

UNIVERZITET U BEOGRADU  
ELEKTROTEHNIČKI FAKULTET

Miodrag Z. Malović

**RAZVOJ I ANALIZA SENZORSKOG  
SISTEMA ZA MERENJE VIBRACIJA  
GRAĐEVINSKIH OBJEKATA**

doktorska disertacija

Beograd, 2015.

UNIVERSITY OF BELGRADE  
FACULTY OF ELECTRICAL ENGINEERING

Miodrag Z. Malović

**DESIGN AND ANALYSIS OF A SENSOR  
SYSTEM FOR VIBRATION  
MEASUREMENTS  
IN CIVIL ENGINEERING**

Doctoral Dissertation

Belgrade, 2015

## PODACI O MENTORU I ČLANOVIMA KOMISIJE

### **Mentor:**

Prof. dr Tomislav Šekara, Univerzitet u Beogradu – Elektrotehnički fakultet

### **Članovi komisije:**

Prof. dr Tomislav Šekara, Univerzitet u Beogradu – Elektrotehnički fakultet

Prof. dr Vujo Drndarević, Univerzitet u Beogradu – Elektrotehnički fakultet

Prof. dr Ljiljana Brajović, Univerzitet u Beogradu – Građevinski fakultet

Prof. dr Slobodan Petričević, Univerzitet u Beogradu – Elektrotehnički fakultet

Doc. dr Marko Barjaktarović, Univerzitet u Beogradu – Elektrotehnički fakultet

Datum odbrane: \_\_\_\_\_.

# RAZVOJ I ANALIZA SENZORSKOG SISTEMA ZA MERENJE VIBRACIJA GRAĐEVINSKIH OBJEKATA

## Rezime

Nadzor stanja objekata ili *structural health monitoring* (SHM) obuhvata merne tehnike kojima se vrši akvizicija podataka relevantnih za otkrivanje oštećenja i procenu upotrebljivosti raznih vrsta struktura. SHM se najčešće primenjuje u građevinskoj struci, za određivanje stanja zgrada i kapitalnih infrastrukturnih objekata, kao što su mostovi, tornjevi, brane, tuneli, itd. Razlozi za nagli razvoj SHM u poslednjih par decenija su povećane potrebe za ispitivanjem ostarelih objekata, novi zahtevi za izgradnjom jeftinijih a kompleksnijih objekata na težim tipovima terena (što zahteva nadzor i za vreme gradnje i za vreme eksploatacije), i rast zahteva sigurnosnih i drugih standarda. S druge strane, tehnološki razvoj i pojeftinjenje bežičnih tehnologija i mikroelektromehaničkih (MEMS) senzora je uslovio da SHM postane dostupan širokom spektru korisnika.

U radu je opisan bežični senzorski sistem prvenstveno dizajniran za merenje vibracija na građevinskim objektima. SHM kapitalnih građevinskih objekata se najčešće vrši analizom modova oscilacija, što podrazumeva upoređivanje amplituda i faza pojedinih harmonika signala u različitim tačkama. U ovu svrhu se skoro isključivo koriste bežični sistemi, koji imaju izražene prednosti u odnosu na žičane. Nije potrebno nepraktično i skupo razvlačenje kablova, ceo sistem i pojedinačni senzori su lakši za premeštanje, i senzori se lakše mogu postaviti na teško pristupačna mesta. Postoje i mane, a to su manja brzina prenosa podataka, osetljivost na smetnje različitog porekla, uključujući otvorenost medijuma za nameran napad sa strane, i problemi sa ograničenošću napajanja. Senzorski uređaji su realizovani u *surface-mount* tehnologiji (SMT). Koriste MEMS akcelerometre za merenje ubrzanja. Njihova potrošnja je mala što odgovara zahtevima baterijski napajanih bežičnih uređaja. Napravljen je solarni panel koji služi za energetska dopunjavanje senzorskih uređaja u slučaju da se ovi postavljaju na mesta sa prirodnom svetlošću dovoljnog intenziteta. Korišćeni su radio modemi domaće proizvodnje za komunikaciju.

Osim u potpunosti originalnog hardvera, razvijen je i originalan softver velike kompleksnosti, počev od *embedded real-time operating system*-a (ili *embedded RTOS*-a) za senzorski uređaj, preko *embedded RTOS*-a za displej (koji predstavlja poseban uređaj koji se opciono priključuje, nezamenljiv u fazi razvoja uređaja, a koristan i u regularnom radu za dijagnostiku stanja i vršenje kompletne kalibracije), do programa za MS Windows koji se izvršava na centralnoj stanici bežične mreže (PC računaru), a sa senzorskim uređajima komunicira putem radio modema serijski povezanog sa računarom. Ovaj softver vrši funkcije počev od bazičnih, kao što je komunikacija sa periferijom (čitanje tastature, prikaz tačaka na ekranu, komunikacija sa akcelerometrom, upravljanje relejima za uključivanje i isključivanje delova uređaja, komunikacija sa modemom, itd), koja se vrši digitalnim izlaznim pinovima mikrokontrolera i ugrađenim interfejsima tipova I2C, SPI i RS232, do kompleksnih algoritama, kao što je kompresija podataka korišćenjem prediktivne tehnike zasnovane na detekciji periodičnosti signala i entropijskog Hafmanovog kodiranja. Kompresija podataka je od velikog značaja za sistem ovog tipa jer je ubedljivo najveći potrošač energije u sistemu modem u režimu slanja podataka, tako da treba izvršiti transmisiju što manje bajtova podataka. Poseban izazov predstavlja činjenica da je treba izvršiti u uslovima skućenog radnog prostora (male RAM memorije) i relativno sporog procesora. Programiranjem u assembleru povećana je brzina izvršenja procedure kompresije, a upotrebom tehnike preklapanja (*overlay*) memorijskih lokacija koje se ne koriste istovremeno uštedeno je na memoriji. Analizirani su različiti oblici diferencijalne pulsno-kodne modulacije (DPCM) i izabran je prediktivni algoritam zasnovan na obliku signala od pre jedne periode, a pod uslovom da signal zadovolji kriterijum da je pseudoperiodičan (da ima određen ponavljajući obrazac), što zahteva poseban algoritam. Izvršeno je entropijsko kodiranje Hafmanovim algoritmom, pri čemu su analizirane različite verzije Hafmanove tablice kodova radi optimizacije kompresionog odnosa.

Druga značajna stavka u realizaciji sistema za SHM kapitalnih građevinskih objekata jeste obezbeđivanje simultanosti mernih podataka. Simultanost mora biti takva da garantuje grešku manju od značajnog dela periode najviših harmonika signala. Srećom, što su objekti veći, frekvencije oscilacija su manje, tako da ostvarena sinhronizacija reda veličine desetina mikrosekundi premašuje potrebe realnih merenja, koje su oko milisekunde. Posebnu teškoću prilikom ispitivanja sinhronosti podataka

predstavlja činjenica da je teško simulirati događaj koji će biti istovremeno detektovan na udaljenim senzorima, kao i činjenica da su u stvarnosti senzorski uređaji izloženi vremenski varijabilnom dejstvu elemenata, kao što je prvenstveno osunčanost, koja dovodi do značajnih odstupanja u temperaturi, i samim tim frekvenciji kvarcnih kristala, koji služe kao osnova za systemske tajmere. Svako odstupanje frekvencije od nominalne ili kalibrisane vrednosti izaziva grešku vremenskog označavanja koja raste kako merno vreme odmiče.

Ispitivani su različiti efekti koji utiču na greške vremenskog označavanja događaja i time na sinhronost podataka sa različitih senzorskih uređaja. Ovo su neodređenosti vremena propagacije signala kroz predajni i prijemni modem i nestabilnosti frekvencija kristala satova senzorskih uređaja. Predložene su metode korekcije koje najbolje odgovaraju terenskim uslovima (nepredvidljivim varijacijama temperature) i analizirana je tačnost dobijenih podataka serijama eksperimenata.

Troosni akcelerometri, upotrebljeni u razvijenom sistemu, imaju relativno veliku kros-aksijalnu ili poprečnu osnu osetljivost (uticaj bočne pobude na izlaz akcelerometra). Ovaj efekat je eksperimentalno ispitivan i razvijena je metoda tilt-testa u 24 pozicije, koji se može vršiti bez ikakve dodatne opreme (jeftina alternativa skupim mernim sistemima). Pokazano je da se poprečna osetljivost može softverski korigovati i smanjiti nekoliko puta (tačna vrednost zavisi od ose; ograničenje predstavljaju elementi kućišta koji onemogućavaju postavljanje akcelerometra u željene položaje).

Domet radio komunikacije je testiran u uslovima urbanih smetnji. U slučaju upotrebe na manjim rastojanjima, može se koristiti režim smanjene snage, u kome se energetske rezerve sporije troše. Ustanovljeno je da su karakteristike sistema zadovoljavajuće za upotrebu na branama i mostovima, izuzev u slučajevima ekstremnih smetnji. Korišćene su različite vrste antena, od minijaturnih *stick* antena do omnidirekcionih i direkcionih *Yagi* antena.

Potrošnja uređaja je ispitana, kao i mogućnost dopunjavanja realizovanim solarnim panelom. Ako se sistem ne koristi intenzivno, moguć je neograničen rad, a u slučaju da se ne koristi solarni panel, baterije mogu pogoniti uređaj oko mesec dana.

Terensko testiranje uređaja vršeno je u Vojnotehničkom institutu, na pruzi kod termoelektrane Nikola Tesla u Obrenovcu, i na dva beogradska mosta. U VTI-u je ispitivana robusnost uređaja višečasovnim izlaganjem jakim vibracijama, a kojom

prilikom je vršena i analiza simultanosti sa mehaničkom pobudom. Prilikom merenja na pruži ustanovljena je mogućnost korišćenja sistema za određivanje brzine vozova i dinamički ugib pružnih pragova. Nije proizveden dovoljan broj uređaja da bi se izvršila kompletna modalna analiza vibracija mostova, ali su merenja vršena na tramvajskom i na mostu Gazela.

Velikim brojem eksperimenata potvrđena je upotrebljivost sistema i dokazana je vrednost originalnog softvera i originalnih algoritama za korekcije nesavršenosti sistema. Ukazani su pravci mogućeg daljeg razvoja ovakvog sistema.

**Ključne reči:** bežične senzorske mreže, akcelerometri, vibracije, frekvencija, vreme, simultanost, sinhronizacija, embedded RTOS, kompresija bez gubitaka, Hafmanovo kodiranje, periodičnost, kapitalni građevinski objekti

**Naučna oblast:** tehničke nauke, elektrotehnika

**Uža naučna oblast:** senzorski sistemi i merenja

**UDK broj:** 621.3

# **DESIGN AND ANALYSIS OF A SENSOR SYSTEM FOR VIBRATION MEASUREMENTS IN CIVIL ENGINEERING**

## **Abstract**

Structural Health Monitoring (SHM) involves measurement techniques used in damage detection and estimation of usability of different kinds of structures. SHM is usually performed in civil engineering, to determine states of buildings and capital infrastructural objects, such as bridges, towers, dams and tunnels. Reasons for the rapid development of SHM in the past two decades are increasing needs to monitor aging structures, new requests to build cheaper and more complex objects on heavier terrain (which often demands monitoring during both construction and exploitation), and growing demands implied by safety and other standards. On the other hand, technological advances and price decrease in the areas of wireless technologies and micro-electro-mechanical (MEMS) sensors caused SHM techniques to become widely available.

