recursive Least Squares algorithm, it is not limited to a certain form of the matrix *A*.

Remark 2. In MB-NCS we rely on the model and its state to stabilize the plant when no feedback is available. A better knowledge of the dynamics of the plant results in an improvement in the performance of the networked system, in this case, longer times between transmissions can be achieved, making the network more available for other systems to communicate or for other applications.

7. Conclusions and future work.

In this paper, algorithms for recursive system identification applied to Model Based Networked Control Systems are considered. An extension of the Kalman filter for identification of parameters is discussed and it is shown that such extension utilizes a nonlinear model that requires a nonlinear Kalman filter. One type of such filter, the Extended Kalman Filter (EKF), is used in this paper in the identification algorithm. The theory that involves the EKF and the identification of system parameters using Kalman filters was discussed and the application of this identification method to MB-NCS is shown through simulations. Stability

using periodic updates follows from prior work [14] on MB-NCS and sufficient conditions for stability on the event-triggered case with fixed threshold were shown. The robustness properties of this scheme are left for future work. Two implementation cases are proposed; the choice of implementation depends on factors such the availability of resources in the sensor node and the type of updates to be used, periodic or aperiodic based on the state error.

References.

[1] A. Anta and P. Tabuada, "Self-triggered stabilization of homogeneous control systems", Proceedings of the 47th Conference on Decision and Control, 2008.

[2] J. Bellantoni and K. Dodge, "A square root formulation of the Kalman-Schmidt filter", AIAA Journal, 1967

[3] T. Estrada, H. Lin, and P. J. Antsaklis "Model based control with intermittent feedback" Proceedings of the 14th Mediterranean Conference on Control and Automation, 2006.

[4] G. F. Franklin, J. D. Powell, and M. Workman, *Digital control of dynamic systems*, 3rd edition, Addison Wesley Longman, Inc.

[5] D. Graupe, *Time series analysis, identification and adaptive filtering*, 2nd edition, Robert E. Krieger Publishing Company; Malabar, Florida, 1989.

[6] J. P. Hespanha, P. Naghshtabrizi, and Y. Xu, "A survey of recent results in Networked Control Systems" Proceedings of the IEEE vol. 95, no.1, pp. 138-162, January 2007.

[7] P. Ioannou and B. Fidan, *Adaptive control tutorial*, Society for Industrial and Applied Mathematics; Philadelphia, PA, 2006.

[8] C. R. Johnson, *Lectures on adaptive parameter estimation*, Prentice Hall Advanced Reference Series, Englewood Cliffs, New Jersey, 1988.

[9] T. Kobayashi and D.L. Simon, "Application of a bank of Kalman filters for aircraft engine fault diagnosis," *ASME Turbo Expo,* Atlanta, paper GT2003-38550, June 2003.

[10] I. D. Landau, "Unbiased recursive identification using model reference techniques", IEEE Transactions on Automatic Control, vol. AC-21, pp. 194-202, 1976.

[11] L. Ljung and T. Soderstrom, *Theory and practice of recursive identification,* The MIT Press, Cambridge, MA, 1983.

[12] L. A. Montestruque, and P. J. Antsaklis, "Model-based networked control systems: necessary and sufficient conditions for stability" Proceedings 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, and P. J. Antsaklis "On the model-based control of networked systems" Automatica, 2003 pp. 1837 - 1843.

[15] J. R. Moyne and D. M. Tilbury, "The emergence of industrial control networks for manufacturing control, diagnostics, and safety data", Proceedings of the IEEE, vol. 95, no.1, pp. 29-47, January 2007.

[16] S. Sastry and M. Bodson, *Adaptive Control, Stability, Convergence, and robustness*, Prentice Hall; Englewood Cliffs, New Jersey, 1989.

[17] L. Schenato, B. Sinopoli, M. Franceschetti, K. Poolla, and S. Sastry, "Foundations of control and estimation over lossy networks", Proceedings of the IEEE, Vol. 95, No. 1, January 2007.

[18] D. Simon, *Optimal State Estimation,* Wiley; Hoboken, New Jersey, 2006.

[19] 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

[20] P. Tabuada, "Event-triggered real-time scheduling of stabilizing control tasks", IEEE Transactions on Automatic Control, Vol. 52, No. 9, September 2007.

Appendix A. Recursive Least Squares on Model Based Networked Control Systems.

In section 2, the Recursive Least Squares (RLS) algorithm was studied, now we offer an example in which RLS is implemented in the controller node of a MB-NCS with intermittent feedback with updates based on the state error. The first implementation in section 6 has been used for the results shown in fig. A.1.

Consider the discrete time system:

$$x_{k+1} = Ax_k + Bu_k \tag{A.1}$$

$$y_k = Cx_k$$

With:

$$A = \begin{pmatrix} a_1 & a_2 & a_3 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix} \quad B = \begin{pmatrix} b_1 \\ 0 \\ 0 \end{pmatrix} \quad C = \begin{pmatrix} 1 & 0 & 0 \end{pmatrix}$$

The initial parameters are given by $a_1 = -1.2, a_2 = -0.6, a_3 = 0.6$ and $b_1 = 10$. It is assumed that no prior knowledge of the parameters is available so the initial estimate for all the parameters is zero; if some previous estimate exist it can also be used for faster convergence. Discrete variations in the value of the parameters

were introduced at specific times and successful identification was achieved in the controller node as shown in the second graph of fig. A.1. The real parameters variations and the times of occurrence are:

$a_1 \rightarrow -1.02$ at $t = 3$ sec

$a_2 \rightarrow -0.76$ at $t = 8$ sec.

