STABLE SAMPLING PERIOD ADAPTATION FOR ENERGY SAVING IN AMBIENT INTELLIGENCE NETWORKS

THOMAS STEFFEN, LOTHAR LITZ

Institute of Automatic Control, Department of Electrical Engineering, University of Kaiserslautern
Erwin-Schrödinger-Str. 12, 67663 Kaiserslautern, GERMANY
E-mails: steffen@eit.uni-kl.de, litz@eit.uni-kl.de

JOÃO Y. ISHIHARA, ADOLFO BAUCHSPIESS

GRAV, Departamento de Engenharia Elétrica, Universidade de Brasília
70910-900 Brasília –DF, BRAZIL
E-mails: ishihara@ene.unb.br, adolfobs@unb.br

Abstract— In this paper, a new control scheme aiming at a stable sampling period adaptation for energy saving in Ambient Intelligence (AmI) Wireless Networked Control Systems (WNCS) is proposed. We introduce a new device to generate the control output, the "Smart Actuator", which is based on the explicit use of a system model in the controller-actuator unit. The model is updated with the plant state via an AmI wireless network. We show that even with a simple heuristic adaptation law for the sampling period, significant enhancement in energy saving is obtained. Also, it opens the possibility of a systematic AmI system design concerning a trade-off between sensor energy saving and control performance. The new control framework was tested in simulations to show its effectiveness.

Keywords— Ambient intelligence, Wireless Networked Control Systems, model-based control, sampling period adaptation

1 Introduction

In recent years, the field of networked control systems (NCS) has seen vivid research activities, see, e.g., Hespanha et al. (2007) for a survey. A natural extension of NCS are wireless NCS (WNCS), which are given some attention, e.g., by Song et al. (2009). In WNCS several additional limitations may increase the complexity of the control problem. Among them are limited energy resources, limited computational power, the lack of powerful protocols (which result in, e.g., a lack of time synchronization). These limitations appear especially in the so called Ambient Intelligence (AmI) systems, see Litz et al. (2005).

AmI systems interact with humans and are thus sensitive and usually adaptive. Typical AmI systems consist of several cheap wireless nodes, which are very small, have low computational capabilities and limited transmission range. They either run on battery or harvest energy from their environment. Therefore, energy saving has a high priority in these systems. Low power transmitters, as well as simple protocols contribute to this goal. Furthermore, AmI nodes feature a so called sleeping mode in which no packets can be sent or received and no computations can be carried out. In sleeping mode extremely few energy is being consumed, thus it should be used as often as possible. For the use in control this implies that sampling periods should be maximized under the constraint of an acceptable QoC and inter-sampling computations must be avoided.

An approach to overcome some of the AmI-WNCS problems is the incorporation of a system model in the control scheme, such as model-based NCS (MB-NCS), which was first proposed by Montestruque and Antsaklis (2003).

Model uncertainties and disturbances are the main reasons why feedback control is used instead of feed-forward control. However, both model uncertainties and disturbances are of varying impact on the control performance depending on the state of the system. This insight gives rise to the idea, that the control loop is only closed when necessary to avoid large differences between real system and model. This idea is followed by Lunze and Lehmann (2010), where asynchronous sampling is triggered by the deviation between the system and the model. One drawback of this event-based control scheme is the necessity of an additional system model in the sensor, which has to be synchronized with the model in the actuator. Furthermore, the discrete event character prevents the sensor from using the energy efficient sleeping mode, because the system state must be tracked continuously.

Event-based control results in varying sampling times. But also other control schemes with varying sampling times were studied in the literature. Zhiwen et al. (2007) use the framework of MB-NCS, however they assume that the sensor sends measurements at a constant rate, whereas the inputs are applied to the plant with different sampling periods. Colandairaj et al. (2007) adjust the sampling period according to the latest round trip time in order to achieve a maximal usage of the network. They give a system description by means of Markov Jump Linear Systems and prove stability in a probability sense.

