Proceedings of the
44th IEEE Conference on Decision and Control, and
the European Control Conference 2005
Seville, Spain, December 12-15, 2005

NCS-Controllers for Ambient Intelligence Networks – Control Performance versus Control Effort

L. Litz, O. Gabel, I. Solihin

Abstract – **Ambient Intelligence (AmI) Systems are typically based on low-energy and low-performance nodes which are connected by wireless ad-hoc networks. They imply some additional constraints to the general Networked Control System problem which are discussed in some detail. The main AmI-effects limiting the control performance and threatening the control stability are shown to be the variable network delays, the stochastic packet losses, the quasi packet losses and the limited resolution. With respect to the constraints three classes of solutions are introduced and compared. It is shown that the solutions result in trade-offs between the control performance and the limited control effort. Simulations are performed to make these trade-offs transparent for a special test system.**

I. INTRODUCTION

NETWORKED control systems (NCS) have obtained more and more attention during the last years, see the overview papers [1], [2]. The components of an NCS are sensors, controllers and actuators which are connected by permanent communication. This results in feedback control via digital wired and wireless networks influencing the control loop behavior in different ways. Depending on the network load the transmission times may vary yielding variable delays (or jitter) in the loop. Some other effects - e.g. a buffer overflow in routing nodes or a corruption of packets on the transmission line - generate packet losses. The rate of packet losses and the probability distribution of the time delays are the most crucial parameters of the Quality-of-Service (QoS), describing the network behavior in the context of feedback control. Of course, these QoS-parameters depend on the special type of the network. The best case would be deterministic fieldbus networks which are originally constructed for control applications. While more and more feedback control is carried out via stochastic networks methods are needed to deal with the emerging problems. There is a wide range of different stochastic networks and, unfortunately, no solution is appropriate to solve all problems without regard to the special network type.

As will be shown in section II the emerging paradigm of Ambient Intelligence (AmI) creates very special networks which impose additional NCS-problems. Most of them stem from the nodes in those networks. A typical AmI-node is a very small low-power, low-weight and low-cost sensor node, fixed to or near the human body. Parts of the nodes or even all nodes are moving and connected by a wireless ad-hoc network. Nodes may enter and leave the network.

Many AmI-scenarios need feedback control from distributed sensors to the actuators in the AmI-network. Feedback control is one possibility to generate an intelligent behavior for AmI-systems. Because of the various limitations of the AmI-nodes and the AmI-network, as well, designing the controller with a sufficient performance is a challenging task. In addition to the classical trade-offs, which are needed in the NCS context [3], the limited resources drive into additional trade-offs. This is shown in Section III and clarified by an example described in Sections IV and V.

II. AMI-SYSTEMS AND THEIR IMPLICATION ON NCS

There are some new papers and books describing the field of Ambient Intelligence (AmI), e.g. [4], [5], [6], [7]. AmI-systems can be shortly defined by four characteristics. They

- include distributed sensing, actuating and computing components,
- which are connected by seamless communication trough networks,
- use pervasive computing to generate intelligent behavior and they
- use an intelligent and unobtrusive human system interface.

The ambition of AmI-systems is to enhance the quality of human live when living, working, in leisure time and so on. Several scenarios have been published like assisted living [8], [9] or assisted training [7], [15].

The first three characteristics cited above yield direct consequences for the resulting NCS-problems. As mentioned before the AmI-nodes at the sensor level are low-power, low-weight and low-cost nodes. A typical technical basis is the MICA-node from Berkeley [10], [11]. Most of the time intervals the node should be sleeping (with electrical power of 6 μW) because computing (24 mW) needs much more power. The most power consuming task is sending and receiving of data (75 mW). For these reasons and because of the low cost, low accuracy character of sensors (typically 2% to 5% accuracy) it does not make sense to realize more than 6 bit resolution in the A/D-Converter. As a consequence, short data packets result, at least if the packet overhead is kept small.

