Synchronization and Sampling in Wireless Adaptive Signal Processing Systems

György Orosz, László Sujbert, Gábor Péceli

Department of Measurement and Information Systems,
Budapest University of Technology and Economics,
{orosz, sujbert, peceli}@mit.bme.hu

Abstract

This paper deals with the synchronization in wireless adaptive signal processing systems. Wireless communication offers high flexibility, however, the distributed structure of wireless systems requires the synchronization of the subsystems. The synchronization becomes particularly important if the signal bandwidth is in the kHz range, and it is inevitable in distributed control systems. The demand on the synchronization is presented through the introduction of a wireless active noise control system. In this system wireless sensors (microphones) receive the signal, and the main signal processing algorithm is implemented on a central unit which produces the signal for the actuators (loudspeakers). In spite of the special application, the system has a general structure, so the results are valid for other adaptive systems. First a PLL like algorithm is described for the synchronization of the sampling in a wireless real-time signal processing system, then a higher-level synchronization is introduced for a distributed Fourier-analyzer based noise control system. The effectiveness of the algorithms is demonstrated by measurement results.

Keywords: Synchronization, Digital signal processing, Distributed signal processing, Wireless sensors, Real-time adaptive signal processing

1 Introduction

Due to the rapid development in the technology, wireless sensor networks (WSN) become more and more wide-spread in every field of our life, and further growth in the number of potential applications of wireless devices can be expected [1, 30]. Well-known fields of the utilization of WSNs are e.g., environmental monitoring, disaster forecast, health care, agriculture, security, military applications, traffic control, production automation [1]. The advantages of wireless technology are e.g., flexibility, scalability, costeffectiveness, easy deployment, installation and maintenance [2]. In certain cases, wireless communication is more reliable then wired communication links since it is not subjected to some typical problems that emerge in wired systems, e.g., cable breaking. Because of these attractive features of WSNs and the permanent decreasing of their price, the deployment of sensor networks is also promising in such applications where their utilization has many open questions yet. One of these fields is the application of WSNs in real-time, closed-loop signal processing and control systems [4, 13, 15, 18, 25, 31].

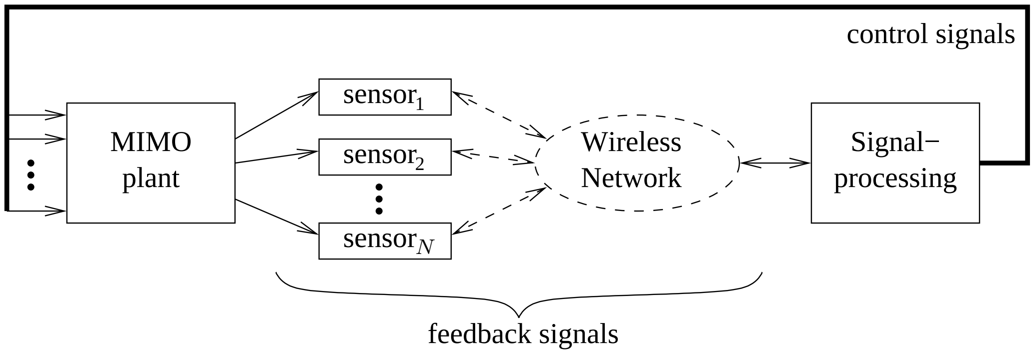

Figure 1: Typical configuration of wireless closed-loop systems

The application of WSNs for signal processing poses plenty of problems, which are not present in traditional wired systems. Perhaps the most unpleasant one is the uncertain amount of delay in the data transmission [25] which is caused primarily by the network protocol and routing. The system should tolerate missing or faulty samples or possibly missing packets containing many samples. Nevertheless, the nodes of the WSN are responsible not only for collecting samples but processing them, i.e. each node has its own processing unit with its own clock generator. Different clock rates result in different sampling frequencies, which results in inconsistent sampling in the network. The latter problem can be solved by the synchronization of the network nodes [27].

The typical configuration of a wireless feedback signal processing and control system is shown in Fig. 1. Nowadays, typical applications of WSNs in the field of process control is generally the control of slow processes (e.g., temperature) or the so-called open-loop control where sensors supply only auxiliary information about the process to be controlled (e.g., the temperature of a rotating machine) that are used for the fine tuning of the control algorithm [18].

This paper focuses on the problem of the synchronization of the network nodes in closed-loop, real-time applications particularly in adaptive signal processing systems. Multi-hop networks and the correction of faulty packets are not considered. The issue of synchronization will be introduced by a wireless active noise control (ANC) system [10,14]. We do not deal with the ANC problem in detail, but we use it to demonstrate the problems that emerge in a real-time, closed-loop application. ANC systems are used for the suppression of acoustic disturbances by means of the phenomenon of the destructive interference. The noise to be suppressed is sensed by microphones, and the so-called anti-noise is radiated by loudspeakers. The anti-noise is generated so that it minimizes the total power of the acoustic noise at the microphones' positions. In our wireless ANC system the noise is sensed by wireless sensors. Since the acoustic systems are generally multiple-input multiple-output (MIMO) structures, the synchronization is inevitable.

A question arises, why ANC is chosen as test application. It has been found that ANC has several benefits which make it particularly suitable for test purposes. Although ANC is closely related to acoustical problems, the results can be generalized for other physical systems, as well. The advantageous features of ANC as test application can be summarized as follows.

An ANC system is easy to install. Sensors and actuators, i.e. microphones and loudspeakers are commercial products so they are easily available. The plant is an acoustic system which is present everywhere and does not require designing and fabricating of electrical, mechanical, chemical, etc. systems. The system is flexible: the structure of the system can be changed by simple geometrical rearrangement, so an algorithm can be tested in different system configurations. Although ANC is closely related to acoustical problems, the results can be generalized for other physical systems, as well.

An ANC system is a MIMO system. It consists of several microphones and loudspeakers, so a real sensor network can be built using wireless microphones. The loudspeakers are connected to the controller by wires since they need more power than the microphones. An ANC system is scalable. The number of microphones and loudspeakers in the ANC applications vary in a wide range, from single input - single output (SISO) systems to bigger ones, in which 1-2 dozens of microphones sense the acoustic field. Complicated algorithms can be tested with simple systems, but other, well-understood procedures can be checked out in a challenging MIMO environment.

The paper is structured as follows. In Section 2, the hardware and software configuration of the wireless active noise control system is presented. Section 3 briefly discusses the ANC problem, and summarizes the challenges of its implementation in a wireless environment. Following the overview of the possible synchronization alternatives, in Section 4, a PLL like synchronization mechanism is introduced that is used for the synchronization within the wireless network. In Section 5, two pilot applications are presented, and special emphasis is put on the synchronization algorithm between the central unit and the sensors.

2 System Description

2.1 Hardware Configuration

The block diagram of the wireless active noise control system can be seen in Fig. 2. The system consists of two main kinds of units.

The computationally critical signal processing algorithms are implemented on a DSP board which is an ADSP EZ-KIT LITE evaluation board [2]. The processor is an ADSP 21364 (SHARC) DSP which has a 32 bit floating point dual arithmetic unit and operates with the clock frequency of 330 MHz. The DSP is connected to an AD1835 codec that has two analog input and eight analog output channels through which the signals can be fed to the loudspeakers. The analog inputs can be used as auxiliary input signals, for example, lots of ANC algorithms use a so-called reference signal that carries information about the noise.

The acoustic signal is sensed by the elements of the wireless sensor network which is composed of Berkeley micaz motes [8]. These motes are intelligent sensors that consist of an ATmega128 [3] eight bit microcontroller with a clock frequency of 7.3728 MHz, a CC2420 2.4 GHz ZigBee compatible radio transceiver and an MTS310 sensor board [8]. The data transfer rate of the transceiver IC is 250 kilobit per second (kbps) including a preamble section, a header and a footer that are handled by hardware. The sensor board also includes a microphone with a variable gain amplifier whose output signal is converted by the 10 bit analog to digital converter (ADC) of the microcontroller.