Our approach aims at a minimal use of sensor energy and uses the framework of model-based control. So far, we combine AmI-like low cost, low performance sensors which may sleep most of the time during the sampling period with smart actuators performing model-based control and producing the adaptation command. By using an adaptive sampling period, our approach can be more effective than MB-NCS in the sense that it implements a dynamic trade-off between QoC and network occupation. Achieving the same QoC with fewer sampling instances directly translates into less energy usage in the AmI sensor and thus longer runtime of sensors.

Compared to the work of Lunze and Lehmann (2010) our scheme omits the model in the sensor for the reasons given above. It adapts the sampling period by a heuristic adaptation law in order to save electrical sensor energy and network bandwidth and at the same time, which makes it applicable in AmI-WNCS systems.

As a main result, a stability proof for arbitrary switching within a set of sampling periods is given. Although our proof uses a different technique, it resembles the result of Montestruque and Antsaklis (2003) for constant sampling period.

The paper is structured as follows: Section 2 describes the system structure and the goal of the control scheme. An adaptation policy for linear SISO systems is described. In Section 3 a stability proof for these systems is given. In Section 4 the adaptation policy is studied. A design methodology based on pareto optimization is presented in an example. Finally the effectiveness of our approach in the example is tested. Section 5 summarizes the results and gives an outlook on future work.

2 Problem Formulation

In this section, the overall system structure will be described. The functionality of the “Smart Actuator” and the AmI Sensor are explained in detail. A mathematical formulation of the control scheme is given for arbitrary adaptation policies. Finally an adaptation policy for linear SISO systems is proposed.

2.1 Control System Architecture

The system structure is shown in Figure 1. It contains the plant, which is subject to a disturbance, a so called “Smart Actuator”, which is described in details later and an AmI sensor, which sends a measurement of the plant state periodically. The sampling period of the sensor can be adjusted by a notification of the Smart Actuator. This possibility is the major difference compared to the control scheme of Montestruque and Antsaklis (2003), as it allows the sampling period to be adapted to the current system state. Besides from receiving a message and adjusting the sampling period accordingly, the sensor is quite simple. Particularly, it does not simulate a system model, which saves computational effort and allows the AmI sensor to go to sleep mode between sampling instances. Consequently energy can be saved, which is of special interest if the sensor runs on batteries or uses energy harvesting.

The Smart Actuator consists of a plant model which is controlled by a state-feedback controller. Its control input is applied to both, the model and the plant. The Smart Actuator embodies the controller and the actuator, as well as the adaptation unit, which executes the adaptation policy. As the actuator is usually connected to a power supply, energy saving or computational complexity concerning the actuator were not given further attention throughout this paper. Whenever a new measurement is received, the model is updated with the state measurement. Between two sampling instances, the system runs model-driven open loop. This method is similar to the one used by Montestruque and Antsaklis (2003), as well as by Lunze and Lehmann (2010) and can be considered as a reasonable compromise between feedback and feed-forward control.

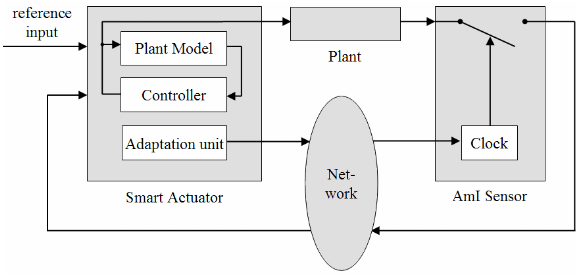

Figure 1: Structure of the system

The adaptation unit incorporated in the Smart Actuator notifies the sensor in case the sampling period should be adjusted. This is done directly after the new sensor values have been received, to allow the sensor to sleep until just before the end of the sampling period for transmitting the next sensor values. The adaptation unit executes the adaptive sampling policy, which is subject to the design procedure. The policy described in Section 2.3 is just one possible solution to show the effectiveness in the example of Section 4.

It is important to note that the stability proof given in Section 3 holds for arbitrary adaptation policies. This gives the control designer the great flexibility to customize the adaptation policy due to the characteristics of a specific plant and a specific network.

