On the Application of Block Transmissions For Improving Control over Lossy Networks

Milton Armando Cunguara,Tomás António Mendes Oliveira e Silva
DETI/IEETA/University of Aveiro
Aveiro, Portugal
{milton.cunguara,tos}@ua.pt

Paulo Bacelar Reis Pedreiras
DETI/IT/University of Aveiro
Aveiro, Portugal
pbrp@ua.pt

***Abstract*—Over the last few years the problem of optimal estimation and control over lossy networks has been studied in detail. However, most of such works assume that in the absence of a new actuation value, the actuator either outputs zero or the previous value. Some other extensions have been proposed, but they present similar limitations. This paper shows that this method is not optimal and presents a new approach, in which the controller sends a number of predictions of future actuation values, which are outputted when the actuator does not receive a new message. Under this new assumption it is shown that the controller/estimator herein presented performs equal to or better than any previously introduced controller, being the equality verified only in the absence of communication errors. A modified Kalman filter as well as a specific optimal controller for this novel actuation model are also presented.**

I. Introduction

During the last four decades the average processing and communication capabilities of the nodes employed in distributed control systems have increased steadily. However, these systems are frequently deployed in harsh environments, being subject to strong electromagnetic interference, noisy power supplies, mechanical vibration, among other perturbations. Thus, despite all aforementioned advances, communications are still affected by non-negligible error-rates. The move towards wireless media, currently observed in many application areas, exacerbates this problem. Obviously, the performance of control applications supported by this kind of communication infrastructure can be severely deteriorated. Thus, to achieve an adequate quality of control, it is necessary to develop a control framework that that takes into account the effects of packet losses in control and estimation.

Depending on the existence or not of explicit acknowledgment signals, communication protocols are usually classified into TCP-like and UDP-like categories, respectively [1]. Assuming a fully-distributed architecture (i.e., sampler, controller and actuator reside in separated nodes), under TCP-like protocols the controller receives an acknowledgment from the actuator signaling the correct reception of the last actuation value. Therefore, in TCP like protocols, whenever a state-estimation is performed, the estimator has perfect knowledge of the inputs of the system. Under UDP-like protocols, there are no such acknowledgments, thus the estimator cannot know which input was used in the previous actuation instant.

This work was partially supported by the Portuguese Government through FCT - Fundação para a Ciência e a Tecnologia in the scope of project Serv-CPS -PTDC/EEA-AUT/122362/2010 and Ph.D. grant - SFRH/BD/62178/2009/J53651387T4E

This lack of information regarding the correct reception of messages by the actuator renders the estimation error covariance matrix a function of the input, thus leading to a complex — i.e. non-quadratic, *optimization problem*, which has no known optimal solution, as shown in [2]. On the other hand, the use of TCP-like protocols suffers less from this ailments. However, it requires acknowledgments that, a) are also sent over a lossy network, and therefore are themselves subject to errors b) increase the transmission error probability c) increase the network load and complexity of the communication stacks.

The literature presents two main strategies to cope with missing actuation data. In the first one, if there is no new message the actuator outputs zero (Figure 1 a)). In the second one, called *hold*, in the absence of a new message the actuator outputs the previous value (Figure 1 b)). This paper argues that the best option is to use an estimation of the control value given the set of previous control and output values (Figure 1 c)). This scheme provides a simple and optimal realization. This paper also shows that for TCP-like protocols the separation principle holds, i.e. the error covariance matrix is independent of the controller gain, while for UDP-like protocols the separation principle does not hold in general, though, as will be shown, for the class of controllers presented in this paper, the estimation auto-covariance matrix is independent of the state. Therefore, for the controller presented in this paper, both for TCP and for UDP-like protocols, the optimal estimator is independent of the optimal controller.

The remaining of this paper is organized as follows: in section 2 a review of previous contributions closely related to the subject of this paper is presented. In section 3 the rationale for the main proposal of this paper is described. Section 4 presents the main results, whereas section 5 contains the validations of the proposed method. Finally, section 6 includes a summary of the contributions and presents the main conclusions.

II. Related Work

The subject of estimation and control over lossy networks is relatively old. The first appearances of it in the literature try to find a correspondence between the Shannon information theory [3] and control theory. More specifically, tried to answer

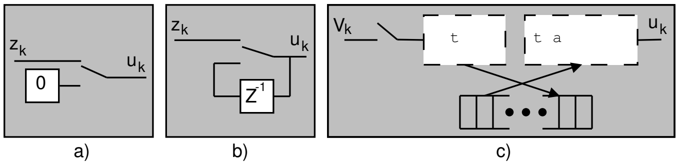

Fig. 1. Types of output strategies

the age old question: *what is smallest bit-rate necessary to stabilize a given system* or, equivalently, what is the maximum transmission error rate that a network can have that ensures the stability of the system? However, *Which techniques achieve this threshold?* is a question that, to the best of the authors' knowledge, is first asked in this paper and it is hopped that our work aids in its response. A number of early attempts to answer the first question are presented in [4][5].

In [6] the authors attempt to extend the classical theory of optimal control [7][8]. However, they conclude that the separation principle, which allows a simplification of the results, does not hold in the general case. However, other simpler results are shown, such as, that a system is stable if the independent and identically distributed (i.i.d.) variables that describe the error rate (α) verify $\alpha < [\max \lambda(\mathrm{A})]^{-2}$, assuming that the system is unstable, where $\lambda(\mathrm{A})$ is the set of all eigenvalues of the matrix A and A is the state-transition matrix. Similar claims are made in [9].

