2011 American Control Conference
on O'Farrell Street, San Francisco, CA, USA
June 29 - July 01, 2011

Event-based NMPC for Networked Control Systems over UDP-like Communication Channels

Paolo Varutti and Rolf Findeisen

***Abstract*— Networked controlled systems have recently received attention from the industry since they allow for flexibility and cost reduction. However, due to the fact that communication media can be subject to random delays, packet dropouts, jitters and other uncertainties, destabilization of the closed loop system can occur. Model predictive control has demonstrated to be a valid solution to cope with these issues. On the other hand, it typically relies on TCP-like (or connection oriented) protocols, i.e. either the received or the lost information is acknowledged.**

In this work, we propose an event-based model predictive control algorithm for nonlinear continuous time systems subject to state and input constraints which is based on UDP-like communication. We show that without the use of any acknowledgment or error message we can derive a compensation algorithm, which used in combination with the controller, under mild conditions, guarantees closed loop stability. The solution is applied to a continuous stirred tank reactor where an exothermic irreversible reaction takes place. The simulations show the effectiveness of the presented algorithm.

I. INTRODUCTION

In recent years, Networked Control Systems (NCSs) have received a lot of attention from the industry, since they allow for fast, flexible, and cheap solutions. For instance, wireless networks can be used for the communication between controller, sensors and actuators, without being invasive. At the same time, they do not require to restructure the already existing network infrastructure, while providing component redundancy. On the other hand, the use of shared and/or wireless communication networks often results in a non-deterministic behavior which can generate random delays, information losses, or other uncertainties. From the control perspective, this is a major problem since the closed loop system can be destabilized.

In [1], [2], a general review on control over communication networks can be found. As presented in [1], [2], the most of the available results about NCSs are mostly limited to linear systems, e.g. [3]–[6]. On the other hand, work on nonlinear ones is typically available only for the discrete time case, e.g. [7]–[10]. Besides, current research focuses mostly on solving particular problems without providing a general framework to cope at the same time with the network nondeterminism and to reduce the exchanged information, cf. [1], [2]. Model based solutions, in particular Model Predictive Control (MPC), have demonstrated to be effective in dealing with both delays and packet dropouts, cf. [8], [10]–[13]. However, these approaches typically rely on a connection-oriented (or TCP-like) communication channel on the input side (see for example [7], [8], [12]). In other words, either error or acknowledgment messages need to be used to make sure that the input is correctly delivered to the actuator. In [11], the use of acknowledgments is avoid by introducing additional set-based constraints on the optimization problem, thus resulting in a non-trivial-to-implement solution.

In this paper, we introduce an Event-based Model Predictive Control (EB-NMPC) algorithm for nonlinear continuous time NCSs, subject to state and input constraints. Differently from our former work [12], [14], the communication channels are supposed to be connectionless, i.e. up- and downlink work in a UPD-like way without using error/acknowledgment messages. Additionally, alternatively to [11], the solution is easily applicable and computationally undemanding. Our novel model-based network compensation algorithm, in combination with EB-NMPC, guarantees closed loop stability for NCSs controlled via UPD-like communications. Additionally, the compensator has the advantage of making the communication network transparent to the controller, thus allowing under mild conditions to re-use any controller already developed for the nominal system. The choice of an event-based controller has also the advantage of reducing the network traffic, thus potentially improving the overall system performance.

The presented solution is tested by simulating a Continuous Stirred Tank Reactor (CSTR), where an irreversible exothermic reaction takes place. The simulations show that the method is effective under the presence of random delays and random information losses, by achieving stability without violating the state and input constrains.

II. PROBLEM STATEMENT

We consider a nonlinear continuous time system

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

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

where $x(t)$ and $u(t)$ represent respectively the constrained state and the constrained input. The following assumptions on the system are made:

- **A1)** $f(x(t), u(t))$ is locally Lypschitz.
- **A2)** $f(0, 0) = 0$, i.e. the origin is an equilibrium point of the system.
- **A3)** The complete state $x(\cdot)$ is available only at discrete times $t_i \in \pi$, where π is a increasing sequence of sampling times (*time partition* or simply *partition*).