2.2 Mathematical Description

In this section, we set the basic description for general set of sampling times without concern about any particular adaptation policy.

The plant is described by

$$\dot{x} = Ax + Bu_S \,, \tag{1}$$

where $x \in \Re^n$ is the plant state and $u \in \Re^m$ is the system input vector. A model of the plant is assumed to be available and described by

$$\dot{x}_S = \hat{A}x_S + \hat{B}u_S \,, \tag{2}$$

where $x_S \in \Re^n$ is the model state and $u_S \in \Re^m$ is the model input vector.

An appropriate stable state-feedback controller is designed and applied to both the model and the plant ($u_S = u$). The control law is

$$u_S = Kx_S \,, \tag{3}$$

where $K \in \Re^{mxn}$. Combining equations (1) - (3) leads to

$$\dot{x} = Ax + BKx_S \,, \tag{4}$$

$$\dot{x}_S = (\hat{A} + \hat{B}K)x_S \tag{5}$$

Introducing the state error

$$e = x - x_S \,, \tag{6}$$

the modeling error matrices

$$\tilde{A} = A - \hat{A} \tag{7}$$

and

$$\tilde{B} = B - \hat{B} \tag{8}$$

the system behavior between the updates can be described by:

$$\begin{pmatrix} \dot{x}(t) \\ \dot{e}(t) \end{pmatrix} = \begin{pmatrix} A + BK & -BK \\ \tilde{A} + \tilde{B}K & \hat{A} - \tilde{B}K \end{pmatrix} \begin{pmatrix} x(t) \\ e(t) \end{pmatrix} \tag{9}$$

The update scheme can be expressed by the following equation

$$\begin{pmatrix} x(t_k) \\ e(t_k) \end{pmatrix} = \begin{pmatrix} x(t_k^-) \\ 0 \end{pmatrix}, \tag{10}$$

where t_k are the sampling instances. Introducing the variables $z(t)$ and Λ defined as

$$z(t) = \begin{pmatrix} x(t) \\ e(t) \end{pmatrix} \tag{11}$$

$$\Lambda = \begin{pmatrix} A + BK & -BK \\ \tilde{A} + \tilde{B}K & \hat{A} - \tilde{B}K \end{pmatrix} \tag{12}$$

simplifies equation (9) to

$$\dot{z}(t) = \Lambda z(t) \,. \tag{13}$$

The trajectory $z(t)$ of this system is of the form

$$z(t) = e^{\Lambda(t - t_k)} z(t_k) \,, \tag{14}$$

where t_k is the last sampling instant.

According to equation (10), $z(t_k)$ is known to be

$$z(t_k) = \begin{pmatrix} I & 0 \\ 0 & 0 \end{pmatrix} z(t_k^-) \,. \tag{15}$$

$z(t_k{}^-)$ can be described by

$$z(t_k{}^-) = e^{\Lambda(t_k^- - t_{k-1})} z(t_{k-1}) \,. \tag{16}$$

Combining equations (15) and (16) the trajectory $z(t)$ can be expressed by

$$z(t) = e^{\Lambda(t - t_k)} \begin{pmatrix} I & 0 \\ 0 & 0 \end{pmatrix} e^{\Lambda(t_k^- - t_{k-1})} z(t_{k-1}) \,. \tag{17}$$

$t_k^- - t_{k-1}$ is the time interval between the k^{th} and $(k\text{-}1)^{th}$ update and is abbreviated by h_k from now on.

The sampling period h_k is chosen from a set $h_k \in \{h_{\min}, ..., h_{\max}\}$ by the adaptation unit.

