2009 American Control Conference
Hyatt Regency Riverfront, St. Louis, MO, USA
June 10-12, 2009



# On the Synchronization Problem for the Stabilization of Networked Control Systems over Nondeterministic Networks

P. Varutti and R. Findeisen

***Abstract*—In recent years, networked control systems have gained the attention of the control community, since they allow to re-use the preexisting infrastructure therefore reducing deployment time and costs. Unfortunately, they also introduce new control challenges due to the nondeterministic network behavior. Predictive and model-based approaches can be used to compensate both delays and packet dropouts. However, a common time-frame among the involved components –sensors, actuators, plants and controllers– is required. This brings normally the necessity of keeping inner-clocks synchronized, which at the current state of art can be hard to realize. In this paper, a possible solution for the synchronization problem is presented. The idea is to keep using predictive techniques but utilizing a unique inner-clock on the system side. Assuming that actuator and sensor are directly connected to the system, the packets are time stamped, the delays are bounded, the maximum round-trip-time is known, and a limited amount of information is lost, it is possible to use model predictive control to stabilize the closed loop system by compensating delays and packet dropouts.**

***Index Terms*—nonlinear continuous systems, networked control systems, stability, nondeterministic networks, packet dropouts, nonlinear model predictive control**

## I. INTRODUCTION

Recent developments in communication technologies and the massive spread of wired and wireless networks have casted a new light on the way of interpreting closed loop control systems. The idea of dedicated communication channels between systems and controllers are leaving room to shared communication media like Internet. Networked Control Systems (NCSs) offer the major advantage of re-using a preexisting infrastructure and consequently reduce considerably startup time and costs. At the same time, they provide a robust framework to counteract component failures more efficiently thanks to components redundancy. Nevertheless, NCSs introduce a lot of new challenges, such as (nondeterministic) delays and/or (unpredictable) information losses. This can obviously reduce the system performance and also lead to instability.

At the state of art, most of the attention has been paid on linear NCSs, while only a few works have focused on nonlinear ones –see [1]–[6] for an overview on NCSs–. Model Predictive Control (MPC) has demonstrated to be a good tool to counteract delays, as shown inter alia in [7], [8]. In [8], a MPC approach able to guarantee asymptotic convergence under the presence of nondeterministic delays and packet dropouts was presented. However, as commonly assumed in the NCSs literature, the proposed solution requires a set of synchronized clocks among the components –controllers, actuators, sensors, and systems– in order to provide a common time-frame for solving the delays. Although this might work for slow dynamical systems, it becomes an issue for fast ones, since it might be hard to keep sufficiently highly precise synchronization.

In this paper, a possible way to avoid synchronization is presented. Considering a nonlinear system where actuator and sensor are directly incorporated into the system, an MPC scheme which does not require clocks synchronization is introduced. In this way, it can be easily applied both to slow and fast dynamical systems. The presented method can deal not only with nondeterministic delays in both the sensor- and the actuator-channel, but also with limited amounts of information losses.

In Section II, the overall problem is introduced. The proposed method is presented in Section III. Finally, simulation results on a continuous stirred tank reactor (CSTR) and an inverted pendulum on a cart are provided in Section IV.

P. Varutti and R. Findeisen are with the Institute of Automation Engineering, Otto-von-Guericke University, 39016 Magdeburg, Germany {paolo.varutti, rolf.findeisen}@ovgu.de

## II. PROBLEM STATEMENT

Consider the nonlinear continuous time system

$$\dot{x}(t) = f(x(t), u(t)), \quad x(0) = x_0, \tag{1}$$

$$x(t) \in X \subseteq \mathbb{R}^n, \text{ and } u(t) \in U \subset \mathbb{R}^m, \tag{2}$$

where (2) denote state and input constraints. It is assumed that $U$ is compact, $X$ is connected, and $(0,0) \in X \times U$. $f : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^n$ is locally Lipschitz continuous, and such that $f(0,0) = 0$. It is assumed that the whole state is available only a discrete instants $t_i \in \pi$, where $\pi$ is a time partition defined as follows:

**Definition (Partition):** Every series $\pi = (t_i)$, $i \in N$ of positive real numbers such that $t_0 = 0$, $t_i < t_{i+1}$ and $t_i \to \infty$ is called partition.

The objective is to stabilize the system around the origin under the state and input constraints (2), i.e. $\|x(t)\| \to 0$ for $t \to \infty$.

### A. Sampled-data MPC

Consider the nonlinear continuous time system (1)-(2). Sampled-data MPC is based on the repeated solution of an open loop control problem, based on the state measurement at the time $t_i$, under the constraints (2). The controller predicts the system behavior over a prediction horizon $T_p$, such that a specific objective functional is minimized. The procedure is repeated at every recalculation instant $t_i \in \pi$. This is mathematically formulated as[1]

[1]Assuming for simplicity that a minimum is obtained.