Seamless communication of the AmI-network relates to the data transport through different networks which, as a rule, are hierarchically structured. In [6] five shells are de-

---

Lothar Litz is head of the Institute of Automatic Control at the University of Kaiserslautern, Erwin-Schrödinger-Str. 12, 67663 Kaiserslautern, Germany (e-mail: litz@eit.uni-kl.de).

Oliver Gabel and Ivan Solihin are with the Institute of Automatic Control; e-mail: oliver.gabel@eit.uni-kl.de, solihin@eit.uni-kl.de).

0-7803-9568-9/05/$20.00 ©2005 IEEE

fined with the first four ones as

- BAN (Body Area Network, 1m range)
- PAN (Personal Area Network, 10m range)
- LAN (Local Area Network, 100m range)
- WAN (Wide Area Network, worldwide)

Feedback control applications usually comprise the first three levels from BAN to LAN. The second and the third one (PAN, LAN) typically are multi-hop ad-hoc networks, at present using the ZigBee-Standard [12] for example. As a consequence, a highly stochastic behavior results.

Packets can get lost because of the limited network stability. This is not only influenced by the network load generated by the applications, but also by the steadily changing network structure. This means that the need for reconfiguration and for refreshing the routing tables is high. The limited storage of the routing nodes causes more or less packet losses. Additional quasi packet losses result from two reasons: Because of small packet overheads and limited computing resources the FEC-methods (forward error correction) are not common on the lower AmI network levels. Therefore packet corruption can be detected but not corrected. Corrupted packets are like lost packets. A similar effect occurs if one packet passes another because of different routes through the network. In control applications a packet is worthless if it has been passed by a newer one and therefore arrives later with older information. Typical packet loss rates (including quasi packet losses) lie between 20% and 40%.

A second stochastic effect is due to the routes through the ad-hoc network that change steadily resulting in variable transmission delays from originator to addressee. The absolute value of the delay depends on the possibly changing distance between originator to addressee and the number of hopping nodes between them. Values for the network delays may amount from fractions to multiples of the sampling time.

In Fig. 1 the structure of single control loop in a typical AmI-system is shown. The sensor produces data packets in a time triggered way, according to the sampling time T_S [13]. The sensor information (measured value and packet numbering) enters the network heading for the controller. All nodes on the route may also be sensors, other controllers or actuators which are in an advantageous range to serve as a routing node. Having received the measured value the controller computes the actuating value giving it to the same network heading for the actuator. Note that closing the control loop from the AmI-plant via sensor, controller and actuator involves the network twice.

Following the analysis of time delays in NCS-networks given in [13] the main effects for the resulting loop delay can be summarized as

$$\tau = \tau_{SC} + \tau_C + \tau_{CA} \tag{1}$$

with

τ_{SC}: Delay time sensor to controller
τ_C: Delay time for executing the control algorithm
τ_{CA}: Delay time controller to actuator

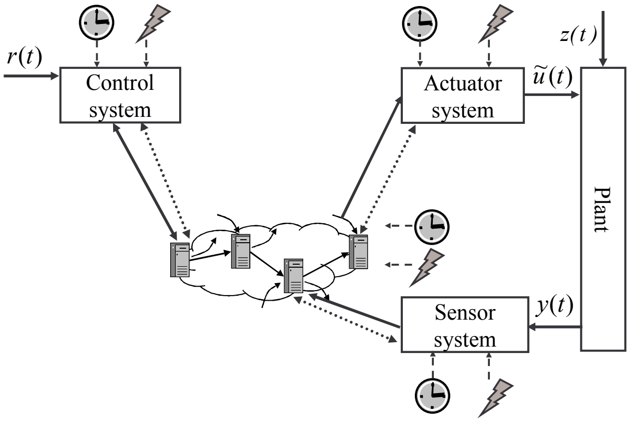

Fig. 1. General architecture of NCS. Clocks and flashes denote time based and event based communication respectively.
r : reference, y : control variable, z : disturbance, $\widetilde{u}$: plant input

The times τ_{SC} and τ_{CA}, respectively, comprise all parts which are relevant for the data transfer from originator to addressee, see [13] for deeper analysis. In particular these are the waiting time due to the medium access control and the routing time due to the multi-hop mechanism. Low control effort means that the controller execution time τ_C may be neglected compared to τ_{SC} and τ_{CA}.