[2] makes a more in-depth study of such issues, presenting formally the UDP and TCP-like protocol cases. TCP-like protocols are more malleable to the existing optimal control theory, whereas UDP-like protocols, under the assumptions made therein (output zero) lead to a non simple optimization problem, since the estimation error covariance matrix at any given time step will depend on previous control values. It is shown in the same reference that, under their assumptions, TCP-like protocols have a error-rate similar to the one found in the previous papers, but UDP-like protocols have lower error-rate bounds that guarantee stability.

Another somewhat related development is given in [10], in which state information is sent in more that one packet. This fragmentation is done primarily in an attempt to reduce the impact of packet losses, since loosing a packet with a significant large number of state variables would be worst than loosing one packet with only one variable.

[11] takes a radically different approach. It uses passive networks as a mean to provide jitter immunity to the controlled system. However, no reasons to believe that passive networks are more immune to jitter is presented, and no mechanism to design passive controllers to achieve certain metrics, e.g. state-tracking or input tracking under noisy conditions, are provided.

In [12] the authors introduce a new concept, i.e., the packet acknowledgment is also probabilistic. By varying the probability of an acknowledgment, it is possible to go from no acknowledgment (UDP-like protocols) through a gray zone up to the situation in which there is an acknowledgment with probability one (TCP-like protocols). Therefore, in a certain way, they extended the notion of UDP-like versus TCP-like protocols. This extension was not employed in this paper because the authors are not aware of any network protocols that support such extension.

In [13] the output schemes are extended from the zero versus hold spectrum by allowing the value that is held at the actuator to decay with time, that is,

$$x_{k+1} = Ax_k + Bs_k + w_k \tag{1a}$$

$$s_k = L_k r_k \tag{1b}$$

$$r_k = \theta_k u_k + (1 - \theta_k) M_k r_{k-1} \tag{1c}$$

in which $\mathrm{s_k}$ is the value that is actually outputted, $\mathrm{r_k}$ is the actuator internal state representation which is allowed to be a matrix, $\mathrm{u_k}$ is such representation computed at the controller using information from the sensors and it is transmitted in a bulk message, $\mathrm{M_k}$ is a state transition matrix of the actuator state and $\mathrm{L_k}$ is the respective controller gain (actually, in the paper, $\mathrm{s_k}$ it is assumed to be equal to the first column of $\mathrm{r_k}$ and constraints are placed on $\mathrm{M_k}$). The authors of the paper in question go on to propose a suboptimal controller given their respective controller design. However, it can be argued that the initial representation is flawed. The first aspect is related to the fact that this approach both sends a large number of messages and executes a rather complex actuator. Second, it has an internal actuator state which is a matrix and it is well known from the theory of realizations that the total number of entries of such state can be chosen such that it is not larger that the size of the state *per se*.

In [14] it is proposed the use of two mechanisms to cope with missing messages. In the controller it is always assumed that control messages arrive at its destination whereas the process errors are filtered through an observer gain, executed only when there is a new sensor message. A similar mechanism is employed at the actuator side, i.e. it assumes that the controller always receives the sensor message and it uses an observer gain also used only when a new controller message is received. The authors also provide a stability analysis (of dubious correctness) for systems in which their assumptions are valid. In essence, the paper is similar to [13] in which the matrix $\mathrm{M_k}$ is chosen as the closed-loop state transition matrix. It is given no reason for the use of this particular choice nor it is proven that their choice has the behaviour in question, two points that are covered in this paper. Furthermore, the authors failed to notice 1) the suboptimality of observers as opposed to the Linear Quadratic Gaussian (LQG), which implies that the optimal (LQG) gain applied to the observer in the actuator simply returns the value received from the controller which in turn renders the observer part of the actuator pointless. 2) the use of an actuator with observing capabilities renders the use of the controller pointless, since the sensor could send its data directly into the actuator where the data would be properly treated. In fact, the use of an intermediary network element only increases the end-to-end error probability.

An aspect related to the critique of the previous paper is explored in [15] in which it is shown that in physically distant networks the closer the controller is to the actuator the better the quality of control and (the relevant part) the best results are achieved when the controller and the actuator are collocated. This implies that the quality of control may be improved by moving some of the functionalities of the controller into the actuator. However, some proposals fully replicate the controller into the actuator. In this paper, the internal function of the actuator are simulated in the controller but the converse is not true.

In [16] two new concepts are introduced. The first one consists in the use of differentiated services (DiffServ) on control networks, which is intended to provide different Quality of Service (QoS) to different networked control processes. The second new concept is the use of Model Predictive Control (MPC) in which the cost function is minimized over a limited number of steps into the future and only the current control value is outputted whereas the remaining are discarded. The two *parts* are connected by a scheduler that uses Linear Programming (LP) to schedule the messages. However, due to the connection of various areas that are still open problems, the overall solution presented itself to be severely suboptimal.

In [17] the authors propose to estimate whether a given controller message reached the actuator. The remaining of the paper further develops the idea. Nevertheless, the assumption related to the type of system, i.e. rank of the various matrices, renders the work of limited utility.

In [18] it is considered the behaviour of the auto-covariance matrix of the state estimation error. In particular, the fact that it does not converge as in the case of the regular Kalman filtering. The analysis is done using the Stieltjes transform which transforms the set of (expected) eigenvalues of a given stochastic matrix into a polynomial in the complex plane.

III. Controller Designed Rationale

The idea that motivated this work is that even if the optimal value is missing, due to communication errors, the actuator can output an acceptable value if the controller had previously predicted and sent a set of future actuation values, which are stored locally, at the actuator, as depicted in Figure 3. In this figure, DAC denotes the Digital-to-Analog converter, whereas the remaining variables will be defined on section IV.

Due to the random nature of the network errors, the controller cannot predict which messages will be successfully delivered. Therefore, in each execution, the controller computes several future control values, given the past information that it has, and then sends them to the actuator. Whenever a message is either lost or delivered late, the actuator can simply output an estimated value from the last successfully received message. Figure 2 provides an example of such process, where the message sent at time $\mathrm{K}+2$ is lost and thus it is used an estimate sent at time $\mathrm{K}+1$.