A wireless sensor system primarily designed to measure vibrations of civil engineering structures is described. SHM of capital civil engineering objects is often performed by modal analysis of vibrations, which includes comparison of amplitudes and phases of specific signal harmonics in different points. Wireless systems are commonly used for this, since they have significant advantages over the wired ones. There is no need to use impractical and expensive large cables, both the whole system and sensor nodes can be easily relocated, and sensors are easier placed at hard-to-reach spots. There are shortcomings as well, mainly lower data transfer bit rate, sensibility to different electromagnetic interferences, including openness of the medium for attack, and limited energy supply. Sensor devices of the developed system are made in surface-mount technology (SMT). They use MEMS accelerometers, with small energy consumption, suitable for battery powered wireless systems. A solar panel was constructed to harvest sun power and supply the devices with additional energy, in case they are placed on spots with sufficient intensity natural light. Radio modems of domestic production were used for communication.



Completely original hardware design was followed by original software of great complexity, including embedded real time operating system (RTOS) for the sensor device, embedded RTOS for the display device, and the program for MS Windows that is executed on the hub of the wireless system (a PC computer). The display is a separate processor device that can optionally be attached to the sensor device. It's irreplaceable during the device development and can be used during regular operation, to diagnose device state and to perform full parameter calibration. Sensor device embedded RTOS performs miscellaneous operations starting from basic ones, like the communication with peripheral components (e.g. keyboard read, screen pixels display, communication with the accelerometer and radio modem, or switching the relays that control the power supply to device parts), which is performed by digital output pins of microcontroller and implemented I2C, SPI and RS232 interfaces; up to most complex algorithms such as the data compression using predictive technique based on signal periodicity detection and Huffman entropy coding. Data compression is important for a system of this type because the greatest energy consumer on the board is the radio modem in the transmission regime, so there is an obvious need to transmit as little data as possible. Special challenge was the fact that this needed to be performed under the conditions of small memory space (RAM) and using a relatively slow processor. Assembler programming increased the speed of compression code execution, and the overlay technique (placing the different non-simultaneously used variables at the same locations) saved the memory. Different forms of differential pulse-code modulation (DPCM) coding were considered and a predictive algorithm was chosen which is based on the signal shape one period prior. It requires that it is first confirmed that the signal really exhibits sufficient level of periodicity or pseudo-periodicity (a certain repeating pattern), and this requires a special algorithm itself. Finally, entropy coding using Huffman algorithm was used, and different variations of Huffman tables were considered to optimize the compression ratio.

Another important item in the development of the system for SHM of capital civil engineering objects is securing sufficient simultaneity of measured data. The error must be guaranteed to be less than a significant part of a period of the highest harmonics of the signal. Fortunately, bigger the object, lower the resonant frequencies, so the achieved synchronicity in the order of ten microseconds exceeds the needs of real civil

engineering measurements, which are around a millisecond. Special difficulty in synchronicity verification was the fact that it is difficult to simulate an event that can be simultaneously detected by remote sensor devices. In addition, the sensor devices in the field are exposed to the elements, varying in time, such as the sunlight, which can cause significant temperature differences. This, in turn, leads to significant frequency drifts of quartz crystals, which serve as the base for system timers on all devices. Any frequency drift from its calibrated value causes the timestamping error that grows over time.

Effects causing the timestamping errors (and therefore synchronicity of data acquired at different nodes) were examined. They are uncertainty of signal propagation time through the output and input electronics, and instability of frequencies of quartz crystals used in devices clocks. Methods for correction have been developed and implemented, best suited for real life (unpredictable temperature variations) application of the system. Precision of acquired data was analyzed in numerous experiments.

Used three-axial accelerometers have relatively high cross-axial (or transverse) sensitivity (transverse excitation influence on the output signal). This effect has been experimentally observed and a 24-position tilt-test method was developed, which can be performed without any additional equipment, so it's a cheap alternative to expensive measurement systems. It has been shown that the transverse sensitivity can be corrected by software and decreased several times (the exact value depends on the axis; the limitations are implied by the elements of the housing which disable strict placement of the device into the desired, repeatable positions).

Range of the radio communication was tested under the conditions of urban interferences. In case of system use in closer ranges, decreased power regime can be used, in which energy reserves of nodes are depleted more slowly. It has been shown that the system characteristics are satisfying for the use on bridges and dams, except in cases of extreme interferences. Different sorts of antennas were used, from miniature stick antennas, to large omnidirectional and directional Yagi antennas.

Energy consumption of the sensor device was examined as well as the possibility of energy harvesting by the developed solar panel. If the system is not used intensively, practically infinite lifetime is achievable. In case solar panel is not used, batteries can power a sensor device for about a month.

Field testing of the devices was performed in Military technical institute (MTI) in

Belgrade, at the railroad near the thermal power plant Nikola Tesla in Obrenovac, and on two major Belgrade bridges. Robustness of the devices was tested at MTI Belgrade, by exposing them to many hours of strong vibrations (simultaneity analysis with uniform mechanical excitation was performed on this occasion). Railroad measurements gave insights into possibilities of use of this system to measure train velocity as well as dynamic deflection of wooden sleepers. Sufficient number of devices was not produced to enable complete modal analysis of bridge vibrations, but the measurements have been performed on the tram bridge and Gazela bridge in Belgrade.

Large number of experiments demonstrated the applicability of the developed system and proved the value of original software algorithms used to correct hardware imperfections. Possible directions of further research have been given.

**Keywords:** wireless sensor networks, accelerometers, vibrations, frequency, time, simultaneity, synchronization, embedded RTOS, lossless compression, Huffman coding, periodicity, capital civil engineering objects

**Scientific area:** technical sciences, electrical engineering

**Specific scientific area:** sensor systems and measurements

**UDK number:** 621.3

# SADRŽAJ

<a href="#">PREDGOVOR</a> .....	1
<a href="#">1. UVOD</a> .....	4
<a href="#">1.1. Monitoring građevinskih objekata</a> .....	4
<a href="#">1.2. Bežične senzorske mreže</a> .....	6
<a href="#">1.2.1. Senzorski uređaji</a> .....	7
<a href="#">1.2.2. Topologija mreže</a> .....	9
<a href="#">1.2.3. Senzori</a> .....	11
<a href="#">1.2.4. Softver</a> .....	12
<a href="#">1.3. Modovi oscilovanja građevinskih struktura</a> .....	13
<a href="#">1.4. Polazne pretpostavke i cilj istraživanja</a> .....	14
<a href="#">2. OSNOVE MODALNE ANALIZE</a> .....	16
<a href="#">2.1. Vibracije žice</a> .....	19
<a href="#">2.2. Vibracije pravougaonih membrana</a> .....	21
<a href="#">2.3. Vibracije kružnih membrana</a> .....	24
<a href="#">2.4. Longitudinalne vibracije štapova</a> .....	26
<a href="#">3. DIZAJN RAZVIJENOG SENZORSKOG SISTEMA</a> .....	30
<a href="#">3.1. Hardver senzorskog uređaja</a> .....	30
<a href="#">3.1.1. Procesor</a> .....	32
<a href="#">3.1.2. Akcelerometar</a> .....	35
<a href="#">3.1.3. Radio modem</a> .....	36
<a href="#">3.1.4. Displej</a> .....	38
<a href="#">3.1.5. Baterije i solarni panel</a> .....	40
<a href="#">3.1.6. Kućište</a> .....	40
<a href="#">3.2. Algoritam rada sistema</a> .....	41
<a href="#">3.3. Operativni sistem senzorskog uređaja</a> .....	44
<a href="#">3.4. Softver na centralnoj stanici</a> .....	46
<a href="#">4. KOMPRESIJA PODATAKA</a> .....	50
<a href="#">4.1. Algoritmi kompresije</a> .....	50
<a href="#">4.2. Detekcija periodičnosti signala</a> .....	54
<a href="#">4.2.1. Kros-korelaciona i autokorelaciona funkcija</a> .....	54

4.2.2. Estimacija kašnjenja .....	55
4.3. Algoritam kompresije sa TDE-DPCM preprocesorom .....	58
4.4. Hafmanove tablice i memorijska optimizacija .....	61
5. ISPITIVANJE KARAKTERISTIKA SISTEMA .....	64
5.1. Ispitivanje dometa radio modema .....	64
5.2. Ispitivanje ciklusa punjenja i pražnjenja solarnih ćelija .....	65
5.3. Ispitivanje poprečne osne osetljivosti akcelerometara .....	69
5.3.1. Matematički model .....	69
5.3.2. Tilt test u 24 pozicije .....	70
5.3.3. Eksperimenti .....	72
6. SIMULTANOST PODATAKA U BEŽIČNOJ MREŽI .....	75
6.1. Metode sinhronizacije u kompjuterskim i senzorskim mrežama .....	75
6.1.1. Izvori nepreciznosti satova .....	75
6.1.2. Spisak često korišćenih pojmova u metodima sinhronizacije .....	75
6.1.3. Uobičajeni mrežni algoritmi sinhronizacije .....	78
6.1.4. Sinhronizacija u bežičnim mrežama .....	79
6.2. Analiza greške u zvezdastoj mreži sa RBS signalom .....	82
6.3. Eliminacija neodređenosti vremena paketizacije modema .....	85
6.4. Metodologija .....	87
7. EKSPERIMENTALNO ODREĐIVANJE SIMULTANOSTI PODATAKA .....	88
7.1. Stabilnost i preciznost kvarenih oscilatora .....	88
7.2. Osnovna kalibracija procesora .....	89
7.3. Temperaturska kalibracija .....	91
7.4. Određivanje neodređenosti vremena detekcije prozivnog signala .....	92
7.5. Eksperimenti sa električnom eksitacijom .....	93
7.6. Metod brojanja na licu mesta .....	95
7.7. Zaključak o postignutoj simultanosti .....	98
7.8. Mehanički testovi simultanosti .....	98
8. PRIMENA SISTEMA U ŽELEZNICI .....	102
8.1. Značaj i tehnike merenja pomeraja pragova .....	102
8.2. Mogućnosti merenja dinamičkog ugiba pragova postojećim sistemom .....	104
8.3. Određivanje brzine vozova .....	105

<u>9. PROVERA OPRAVDANOSTI TDE-DPCM PREPROCESIRANJA U REALNIM MERENJIMA</u> .....	110
<u>10. ZAKLJUČAK</u> .....	119
<u>11. LITERATURA</u> .....	121
<u>12. PRILOG 1: ELEKTRIČNE ŠEME</u> .....	131
<u>13. PRILOG 2: IZVODI IZ PROGRAMA</u> .....	134
<u>13.1. Kompletan RTOS displeja</u> .....	134
<u>13.2. Hafmanovo kodiranje na senzorskom uređaju</u> .....	150
<u>13.3. Hafmanovo dekodiranje na PC računaru</u> .....	164
<u>BIOGRAFIJA</u> .....	166