To summarize, in AmI-systems, there are three main effects limiting the control performance and threatening the control stability:

- the limited resolution of measuring values may produce quantization errors,
- the packet transport along different ways with hopping through the nodes creates variable network delays τ (up to multiples of sampling time T_S),
- the changing network stability, passing of packets and packet corruption cause stochastic packet losses (including quasi losses) of 20% to 40%.

Methods for designing controllers in AmI-systems have to master these problems under the AmI-typical constraints like e.g. limited computational power and limited packet length due to the high electrical power amount for sending and receiving data. Possible trade-offs between control performance and control effort are shown in the next sections.

III. ATTEMPTS FOR CONTROL VIA AMI-NETWORKS

Two adaptive attempts are compared. The first one tries to keep the constant sample interval for sensor and actuator and uses Model Predictive Control (MPC)-based control for an adaptation to the network state. The second one relies on standard controllers which are adapted to the network state allowing asynchronous actuating sequences. Both attempts are based on some general mechanisms described next.

A. General Mechanisms

1) Packet Numbering

All data packets contain the main information (process value from the sensor or control value from the controller) and are numbered with packet numbers *pn*, consecutively given by the originators of the information.

This numbering has two advantages:

- Comparing the numbers pn_{new} and pn_{last} of two packets which have been received one after another the addressee can easily decide whether packets have been lost or have been overtaken.
- The use of such packet numbers does not affect the network load significantly if the numbers are assigned in a circular way with limited bit-coding, e.g. 4 bits.

2) Time Stuffing

As opposed to NCS based on deterministic fieldbus technology, AmI systems never behave like classical discrete time control. Disregarding the packet losses for a while and only concentrating on the variable time delays the problem can be approximately solved by the *time stuffing* mechanism. If a constant delay time T_N summarizing all network effects is added to the plant model this simple method enables a correct controller design as long as from Eq. (1)

$$\tau_{SC} + \tau_C + \tau_{CA} < T_N$$

holds. To adjust the real behavior to the theoretical preliminary of a constant delay time T_N the actuator has to wait the time

$$\tau_W = T_N - \left(\tau_{SC} + \tau_C + \tau_{CA}\right) \tag{2}$$

before activating the new value. This mechanism works easily if T_N is chosen a whole-numbered multiple of the sampling time T_S :

$$T_N = iT_S,\ \ \mathrm{i} = 1,2,... \tag{3}$$

While making the theory of disrete time systems applicable – at least in the absensce of packet losses – this method may not yield the best performance because of introducing more delay than necessary. Nevertheless, this mechanism can be combined with other methods as it is done in the next section.

B. MPC-based Adaptive Control

1) Short Overview

A successful implementation of MPC in NCS was reported by Tang and de Silva in [16]. They used the recursive least squares estimator for online model identification.

In this section we introduce the MPC methodology from [17] for NCS where the model of the plant is assumed to be known. The aim of the methodology is to use the property of MPC in such a way that the sensor and the controller are sending packets synchronously and the actuator influences the plant synchronously, as well. The sampling time is T_S for all three components.

2) Detection of Packet Losses and Quasi Packet Losses

Defining two thresholds T_{SC} and T_{CA} with the condition (compare Eq. (2))

$$T_{SC} + T_{CA} = iT_S,\ \ i = 1,2,..., \tag{4}$$

a packet is considered to be lost in the MPC context if it fulfils one of the following criteria:

- $pn_{new} < pn_{last}$ holds for two consecutively received packets (overtaking),
- packet is corrupted,
- $\tau_{SC} > T_{SC}$ for Controller, $\tau_{CA} > T_{CA}$ for actuator.