Because $z(t_{k-1}) = \begin{pmatrix} I & 0 \\ 0 & 0 \end{pmatrix} z(t_{k-1})$ holds, equation (17) can be rewritten as

$$z(t) = e^{\Lambda(t - t_k)} \begin{pmatrix} I & 0 \\ 0 & 0 \end{pmatrix} e^{\Lambda h_k} \begin{pmatrix} I & 0 \\ 0 & 0 \end{pmatrix} z(t_{k-1}) \,. \tag{18}$$

From equation (18) it is easy to see that

$$z(t_k) = \begin{pmatrix} I & 0 \\ 0 & 0 \end{pmatrix} e^{\Lambda_k h_k} \begin{pmatrix} I & 0 \\ 0 & 0 \end{pmatrix} z(t_{k-1}) \,. \tag{19}$$

Iterating the recursive equation (19) leads to

$$z(t) = e^{\Lambda(t - t_k)} \prod_{i=1}^{k} \left\{ \begin{pmatrix} I & 0 \\ 0 & 0 \end{pmatrix} e^{\Lambda h_{k-i}} \begin{pmatrix} I & 0 \\ 0 & 0 \end{pmatrix} \right\} z(t_0) \,, \tag{20}$$

which describes the controlled system for all times.

$$z(t_0) = \begin{pmatrix} x(0) \\ 0 \end{pmatrix} \tag{21}$$

denotes the initial condition of the system.

Note 1:

If $h_k = h \; \forall k$, equation (18) becomes:

$z(t) = e^{\Lambda(t - t_k)} \left\{ \begin{pmatrix} I & 0 \\ 0 & 0 \end{pmatrix} e^{\Lambda h} \begin{pmatrix} I & 0 \\ 0 & 0 \end{pmatrix} \right\}^k z(t_0)$ and thus resembles the result of Montestruque and Antsaklis (2003).

So far, a common controller $u_S = Kx_S$ is assumed to be used for all sampling periods. This assumption can be given up though, in order to make the control scheme more general and give more flexibility to the control designer. Therefore, it is assumed that an individual controller is assigned to every sampling period:

$$u_S = K(h_k) x_S \,, \tag{22}$$

where $K(h_k) : \{h_{\min}, ..., h_{\max}\} \rightarrow \Re^{mxn}$

So equation (19) becomes

$$z(t_k) = \begin{pmatrix} I & 0 \\ 0 & 0 \end{pmatrix} e^{\Lambda_k h_k} \begin{pmatrix} I & 0 \\ 0 & 0 \end{pmatrix} z(t_{k-1}) \,. \tag{23}$$

Let

$$\Phi_k = \begin{pmatrix} I & 0 \\ 0 & 0 \end{pmatrix} e^{\Lambda_k h_k} \begin{pmatrix} I & 0 \\ 0 & 0 \end{pmatrix}, \tag{24}$$

where

$$\Lambda_k = \begin{pmatrix} A + BK(h_k) & -BK(h_k) \\ \tilde{A} + \tilde{B}K(h_k) & \hat{A} - \tilde{B}K(h_k) \end{pmatrix}. \tag{25}$$

A system described by equations (1), (2), (6) - (8), (11), (21) and (22) - (25) will be called adaptive sampling period model-based controller (AS-MB-NCS) from now on.

2.3 The Proposed Adaptation Policy

Aim of the AS-MB-NCS is to reduce the communication of the sensor, while maintaining an acceptable QoC. The Integral of Absolute Error (IAE) defined as $IAE = \int \left| r(t) - y(t) \right| dt$, where $r(t)$ is the reference value and $y(t)$ is the system output, serves as QoC measure. The sensor communication cost will be measured by the number of packets sent in a certain time period. A trade-off between IAE and sampling rate to characterize the quality of a control scheme was seen in the literature before, e.g., Peng et al. (2009).

Although the stability proof holds for arbitrary adaptation policies, the choice of adaptation rules largely influences the performance of the AS-MB-NCS. Therefore, the adaptation policy should be chosen wisely. In general, an analysis of the system is necessary either by theoretically derived properties or simulations in order to identify the critical situations, that should be addressed by the adaptation policy. In this paper, we present a simple but effective adaptation policy which is expected to perform well for many linear SISO systems. It makes use of two sampling periods h_{slow} and h_{fast} and is based on the fact that in AmI WNCS sending measurements by the sensor is a costly process.