## SPISAK SKRAĆENICA

<u>Skraćenica</u>	<u>Pun naziv</u>
A/D	Analogno-digitalna (konverzija)
BIOS	<i>Basic Input/Output System</i>
CCS	<i>Consensus Clock Synchronization</i>
COM	Serijski (komunikacioni) port PC računara
DFT	Diskretna Furijeova transformacija
DPCM	<i>Differential Pulse Code Modulation</i>
DSP	<i>Digital Signal Processor</i>
EEPROM	<i>Electrically Erasable Programmable Read-Only Memory</i>
FPGA	<i>Field-Programmable Gate Array</i>
FSK	<i>Frequency-Shift Keying</i>
FTSP	<i>Flooding Time Synchronization Protocol</i>
GPS	Globalni pozicioni sistem
GTSP	<i>Gradient Time Synchronization Protocol</i>
HRTS	<i>Hierarchy Referencing Time Synchronization</i>
I2C	<i>Inter-Integrated Circuit</i>
IBS	<i>Interval Based Synchronization</i>
IDE	<i>Integrated Development Environment</i>
IEEE	<i>Institute of Electrical and Electronics Engineers</i>
IP	<i>International Protection (Marking)</i>
IR	Infracrveno (zračenje)
ITR	<i>Individual-based Time Request</i>
KB	Kilobajt
LCD	Displej sa tečnim kristalom
LSB	Najniži (najmanje značajan) bit
LSD	<i>Low Self-Discharge</i>
LSSA	<i>Least Squares Spectral Analysis</i>
LTS	<i>Lightweight Time Synchronization</i>
MCXO	<i>Microcomputer Controlled Crystal Oscillator</i>
MEMS	<i>Micro-Electro-Mechanical System(s)</i>
MS	<i>Microsoft (Corporation)</i>
MSB	Najviši (najznačajniji) bit

NiMH	Nikl-metal hidridne (baterije)
NTP	<i>Network Time Protocol</i>
OCXO	<i>Oven Controlled Crystal Oscillator</i>
PC	Personalni kompjuter
PLL	Sinfazna petlja
PPM	Milioniti deo ( $10^{-6}$ )
PPS	Impuls u sekundi
PTP	<i>Precision Time Protocol</i>
PWM	Širinsko-impulsna modulacija
RAM	<i>Random Access Memory</i>
RBS	<i>Reference Broadcast Synchronization (ili Signal)</i>
RF	Radio-frekventni
RITS	<i>Routing Integrated Time Synchronization</i>
RLE	<i>Run-Length Encoding</i>
ROM	<i>Read-Only Memory</i>
RTOS	<i>Real-Time Operating System</i>
RTXO	<i>Room Temperature Crystal Oscillator</i>
SDRAM	<i>Synchronous Dynamic Random Access Memory</i>
SHM	<i>Structural Health Monitoring</i>
SLTP	<i>Scalable Lightweight Time-synchronization Protocol</i>
SMD	<i>Surface-Mount Device</i>
SMT	<i>Surface-Mount Technology</i>
SOC	<i>State of Charge</i>
SPI	<i>Serial Peripheral Interface</i>
SRAM	<i>Static Random Access Memory</i>
TCXO	<i>Temperature Compensated Crystal Oscillator</i>
TDE	<i>Time Delay Estimation</i>
TDP	<i>Time Diffusion Protocol</i>
TPSN	<i>Timing-sync Protocol for Sensor Networks</i>
TSS	<i>Time-Stamp Synchronization</i>
UART	<i>Universal Asynchronous Receiver/Transmitter</i>
UV	Ultraljubičasto (zračenje)
WSN	<i>Wireless Sensor Network(s)</i>



## SPISAK SLIKA

<b>Slika 1.1</b> Blok šema bežičnog uređaja .....	7
<b>Slika 1.2</b> BTnode, ETH Ciriha.....	7
<b>Slika 1.3</b> SunSPOT .....	9
<b>Slika 1.4</b> Topologija zvezde.....	9
<b>Slika 1.5</b> Topologija magistrale .....	9
<b>Slika 1.6</b> Mrežasta topologija.....	10
<b>Slika 1.7</b> Realan primer složene topologije multi-hop mreže .....	10
<b>Slika 1.8</b> Tacoma Narrows most .....	13
<b>Slika 2.1</b> Neki od registrovanih modova oscilacija brane Norsjö u Švedskoj [33] .....	16
<b>Slika 2.2</b> Deliće žice dužine $dx$ izložen istežanju .....	19
<b>Slika 2.3</b> Modovi oscilovanja žice .....	21
<b>Slika 2.4</b> Deliće membrane dimenzija $dx dy$ izložen transverzalnom istežanju .....	22
<b>Slika 2.5</b> Modovi oscilovanja pravougaone membrane .....	24
<b>Slika 2.6</b> Modovi oscilovanja kružne membrane .....	26
<b>Slika 2.7</b> Modovi oscilovanja štapa učvršćenog na levom kraju .....	28
<b>Slika 2.8</b> Modovi oscilovanja štapa slobodnog na oba kraja .....	28
<b>Slika 3.1</b> Glavna ploča senzorskog uređaja.....	30
<b>Slika 3.2</b> Mikrokontroler ADuC845 .....	32
<b>Slika 3.3</b> Blok šema procesora [35] .....	33
<b>Slika 3.4</b> Ploča sa akcelerometrom .....	35
<b>Slika 3.5</b> Blok šema LIS3LV02DL [37] .....	35
<b>Slika 3.6</b> Radio modem PRM-4 [39] .....	37
<b>Slika 3.7</b> Blok šema displeja [42] .....	39
<b>Slika 3.8</b> Displej izvan svog kućišta, sa funkcijama dugmića (levi red: levo, gore, dole, Escape; desni red: desno, buđenje, svetlo, Enter).....	39
<b>Slika 3.9</b> Realizovani solarni panel.....	40
<b>Slika 3.10</b> Kućište OKW C2016241 .....	40
<b>Slika 3.11</b> Senzorski uređaj sa centralnom stanicom i antenama .....	41
<b>Slika 3.12</b> Algoritam rada sistema .....	44
<b>Slika 3.13</b> Nadzorni program na PC računaru, glavni ekran.....	47
<b>Slika 4.1</b> Primer Hafmanovog drveta.....	52
<b>Slika 4.2</b> Primer aritmetičkog kodiranja .....	53

<b>Slika 4.3</b> Sinusna funkcija promenljive amplitude .....	57
<b>Slika 4.4</b> Poređenje TDE i DFT (diskretne Furijeove transformacije) primenjenih na signal sa sl. 4.3.....	57
<b>Slika 4.5</b> Približan algoritam kompresije na senzorskom uređaju; AC označava autokorelaciju .....	60
<b>Slika 5.1</b> Verovatnoća prenosa paketa kao funkcija predajne snage i udaljenosti.....	65
<b>Slika 5.2</b> SOC u funkciji napona tipične NiMH baterije [44].....	66
<b>Slika 5.3</b> Struja punjenja panela kao funkcija osvetljenosti.....	67
<b>Slika 5.4</b> Ciklusi punjenja i pražnjenja baterija senzorskog uređaja prilikom upotrebe solarnog panela.....	68
<b>Slika 5.5</b> Rotacija akcelerometra.....	70
<b>Slika 5.6</b> Senzorski uređaj u plastičnom kućištu.....	72
<b>Slika 6.1</b> Vremenski dijagram uspešnog i neuspešnog RBS-a .....	82
<b>Slika 7.1</b> Tipična promena rezonantne frekvencije kvarcnih kristala usled starenja .....	88
<b>Slika 7.2</b> Teorijska zavisnost $f(\theta)$ za kvarcne kristale tipa zvučne viljuške.....	91
<b>Slika 7.3</b> Zavisnost frekvencije procesora od temperature u eksperimentu sa temperaturskom komorom.....	92
<b>Slika 7.4</b> Histogrami razlike u vremenu propagacije prozivnog signala (levo) i histogrami razlike u vremenskom označavanju događaja sa softverskom korekcijom (desno) .....	95
<b>Slika 7.5</b> Rezultati kalibracije odnosa frekvencija dva uređaja metodom brojanja na licu mesta.....	96
<b>Slika 7.6</b> Normalizovana korelacija (levo) i histogram broja pojavljivanja maksimalne korelacije (desno), u funkciji vremenskog pomaka, za eksperimente na stolu .....	99
<b>Slika 7.7</b> Senzorski uređaji na vibracionoj platformi.....	100
<b>Slika 7.8</b> Normalizovana korelacija (levo) i histogram broja pojavljivanja maksimalne korelacije (desno), u funkciji vremenskog pomaka, za eksperimente na platformi .....	101
<b>Slika 8.1</b> Pruga za prevoz uglja u termoelektranu Nikola Tesla u Obrenovcu. ....	102
<b>Slika 8.2</b> Par koševa koji čine teretni dvodelni EDF vagon za prevoz uglja .....	102
<b>Slika 8.3</b> Laserski sistem za merenje statičkog ugiba šina.....	103
<b>Slika 8.4</b> Primeri određivanja vertikalnog pomeraja (dinamičkog ugiba) dvostrukom integracijom za slučaj relativno pravilnog signala .....	104
<b>Slika 8.5</b> Primer 1: vertikalno ubrzanje praga.....	106
<b>Slika 8.6</b> Primer 2: vertikalno ubrzanje praga.....	106
<b>Slika 8.7</b> Primena TDE metoda na signal iz primera 1 .....	107
<b>Slika 8.8</b> Primena TDE metoda na signal iz primera 2.....	107
<b>Slika 8.9</b> Vizuelni prikaz nastanka 'među' i glavnih pikova korelacije .....	108

<b>Slika 8.10</b> Dijagrami amplitude u funkciji frekvencije, dobijeni DFT-om, za signale iz primera 1 i 2.....	109
<b>Slika 9.1</b> Signali sa vibracione platforme, snimani na 160 Hz (a) i na 2560 Hz (b).....	111
<b>Slika 9.2</b> Vibracije pružnih pragova, dobro (a) i loše (b) podbijen prag .....	112
<b>Slika 9.3</b> Tramvajski most u Beogradu .....	113
<b>Slika 9.4</b> Vibracije tramvajskog mosta prilikom tramvajske (a) i eksitacije od strane motornih vozila (b) .....	114
<b>Slika 9.5</b> Dilataciona sprava stuba mosta Gazela, izgled sa unutrašnje strane .....	115
<b>Slika 9.6</b> Vibracije na sredini (a) i na stubu (b) mosta Gazela.....	116
<b>Slika 9.7</b> Dijagrami amplitude u funkciji frekvencije, za signale snimljene na sredini mosta.....	117
<b>Slika 9.8</b> Poređenje odnosa kompresije (KO) upotrebom DPCM susednih odmeraka i upotrebom TDE-DPCM .....	118
<b>Slika 12.1</b> Procesor sa pratećom elektronikom .....	131
<b>Slika 12.2</b> Stabilizatori i punjač .....	131
<b>Slika 12.3</b> Analogni deo uređaja .....	132
<b>Slika 12.4</b> Akcelerometar.....	132
<b>Slika 12.5</b> Displej .....	133