As long as

$$\tau_{SC} < T_{SC}\ ,\quad \tau_{CA} < T_{CA}$$

and packets are not corrupted and not overtaken the time stuffing is applied for the controller and the actuator, respectively. In this way, the control system is modified from a system with time varying delays to a system with constant delays. The choice of the delays T_{CA} and T_{SC} has to consider the fact that the longer the delays T_{CA} and T_{SC}, the rarer the quasi packet losses. However, it is also well known that a long delay deteriorates the performance of the controller. A trade-off between both effects is needed.

3) Structure of the MPC-based Adaptive Control

By using the above procedure variable time delays greater than the threshold are transformed into quasi packet losses. To solve the packet loss problem, we use the nature of MPC which predicts the future system behaviour and calculates the future control signals. Fig. 2 shows a schematic diagram of the solution. The sensor samples synchronously the outputs of the plant and transmits them into numbered packets to the controller. If there are packet losses or quasi ones, the estimator calculates an approximation based on the defined model and the previous control signal. Therefore the MPC controller can work at every sampling cycle and can produce the control trajectory. Unlike a common MPC that only sends the current control signal, in this case the MPC sends also several future control signals which are stored in the buffer of the actuator. The control signal trajectory is also sent in a numbered data packet. If one or more data packets from the controller are lost, the control signal from the buffer will be used to actuate the system. In this way, the plant can be actuated at synchronous intervals.

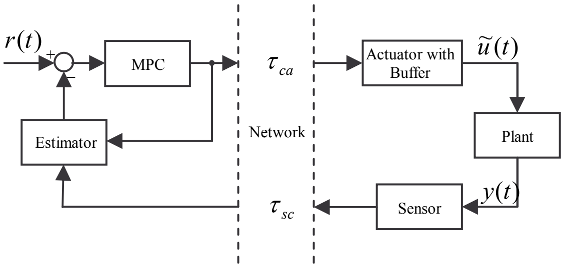

Fig. 2. NCS with MPC-based strategy

C. QoS-adaptive Heuristic Control

1) Overview

First, the mechanisms are introduced how to measure the two crucial QoS-parameters τ (actual delay time) and p_L (actual number of packets lost). From Eq. (1) τ has three parts. Because the adaptation will be based on standard controllers the controller computing time τ_C can be neglected. τ_{SC} will be measured as shown below and will be used as a controller adaptation input. There is no need to measure τ_{CA} because the controller can not react on this information. In the calculation of the controller parameters the expected value $E(\tau_{CA})$ is used instead of τ_{CA} in Eq.(1). Of course, in AmI-systems measuring τ and p_L has to be done without enhancing the network burden.

Starting from a continuous standard controller, the adaptive control is developed in two steps. The first one is the transformation of the controller into the time discrete domain space. In the second step the controller is extended by the delay adaptation and the packet loss adaptation. In the design stage, the parameters for the delay adaptation are optimized by a genetic algorithm. This optimization procedure is performed on the basis of both the plant model and the network model incorporating the stochastic effects described above. Model based optimization is an attractive way because the theory of linear and nonlinear time discrete systems needs constant sampling time and therefore does no longer hold in the AmI-case. It can be shown that QoS-adaptive control is superior to non adaptive control if some preliminaries concerning the rate of change of the QoS-parameters hold. Because the time stuffing mechanism is not used here, the controller and the actuator work asynchronously.

2) *Mechanisms for Measuring QoS*

a) Measuring the Delay Time

The QoS-adaptive approach assumes the following architecture: the sensors work time-triggered, the controller and actuator event-triggered. In other words: the sensors measure with constant sampling time T_s and send packets to the controller. The controller and the actuator are only activated upon reception of a new packet. If the packets are dropped by the network the controller does not evaluate a new control signal and the actuator holds the last received value.