Whenever a plant measurement is received, the Smart Actuator gets information about how well its model fits reality. In case of full state feedback, the deviation between every state variable of model and plant is available. A threshold for the absolute error between model and system state can serve to decide whether fast or slow sampling is carried out. As the deviation relies on the length of the sampling interval, it might be appropriate to use individual thresholds for fast and slow sampling. By this adaptation policy the efficient use of network bandwidth is expected as well as energy saving aspects from the sensor point of view. Another adaptation rule forces the sampling rate to h_{fast} whenever the reference value changes. In other words, slow sampling is only carried out during stationary inputs. A similar rules has shown to be effective in Litz et al. (2005). The adaptation rules are given below:

1. If $\dfrac{dr(t)}{dt} \neq 0$, then $h = h_{fast}$ (26)
2. If $h = h_{slow} \wedge \left| x - x_M \right| > c_1$, then $h = h_{fast}$ (27)
3. If $h = h_{fast} \wedge \left| x - x_M \right| < c_2$, then $h = h_{slow}$ (28)

3 Stability

Stability of the AS-MB-NCS can be shown by showing that trajectory (20) converges. However, showing convergence of a product of matrices is not easy, because in general, matrices do not commute. Therefore, a different approach was taken. An energy function like in Laypunov theory measures the system energy. If it can be shown that this energy decreases between any two consecutive sampling instances, the system is stable. This could be shown for AS-MB-NCS.

Lemma:

Suppose q matrices $\Phi_i \in \Re^{nxn}$ $i \in \{1,...,q\}$, were given. If there exists a matrix $P \in \Re^{nxn}, P > 0$ such that $P - \Phi_i P \Phi_i > 0 \;\; \forall i \in \{1,...,q\}$, then the system $z_{k+1} = \Phi_k z_k$ is stable for any $\Phi_k \in \{\Phi_1,...,\Phi_q\}$.

Proof:

Choose a positive definite energy function $V(z) = z^T P z,\; P > 0$. The system given by $z_{k+1} = \Phi_k z_k$ is stable if inequality

$$V(z_{k+1}) < V(z_k) \;\; \forall k \tag{29}$$

holds, that is the system energy decreases between every two consecutive sampling instances.
With the chosen energy function, the inequality becomes

$$z_{k+1}{}^T P z_{k+1} < z_k{}^T P z_k \tag{30}$$

$$z_k{}^T \Phi_k{}^T P \Phi_k z_k < z_k{}^T P z_k \tag{31}$$

$$\Phi_k{}^T P \Phi_k < P \tag{32}$$

$$P - \Phi_k{}^T P \Phi_k > 0 \tag{33}$$

From this Lemma a stability criterion for the adaptive sampling period model-based control scheme can be derived.

Stability criterion:

An AS-MB-NCS, described by equations (1), (2), (6) - (8), (11), (21) and (22) - (25) is stable if a positive definite matrix P can be found, such that the following inequality holds:

$$P - \left(\begin{pmatrix} I & 0 \\ 0 & 0 \end{pmatrix} e^{\Lambda_k h_k} \begin{pmatrix} I & 0 \\ 0 & 0 \end{pmatrix} \right)^T P \begin{pmatrix} I & 0 \\ 0 & 0 \end{pmatrix} e^{\Lambda_k h_k} \begin{pmatrix} I & 0 \\ 0 & 0 \end{pmatrix} > 0 \tag{34}$$

, where

$$\Lambda_k = \Lambda(h_k) = \begin{pmatrix} A + BK(h_k) & -BK(h_k) \\ \tilde{A} + \tilde{B}K(h_k) & \hat{A} - \tilde{B}K(h_k) \end{pmatrix} \tag{35}$$

for all $h_k \in \{h_{\min},...,h_{\max}\}$, that is, for all sampling periods used in the AS-MB-NCS controller.

Note 2:

Stability of every matrix $\Phi_k = \begin{pmatrix} I & 0 \\ 0 & 0 \end{pmatrix} e^{\Lambda_k h_k} \begin{pmatrix} I & 0 \\ 0 & 0 \end{pmatrix}$ is a necessary condition for overall stability.