P. Varutti and R. Findeisen are with the Laboratory for Systems Theory and Automatic Control, Institute for Automation Engineering, Otto-von-Guericke University, Magdeburg, Germany. {paolo.varutti, rolf.findeisen}@ovgu.de

P. Varutti is also member of the International Max Planck Research School for Analysis, Design and Optimization in Chemical and Biochemical Process Engineering, Magdeburg, Germany

978-1-4577-0079-8/11/$26.00 ©2011 AACC

The system is controlled by a remote regulator connected through a *(shared) nondeterministic communication network* (see Figure 1). We look for an *application-level solution*,

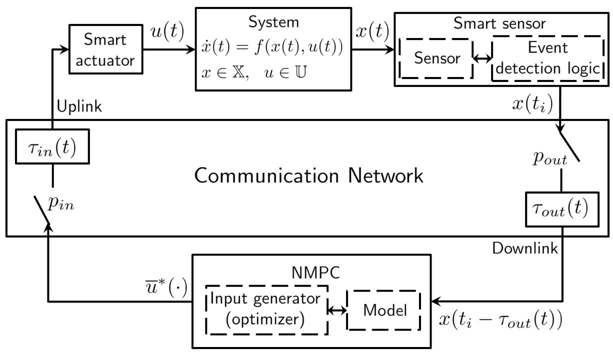


Fig. 1. Graphical representation of the problem under investigation.

where the network is abstracted as merely delays and packet dropouts. Additionally, we assumed that the network has the following properties:

A4) A set of *synchronized clocks (or a global clock)* is available to each component — controller, sensor, actuator, system —.

A5) The network is subject to *bounded random delays*

$$\tau_{in}(t) \in [0, \overline{\tau}_{in}], \quad \tau_{out}(t) \in [0, \overline{\tau}_{out}], \tag{3}$$

respectively in the up- and downlink.

A6) The network is subject to *random information losses* with probability-loss

$$p_{in}(t) \in [0, 1), \quad p_{out}(t) \in [0, 1), \tag{4}$$

i.e. *at least some exchanged packets are delivered correctly*.

A6) All exchanged *packets are time-stamped*. The time-stamping depends on the algorithm which is utilized (more details are to be found in Section III).

A7) Both *down- and uplink are connectionless*, i.e. they both work in a UPD-like way.

Remark 2.1: Assumptions **A4)-A6)** are standard assumptions for NCSs. Small uncertainties on clocks and time-stamps are, in general, non-trivial to handle. An alternative solution was presented by the authors in [15].
Disordered packet arrivals are solved by keeping the packet with the most recent time-stamp. All other information is discarded.

All the components are supposed to be event-based. In this framework, we can identify four different ways how the network works:

a) *No dropouts:* no information is lost; the network is only subject to delays.
b) *Consecutive Measurement Dropouts:* only measurement are lost; new inputs are not generated since measurements are not delivered correctly.
c) *Consecutive Actuation Dropouts:* only inputs are dropped; measurements keep arriving regularly.
d) *Consecutive Actuation and Measurement Dropouts:* input and measurement packets are consecutively lost, leaving the system work in open loop.

Notice that, since a new input is generated only when a measurement arrives, consecutive losses of measurement and actuation messages cannot occur.

To control the system, we use an EB-NMPC algorithm. For sake of clarity, before introducing the main results of this work in the following section a short review on EB-NMPC is given, cf. [12], [16].

A. Event-based Nonlinear Model Predictive Control

The basic idea behind predictive control is the repeated solution of an Optimal Control Problem (OCP) over a finite prediction horizon T_p:

$$J^*(x(t_i)) = \min_{\overline{u} \in \mathbb{U}} \int_{t_i}^{t_i+T_p} F(\overline{x}(\tau), \overline{u}(\tau))d\tau + E(\overline{x}(t_i + T_p)) \tag{5a}$$

$$\text{s.t.} \quad \dot{\overline{x}}(t) = f(\overline{x}(t), \overline{u}(t)), \quad \overline{x}(t_i) = x(t_i) \tag{5b}$$

$$\overline{x}(t) \in \mathbb{X}, \quad \overline{u}(t) \in \mathbb{U} \tag{5c}$$

$$\overline{x}(t_i + T_p) \in \epsilon, \tag{5d}$$

where $\bar{\cdot}$ refers to the controller's internal variables (model at the controller side), whereas $x(t_i)$ represents the measurement taken from the real system (1) at time t_i. Only the first piece of optimal input trajectory is applied to the system, i.e.

$$\overline{u}^*(\tau; x(t_i)), \quad \tau \in [t_i, t_{i+1}), \quad t_i, t_{i+1} \in \pi. \tag{6}$$

