

Predictive Compensation for Communication Outages in Networked Control Systems

Erik Henriksson

Henrik Sandberg

Karl Henrik Johansson

Abstract—A predictive outage compensator co-located with the actuator node in a networked control system can be used to counteract unpredictable losses of data in the feedback control loop. When a new control command is not received at the actuator node at an appropriate time instance, the predictive outage compensator suggests a replacement command based on the history of past control commands. It is shown that a simple tuning phase together with the monitoring of the control history can lead to a compensator that can improve the closed-loop control performance under communication outages considerably compared to traditional schemes. Worst case performance bounds are given that relate the quality of the tuning phase and the complexity of the compensator with the length of the communication outage period. Zero-order-hold (holding the past control command if the current is lost) and applying an a priori decided constant signal (using a predefined value on the control command if the current is lost) are special cases of the more general compensation scheme presented. The predictive outage compensator is illustrated through computer simulation with communication outages.

I. INTRODUCTION

Communication networks are commonly used in distributed control systems since the seventies [1]. The recent introduction of wireless technology has led to new challenges due to the large variations in reliability and quality that radio links impose. These problems have recently been tackled through the design of communication protocols suitable for control (e.g., [2], [3], [4]), through network aware compensation schemes for control and estimation algorithms (e.g., [5], [6], [7]), and through joint communication and control designs (e.g., [8], [9], [10]).

In this paper we focus on the problem of communication outages in networked control systems. These outages correspond to short time intervals during which sensor data do not reach the controller node or control commands do not reach the actuator node. They are due to variations in radio conditions, because of moving objects, interference etc. Typical scenarios in industrial control settings are reported in [2]. It is hard to prevent communication outages to occur, and it is difficult to provide accurate stochastic models for them. Stationary models commonly used in the literature on networked control can be hard to justify in practice [6], [7].

A motivating scenario for the contribution of this paper is the control of the floatation tanks in an ore concentrator

This work was supported by the European Integrated Project SOCRADES, the Swedish Research Council, and the Swedish Foundation for Strategic Research.

E. Henriksson, H. Sandberg and K. H. Johansson are with the ACCESS Linnaeus Center, School of Electrical Engineering, Royal Institute of Technology, 10044 Stockholm, Sweden {erike02, hsan, kallej}@ee.kth.se

at Boliden in Sweden. This system is being investigated within the SOCRADES project [11]. It consists of four tanks in series and is today controlled using four individual PI-controllers [12]. The control objective is to maintain stable levels in all tanks, compensating for fluctuations in inflow and for load disturbances. For this process we are interested in replacing the wired level and flow sensors with wireless sensors. In doing so, it is desirable not to have to change the overall control structure or not even the control parameter tuning. We look instead for a solution where some additional compensation is done at the actuator node, but no other changes are needed in the closed-loop system.

The main contribution of this paper is a new predictive outage compensator (POC) co-located with the actuator nodes in a networked control system. The POC counteracts unpredictable losses of data in the feedback control loop. No modifications to the existing controller implementation is needed, which is a desirable feature in many practical systems. The POC suggests a replacement command based on the history of past control commands. So by a simple tuning phase together with the monitoring of the control history, the closed-loop control performance under communication outages can be considerably improved. Predictive control has been extensively used in various networked control settings (e.g., [13]), but we believe that our study on the complexity and actuator implementation of the compensator scheme is new.

The outline of the paper is as follows. In Section II the general idea of the POC is given. Section III describes the procedure to use it. Section IV shows and describes POC variants. Qualitative bounds for these are given in Section V. The POC methodology is then evaluated in Section VI. Concluding discussions and future work are given in Section VII.

II. PREDICTIVE OUTAGE COMPENSATION

The proposed POC is a generalization of the communication outage compensation algorithms used today such as zero-order-hold, applying constant outputs, and is related to a generalized hold function (e.g., [14]). The general idea is to monitor the control signal and use a signal model to extrapolate the signal in the event of a communication outage. The proposed control setup is shown in Fig. 1. The POC listens to the control signal sent from the controller. If the signal is received, the POC passes the control signal forward to the actuator and updates its own internal states using the received signal. In the case that no control action is received the POC uses its internal model to extrapolate the control signal based on the signal model and previous