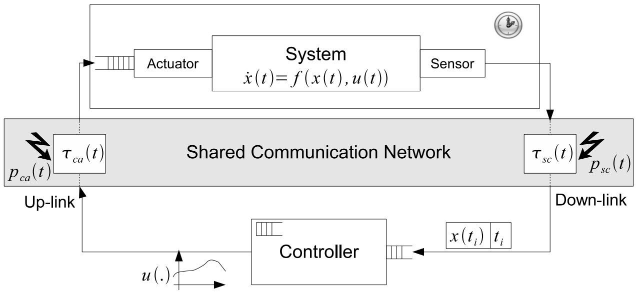


Fig. 1. Sketch of an NCS. The black arrows indicate that packet dropouts can occur in both links.

$$\min_{\overline{u}(\cdot)} \int_{t_i}^{t_i+T_p} F(\overline{x}(\tau),\overline{u}(\tau))d\tau + E(\overline{x}(t_i+T_p)), \tag{3a}$$

$$\text{s.t.} \quad \dot{\overline{x}}(t) = f(\overline{x}(t),\overline{u}(t)), \quad \overline{x}(t_i) = x(t_i), \tag{3b}$$
$$\overline{u}(t) \in U, \quad t \in [t_i, t_{i+1}), \tag{3c}$$
$$\overline{x}(t) \in X, \tag{3d}$$
$$\overline{x}(t_i+T_p) \in \mathscr{E}, \tag{3e}$$

where $\bar{\cdot}$ denotes the controller internal variables. It is assumed that the cost function $F : X \times U \rightarrow \mathbb{R}$ is locally Lipschitz continuous with $F(0,0) = 0$ and $F(x,u) > 0$, $\forall (x,u) \in X \times U \setminus (0,0)$. The obtained input is applied open loop in between recalculation instants, i.e.

$$u(\tau) = u^*(\tau; x(t_i)), \tau \in [t_i, t_i + \delta_c), \tag{4}$$

where $\delta_c$ is defined as:

**Definition (Recalculation time):** Given two consecutive recalculation instants $t_i$, $t_{i+1}$, the time interval between them $\delta_c = (t_{i+1} - t_i)$ is called *recalculation time*.

By properly choosing the cost functional $F(\cdot)$, the terminal cost $E(\cdot)$, the terminal region $\mathscr{E} \subset X$, and the prediction horizon $T_p$, stability of the closed loop can be achieved (refer to [9], [10] for more details).

### B. Control over Communication Networks

When the control loop is closed through a shared communication network the exchanged information can be subject to delays and packet dropouts. In Fig. 1 a sketch of an NCS is reported, where $\tau_{sc}(t)$ represents the measurement delays, while $\tau_{ca}(t)$ the actuation delays. The loss probabilities in the actuation and the measurement are respectively referred as $p_{ca}(t)$ and $p_{sc}(t)$. The following assumptions are made:

**Assumption 1:** All exchanged information is time-stamped.

**Assumption 2:** All delays are bounded, i.e.

$$\tau_{sc}(t) \in [0,\ \tau_{sc}^{max}], \text{ and } \tau_{ca}(t) \in [0,\ \tau_{ca}^{max}], \tag{5}$$

and $\tau_{sc/ca}^{max}$ are known.

**Assumption 3:** Sensors and actuators are directly connected to the system and they share a local clock.

**Remark 1:** Assumption 1 and 2 are common assumptions for NCSs. Moreover, having sensors and actuators directly connected to the system does not represent such a strict assumption. In fact, it is quite a common setup when dealing, for instance, with teleoperation and autonomous unmanned vehicles (AUV), e.g. space rovers, where the controller cannot be directly attached to the system.

## III. PROPOSED METHOD

In this paper, MPC is used to cope with the network nondeterminism. As already seen in [7], [8], by using time-stamped information and a local model of the system at the controller side, (nondeterministic) bounded measurement delays can be compensated. Furthermore, as investigated in [8], [11], [12], dispatching long pieces of input trajectories, together with the use of playback buffers, can compensate actuation and computational delays, as well as information losses. In this work the notion of *prediction consistent feedback*[2] introduced in [8] is used to deal with packet dropouts.

### A. Compensation of measurement and actuation delays

Assume for the moment that no information is lost. The presence of $\tau_{sc}(t)$ in the down-link means that the state of the system measured at $t_i$, $x(t_i)$, is available to the controller only at $(t_i + \tau_{sc}(t_i))$, i.e. the information received by the controller does not correspond to the current state of the systems. Additionally, the absence of a common synchronized clock between system and controller does not allow to easily compensate neither $\tau_{sc}(t_i)$ nor $\tau_{ca}(t_i + \tau_{sc}(t_i))$, as shown in [8]. The main challenge is to ensure that, although the delays are nondeterministic, the input trajectory is well defined. From (5) both $\tau_{sc}(\cdot)$ and $\tau_{ca}(\cdot)$ are bounded, i.e. the maximum Round Trip Time (RTT) is known.