By choosing a suitable cost functional $F(\cdot)$, terminal cost $E(\cdot)$, terminal region ϵ, prediction horizon T_p, and a proper partition π, closed loop stability of the EB-NMPC scheme can be shown, in the sense of asymptotic convergence (see [12], [16]–[18] for additional details). For sake of clearness, we recall here the definition of *proper partition*, firstly introduced in [12], [14]:

Definition 2.1 (Proper Partition):
An increasing sequence of recalculation times $t_i \in \mathbb{R}^+$ is called *proper partition* π if $\exists \beta \in \mathbb{R}^+$ such that

$$0 < \beta \leq (t_i - t_{i-1}) < T_p, \ \forall t_{i-1}, t_i \in \pi, \tag{7}$$

i.e. the maximum recalculation interval is strictly smaller than the prediction horizon T_p.

Notice that, in an event-based framework, condition (7) is necessary to ensure that the input $\overline{u}^*(\tau; x(\cdot))$ is always correctly defined.

The presence of delays and/or packet dropouts, however, can destabilize the closed loop system. Therefore, a network compensator is needed to cope with these issues. In this work, we utilize a *model-based network compensator* (see Figure 2). The choice of a model-based network compensator is motivated by the fact that a model of the system is already available for the EB-NMPC controller. In this situation, it seems logical to re-use such information to cope with the induced network nondeterminism. In practice, EB-NMPC controller and network compensator could be merged

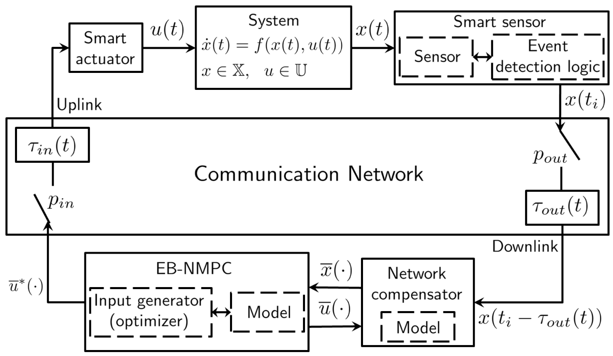


Fig. 2. Model-based network compensator scheme.

together and become a single component. How the compensation algorithm is realized depends on the underlying network infrastructure and protocol stack. Additional details on the compensation algorithm that we use in this work can be found in Section III-A.

III. MAIN RESULT

A. Model-based Network Compensator

The model-based network compensator is an *application level algorithm* designed to counteract the delays and the packet dropouts. In this way, the compensator can be used without loss of generality with any kind of network infrastructure, see [19].

As long as the compensator is *input-consistent*, stability of the closed loop system can be shown as in [12], [19]. For sake of clarity, we recall here the definition of *input-consistent algorithm*:

Definition 3.1 (Input-consistent Algorithm): Let $u(t) \in \mathbb{U}$ be the input applied to the system; $\overline{u}^*(t; x(\cdot)) \in \mathbb{U}$ the input generated by the controller and used by the compensator. We call a model-based network compensator *input-consistent*, if

$$\forall t \in \mathbb{R}^+, \quad u(t) \equiv \overline{u}^*(t; x(\cdot)), \tag{8}$$

i.e. the applied input is always equal to the input generated by the controller.

Input-consistency is important when a model-based network compensator is utilized, since the input $\overline{u}(t; \overline{x}(\cdot)) \in \mathbb{U}$ is adopted to forward predict the system evolution by using the available model. If there were a discrepancy between the two inputs, the real state $x(t)$ would differ from the predicted behavior $\overline{x}(t)$.

Similarly to [12], [14], the compensation of random delays is done by worst-case forward prediction, i.e. the worst possible delays in both the input and output channel are considered. This is necessary because, although we can obtain the exact value of $\tau_{out}(t)$, the presence of a random delay $\tau_{in}(t)$ does not allow to be sure when the input arrives. However, we can make the process deterministic again by taking $\overline{\tau}_{in}$, $\overline{\tau}_{out}$, and using these values to predict the behavior of the system in the future. Utilizing this worst-case approach is also necessary to counteract information losses, as clarified later. If we *time-stamp the measurements* $x(t_i)$ *with* t_i, instant in which they are collected from the plant, then we can calculate

