Metrol. Meas. Syst., Vol. XXII (2015), No. 2, pp. 275–288.

PAN POLSKA AKADEMIA NAUK

METROLOGY AND MEASUREMENT SYSTEMS

Index 330930, ISSN 0860-8229

www.metrology.pg.gda.pl

M&MS

SIMULTANEITY ANALYSIS IN A WIRELESS SENSOR NETWORK

Miodrag Malović[1)], Ljiljana Brajović[2)], Zoran Mišković[2)], Tomislav Šekara[3)]

1) University of Belgrade, Faculty of Technology and Metallurgy, Karnegijeva 4, 11000 Beograd, Serbia (✉ office@malovic.in.rs, +381 11 321 8585)

2) University of Belgrade, Faculty of Civil Engineering, Kralja Aleksandra 73, 11000 Beograd, Serbia (brajovic@grf.rs, mzoran@imk.grf.bg.ac.rs)

3) University of Belgrade, Faculty of Electrical Engineering, Kralja Aleksandra 73, 11000 Beograd, Serbia (tomi@etf.rs)

Abstract

An original wireless sensor network for vibration measurements was designed. Its primary purpose is modal analysis of vibrations of large structures. A number of experiments have been performed to evaluate the system, with special emphasis on the influence of different effects on simultaneity of data acquired from remote nodes, which is essential for modal analysis. One of the issues is that quartz crystal oscillators, which provide time reading on the devices, are optimized for use in the room temperature and exhibit significant frequency variations if operated outside the 20–30°C range. Although much research was performed to optimize algorithms of synchronization in wireless networks, the subject of temperature fluctuations was not investigated and discussed in proportion to its significance. This paper describes methods used to evaluate data simultaneity and some algorithms suitable for its improvement in small to intermediate size ad-hoc wireless sensor networks exposed to varying temperatures often present in on-site civil engineering measurements.

Keywords: time, frequency, simultaneity, synchronization, temperature, wireless sensor networks

1. Introduction

Clock synchronization is a well-known problem in computer and engineering science. Distributed systems cannot be physically attached to a global time source, so differences in local clocks inevitably occur as the consequence of hardware imperfections. A variety of techniques have been designed and used in the past decades to limit the growth of these differences over time and provide a certain level of dependability of clock readings.

Some algorithms perform well in office networks with moderate temperature fluctuations. Others are optimized for low power, important for most wireless networks. Sensor networks in civil engineering are often exposed to harsh and varying atmospheric conditions. Their goal is usually structural health monitoring (SHM), *i.e.*, damage detection and estimation of usability of large structures [1]. SHM includes real-time measurements using sensor arrays, signal processing to extract appropriate features from the abundant raw data, and mathematical analysis to estimate the condition of observed structure. Modal analysis of vibrations is a method often applied in this area. To perform modal analysis, data are acquired from different points, and amplitudes and phases of signal harmonics are compared. Good data synchronization is therefore required on site, where some sensors are commonly placed in a shade (or even in air-conditioned rooms) while others are exposed to direct sunrays and enclosed in plastic housings with temperatures soaring well above 50°C. Not enough research was performed to examine an influence of varying temperatures often present in civil engineering measurements and to establish best methods to rectify the situation.

A wireless sensor network (WSN) for vibration measurements on capital objects (such as

Article history: received on Jan. 30, 2015; accepted on Mar. 25, 2015; available online on May 29, 2015; DOI: 10.1515/mms-2015-0022.

bridges, dams and towers) was developed at the Faculty of Civil engineering in Belgrade [2].

An original embedded real time operating system (RTOS) has been implemented on a 6 MHz 8-bit microcontroller from the 8051 family. Despite the predictions that 8-bit microcontrollers would die out, reputable manufacturers still develop 8051 derivatives with new features, and there is a vast base of knowledge and freeware libraries, so they account for nearly half of the world's market today [3].

2. Clock synchronization

2.1. Main sources of synchronization errors

Reading of a network node clock, based on a quartz crystal oscillator, is expressed by:

$$t_{\mathrm{n}} = t_0 + \frac{N}{f}, \tag{1}$$

where t_0 is the time offset (assumed initial moment of the counting), N is the number of counter (timer) ticks, and f is the counter frequency.

Clock frequency stability is limited by many physical factors. Even in the case when frequencies are perfectly calibrated in one moment, drifts have to occur and an error accumulates over time (which is referred to as a skew), necessitating the exchange of new messages to resynchronize the nodes (establish new t_0 and reset N). These messages have transmission time that is not sufficiently determined, especially in the case when radio communication is used between wireless network nodes. Hence, the clock frequency instability and message propagation time uncertainty are two main sources of time stamping errors in distributed systems. Network protocols coping with the problem perform series of actions that are based on distribution of timestamps from reference sources (referred to as offset synchronization) and adjustment of coefficients used for calibration of clock speeds (referred to as rate synchronization) [4].