Note 3:
If no adaptation policy is carried out and only one sampling time being used, equation (34) simplifies to $P - \Phi^T P \Phi > 0$, which is known to be fulfilled if and only if the eigenvalues of Φ lie within the unit circle. This shows that our result contains the result of Montestruque and Antsaklis (2003) as a special case.

Note 4:
The stability criterion ensures that the system energy decreases between any two consecutive sampling instances independent of the particular matrix $\Phi_k = \begin{pmatrix} I & 0 \\ 0 & 0 \end{pmatrix} e^{\Lambda_k h_k} \begin{pmatrix} I & 0 \\ 0 & 0 \end{pmatrix}$, as long as h_k is chosen from the set of allowed sampling periods, that is $h_k \in \{h_{\min}, \dots, h_{\max}\}$. This means, that stability is independent of the choice of the adaptation policy.

4 Simulation

In order to show the effectiveness of the proposed control scheme, consider the following second order system $G_{PT2}(s) = \frac{1}{(1+Ts)^2}$, where $T = 0.2$, that can be represented in the following state-space form:

$$A = \begin{pmatrix} 0 & 1 \\ -25 & -10 \end{pmatrix},\ B = \begin{pmatrix} 0 \\ 25 \end{pmatrix},\ C = \begin{pmatrix} 1 & 0 \end{pmatrix},\ D = 0 .$$

The model of the system has a dynamic error but correct gain: $\hat{G}_{PT2}(s) = \frac{1}{(1+\hat{T}s)^2}$, where $\hat{T} = 0.4$.

The corresponding state space representation is:

$$\hat{A} = \begin{pmatrix} 0 & 1 \\ -6.25 & -5 \end{pmatrix},\ \hat{B} = \begin{pmatrix} 0 \\ 6.25 \end{pmatrix},\ C = \begin{pmatrix} 1 & 0 \end{pmatrix},\ D = 0 .$$

In this example, a controller independent of the sampling period is used for simplicity. $K = \begin{pmatrix} -3 & -0.4 \end{pmatrix}$ is the controller law, which moves the poles of the system from -5 to -10. Both systems together are described by the system matrix

$$\Lambda = \begin{pmatrix} 0 & 1 & 0 & 0 \\ -100 & -20 & 75 & 10 \\ 0 & 0 & 0 & 1 \\ -75 & -12.5 & 50 & 2.5 \end{pmatrix}. \quad \Phi_h = S e^{\Lambda h} S, \text{ where}$$

$$S = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$ and h is the sampling period.

According to the stability criterion presented in Section 3, for stability of an AS-MB-NCS system, the existence of an energy function $P > 0$ must be shown, such that $P - \Phi_{slow}{}^T P \Phi_{slow} > 0 \wedge P - \Phi_{fast}{}^T P \Phi_{fast} > 0$.

Let $P = \begin{pmatrix} 1 & 0.13 & 0 & 0 \\ 0.13 & 0.04 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} > 0$.

For the chosen P, Figure 2 shows the real part of the smallest eigenvalue of the matrix $P - \Phi_h{}^T P \Phi_h$ for sampling periods h between 0 and 500 ms. It reveals that $P - \Phi_h{}^T P \Phi_h$ is positive definite for all sampling periods, thus the AS-MB-NCS in this example is stable for any choice of h_{slow} and h_{fast} within this range.

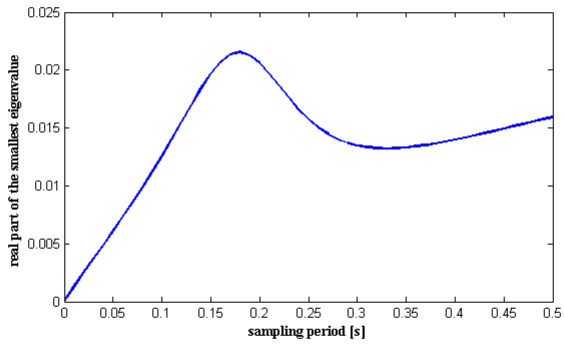

Figure 2: Real part of the smallest eigenvalue of $P - \Phi_h{}^T P \Phi_h$

In order to find suitable sampling periods h_{slow} and h_{fast}, an optimization setup was used. First, a reference trajectory was defined. In practice, such a trajectory should be as realistic as possible. The reference trajectory used throughout this example incorporates 10 steps of different heights. An important feature of this trajectory is, that the time between the steps is varying. This ensures that no sampling period is favoured, in the sense that it always sends a measurement of the plant right after a change in the reference trajectory, when a measurement is most valuable. Note that both MB-NCS and AS-MB-NCS are time-varying systems.