$$\overline{x}(t_i + \overline{\tau}_{out} + \overline{\tau}_{in}) = \int_{t_i}^{t_i + \overline{\tau}_{out} + \overline{\tau}_{in}} f(x(\tau), u(\tau)) d\tau + x(t_i), \tag{9}$$

and use (9) to solve the OCP (5). For simplicity of notation, we define

$$ts_i := t_i + \overline{\tau}_{out} + \overline{\tau}_{in}. \tag{10}$$

Notice, however, that the resulting optimal input

$$\overline{u}^*(\tau; \overline{x}(ts_i)), \ \tau \in [ts_i, ts_i + T_p), \tag{11}$$

refers to a future time-instant ts_i. Thus, it cannot be used by the actuator immediately at its arrival. Instead, it must be buffered until ts_i. This solution works well only when no dropouts occur. In [7], [9], [14], this problem was solved, even if not explicitly mentioned, by utilizing connection-oriented (or TCP-like) protocols on the input side, i.e. either error messages, or acknowledgments, are sent. In this case, however, we work with UDP-like connections, which means that once the message is sent, there is no way to be sure it is arrived at destination.

We propose to utilize additional information to deal with packet dropouts: by *adding to the measurement packets the time-stamp* t_i *and also the time-stamp of the latest successfully received actuation message* ts_j, we can confirm the arrival of an input. Instead, measurement losses are ignored. In practice, the acknowledgment mechanism is included in the downlink, without requiring a full-duplex connection-oriented communication on the input side. Such a solution is reasonable since we want to control the system in closed loop. Therefore, as assumed in **A6)** some measurement must arrive to the controller in a finite time.

Notice also that before sending a new measurement message, the sensor needs to make sure that either a new input is arrived, or it went lost. Therefore, a new measurement $x(t_i)$ cannot be sent before the arrival time $t_i + \tau_{out}(t) + \tau_{in}(t)$ and, in case of a dropout, it has to wait till the next expected time-stamp ts_i.

In addition, instead of sending only the first piece of input trajectory, the complete optimal input (11) is dispatched. It is up the smart-actuator to store the input and use it when necessary. A meta-code description of the algorithm is given in Table I.

Remark 3.1: The maximum length of the input trajectory depends on the actual underlying protocol, as well as on the problem under investigation, i.e. prediction horizon, problem discretization. A trade-off between packet size and problem requirements should be considered in order to obtain good closed-loop performance.

B. Stability Results

By using the algorithm described in Table I, we can prove the closed loop stability of the NCS under the presence of both random delays and information losses. Before going any further, we introduce the following quantities, which are important for the main result:

- N_{in} = maximum number of consecutive losses in the uplink.
- N_{out} = maximum number of consecutive losses in the downlink.
- $RTT = \tau_{out}(t) + \tau_{in}(t)$, round-trip-time.
- $RTT^{max} = \overline{\tau}_{out} + \overline{\tau}_{in}$, maximum round-trip-time.

We can now present the following theoretical result:

Theorem 3.1 (Stability over UDP-like channels): Consider the nonlinear continuous time NCS (1) subject to random delays $\tau_{in}(t)$, $\tau_{out}(t)$ and random information losses with probabilities $p_{in}(t)$, $p_{out}(t)$. If

i) Assumptions **A1)-A7)** are verified.
ii) The model-based network compensator of Table I is used.
iii) The EB-NMPC controller (11) is chosen such that the prediction horizon

$$T_p > (N_{in} + N_{out}) \cdot RTT^{max}, \tag{12}$$

and minimum sampling time

$$\beta = \begin{cases} RTT, & \text{if a new input arrives} \\ RTT^{max}, & \text{if the last input is lost} \end{cases}. \tag{13}$$

Then, the closed loop system is asymptotically stable, in the sense of asymptotic convergence, i.e.

$$\lim_{t\to\infty} \|x(t)\| = 0. \tag{14}$$

Proof: To prove closed loop stability we first need to show that the compensator is input-consistent. Then, stability follows from [12], [19].