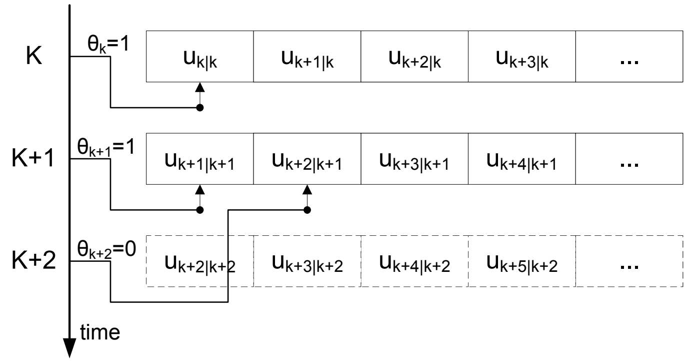

Fig. 2. Example of Actuator Output Sequence.

Note that systems with dominant, discrete-time, closed-loop poles close to 1, produce control sequences that change relatively slowly. Hence, in these circumstances and if the cost of control is relatively small, applying the hold strategy provides a good performance. Conversely, for systems in which present actuation values are not correlated with past values and the cost of control is relatively high, it makes sense to output zero in the event of message losses. This is the case where the output zero strategy performs better than the hold one. However, the idea presented in this paper is optimal in either extreme and in the cases in between.

Furthermore, though it is presented as a mechanism in which the controller sends a bulk of control values to the actuator, it is possible to implement a similar control system in which the controller sends its estimate of the system state and the actuator computes future estimates whenever it does not receive a new message. However, the approach adopted in this paper has the advantage of requiring only one network element with a relatively high computational power.

It is assumed that the network drops the control messages

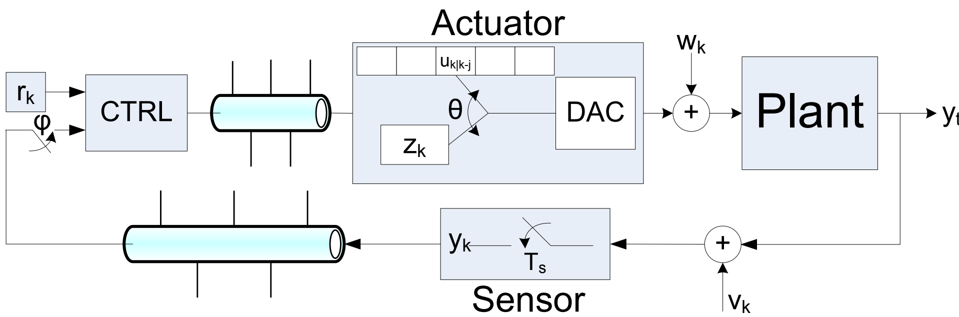

Fig. 3. Proposed Control Architecture

as an i.i.d process. However, this assumption can be relaxed, since error bursts that occur consecutively but within the span of the same message are already considered in the analysis. Furthermore, the expected number of message transmissions that provides a probabilistic guarantee that at every instant there is a control value with a given probability informs on the necessary size of the buffers. This already deals with non i.i.d. packet drop processes because if, for example, the errors occur in burst of N messages, then the buffer need to have only $\mathrm{N}+1$ entries.

Obviously, the approach proposed in this paper incurs in overhead. Regarding the network, current fieldbuses have a payload that accommodates easily the additional amount of data. In some cases, e.g. Ethernet, it may easily happen that the additional data still fits in the minimum payload. In these cases there is no additional cost at all. As for the computational capabilities, most of the additional burden is put on the controller node, which normally is a node with relatively high computational capabilities. Therefore, the additional computations should imply only a marginal increase of the computational processing power of the controller node.

IV. Problem Statement and Solution

Consider a fully distributed system, as depicted in Figure 3, described by its state-space representation as follows:

$$x_{k+1} = Ax_k + Bu_k + w_k \tag{2a}$$

$$y_k = Cx_k + v_k \tag{2b}$$

where x is the state variable, u is the control value computed by the controller, w is the input noise, y is the output, v is the output noise, and k is the index of the sample. A, B and C are the state transition, input and output matrices, respectively. Let $\hat{\mathrm{x}}_{\mathrm{k|k-j}}$ be the best estimation of $\mathrm{x_k}$ given all the information known at instant $\mathrm{k}-\mathrm{j}$. Lets define also

$$P_{k|k-j} = \mathbb{E}\left[\left(\hat{x}_{k|k-j} - x_k\right)\left(\hat{x}_{k|k-j} - x_k\right)'\right] \tag{3a}$$

$$\mathbb{E}\left[[w_k \;\, v_k][w_i \;\, v_i]'\right] = \begin{bmatrix} Q & 0 \\ 0 & R \end{bmatrix} \delta_{ki} \tag{3b}$$

where $\mathrm{P_{k|k-j}}$ is the auto-covariance matrix of the estimation of $\mathrm{x_k}$ given all the information received up to the instant $\mathrm{k}-\mathrm{j}$. Q and R are the input and output error covariance matrices, respectively. These last two matrices are not necessarily constant. However, since this fact does not have a direct impact in the results presented herein, it will be assumed that they are constant. δ_{ki} is the Kronecker delta which is 1 if $\mathrm{k}=\mathrm{i}$ and 0 otherwise. It is also assumed that w and v are uncorrelated white Gaussian noise sequences.