Most of motes (mote$_1$ ... mote$_N$ in Fig. 2) are responsible for noise sensing. They transmit the noise data towards the DSP. Data from the wireless network are forwarded to the DSP by the gateway mote (mote$_0$ in Fig. 2).

The DSP and the gateway mote are connected to each other via the asynchronous serial port. The data rate of the communication between the two units is 115.2 kbps. The programming of the motes is carried out with an interface board of type MIB510 [8]

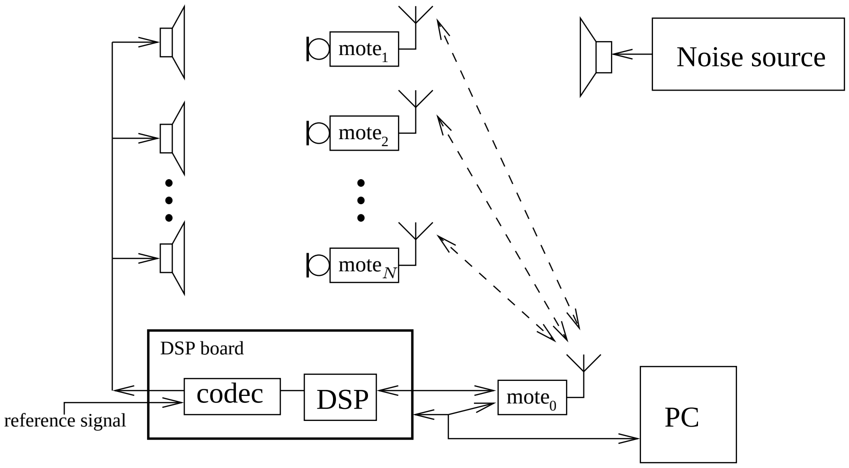

Figure 2: Block diagram of the wireless ANC system

which also serves as a power supply and RS232 line driver for the gateway mote. The PC is used for the processing of data sent by the gateway or the DSP over serial port.

An independent loudspeaker driven by a signal generator is used to generate external sound. It can be used as an artificial noise source for testing the ANC system or as general excitation signal for test purpose.

2.2 Software and Network Configuration

The operation of the DSP and the motes is time triggered. This means that the signal processing on the DSP is performed with the periodicity T_{sDSP}, and the sampling, basic signal processing and communication tasks on the motes are performed with the periodicity T_{sm}. This scheduling scheme is realized using the own timers of the units. The time triggered scheduling ensures the deterministic operation of the system which is important because of the stability of the algorithms that are implemented in the system [15]. The only asynchronous task is the processing of the messages that are sent between the nodes of the system. Since these messages have crucial role in the synchronization, special emphasize must be put on the detection of the arrival time of radio messages during the software development.

The system uses single-hop star topology, i.e. the sensor nodes and the base station communicate directly with each other. The network uses Time Division Multiple Access (TDMA) scheme since the real-time operation of the system requires deterministic network protocol and medium access. It has been demonstrated in [16] that TDMA medium access provides better performance in the wireless closed loop systems than random access MAC protocols even with collision detection and avoidance. One can also find standard protocols for sensor networks that support TDMA MAC, for example in ZigBee the beacon-enabled mode with guaranteed time slots (GTS) can be used for TDMA communication [5, 6]. Furthermore, TDMA ensures the high utilization ratio of the wireless channel.

The network uses UDP like data transmission, i.e. no acknowledge signal is used for the detection of the packet loss. It results in higher data loss ratio then TCP like data transmission (where packet loss detection and correction is used), however, it suits better to the hard real-time behavior of the system. It shows similarities with video and sound streaming in multimedia applications where UDP is also favorable. The following considerations led to the choice that the handling of packet loss is neglected in the network protocol.

- Closed-loop systems require deterministic and as low delay as possible.
- The system requires continuous data flow, and the blocking of the data flow because of the retransmission of lost packets is not advantageous since it causes the uncertainty of the delay.
- The retransmission of the packets requires extra time gaps. These extra time gaps decrease the achievable bandwidth, which is not advantageous since the sensing of the acoustic signal requires high bandwidth.
- Simulations and analytical results show that adaptive closed-loop systems tolerate well the random data loss that is characteristic of wireless communication, for example random-independent (i.e. Bernoulli) or random-bursty (i.e. Gilbert-Elliot) [21] data loss.

3 Active Noise Control

3.1 Introduction to Active Noise Control

The purpose of ANC systems is to suppress low frequency acoustic noise by means of the destructive interference [10, 14]. Effective noise suppression can be achieved for noises of bandwidth about 1-2 kHz. ANC systems can also be regarded as control systems where the plant to be controlled is an acoustic system, the inputs of which are the loudspeakers that radiate the so-called anti-noise, and the outputs are the signals of microphones that sense the acoustic signals. The plant incorporates not only the acoustic system but also the wireless sensors and network delay. They are described altogether by a matrix $\mathbf{A}(z)$ which consists of the transfer functions between each output and each input of the noise control algorithm. $\mathbf{A}(z)$ is often called secondary path in the ANC systems. Since the acoustic environment is time varying, adaptive signal processing algorithms have to be used.

In the ANC algorithms, a kind of inverse model of the matrix $\mathbf{A}(z)$ is applied which is denoted by $\mathbf{W}(z)$. In order to ensure the stability of the system, $\mathbf{W}(z)$ is often chosen as follows [9, 29]:

$$\mathbf{W}(z) = \mathbf{A}(z)^{\#}, \tag{1}$$

where $^{\#}$ denotes the pseudo- (or Moore-Penrose) inverse. The secondary path $\mathbf{A}(z)$ should be identified in advance. The stability criterion of the system is [14, 29]:

$$-\pi/2 < \mathrm{arc}(\lambda_l) < \pi/2, \tag{2}$$

$$\lambda_l = \lambda_l(\mathbf{A}(z)\mathbf{W}(z)); \quad l = 1 \ldots L, \tag{3}$$

where L is the number of inputs. (2) and (3) mean that all eigenvalues λ_l of the term $\mathbf{A}(z)\mathbf{W}(z)$ must have positive real part for each frequencies where the noise is present. For the single channel case (only one microphone and one loudspeaker are used) it means that the phase shift of $\mathbf{A}(z)$ must be known at least with the accuracy of 90°. In practice, it is a crucial problem to ensure the permanency of $\mathbf{A}(z)$.

3.2 Difficulties in the Wireless ANC

The problems that emerge in ANC applications are generally present in closed-loop signal processing systems. ANC systems are generally multiple-input multiple-output (MIMO) systems, which means that even 1-2 dozens of sensors can be used for noise sensing. The high number of sensors is required in order to achieve effective noise suppression in large space. Furthermore, ANC systems are real-time signal processing systems which require relatively fast data flow, so they maximally exploit the resources of the wireless sensor network from the viewpoint of the communication and computation.

The realization of the noise sensing with wireless sensors requires the accurate synchronization of the sensors, since they must provide consistent information for the central controller. Taking into account that the bandwidth of the noise may be some kHz, the accuracy of the synchronization must be at least in some ten μsec range.

ANC systems are extremely sensitive to the change of the transfer function matrix $\mathbf{A}(z)$. Each element of the matrix $\mathbf{A}(z)$ is a high-order transfer function, the phase of which can change rapidly in frequency, thus, in spite of the simplicity of (3), small changes in the transfer function (e.g., frequency shift of a zero) can cause the instability of the algorithm. Such changes often occur in acoustic systems. On the other hand, the presence of the wireless communication increases the uncertainty in the system by the network induced delay. As it will be highlighted in the next sections, the improper synchronization also contributes to the delay, so it increases the risk of the instability.