Without loss of generality, we suppose that initially an actuation and a measurement packet arrived consecutively[1], i.e. controller knows that $\overline{u}^*(\tau; x(ts_{i-1}))$, $\tau \in [ts_{i-1}, ts_{i-1} + T_p)$, has arrived from the latest measurement $x(t_i)$. We can then distinguish four different scenarios (see Figure 3):

a) *No Dropouts:*
Input-consistency easily follows from the fact that every measurement is generate only after a new input arrived (Condition (13)) and thus the controller always knows the applied input $\overline{u}^*(t; \overline{x}(\cdot))$.

b) *Consecutive Measurement Dropouts:*
The last input $\overline{u}^*(\tau; \overline{x}(ts_{i-1}))$, $\tau \in [ts_{i-1}, ts_{i-1}+T_p)$, is re-used until a new measurement $x(t_j)$, $t_j > t_i$, arrives, and the input $\overline{u}^*(\tau; \overline{x}(ts_j))$, $\tau \in [ts_j, ts_j+T_p)$, is generated.
Recursively, the latest input is known-to-be-arrived when a new measurement $x(t_k)$, $t_k > t_j$, is successfully delivered. From **A6)** and (12), we know that a measurement $x(t_i)$ must arrive before the end of the prediction horizon, i.e. before the latest input is completely over. Therefore, input-consistency is verified.

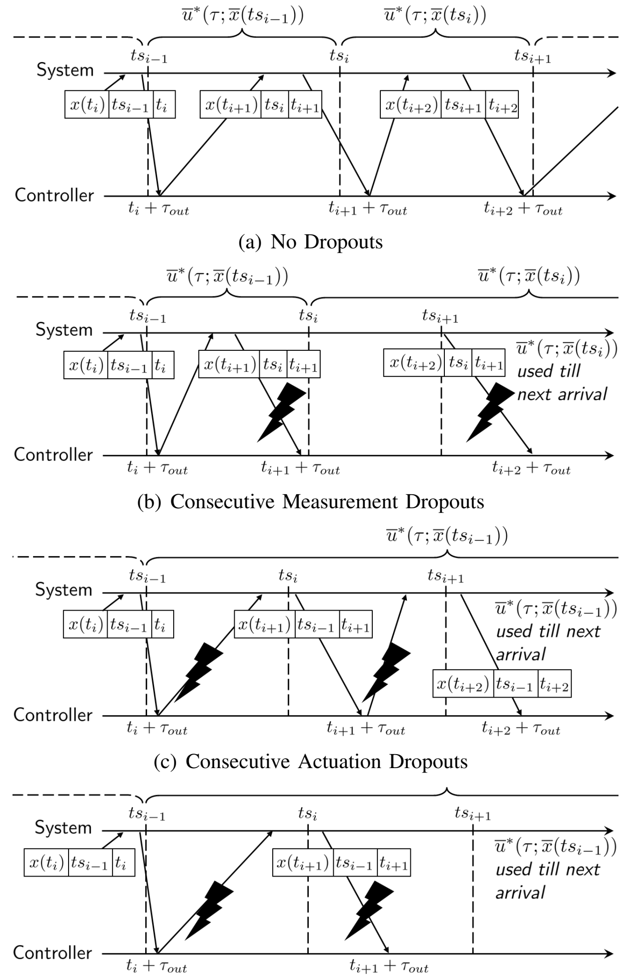
c) *Consecutive Actuation Dropouts:*
Similarly to the former case, $\overline{u}^*(\tau; \overline{x}(ts_{i-1}))$, $\tau \in [ts_{i-1}, ts_{i-1} + T_p)$ is re-used until a new input $\overline{u}^*(\tau; \overline{x}(ts_j))$, $\tau \in [ts_j, ts_j + T_p)$, corresponding to the measurement $x(t_j)$, $t_j > t_i$, is correctly dispatched.
Since the measurements $x(t_i)$ are assumed to arrive correctly, the controller knows always the most recently received input. Again, from **A6)** and (12) input-consistency holds.

d) *Consecutive Actuation and Measurement Dropouts:*
The controller knows the latest successfully received input $\overline{u}^*(\tau; \overline{x}(ts_{i-1}))$, $\tau \in [ts_{i-1}, ts_{i-1} + T_p)$. Similarly to b)-c), from **A6)** and (12), a new measurement $x(t_i)$ and a new input $\overline{u}^*(t; \overline{x}(\cdot))$ must arrive before the end of the prediction horizon. Input-consistency easily follows.

Due to the fact that a new input is generated only when a measurement is lost, there cannot be a measurement loss followed by an input dropout. Consequently, the algorithm is always input-consistent.

From [12], [19], since the model-based network compensator is input-consistent, and the chosen prediction horizon T_p is sufficiently long, the closed loop NCS is asymptotically stable, in the sense of asymptotic convergence, i.e. $\lim_{t\to\infty} \|x(t)\| = 0$. ∎

[1]This essentially means that at least for a very short period of time the NCS works without any problem.