Considering a linear controller and that the certainty equivalence principle holds, as will be shown in a subsequent section, each possible output of the controller can be written as:

$$u_{k|k-i} = -L_k \hat{x}_{k|k-i} \tag{4}$$

in which $\hat{\mathrm{x}}_{\mathrm{k|k-i}}$ is the state estimation as defined above, $\mathrm{L_k}$ is the controller gain matrix at the $\mathrm{k^{th}}$ step and $\mathrm{u_{k|k-i}}$ is the value that the controller expected at instant $\mathrm{k}-\mathrm{i}$ to be the optimal control value at instant k.

Due to the fully distributed architecture considered in this paper, communication errors may happen both between the sensor and the controller and between the controller and the actuator. It is assumed that the underlying communication network provides suitable error detection mechanisms so that messages are either correctly transmitted or discarded. As illustrated in Figure 3, transmission errors are modeled by the boolean variables φ_{k} (sensor-controller) and θ_{k} (controller-actuator), which take the logical value 1 if the transmission is carried out correctly and 0 otherwise. Furthermore, regarding their stochastic behavior, it is assumed that they are independent and identical distributed (i.i.d) Bernoulli variables, i.e. $Pr(\theta_k = 1) = Pr(\varphi_k = 1)$ and $Pr(\theta_k = 0) = Pr(\varphi_k = 0)$.

According to the methodology presented in this paper, in each execution $(\mathrm{k}-\mathrm{j})$ the controller computes $\mathrm{u_{k-j}}$ as well as a set of j future values $\{\hat{\mathrm{u}}_{\mathrm{k-j+1|k-j}}, \hat{\mathrm{u}}_{\mathrm{k-j+2|k-j}} ... \hat{\mathrm{u}}_{\mathrm{k|k-j}}\}$. These future values are computed based in the expectation of the system state x at the respective time instants. It is important to stress that these estimates are updated by the controller and sent to the actuator in each controller execution. Therefore, each one of the estimated control values $\hat{\mathrm{u}}_{\mathrm{k-i|k-j}}, \mathrm{i} \in \{0..(\mathrm{j}-1)\}$ is actually outputted if and only if its predecessor $\hat{\mathrm{u}}_{\mathrm{(k-i)-1|k-j}}$ was also outputted. This is so because the correct reception of a message at instant $(k-m)$ replaces all previous estimates in the interval $\{(\mathrm{k}-\mathrm{m})...\mathrm{k}\}$. This fact significantly simplifies the computation of $\hat{\mathrm{u}}_{\mathrm{k|k-j}}$.

Under the above presented conditions, the value that is actually used by the actuator at execution k, $\mathrm{z_k}$, may be different from the value generated by the controller at the same execution $\mathrm{u_k}$, being equal to:

$$z_k = \sum_{j=0}^{k} \theta_j \left[\prod_{l=j+1}^{k} (1-\theta_l)\right] u_{k|j} \tag{5}$$

The boolean part of the argument of summation is 1 if the $\mathrm{j^{th}}$ message was received and all the future messages up to the instant k were not. Therefore, the whole summation tells 1) what was the last received message and 2) what was the estimate for $\mathrm{u_k}$ at that time.

Hence, the state in the next step, can be written as

$$x_{k+1} = Ax_k + Bz_k + w_k \tag{6a}$$

$$= Ax_k + B\sum_{j=0}^{k} \theta_j \left[\prod_{l=j+1}^{k} (1-\theta_l)\right] u_{k|j} + w_k \tag{6b}$$

A. Noise Filtering

The computation of the estimates involves two complementary steps. In the first one, designated **innovation**, an estimation of the state in the next instant is made, followed by the the computation of its associated error covariance matrix. The second step, designated **correction**, consists in a comparison of actual observations (with certain error covariance matrices) with the previously made estimations and the computation of new estimations with a lower error covariance matrix. Both of these steps are presented in detail in the following sections. Since UDP- and TCP-like protocols behave differently, a separate analysis for each of them is presented.

1) innovation: The innovation step is based on equation (6), consisting in the computation of the system state at the next instant given information available prior to the reception of the sample of the instant in question. UDP-like protocols lack acknowledgment mechanisms, therefore the values of θ_j are unknown on the controller side. Given the type of control that is proposed, this behavior can be modeled with the help three new variables, t_k, $\tilde{e}_k$ and ϵ_k, defined as follows:

- $t_k = \theta_k \hat{x}_{k|k} + (1-\theta_k)(A - BL_{k-1})\, t_{k-1}$ is the state estimate that results from equation (5), representing the state estimation that was used when computing the control value that the actuator actually applied to the environment;
- $\tilde{e}_k = (A - BL_{k-1})\hat{x}_{k-1|k-1} - \hat{x}_{k|k}$ is the information acquired about the state by performing the k^{th} correction step, being equal to the difference between the uncorrected and corrected state estimation;
- ε_k is a state variable associated with the input $\tilde{e}_k$, i.e., $\varepsilon_{k+1} = (A - BL_k)\varepsilon_k + \varphi_{k+1}\tilde{e}_{k+1}$, where φ_{k+1} is the boolean variable defined above, which models the communication errors between the sensor and the controller.

Since t and ε were defined recursively, it is also necessary to define boundary conditions. Given the physical meaning associated with the variables, it is sane to define $t_0 = \hat{x}_0$ and $\varepsilon_0 = 0$. Note that $\varphi_{k+1}\tilde{e}_{k+1} = \tilde{e}_{k+1}$ since whenever $\varphi_{k+1} =$ '0' there is no update hence $\tilde{e}_{k+1} = 0$, hence the multiplication by φ_{k+1} is redundant, and whenever $\varphi_{k+1} =$ '1' it is the neutral element of multiplication, therefore, it is always redundant.