4 Synchronization

4.1 General Considerations

The synchronization is a crucial task in the system because of the autonomous operation of the subsystems (each mote and the DSP). If the sampling on the motes and the processing of the sampled data on the DSP occurred independently of each other, the delay between the sampling and the processing of the signal would vary at least within one sampling period long interval. The phenomenon of the changing delay is illustrated in Fig. 3. Let's consider the case when the motes are sampling the noise signal, they send these data to the DSP, and the DSP processes the most recent samples received from motes. A sample can arrive at any time instant between the actual and the previous data processing points of the DSP. The delay between the sampling and the processing of data at the time instant t_n is $(T_{tr} + T_{d1})$, and the delay at the time instant t_{n+3} is $(T_{tr} + T_{d2})$. The two delays are different, which is the result of the different sampling and signal processing frequencies. In the figure, T_{tr} denotes the transmission time of one sample, the horizontal axes are time axes, and the short vertical lines denote the signal processing and sampling instants.

A delay ΔT contributes to the transfer function $\mathbf{A}(z)$ of the feedback path by a phase shift $e^{-\jmath\omega\Delta T}$. Since $\mathbf{A}(z)$ includes the delay between the arrival and the processing of data, if the delay changes, $\mathbf{A}(z)$ also changes continuously during the operation so it differs from its identified value. Hence, $\mathbf{W}(z)$ is no more optimal. Moreover, it can occur that $\mathbf{W}(z)$ does not satisfy the stability condition. For example, even one sample interval (T_{sDSP}) change in the delay causes 90° phase shift at a signal frequency of $f = f_{sDSP}/4$, which causes the instability of a SISO system.

The synchronization that is required for the proper operation of the system can be solved with different methods [20]. The units can be synchronized by fixing the relative positions of the time instants of the sampling and signal processing, which requires the

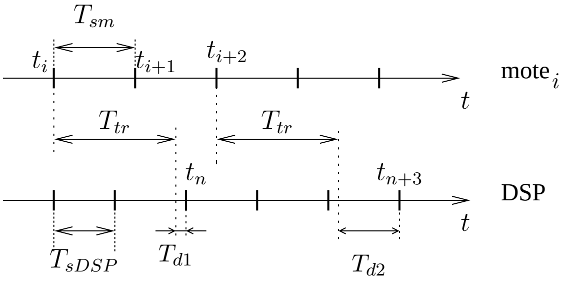

Figure 3: The changing delay without synchronization

tuning of the clock (i.e. the scheduler) of the units. According to Fig. 3, it means that $T_{sm} = T_{sDSP}$ and $T_{d1} = T_{d2} = const.$ should be ensured. It is called physical synchronization, since the events (e.g., sampling) must be physically influenced.

Another approach for the synchronization is the interpolation, where the effect of the unaligned sampling and processing time instants are compensated by computation. By means of the interpolation, the exact value of the signal can be estimated at an arbitrary time instant, so eliminating the changing delay. The interpolation is a very general task emerging in several applications, so it has very wide literature [7, 19, 23, 28]. Standard methods are, e.g., polynomial interpolation (e.g., Lagrange, Hermite polynomials, or least-squares fitting of polynomials, splines), sinc interpolation or interpolation by filtering the signal. The interpolation can be interpreted as the conversion of the sampling frequencies between the sensors and the DSP, so standard methods that are used in this field can also be efficiently used, e.g., realization of integer interpolation using filtering followed by polynomial interpolation [7, 23]. The main requirement in the closed-loop systems is to ensure that the interpolation introduces as low delay as possible while it provides sufficient precision.

The interpolation can be realized either on the motes or on the DSP. In the first case, the DSP acquires the data asynchronously, and the motes calculate the signal value at the request time instants. This means that the computation is distributed in the network but the transmission of request messages causes extra load for the network. In the second case, the motes send the data asynchronously, and the DSP estimates the value of the signal in the processing points according to the previous samples and the time of the arrival of the data. With this method, the interpolation leads to high computational load on the DSP since the data from every mote should be handled individually. The method of the interpolation can be chosen in both cases according to trade-offs between computational complexity and precision. Algorithms that ensure lower distortion of the signal shape generally require more processor capacity. At the present state of the technology, it is not realistic to implement complicated interpolation algorithms on the sensor nodes because of their computational, energy and memory constraints.

4.2 Synchronization of the Sampling

4.2.1 Algorithm Description

In our system the synchronization is divided into two stages: synchronization within the sensor network, and synchronization between the sensor network and the DSP. These two stages are reasonable because of the different configuration, functionality and resources of the motes and the DSP.

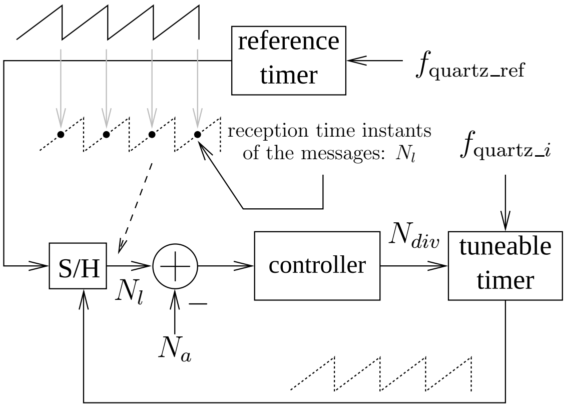

Figure 4: Block diagram of the synchronization algorithm. Sawtooth signal with solid line: reference counter's value. Sawtooth signal with dotted line: the value of the counter of the timer to be synchronized.

In this subsection, the synchronization of the sampling within the sensor network is presented. The synchronization between the sensor network and the DSP depends also on the ANC algorithm, so it will be presented together with the concrete system architectures.

The sampling on the motes is physically synchronized. The physical synchronization is chosen for the motes because it can be solved with a simple synchronization algorithm, which is an important aspect because of the limited resources of the sensor motes. Physical synchronization of the motes has the supplementary advantage that it makes the timing of the network activity easier, since the motes have the same time reference. The synchronization of the motes with interpolation is not preferred because it increases the complexity and has no additional advantage.

In order to ensure the synchronization of the sampling on the motes, a PLL like method was developed. The wide-spread synchronization algorithms that are developed for sensor networks [11, 12, 17, 26, 27] can not be directly used since they are too general methods, and focus on general time synchronization. In our case, however, the main task is the synchronization of the sampling with high precision. Since we give a special method for the synchronization of the sampling, it has smaller resource requirement than the general algorithms.

The synchronization method requires a tuneable timer, since synchronization is achieved by slowing down or speeding up the sampling. The tuneable timer is available on the microcontroller of each mote [3]. The timer is realized with a counter which operates with the clock frequency of the mote ($f_{quartz} = 1/T_{clk}$). After the counter has reached its programmed maximal value (N_{div}), it is cleared, and an interrupt is generated where the sampling of the microphone's signal is performed. The value N_{div} determines the sampling frequency: $f_{sm} = T_{sm}^{-1} = f_{quartz}/N_{div} = 1/(T_{clk}N_{div})$ (where T_{sm} is the sampling interval). The time function of the counter's value is a sawtooth signal, at the falling edge of which the sampling is carried out. The synchronization of the sampling frequency on the motes is achieved by holding the phase difference between the sawtooth signals constant with a PLL like structure that can be seen in Fig. 4. Since the falling edges of the sawtooth signals represent the sampling points, this algorithm ensures the sinchronization of the sampling.

The synchronization is performed in the following way. The algorithm requires a reference mote. The reference mote performs the same tasks as the other sensor motes, but it sends its messages over the radio channel at deterministic time instants with constant periodicity that is determined by its timer. Hence, these data messages can be used as synchronization messages and as time reference. At the reception time instants of these messages, the other motes read the value of their own counter. In Fig. 4, N_l denotes this value. Since the value of the sawtooth signal is proportional to its phase, this sampling and hold (S/H) operation is analogous to the phase detector function. Hence, with the tuning of the sampling frequency (changing N_{div}), the phase difference between the sawtooth signals on the reference and on the other motes (i.e., N_l) can be kept constant.