(a) No Dropouts
(b) Consecutive Measurement Dropouts
(c) Consecutive Actuation Dropouts
(d) Consecutive Actuation and Measurement Dropouts

Fig. 3. Different dropout scenarios. As soon as some information is lost, the old input is reused to preserve stability.

Remark 3.2: Condition (13) on the minimum sampling time means that we cannot send a new measurement $x(t_i)$, before either the latest input has arrived at $t_i + \tau_{out}(t) + \tau_{in}(t)$, or it is sure that it went lost, i.e. more than the RTT^{max} has passed after the last arrival.

Remark 3.3: Notice that the presented compensation algorithm based on the use of redundant information can be actually implemented only in an event-based framework. In fact, with a synchronous controller — a regulator which uses constant sampling intervals — since an input needs to be calculated at every sampling time, the forward prediction should take into account every possible sequence of losses. This would result in a computationally intractable problem.

IV. SIMULATION RESULTS

The model-based network compensator presented in this work is applied to the CSTR represented in Figure 4, where an irreversible exothermic reaction, $A \to B$, takes place in a constant volume, cooled by a single coolant stream at temperature T_c. The system is described by the following

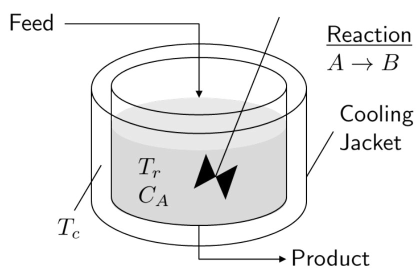


Fig. 4. CSTR where an irreversible exothermic $A \to B$ takes place.

equations:

$$\dot{C}_A(t) = \frac{F}{V}(C_{Af} - C_A(t)) - k_0 C_A(t) e^{-\frac{E}{RT_r(t)}},$$
$$\dot{T}_r(t) = \frac{F}{V}(T_{rf} - T_r(t)) - \frac{k_0 \Delta H}{\rho c_p} C_A(t) e^{-\frac{E}{RT_r(t)}} + \frac{UA}{\rho c_p V}(T_c(t) - T_r(t)),$$

where $C_A(t)$ represents the concentration of the reactant A, and $T_r(t)$ is the reactor temperature. A detailed description and the exact values of all the parameters for the nominal condition $T_c^{nom} = 103.4\ K$ can be found in [20]. Both $T_r(t)$ and $T_c(t)$ are subject to hard safety constraints. In particular, $T_r(t)$ must not overcome $500\ K$, while $T_c(t)$ lies between $275\ K$ and $350\ K$. Under nominal conditions, the system has three equilibrium points: $(0.19, 432.08)$ (stable), $(0.52, 398.97)$ (unstable), and $(0.90, 361.14)$ (stable).

The objective is to stabilize the unstable equilibrium point $(0.52, 398.97)$ by manipulating the temperature of the cooling jacket $T_c(t)$. To do that, we use a remote EB-NMPC controller connected to the system through a nondeterministic network having the following properties: $\tau_{in} \in [0, 10]\ sec$, $\tau_{out} \in [0, 20]\ sec$, $p_{in} = 0.15$, $p_{out} = 0.05$. For simplicity, all stochastic variables are modeled as uniform distributions. From several simulations, it was seen that with the former parameters the max number of consecutive losses in the up- and downlink are respectively 2 and 1. For security, $N_{in} = 4$ and $N_{out} = 2$ were chosen. This is due to the fact that generally the information provided by simulation is less than the one obtained from the analysis of a real network. By choosing $N_{in} = 4$, $N_{out} = 2$ we have that the prediction horizon T_p needs to be longer than $150\ sec$. A prediction horizon of $160\ sec$ is chosen and the following cost function utilized:

$$J(x(t_i)) = \int_{t_i}^{t_i+T_p} \big((x - x_{sp})^T Q (x - x_{sp}) + u^T R u\big) d\tau + (x(t_i + T_p) - x_{sp})^T S (x(t_i + T_p) - x_{sp}),$$

where $x(t) = [C_A(t)\ \ T_r(t)]^T$, x_{sp} represents the desired set-point, $u = T_c(t)$, $Q = I$, $R = 1$, and $S = I$, with I equal to the identity matrix. For simplicity of implementation, in the simulations the input is held constant between consecutive recalculation times. In Figure 5, we can see that without