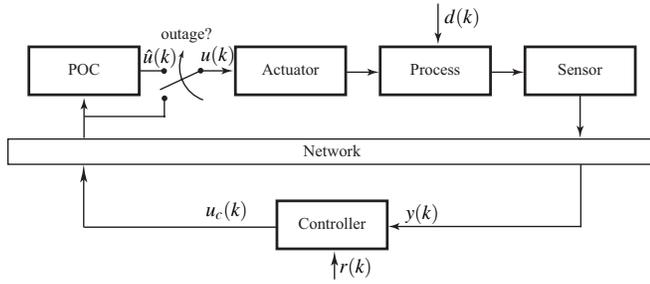


Fig. 1. Networked Control System with Predictive Outage Compensator (POC)

received data. An outage in the sensor signal link can be treated using the same POC simply by instructing the controller not to send a control command in the case that the corresponding sensor signal is lost. Since no control command is received at the POC when the sensor signal is lost, it will act and work in the same way as usual when a control signal is lost, regardless of where the outage occurred.

A. System Description

First we define the signals and transfer functions describing the system. In the system modelling it is assumed that the disturbances entering the system are load disturbances, here denoted by $d(k)$. The system output is given by

$$y(k) = P(q)(u(k) + d(k)) = \frac{B(q)}{A(q)}(u(k) + d(k))$$

where $B(q)$ and $A(q)$ are polynomials of degree n_B and n_A respectively and q denotes the one step forward shift operator. The controller output is given by

$$u_c(k) = C(q)(r(k) - y(k)) = \frac{S(q)}{R(q)}(r(k) - y(k))$$

where $S(q)$ and $R(q)$ are polynomials of degree n_S and n_R respectively.

B. Predictive Outage Compensator

The POC control signal estimate is given by

$$\hat{u}(k) = G(q)\hat{d}(k) = \frac{E(q)}{F(q)}\hat{d}(k) \quad (1)$$

where $E(q)$ and $F(q)$ are polynomials of degree n_E and n_F respectively and $\hat{d}(k)$ can be interpreted as a virtual disturbance. The control action actuated on the plant is decided by the switching mechanism inside the POC, given by.

$$u(k) = \begin{cases} u_c(k) & \text{Command from controller received} \\ \hat{u}(k) & \text{Command from controller lost} \end{cases}$$

C. Closed Loop Models

We will from this point onwards for the sake of simplicity assume that the reference is $r(k) \equiv 0$. This can be done without loss of generality since the influence of the reference can be modelled in the load disturbance. Using this we can conclude that when communication is working, i.e. the loop

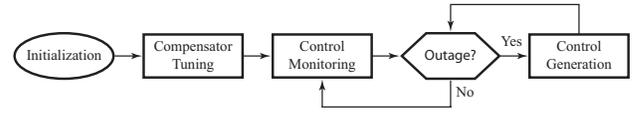


Fig. 2. Flow diagram describing the POC

is closed, this gives the system output (2) and the controller signal (3).

$$(A(q)R(q) + B(q)S(q))y(k) = B(q)R(q)d(k) \quad (2)$$

$$(A(q)R(q) + B(q)S(q))u_c(k) = -B(q)S(q)d(k) \quad (3)$$

We introduce the transfer function from $d(k)$ to $u_c(k)$ (4), compare with (3), where $E^0(q)$ and $F^0(q)$ are polynomials of degree n_{E0} and n_{F0} . Notice that (4) is $-T(q)$, where $T(q)$ is the complementary sensitivity function for the closed loop system.

$$u_c(k) = -\frac{B(q)S(q)}{A(q)R(q) + B(q)S(q)}d(k) = \frac{E^0(q)}{F^0(q)}d(k) \quad (4)$$

When communication is lost, i.e. the loop is open, the system output is as

$$y(k) = P(q)(\hat{u}(k) + d(k)) \quad (5a)$$

$$F(q)A(q)y(k) = B(q)E(q)\hat{d}(k) + B(q)F(q)d(k) \quad (5b)$$

By comparing (1) and (4) one can see that for a smart choice of $F(q)$, $E(q)$ and $\hat{d}(k)$, and under some assumptions on the reference and disturbances, we should be able to get similar or same behavior of the system output when communication is lost as when it is functioning. Since $E(q)/F(q)$ should describe $-T(q)$, i.e. the negated complementary sensitivity function, it should be a well-behaved function with lowpass characteristics, given a reasonable control design. However, when communication is lost the system is operated in open-loop and unstable systems are therefore inherently difficult to handle.