Let N_a denote the desired value of N_l. The synchronization algorithm is as follows:

- if $N_l < N_a - N_{tol}$, f_{sm} should be increased (i.e., N_{div} should be decreased)
- if $N_l > N_a + N_{tol}$, f_{sm} should be decreased (i.e., N_{div} should be increased)
- if $N_a + N_{tol} \geq N_l \geq N_a - N_{tol}$, f_{sm} should be held on its nominal value (i.e., $N_{div} = N_{div}^{nom}$)

where N_{tol} is a tolerance parameter. N_{tol} is used because of the measurement uncertainty of the reception time of the messages (N_l), due to the jitter in the communication and software delays. $N_{tol} = 0$ would result in the unnecessary tuning of the timer. This is a very simple control algorithm, so it does not require considerable computational resources. It is important, since the algorithm is implemented on an eight bit microcontroller.

A measurement result of the synchronization can be seen in Fig. 5 where the output of the phase detector is observed. As shown by the dashed line in Fig. 5, without synchronization, N_l changes continuously because of the frequency error of the clock generators of the motes. Since the maximal value of the counter is N_{div}^{nom}, the maximal output signal of the phase detector is also N_{div}^{nom} that is in our case 4096. When the motes are synchronized, N_l remains in a ± 10 μsec interval around N_a that was 3000 (solid line). In the experiment N_{tol} was set to 32.

Since the synchronization messages carry information only in their time of arrival, any data messages can be used for this purpose. The only constraint is that they should be sent by the reference mote at predefined time instants. It means that the synchronization of the sampling in this system means no extra network traffic since it is realized with data messages that should be sent anyway. The synchronization does not need to be performed in each sampling period because the clocks of the motes are accurate enough.

In the wireless ANC system every mote is physically synchronized, since in this case the observations of the sensors are consistent. The synchronization of the gateway is required in order to ensure the real-time data forwarding between the network and the DSP (see later sections).

4.2.2 The Effect of Packet Loss

Since wireless communication systems are prone to packet loss, some considerations are given how to the take this effect into account.

It is assumed that if a packet is lost by a node, it uses the nominal sampling period, i.e. the synchronization algorithm is inactive. The detection of the packet loss can be solved using a timeout mechanism. The synchronization packets are sent regularly, so the nodes can notice that a synchronization packet is lost by measuring the time elapsed since the last received packet. If this time exceeds a limit, the synchronization should be inactivated.

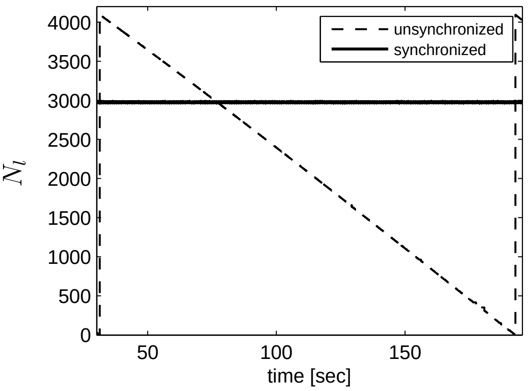

Figure 5: Synchronization measurement results

The algorithm could be further improved by estimating the drift of the clocks—see e.g., [17], and performing the synchronization in the case of packet loss using this calculated drift.

The synchronization should be disabled when a packet is lost because the unnecessary changing of the sampling frequency could cause the overcompensation of the drift which can even increase the error between the clocks. The reason is that the frequency error of the commercial crystal oscillators is in some or at most some ten ppm range, however, if the sampling frequency is changed, it can cause much higher drift compensation. Taking a practical example that is presented in Section 6, the drift between the clocks of two sensor nodes is $\rho = \frac{555 \cdot 10^{-6}\text{sec}}{152.8\text{sec} - 47\text{sec}} = 5.25$ ppm (it is the slope of the sawtooth signal in Fig. 12.(d)). Furthermore, if the synchronization algorithm changes the sampling frequency even to the smallest extent, i.e. the nominal value of the division factor N_{div} is changed by one: $N_{div} = N_{div}^{nom} \pm 1$, it causes a compensating drift approximately $\rho_{comp} = \frac{1}{N_{div}^{nom}}$. Assuming that $N_{div}^{nom} = 4096$ which is used in Section 6, then $\rho_{comp} = \frac{1}{N_{div}^{nom}} = \frac{1}{4096} = 244$ ppm. This is considerably higher then the error of the quartz oscillators. Hence, it is more dangerous to apply a modified sampling frequency than the nominal one if a synchronization packet is lost, since the node to be synchronized is not aware of the error of the synchronization, and it has no information whether a compensation is required or not.

5 Pilot Applications

In this section, two kinds of ANC systems are introduced that have already been implemented. Both of them utilize the resonator based ANC algorithm [29], however, the structures of these systems are quite different. These two different structures show the difficulties and advantages of the deployment of wireless sensors.

The resonator based ANC algorithm can be used for suppressing periodic noises providing better performance for periodic noises than the general methods [29]. The noise control algorithm requires the exact estimation of the fundamental frequency of the noise that is denoted by f_1. The frequency f_1 is measured by an adaptive Fourier-analyzer (AFA) algorithm [22] using the reference signal. The reference signal is directly connected to the DSP board, as it can be seen in Fig. 2. The reference signal can be any signal the frequency of which equals to f_1. The AFA is implemented on the DSP as it requires floating point computations.

The two kinds of ANC systems that will be introduced in this section have the same hardware configuration as shown in Fig. 2, the difference lies in the implementation of the ANC algorithm. The sampling frequency on the motes is $f_{sm} = 1.8$ kHz and $f_{sDSP} = 2$ kHz on the DSP. The different sampling frequencies are constrained by the hardware and by the bandwidth of the wireless network.

5.1 ANC with Simple Data Transmission Network

In this system an obvious method of the signal observation is used, which means that the motes perform the sampling of the noise, and they transmit the samples to the base station over radio. The base station collects the data from the sensors and forwards one sample from each mote to the DSP at each sampling instant. Since every mote (including also the gateway) is physically synchronized, they provide data with the same average sampling frequency, and the relative positions of the sampling instants on the different motes are fixed. Furthermore, since the gateway is also synchronized, the time difference between the reception and the forwarding of the sensors' data is constant, so there is no changing delay in the data path. Hence, the whole network can be regarded as a multiple channel compact data acquisition system, and the operation of the sensor network is transparent from the viewpoint of the DSP.

The network operation is time division multiplexed (TDM): the motes transmit their data periodically, in predefined order, in their own timeslots. This kind of network protocol is proven to be a suitable solution [15] as the probability of the message collision, data loss and fluctuation in the data transmission time is less than in the case of random access protocols [15]. The sensors send data in 25 samples size packets, and the gateway performs the serialization of the data.

The fitting of the sampling frequency of the motes to that of the DSP is achieved by linear interpolation. Linear interpolation is used since it provided sufficient performance in practice, and higher order interpolation would introduce more delay, which is not advantageous in a closed-loop system.

The interpolation algorithm is explained in Fig. 6, where $f(t)$ stands for the signal to be interpolated and the dots symbolize its sampled values. The samples are supplied by the motes and the interpolation is carried out on the DSP. The DSP estimates the signal value at the signal processing point t_n as follows:

$$\hat{f}(t_n) = f(t_{i-1})\frac{T_{sm} - T_d}{T_{sm}} + f(t_i)\frac{T_d}{T_{sm}}. \tag{4}$$

In order to ensure the causality, one sample delay should be introduced since $f(t_i)$ is not known in t_n. With this method, the effect of the continuously changing delay, T_d, is alleviated (see T_{d1} and T_{d2} in Fig. 3), since the sampling instant of the mote is virtually replaced to t_n.

5.2 ANC with Distributed Signal Processing

The bottleneck of the system introduced in the previous subsection is the bandwidth of the wireless network which limits the number of motes at a given sampling frequency. The sampling frequency is determined by the bandwidth of the acoustic signals. To overcome this problem, the signal processing capability of the intelligent sensor nodes was exploited.