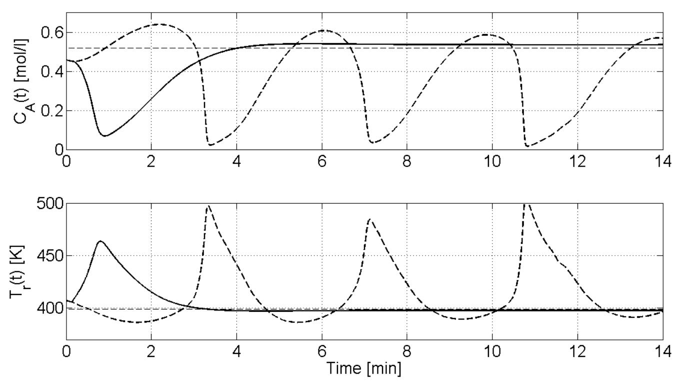


Fig. 5. Comparison between the compensated (black solid line) and non-compensated closed loop system (black dashed line).

model-based network compensator, the EB-NMPC controller is not able to steer the system to the desired set-point (black dashed line), generating an oscillatory behavior and violating the safety constraints on $T_r(t)$. On the other hand, when the compensation mechanism presented in Section III is introduced (black solid line in Figure 5), the CSTR can be safely brought to the desired equilibrium point. The initial transition phase is due to the fact that both measurement and input are delayed and lost, and the controller and the actuator are initialized as if the system were already on the desired set-point.

V. CONCLUSIONS AND FUTURE WORK

In this paper, an event-based predictive control solution for nonlinear continuous time NCSs was presented. Differently from former work, which relies on the use of connection-oriented (or TCP-like) communication channels, here a connectionless (or UDP-like) solution is presented. In particular, we showed that by adding extra information to the header of the measurement packet, it is possible to obtained an input-consistent model-based network compensator. The time-stamp of last correctly arrived input packet is included in the header of the output. The former solution seems to be reasonable since we want to remotely closed loop control the system, thus expecting that at least some measurement packets sooner or later arrive. We can prove that under mild conditions, EB-NMPC in combination with this network compensator achieves closed loop stability, in the sense of asymptotic convergence.

Future work should concentrate on the development of robust methodologies for systems subject to disturbances or uncertainties. Additionally, the algorithm could be improved to obtain less restrictive conditions on the prediction horizon length, e.g. by updating the values N_{in}, N_{out} in function of the network traffic. It is also interesting to investigate other possible input-consistent compensation algorithms, which can be used with connectionless communication protocols.

VI. ACKNOWLEDGEMENTS

This work is partly supported by the German Research Foundation (DFG) within the scope of the priority research program SPP 1305.

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. Chow, "Control methodologies in networked control systems," *Cont. Eng. Pract.*, vol. 11, pp. 1099–1111, 2003.

[3] E. Garone, B. Sinopoli, and A. Casavola, "LQG control for distributed systems over TCP-like erasure channels," in *Proc. 46th IEEE Conf. on Dec. and Cont.*, 12–14 Dec. 2007, pp. 44–49.

[4] A. Casavola, E. Mosca, and M. Papini, "Predictive teleoperation of constrained dynamic systems via internet-like channels," vol. 14, no. 4, pp. 681–694, July 2006.

[5] P. L. Tang and C. W. De Silva, "Stability validation of a constrained model predictive networked control system with future input buffering," *International Journal of Control*, vol. 80, pp. 1954–1970, 2007.

[6] M. Cloosterman, N. van de Wouw, W. Heemels, and H. Nijmeijer, "Stabilization of networked control systems with large delays and packet dropouts," in *Proc. Amer. Contr. Conf., ACC'08*, 2008, pp. 4991–4996.

[7] G. Pin and T. Parisini, "Stabilization of networked control systems by nonlinear model predictive control: a set invariance approach," in *Nonlinear Model Predictive Control: Towards New Challenging Applications*, ser. Lecture Notes in Control and Information Sciences LNCIS 384, L. Magni, D. Raimondo, and F. Allgöwer, Eds. Springer, 2009, pp. 195–204.

[8] L. Grüne, J. Pannek, and K. Worthmann, "A prediction based control scheme for networked systems with delays and packet dropouts," in *Proceedings of the 48th IEEE CDC*, Shanghai, China, 2009, to appear.