III. PREDICTIVE COMPENSATION PROCEDURE

The method used when designing and running the proposed POC is given in Fig. 2 which shows the different parts of the work flow in commissioning and running the proposed algorithm.

A. Initialization

When first commissioning the POC one needs to make some initial design decisions. The first is to decide the order of $E(q)$ and $F(q)$. The choice of model order will later show to be a key factor in the POC performance as it turns out that zero-order-hold and applying constant signals are special cases of the general POC. Another design parameter is the methodology for detecting packet losses. This problem is left outside the scope of this paper, however it is assumed to be directly detectable if a packet has been lost or not.

B. Compensator Tuning

The next stage is to tune the POC by identifying $E(q)$ and $F(q)$ according to the previously decided model order. The POC tuning is based on conventional recursive identification [15]. The identification problem is not treated further in the paper, instead $E(q)$ and $F(q)$ are assumed to have been identified using an appropriate algorithm. The identification experiment can be done by adding an excitation signal to the control signal and measure the response from the controller at the location of the POC.

C. Control Monitoring

This step is the core part of the POC. During this phase the POC listens to the received control signal $u_c(k)$ and uses it to filter out an estimate of $\hat{d}(k)$ as

$$\hat{d}(k) = \frac{F(q)}{E(q)} u_c(k)$$

or more precisely shifted as $q^{-n_F} E(q) \hat{d}(k) = q^{-n_F} F(q) u_c(k)$ which after rearranging gives

$$\begin{aligned} e_0 \hat{d}(k + (n_E - n_F)) &= \\ &= u_c(k) + \dots + f_{n_F} u_c(k - n_F) - \\ &\quad - e_1 \hat{d}(k - n_F - 1 + n_E) - \dots - e_{n_E} \hat{d}(k - n_F) \end{aligned}$$

where e_i and f_i are the coefficients of $E(q)$ and $F(q)$ respectively. As seen, what one can actually estimate is $\hat{d}(k)$ shifted back $n_F - n_E$ which is the relative degree of the POC.

D. Outage Control Generation

When an outage occur an estimate of $\hat{d}(k)$ is used to drive the POC filter, in order to extrapolate the signal and provide a control signal to actuate. When communication is lost and we no longer receive any control signals, we can no longer estimate $\hat{d}(k)$ using $u_c(k)$. Instead we have to decide on a new policy. This can be done in several different manners. In this paper we use the method to hold the last known $\hat{d}(k)$ which we denote by \bar{d} , another policy is to set $\bar{d} = 0$. The general POC is now given by

$$\hat{u}(k) = \frac{E(q)}{F(q)} \bar{d} \quad (6)$$

again shifted as $q^{-n_F} F(q) \hat{u}(k) = q^{-n_F} E(q) \bar{d}$ which in the same way as before gives

$$\begin{aligned} \hat{u}(k) &= (e_0 + \dots + e_{n_E}) \bar{d} - \\ &\quad - f_1 \hat{u}(k-1) - \dots - f_{n_F} \hat{u}(k-n_F) \end{aligned}$$

if the true value for u_c is known for any time instant in the recursion, the true value is used instead of the predicted \hat{u} .

IV. POC INTERNAL MODEL COMPLEXITY

Within the proposed framework we have the freedom to choose different complexity for the general POC (6). Next we highlight three special cases.

Zero POC: This is the simplest version of outage compensation. If no new control signal arrives to the POC an a priori decided constant command, zero say, is actuated. This corresponds to setting $F(q) = 1$ and $E(q) = 0$ in (6) so that $\hat{u}(k) = 0$.

Hold POC: Another common version of outage compensation is zero-order-hold in which the action when no command is received is to keep the last received value. In this framework it corresponds to choosing $F(q) = q - 1$ and $E(q) = 0$ giving $(q - 1)\hat{u}(k) = 0$ or shifted back in time, $\hat{u}(k) = \hat{u}(k - 1)$.

Optimal POC: The best POC one can use in this framework is to choose $F(q) = F^0(q)$ and $E(q) = E^0(q)$, i.e. their true value. This corresponds to $F^0(q)\hat{u}(k) = E^0(q)\bar{d}$

V. WORST-CASE ERROR BOUNDS FOR PREDICTIVE OUTAGE COMPENSATOR

In order to bound the error in the POC model, the following lemma is useful.