2.2. Quartz crystal resonant frequency issues

Quartz crystals, commonly used in electronic clocks, work on the principle of piezoelectricity. The fundamental frequency of a crystal is defined mainly by its cut, shape and size. A number of miscellaneous physical factors also influence the frequency, such as mechanical stress and the temperature. Two main quantitative measures of crystal quality are the error (frequency imprecision) and the uncertainty (frequency instability). The imprecision corresponds to a difference between the actual and nominal frequencies, whereas the uncertainty describes deviations around the average value occurring over time. Variations are, naturally, dependent on the observation period. Therefore, they are usually divided into short-term and long-term instabilities [5]. The expression "intermediate-term" is also used.

Longest-term changes of the frequency are referred to as aging. Aging is the consequence of movement of impurity particles inside the crystal, microscopic changes in crystal shape due to mechanical stress relaxation, and material (usually dust particles) being adhered to or released from the crystal (which changes its mass and shape). Aging slows with time, so the older the crystal, the better its stability. Since aging progresses regardless of the fact whether the crystal is used or not, unused aged crystals are more valuable. Common aging rates are in the order of 10^{-7} (relative) per month for new crystals.

Vibrations and shocks are another reason causing the frequency discrepancy. While this is not of great importance in civil engineering measurements, where shocks are rare and vibrations are of moderate intensity, they have to be considered in more dynamic

environments (*e.g.* aircrafts). Changing gravity orientation also influences the crystal frequency. Moderate shocks, vibrations and changes in g direction alter the frequency in the order of 10^{-9} [6]. The electric current driving a crystal influences its resonant frequency as well. If the current is strong, long-term stability decreases, whereas if it is weak, an inevitable noise current of higher relative proportion causes short-term instability. Not surprisingly, crystals are also sensitive to the applied voltage, which is usually referred to as frequency pushing and pulling. Finally, retrace is another effect related to the electric supply: after each shutdown and power-up, the frequency achieves a slightly different value [7]. The list of other minor frequency deviation sources is long and includes atmospheric conditions and electromagnetic influences.

The most important cause of short, intermediate, and long-term frequency instability is the temperature. Ordinary quartz crystal oscillators used in everyday electronics are labelled RTXO (*Room Temperature Crystal Oscillators*), which means they are not temperature compensated. TCXO (*Temperature Compensated Crystal Oscillators*) measure the temperature and make appropriate corrections to obtain the stable output rate. An obvious problem with these devices is that their much greater complexity dictates a much higher price. OCXO (*Oven Controlled Crystal Oscillators*) are the most precise form of quartz oscillators, used in high precision measurements, where the crystal temperature is kept constant, so a temperature influence on the frequency is neglected. Typically, wireless sensor devices use RTXO oscillators not only for their much lower cost, but for lower energy consumption (an important issue in battery-powered systems) as well.

Most crystal cuts exhibit a negative parabolic frequency vs. temperature dependence, where the maximum corresponds to the preferable working temperature (usually, the room temperature).

2.3. WSN synchronization algorithms using comparison of timestamps

Many protocols of clock synchronization in wireless sensor networks incorporate the method of saving pairs of time data points (timestamps) acquired from two nodes and then comparing them using a variety of linear regression algorithms. If timestamps from one node are plotted as a function of the other one's, an approximately straight line is obtained, and the initial offset (difference between the clocks' readings) and the ratio between the clocks' speeds (timer frequencies) can be estimated from it. Such protocols include: Tiny-sync and Mini-sync [8], Timing-Sync Protocol for Sensor Networks (TPSN) [9], Scalable Lightweight Time-Synchronization Protocol (SLTP) [10], Tsync [11], Flooding Time Synchronization Protocol (FTSP) [12], Rapid Time Synchronization (RATS) [13], PulseSync [14], and Gradient Time Synchronization Protocol (GTSP) [15]. Some authors claim that the precision of a method increases with the number of timestamps gathered. However, in some cases the temperature varies among nodes (most often due to an unequal exposure to the sun), and this causes considerable variations in resonant frequencies of quartz crystals, which are the base of all system timers. The slope of the line then exhibits variations, and linear regression based on samples made in the past (on different temperatures) becomes unreliable.

2.4. Research questions

This paper presents methods used to investigate main sources of time stamping errors in a star topology wireless network (developed at the Faculty of Civil Engineering in Belgrade), namely signal propagation uncertainties and frequency drifts based on temperature fluctuations and other physical factors. Algorithms have been developed and implemented to deal with these problems in realistic conditions, where nodes are exposed to the elements. The

propagation uncertainty of radio modem output was resolved by a form of "post-factum" [16] synchronization, whereas the method of linear regression to determine ratios of clock frequencies was replaced by on-spot counting either prior to or during the measurement.

3. System design and theoretical simultaneity analysis

3.1. Wireless node design

The developed wireless node consists of a main processor board, an accelerometer board, batteries and a radio modem, packed inside a plastic housing (shown in Fig. 1). Other parts can optionally be attached: a display in its own case, used for diagnostics and debugging; a solar panel, used for energy harvesting; and a directional Yagi antenna, used for long-range radio communication.

The main board and accelerometer board are made in the surface mount technology (SMT). The core component of the device is an ADuC845 microcontroller. It is appropriate for its low power consumption and a possibility to enter the sleep mode. A three-axial MEMS accelerometer LIS3LV02DL is used as the primary sensor. Sampling is normally performed using ±2g range and 160 Hz sampling rate. One measurement cycle acquires 3200 12-bit samples for each axis and lasts approximately 20 seconds. The system architecture is open for an addition of analogue sensors or use of other digital sensor types.