[9] L. Grüne, J. Pannek, and K. Worthmann, "A networked unconstrained nonlinear MPC scheme," in *Proc. of the Europ. Contr. Conf. 2009, ECC'09*, 2009, pp. 371–376.

[10] I. G. Polushin, P. X. Liu, and C.-H. Lung, "On the model-based approach to nonlinear networked control systems," *Automatica*, vol. 44, pp. 2409–2414, 2008.

[11] R. Findeisen and P. Varutti, "Stabilizing nonlinear predictive control over nondeterministic communication networks," in *Nonlinear Model Predictive Control: Towards New Challenging Applications, Lecture Notes in Control and Information Sciences LNCIS 384*, L. Magni, D. Raimondo, and F. Allgöwer, Eds. Springer, 2009, pp. 167–179.

[12] P. Varutti, B. Kern, T. Faulwasser, and R. Findeisen, "Event-based model predictive control for networked control systems," in *Proc. of the 48th IEEE Conf. on Dec. and Cont., CDC'09*, 2009, pp. 567–572.

[13] J. Lunze and D. Lehmann, "A state-feedback approach to event-based control," *Automatica*, vol. 46, no. 1, pp. 211–215, January 2010.

[14] P. Varutti and R. Findeisen, "Predictive control of nonlinear chemical processes under asynchronous measurements and controls," in *Proc. Int. Symp. Adv. Contr. of Chem. Proc., ADCHEM'09*, 2009, pp. 156–161.

$t :=$ global time;

SYSTEM SIDE
$t_i :=$ recalculation time;
$ts_i :=$ time-stamp of the most recently arrived input;
$RTT^{max} :=$ maximum RTT (constant value);
$new :=$ boolean telling if a new input arrived;

Sensor:

$event :=$ generated by the event-detection logic;
while($true$){
  if ($event == true$){
    if($new == true$ or $t \geq t_i + RTT^{max}$){
      send $\boxed{x(t_i)|ts_j|t_i}$;
      $t_i = t$;
      if ($new == true$) $new = false$;}}}

Actuator:

$u(t) :=$ applied input;
$last\_input :=$ last received input;
while($true$){
  if $\left(\boxed{\overline{u}^*(\tau;\overline{x}(ts_j))|ts_j} \text{ arrives} \right)${
    $last\_input = \overline{u}^*(\tau;\overline{x}(ts_j))$;
    $ts_i = ts_j$;
    $new = true$;}
  if ($t == ts_i$){
    $u(t) = last\_input$;}}

CONTROLLER SIDE
$pred :=$ forward state prediction $\overline{x}(ts_i)$;
$buffer :=$ inputs used for compensation;

Compensator:

$\boxed{x(t_k)|ts_m|t_k}$:= last measurement;
while($true$){
  if $\left(\boxed{x(t_i)|ts_j|t_i} \text{ arrives and } t_i > t_k \right)${
    remove all $\overline{u}^*(\tau;\overline{x}(ts_k)) : ts_k > ts_j$
    from $buffer$;
    $pred = \int_{t_i}^{ts_i} f(x,u)d\tau + x(t_i)$;}}

Controller:

while($true$){
  if(new $pred$){
    solve OCP (5) with pre;
    insert $\boxed{\overline{u}^*(\tau;\overline{x}(ts_i))|ts_i}$ in $buffer$;
    send $\boxed{\overline{u}^*(\tau;\overline{x}(ts_i))|ts_i}$;}}

TABLE I

MODEL-BASED NETWORK COMPENSATOR ALGORITHM.

[15] ——, "On the synchronization problem for the stabilization of networked control systems over nondeterministic networks," in *Proc. of the Amer. Contr. Conf., ACC'09*, 2009, pp. 2216–2221.

[16] T. Faulwasser, B. Kern, P. Varutti, and R. Findeisen, "Nonlinear predictive control based on asynchronous measurement and control signals," *at - Automatisierungstechnik*, vol. 6, pp. 279–287, 2009.

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

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

[19] P. Varutti, B. Kern, and R. Findeisen, "Event-based control for networked control systems: an active compensation approach," in *Entwurf komplexer Automatisierungssysteme - EKA2010*, U. Jumar, E. Schnieder, and C. Diedrich, Eds. Ifak, 2010, pp. 29–38.

[20] M. Henson and D. Seborg, *Feedback Linearizing Control*, M. Hensen and D. Seborg, Eds. Prentice Hall, 1997.