Lemma 1: Consider the linear time-invariant input-output model

$$\delta(k) = \sum_{j=k_0}^k \gamma(k-j)\rho(j), \quad k \geq k_0,$$

with impulse response $\gamma(j)$. It holds that

$$|\delta(k)| \leq \left(\sum_{j=0}^{k-k_0} |\gamma(j)| \right) \max_{k_0 \leq j \leq k} |\rho(j)|.$$

For bounded input over the interval $[k_0, k_f]$, the maximum output over the same interval is bounded by

$$\max_{k_0 \leq k \leq k_f} |\delta(k)| \leq \left(\sum_{j=0}^{k_f-k_0} |\gamma(j)| \right) \max_{k_0 \leq k \leq k_f} |\rho(k)|.$$

Both bounds are tight, i.e., there is an input $\rho(k)$ that achieves equality.

Proof: Application of *Theorem 27.2* in [16]. ■

We are going to use the lemma to bound the difference between the ideal input from the nominal model, $u_c(k)$, and the input from the POC, $\hat{u}(k)$, when there is an outage. We assume that the outage occurs in the time interval $k \in [0, k_f]$, without loss of generality. First, we re-define the nominal model and the POC model in the following way:

$$u_c(k) = \frac{q^{n_{F0}-1}}{F^0(q)} p(k), \quad (q^{n_{F0}-1} p(k) = E^0(q) d(k)) \quad (7)$$

$$\hat{u}(k) = \frac{q^{n_F-1}}{F(q)} \hat{p}(k), \quad (q^{n_F-1} \hat{p}(k) = E(q) \hat{d}(k)) \quad (8)$$

where $\deg F = n_F$ and $\deg F^0 = n_{F0}$. The signal $p(k)$ can be thought of as a driving noise signal that contains everything in $u_c(k+1)$ that cannot be explained by a linear combination of $u_c(k), \dots, u_c(k-n_{F0}+1)$. The signal $\hat{p}(k)$ has a similar interpretation for $\hat{u}(k)$.

The model (7) can be realized in the state-space form

$$\begin{aligned} x(k+1) &= c(F^0)x(k) + Kp(k) \\ u_c(k) &= K^T x(k), \end{aligned}$$

where $c(F^0)$ is a companion matrix of the polynomial $F^0(q) = q^{n_{F0}} + f_1^0 q^{n_{F0}-1} + \dots + f_{n_{F0}}^0$,

$$c(F^0) = \begin{pmatrix} -f_1^0 & -f_2^0 & \dots & -f_{n_{F0}}^0 \\ 1 & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 0 \end{pmatrix} \in \mathbb{R}^{n_{F0} \times n_{F0}}$$

and

$$K = \begin{pmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix} \in \mathbb{R}^{n_{F0}}, \quad x(k) = \begin{pmatrix} u_c(k) \\ u_c(k-1) \\ \vdots \\ u_c(k-n_{F0}+1) \end{pmatrix} \in \mathbb{R}^{n_{F0}}.$$

The model (8) can be realized in exactly the same form using a companion matrix $c(F)$. If $\deg F < \deg F^0$, we can define the polynomial coefficients $f_{n_F+1} = \dots = f_{n_{F0}} = 0$ so that $c(F) \in \mathbb{R}^{n_{F0} \times n_{F0}}$, and both models have the same state dimension.

If the POC has been run in the control monitoring mode, as described earlier, and an outage occurs at $k=0$, then the signals $\hat{u}(k), u_c(k)$ when $k \in [0, k_f]$ are given by

$$\hat{u}(k) = K^T c(F)^k x(0) + \sum_{j=0}^{k-1} K^T c(F)^{k-j-1} K \hat{p}(j) \quad (9a)$$

$$u_c(k) = K^T c(F^0)^k x(0) + \sum_{j=0}^{k-1} K^T c(F^0)^{k-j-1} K p(j) \quad (9b)$$

Subtracting (9b) from (9a), the error between ideal and actual input in outage mode is given by

$$\begin{aligned} \hat{u}(k) - u_c(k) &= K^T [c(F)^k - c(F^0)^k] x(0) \\ &+ \sum_{j=0}^{k-1} K^T c(F)^{k-j-1} K \hat{p}(j) - \sum_{j=0}^{k-1} K^T c(F^0)^{k-j-1} K p(j). \end{aligned} \quad (10)$$

In order to derive simple expressions for the error bounds, the following assumptions are made. They should be relatively easy to verify for a given system.