A packet modem Decode PRM-4 is used, with a relatively low carrier frequency of 863−867 MHz, which does not allow a high bit rate, but improves propagation inside closed spaces and in the presence of obstacles. The output power is programmable (up to 19 mW), which enables energy saving. The maximum open field range was determined to be around 800 m in the central city area, or about twice as large in the suburbs.

Fig. 1. Parts of the sensor device: the housing, LCD, the Yagi antenna, the boards, the radio modem, and the solar panel.

The embedded Real-Time Operating System (RTOS) is original, written mostly in the assembler, with the flow control core in embedded C. The algorithm of operation is as follows. The nodes spend most of their time in the sleep mode. Most peripheral components are shut down consuming no energy during this period. The nodes are activated periodically using the embedded timer with programmable period, and there is a possibility for an adaptive

sleep mode (short wake-up period during the day and long period or no wake-up during the night). The sleep mode can also be interrupted by attaching the display and pressing the wake key (in which case a node enters the calibration procedure). Upon the scheduled wake-up, a node turns the radio modem on – to check for the presence of the beacon signal broadcasted from the central station (the system hub; usually a laptop with a radio modem). It enters a prolonged active period if it is detected, awaiting user commands, or reverts to sleep otherwise. The beacon signal also serves for the time offset synchronization of the nodes (this method is referred to as Reference Broadcast Synchronization or RBS [16, 17]). The star network topology is used, suitable for a small to intermediate size network in which time synchronization has priority over energy saving.

Data compression of the vibration signal is performed in order to minimize the data transfer volume (the most energy consuming process). The algorithm was implemented on a relatively slow processor with limited memory resources, so it had to be customized using a number of techniques. Loops and recursive procedures were coded in the assembler to maximize their speed. Data overlaying was used, meaning that different variables - not used at the same moment – use the same memory space (overlap). The signal was coded using the differential pulse code modulation (DPCM) in conjunction with a predictive algorithm based on periodicity detection, and then compressed using the Huffman's entropy coding [18].

3.2. Timestamp uncertainty analysis

Dynamic characteristics of a sensor are excluded from the model. The sensor itself can introduce significant delays, as well as signal distortion; however, this varies greatly between different types of sensors. The effects of propagation uncertainties and frequency instabilities are analyzed. A successful and failed RBS in a single-hop network are illustrated in Fig. 2. If the RBS procedure is successful then interval 1 has to be calculated and the physics of beacon emission process is irrelevant. If the RBS fails (*e.g.* node 3 on the diagram) then interval 2 has to be calculated, and then the indetermination of transmission time through the output electronics of the base station influences event timestamping.

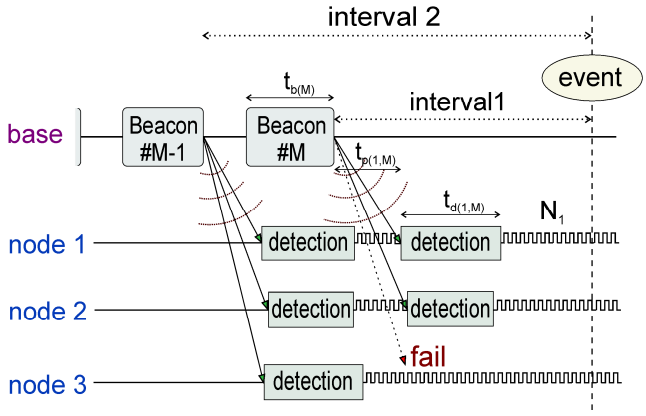

Fig. 2. The time diagram of a successful and failed RBS.

$$t_{\mathrm{n}} = t_{\mathrm{out}} + t_{\mathrm{p.n}} + t_{\mathrm{d.n}} + \frac{N_{\mathrm{n}}}{f_{\mathrm{n}}} \tag{2}$$

describes the timestamp of an event detected by the node n; t_{out} is the moment of beacon emission from the hub, t_p is the time of signal propagation, t_d is the time needed by the hardware to detect the beacon signal and start the counter, N is the number of counter ticks recorded prior to event detection, and f is the timer frequency. If the RBS procedure is successful then the value of t_{out} is equal for all nodes, so the maximum uncertainty of a timestamp is:

$$\Delta t_n = \Delta t_{p.n} + \Delta t_{d.n} + \Delta\left(\frac{N_n}{f_n}\right), \tag{3}$$

where Δ denotes the maximum uncertainty of respective value. Radio waves propagate at $3 \cdot 10^8$ m/s and even in the cases when distances between the base station and different nodes vary significantly, they can be measured with sufficient precision to make Δt_p fall well below a microsecond. Therefore, this term is neglected. Due to the integer nature of counting, its maximum uncertainty equals to one, so we can state:

$$\Delta t_n = \Delta t_{d.n} + \frac{1}{f_n} + N_n \frac{\Delta f_n}{f_n^2}. \tag{4}$$