## SPISAK TABELA

<b>Tabela 1</b> Pregled poznatijih brendova bežičnih uređaja (karakteristike mogu varirati između različitih verzija i zbog modularnog dizajna) .....	8
<b>Tabela 2</b> Pregled poznatijih brendova integrisanih akcelerometara (parametri mogu varirati između različitih verzija i u različitim modovima rada) .....	11
<b>Tabela 3</b> Kompresioni odnosi i vremena kompresije u C-u i assembleru, za različite tipove signala; $f_s$ označava frekvenciju samplinga (odabiranja).....	61
<b>Tabela 4</b> Rezultati eksperimenata na tri različite podloge.....	73
<b>Tabela 5</b> Rezultati bazične kalibracije procesora (p-p označava <i>peak-to-peak</i> ).....	90
<b>Tabela 6</b> Koeficijenti temperaturske zavisnosti frekvencije procesora .....	91
<b>Tabela 7</b> Srednje vrednosti i standardne devijacije razlika u vremenima detekcija prozivnog signala i korigovanih vremenskih žigova, kao i proračunate devijacije odnosa frekvencija tajmera za parove senzorskih uređaja .....	95
<b>Tabela 8</b> Originalni kompresioni odnosi (KO) i dobiti upotrebom TDE-DPCM preprocesiranja u odnosu na klasično DPCM preprocesiranje .....	118

## PREDGOVOR

Doktorska teza pod naslovom „Razvoj i analiza senzorskog sistema za merenje vibracija građevinskih objekata” je rezultat četvorogodišnjeg (2009-2013) praktičnog i istraživačkog rada autora u Laboratoriji za elektroniku Građevinskog fakultetu u Beogradu. 2009/2010. razvijena je osnovna arhitektura hardvera i softvera mernog sistema, a u narednom periodu vršene su dorade istih kao i višestruka eksperimentalna ispitivanja. Evaluacija sistema, osim u laboratoriji fakulteta i na terenu, izvršena je i u saradnji sa kolegama iz Vojnotehničkog instituta (VTI), Direkcije za mere i dragocene metale (DMDM), i Auto-moto saveza Srbije - centra za motorna vozila (AMSS-CMV), čija je merna oprema značajno doprinela radu.

Sledeći objavljeni naučni radovi opisuju metode i rezultate koji su doprinos ove doktorske teze:

### ***A. Radovi u međunarodnim časopisima***

- 1) **Malović M.**, Brajović Lj., Mišković Z., Šekara T., „Simultaneity analysis in a wireless sensor network“, Metrology & Measurement Systems, 22 (2015) 2, DOI: 10.1515/mms-2015-0022
- 2) Brajović Lj., **Malović M.**, Popović Z., Lazarević L., „Wireless System for Sleeper Vibrations Measurement“, Communications - Scientific Letters of the University of Žilina, 16 (2014) 4, pp. 21-26, ISSN 1335-4205

### ***B. Radovi u domaćim časopisima***

- 1) **Malović M.**, Brajović Lj., Mišković Z., Todorović G., „Vibration measurements using a wireless sensors network“, Technics - Our Civil Engineering, 68 (2013) special edition, UDC:62(062.2)(497.1), ISSN 0040-2176, pp. 19-26, UDC:624.042.3:693.547.7
- 2) **Malović M.**, Brajović Lj., Mišković Z., Todorović G., „Merenje vibracija mrežom bežičnih senzora“, Tehnika - Naše Građevinarstvo, 66 (2012) 6, str. 883-889
- 3) **Malović M.**, Brajović Lj., Mišković Z., „Primena bežičnih senzorskih mreža u monitoringu građevinskih konstrukcija“, Građevinski Kalendar, 42 (2010), str. 201-232

### **C. Rad u zborniku sa međunarodne konferencije**

- 1) **Malović M.**, Brajović Lj., Radić V., Lazarević L., Popović Z., „Merenje vibracija pružnih pragova i određivanje njihovog dinamičkog ugiba i brzine vozova“, Zbornik radova XIII međunarodnog naučno-stručnog simpozijuma Infotech, pp. 13-17, Jahorina, Bosna i Hercegovina, 19-21. mart 2014, ISBN 978-99955-763-3-2

### **D. Radovi u zbornicima sa nacionalnih konferencija**

- 1) **Malović M.**, Brajović Lj., Mišković Z., Todorović G., Cvetković J., Banović N., „Analiza vremenske sinhronizovanosti bežičnih senzorskih uređaja za merenje vibracija u građevinarstvu“, Zbornik radova Kongresa Metrologa 2013, Bor, 16-18(19). oktobar (sekcija 1, rad 5), ISBN 978-86-7287-040-4
- 2) **Malović M.**, Brajović Lj., Mišković Z., „Mreža bežičnih senzora za merenje vibracija na građevinskim objektima“, Zbornik radova Kongresa Metrologa 2011, Kladovo, 17-19. oktobar 2011, str. 39-46

Doktorska teza se sastoji iz 10 poglavlja, literature, 2 priloga i biografije autora.

U prvom poglavlju teze je dat opis tehnika nadzora stanja objekata, kao i bežičnih senzorskih mreža koje se primenjuju u te svrhe. Opisana je modalna analiza vibracija građevinskih objekata. Dat je prikaz nekih uređaja renomiranih proizvođača koji se koriste u sličnim sistemima u svetu.

U drugom poglavlju opisani su matematički osnovi teorije modalne analize.

U trećem poglavlju je dat opis razvijenog mernog sistema. Opisani su delovi hardvera senzorskog uređaja (kao i displej i solarni panel koji se opciono mogu povezati). Takođe je opisan originalni embedded RTOS uređaja, i programski paket koji je razvijen za centralnu stanicu sistema, koju predstavlja PC računar. Algoritam rada celog sistema je prikazan.

U četvrtom poglavlju je dat prikaz postojećih tehnika za kompresiju podataka bez gubitaka, kao i tehnike detekcije periodičnosti signala. Obrazloženo je tzv. TDE-DPCM preprocesiranje (od *Time Delay Estimation* i *Differential Pulse Code Modulation*) i zatim izvođenje entropijskog kodiranja Hafmanovom metodom. Parametri ovog procesa su analizirani. Kompresija podataka je od velikog značaja za uštedu energije prilikom transmisije podataka u bežičnoj mreži.

U petom poglavlju opisani su eksperimenti izvršeni u cilju ispitivanja osnovnih karakteristika sistema. Domet modema je ispitivan u uslovima urbanog šuma. Kros-

aksijalna (ili poprečna) osetljivost akcelerometara je ispitivana metodom tilt-testa sa 24 pozicije. Ispitivanje funkcije solarnog panela za dopunjavanje je vršeno u laboratorijskim uslovima.

U šestom poglavlju dat je pregled tehnika sinhronizacije u postojećim kompjuterskim i bežičnim senzorskim mrežama i opisani su problemi sinhronizacije u mrežama sličnim realizovanoj, te je izvršena analiza kritičnih faktora. Sinhronizacija je bitna u merenjima u građevini zbog poređenja tačnih faza pojedinačnih harmonika signala u udaljenim tačkama.

U sedmom poglavlju opisano je eksperimentalno određivanje parametara bežične mreže, koji utiču na vremensku sinhronizaciju. Pod ovim se podrazumevaju parametri propagacije signala kroz radio modeme i parametri koji opisuju neodređenost frekvencije kvarnih kristala koji služe kao osnova merenja vremena na senzorskim uređajima (i u elektronici uopšte). Algoritmi su razvijeni, implementirani, i eksperimentalno verifikovani, u cilju poboljšanja simultanosti podataka. Takođe su prikazani eksperimenti sa mehaničkom pobudom sistema.

U osmom poglavlju opisana je upotreba mernog sistema u merenjima na pruži. Ova merenja, bazirana na vibracijama pragova, impliciraju mogućnost primene sistema za određivanje brzine vozova i za određivanje dinamičkog ugiba (pomeraja) pragova.

U devetom poglavlju dato je poređenje rezultata četiri seta eksperimenata sa i bez primene TDE-DPCM metode kompresije. Rezultati potvrđuju opravdanost primene metoda kompresije u većini slučajeva, i date su preporuke za unapređenje algoritma, tj. izbora parametara koji vode optimalnoj kompresiji za date slučajeve.

Deseto poglavlje predstavlja zaključak rada.

U prilogima su prikazane šeme hardvera uređaja i dati izvodi iz listinga softvera senzorskog uređaja, displeja, i programa za MS Windows (centralnu stanicu sistema).

# 1. UVOD

## 1.1. Monitoring građevinskih objekata

Unapređenje održavanja, eksploatacije i sigurnosti kapitalnih građevinskih objekata (visoke spratnosti, velikih raspona i značajne infrastrukturne važnosti, kao što su mostovi, brane, tornjevi ili tuneli) zahteva permanento ili povremeno osmatranje njihovog stanja i ponašanja. Razvoj infrastrukture i zahtevi za racionalnim održavanjem, sa povećanjem pouzdanosti ovakvih objekata, nameću neizbežnu potrebu za primenom raznovrsnih senzorskih sistema za monitoring. Primenom metoda monitoringa u fazi građenja dolazi do značajne uštede i povećavanja mogućnosti izvođenja komplikovanih tehnoloških operacija, uz smanjenje rizika od nesreća i raznih ekscenčnih pojava [1,2].

Sam monitoring obuhvata osmatranje naponsko-deformacijskih stanja, geometrije, meteoroloških i drugih parametara, koji imaju uticaja na konstrukciju ili mogu da pokažu u kom stanju se konstrukcija nalazi. Projekat monitoringa je multidisciplinaran, obuhvata građevinsku i geodetsku problematiku, kao i aspekte akvizicije, prenosa, skladištenja i analize podataka o fizičkim veličinama koje su predmet ispitivanja [3,4].

Istorijski gledano, začeci primene konstrukcijskog monitoringa mogu se naći u mašinskoj industriji, a prvenstveno u vazduhoplovnim i kosmičkim tehnologijama. Monitoring konstrukcija razvijanih u ovim oblastima bio je imperativ za realizaciju projekata. Tu se mogu uočiti i prve primene monitoringa građevinskih konstrukcija, koji je bio ograničen na relativno mali broj tipova objekata, u prvom redu brana, gde je u primeni od šezdesetih godina dvadesetog veka. Ograničena i selektivna primena bila je uslovljena nesavršenošću senzorske tehnologije (u smislu merne tačnosti, praktične pouzdanosti, i visoke cene), ograničenim mogućnostima skladištenja i prenosa podataka (informacione tehnologije), i pouzdanošću algoritama za identifikaciju konstrukcijskih karakteristika i mogućnosti predviđanja konstrukcijskog odgovora na pobudu (uključujući nedovoljne mogućnosti efikasnog procesiranja veće količine podataka) [5,6].