Under these assumptions $t_k = \hat{x}_{k|k} + (1-\theta_k)\varepsilon_k$. The last statement can easily be proven by induction, i.e., the initial value already satisfies the assumption and

$$
\begin{aligned}
t_{k+1} &= \theta_{k+1}\hat{x}_{k+1|k+1} + (1-\theta_{k+1})(A - BL_k)\, t_k && \text{(7a)}\\
&= \theta_{k+1}\hat{x}_{k+1|k+1} + (1-\theta_{k+1})(A - BL_k)\left(\hat{x}_{k|k} + \varepsilon_k\right) && \text{(7b)}\\
&= \theta_{k+1}\hat{x}_{k+1|k+1} + (1-\theta_{k+1})\,\bullet\\
&\qquad \bullet \left[\left(\hat{x}_{k+1|k+1} + \varphi_{k+1}\tilde{e}_{k+1}\right) + (A - BL_k)\,\varepsilon_k\right] && \text{(7c)}\\
&= \hat{x}_{k+1|k+1} + (1-\theta_k)\left[\varphi_{k+1}\tilde{e}_{k+1} + (A - BL_k)\,\varepsilon_k\right] && \text{(7d)}\\
&= \hat{x}_{k+1|k+1} + (1-\theta_k)\,\varepsilon_{k+1} && \text{(7e)}
\end{aligned}
$$

Using this new set of variables, the state and the state estimation variables can be written as:

$$
\begin{aligned}
x_{k+1} &= Ax_k - BL_k t_k + w_k && \text{(8a)}\\
\hat{x}_{k+1|k} &= A\hat{x}_{k|k} - BL_k\hat{t}_k && \text{(8b)}\\
\hat{t}_{k+1} &= \bar{\theta}\hat{x}_{k+1|k} + (1-\bar{\theta})(A - BL_k)\,\hat{t}_k && \text{(8c)}\\
\tilde{e}_k &= (A - BL_{k-1})\hat{x}_{k-1|k-1} - \hat{x}_{k|k} && \text{(8d)}\\
\varepsilon_{k+1} &= (A - BL_k)\varepsilon_k + \varphi_{k+1}\tilde{e}_{k+1} && \text{(8e)}
\end{aligned}
$$

which implies that $e_{k+1|k} = \hat{x}_{k+1|k} - x_k$ is equal to

$$
e_{k+1|k} = Ae_{k|k} - w_k - \left(\bar{\theta} - \theta_k\right) BL_k\varepsilon_k. \qquad (9)
$$

The error auto-covariance matrix, i.e. $\mathbb{E}\left[e_{k+1|k}e'_{k+1|k}\right]$, is therefore

$$
P_{k+1|k} = APA' + Q + \bar{\theta}\left(1-\bar{\theta}\right) BL_k\varepsilon_k\varepsilon'_k\left(BL_k\right)' \qquad (10)
$$

in which there are no terms in $\mathbb{E}\left[e_{k+1|k}\varepsilon'_k\right] = 0$ despite the correlation of the two variables being non-zero, because all such terms come multiplied by $\mathbb{E}\left[\left(\bar{\theta} - \theta_k\right)\right]$ which is zero by definition.

Under TCP-like protocols, it is always known whether or not a message was delivered. Therefore, even though TCP-like protocols have a similar mechanism (involving ε), the applied value is always known. Hence, the innovation step is equal to the innovation step of the standard Kalman Filter.

$$
P_{k+1|k} = AP_{k|k}A' + Q_k \qquad (11)
$$

2) Correction: The correction step is independent of the type of protocol used to exchange messages between the controller and the actuator, because it depends only of the sensor-controller message transfer, and at the time of the correction the controller knows whether or not it has received a new sensor message. The correction is made only when a sensor message is received and is equal to the correction step of a standard Kalman filter, i.e.,

$$
\begin{aligned}
K_{k+1} &= \varphi_{k+1}P_{k+1|k}C'(CP_{k+1|k}C' + R_k) && \text{(12a)}\\
P_{k+1|k+1} &= (I_n - K_{k+1}C')P_{k+1|k} && \text{(12b)}\\
\hat{x}_{k+1|k+1} &= \hat{x}_{k+1|k} + K_{k+1}(y_{k+1} - C\hat{x}_{k+1|k}) && \text{(12c)}
\end{aligned}
$$

Note that if $\varphi_{k+1} = 0$, then Equation (12) implies that no correction is made, i.e., $\hat{x}_{k+1|k+1} = \hat{x}_{k+1|k}$.

B. Optimal Control over Lossy Networks — TCP-Like Protocols

Thanks to the existence of acknowledgments, the general principles behind the dynamic programming theory are valid for TCP-like protocols. It is important to notice that in this case: 1) the controller knows the inner state of the actuator and its inputs, thus is able to determine the outputted value, 2) once a controller message is delivered, its respective values will be outputted until the reception of a new controller message and 3) the certainty equivalence principle is still valid since the error does not depend on the control variable and according to [19] whenever this condition is verified, the certainty equivalence is valid. Taking these three notes in consideration, the computation of new controller values under the proposed scheme is no different than the computation of such values under classical optimal control, which is presented below.

Let $J_k(x_k)$ be defined as:

$$
J_k(x_k) = x'_N W_N x_N + \sum_{j=k}^{N-1}(x'_j W_j x_j + u'_j U_j u_j) \qquad (13)
$$

with k and N the limits of the minimization window. Lets also define the value function $V_k(x) = x'_k P_k x_k + c_k$ as the usual value function in which $V_k(x_k) = J_k(x_k)$, also known as the cost-to-go function. Then using the Bellman-Hamilton-Jacobi theorem, $V_k(x_k)$ can be written as

$$V_k(x) = x_k' W_k x_k + c_{k+1} + \min_{u_k}\Big(u_k' U_k u_k + (Ax_k + Bu_k + w_k)' P_{k+1}(Ax_k + Bu_k + w_k)\Big) \tag{14}$$

At each control iteration, the controller produces a set of values that are assumed to be applied according to the description provided above. This set of rules that are pre-programmed into the various networked control elements ensures that a given control value is applied only if the previous control value was applied, as discussed above.