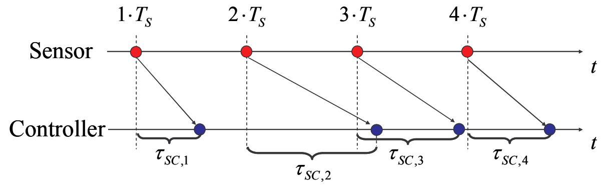

Fig. 3: Definition of network induced delays

The controller can calculate the network induced delay as shown in Fig. 3. Of course, this mechanism assumes that all components of the NCS in the same control loop are synchronized and have the same time base.

b) Measuring the Lost Packets

Whenever a receiver obtains a corrupted packet, this will be discarded. For the packets which are not corrupted the actual number of lost packets is

$$p_L = pn_{new} - pn_{last} - 1 \quad \forall \; pn_{new} > pn_{last} . \qquad (5)$$

With

$$pn_{new} < pn_{last}$$

the actual packet has been overtaken and is discarded.

In order to enhance the control performance in spite of the presence of delays and packet losses, the extensions of the NCS are described in the following

3) *Controller Design*

a) General Policy

Assume that the AmI-NCS has the architecture depicted in Fig. 1 and that we have a stable continuous controller with n parameters designed by standard methodologies. Then a discrete form of the controller is derived by transformation, for example with Tustin's method. Moreover, we know the probability density function $f_T(\tau)$ of the network delays. Then it is possible to increase the control performance by the following heuristics that have been motivated in [13] and [14].

b) Heuristic 1

Old packets that are overtaken by newer packets carrying newer process information must be dropped because newer process information is better than older one. Corrupted packets are worthless because in AmI systems there is no mechanism for an FEC (Forward Error Correction).

c) Heuristic 2

As a first step, we choose l values of τ distributed over a range between $\tau_{\min}$ and $\tau_{\max}$:

$$\tau_j \in \left[\tau_{\min}, \tau_{\max}\right] \quad j = 1, \ldots, l \qquad (6)$$

Note that with $\tau_{\max}$ only the upper bound of the distribution function is mentioned and not any boundry in relationship with the sampling time. As second step, we add these in the control loop and model them as constant delays. For each τ_j we search the n optimal control parameters that maximize a certain optimization criteria by simulation. In that way, we get l tuples of the form

$$\mathbf{k}_{\tau j} = (k_{j1} \ldots k_{jn}) \,, \quad j = 1, \ldots, l \,. \qquad (7)$$

Each n-tuple belongs to a specific constant delay and represents the optimal set of parameters in terms of optimization criteria.

Now, for each packet we measure the delay τ_{SC} and depending on

$$\tau = \tau_{SC} + E\left(\tau_{CA}\right) \qquad (8)$$

we choose the optimal set of parameters. Since there are only l tuples but the delays are continuous we always choose the set of parameters that is closest to the measured delay. This leads to a switching controller that changes the control parameters every time the measured τ_{SC} has significantly changed.

d) Heuristic 3

To deal with the packet losses we use an adaptation strategy. So, instead of using a fixed sampling time we adjust the sampling time according to the occurrence of packet losses. The sampling time is then derived by

$$T_m = \left(p_L + 1\right) T_S \,, \qquad (9)$$

where T_m is the sampling time used in the m^{th} calculation of the controller. If no packet losses occur, $p_L = 0$ and T_m is equal to T_s.

e) Heuristic 4

Remember that the controller is event-triggered. So, it is no longer necessary to wait for the next invocation of the controller by a new sensor packet when the set point has been changed. The occurrence of a new set point triggers the controller directly.

IV. THE NUMERICAL TEST EXAMPLE

A. Chosen Plant and Network Model

The following example is taken from [1] and was also used in [13]. The plant representing a DC drive is given by

$$G(s) = \frac{2029.826}{(s+26.29)(s+2.296)} . \tag{10}$$

A disturbance $z(t)$ is added to the output of the plant with $\mu_S = 10$ and $\sigma_S^2 \in \{3,10\}$. The network delays vary uniformly distributed with $\tau \in [10ms, 110ms]$ yielding expected values $E(\tau)$ = $60ms$ and $E(\tau_{sc})$ = $E(\tau_{ca})$ = $30ms$ if the computing time τ_C can be neglected. The network packet loss rates p_{SC} and p_{CA} are set to 20%. For all controllers the sampling time T_s is set to $30ms$.