Monitoring građevinskih konstrukcija doživljava nagli uspon početkom dvadesetprvog veka. Razlozi su razvoj mikroelektronike, uglavnom bežičnih tehnologija



i mikro-elektro-mehaničkih (MEMS) senzora, i računarske tehnologije, pošto je praktično svaki akvizicioni sistem baziran na personalnom računaru ogromne procesorske moći i praktično neograničenog kapaciteta za skladištenje podataka. Pored novih mogućnosti koje savremena senzorska i informaciona tehnologija nudi, razvoj monitoringa građevinskih konstrukcija je uslovljen i povećanim potrebama usled sve smelijih konstrukcijskih rešenja, koja zahtevaju monitoring i tokom izgradnje i tokom eksploatacije, i povećanim zahtevima u smislu kvaliteta, ekonomičnosti izgradnje, i sigurnosnih propisa. Takođe, starost mnogih kapitalnih infrastrukturnih objekata raste, a njihova zamena nije svrsishodna iz praktičnih i ekonomskih razloga, pa je potrebno povremeno ili stalno praćenje stanja radi blagovremenog planiranja sanacija. Konačno, povećani su zahtevi za gradnjom na lokacijama nepovoljnim u geomehaničkom pogledu, tako da sistemi oslanjanja građevinskih objekata (a posebno šipovi i drugi oblici dubokog fundiranja) zahtevaju proveru sleganja i nosivosti terena. Sve ovo uslovalo je i upotrebu nove kovanice *Structural Health Monitoring* (SHM) u naučnim radovima, što se najčešće prevodi kao „nadzor stanja objekata“ [7].

U užem smislu, SHM podrazumeva posmatranje sistema snimanjem podataka dobijenih u stvarnom vremenu sa matrice senzora (*sensor array*), i zatim identifikaciju specifičnih informacija (prepoznavanje oblika signala) koje upućuju na neki poremećaj ili oštećenje objekta, a što se često čini ispitivanjem odnosa tj. korelacija između signala snimljenih u različitim tačkama. U širem smislu, pod SHM se podrazumeva bilo kakva vrsta merenja na mašinskim, građevinskim, i drugim objektima, čiji je cilj određivanje kvaliteta odnosno stanja posmatranog elementa. Merene fizičke veličine su najčešće mehaničke (dinamički ili statički pomeraj ili istežanje, brzina, ubrzanje, sila), a mogu uključivati i temperaturu, razne meteorološke parametre, osvetljenost, buku, i dr. Metode merenja, tj. primenjeni senzori mogu varirati, a u novije vreme sve češće se koriste MEMS senzori u vidu čipova na minijaturizovanim integrisanim pločama (čije je pojeftinjenje i uslovalo razvoj ove grane nauke) [8].

## 1.2. Bežične senzorske mreže

Mreže bežičnih senzora (*wireless sensor networks* ili WSN) u ekonomskoj kategoriji danas se koriste u raznim oblastima nauke i tehnike i svakodnevnog života, kao što su meteorologija, medicina (daljinski nadzor pacijenata), saobraćaj (od statistike saobraćaja do merenja zagađenja), zaštita životne sredine (praćenje ugroženih vrsta životinja), seizmologija, industrijska i kućna automatika, itd. Građevinska struka je posebno pogodna za primenu bežičnih tehnologija, jer je lakše, i najčešće jeftinije, postaviti mrežu bežičnih senzora nego žičani sistem na građevinske objekte [9,10].

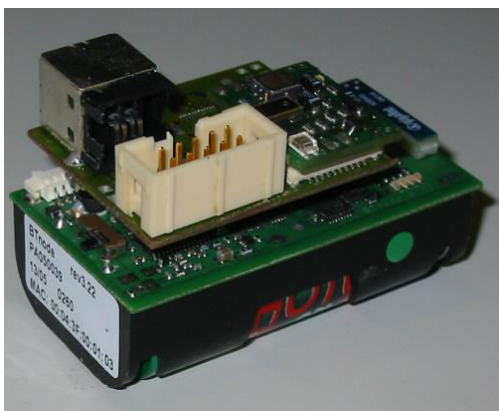
Prilikom izvođenja SHM na kapitalnim objektima, praktično je mandatorno da se koriste bežične tehnologije prenosa podataka. Autonomni bežični senzorski uređaji imaju velike prednosti u odnosu na klasične u merenjima u građevinskoj struci. Oni pojednostavljaju merenja i čine ih jeftinijim. Izbegava se skupo i nepraktično razvlačenje velike količine kablova po objektima velikih dimenzija. Merenja postaju fleksibilnija jer se senzorski uređaji (*wireless sensor nodes*) mogu lako pomerati, ceo sistem se lako prenosi sa jedne lokacije na drugu, i logička topologija mreže se može konfigurisati bez fizičke intervencije. Takođe raste mogućnost postavljanja senzora na teško dostupna mesta. Glavne mane bežičnih sistema su niža brzina prenosa podataka, osetljivost na prirodne i veštačke elektromagnetske smetnje, i nedovoljna sigurnost, tj. podložnost napadu sa strane zbog lakog pristupa medijumu [11].

Senzorski uređaji se često, a posebno u aplikacijama u građevinskoj struci, postavljaju na otvorenom prostoru i moraju odoleti nepovoljnim uslovima u datoj sredini. Moraju se postaviti u vodootporne kutije i funkcionisati u različitim vremenskim uslovima (na niskim i visokim temperaturama i pri visokoj vlažnosti). Takođe moraju se uzeti u obzir mnogobrojne moguće elektromagnetske smetnje prirodnog i veštačkog porekla. Izbor parametara komunikacije (noseća frekvencija, tip modulacije, i bodna brzina) zavisi između ostalog i od očekivanih smetnji na datom mestu. Difrakcija u zatvorenim prostorima, kao i u slučajevima kada se uređaji ne nalaze u liniji optičke vidljivosti, takođe predstavlja važan faktor pri projektovanju mreže bežičnih senzora [12].

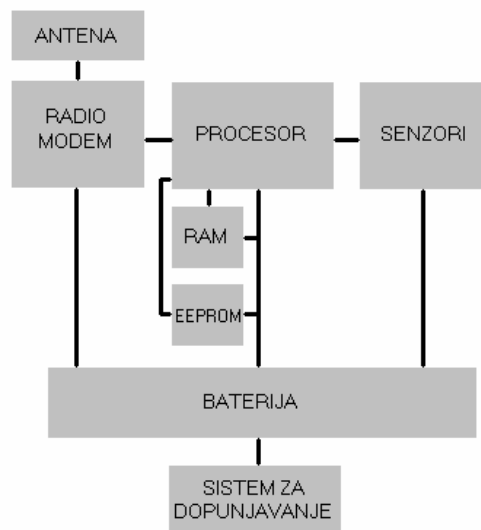
## 1.2.1. Senzorski uređaji

Bežične senzorske mreže su sastavljene od senzorskih uređaja ili čvorova (*nodes*). Principijelni delovi svakog senzorskog uređaja su procesor, jedinica za komunikaciju, antena, memorijske jedinice (RAM/EEPROM čipovi), senzori, baterijsko napajanje, i sistem za dopunjavanje energije (tzv. *energy harvesting*), koji nije obavezan sastavni deo. Uobičajena blok šema uređaja je data na sl. 1.1.

Procesorska jedinica je najčešće mikrokontroler, a mogu se koristiti i drugi tipovi kao što su *digital signal processor* (DSP), ili *field programmable gate array* (FPGA). Jedinica za komunikaciju najčešće koristi radio talase (*radio frequency* ili RF; mada se u opštem slučaju mogu koristiti i mikrofrekventni, infracrveni, i laserski). RF komunikaciona jedinica može biti samostalan modemska uređaj ili ugrađeni transiver. Antene koje se povezuju sa njima su najrazličitijih konstrukcija, od minijaturnih *stick* antena do velikih antena sa linijskim (*Yagi-Uda*) ili ravanskim (omnidirekcione) usmeravanjem talasa. Za baterijsko napajanje koriste se različiti tipovi baterija kao što su litijum-jonske ili niki-metal-hidridne (NiMH), ukupnog napona najčešće 3-5 V. Obzirom da različiti uređaji imaju različite režime rada i različite energetske zahteve,



Slika 1.2 BTnode, ETH Cirih



Slika 1.1 Blok šema bežičnog uređaja

tipično vreme rada uređaja pre nego što je potrebno izvršiti zamenu baterija varira. *Energy harvesting* sistemi se često koriste za dopunjavanje baterija energijom iz okoline [13]. Najčešći princip rada ovakvih sistema je konverzija sunčeve energije putem fotonaponskih solarnih ćelija, ali koriste se i drugi tipovi kao što su piezoelektrični (koji koriste energiju vibracija), termoparovi, itd. Često ovakvi sistemi mogu da obezbede

dovoljno energije za neograničenu upotrebu senzorskog uređaja.

Na tržištu postoji veliki broj standardnih tipova bežičnih uređaja koji su modularno dizajnirani i mogu biti opremljeni različitim sensorima. Mnogi od njih su razvijeni u renomiranim naučnim ustanovama, kao što su univerzitet Berkli, Kalifornija, ili institut ETH, Ciri, Švajcarska. U tabeli 1 dat je pregled nekih poznatijih tipova.

Većina uređaja je programabilna i razvoj softvera se najčešće vrši u *embedded* C-u. Ređe se koriste viši programski jezici kao objektni C ili Java [14].

**Tabela 1** Pregled poznatijih brendova bežičnih uređaja (karakteristike mogu varirati između različitih verzija i zbog modularnog dizajna)

naziv	proizvođač	procesor, memorija, i transsiver
BTnode	ETH Zürich	ATmega128 (8 MHz) 64+180K RAM, 128K FLASH, 4K EEPROM CC1000 (433-915 MHz)
Imote2	Intel Crossbow	PXA271 (13-416 MHz) 256K SRAM, 32M SDRAM, 32M FLASH CC2420 (IEEE 802.15.4)
Iris	Memsic	ATmega1281 (8 MHz) 8K RAM, 128K FLASH AT86RF230 (IEEE 802.15.4)
Mica	Berkeley univ. Crossbow	ATmega128L (8 MHz) 4K RAM, 128K FLASH Mica2: Chipcon 868/916 MHz; MicaZ: CC2420
MITes	MIT	nRF24E1 (16 MHz) 4K RAM, 512K FLASH nRF24E1 sadrži 2.4GHz 100 kbps transsiver
Shimmer	Shimmer	MSP430F1611 (8 MHz) 10K RAM, 48K FLASH, 2G memorijska kartica Shimmer SR7 (IEEE 802.15.4)
SunSPOT	Sun	ARM920T (180 MHz) 512K RAM, 4M FLASH IEEE 802.15.4
TelosB, Tmote Sky	Berkeley univ. Memsic MoteIV, Sentilla	MSP430 (8 MHz) 10K RAM, 48K FLASH CC2420 (IEEE 802.15.4)
TinyNode	Shockfish	MSP430F1611 (8 MHz) 8K RAM, 512K FLASH Semtech SX1211 (868 MHz)



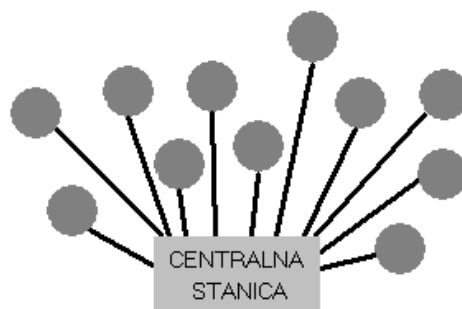
Slika 1.3 SunSPOT

Vidimo da se SunSPOT ističe velikim procesorskim i memorijskim kapacitetom, ali ima relativno veliku snagu što znači da ovaj uređaj nije pogodan za aplikacije gde je štednja energije prioritet. Uređaji uglavnom koriste 2,4 GHz noseću frekvenciju sa bodnim brzinama do 250 kbps (IEEE 802.15.4 standard). Manje brzine prenosa na nižim nosećim frekvencijama mogu biti pogodne kada treba ostvariti komunikaciju u prisustvu prepreka i u zatvorenom prostoru.