Using the reference trajectory, MB-NCS was simulated with different constant sampling periods. Measurement noise was not incorporated. The IAE was calculated from the simulation of length 60 s. As expected, the QoC decreases with rising sampling periods, see the blue dots in Figure 3.

In the next step, AS-MB-NCS was simulated with different combinations of h_{slow}, h_{fast}, c_1, and c_2. The results are depicted as red diamond shapes in Figure 3. Sampling period in this case means the average sampling period during simulation, which is $h_{avg} = \frac{p}{T_{sim}}$, where p is the number of sampling instances during the simulation and T_{sim} is the simulation time.

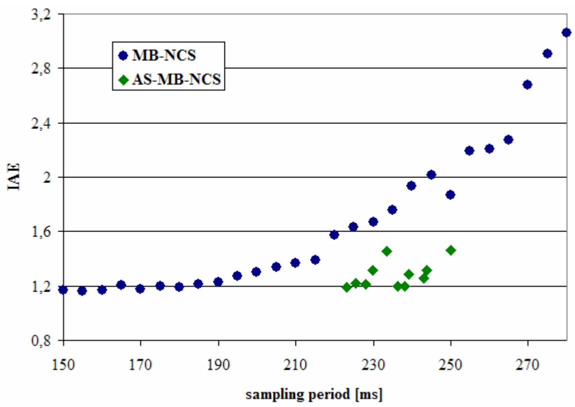

Figure 3: MB-NCS (blue dots) vs. AS-MB-NCS (green diamond shapes) for different adaptation policy parameters

Several pareto-optimal combinations of the design parameters could be found. Pareto optimality means, that an increase in QoC can only be achieved by accepting a higher number of sampling instances. Pareto optimization is a method to do multi-objective optimization, see, e.g., Zitzler et al. (2001). The control designer has to pick one of those combination, depending on the importance of the opposing goals. In this example the following pareto-optimal parameters were chosen: h_{fast} = 160 ms, h_{slow} = 240 ms, $c_1 = c_2 = \begin{pmatrix} 0.05 \\ 0 \end{pmatrix}$

This AS-MB-NCS controller works with an average sampling period h_{avg} of 229 ms. Therefore, it was tested against a MB-NCS controller with a fixed sampling rate of 230 ms. A detail of the result is shown in Figure 4.

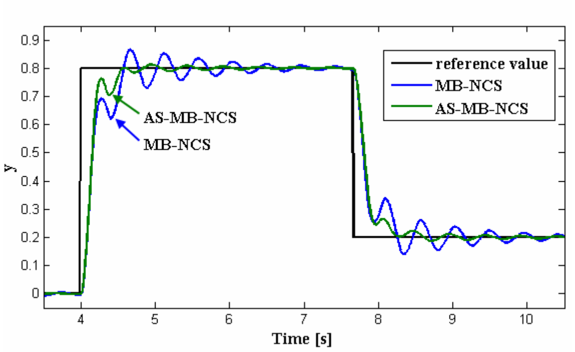

Figure 4: comparison between MB-NCS and AS-MB-NCS (detail)

In Table 1, the IAE and number of packets is compared for AS-MB-NCS and MB-NCS. Not very surprisingly MB-NCS with 160 ms sampling period has the best IAE, but worst number of packets and MB-NCS with 240 ms has the worst IAE, but best number of packets. AS-MB-NCS shows a pretty good IAE, while sending a number of packets which is equivalent to MB-NCS with 230 ms. Compared to MB-NCS with 230 ms, AS-MB-NCS shows a much better IAE, which supports the impression of Figure 4.