B. MPC-based Adaptive Control

For the simulation we choose a special MPC algorithm from [17], the so called unified approach algorithm with an infinite horizon for both the control and the prediction horizon. The control signal is obtained by minimizing the cost function:

$$J = \sum_{i=1}^{\infty} \left(y(k+i) - r \right)^2 + 5000 \Delta\tilde{u}(k+i-1)^2 \tag{11}$$

Although the control horizon is infinite, only up to three steps in the future of the control signal trajectory are sent through the network. The thresholds T_{SC} and T_{CA} are chosen to be 45 *ms* each. Thus, the model has a total dead time of 90 *ms*. The optimal control is calculated analytically and can be expressed as difference equation for $\Delta\tilde{u}$ and y since no constraints are considered.

C. QoS-adaptive Heuristic PI-Control

By using Tustin's method we obtain the discrete algorithm

$$\begin{aligned} u(kT_S) = u((k-1)T_S) &+ \left[k_P + 0.5 \cdot k_I \cdot T_S \right] \cdot e(kT_S) \\ &+ \left[0.5 \cdot k_I \cdot T_S - k_P \right] \cdot e((k-1)T_S) \end{aligned} \tag{12}$$

for a PI-controller with transfer function

$$G_{PI}(s) = k_P + k_I / s . \tag{13}$$

In eq. (12) e is the control error. The optimal control parameters are calculated for five different delays

$$\tau_i / ms \in [35, 50, 65, 80, 95] \tag{14}$$

resulting in five parameter sets [$k_P(\tau_i)$, $k_I(\tau_i)$].

Based on the measured τ_m the parameters $k_P(\tau_m)$, $k_I(\tau_m)$ in the control algorithm for m-th step,

$$\begin{aligned} u_m = u_{m-1} &+ \left[k_P(\tau_m) + 0.5 \cdot k_I(\tau_m) \cdot T_m \right] \cdot e_m \\ &+ \left[0.5 \cdot k_I(\tau_{m-1}) \cdot T_{m-1} - k_P(\tau_{m-1}) \right] \cdot e_{m-1} \end{aligned} \tag{15}$$

are approximated by those $k_P(\tau_i)$, $k_I(\tau_i)$ for which τ_i in Eq. (14) is nearest to τ_m.

V. SIMULATION RESULTS

For comparison reasons a normalized measure is defined to jugde the control performance, called quality-of-control,

$$QoC = 100 \left[1 - \frac{error}{error_{\max}} \right] , \tag{16}$$

with

$$error = (1-\lambda) \cdot IAE \; + \; \lambda \cdot ITAE . \tag{17}$$

A. Comparison of MPC- and PI-based Control

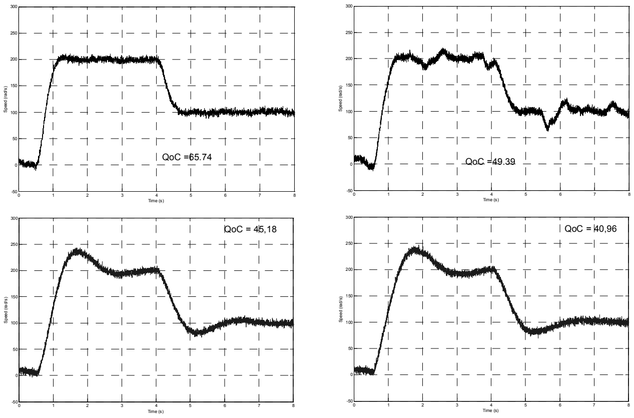

Fig. 4: Step Responses for
MPC-based Control without network (top left) QoC = 65,74
PI-based Control without network (bottom left) QoC = 46,18
MPC-based adaptive Control with network (top right) QoC = 49.39
PI-based adaptive Control with network (bottom right) QoC = 40.96
Noise with $\sigma^2 = 10$

To show the best results that can be achieved with both controllers the best case is shown, a fictitious one without time delays and without packet losses, see Fig. 4. As expected, the MPC-based result above is superior with a much better QoC. At the same time the computational amount when carrying out the MPC-based control is higher than that of PI-control.