The last paragraph plus the fact that the separation principle holds for TCP-like protocols implies that the minimization can ignore the network side and simply minimize in order to u_k, i.e.

$$\frac{\partial V_k(x_k)}{2\partial u_k} = u_k' U_k + (Ax_k + Bu_k + w_k)' P_{k+1} B = 0 \tag{15}$$

The last equation was equated to zero because it is known that the minimum is an extremal of $V_k(x_k)$. Furthermore, it is solved into (after transposition)

$$u_k^* = -\left(B' P_{k+1} B + U_k\right)^{-1} B' P_{k+1} A x_k. \tag{16}$$

The previous equation implicitly defines L_k. The equation defines the optimal controller in terms of the actual state variable as opposed to the estimation of the state variable. This is allowed due to the certainty equivalence principle which is never broken under TCP-like protocols.

Subtracting an appropriate form of Equation (15) from Equation (14) yields

$$V_k(x_k) - \frac{\partial V_k(x_k)}{2\partial u_k} u_k = x_k' W_k x_k + c_{k+1} + u_k' U_k u_k + (Ax_k - BL_k x_k + w_k)' P_{k+1}(Ax_k + w_k). \tag{17}$$

Since $\mathbb{E}\left[x_k w_k'\right] = 0$ by assumption (otherwise the system would either be non-causal or it would violate the assumption that w_k is a white noise sequence) and $\mathbb{E}\left[w_k' P_{k+1} w_k\right] = \mathrm{trace}(QP_{k+1})$, then a comparison of the terms in the definition of $V_k(x_k)$ with Equation (17) implies that:

$$P_k = W_k + (A - BL_k)' P_{k+1} A \tag{18a}$$

$$c_k = c_{k+1} + trace\left(QP_{k+1}\right) \tag{18b}$$

with the boundary conditions $P_N = W_N$ and $c_N = 0$.

Note that the controller presented above is structural very close to the centralized that can be found for example in [20].

C. Optimal Control Over Lossy Networks — UDP-Like Protocols

The computation of optimal control values under UDP-like protocols is normally more complex than the computation of optimal control values under TCP-like problems. As pointed out in the introduction section, under this type of protocol the separation and certainty equivalence principles are not guaranteed to hold. Furthermore, when these principles do not hold, the minimization problem from which the optimal control value stems is no longer quadratic.

However, as shown in Section IV, $e_{k|k-1} = \hat{x}_{k|k} - x_k$ and consequently $P_{k|k-1} = \mathbb{E}\left[e_{k|k-1} e_{k|k-1}'\right]$ are independent of x_k even over UDP-like protocols. In fact, it is possible to prove that some of the other extended inputs techniques discussed in Section II also have this property.

Having the above mentioned property is a sufficient condition for the certainty equivalence principle to hold, as shown in [19]. This implies that the optimal controller can be derived as the *classical* optimal controller, as shown in the previous subsection.

Nevertheless, the actual control values will be different stemming from the fact that they are computed from different state estimates, since they, for example, have different error auto-covariance matrices.

V. Simulations And Results

In order to evaluate the validity of the methods proposed in this paper, two sets of simulations were performed. The first set was centered in the validation of the method per se, and to show its benefits when compared to other methods, whereas the second set of simulations is a remake of a simulation performed in another paper [2] which was modified in order to include the methods presented herein. Both simulations were made using the `Matlab`®. The first set of simulations used were performed in a system that had the parameters of the discrete-time dynamic (see Equation (2)):

$$A = \begin{bmatrix} 1.2 & 0.8 \\ 0.5 & 0.9 \end{bmatrix} \tag{19a}$$

$$B = [1.1 \ 0.9]' \tag{19b}$$

$$C = [-0.9 \ 1.1] \tag{19c}$$

$$\mathbb{E}\left[[w_k \ v_k][w_j \ v_j]'\right] = \begin{bmatrix} 1.2 & 0.7 & 0.00 \\ 0.7 & 1.1 & 0.00 \\ 0.0 & 0.0 & 0.60 \end{bmatrix} \delta_{kj} \tag{19d}$$

The simulated system had poles $(\mathrm{z-plane})$ at 1.7 and 0.4. The simulations, in essence, varied the transmission error probability. For each transmission error probability there were made $2.3\mathrm{e}5$ simulations to filter out the stochastic nature of each run. At each run, there were made 300 steps, i.e., k varied from 0 at initialization to 300 at the end. The system had an initial value of $x_0 = [1.32,\ 0.97]'$, chosen once randomly, but used in all experiments, to increase the degree of repeatability of the experiments, though it ended up proving itself of low value given the rather long duration of each run.

The cost matrices were

$$W = \begin{bmatrix} 0.81 & -0.99 \\ -0.99 & 1.21 \end{bmatrix} \tag{20a}$$

$$U = 0.70 \tag{20b}$$

where W and U were randomly selected.