Replacing N/f with the time that elapsed between the timer start and the event t_e, and introducing the maximum relative variation of timer frequency δf, we obtain:

$$\Delta t_n = \Delta t_{d.n} + \frac{1}{f_n} + \delta f_n \cdot t_{e.n} \approx \Delta t_{d.n} + \delta f_n \cdot t_{e.n}. \tag{5}$$

It will be shown that Δt_d is about 40 times larger than the machine cycle $1/f$, so the latter is neglected. The maximum deviation between timestamps from two different nodes n and m is the sum of maximum deviations of these single nodes:

$$\Delta(t_n - t_m) = \Delta t_{d.n} + \Delta t_{d.m} + t_{e.n}\left(\delta f_n + \delta f_m \frac{t_{e.m}}{t_{e.n}}\right) \approx 2\Delta t_d + t_e(\delta f_n + \delta f_m). \tag{6}$$

Although t_e has a unique value for each node due to different start moments of counting, we can approximate it with a single value, because if t_e is small then Δt_d is dominant in (5), whereas if t_e is large, the two values converge. We can also assume that t_d has an indetermination that depends on hardware design and does not vary significantly between nodes ($\Delta t_{d.n} \approx \Delta t_{d.m}$). Replacing the sum of relative uncertainties with the uncertainty of the ratio, we obtain:

$$\Delta(t_n - t_m) \approx 2\Delta t_d + \delta\left(\frac{f_n}{f_m}\right) \cdot t_e. \tag{7}$$

In practice, radio communication is susceptible to losses and it is not likely that a large number of nodes could all receive a single beacon signal and synchronize the timers thus performing a successful RBS procedure. The base station is therefore programmed to emit several beacons in short succession, to maximize the odds for a successful measurement trigger. Each of these beacons contains a variable number of node timer cycles after which the measurement should start (calculated using the base station clock). If a node misses the final beacon but receives one of the previous ones, there is an additional term in (5) and then it reads:

$$\Delta t_n \approx \Delta t_b + \Delta t_{d.n} + \delta f_n \cdot t_{e.n}, \tag{8}$$

because t_{out} is no longer in common. t_b represents the beacon emission time. If standard deviations of terms on the right side are known, we can state that:

$$\sigma t_{\mathrm{n}} \approx \sqrt{(\sigma t_{\mathrm{b}})^2 + (\sigma t_{\mathrm{d.n}})^2 + (\sigma(f_{\mathrm{n}}) \cdot t_{\mathrm{e.n}})^2}\,, \tag{9}$$

the first term on the right being optional (for a failed RBS procedure). For the standard deviation of timestamp difference between nodes *n* and *m*, we can state that:

$$\sigma(t_{\mathrm{n}} - t_{\mathrm{m}}) \approx \sqrt{2(\sigma t_{\mathrm{b}})^2 + 2(\sigma t_{\mathrm{d}})^2 + (\sigma(\frac{f_{\mathrm{n}}}{f_{\mathrm{m}}}) \cdot t_{\mathrm{e}})^2}\,, \tag{10}$$

t_{b} term being optional. Without perfect instruments to measure single t_{b} or t_{d}, only the differences can be determined. Single terms have no practical significance and they are replaced with differences between pairs, labelled τ:

$$\sigma\tau_{\mathrm{nm}} \approx \sqrt{\sigma\tau_{\mathrm{b}}{}^2 + \sigma\tau_{\mathrm{d}}{}^2 + (\sigma(\frac{f_{\mathrm{n}}}{f_{\mathrm{m}}}) \cdot t_{\mathrm{e}})^2}\,, \tag{11}$$

where τ_{d} refers to a pair of nodes and $\sigma\tau_{\mathrm{d}}$ equals $\sqrt{2}$ times standard deviation of the beacon detection time; τ_{b} refers to a pair of beacon signals and $\sigma\tau_{\mathrm{b}}$ equals $\sqrt{2}$ times standard deviation of the modem "packetization time".

3.3. Elimination of packetization time uncertainty from the equation

The packetization time is the time that elapsed between writing the last byte to the modem (RS232 input) and the beginning of physical radio transmission. It has the uncertainty of 0.5 ms due to the manufacturer's specification. This is a very large value compared to other uncertainties in the system.

The base station emits several beacons in short succession, and nodes synchronize with each one they receive (beacons contain variable information about the delay prior to the measurement start). Those communicating regularly synchronize with the last one. Nodes that miss it experience the problem of large t_{b} indetermination. Operating system interruption of the process would make it even bigger. This chance is reduced by triggering the parallel process of serial transmission as fast as possible after the processor's counter readout; the program performs delay calculation only, taking a fraction of microsecond of the processor time.