Including the network as described before the performance deteriorates with both, see Fig. 4. But surprisingly, the difference between the two results has become very small, as can also be seen by the QoC values.

This result is not by accident. To show this, 10 simulations have been performed for each controller with 10 different start values of the random generators. The results are compared by the QoC-values. We get

$$\overline{QoC_{MPC}} = 42.64, \quad \mathrm{var}(QoC_{MPC}) = 27.56 , \tag{18}$$

$$\overline{QoC_{PI}} \; = \; 38.08, \quad \mathrm{var}(QoC_{PI}) = 16.97 . \tag{19}$$

B. Noise Sensitivity of MPC- and PI-based Control

The reason can be analyzed when comparing the sensitivity to noise in the presence of packet losses and variable delays. This is shown in Fig. 5 for the case of control over network with packet losses and variable delays. There is a big difference between noisy (QoC = 47.26) and noise free (QoC = 73.22) case with the MPC-algorithm (Fig. 5 above). In contrast, for the PI-based solution the difference between the noisy (QoC = 44.44) and the noise free case (QoC = 46.77) can hardly be detected.

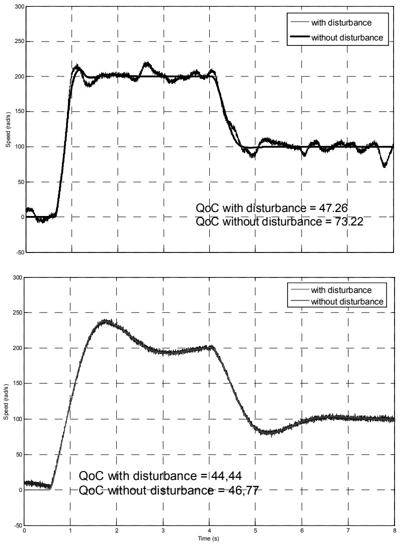

Fig. 5: Step Responses for
MPC-based Control with and without noise (above)
PI-based Control with and without network (below)
Noise with $\sigma^2 = 3$

The high noise sensibility of the MPC-algorithm together with variable delays and packet losses is inherent in the forgoing. The predicted control values to overcome the packet losses are the more irrelevant the higher the noise ratio is. Control values are always predicted without knowledge of the future noise trajectory.

VI. CONCLUSIONS

AmI-systems are special NCS with additional restrictions originating from the AmI-nodes and the AmI-network as well. Even with these challenges – e.g. variable delays till multiples of the sampling time and packet losses till 40% in the whole loop – satisfactory controllers can be designed. In addition to the classical NCS trade-offs [3] between control, physical layer and MAC-layer, much more trade-offs are needed. They relate to the control effort and may favor algorithms with small computational amount like the shown PI-based QoS-adaptive control against more comfortable ones.

Concerning the two attempts introduced and compared in this paper we conclude:

- The MPC-based solution is superior from the QoC-point of view, especially with low noise.
- The QoC of the MPC-based solution deteriorates rapidly when the noise is increased.
- The PI-based solution can also manage high noise ratios and needs a very low computational amount which is independent of the model order.

There are also network specific trade-offs hidden in the respective methods influencing the performance to a high extend. As an example for the MPC-method consider the integer *i* in Eq. (4) which was chosen to 3 in the example. The best choice depends on the plant dynamics in connection with the network properties like probability density function of the delays and the packet loss rate.