Neki uređaji su prikazani na sl. 1.2 i sl. 1.3.

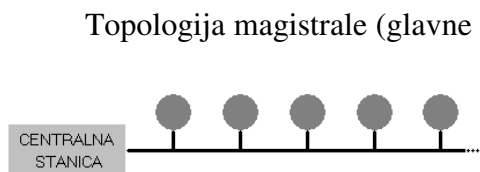
## 1.2.2. Topologija mreže

Pod topologijom mreže podrazumeva se raspored uređaja i njihove međusobne veze [15]. Logička topologija definiše logički plan mreže, tj. zadaje načine i putanje komunikacije. Fizička topologija definiše fizički plan mreže, tj. raspored uređaja i njihovih veza u prostoru. Logička i fizička topologija nisu neophodno identične, ali iz razumljivih razloga projektovanje logičke topologije je tesno povezano sa fizičkom topologijom. Tri najčešće korišćena tipa topologije kod bežičnih mreža su topologija zvezde, magistrale, i mrežasta topologija [16].



Slika 1.4 Topologija zvezde

Topologija zvezde znači da je svaki uređaj spojen sa centralnom stanicom preko odvojenog medijuma (žice, ili frekvencije u slučaju bežičnog prenosa). Primer je dat na sl. 1.4. Povećana je potreba za procesorskom moći centralne stanice, jer ona mora da komunicira sa više uređaja istovremeno. Troškovi su veliki u slučaju žičane mreže. Protokol komunikacije je jednostavan.



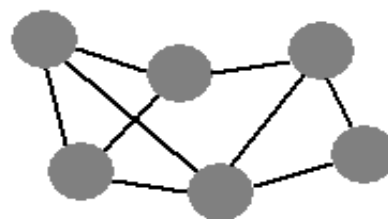
Slika 1.5 Topologija magistrale

Topologija magistrale (glavne linije odnosno busa; primer na sl. 1.5) znači da su svi uređaji spojeni sa centralnom stanicom putem zajedničke magistrale (medijuma). Centralna stanica komunicira samo sa jednim uređajem u jednom trenutku. Povećana je

potreba za pouzdanim protokolom. Troškovi ožičavanja u slučaju žičane mreže su smanjeni.

Mrežasta topologija (*web* ili *mesh*) znači da je svaki uređaj spojen sa bar dva druga uređaja (a možda i svaki sa svakim), uključujući i centralnu stanicu (ako takva postoji). Primer je dat na sl. 1.6. Internet radi na ovaj način. Mnoge bežične mreže inkorporiraju u određenoj meri ovaj princip rada zbog dinamičke prirode skupa ostvarivih veza.

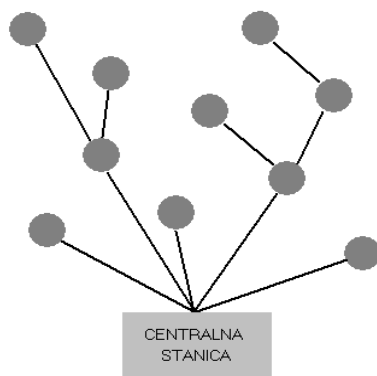
U praksi se koriste i druge podele, npr. na tačka do tačke (*point-to-point*) i višerazvodnu (*multi-drop*). *Point-to-point* topologija znači da nema grananja veza (ne računajući centralnu stanicu). Primeri ove topologije su zvezda i linija (mreža u kojoj je centralna stanica spojena sa prvim uređajem, prvi sa drugim, drugi sa trećim, itd). *Multi-drop* mreža je ona u kojoj svaki uređaj ima više veza sa drugim uređajima, dakle *web* topologija spada u *multi-drop* kategoriju.



Slika 1.6 Mrežasta topologija

Podela na *single-hop* i *multi-hop* mreže govori nam da li se informacija prenosi direktno iz svakog uređaja do centralne jedinice, ili se može prenositi iz više koraka preko različitih uređaja dok ne stigne na odredište. Usled ograničenog dometa portabl radio uređaja, kod bežičnih mreža česta je upotreba multi-hopa.

Topologija mreže se često ne može striktno podvesti pod okvire gornjih podela. Npr, čest način realizacije mreže je razgranata (ili „distribuirana“) zvezda, gde neki



Slika 1.7 Realan primer složene topologije multi-hop mreže

uređaji u osnovnoj zvezdi predstavljaju čvorišta (habove) sekundarnih zvezda. To je primer multi-hop mreže. Na sl. 1.7 je dat primer tzv. razgranate zvezde.

Optimalna konfiguracija mreže zavisi od osobina sredine i funkcije i položaja uređaja. Za razliku od žičanih mreža, kod bežičnih mreža ne postoje materijalne veze između uređaja, i ovo im daje prednost da se topologija može konfigurisati u toku rada (u zadatim

granicama). Ovo je naročito bitno kod mreža sa velikim brojem uređaja i kod mreža

koje su dizajnirane da budu mobilne. Konfigurisanje mreže može da se vrši jednom, prilikom postavljanja, ili periodično, u toku korišćenja.

### 1.2.3. Senzori

Senzor može biti bilo koji pretvarač sa izlazom u vidu analogne ili digitalne električne veličine. Od pretvarača mehaničkih veličina u električne, u bežičnim mrežama se najčešće koriste akcelerometri, merne trake i piezoelektrični senzori, a od ostalih termometri, detektori svetlosti i zvuka, detektori hemijskih supstanci, itd. Ovi senzori su uglavnom digitalni, a ređe mogu biti i analogni pri čemu se vrši A/D konverzija njihovog izlaznog signala. Većina modernih mikrokontrolera ima ugrađene A/D konvertore za ovu namenu. Iako analogni senzori imaju značajno bolju preciznost, mana im je veća cena, veći radni napon, i veća potrošnja energije (ograničena količina energije kojom tipičan bežični uređaj raspolaže je značajan faktor pri izboru komponenti).

**Tabela 2** Pregled poznatijih tipova integrisanih akcelerometara (parametri mogu varirati između različitih verzija i u različitim modovima rada)

naziv	proizvođač	osa	opseg $a$ [ $\pm g$ ]	opseg $f$ [Hz]	izlaz	napajanje	komentar
ADXL203	Analog Devices	2	1,7 - 18	2500	analogni signal	700 $\mu A$ (5V)	klasični model
ADXL362	Analog Devices	3	2 - 8	200	12-bitni SPI	2,5 $\mu A$ (2V)	niska potrošnja
KXP84	Kionix	3	1,5 - 6	3500	12-bitni I <sup>2</sup> C, SPI	1 mA (3V)	veliki opseg frekvencija
LIS331DL	STMicro-electronics	3	2 - 8	200	8-bitni I <sup>2</sup> C, SPI	400 $\mu A$ (3V)	malih dimenzija
MMA726x	Freescale Semiconductor	3	1,5 - 6	350	analogni signal	500 $\mu A$ (3V)	klasični model
MX2125	Memsic	2	3	160	PWM signal	3 mA (5V)	velika preciznost
SMB26x	Bosch	1-2	35 - 100	400	analogni signal	5 mA (5V)	veliki opseg ubrzanja

MEMS akcelerometri koji se koriste u merenjima u građevini su najčešće troosni [17]. Njihova rezolucija iznosi 8 do 12 bita, imaju opseg merenja reda veličine  $\pm g$  ili

$\pm 10g$ , a maksimalne frekvencije (propusni opseg, odnosno *bandwidth*) su od desetak do više stotina Hz. Napajaju se naponima od 3 do 5 V, i spajaju se sa procesorom putem I2C ili SPI interfejsa. Loša osobina im je visoka unakrsna osna osetljivost (tipično 3-5%), za red veličine veća nego kod analognih senzora. Ova se ipak može dobrim delom kompenzovati softverski. Među tipične predstavnike spadaju modeli iz familija ADXL (Analog Devices), LIS (STMicroelectronics), MX (Memsic), SMB (Bosch), KXP (Kionix), i drugi [18]. Spisak nekih često korišćenih modela sa karakteristikama dat je u tabeli 2.

#### 1.2.4. Softver

U sistemima bežičnih senzora mogu se koristiti gotovi operativni sistemi (*embedded real time operating systems* ili skraćeno embedded RTOS) koji su uglavnom razvijeni na raznim univerzitetima, i *open-source*, odnosno otvoreni za prepravke i nadgradnju, ili se mogu razviti novi, što povećava fleksibilnost, a može biti od značaja ako je aplikacija nestandardna. Među poznatije operativne sisteme koji se koriste u mrežama bežičnih senzora spadaju TinyOS [19], MANTIS OS [20], SOS [21], EPOS [22], Contiki [23], Pixie [24], LiteOS [25], nano-RK [26] i drugi. Postoje i specijalni softverski paketi koji se ne mogu klasifikovati ni kao programski jezici ni kao operativni sistemi, a pogodni su za upotrebu u WSN, kao što je Eon [27].

Većina procesora može se programirati putem serijske veze sa PC računarom, pri čemu se koristi neko *integrated development environment* (IDE) okruženje. Najčešći programski jezik koji se koristi za razvoj embedded RTOS-a je C niskog nivoa (embedded C), opciono u kombinaciji sa assemblerom koji se koristi za delikatne operacije (kao što je rukovanje osetljivim hardverskim interaptima).

Softver na uređaju (embedded RTOS) osim osnovnog upravljanja hardverom može vršiti određenu obradu signala na licu mesta. Ovo se radi da bi se uštedelo prilikom prenosa, jer je komunikacija najveći potrošač energije u bežičnim mrežama. Softver na centralnoj stanici bežične mreže (najčešće PC računaru) ima veću slobodu jer nije ograničen vremenski i energetska.