The idea behind the algorithm is to use exclusively the clock at the system side, available –from Assumption 3– both to the sensor and the actuator, and let the controller solve the optimal control problem for the maximum RTT *(worst case compensation)*. The obtained input trajectory is then entirely sent to the actuator, but applied only starting from $(t_i + RTT^{max})$. In this way, no synchronization between the controller and system is anymore required. The overall algorithm is reported in Table I.

If MPC with this compensation mechanism is used, then one can prove that the following theorem holds.

*Theorem 3.1 (Worst Case Compensation):* Consider the closed loop system given by (1)-(3). Suppose there is an invariant set $\mathscr{E}$ and terminal penalty $E$ such that

i) $E \in C^1$, $E(0) = 0$, and $\mathscr{E} \subset X$ is closed, connected and containing the origin.
ii) $\exists T_p$ such that

$$T_p > \delta_c^{max} + \tau_{sc}^{max} + \tau_{ca}^{max} \equiv \delta_c^{max} + RTT^{max}, \tag{9}$$

[2] The definition of *prediction consistent feedback* will be recalled later.

TABLE I

**Algorithm 1 (Worst Case Compensation):**

$\forall t_i \in \pi$; $t$ = current time (system's side);

**Sensor:**

1) Measure $x(t_i)$.
2) Send $[x(t_i)|ts_i]$, with $ts_i = t_i$, to the controller.
3) Go to 1.

**Controller:**

`control_input` $= \{[u^*(\cdot)|ts_0]\}$;

1) $[x(t_i)|ts_i]$ arrives.
2) Calculate

$$\bar{x}(t_i + \tau_{sc}^{max} + \tau_{ca}^{max}) = x(t_i) + \int_{t_i}^{t_i+\tau_{sc}^{max}+\tau_{ca}^{max}} f(\bar{x}(\tau), \bar{u}(\tau))d\tau, \quad (6)$$

where

$$\bar{u}(\tau) \equiv u^*(\tau; x(t_i)) \in \texttt{control\_input}, \quad (7)$$
$$\text{for } \tau \in [t_i, t_i + \tau_{sc}^{max} + \tau_{ca}^{max}].$$

3) Solve the optimal control problem for (6) $\longrightarrow$

$$u^*(\tau; \bar{x}(t_i + \tau_{sc}^{max} + \tau_{ca}^{max})), \tau \in [t_i + \tau_{sc}^{max} + \tau_{ca}^{max}, t_i + T_p). \quad (8)$$

4) Send $[u^*(\tau; \bar{x}(t_i + \tau_{sc}^{max} + \tau_{ca}^{max}))|ts_i]$, with $ts_i = (t_i + \tau_{sc}^{max} + \tau_{ca}^{max})$.
5) Insert $[u^*(\tau; \bar{x}(t_i + \tau_{sc}^{max} + \tau_{ca}^{max}))|ts_i]$ in `control_input`.
6) Go to 1.

**Actuator:**

`buffer` $= \{[u^*(\cdot)|ts_0], \ldots, [u^*(\cdot)|ts_n]\}$, for $ts_0 < t < ts_1 \ldots < ts_n$;
`applied_input` $= [u^*(\cdot)|ts_0]$;

1) If $[u^*(\cdot)|ts_i]$ arrives
   a) Insert $[u^*(\cdot)|ts_i]$ in `buffer`.
   b) "Sort" `buffer` by increasing $ts_i$.
   c) `temp` = first element of `buffer`.
2) If $ts_{\texttt{temp}} = t$
   a) `applied_input` = `temp`.
   b) Remove first element from `buffer`.
3) Go to 1.

where $\delta_c^{max}$ = maximum recalculation time, while $RTT^{max} = \tau_{sc}^{max} + \tau_{ca}^{max}$ = maximum RTT.

iii) $\forall x_0 \in \mathscr{E}$, $\exists \bar{u}(\tau) \in U$, $\tau \in [0, T_p]$ such that

$$x(\tau) \in \mathscr{E}, \quad (10a)$$
$$\text{for } \dot{x}(\tau) = f(x(\tau), \bar{u}(\tau)), x(0) = x_0, \quad (10b)$$
$$\text{and } \frac{\partial E}{\partial x} f(x(\tau), \bar{u}(\tau)) + F(x(\tau), \bar{u}(\tau)) \leq 0. \quad (10c)$$

iv) The optimal control problem is feasible for a time $t_0$.

v) The compensation algorithm in Table I is used.