In order to increase the effective throughput of the network, a transformed domain data transfer was developed. This kind of signal transmission can be applied in the case of

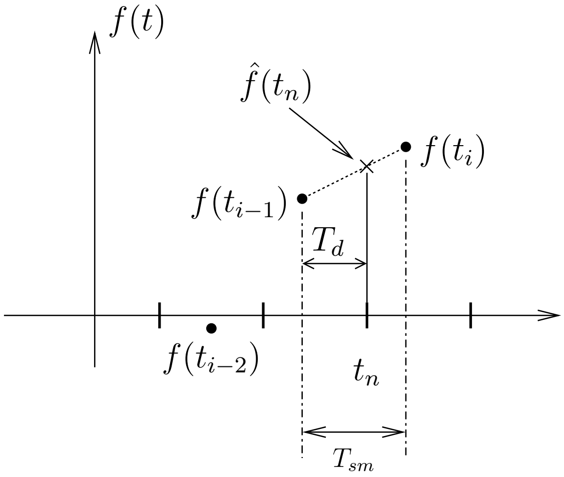

Figure 6: Synchronization with linear interpolation

periodic noise. In this system, the sensors realize a distributed Fourier-analyzer algorithm, and they send only the Fourier coefficients of the periodic signals to the DSP. The Fourier coefficients are computed by the motes with an observer based Fourier-analyzer (FA) structure [24]. These coefficients change slower than the signal, so a lower transmission rate can be achieved. Hence, the small communication bandwidth does not mean a hard limit for the number of the sensor motes. This allows the deployment of more motes without decreasing the sampling frequency. The number of motes is limited by the properties of the noise and the performance requirements (e.g., settling time).

The assumption that the Fourier coefficients change slowly is true if the parameters of the observed periodic signal y_n change slowly or rarely, i.e. at least the short term stationarity of the signal is required. Formally, this means that if y_n is composed of sinusoidal components, i.e. $y_n = \sum_i a_i(n) \cos\big(\omega_i(n)n + \phi_i(n)\big)$, then the amplitude, phase and frequency of the ith component do not change considerably between the transmission time instants, or they change rarely.

This condition is generally true for acoustic signals that are generated by mechanical equipments (e.g., rotating machines) since the inertia of these systems and the time constants of the mechanical movements do not allow the rapid or frequent change of the parameters of the generated periodic signal (i.e. the noise of the machine). For example, let's assume a 0.1 sec time segment. Due to the physical circumstances, the angular frequency of a machine does not change considerably within this time segment. If a sampling frequency of several kHz is applied, it means that a sensor collects and transmits several hundreds of samples during this time. However, since the frequency does not change considerably in this interval, it is sufficient to send the signal parameters only once.

Fig. 7 depicts the structure of a resonator based distributed ANC system. In the original configuration of the algorithm [29], the FA and the ANC blocks are implemented on the same hardware, so in this distributed structure a new synchronization problem emerges: the consistency of the basis functions of the Fourier-decomposition has to be ensured on each mote and on the DSP, because the phase of the Fourier coefficients can be interpreted by the ANC algorithm only by using a common reference in the system. This reference is the basis function set of the Fourier-decomposition. The n^{th} basis function is

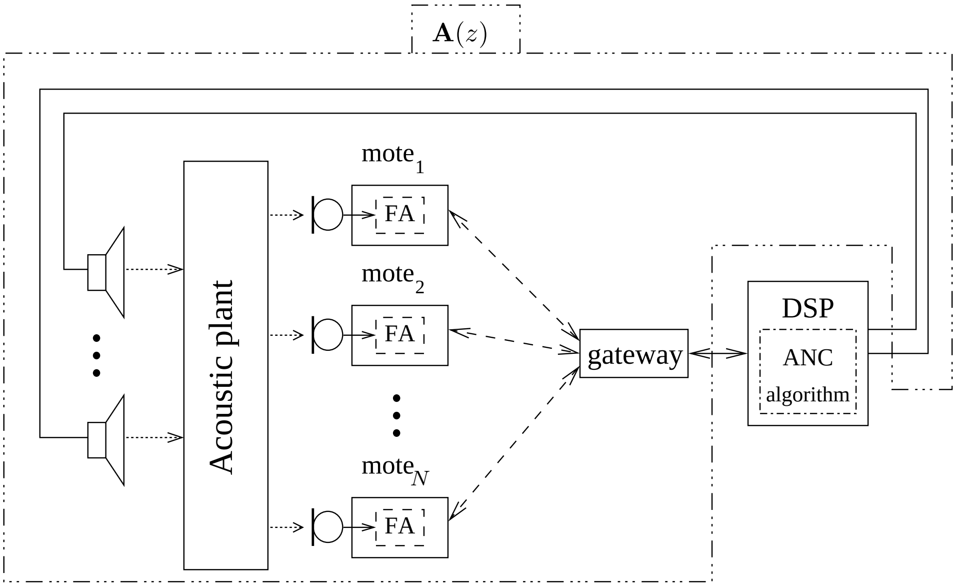

Figure 7: Distributed Fourier-analyzer based ANC system

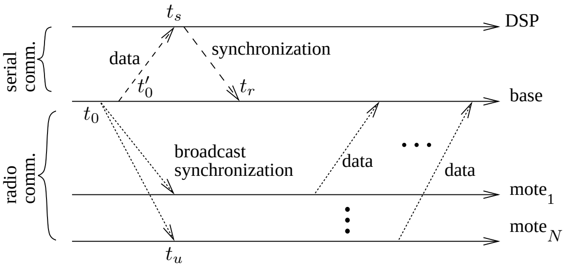

Figure 8: Network operation in the distributed ANC system

defined as follows:

$$c_n(t) = e^{\jmath 2\pi[n\varphi_1(t)]}, \quad \varphi_1(t) = \sum_{\tau=0}^{t} f_1(\tau)\Delta\tau, \tag{5}$$

where φ_1 and f_1 are the phase and the frequency of the fundamental harmonic components, respectively. The fundamental frequency of the basis functions equals to the fundamental frequency of the noise.

The operation of the system is the following. The activity of the wireless network is periodic, each period consists of two phases, as shown in Fig. 8. First, the base station sends a synchronization message to the motes which is used for the synchronization of both the sampling frequency and the basis functions. After this message, each mote gets the right in a predefined order to transmit the Fourier coefficients to the base station. After the synchronization message, the base station begins to forward the data collected from the network to the DSP.

The synchronization of the basis functions is carried out in two phases. The two stages are the synchronization between the base station and the sensors, and the synchronization between the DSP and the base station. Thus the sensors are synchronized to the DSP indirectly, through the base station.

The timing diagram of the synchronization of the basis functions between the base

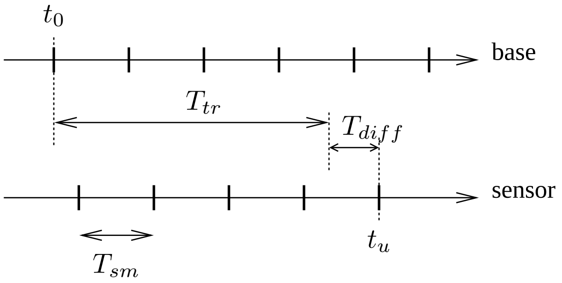

Figure 9: Synchronization of the basis functions (sensors to base station)

station and the sensors is depicted in Fig. 9. The synchronization is carried out by means of the synchronization message of the base station that is sent at the time instant t_0 (see Fig. 8), and it includes $f_1(t_0)$ and $\varphi_1(t_0)$ that are the actual value of the frequency and the phase of the fundamental harmonic basis function at the time instant of the transmission of the message, respectively. Let's denote the transmission instant of this message by t_0 and the time instant of the utilization of the information sent in the message by t_u, so the delay between these two events is $(t_u - t_0)$. Since these operations are carried out at sampling instants, and the sampling on the motes is physically synchronized (i.e. $T_{diff} = const$), so the delay $(t_u - t_0)$ is constant. Hence, the delay $(t_u - t_0)$ can be regarded as the part of $\mathbf{A}(z)$.

The phase of the fundamental harmonic basis function can be estimated at an arbitrary time t in the following way:

$$\varphi_1(t) = \varphi_1(t_u) + f_1(t_u)(t - t_u). \tag{6}$$

The phase of the n^{th} harmonic basis function is $\varphi_n(t) = n\varphi_1(t)$, because the initial value of the phases are zero and the frequencies are in harmonic relationship, so the synchronization of only the first basis function is sufficient. Since the frequency of the periodic acoustic disturbance to be suppressed (f_1) changes generally slowly, it can be regarded constant between two synchronization messages.