$a_3 \rightarrow 0.45$ at $t = 15$ sec

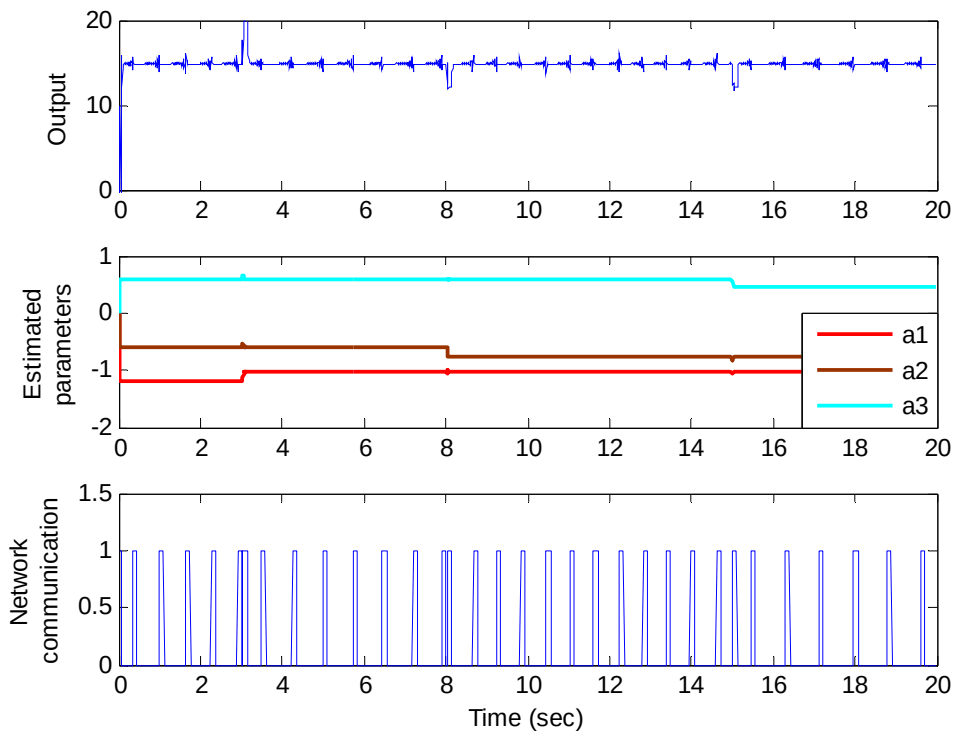

Fig. A.1. RLS implemented over a MB-NCS.

The higher peaks seen in the output of the system correspond to the variations on the parameters, the rest are due to very small inaccuracies between the parameters

and their estimates. In order to obtain a successful identification using RLS it is necessary a sufficiently rich input; for this example a unit step input is used. Identification of parameters under a zero-input scenario is not possible, limiting the use of RLS for applications in which stabilization from initial conditions is needed.

Implementation of RLS using the second implementation described in section 6 is also possible; the identification algorithm is now processed in the sensor node. Similar to the Kalman filter case we construct the network communication signal (25). System (A.1) is used here with the same parameter variations.

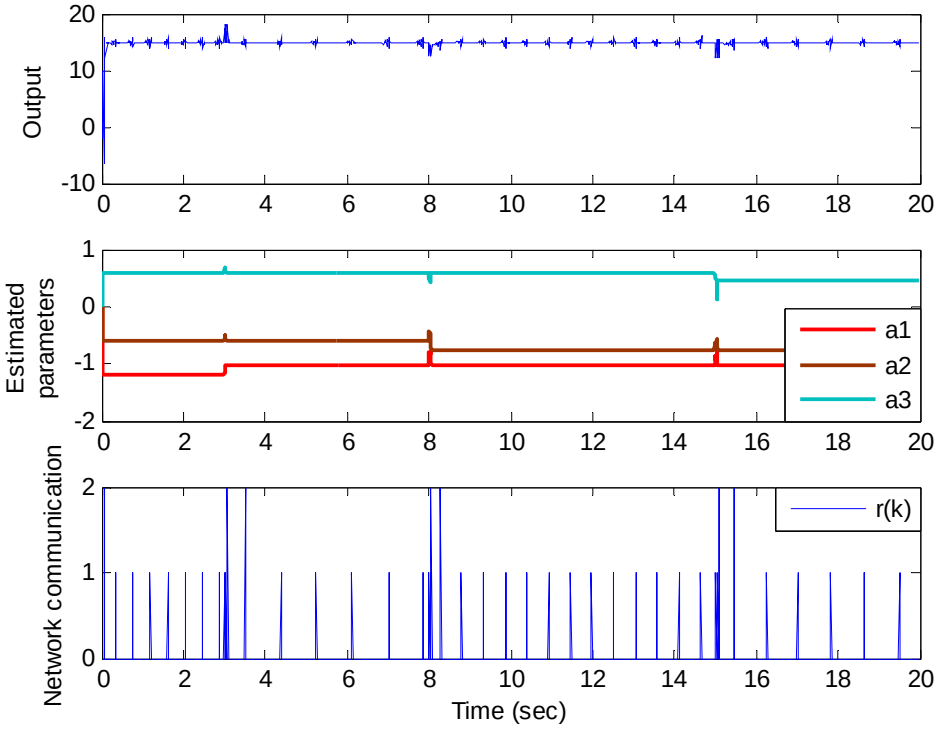

Fig. A.2. RLS second implementation over MB-NCS

With this implementation is also possible to use instantaneous feedback, a single packet updates both the parameters and the state of the model. Note that in both implementations of the RLS algorithm we know exactly the remaining parameters of the system matrices in addition to the restriction of the use of the canonical form.