Assumptions 1: It is assumed that both (7) and (8) are exponentially stable, i.e., there are constants $c_0 > 0$, $c > 0$, $0 \leq \lambda_0 \leq 1$, $0 \leq \lambda \leq 1$ such that $\|K^T c(F^0)^k\|_1 \leq c_0 \lambda_0^k$, $\|K^T c(F)^k\|_1 \leq c \lambda^k$. Where $\|\cdot\|_1$ is the 1-norm of a vector (sum of magnitude of elements). Furthermore, we assume the actual input and the innovations are bounded $|u_c(k)| \leq \rho_u$, $|p(k)| \leq \rho_p$, $\forall k$ and $|\hat{p}(k)| \leq \rho_{\hat{p}}$ for $k \geq 0$.

The constants λ and λ_0 are measures of how fast the systems are. Using the error model (10) and the assumptions we are going to analyze the error behavior. A simple example is also given at the end of this section.

A. Zero POC

In the zero POC, the input is simply set to zero in outage mode, i.e., $\hat{u}(k) = 0$, $k > 0$. In the model framework developed here, this means $\hat{u}(k+1) = 0 \cdot \hat{u}(k) + \hat{p}(k)$, that is $F(q) = 1$, together with the outage policy $\hat{p}(k) = 0$, $k \geq 0$. The error model (10) reduces to

$$\hat{u}(k) - u_c(k) = -K^T c(F^0)^k x(0) - \sum_{j=0}^{k-1} K^T c(F^0)^{k-j-1} K p(j),$$

for $k > 0$. Applying the triangle inequality and Lemma 1, we have

$$\begin{aligned} |\hat{u}(k) - u_c(k)| &\leq \rho_u \|K^T c(F^0)^k\|_1 + \rho_p \sum_{j=0}^{k-1} |K^T c(F^0)^{k-j-1} K| \\ &\leq \rho_u c_0 \lambda_0^k + \rho_p c_0 \frac{1 - \lambda_0^k}{1 - \lambda_0} =: \Gamma_0(\rho_u, \rho_p, k). \end{aligned}$$

The error bound Γ_0 converges exponentially fast with rate λ_0 to $\frac{\rho_p c_0}{1 - \lambda_0}$.

B. Hold POC

The hold POC can be modelled as before by the model $\hat{u}(k+1) = \hat{u}(k) + \hat{p}(k)$, that is $F(q) = q - 1$, together with the outage policy $\hat{p}(k) = 0$, $k \geq 0$. Hence, if there is an outage at $k=0$, we have $\hat{u}(k) = u_c(0)$, for $k \geq 0$. The error model (10) reduces to

$$\hat{u}(k) - u_c(k) = K^T [I - c(F^0)^k] x(0) - \sum_{j=0}^{k-1} K^T c(F^0)^{k-j-1} K p(j),$$

since $u_c(0) = K^T x(0)$. We make the following assumptions.

Assumptions 2: Assume there are constants $c_1 \geq c'_1$ such that $\|K^T [I - c(F^0)^k]\|_1 \leq c_1 - c'_1 \lambda_0^k$. Conservative choices for c_1, c'_1 that work under Assumptions 1 are $c_1 = 1 + c_0$ and $c'_1 = 0$.

Applying the assumptions, the triangle inequality, and Lemma 1, we have

$$\begin{aligned} |\hat{u}(k) - u_c(k)| &\leq \rho_u (c_1 - c'_1 \lambda_0^k) + \rho_p c_0 \frac{1 - \lambda_0^k}{1 - \lambda_0} \\ &=: \Gamma_1(\rho_u, \rho_p, k). \end{aligned}$$

The error bound Γ_1 converges exponentially fast to $\rho_u c_1 + \frac{\rho_p c_0}{1 - \lambda_0}$, at a rate λ_0 .

C. General POC

For the general POC,

$$\hat{u}(k+1) = -f_1 \hat{u}(k) - \dots - f_{n_F} \hat{u}(k - n_F + 1) + \hat{p}(k),$$

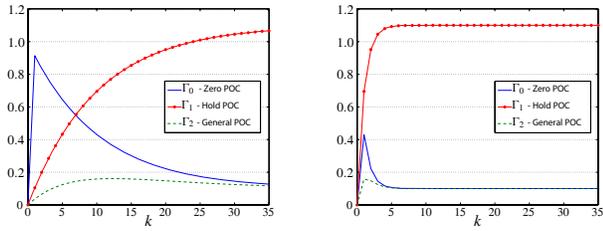
we make the following assumptions.