Adding an occasional uncertainty of a millisecond magnitude to the equation is still acceptable for many SHM applications, but there exists a solution to this problem, which reduces the error to the order of reception uncertainty. All nodes timestamp all beacons they receive from the hub and append this information to the measurement results. Data from any node that captured both critical beacons can be used to calculate the actual time that elapsed between physical emissions of beacons and therefore eliminate the packetization time offset from the equation. The packetization time uncertainty is replaced by another detection uncertainty, so for a failed RBS we can state that:

$$\sigma\tau_{nm} \approx \sqrt{(\sigma\tau_{\mathrm{d}}^*)^2 + (\sigma(\frac{f_{\mathrm{n}}}{f_{\mathrm{m}}}) \cdot t_{\mathrm{e}})^2}\,, \tag{12}$$

$$\sigma\tau_{\mathrm{d}}^* = \sqrt{2(\sigma t_{\mathrm{d.n}})^2 + (\sigma t_{\mathrm{d.m}})^2} = \sqrt{\frac{3}{2}}\sigma\tau_{\mathrm{d}}\,, \tag{13}$$

in case RBS failed on node *m*, and succeeded on node *n*, or:

$$\sigma\tau_{\mathrm{d}}^* = \sqrt{2(\sigma t_{\mathrm{d.n}})^2 + 2(\sigma t_{\mathrm{d.m}})^2} = \sqrt{2}\sigma\tau_{\mathrm{d}}\,, \tag{14}$$

in the case when the RBS failed on both nodes and synchronization was performed by using data from a third node. This procedure is a form of post-factum synchronization.

Since the interval between successive beacons is small (under a second), a typical frequency uncertainty causes an error of up to a microsecond, which is neglected in the above calculation, since other terms in (12) are of the 10 μs order.

4. Experiments

4.1. Frequency and frequency vs. temperature calibration

First, processors were tested under laboratory conditions. They are driven by 32768 Hz tuning fork quartz crystals. A phase-locked loop (PLL) frequency multiplier is used to achieve the clock rate of 6.29 MHz. Nodes were programmed to flip a digital output pin and produce a square wave with its frequency several times lower. in this way, a potential frequency multiplier error (which should be negligible [19]) adds to the result. The measurement was performed on 6 boards using a high precision counter Pendulum CNT-90. The obtained relative differences are about 10^{-5} between the processors from the same set, and about 10^{-4} between the different sets (processors were acquired on two different occasions). The standard deviation of core frequency is between $2.7 \cdot 10^{-9}$ and $5.6 \cdot 10^{-8}$ relative, observed over a five-minute period (absolute values are shown in Table 1). For good quality RTXO oscillators, this deviation can go as low as 10^{-9} [20].

Although using the measured frequencies instead of the nominal ones to calculate timestamps improves accuracy, validity of this method alone is limited due to the crystal aging and frequency instability. With the embedded thermometer in the processor, it is possible to make a software correction for the temperature based frequency drift. Similar methods have been used by [21] and [22]. Experiments were performed in order to test accuracy of the theoretical model for low cost oscillators. Differences between experimental and theoretical data for each piece were examined, as well as variations in the obtained coefficients among different pieces.

Theoretically, low frequency tuning fork quartz crystals have an approximately parabolic function $f(\theta)$, where f is the crystal resonant frequency and θ is the temperature [5]:

$$f(\theta) = f_0(1 - c(\theta - \theta_0)^2). \tag{15}$$

A temperature chamber and a counter were used in the experiment. Temperatures ranged from about 0°C to about 50°C. Obtained coefficients c vary from $3.4 \cdot 10^{-8}$ to $4.6 \cdot 10^{-8}$ °C^{-2} (they are presented in Table 1) and temperatures θ_0 are 20–25°C. The results are not perfect due to a number of factors, mainly inability to accurately determine the crystal temperature and keep it stable. Effects of thermal hysteresis [23] and thermal gradient, among others [5, 6], contribute to the error. The thermal hysteresis means that the crystal resonant frequency is not only a function of its current temperature, but depends on the history of temperature variations as well. The thermal gradient describes spatial variations of the temperature, which can never be identical in all points of the chamber.

A conclusion based on the observed differences between the measurement data and fitted theoretical curves, shown in Fig. 3, is that this sort of correction keeps the error under a ppm if the system is used indoors, whereas its values of up to 10^{-5} in the case of larger (tens of degrees) temperature fluctuations are possible. Using the average coefficients provided by the manufacturer, without the calibration of each piece, would reduce the precision significantly.

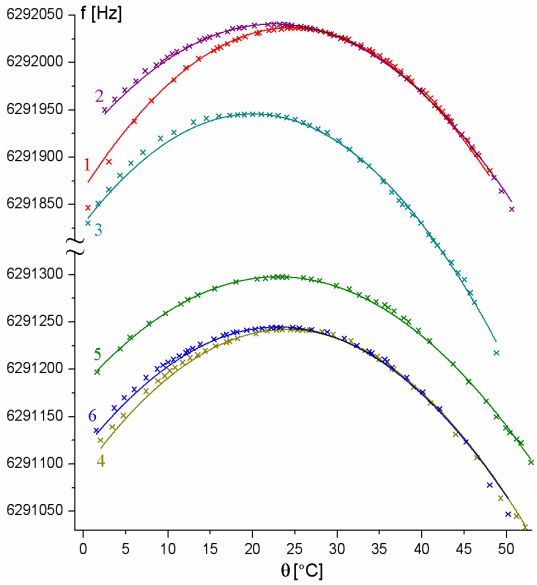

Fig. 3. Core frequencies vs. temperature.

Table 1. Core frequencies of processors, standard and peak-to-peak deviations, and temperature dependence coefficients.