Then $\lim_{t\to\infty} \|x(t)\| = 0$, i.e. asymptotic converge to the origin is achieved.

*Proof:* See Appendix. ■

**Remark 2:** Condition (9) follows from the fact that the input must be defined for every recalculation interval $\delta_c$. It is easy to show that since (8) is dispatched to the actuator, and that the definition interval must be

$$[t_i + \tau_{sc}^{max} + \tau_{ca}^{max},\ t_{i+1} + \tau_{sc}^{max} + \tau_{ca}^{max}),$$

whose right extreme can be upper-bounded by $(t_i + \delta_c^{max} \tau_{sc}^{max} + \tau_{ca}^{max})$, (9) is a necessary condition.

**Remark 3:** Only knowledge on the maximum RTT is required in order to choose a properly long $T_p$ and thus cope effectively with the delays. Algorithms and communication protocols are already available to estimate in real-time the RTT –see [13] for an overview–.

## B. Packet dropouts compensation

The presence of packet dropouts introduces a further degree of uncertainty on the system. In fact, while when a packet is dropped on the down-link the system still works as an open loop, if some information is dropped on the actuation side the controller cannot be sure anymore on which input is applied to the system, and therefore not only stability might be compromised, but it is also impossible to obtain a correct prediction (6), since (7) might differ from the input applied by the actuator.

A solution similar to Algorithm 1 can be used. If a sufficiently long prediction horizon $T_p$ is utilized, and the mismatch between (7) and the applied input is negligible, it shall be possible to counteract packet dropouts, as soon as the number of consecutive losses are shorter than $T_p$. Thus, to ensure asymptotic convergence $u(\tau)$ must meet further requirements. As presented in [8], we require the control trajectories to be prediction consistent.

**Definition (Prediction consistent feedback):** Given the recalculation partition $\pi$, and the terminal set $\mathbb{T} \subseteq X \subseteq \mathbb{R}^n$, the feedback $u(\cdot)$ is called *prediction consistent* if for every recalculation time $t_i \in \pi$, $u(\cdot; x(t_i)) \in U$, and, given two input trajectories $u_k(\cdot; x(t_k))$, $u_h(\cdot; x(t_h))$ obtained at the recalculation times $t_k < t_h$, the predicted states of the system (1), $\bar{x}(t_j; u_k(\cdot; x(t_k)))$, $\bar{x}(t_j; u_h(\cdot; x(t_h)))$ at the recalculation time $t_j > t_h$, obtained by applying the former inputs, belong to the same controllable set $S_i(X, \mathbb{T})$ of the corresponding sampled-data system, $\forall t_j \in \pi$.

This means that the concatenation of successive input trajectories must satisfy some smoothness properties in the state prediction in order to avoid the system destabilization. An exemplification of the former definition can be found in in Fig. 2.

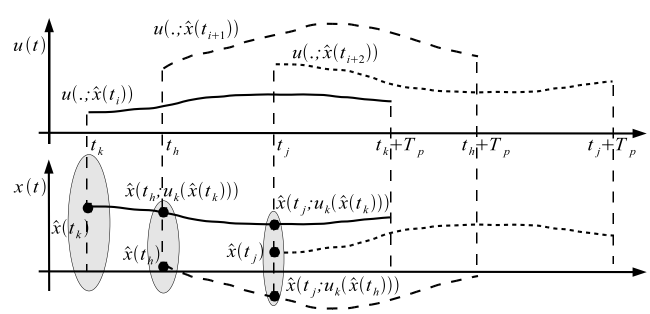


Fig. 2. Example of prediction consistent feedbacks.

**Remark 4:** The use of an acknowledgment mechanism would guarantee (7) to be equal to the input applied by the actuator. This can be seen as a special case of prediction consistency, which can be referred as *input consistency*, i.e. since the trajectories on both side are the same, predictions will be consistent.

It can be proved that by using prediction consistent feedbacks and the compensation approach in Table I, the following theorem holds.

*Theorem 3.2 (Packet Dropouts Compensation):* Consider the closed loop system (1)-(3). Suppose there is an invariant set $\mathscr{E}$, a terminal penalty $E$, and a prediction horizon $T_p$ such that

i) Theorem 3.1 is verified.
ii) $T_p$ is chosen such that

$$T_p > \tau_{sc}^{max} + \tau_{ca}^{max} + n \cdot \delta_c^{max},$$

where $n =$ number of consecutive losses in both the up- and down-link.

iii) $u(\tau) \in U$ is prediction consistent for the terminal region $\mathscr{E}$.

Then $\lim_{t\to\infty} \|x(t)\| = 0$, i.e. asymptotic converge to the origin is achieved.