Assumptions 3: Assume there are constants $c_2 \geq c'_2$ and $1 \geq \lambda_2 \geq \lambda'_2 \geq 0$ such that $\|K^T [c(F)^k - c(F^0)^k]\|_1 \leq c_2 \lambda_2^k - c'_2 (\lambda'_2)^k$. Conservative choices that work under Assumptions 1 are $c_2 = c + c_0$, $\lambda_2 = \max\{\lambda, \lambda_0\}$, and $c'_2 = 0$.

The error bound is now

$$\begin{aligned} |\hat{u}(k) - u_c(k)| &\leq \rho_u (c_2 \lambda_2^k - c'_2 (\lambda'_2)^k) + \rho_p c_0 \frac{1 - \lambda_0^k}{1 - \lambda_0} \\ &+ \rho_{\hat{p}} c \frac{1 - \lambda^k}{1 - \lambda} =: \Gamma_2(\rho_u, \rho_p, \rho_{\hat{p}}, k) \end{aligned}$$

To make the worst-case bound Γ_2 small, it is clear that it is best to use a zero policy for $\hat{p}(k)$, i.e., $\rho_{\hat{p}} = 0$. It is important to remember that this is a strict worst-case analysis that assumes that we have no knowledge whatsoever of $p(k)$. If we have knowledge of how quickly $p(k)$ evolves, then it can be very beneficial to choose a nonzero $\hat{p}(k)$ to counteract it, as shall be seen in Section VI.



(a) Low bandwidth, $\omega_b h = 0.1$. (b) High bandwidth, $\omega_b h = 1.0$.

Fig. 3. The worst-case bounds for systems with different bandwidth.

D. Example

As described in Section II-C the transfer function from disturbance $d(k)$ to control $u_c(k)$ is given by the negated complementary sensitivity function. To illustrate the worst case bounds we use a simple example. Let us assume the feedback control system is operating well and is well designed so that it can be described by a first-order system (in continuous time) $-\omega_b/(s + \omega_b)$, i.e., the control system has bandwidth ω_b . A zero-order-hold sampled realization with sampling period h is $u_c(k+1) = \lambda_0 u_c(k) + p(k)$, where $\lambda_0 = e^{-\omega_b h}$, and $p(k) = (e^{-\omega_b h} - 1)d(k)$. As general POC, let us use the model $\hat{u}(k+1) = \lambda \hat{u}(k) + \hat{p}(k)$, where $\lambda = e^{-1.3\omega_b h}$. This means that we have overestimated the actual bandwidth of the system with 30% in the POC. The other constants in Assumptions 1 are $c_0 = c = 1$, and we assume control signals $u_c(k)$ must be smaller than one, $\rho_u = 1$, and that disturbances $d(k)$ are smaller than 0.1. This means that $\rho_p = 0.1(1 - e^{-\omega_b h})$. Based on the discussion in Section V-C, we also choose the zero policy for $\hat{p}(k)$, i.e., $\rho_{\hat{p}} = 0$. It is also easy to verify that the constants in Assumptions 2 and 3 can be chosen as $c_1 = c'_1 = 1$, $c_2 = c'_2 = 1$, $\lambda_2 = \lambda_0$, and $\lambda'_2 = \lambda$.

We plot the error bounds Γ_0 , Γ_1 , and Γ_2 as functions of outage time k in Figs. 3(a) and 3(b). In Fig. 3(a), the feedback control system is slow with $\omega_b h = 0.1$ and in Fig. 3(b) the system is fast with $\omega_b h = 1.0$. As can be seen, for the slow system, the general POC is best for all times, whereas the hold POC is better than the zero POC for outages shorter than seven samples. If the bandwidth is increased with a factor 10, then the hold POC is by far worst for all times. The reason is that the system is capable of very fast changes, and to hold a constant input can quickly push the system in the wrong direction. Also in the fast case is the general POC best for all times, even though the zero POC does quite well also.

The example has shown that a general POC can do much better than the traditional hold and zero POCs compensators both for slow and fast systems, even though the model had a parameter error of 30%.