The synchronization between the DSP and the base station begins with the initiation message of the gateway at t_0' as it can be seen in Fig. 8 and Fig. 10. At the reception instant of this message (t_s), the DSP calculates the actual value of the phase of the fundamental basis function as follows:

$$\varphi_1(t_s) = \varphi_1(t_p) + f_1(t_p)(t_s - t_p), \tag{7}$$

where $\varphi_1(t_p)$ and $f_1(t_p)$ denote the values of the phase and the frequency in the previous processing point, respectively. The time diagram can be seen in Fig. 10. $\varphi_1(t_s)$ and $f_1(t_s)$ are then sent to the gateway which updates its old data with these new ones. This procedure is important because the gateway and the DSP run asynchronously, so the DSP has to estimate the value of the phase whenever the gateway initiates the synchronization. The transmission of $\varphi_1(t_p)$ would introduce a changing delay $(t_s - t_p)$ into $\mathbf{A}(z)$, thus $\mathbf{W}(z)$ could not be determined correctly. This $(t_s - t_p)$ delay is eliminated by the last term of (7). Constant delays of the data transmission (T_{tr1} and T_{tr2}) cause no trouble since they are represented in $\mathbf{A}(z)$.

In the simple data collection system, the maximal applicable number of motes was three, while in the distributed ANC system the experimental system consisted of eight motes at the same sampling frequency. This proves the signal compression capability of

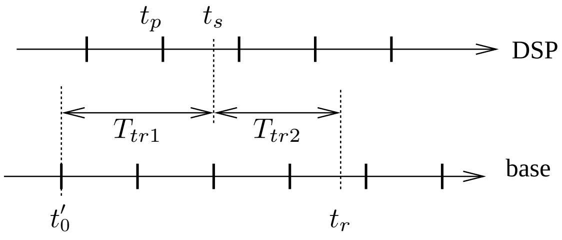

Figure 10: Synchronization of the basis functions (base station to DSP)

the latter, distributed signal processing based method. In this case, the number of sensors could be extended up to some dozens. The size of the network is limited by the properties of the noise to be suppressed (e.g., the rate of the change of the noise parameters) and by the resources of the DSP. However, the computational burden means marginal restriction since in the case of eight motes and four loudspeakers only about five percent of the available time was used for computation of the ANC algorithm on the DSP. On the other hand, the storage of the transfer function matrix $\mathbf{A}(z)$ has large memory demand.

The application of the distributed Fourier-analyzer structure that is presented in this subsection is not limited to ANC systems, but it can be used in any applications where the synchronous analysis of periodic signals is important (e.g., power systems).

6 Results

In this section, some results are presented which prove that the synchronization is inevitable in a wireless, closed-loop application, and demonstrate the proper operation of the synchronization algorithms that have been introduced in this paper.

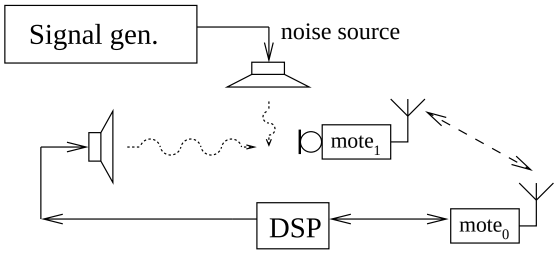

Figure 11: Measurement arrangement

The tests were performed both for the simple data collecting and the distributed ANC systems. The measurements were carried out on a single-input single-output (SISO) system—i.e., one loudspeaker and one microphone were used. The sampling frequency was set to $f_{sm} = 1800$ Hz. The block diagram of the physical arrangement can be seen in Fig. 11. A signal generator was used as a noise source which was connected to a loudspeaker. The noise was a sinusoidal signal with the frequency of 115 Hz. The measurements were performed in a laboratory room, the distance between the sensor mote and the loudspeaker was about 0.5 m. In order to examine the performance of the system, the noise level was measured by an external microphone that was situated about 1 cm from the noise sensing microphone of the sensor mote, and the signals were recorded by a PC sound card. The transfer function $\mathbf{A}(z)$ was measured under normal operating conditions, when the synchronization worked properly.

The basic concept of the measurements is that the noise suppression is to be examined when the synchronization is turned on and off. It is expected that the system without synchronization becomes unstable in some seconds as the clock frequencies differ from each other on the motes. When the synchronization is on, the system remains stable during the whole observation period.

6.1 The Simple Data Collecting ANC System

The synchronization algorithm of this system is described in Section 4.2. By turning off the synchronization in the wireless network, the sampling of the acoustic signal on the sensor and the data forwarding on the base station slide away from each other, which results in the continuous changing of the delay in the feedback path. This phenomenon is explained in Fig. 3.

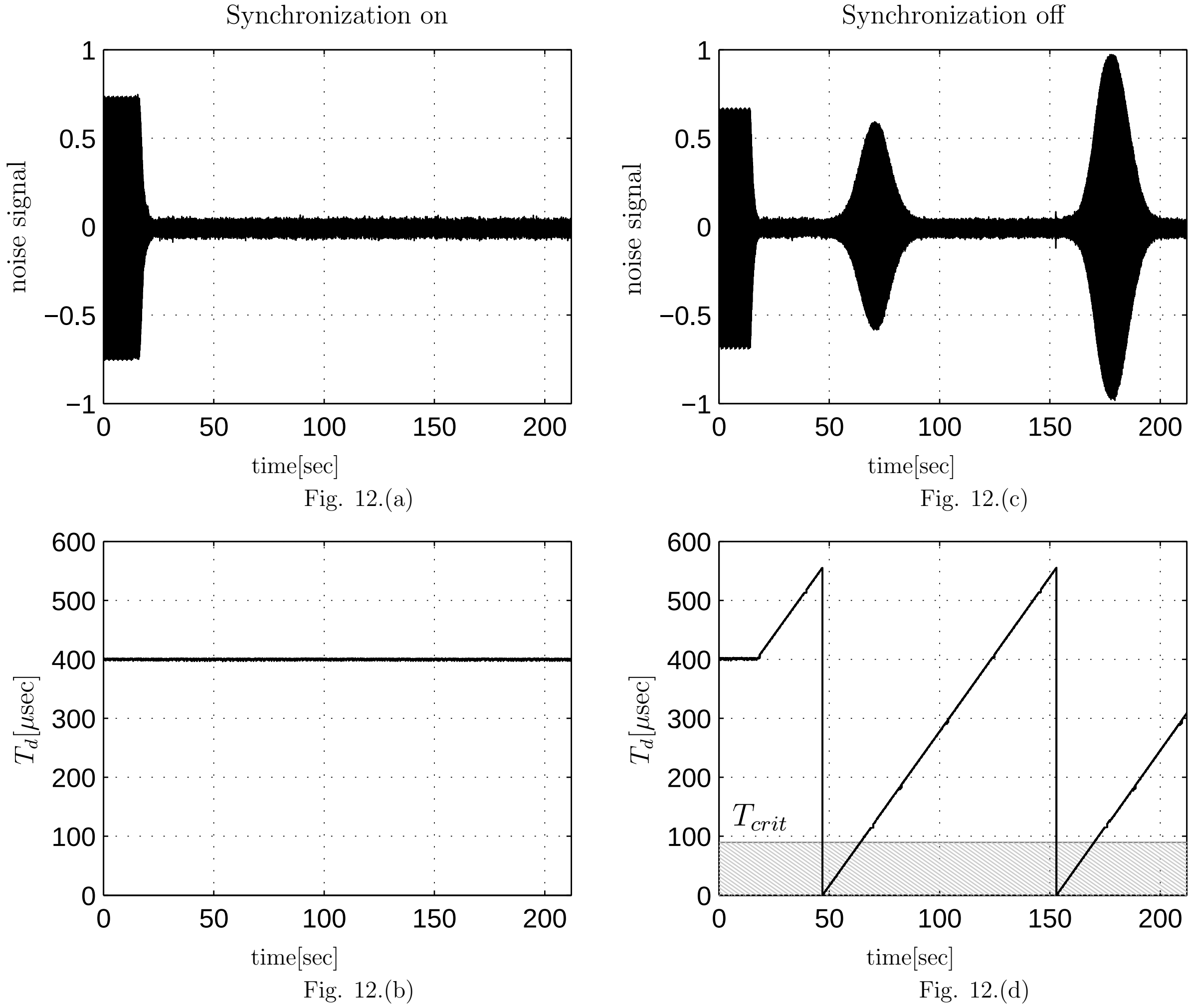

Fig. 12.(a)

Fig. 12.(c)

Fig. 12.(b)

Fig. 12.(d)

Figure 12: Measurement results in the simple data collecting ANC system. Fig (a) and (c): noise signal measured by the external microphone when synchronization is on and off, respectively. Fig (b) and (d): delay between the motes when synchronization is on and off, respectively.