#	f_o [MHz]	σf [Hz]	p–p f[Hz]	c [°C^{-2}]	θ_o [°C]
1	6.292034	0.3554	0.8542	$4.50 \cdot 10^{-8}$	24.5
2	6.292040	0.1184	0.2949	$3.76 \cdot 10^{-8}$	22.5
3	6.291937	0.0187	0.0591	$4.60 \cdot 10^{-8}$	20.0
4	6.291240	0.0170	0.0574	$4.12 \cdot 10^{-8}$	24.0
5	6.291302	0.0991	0.2756	$3.43 \cdot 10^{-8}$	23.0
6	6.291245	0.0453	0.1408	$3.89 \cdot 10^{-8}$	23.0

Frequency deviations of 10^{-5} order cause the time stamping uncertainty of 0.2 ms magnitude near the end of a usual 20-second measurement period. A software temperature correction may therefore be applied under the conditions of moderate temperature variations around the room temperature (near the top of curves from Fig. 3; the preferred temperature range is approximately 18–30°C), or if data simultaneity of a microsecond order is not required.

4.2. Determination of beacon detection time uncertainty

Series of experiments with a storage oscilloscope have been performed to observe differences in reaction to the beacon on different nodes. They were programmed to switch the digital output upon reception of a beacon, and this was used to trigger the oscilloscope. Differences in reaction times obtained in 150 broadcasts for three pairs of modems exhibit system and random deviations in the order of microseconds. Results are presented in Table 2 (τ_d) and histograms in the left column of Fig. 4. Modem pairs 1–2 and 3–1 showed exactly the same system offset of 4.9 μs (the first mentioned being slower), and the pair 3–2 matched perfectly with 9.8 μs. This could be a consequence of a quantized inner modem setting

invisible to the user. More important than the system offset, which is easily compensated by software, are the distribution and standard deviation. All pairs showed the standard deviation between 6.1 and 6.8 μs, the maximum deviation between 17 and 21 μs, and an approximately Gaussian distribution. This is caused mainly by modem imperfection, but it includes an uncertainty caused by node software, related to event detection. The embedded RTOS loops checking for flags from different interrupts, thus introducing a random offset within ±1.75 μs range.

The experiments were performed with the intentional miss of the final beacon on one device, to determine the packetization time uncertainty (Δt_b). Its value never exceeded 0.18 ms.

4.3. Electric excitation test

The next step was to check if the real event time stamping error matches the combination of beacon propagation uncertainty and laboratory results for frequency instability. The latter were obtained under ideal conditions with processors in very stable regime, unlike in practice, where devices are exposed to mechanical shocks, processors are turned on and off (as well as peripheral components), the temperature varies, radio waves are emitted in the immediate proximity, *etc*.

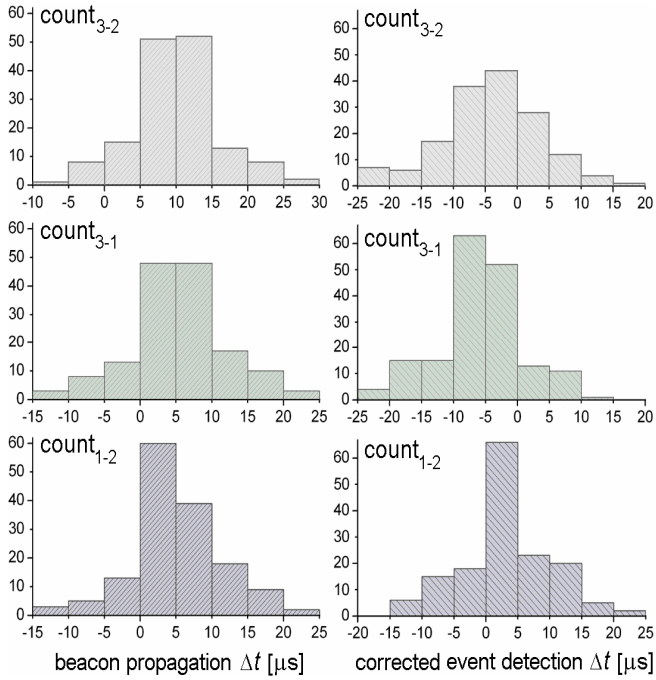

Fig. 4. Histograms of differences in beacon detection times and corrected event timestamps for pairs of nodes.

A signal generator was connected to the input pins of pairs of nodes in parallel. It was set to generate a low frequency square wave. Its shape was observed on a storage oscilloscope and verified to have 100 ns rising and falling edge (causing a negligible hardware detection uncertainty). Pairs of nodes were ordered to execute an ordinary measurement, 20 seconds long, then detect the signal edge and timestamp it. It actually took 22–24 seconds from the

timer start to the event (due to the initial and event-waiting delay). The following corrections have been made: counter ticks were divided by respective frequencies and beacon propagation offsets for respective pairs were compensated. The results (timestamp differences τ_{nm} in Table 2 and right column of Fig. 4) show that standard deviations have risen by up to 1.8 μs. The obtained system offsets range up to 5.8 μs, and they do not add up, like in the experiment with beacon signal detection. The obvious reason for both effects is that frequencies have not drifted to new stable values, but exhibit considerable instability during approximately two hours the experiment has taken. Another uncertainty introduced is the event detection delay (random ±1 μs offset).