The TCP-like case was simulated both with the approach proposed here and with the output zero strategy, i.e. $u_k = \theta z_k$. However, it was not simulated for the *hold* case. That's because the optimal controller for the output zero strategy was already presented in the literature (for example [2]) whereas

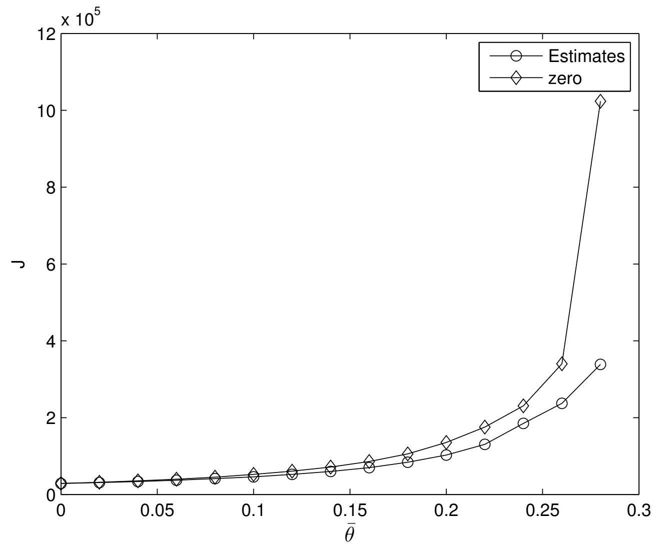

Fig. 3. Output Zero versus Proposed Approach under TCP-like Protocols.

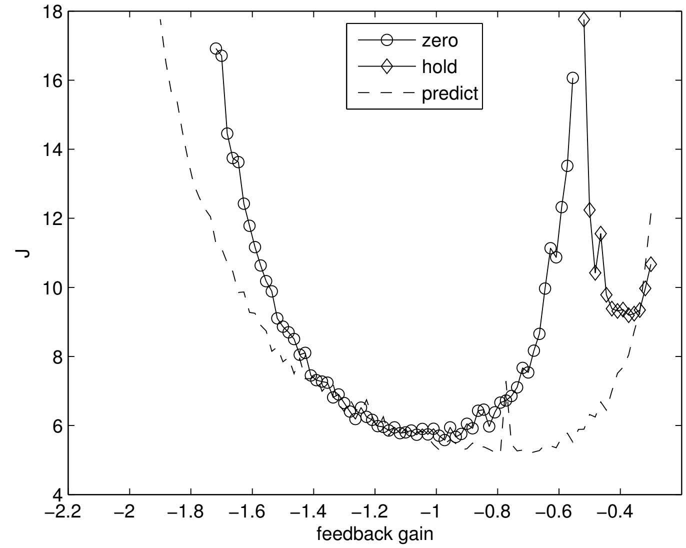

Fig. 5. Output strategies comparison

the *hold* case, to the best of the authors knowledge, has never been extensively studied. The UDP-like protocol was simulated only with the solution provided in this paper, for reasons similar to the previous one, i.e. other output methods do not provide any clue on optimal control values.

Figure 3 compares the behaviour of the output zero strategy with the strategy proposed herein, for TCP-like protocols. The graph is shown with a (natural) logarithmic y-axis due to the rather wide range of total cost J. At low error probabilities the performance of both strategies is indistinguishable. However, as the error probability grows the performance difference grows dramatically, in favour of the estimate-based approach proposed in this paper.

Figure 4 compares the behaviour of the strategy proposed herein for the TCP and UDP-like protocols. Once again, at low error probabilities both methods are indistinguishable. However, as the error probability increases the differences become evident. It is unfortunate, however, that the authors could not find another proposals in the literature claiming optimality under UDP-like protocols for a fair comparison.

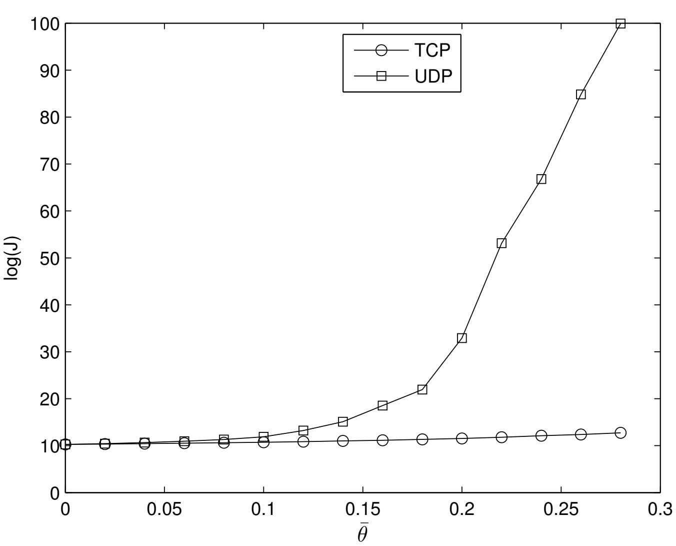

Fig. 4. TCP-like versus UDP-like Protocols for Estimation Output Extensions.

A. Comparison with other solutions

The second set of experiments is based on a comparison previously made in [2] between the zero and hold strategies, extended with the prediction-based method presented herein.

It consists in a first order discrete-time system, in which $\mathrm{A} = 1.2$, $\mathrm{B} = \mathrm{C} = 1$ and $\mathrm{Q} = \mathrm{R} = 1$. In the original experiment the authors considered $\bar{\varphi} = 1$, i.e. no errors in sensor to controller transmissions, and $\bar{\theta} = 0.5$. This is unrealistic since any source of errors in one segment of the network is very likely to affects the other segment as well, with a similar probability. Therefore, in this paper we considered $\bar{\varphi} = \bar{\theta} = 0.5$.

As previously discussed in the rationale section, the hold strategy performs best when the discrete-time closed-loop system has all of its poles close to 1, because in this situations the control value does not vary significantly between successive control periods. For a similar reason, the output zero strategy performs better when the closed loop system has all of its poles close to origin of the $\mathrm{z-plan}$, because in this situations the control value is almost always close to zero, thus outputting zero approximates the optimal control value.

As shown in Figure 5, the performance of the zero and hold strategies depends strongly on the feedback gain, as expected, since the position of the poles depends on this parameter. It should also be remarked the high sensibility to the feedback gain. Thus, minimizing the ISE is hard, mainly for practical systems where the exact parameter values are often difficult of even impossible to obtain.