The measurement results are presented in Fig. 12. Fig. 12.(a) and (b) belong to the synchronized state, while Fig. 12.(c) and (d) belong to the unsynchronized state. In

Fig. 12.(a) and (c), the time functions of the noise can be seen during the operation of the noise control algorithm. Fig. 12.(b) and (d) contain the time functions of the delay, T_d, between the events when the base station receives the sensor's sample and forwards it to the DSP. Note that the time scale in each column is the same, therefore the change in T_d and the working of the ANC system can easily be connected. T_d is denoted by T_{di} in Fig. 3, and it causes an extra phase shift in the transfer function of the feedback path. Theoretically, T_d is the only changing delay which can influence the stability of the system. T_d can change in the interval of $[0 \dots 555]$ μsec, since the new samples of the sensor are available with the periodicity $T_{sm} = \frac{1}{1800\text{Hz}} = 555$ μsec, and the gateway forwards the most recent sample.

In Fig. 12.(a) and (b) the time functions can be seen that were recorded when the synchronization worked properly. The delay T_d stayed in the neighborhood of the nominal value $T_{d\_nom} = 400$ μsec. Since this is included in the transfer function $\mathbf{A}(z)$, the system is stable. The ANC algorithm was started at the time instant of 15 sec, and it reduced the noise level by 26 dB. The algorithm stayed in stable state during the whole observation time.

The result is completely different when no synchronization was applied between the sensor and the gateway as Fig. 12.(c) and (d) show. In this measurement, the synchronization was active in the first 20 sec, then the synchronization was turned off. The noise control algorithm was started at the time instant of 15 sec. Since the synchronization worked properly, the noise control algorithm achieved the usual suppression. After the synchronization had been turned off, the delay T_d began to change, which led to instability.

The extra delay that surely causes the instability of the system can be calculated according to (3). The system is the most sensitive at the highest operating frequency where the delay causes the highest phase shift. Since the stability region is at most $\Delta\phi = 90°$, and the highest operating frequency is 805 Hz=115 Hz·7 (the 7th harmonic component), the extra delay which surely causes instability is $\Delta T_{crit} = \frac{\Delta\phi}{360° \cdot 805 \text{ Hz}} = 310$ μsec. Note that the crucial delay in practice is a bit less than ΔT_{crit}. In Fig. 12.(d), the grey area indicates the theoretical instability region where the difference from the nominal delay (T_{d_nom}) exceeds the critical value (ΔT_{crit}). The border of this region is: $T_{crit} = T_{d\_nom} - \Delta T_{crit} = 90$ μsec. Indeed, one can see in Fig. 12.(c) and (d) that the system became unstable after the delay had approached the critical region, and the system recovered from the unstable state after the delay had left the critical region. Since the delay changes continuously, the stable and unstable periods are repeated.

As one can see in Fig. 12, the system remained stable over $T_{us} \approx 28$ sec after the synchronization had been turned off. This measurement also characterizes the synchronization algorithm from the viewpoint of the data loss. The reason is that, as recommended in Subsection 4.2.2, the synchronization should be switched off by a node if a packet is lost. Hence, the time T_{us} during which the system remains stable with disabled synchronization equals the time interval during which the consecutive loss of the synchronization packets is allowed without losing the stability of the system. In this particular example, several hundred synchronization packets were sent in the network during the time when the synchronization was intentionally disabled and the system became finally unstable, so it has very small probability that in a practical case such an extreme data loss scenario occurs which causes instability.

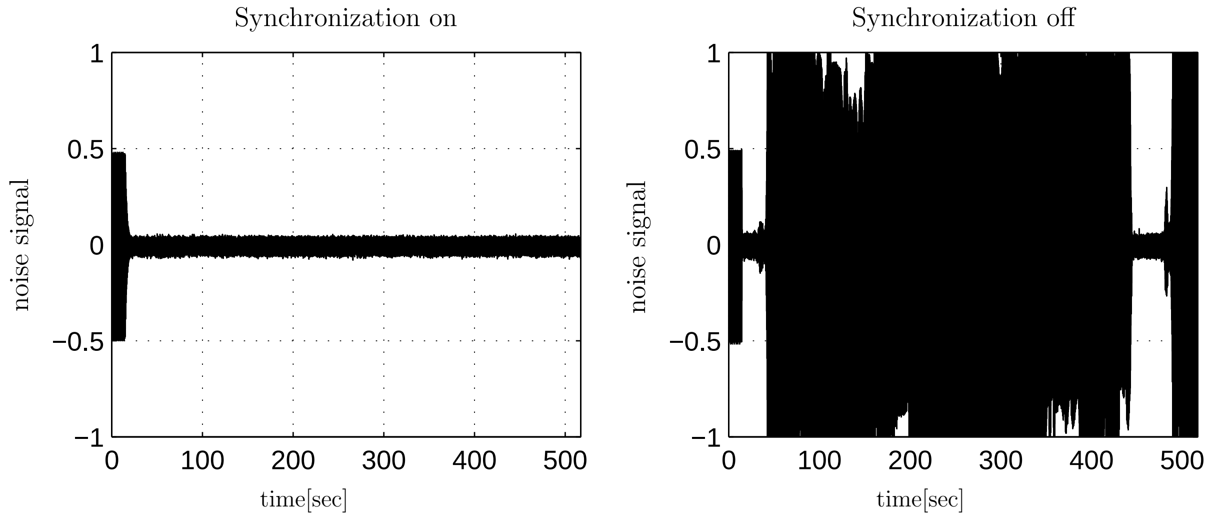

Fig. 13.(a) Fig. 13.(b)

Figure 13: Measurement results in the distributed ANC system.
Fig (a): synchronization is on, Fig (b): synchronization is off.

6.2 The Distributed ANC System

The synchronization algorithm of this system is described in Section 5.2. The effect of the unsynchronized operation was tested by turning off the synchronization of the basis functions between the DSP and the base station. Since the sensor gets the synchronization information from the base station, it means that the sensor mote is not synchronized, as well. The measurement results are found in Fig. 13. In this case no internal states are available, only the time functions of the noise are plotted.

The measurement result that belongs to the synchronized system can be seen in Fig. 13.(a). The noise control algorithm was started at 15 sec, and it was stable during the whole observation time. The noise suppression was 24 dB.

Fig. 13.(b) shows the measurement result of the unsynchronized ANC system. The noise control algorithm was started at the time instant of 15 sec, and the synchronization was active in the first 20 sec. The system became unstable at about 40 sec which caused increasing noise level. The reason of the instability is that after turning off the synchronization, the phases of the basis functions begin to slide away from each other on the sensor and on the DSP. The difference between the basis functions results in a phase error in the measurement of the Fourier coefficients of the noise that can be interpreted as a phase error in the transfer function. If this phase error exceeds the stability limit ($\Delta\phi = \pm 90°$), the system becomes unstable. The ANC system produced larger sound during the unstable state than in the normal operating region, so the noise sensing microphone was saturated. The system became stable again in the interval [445 sec ... 490 sec]. This temporary stability interval appears every time the phase error between the basis functions approaches $k \cdot 2\pi$, since the basis functions are periodic with period 2π. Similarly to the simple data collecting system, this phenomenon causes that the stable and unstable intervals are repeated after each other.