The results from Table 2 imply that instabilities in frequency ratios (calculated based on $\overline{\tau_{nm}}$ and increase of $\sigma\tau$) range up to $2.5 \cdot 10^{-7}$ relative. This is several times larger than the highest deviation from Pendulum counter calibration. Since the times involved are longer, conditions are not ideal, and the values describe pairs (not single processors), it can be concluded that the results are within expected limits.

Table 2. Average values and standard deviations of differences in beacon detection time sand corrected event time stamps.

pair	$\overline{\tau_d}$ [μs]	$\sigma\tau_d$ [μs]	$\overline{\tau_{nm}}$ [μs]	$\sigma\tau_{nm}$ [μs]
1–2	4.9	6.3	3.1	6.9
3–1	4.9	6.8	–5.8	6.8
3–2	9.8	6.1	0.8	7.9

4.4. On-spot frequency ratio calibration

A simple procedure taking several minutes can help keeping track of the ratio of processors' frequencies. Nodes are synchronized with a single broadcast and left on stand-by until the user decides to read their timers. Ratio of nodes' counts equals to the ratio of their processors' current core frequencies. Typical results of series of these experiments with pairs of nodes are presented in Fig. 5. One node was exposed to a higher temperature, so the frequency ratio drifted from its initial value established for the room temperature. The longer the period, the better the ratio converges. The gray area represents the frequency ratio within standard deviation limits, calculated on the basis of two beacon detection times.

A conclusion can be drawn that 2–3 minutes is just the right margin to obtain satisfactory results (the frequency ratio standard deviation based on two beacon receptions goes under 10^{-7} for periods over 1.5 minutes). Long periods (*e.g.* 10 minutes or more) are pointless because the achieved precision exceeds the limit of short-term frequency instability determined by Pendulum counter calibration.

When "full" on-spot frequency ratio calibration is impractical, for lack of either operator's time or nodes' energy reserves, a simplified version of this algorithm can be employed. After the end of measurement, the base station sends an additional beacon to let nodes timestamp another reference point in time. A method of sending several beacons in short succession and making post-factum corrections, described earlier (used for correcting the packetization uncertainty if the RBS fails), can be used here too, although it causes a deterioration of the results. With two reference time-points, the frequency ratio can be calculated in the same way, only this time the measurement of vibrations is performed while counting, and the precision is worse because the calibration period is not long enough.

Although the results are displayed for a pair, the identical procedure can be applied on an arbitrary number of nodes simultaneously.

The conclusion is that, when using on-spot frequency ratio calibration, the accuracy increases to about 10^{-7} and closes in on the magnitude of short-term frequency instability determined by the Pendulum counter (around 10^{-8}). Although the simultaneity is important in SHM applications, absolute values of frequencies do not have to be accurate anywhere near this order of magnitude, so regular absolute frequency calibrations do not have to be performed if this method is applied. Practically, a single calibration using an ordinary counter (frequency meter) can serve for a crystal-processor lifetime.

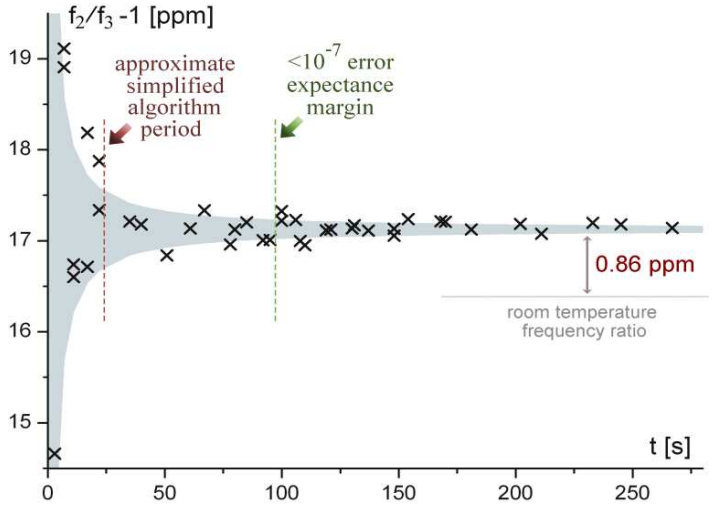

Fig. 5. Typical results of on-spot frequency ratio calibration for a pair of nodes.

5. Conclusions

A wireless sensor node for vibration measurements in civil engineering was designed. The proposed wireless sensor network is optimized for satisfactory time synchronization in realistic field conditions. Methods for estimation of simultaneity of data acquired from remote nodes were described.

Post-factum correction was applied to rectify the uncertainty based on the radio modem signal transmission time, and a method of on-spot frequency ratio determination by counting was proposed instead of usual frequency ratio determination based on sets of saved samples, which may have been taken at different temperatures, and might be inaccurate if the temperature varies among sensor nodes. Software correction of the timer frequency based on the embedded thermometer reading was presented as an option, too, but the problems associated with it are a lengthy calibration procedure requiring bulky equipment, and a limited precision if it is used in far from the room temperature.

The achieved synchronization accuracy for measurement periods under a minute is in the order of 10 μs. Commercial MEMS sensors, currently employed in the system, have the maximum sampling frequency in the order of kilohertz. Therefore, the established precision is more than suitable for vibration measurements in civil engineering, and the presented system could be used with higher bandwidth sensors.