In MB-NCS a sampling period of 185 ms would be necessary to achieve the same IAE, resulting in 324 sensor packets, which is an 23% increase in sensor packets, compared to AS-MB-NCS.

Table 1. Comparison of QoC and network usage for AS-MB-NCS and MB-NCS

controller type	MB-NCS				AS-MB-NCS
sampling period	160 ms	185 ms	230 ms	240 ms	229 ms (average)
IAE	1,166	1,212	1,672	1,936	1,213
packets	375	324	260	250	263

In Table 1, the number of packets sent using AS-MB-NCS is 263. In contrast to MB-NCS, the AmI sensor in AS-MB-NCS also receives packets, namely the notifications from the Smart Actuator to change the sampling period. Counting these packets as well, the number of packets rises by 21 to 284. However, these additional packets can be omitted, as the sensor is awake while receiving notifications anyways. In practice, the Smart Sensor sends its notifications at the time a new measurement is expected according to the current sampling period.

5 Conclusion and outlook

In this paper a novel adaptive sampling period control scheme was presented, which is suitable for AmI systems. A stability proof for AS-MB-NCS was presented. An example of an adaptation policy for linear SISO systems was given. A pareto optimization was carried out to determine the design parameters of the proposed adaptation policy. The resulting AS-MB-NCS was tested against MB-NCS in a simulation example. It was shown that AS-MB-NCS can be more effective than MB-NCS. A similar QoC can be achieved with fewer sensor packets. Therefore, sensor energy is being saved.

In the future, the control scheme will be extended to output feedback control systems, as well as for delayed measurements. Both extensions should be easily incorporated as they are already handled in the non-adaptive case in Montestruque and Antsaklis (2003). Whenever a model has a gain error, a constant control error will remain no matter how fast sampling is carried out. Therefore, the control scheme should be extended by an integral action, in order to improve applicability of the proposed control scheme. Also typical network induced effects like packet losses and jitter need to be addressed. The effects of disturbances on the control scheme is another field of interest. Further investigations on adaptation policies will be done, particularly considering the needs in AmI systems and cooperative control.

References

Colandairaj, J.; Irwin, G.W.; Scanlon, W.G. (2007). Wireless networked control systems with qos-based sampling. Control Theory & Applications, IET, Vol.1, No.1, pp. 430-438.

Hespanha, J.P.; Naghshtabrizi, P. and Yonggang Xu (2007). A Survey of Recent Results in

Networked Control Systems. Proceedings of the IEEE, Vol.95, No.1, pp. 138-162.

Litz, L.; Gabel, O. and Solihin, I. (2005). NCS-Controllers for Ambient Intelligence Networks - Control Performance versus Control Effort. IEEE Conference on Decision and Control, 2005 and 2005 European Control Conference.

Lunze, J. and Lehmann, D. (2010). A state-feedback approach to event-based control. Automatica, Vol. 46, Issue 1, pp. 211-215.

Montestruque, L. and Antsaklis, P. (2003). On the model-based control of networked systems. Automatica, Vol. 39, Issue 10, pp. 1837-1843.

Peng, C.; Yue, D.; Gu, Z. and Xia, F. (2009). Sampling period scheduling of networked control systems with multiple-control loops. Mathematics and Computers in Simulation, Vol. 79 , Issue 5, pp. 1502-1511

Song, Y. H.; Park, J. H.; Lee, K. C. and Lee, S. (2009) Performance evaluation of wireless networked control system using time-triggered IEEE 802.15.4. ICCAS-SICE, 2009, pp. 2187-2191.

Zhiwen W.; Ge G.; Yi H. and Dongsong L. (2007). Output Feedback of Model-based Networked Control Systems with Multi-rate Input Sampling. Chinese Control Conference 2007, CCC 2007, pp. 269-272.

Zitzler, E.; Laumanns, M. and Thiele, L. (2001). SPEA2: Improving the Strength Pareto Evolutionary Algorithm. Tech. Rep. 103, Gloriastrasse 35, CH-8092 Zurich, Switzerland.