These measurements prove the long term stability of the operation of the synchronization algorithm, and it can be concluded that the unsynchronized distributed signal processing causes the improper and unpredictable operation of the system.

7 Conclusions

In this paper, the issue of the synchronization and sampling in the wireless, closed-loop signal processing systems was introduced. The sampling, i.e., the signal observation is one of the most basic tasks in a signal processing system, so it requires special attention. While in the centralized signal processing systems the simultaneous sampling of the observed signals can be solved easily, in a wireless system the data are collected in a distributed manner by independent sensors. This distributed structure of the wireless systems requires accurate synchronization.

The synchronization was introduced within the frames of two variants of a wireless active noise control system. Although ANC systems are acoustic systems, the results can be generalized for any adaptive signal processing systems. A PLL like method was introduced which ensures the synchronous sampling of the acoustic noise with the continuous tuning of the sampling frequency of the motes. The synchronization between the sensors and the central controller was solved by interpolation. A distributed wireless ANC system was also introduced where the sensors perform the Fourier-decomposition of spatially separated signals in order to achieve data compression. In this case, the synchronization of the basis functions of the Fourier-decomposition has to be solved in order to ensure a global reference in the system.

Both synchronization methods were tested with experimental measurements which prove the necessity of the proposed synchronization, and demonstrate the proper operation of the synchronization algorithms. Further research aims at the development of closed-loop multi-hop wireless signal processing systems.

Acknowledgment

This work is connected to the scientific program of the "Development of quality-oriented and cooperative R+D+I strategy and functional model at BME" project. This project is supported by the New Hungary Development Plan (Project ID:
TÁMOP-4.2.1/B-09/1/KMR-2010-0002).

References

[1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci. Wireless sensor networks: a survey. *Comput. Netw.*, 38(4):393–422, 2002.

[2] Analog Devices. *EZ-KIT Lite for the ADSP-2136x SHARC® Processor Family.* Analog Devices Inc., `www.analog.com`, 2005.

[3] Atmel. *ATmega128A® datasheet.* Atmel co., `www.atmel.com`, 2002.

[4] J. Baillieul and P. J. Antsaklis. Control and communication challenges in networked real-time systems. *Proceedings of the IEEE*, 95(1):9–28, Jan. 2007.

[5] Najet Boughanmi, Ye-Qiong Song, and Eric Rondeau. Wireless networked control system using IEEE 802.15.4 with GTS. In *2nd Junior Researcher Workshop on Real-Time Computing, JRWRTC 2008*, 16-17 Oct. 2008.

[6] Dong-Hyuk Choi and Dong-Sung Kim. Wireless fieldbus for networked control systems using LR-WPAN. *International Journal of Control, Automation and Systems*, 6(1):119–125, Feb. 2008.

[7] R. E. Crochiere and L. R. Rabiner. *Multirate Digital Signal Processing.* Englewood Cliffs, NJ: Prentice-Hall, 1983.

[8] Crossbow. *Crossbow Products Overview.* Crossbow Technology Inc., `www.xbow.com`, 2005.

[9] S. J. Elliott, C. C. Boucher, and P. A. Nelson. The behavior of a multiple channel active control system. *IEEE Trans. on Signal Processing*, 40(5):1041–1052, May 1992.

[10] S. J. Elliott and P. A. Nelson. Active noise control. *IEEE Signal Processing Magazine*, 10(4):12–35, Oct. 1993.

[11] J. Elson, L. Girod, and D. Estrin. Fine-grained network time synchronization using reference broadcasts. In *OSDI '02: Proceedings of the 5th Symposium on Operating Systems Design and Implementation*, pages 147–163, Boston, Massachusetts, Dec. 2002.

[12] S. Ganeriwal, R. Kumar, and M. B. Srivastava. Timing-sync protocol for sensor networks. In *SenSys '03: Proceedings of the 1st International Conference on Embedded Networked Sensor Systems*, pages 138–149, Los Angeles, California, USA, Nov. 5-7, 2003.

[13] D. Hristu-Varsakelis and W. S. Levine. *Handbook of Networked and Embedded Control Systems.* Birkhäuser Boston, 2005.

[14] S. M. Kuo and D. R. Morgan. Active noise control: A tutorial review. *Proceedings of the IEEE*, 87(6):943–973, June 1999.

[15] X. Liu and A. Goldsmith. Wireless network design for distributed control. In *2004. CDC. 43rd IEEE Conference on Decision and Control*, volume 3, pages 2823–2829, Atlantis, Paradise Island, Bahamas, 14-17 Dec. 2004.

[16] X. Liu and A. Goldsmith. Wireless medium access control in networked control systems. In *Proceedings of American Control Conference 2004.*, volume 4, pages 3605–3610, Atlantis, Paradise Island, Bahamas, 30 June-2 July 2004.

[17] M. Maróti, B. Kusy, Gy. Simon, and Á. Lédeczi. The flooding time synchronization protocol. In *SenSys '04: Proceedings of the 2nd International Conference on Embedded Networked Sensor Systems*, pages 39–49, Baltimore, MD, USA, Nov. 3-5, 2004.

[18] M. Mathiesen, G. Thonet, and N. Aakvaag. Wireless ad-hoc networks for industrial automation: Current trends and future prospects. In *Proceedings of the IFAC World Congress*, Prague, Czech Republic, July 4-8, 2005.

[19] Erik Meijering. A chronology of interpolation: From ancient astronomy to modern signal and image processing. *Proceedings of the IEEE*, 90(3):319–342, March 2002.

[20] K. Molnár, L. Sujbert, and G. Péceli. Synchronization of sampling in distributed signal processing systems. In *Int. Symp. on Intelligent Signal Processing, WISP 2003.*, pages 21–26, Budapest, Hungary, Sept. 2003.

[21] M. Mushkin and I. Bar-David. Capacity and coding for the Gilbert-Elliot channels. *IEEE Transactions on Information Theory*, 35(6):1277–1290, 1989.

[22] F. Nagy. Measurement of signal parameters using nonlinear observers. *IEEE Trans. on Instrumentation and Measurement*, IM-41(1):152–155, Feb. 1992.

[23] Olli Niemitalo. Polynomial interpolators for high-quality resampling of oversampled audio, Aug 2001. `http://www.student.oulu.fi/~oniemita/DSP/INDEX.HTM`.

[24] G. Péceli. A common structure for recursive discrete transforms. *IEEE Trans. Circuits Syst.*, CAS-33(10):1035–1036, Oct. 1986.

[25] N. J. Ploplys, P. A. Kawka, and A. G. Alleyne. Closed-loop control over wireless networks. *IEEE Control Systems Magazine*, 24(3):58–71, Jun. 2004.

[26] K. Römer. Time synchronization in ad hoc networks. In *MobiHoc '01: Proceedings of the 2nd ACM International Symposium on Mobile ad hoc Networking & Computing*, pages 173–182, Long Beach, CA, USA, Oct. 2001.

[27] K. Römer, P. Blum, and L. Meier. Time synchronization and calibration in wireless sensor networks. In Ivan Stojmenović, editor, *Handbook of Sensor Networks: Algorithms and Architectures*. Wiley, 2005.

[28] R. W. Schafer and L. R. Rabiner. A digital signal processing approach to interpolation. *Proceedings of the IEEE*, 61(6):692–702, June 1973.

[29] L. Sujbert and G. Péceli. Periodic noise cancellation using resonator based controller. In *Proceedings of the Active '97, The International EAA Symposium on Active Control of Sound and Vibration*, pages 905–916, Budapest, Hungary, Aug. 1997.

[30] M. Tubaishat and S. Madria. Sensor networks: an overview. *IEEE Potentials*, 22(2):20–23, 2003.

[31] S. Zampieri. Trends in networked control systems. In *Proceedings of the 17th World Congress of IFAC*, pages 2886–2894, Seoul, Korea, July 6-11, 2008.