*Proof:* The proof follows from Theorem 3.1. The use of prediction consistent feedbacks guarantees a negligible mismatch between (7) and the applied input, allowing to prove feasibility and convergence similarly to 3.1. ∎

**Remark 5:** Theoretically, if it were possible to choose $T_p = \infty$, packet dropouts would not have any effect on the system. However, the prediction consistent feedback requirement will decrease the robustness performance.

**Remark 6:** Notice that Theorem 3.1 and 3.2 can guarantee only asymptotic convergence but not asymptotic stability in the sense of Lyapunov. This means that the system can temporary drift away from the origin, but in the long run it will eventually reach the equilibrium point –see, for example, Fig. 4, Section IV–.

## IV. Simulation Results

To test the effectiveness of the method presented in Section III two benchmark systems are considered:

- A continuous stirred-tank reactor (CSTR).
- And, an inverted pendulum on a cart.

### A. Example 1: CSTR

The CSTR taken into account, presented in [14], consists of an irreversible exothermic reaction, $A \to B$, in a constant volume reactor, cooled by a single coolant stream. The system is modeled by the following equations:

$$\begin{aligned} \dot{C}_A(t) &= \frac{q}{V}(C_{A0} - C_A(t)) - k_0 C_A(t) e^{-\frac{E}{RT(t)}} \\ \dot{T}(t) &= \frac{q}{V}(T_0 - T(t)) - \left(\frac{k_0 \Delta H}{\rho C_p}\right) C_A(t) e^{-\frac{E}{RT(t)}} \\ &+ \left(\frac{\rho_c C_{pc}}{\rho C_p V}\right) q_c(t) \left[1 - e^{\frac{-hA}{q_c(t)\rho_c C_{pc}}}\right] (T_0 - T(t)). \end{aligned}$$

The values of the parameters are reported in Table II. The

TABLE II
Nominal CSTR parameter values.

| | | |
|---|---|---|
| Process flow rate | Q | 100 l/min |
| Feed concentration | $C_{A0}$ | 1 mol/l |
| Feed temperature | $T_0$ | 350 K |
| Inlet coolant temperature | $T_{c0}$ | 350 K |
| CSTR volume | V | 100 l |
| Heat transfer term | hA | $7 \cdot 10^5$ cal/min K |
| Reaction rate constant | $k_0$ | $7.2 \cdot 10^{10}$ l/min |
| Activation energy term | E/R | $1 \cdot 10^4$ K |
| Heat of reaction | ΔH | $-2 \cdot 10^5$ cal/mol |
| Liquid densities | $\rho$, $\rho_c$ | $1 \cdot 10^3$ g/l |
| Specific heats | $C_p$, $C_{pc}$ | 1 cal/g K |

objective is to control the concentration $C_A(t)$, by manipulating the coolant flow rate $q_c(t)$. It is assumed that the system is connected to a remote controller through a shared communication network, where the measurement channel is affected by a delay $\tau_{sc}(t)$ modeled as a uniform probability distribution $\mathscr{U}(18,36)$, i.e. $\tau_{sc} \in [18,36]$ seconds. Similarly, the actuation link is supposed to be subject to a delay $\tau_{ca}(t)$ modeled as well as a uniform probability distribution $\mathscr{U}(9,18)$ seconds. No information loss affects the communication. The results for the CSTR are shown in Fig. 3, where the non-compensated and the compensated closed loop system behaviors are depicted.

**Remark 7:** For the given configuration one can also consider that the delays/losses are a result of operators collecting measurements and implementing the control on the plant.

As one can see from simulation example, the proposed MPC approach is able to compensate effectively both actuation and measurement delays. On the contrary, if no compensation is utilized, the controller either is no able to stabilize the system or unpleasant oscillations in $C_A(t)$ occur.

### B. Example 2: Inverted pendulum on a cart

To demonstrate the effectiveness of the method also against packet losses, an inverted pendulum on a cart is taken into consideration. The system is described by the equations

$$\begin{aligned} \dot{x}_1(t) &= x_2(t) \\ \dot{x}_2(t) &= \frac{mL\cos(x_1(t))\sin(x_1(t))x_2^2(t) - g(M+m)\sin(x_1(t)) + \cos(x_1(t))u(t)}{mL\cos^2(x_1(t)) - \frac{4}{3}(m+M)L}, \end{aligned}$$

where attention was paid solely on the pendulum dynamics, described by $x_1$, pendulum position, and $x_2$, angular speed. The parameters of the pendulum are respectively $m = 0.3$ Kg, $M = 1$ Kg, and $L = 1.2$ m.