Ovaj softver može vršiti i korekciju snimljenih podataka usled poznatih statičkih i dinamičkih karakteristika senzora. Naime, nijedan senzor ne preslikava ulaznu u izlaznu veličinu savršeno. Pod statičkom korekcijom podrazumeva se korigovanje zbog



nesavršenosti kao što su npr. nelinearnost ili unakrsna osna osetljivost kod višeosnih akcelerometara. Iako uglavnom relativno stabilne tokom vremena, ove nesavršenosti se moraju eksperimentalno ispitati za svaki pojedinačni senzor. Efekti inercije i prigušenja (koji potiču od pokretne mase i trenja kod senzora sa mehaničkim principom rada, ili induktivnosti i kapacitivnosti kod električnih senzora) čine da senzori ne odgovaraju na pobudu trenutno niti signalom istog oblika u vremenu, pa se zbog toga vrši dinamička korekcija (dinamička korekcija, za razliku od statičke, uključuje prethodne odmerke signala u proračun). Da bi bilo moguće izvršiti dinamičku korekciju, potrebno je modelovati senzor odgovarajućom jednačinom. Najčešće se koristi diferencijalna jednačina drugog reda koja zadovoljavajuće aproksimira ponašanje većine realnih senzora (sistem drugog reda) [28]. Zatim je neophodno odrediti koeficijente u jednačini prenosa, što se čini kalibracijom senzora (ispitivanjem njegovog odziva na impulsnu ili odskočnu pobudu). U slučaju merenja mehaničkih veličina treba voditi računa i o promeni karakteristika celog mernog sistema prilikom stavljanja senzora u kućište i u zavisnosti od načina na koji se sve spaja sa objektom koji se ispituje.

### 1.3. Modovi oscilovanja građevinskih struktura

Sve mehaničke strukture su podložne skladištenju energije prilikom dejstva spoljnih sila. Energija koja dolazi od spoljnih sila (kao što je npr. vetar u slučaju tornjeva ili mostova) se pretvara u potencijalnu energiju putem elastičnih sila između čestica materijala, bilo da se radi o linearnoj promeni dimenzija (skupljanje i širenje) ili



Slika 1.8 Tacoma Narrows most

uvijanju (torzija). Ova energija se u nekim slučajevima „tiho“ disipira, bez izazivanja ikakvih vidljivih efekata. Međutim, u proces ove disipacije, po prirodi prigušeno-oscilatornog karaktera, su uključene vremenske konstante zavisne od vrste i dimenzija objekta. Čak i minimalne periodičnosti u dejstvu spoljašnjih sila (npr. postojanje periodičnih vrtloženja u slučaju vetra) mogu dovesti do

nepredvidljivih efekata na inače statički stabilnim strukturama, ako se periodi poklope sa ovim vremenskim konstantama. U tom slučaju govorimo o rezonantnim efektima.

Rezonantne efekte na kompleksnim objektima kao što su kapitalne građevinske strukture je teško analitički proračunati sa velikom preciznošću. U svrhu izbegavanja katastrofa, kao što je poznato rušenje Tacoma Narrows mosta [29], prikazano na sl. 1.8, koriste se, osim teorijskih, dve osnovne empirijske tehnike. Prvo, pravilni elementi koji imaju jasno definisane rezonantne frekvencije, kao što su kablovi, stubovi, i sl. projektuju se tako da im se rezonantne učestanosti ne poklapaju. Drugo, prave se makete objekata i ispituju u vazдушnim tunelima. Ovo je uobičajeno i prilikom ispitivanja novih modela ne samo aviona već i automobila i drugih tipova prevoznih sredstava (pri čemu se ispituju i makete i prototipovi) [30].

Ipak, empirijske metode nisu jedino oružje u rukama današnjim inženjera. Metoda konačnih elemenata (*finite element modeling* ili FEM), koja predstavlja vrstu numeričke analize, je sa razvojem kompjuterske tehnike (bržeg hardvera, veće memorije, i specijalizovanih programskih paketa) omogućila vrlo tačne simulacije koje su pre nekoliko decenija bile nezamislive [31,32].

## **1.4. Polazne pretpostavke i cilj istraživanja**

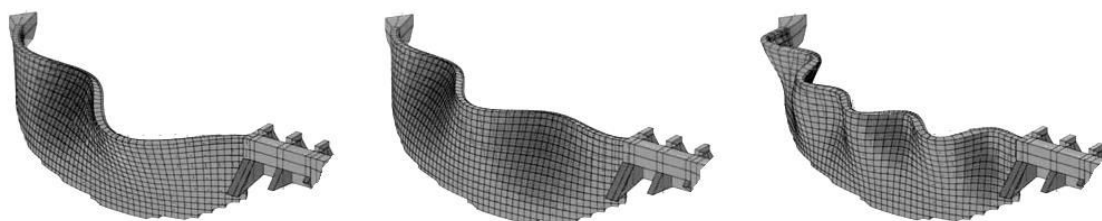
Može se konstatovati da se osim klasičnih mernih sistema sve više koriste bežične senzorske mreže, pogotovo u građevinskoj struci. Postoji potreba za domaćim senzorskim sistemom za merenje vibracija u cilju monitoringa postojećih i novih građevinskih konstrukcija. Razlozi su sve smelija rešenja koja zahtevaju praćenje i za vreme gradnje i tokom eksploatacije, inovacije u oblasti tehnologije građenja, povećani zahtevi za gradnjom kapitalnih objekata po što nižoj ceni, novi propisi u vezi sa sigurnošću i održavanjem građevinskih objekata, upotreba novih materijala o čijem ponašanju u toku dužeg vremenskog perioda nema dovoljno podataka, starenje infrastrukturnih objekata za čiju zamenu nema sredstava, kao i objekata od istorijskog značaja, zahtevi za izgradnju na nepovoljnim i slabo pristupačnim terenima, tektonska pomeranja koja ugrožavaju kapitalne objekte, itd.

Ciljevi istraživanja su: razvoj originalnog senzorskog uređaja visoke fleksibilnosti (u smislu arhitekture otvorene za dodavanje različitih digitalnih ili analognih senzora, i mogućnosti modifikacije softvera na najnižem mogućem nivou);

razvoj metoda za korekciju vremenske neodređenosti izazvane propagacijom signala kroz mrežu i kratkoročnim i dugoročnim neodređenostima frekvencije kvarcnih kristala; razvoj metoda za merenje i korekciju visoke kros-aksijalne osetljivosti upotrebljenih akcelerometara bez upotrebe skupe visoko sofisticirane opreme; razvoj novih algoritama kompresije podataka; i dr.

## 2. OSNOVE MODALNE ANALIZE

Termin „modalna analiza“ podrazumeva proces određivanja dinamičkih karakteristika sistema: rezonantnih frekvencija, faktora prigušenja, i oblika modova oscilovanja, i upotrebu ovih podataka da bi se matematički modelovao odziv strukture na pobudu. Modalna analiza je bazirana na činjenici da se vibracioni odziv linearnog, vremenski nepromenljivog sistema, može izraziti kao linearna kombinacija skupa prostih harmonijskih oscilacija koje se nazivaju osnovni ili prirodni modovi. Osnovni modovi vibracija su karakteristični za svaki sistem, i određeni su njegovim fizičkim osobinama (masom, modulom elastičnosti, koeficijentom prigušenja, i prostornom raspodelom istih). Svaki mod oscilacija se odlikuje određenim „oblikom“, frekvencijom, i prigušenjem. Pod oblikom se podrazumeva raspored amplituda u prostoru. Ovaj se često grafički prikazuje na dijagramima na kojima su amplitude preuveličane radi preglednosti. Jedan takav dijagram dat je na sl. 2.1, gde vidimo registrovane modove oscilovanja brane Norsjö u Švedskoj [33]. Intenzitet vibracija svakog od modova zavisi ne samo od karakteristika sistema, već i od oblika pobude.



**Slika 2.1** Neki od registrovanih modova vibracija brane Norsjö u Švedskoj [33]

Modalna analiza obuhvata i teorijske i praktične tehnike. Teorijska modalna analiza je bazirana na fizičkom modelu koji koristi masu i parametre koji opisuju elastičnost i prigušenje. Ovi parametri su najčešće dati u obliku parcijalnih diferencijalnih jednačina. Jednostavan primer je jednačina oscilovanja žice, koja je izvedena na osnovu njene podužne distribucije mase i modula elastičnosti. Rešenja ove jednačine otkrivaju prirodne frekvencije i oblike modova oscilovanja žice koji će se pojaviti prilikom eksterne pobude. Realističniji model kompleksnijih struktura je dat u prostoru stanja i sadrži matrice mase, elastičnosti i prigušenja. Princip superpozicije linearnih dinamičkih sistema omogućava da se ove jednačine transformišu u matrice sopstvenih vrednosti i sopstvenih vektora.

Praktične eksperimentalne tehnike se često nazivaju „modalno testiranje“. Modalno testiranje podrazumeva ustanovljavanje relacija između vibracionih odgovora strukture u različitim tačkama, pri primeni različitih pobudnih frekvencija. Ove veze su često vrlo kompleksne, i upotrebljava se termin *frequency response function* (FRF) da bi se opisale. Merenja se vrše meračima sile (na mestu pobude) i akcelerometrima na mestima gde se snima odziv sistema. Primenjena sila može biti različitih oblika, kao što su harmonijske oscilacije određene frekvencije, kombinacija oscilacija (skup frekvencija), impulsna ili odskočna funkcija, beli šum, itd. Praktičan problem prilikom modalnog testiranja jeste odrediti koliko tačno podataka treba da se snimi (koliko vrsta pobuda, na koliko mesta, da se upotrebi i u koliko tačaka da se snima odziv) da bi se dobio dovoljno realan model sistema.

Teorija matrica ima veliki značaj u modalnoj analizi. Ovo je zato što se modalna analiza uglavnom bazira na proučavanju dinamičkih karakteristika sistema sa više stepeni slobode, koji se oslanja na teoriju matrica. Furijeova analiza je takođe nezaobilazna metoda korišćena prilikom procesiranja signala. Bez FFT-a, modalna analiza je teško zamisliva. Teorija upravljanja, kao i modalna analiza, se bavi identifikacijom prenosne karakteristike sistema. Zato modalna analiza uključuje različite tehnike poznate iz teorije upravljanja, kao što su Laplasova transformacija, Hilbertova transformacija, i analiza vremenskih serija (*time series analysis*).

Teorija vibracija se retko primenjuje na realne strukture sa izuzetkom onih čija se dinamika može tačno opisati korišćenjem konačnog broja parcijalnih diferencijalnih jednačina. Ovo je zato što je teorija obično razvijena za idealizovanu verziju stvarnog problema (dobijenu korišćenjem različitih pretpostavki). Matematičko modelovanje strukture uključuje proces idealizacije. Ovaj proces nije jednoznačno određen. Potreba da se razvije jednostavan i efikasan model je u suprotnosti sa dubinom shvatanja dinamike strukture. Za modalnu analizu koristi se najčešće diskretan model sa konačnim brojem koordinata. Osobine strukture, kao što su oblici modova oscilacija, se reprezentuju korišćenjem konačnog broja tačaka. Model je zato često predstavljen konačnim brojem diferencijalnih jednačina. Posle Furijeove ili Laplasove transformacije, ove jednačine se konvertuju u algebarski oblik, čime se potencira potreba za matričnom algebrom u modalnoj analizi.