VI. SIMULATION EVALUATION

To give intuition around the boundaries for the three POC versions in Section V for a more complex system, simulations are performed on a system consisting of a double integrator controlled with a lead-lag controller. The

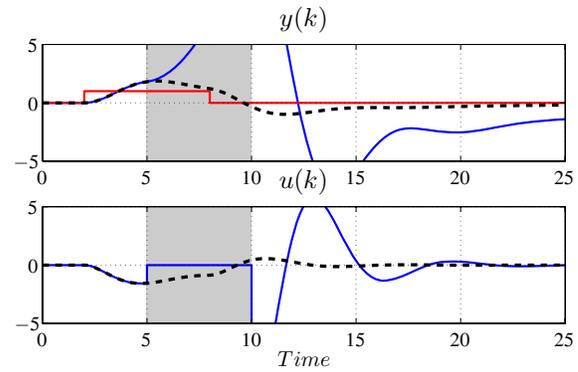


Fig. 4. Behavior of the zero POC policy (solid) compared to the Nominal system behavior (dashed)

continuous time controller is discretized using sampling and zero-order-hold with a sampling interval of $T_s = 0.1s$.

A. Simulation Scenario

The scenario considered in the simulation examples is the following:

- t = 0** The system starts at rest
- t = 2** A load disturbance with amplitude 1 hits the system
- t = 5** The communication between the controller and the actuator is lost and the backup policy is activated
- t = 8** The load disturbance disappears. Since there is no communication between controller and POC the disturbance is invisible and hence it can not be compensated for. This is a fundamental limitation for all compensators
- t = 10** Communication is restored and the controller starts to actuate the system back into rest

In the following three subsections we discuss the system behavior under this scenario for the three compared POC versions.

B. Zero POC

The response for the system with the zero POC can be seen in Fig. 4. As seen, when the outage occur, the system output instantly starts to grow rapidly, taking the system far away from the desired setpoint. As a consequence of the large perturbation caused during the outage the controller has to use a large control signal to stabilize the system once communication is restored.

C. Hold POC

The hold POC system response can be seen in Fig. 5. Compared to the zero POC the hold POC initially, for the first few samples after the outage, manage to keep the system quite close to the nominal trajectory. However, after these initial samples the system trajectory starts growing away from the setpoint, although not as fast as in the zero POC setup. When communication is restored the controller just as in the zero POC case needs to use a large control signal to stabilize the system again. The large magnitude of the control signal for the zero POC and the hold POC, once

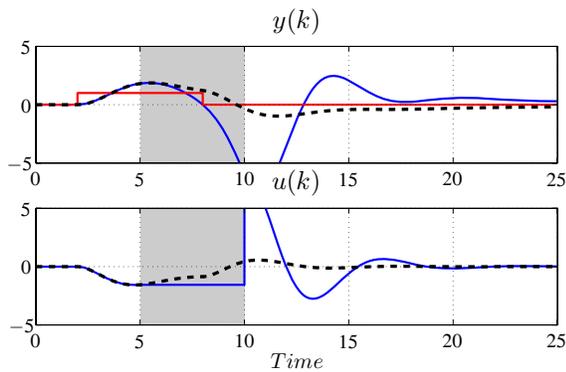


Fig. 5. Behavior of the hold POC policy (solid) compared to the Nominal system behavior (dashed)

communication is restored, is dependent on two factors. The first and most apparent reason is the fact that the system drifts far away from the setpoint and therefore a large signal is needed. The second and more subtle reason is that there is a large difference between the control signal that the controller is computing and the one that is actuated. This difference will cause wind-up effects in the controller which also appear in the transient after communication is restored.

D. General POC

The system response with the general POC is shown in Fig. 6. In this simulation scenario the internal model of the POC is chosen as the nominal system, giving the optimal POC. As seen the predicted trajectory follows the nominal trajectory exactly during the outage up until $t = 8s$ when the disturbance changes. This is the best we can do since we can only follow the nominal trajectory that was observed prior to the outage. As the disturbance changes so does the nominal trajectory. However since communication is lost the POC can not detect this change of nominal trajectory since it can not measure it.

When communication is restored the controller can use a small input correction term to get the system stabilized again after the outage. This is due, relating to Section VI-C, both to the fact that the drift compared to the nominal trajectory is small, causing a small control error, and that this in turn yields a much smaller wind-up term in the controller. We can also note that the settling time for the nominal system and the system using the optimal POC are almost identical.

VII. CONCLUSIONS AND FUTURE WORK

We have presented a new general methodology for compensating for communication losses in networked control systems. It has been shown that this method gives significantly improved performance compared to previously used compensation schemes without increasing the complexity too much.