A first simulation with only packet losses is considered. This can happen for instance when a UDP-like protocol is used for communication purposes. It is assumed that, due to the network dynamics, 30% of the packets in the actuation link are lost, i.e. $p_{ca} = 0.3$. The loss probability is modelled as uniformly distributed variable. Fig. 4 shows the comparison between the non-compensated case and the formerly described approach. As one can see, while the nominal controller without compensation is not able to stabilize the origin, the presented

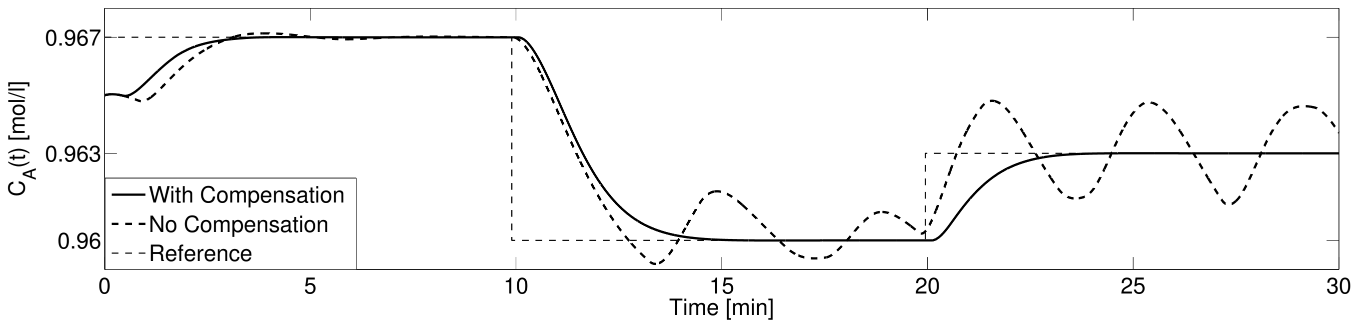


Fig. 3. CSTR: Evolution of $C_A(t)$ with and without delay compensation.

method can bring the system to the unstable equilibrium point even though such a considerable amount of the control information is lost during the communication.

In a second simulation, delays and losses are taken into account. This time it is assumed that an actuation delay of 0.05 seconds affects the up-link, while there is a measurement delay of 0.1 seconds in the down-link. Furthermore, a loss probability $p_{ca} = 0.2$ is considered. Fig. 5 shows the closed loop behavior of the system for the non-compensated case (dashed line) and the results obtained with the proposed method (solid line).

As one can see from the graph, the controller both stabilizes the system and exhibits good performance.

## V. Conclusions

In this paper, the problem of inner-clock synchronization in NCSs has been studied. The problem is commonly present in the available control design methods for NCSs, since it is generally assumed that a common time-frame is available among the network components –sensors, actuators, systems, controllers–. However, this represent a major challenge in the automation industry especially for fast dynamical systems since it is hard to keep inner-clocks synchronized with each other. A possible solution for the problem has been presented. In particular, by imposing the further restriction that sensors and actuators are directly attached to the system, and that they are able to access the system's inner-clock, it is possible, without requiring any synchronization, to counteract both measurement and actuation delays by means of predictive control techniques. This requires some special actuators, called smart-actuators, which have the additional capability of buffering the received information and applying it at the proper time. The proposed approach abstracts from the underlying network protocols, since they can be always represented as additional delays. Moreover, the proposed solution is valid also for nonlinear NCSs, for which only a few results are available so far. Eventually, the method can be easily extended to deal with packet dropouts. In this case, in order to ensure closed loop stability the generated control trajectories must be *prediction consistent*. The proposed method was tested by simulation on a CSTR afflicted by measurement and actuation delays, and an inverted pendulum on a cart, where also packet dropouts were included. As it can be seen, asymptotic convergence and good performance can be guaranteed.

## VI. Acknowledgments

This work was supported by the German research foundation in the frame of the SPP 1309 grant, project "Entwicklung asynchroner prädiktiver Regelungsverfahren für digital vernetzte Systeme".

## Appendix

*Proof of Theorem 3.1*

The problem is to ensure that an input $u(\tau) \in U$ is well defined at every time. The proof is articulated in two parts: Feasibility and convergence.

*Feasibility:* Consider any time $ts_i = (t_i + \tau_{sc}^{max} + \tau_{ca}^{max})$ for which the optimal control problem is feasible, e.g. $t_0 = 0$. By applying the algorithm reported in Table I, the corresponding optimal input trajectory $u^*(\tau;\bar{x}(ts_i)) \in U$, resulting from (6) based on the measurement $x(t_i)$, is implemented for $\tau \in [ts_i, ts_{i+1})$. Thus, by using the former input the system is led to $\bar{x}(ts_{i+1})$, which, assuming there is no model mismatch, must be equal to the real state $x(ts_{i+1})$. Moreover, the remaining piece of optimal trajectory $u^*(\tau;\bar{x}(ts_{i+1}))$, for $\tau \in [ts_{i+1}, ts_i + T_p]$ is admissible and such that the state $\bar{x}(ts_i + T_p) \in E$ is reached. From Theorem 3.1.iii), $\exists \bar{u}(\cdot)$ for which $\mathscr{E}$ is an invariant set. One can consider for example the following candidate input:

$$\tilde{u}(\tau) = \begin{cases} u^*(\tau;\bar{x}(ts_i)) & ,\tau \in [ts_{i+1}, ts_i + T_p] \\ \bar{u}(\tau) & ,\tau \in (ts_i + T_p, ts_{i+1} + T_p] \end{cases}, \quad (11)$$

where $u^*(\tau;x(ts_i))$ is the remaining part of the old optimal input. (11) is admissible and leads to $\bar{x}(ts_{i+1} + T_p) \in \mathscr{E}$. It follows that by induction feasibility at time $ts_i$ implies feasibility at $ts_{i+1}$.

*Convergence:* Denote the optimal cost function at $ts_i$ as the value function $V(\bar{x}(ts_i)) = J^*(u^*(\cdot), \bar{x}(ts_i))$, where $\bar{x}(ts_i)$ is obtained as in (6). This can be written as

$$V(\bar{x}(ts_i)) = \int_{ts_i}^{ts_i+T_p} F(\bar{x}(\tau;\bar{x}(ts_i)), u^*(\tau;\bar{x}(ts_i))d\tau + E(\bar{x}(ts_i + T_p)).$$

The cost resulting from the application of $\tilde{u}(\cdot)$ at time $ts_{i+1}$, starting from the compensated prediction $\overline{x}(ts_{i+1})$ is provided by

$$J(\tilde{u}(\cdot),\overline{x}(ts_{i+1})) = \int_{ts_{i+1}}^{ts_{i+1}+T_p} F(\overline{x}(\tau;x(ts_{i+1})),\tilde{u}(\tau))d\tau + E(\overline{x}(ts_{i+1}+T_p)).$$

This can be reformulated in terms of $V(\overline{x}(ts_i))$ as

$$\begin{aligned} J(\tilde{u}(\cdot),x(ts_{i+1})) = {} & V(\overline{x}(ts_i)) \\ & + \int_{ts_i+T_p}^{ts_{i+1}+T_p} F(\overline{x}(\tau;\overline{x}(ts_{i+1})),\tilde{u}(\tau;\overline{x}(ts_{i+1}))d\tau \\ & - \int_{ts_i}^{ts_{i+1}} F(\overline{x}(\tau;\overline{x}(ts_i)),u^*(\tau;\overline{x}(ts_i)))d\tau \\ & - E(\overline{x}(ts_i+T_p)) + E(\overline{x}(ts_{i+1}+T_p)) \end{aligned}$$

By integrating (10c) over $\tau \in [ts_i+T_p, ts_{i+1}+T_p)$, the last three terms can be upper bounded by zero. Thus,

$$V(\overline{x}(ts_i)) - J(\tilde{u}(\cdot),\overline{x}(ts_{i+1})) \leq - \int_{ts_i}^{ts_{i+1}} F(\overline{x}(\tau;\overline{x}(ts_i)),u^*(\tau;\overline{x}(ts_i)))d\tau.$$

But since $\tilde{u}(\cdot)$ is not necessarily optimal, i.e. $J(\tilde{u}(\cdot),\overline{x}(ts_{i+1})) \leq V(\overline{x}(ts_{i+1}))$, the former expression can be rewritten as

$$V(\overline{x}(ts_i)) - V(\overline{x}(ts_{i+1})) \leq - \int_{ts_i}^{ts_{i+1}} F(\overline{x}(\tau;\overline{x}(ts_i)),u^*(\tau;\overline{x}(ts_i)))d\tau,$$

which is strictly decreasing for $(x,u) \neq (0,0)$. Since no model mismatch is assumed, similarly to [9], by applying a variant of the Barbalat's lemma convergence to the origin for $t \to \infty$ is established. ■

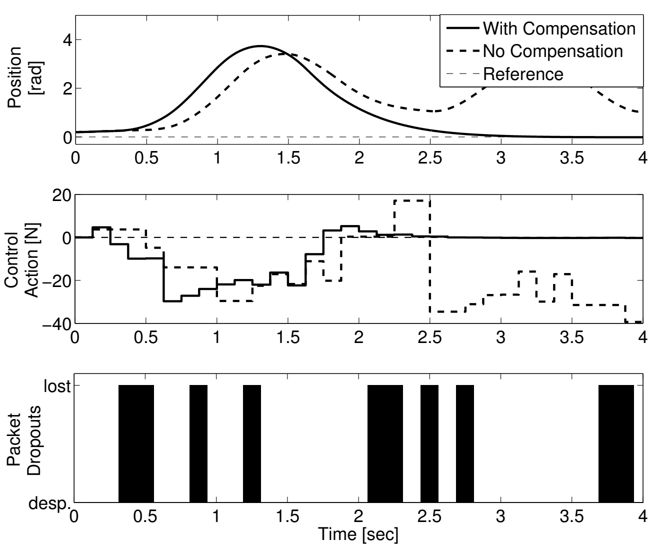


Fig. 4. Simulation 1: Inverted pendulum with packet dropouts.