Acknowledgements

This work was supported in part by the Ministry of Science and Technological Development of Republic of Serbia, under the grant TR-36048.

References

[1] Karbhari, V.M., Ansari, F. (2009). *Structural Health Monitoring of Civil Infrastructure Systems*. Woodhead Publishing, Ltd.

[2] Malović, M., Brajović, L., Mišković, Z., Todorović, G. (2013). Vibration measurements using a wireless sensors network. *Technics*, 68, 19–26.

[3] Mahalakshmi, M. (2012). *8051 Microcontroller Architecture, Programming and Application*. Laxmi Publications.

[4] He, L.M. (2009). Time synchronization for wireless sensor networks. *Proc. of the 10th Int. Conf. on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing*. Daegu, Korea, 438–443.

[5] Grzelak, S., Kowalski, M., Czoków, J., Zieliński, M. (2014). High resolution time-interval measurement systems applied to flow measurement. *Metrol. Meas. Syst.*, 21(1), 77–84.

[6] Vig, J.R., Ballato, A. (1999). Frequency control devices. Thurston, R.N., Pierce, A.D., Papadakis, E. (eds.). *Physical Acoustics 24: Ultrasonic Instruments and Devices II*. London, Academic Press, 209–273.

[7] Zhou, H., Nicholls, C., Kunz, T., Schwartz, H. (2008). Frequency accuracy & stability dependencies of crystal oscillators. *Carleton University, Systems and Computer Engineering, Technical Report SCE-08-12*.

[8] Yoon, S., Veerarittiphan, C., Sichitiu, M.L. (2007). Tiny-sync: Tight time synchronization for wireless sensor networks. *ACM Trans. Sens. Netw.*, 3(2), 1–34.

[9] Ganeriwal, S., Kumar, R., Srivastava, M.B. (2003). Timing-sync protocol for sensor networks. *Proc. of the 1st Int. Conf. on Embedded Networked Sensor Systems, ACM*. Los Angeles, CA, 138–149.

[10] Gelyan, S.N., Eghbali, A.N., Roustapoor, L., Abadi, S.A.Y.F., Dehghan, M. (2007). SLTP: Scalable lightweight time synchronization protocol for wireless sensor network. Zhang, H., Olariu, S., Cao, J. (eds.). *Mobile Ad-Hoc and Sensor Networks*. Berlin, Heidelberg, Springer, 536–547.

[11] Dai, H., Han, R., (2004). TSync: a lightweight bidirectional time synchronization service for wireless sensor networks. *ACM SIGMOBILE Mob. Comput. Commun. Rev.*, 8(1), 125–139.

[12] Gheorghe, L., Rughinis, R., Tapus, N. (2010). Fault-tolerant flooding time synchronization protocol for wireless sensor networks. *Proc. of the 6th Int. Conf. on Networking and Services (ICNS), IEEE*. Cancun, Mexico, 143–149.

[13] Kusý, B. (2007). *Spatiotemporal Coordination in Wireless Sensor Networks*. Ph.D. Thesis. Vanderbilt University.

[14] Lenzen, C., Sommer, P., Wattenhofer, R. (2009). Optimal clock synchronization in networks. *Proc. of the 7th Conf. on Embedded Networked Sensor Systems*. Berkeley, CA, 225–238.

[15] Sommer, P., Wattenhofer, R. (2009). Gradient clock synchronization in wireless sensor networks. *Proc. of the 8th ACM/IEEE Int. Conf. on Information Processing in Sensor Networks*. San Francisco, CA, 37–48.

[16] Elson, J. (2003). *Time Synchronization in Wireless Sensor Networks*. Ph.D. Thesis. University of California.

[17] Sivrikaya, F., Yener, B., (2009). Time synchronization. Zheng, J., Jamalipour, A. (eds.). *Wireless Sensor Networks: A Networking Perspective*. Hoboken, John Wiley & Sons, 285–306.

[18] Shahbahrami, A., Bahrampour, R., Rostami, M.S., Mobarhan, M.A. (2011). Evaluation of Huffman and arithmetic algorithms for multimedia compression standards. *Int. J. Comput. Sci. Eng. Appl.*, 1(4), 34–47.

[19] Kundert, K. (2012). Predicting the phase noise and jitter of PLL-based frequency synthesizers. *http://www.designers-guide.org/Analysis/PLLnoise+jitter.pdf*

[20] Carr, J.J. (2002). *RF Components and Circuits*. Newnes.

[21] Marchetto, P., Strickhart, A., Mack, R., Cheyne, H. (2012). Temperature compensation of a quartz tuning-fork clock crystal via post-processing. *Proc. of Frequency Control Symp. IEEE Int.* Baltimore, MD, 1–4.

[22] Castillo-Secilla, J.M., Palomares, J.M., Olivares, J. (2013). Temperature-compensated clock skew adjustment. *Sensors*, 13(8), 10981–11006.

[23] Filler, R. L. (1990). Thermal hysteresis in quartz crystal resonators and oscillators. *Proc. of the 44th Annual Symp. on Frequency Control, IEEE*. Baltimore, MD, 176–184.