The ongoing work consists of examining how measurements of the disturbances entering the system can be utilized to get even better performance and tighter bounds on the error modelling. Also work is currently being done on implementing the proposed algorithm on a physical laboratory

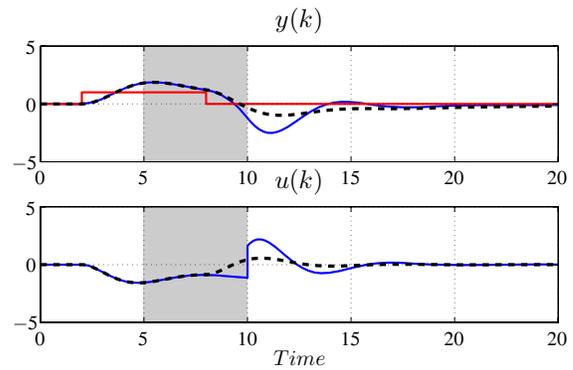


Fig. 6. Behavior of the optimal POC policy (solid) compared to the Nominal system behavior (dashed)

process as well as further developing the method for more general disturbances and systems.

REFERENCES

- [1] T. Samad, P. McLaughlin, and J. Lu, "System architecture for process automation: Review and trends," *Journal of Process Control*, vol. 17, pp. 191–201, 2007.
- [2] A. Willig, M. Kubisch, C. Hoene, and A. Wolisz, "Measurements of a wireless link in an industrial environment using an IEEE 802.11-compliant physical layer," *IEEE Transactions on Industrial Electronics*, vol. 49, no. 6, pp. 1265–1282, 2002.
- [3] X. Liu and A. Goldsmith, "Wireless network design for distributed control," in *Proceedings of the IEEE Conference on Decision and Control*, Atlantis, Paradise Island, Bahamas, 2004, pp. 2823–2829.
- [4] P. G. Park, C. Fischione, A. Bonivento, K. H. Johansson, and A. Sangiovanni-Vincentelli, "Breath: a self-adapting protocol for wireless sensor networks in control and automation," in *IEEE SECON*, San Francisco, CA, USA, 2008.
- [5] Q. Ling and M. Lemmon, "Optimal dropout compensation in networked control systems," in *Proceedings IEEE Conference on Decision and Control*, Maui, HI, USA, 2003.
- [6] J. Hespanha, P. Naghshtabrizi, and Y. Xu, "A survey of recent results in networked control systems," *Proceedings of the IEEE, special issue*, vol. 95, no. 1, pp. 138–192, 2007.
- [7] L. Schenato, B. Sinopoli, M. Franceschetti, K. Poolla, and S. Sastry, "Foundations of control and estimation over lossy networks," *Proceedings of the IEEE, special issue*, vol. 95, no. 1, pp. 163–187, 2007.
- [8] S. Graham and P. R. Kumar, "The convergence of control, communication, and computation," in pp. 458–475, *Proceedings of PWC 2003: Personal Wireless Communication, Lecture Notes in Computer Science, Volume 2775*. Springer-Verlag, Heidelberg, 2003.
- [9] L. Bao, M. Skoglund, and K. H. Johansson, "Encoder/decoder design for feedback control over the binary symmetric channel," in *IEEE International Symposium on Information Theory*, Seattle, WA, USA, 2006.
- [10] K.-E. Årzén, A. Bicchi, G. Dini, S. Hailles, K. H. Johansson, J. Lygeros, and A. Tzes, "A component-based approach to the design of networked control systems," *European Journal of Control*, vol. 2-3, pp. 261–279, 2007, Invited Paper.
- [11] SOCRADES, Integrated Project. EU 6th Framework Programme, <http://www.socrates.eu/>.
- [12] B. Stenlund and A. Medvedev, "Level control of cascade coupled flotation tanks," *Control Engineering Practice*, vol. 10, pp. 443–448, 2002.
- [13] A. Bemporad, "Predictive control of teleoperated constrained systems with unbounded communication delays," in *Proceedings IEEE Conference on Decision and Control*, Tampa, FL, December 1998.
- [14] W. Sun, K. M. Nagpal, and P. P. Khargonekar, " \mathcal{H}_∞ control and filtering for sampled-data systems," *IEEE Transactions on Automatic Control*, vol. 38, no. 8, August 1993.
- [15] L. Ljung and T. Söderström, *Theory and Practice of Recursive Identification*. The MIT Press, 1983.
- [16] W. J. Rugh, *Linear System Theory*, 2nd ed. Prentice-Hall, 1996.