The prediction case presents a very different behavior. Firstly it can be seen that it always offers a lower ISE than any of the other approaches, as claimed. In addition, the ISE is maintained near the minimum for a large range of feedback gain values, thus making the system much more amenable to use in practical scenarios. Therefore, it can be concluded that, in the presence of communication errors, the use of the prediction-based strategy, proposed in this paper, outperforms the zero and hold strategies in terms of ISE and is easier to use in practice due to its lower sensibility to feedback gain.

VI. Summary and Conclusions

The performance of distributed control systems is negatively affected by the presence of communication errors. Classical approaches to deal with missing actuation data consist in letting the controller output zero or the previous control value. This paper shows that these techniques only perform well in particular circumstances and are hard to use in practice, due to their high sensibility to feedback gain. To address these limitations, this paper proposes a novel prediction-based technique, in which the controller sends the current actuation value as well as a set of future estimates, which are used only in case of communication errors. This approach ensures that the value that is outputted by the actuator is as close as possible to the value produced by the controller, even in the presence of network errors.

The basic mechanisms and theoretical foundation of the novel prediction-based technique are presented in the paper. Simulation results show the feasibility of the proposed technique as well as that it outperforms classical approaches in terms of ISE, exhibiting at the same time a significantly lower sensitivity to feedback gain, thus being more amenable to use in practical scenarios.

References

[1] O. C. Imer, S. Yüksel, and T. Başar, "Optimal control of lti systems over unreliable communication links," *Automatica*, vol. 42, pp. 1429–1439, September 2006. [Online]. Available: http://portal.acm.org/citation.cfm?id=1327542.1327723

[2] 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, pp. 163 –187, jan. 2007.

[3] C. E. Shannon, "A mathematical theory of communication," *Bell System Technical Journal*, vol. 27, p. 379423 & 623656, July 1948.

[4] J. Hespanha, A. Ortega, and L. Vasudevan, "Towards the control of linear systems with minimum bit-rate," in *In Proc. of the Int. Symp. on the Mathematical Theory of Networks and Syst*, 2002.

[5] J. P. Hespanha, P. Naghshtabrizi, and Y. Xu, "A survey of recent results in networked control systems," in *Proceedings of the IEEE*, 2007, pp. 138–162.

[6] B. Azimi-sadjadi, "Stability of networked control systems in the presence of packet losses," in *in Proceedings of 2003 IEEE Conference on Decision and Control*, 2003, pp. 676–681.

[7] D. P. Bertsekas, *Dynamic Programming and Optimal Control, Vol. II*, 3rd ed. Athena Scientific, 2007.

[8] B. W. Karl J. Astrom, *Computer Controlled Systems: Theory and Design*, 3rd ed. Prentice Hall, 1997.

[9] S. Hu and W.-Y. Yan, "Stability robustness of networked control systems with respect to packet loss," *Automatica*, vol. 43, no. 7, pp. 1243–1248, Jul. 2007. [Online]. Available: http://dx.doi.org/10.1016/j.automatica.2006.12.020

[10] S. Hu and W. Y. Yan, "Stability of networked control systems under a multiple-packet transmission policy," *Automatic Control, IEEE Transactions on*, vol. 53, no. 7, pp. 1706–1711, 2008.

[11] X. Koutsoukos, N. Kottenstette, J. Hall, P. Antsaklis, and J. Sztipanovits, "Passivity-based control design of cyberphysical systems," in *In International Workshop on CyberPhysical Systems - Challenges and Applications (CPSCA08*, 2008.

[12] E. Garone, B. Sinopoli, and A. Casavola, "Lqg control over lossy tcp-like networks with probabilistic packet acknowledgements," in *Decision and Control, 2008. CDC 2008. 47th IEEE Conference on*, 2008, pp. 2686–2691.

[13] M. Kogel and R. Findeisen, "Robust suboptimal control over lossy networks using extended input schemes," in *Computer-Aided Control System Design (CACSD), 2011 IEEE International Symposium on*, 2011, pp. 210–215.

[14] T. M. P. Gommans, W. P. M. H. Heemels, N. Bauer, and N. Van de Wouw, "Compensation-based control for lossy communication networks," in *American Control Conference (ACC), 2012*, 2012, pp. 2854–2859.

[15] C. Robinson and P. Kumar, "Control over networks of unreliable links: Controller location and performance bounds," in *Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks and Workshops, 2007. WiOpt 2007. 5th International Symposium on*, 2007, pp. 1–8.

[16] R. Muradore, D. Quaglia, and P. Fiorini, "Predictive control of networked control systems over differentiated services lossy networks," in *Design, Automation Test in Europe Conference Exhibition (DATE), 2012*, 2012, pp. 1245–1250.

[17] M. Epstein, L. Shi, and R. Murray, "Estimation schemes for networked control systems using udp-like communication," in *Decision and Control, 2007 46th IEEE Conference on*, 2007, pp. 3945–3951.

[18] A. Vakili and B. Hassibi, "On the steady-state performance of kalman filtering with intermittent observations for stable systems," in *Decision and Control, 2009 held jointly with the 2009 28th Chinese Control Conference. CDC/CCC 2009. Proceedings of the 48th IEEE Conference on*, 2009, pp. 6847–6852.

[19] E. Tse and Y. Bar-Shalom, "Generalized certainty equivalence and dual effect in stochastic control," *Automatic Control, IEEE Transactions on*, vol. 20, no. 6, pp. 817 – 819, Dec 1975.

[20] D. E. Kirk, *Optimal Control Theory: An Introduction*. Dover Books on Electrical Engineering, 2004 (republished).