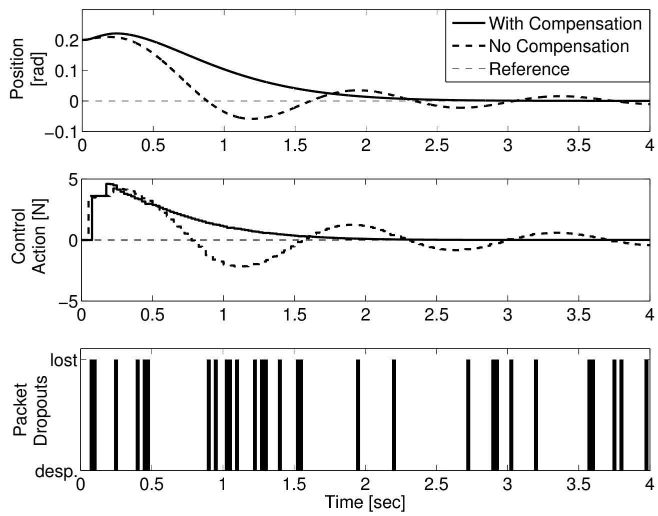


Fig. 5. Simulation 2: Inverted pendulum with delays and packet losses.

## REFERENCES


[1] J. Hespanha, P. Naghshtabrizi, and Y. Xu, "A survey of recent results in networked control systems," *Proc. of the IEEE*, vol. 95, no. 1, pp. 138–162, 2007.

[2] Y. Tipsuwan and M. Y. Chow, "Control methodologies in networked control systems," *Cont. Eng. Pract.*, vol. 11, pp. 1099–1111, 2003.

[3] G. Walsh, O. Beldiman, and L. Bushnell, "Asymptotic behavior of nonlinear networked control systems," *IEEE Trans. Automat. Contr.*, vol. 46, no. 7, pp. 1093–1097, 2001.

[4] D. Carnevale, A. R. Teel, and D. Nešić, "A lyapunov proof of an improved maximum allowable transfer interval for networked control systems," *IEEE Trans. Automat. Contr.*, vol. 52, no. 5, pp. 892–897, 2007.

[5] G. Nair, R. Evans, I. Mareels, and W. Moran, "Topological feedback entropy and nonlinear stabilization," *IEEE Trans. Automat. Contr.*, vol. 49, no. 9, pp. 1585–1597, 2004.

[6] G. Nair, F. Fagnani, S. Zampieri, and R. Evans, "Feedback control under data rate constraints: An overview," *Proc. IEEE*, vol. 95, no. 1, pp. 108–137, 2007.

[7] R. Findeisen and F. Allgöwer, "Computational delay in nonlinear model predictive control," in *Proc. Int. Symp. Adv. Contr. of Chem. Proc.*, Hong Kong, PRC, 2004, pp. 427–432.

[8] R. Findeisen and P. Varutti, "Stabilizing nonlinear predictive control over nondeterministic communication networks," September 2008, int. Workshop on Ass. and Fut. Dir. of NMPC.

[9] F. Fontes, "A general framework to design stabilizing nonlinear model predictive controllers," *Sys. Contr. Lett.*, vol. 42(2), pp. 127–143, 2001.

[10] D. Q. Mayne, J. B. Rawlings, C. V. Rao, and P. O. M. Scokaert, "Constrained model predictive control: Stability and optimality," *Automatica*, vol. 36, no. 6, pp. 789–814, June 2000.

[11] D. Quevedo, E. Silva, and G. Goodwin, "Control over unreliable networks affected by packet erasures and variable transmission delays," *IEEE J. Select. Areas Commun.*, vol. 26, no. 4, pp. 672–685, 2008.

[12] G. Alldredge and V. L. M. S. Branicky, "Play-back buffers in networked control systems. evaluation and design," *Amer. Cont. Conf.*, pp. 3106–3113, 2008.

[13] A. S. Tanenbaum, *Computer Networks*, 4th ed. Prentice Hall PTR, 2002.

[14] J. Morningred, B. Paden, and D. Mellichamp, "An adaptive nonlinear predictive controller," *Chemical Engineering Science*, vol. 47(4), pp. 755–762, 1992.