As a consequence, designing the best solution for a given problem is a crucial task. The optimum depends on the plant dynamics, the chosen type of control algorithm, the network properties and the typical AmI-constraints, only to mention the most important influences. Because of many degrees of freedom with the related trade-offs there is no direct way to find best solution. To come near the optimum is a challenging engineering task, which can only be solved iteratively.

REFERENCES

[1] Tipsuwan, Y.; Chow, M.-Y.: Control methodologies in networked control systems, Control Engineering Practice 11, p.1099-1111, 2003.

[2] Hokayem P. F.; Abdallah C. T. : Inherent Issues in Networked Control Systems: A Survey. Proceedings of the 2004 American Control Conference, Boston, p. 4897-4902, 2004.

[3] Liu, X.; Goldsmith, A.: Wireless Network Design for Distributed Control. Proceedings of the IEEE Conference on Decision and Control, p. 2823 – 2829, 2004

[4] Remagnino, P.; Foresti, G. L.: Ambient Intelligence – A new Multidisciplinary Paradigm. IEEE Trans. Systems, Man, and Cybernetics, Part A, Vol 35, p. 1- 6, 2005.

[5] Remagnino, P.; Foresti, G. L. ; Ellis, T. (Edts): Ambient Intelligence - A Novel Paradigm. Springer Verlag, Berlin, 240 p., 2005.

[6] Riva, G.; Loreti, P.; Lunghi, ,M.; Vatalaro, F.; Davide, F.: Presence 2010: The Emergence of Ambient Intelligence. In: Riva, G. ; Davide, F. ; IJsselsteijn W.A (Eds.): Being There, Concepts, effects and measurement of user presence in synthetic environments. Ios Press, Amsterdam, 2003.

[7] Litz, L.; Wehn, N.; Schürmann, B.: Research Center "Ambient Intelligence" at the University of Kaiserslautern. Tagungsband VDE Kongress 2004, Berlin: VDE Verlag, Band 1, p. 19-24, see also http://www.eit.uni-kl.de/AmI/Li-we-sch-2004/index.html,.

[8] Barger, S.T.; Brown, D.E.; Alwan, M.: Health Status Monitoring Through Analysis of Behavioral Patterns. IEEE Trans. Systems, Man, and Cybernetics, Part A, Vol 35, p. 22- 27, 2005.

[9] Ross, P.E.: Managing Care through the Air. IEEE Spectrum Dec. 2004, p. 26 – 31.

[10] Hill, J. L.; Culler, D. E.: "Mica: A Wireless Platform for Deeply Embedded Networks", in IEEE Micro 22, no. 6, pp.12-24, 2002.

[11] University of Berkely: Mica-Homepage, http://webs.cs.berkeley.edu/tos, accessed on: 15.02.2005.

[12] ZigBee® Alliance, Wireless Control that simply works http://www.zigbee.org/en/index.asp

[13] Gabel, O.; Litz, L.: Adaptive Control in NCS with Variable Delays and Packet Losses - A Heuristic Approach. Proceedings of the IEEE Conference on Decision and Control, p. 1586 – 1591, 2004.

[14] Gabel, O.; Gerkhardt, D.; Litz, L.: Feedback Control in AmI-Networks – Managing the Limited Resource Problem by QoS-Adaptive Control, Proceedings of the IIIS Conference on Computing Communications and Control Technologies, pp. 588-593, 2005.

[15] TU Kaiserslautern: *Website of the Research Center "Ambient Intelligence"*. http://www.eit.uni-kl.de/ami/, 2004.

[16] Tang, P. L.; de Silva, C. W.: Ethernet-based Predictive Control of an Industrial Hydraulic Machine, IEEE Conference on Decision and Control, Maui, Hawaii,p. 695-700, 2003.

[17] Solihin, I: Unified Approach Algorithm for Model Predictive control. Dissertation at University of Kaiserslautern, Electrical and Computer Engineering, Germany, March 2005.