Jednačina kretanja sistema sa više stepeni slobode, kada nema spoljne pobude,

već se radi o slobodnim oscilacijama, glasi:

$$[M]\{\ddot{x}\} + [K]\{x\} = \{0\} \quad (2.1)$$

Pretpostavivši oscilatorni odgovor sistema na pobudu, jednačina se prevodi u algebarski oblik:

$$\{x\} = \{X\} \sin(\omega t) \quad (2.2)$$

odnosno:

$$([K] - \omega^2 [M])\{X\} = \{0\} \quad (2.3)$$

Jednačina kretanja sistema sa više stepeni slobode čije pobudne funkcije imaju istu frekvenciju  $\omega$  i nulti fazni pomeraj glasi:

$$[M]\{\ddot{x}\} + [K]\{x\} = \{F\} \sin(\omega t) \quad (2.4)$$

gde vektor  $\{F\}$  sadrži amplitude pobudne sile. Pretpostavka harmonijskog oscilatornog odgovora na pobudu se koristi da se jednačina (2.4) prevede u algebarski oblik:

$$([K] - \omega^2 [M])\{X\} = \{F\} \quad (2.5)$$

Odgovor sistema na eksternu pobudu je u vremenskom domenu dat sa:

$$\{x(t)\} = ([K] - \omega^2 [M])^{-1} \{F\} \sin(\omega t) \quad (2.6)$$

Za razliku od slobodnih vibracija, ovde je vibraciona frekvencija identična frekvenciji pobude.

Parcijalne diferencijalne jednačine jednostavnijih sistema mogu se rešiti analitički. Ovakvi sistemi uključuju žice, štapove, grede i membrane. Za sisteme sa većom geometrijskom kompleksnošću, potrebno je koristiti numeričke metode i predstaviti kontinualne sisteme kao diskretne sa konačnim brojem stepeni slobode, ili kao kombinaciju više prostih kontinualnih sistema.

Jednačine (2.1) do (2.6) važe za slučaj neprigušenih sistema. U slučaju da postoji prigušenje, jednačina (2.1) postaje:

$$[M]\{\ddot{x}\} + [C]\{\dot{x}\} + [K]\{x\} = \{0\} \quad (2.7)$$

Po Rajlijevoj teoriji zvuka, matrica koeficijenata viskoznog prigušenja se može izraziti kao linearna kombinacija matrica mase i elastičnosti [34]:

$$[C] = \alpha[M] + \beta[K] \quad (2.8)$$

iz čega za slučaj slobodnih vibracija sledi:

$$[M]\{\ddot{x}\} + (\alpha[M] + \beta[K])\{\dot{x}\} + [K]\{x\} = \{0\} \quad (2.9)$$

Jednačine slobodnih oscilacija prigušenih sistema su slične jednačinama koje se dobijaju za neprigušene sisteme. U slučaju prigušenja, dolazi do (uglavnom male) promene rezonantnih frekvencija, i do vremenski eksponencijalno opadajućeg intenziteta vibracija. U nastavku su dati neki jednostavniji slučajevi izvođenja jednačina slobodnih vibracija bez prigušenja.

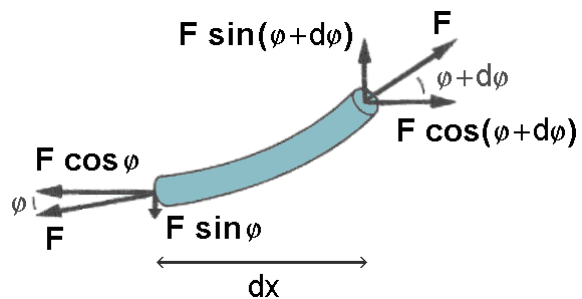
## 2.1. Vibracije žice

Posmatrajmo elastičnu žicu podužne mase  $\mu$  koja se zategnuta silom  $F$ . Ako je transverzalni pomeraj delića žice  $y$  mali, što je u stvarnosti slučaj, ugao  $\varphi$  (sl. 2.2) biće mali, i važiće:

$$\varphi \approx \sin(\varphi) \approx \operatorname{tg}(\varphi) \approx \frac{dy}{dx} \quad (2.10)$$

$$\cos(\varphi) \approx \cos(\varphi + d\varphi) \approx 1 \quad (2.11)$$

Aproksimirajući da je kosinus ugla isti i sa leve i sa desne strane elementa žice  $dx$ , podužna sila je zanemarljiva, ali postoji značajna transverzalna komponenta koja zavisi od razlike nagiba sa leve i desne strane, tako da je:



Slika 2.2 Delić žice dužine  $dx$  izložen istezanju

$$dF_t = F \sin(\varphi + d\varphi) - F \sin(\varphi) = F d(\sin(\varphi)) = F \frac{d\varphi}{dx} dx = F \frac{\partial^2 y}{\partial x^2} dx \quad (2.12)$$

Jednačina žice se izvodi iz drugog Njutnovog zakona:

$$F \frac{\partial^2 y}{\partial x^2} dx = \mu dx \frac{\partial^2 y}{\partial t^2} \quad (2.13)$$

gde član sa desne strane predstavlja masu ( $\mu dx$ ) puta transverzalno ubrzanje, pa je konačno diferencijalna jednačina transverzalne oscilacije žice:

$$\frac{\partial^2 y}{\partial x^2} \frac{F}{\mu} = \frac{\partial^2 y}{\partial t^2} \quad (2.14)$$

Rešavanje ove jednačine se vrši metodom razdvajanja, tako što se pretpostavi da je  $y(x,t)$  funkcija takva da se može izraziti kao proizvod prostornog člana  $Y(x)$  i vremenskog člana  $T(t)$ . U tom slučaju izvod po  $x$  zavisi samo od prostornog člana funkcije, dok izvod po  $t$  zavisi samo od vremenskog člana.

$$\frac{1}{Y} \frac{\partial^2 Y}{\partial x^2} c^2 = \frac{1}{T} \frac{\partial^2 T}{\partial t^2} \quad (2.15)$$

Član  $\sqrt{F/\mu}$  je zamenjen brzinom transverzalnog talasa na žici  $c$ , koja se dobija upoređivanjem jednačine (2.14) sa osnovnom diferencijalnom jednačinom talasa. Pošto je leva strana jednačine (2.15) nezavisna od vremena, a desna strana nezavisna od koordinate, one moraju biti jednake konstanti da bi jednačina uvek važila. Obzirom da je fizički očekivano rešenje oscilatornog oblika, konstanta mora biti negativna zato što ubrzanje mora biti suprotnog znaka od koordinate. Obeležimo ovu konstantu  $-\omega^2$ . Sledi:

$$\frac{1}{Y} \frac{\partial^2 Y}{\partial x^2} c^2 = -\omega^2 \quad \Rightarrow \quad \frac{d^2 Y}{dx^2} + \frac{\omega^2}{c^2} Y = 0 \quad (2.16)$$

$$\frac{1}{T} \frac{\partial^2 T}{\partial t^2} = -\omega^2 \quad \Rightarrow \quad \frac{d^2 T}{dt^2} + \omega^2 T = 0 \quad (2.17)$$

Opšte rešenje jednačina (2.16) i (2.17) je oblika:

$$Y = A \sin(kx) + B \cos(kx) \quad (2.18)$$

$$T = C \sin(\omega t) + D \cos(\omega t) \quad (2.19)$$

gde je  $k=\omega/c$ , a A, B, C i D konstante koje zavise od graničnih uslova. Ako posmatramo žicu dužine  $l$  koja je uklještena između dve tačke, pomeraj u tačkama uklještenja mora biti nula. Ako se početak  $x$  ose postavi u jednu od tih tačaka, konstanta B automatski postaje nula. Rešenje jednačine oscilovanja je onda:

$$y(x,t) = A(C \sin(\omega t) + D \cos(\omega t)) \sin(kx) \quad (2.20)$$

Pošto član  $u$  zagradi predstavlja harmonijsku oscilaciju, izabiranjem odgovarajućeg referentnog vremenskog trenutka on se može svesti na bilo sinus bilo kosinus, tako da se jednačina oscilovanja obično predstavlja kao:

$$y(x,t) = Y_0 \sin(\omega t) \sin(kx) \quad (2.21)$$

gde je  $Y_0$  konstanta. Alternativno, sinusni član može sadržavati proizvoljnu početnu fazu. Za drugu uklještenu tačku ( $x=L$ ) takođe mora da važi:

$$Y(L,t) = 0 \quad (2.22)$$

pa je:

$$\sin \frac{\omega}{c} L = 0 \quad \Rightarrow \quad \frac{\omega}{c} L = z\pi \quad (2.23)$$



gde je  $z$  ceo broj. Sopstvene kružne frekvencije oscilovanja žice su dakle jednake:

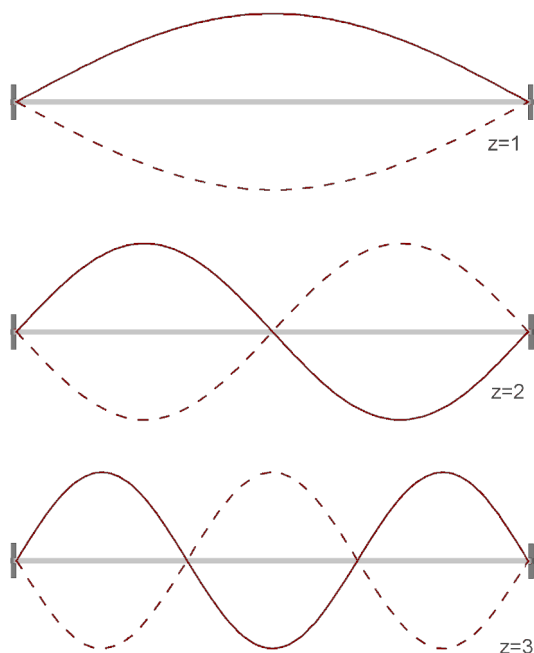
$$\omega_z = z\pi \frac{c}{L} \quad (2.24)$$

što se može prikazati i kao:

$$f_z = \frac{z}{2L} \sqrt{\frac{F}{\mu}} \quad (2.25)$$

gde je  $f_z$  rezonantna frekvencija reda  $z$ . Oblik modova oscilovanja je dat prostornim delom jednačine (2.21):

$$Y(x) = Y_0 \sin \frac{z\pi x}{L} \quad (2.26)$$



Slika 2.3 Modovi oscilovanja žice

gde je  $k$  (talasni broj) izraženo iz jednačine (2.24). Oblici modova u zavisnosti od  $z$  su prikazani na sl. 2.3. Pod normalnim okolnostima, žica će slobodno vibrirati tako da je njena jednačina kretanja suma jednačina za beskonačno modova ( $z=1,2,3,\dots+\infty$ ), koji se razlikuju po amplitudama i početnim fazama u vremenu (ali su čvorovi talasa uvek smešteni u tačke uklještenja). Budući da je energija jednog moda oscilovanja zavisna i od frekvencije i od amplitude, najveće amplitude se u praksi

javljaju za najniže rezonantne frekvencije. Modovi oscilovanja sa visokim  $z$  imaju zanemarljive amplitude.

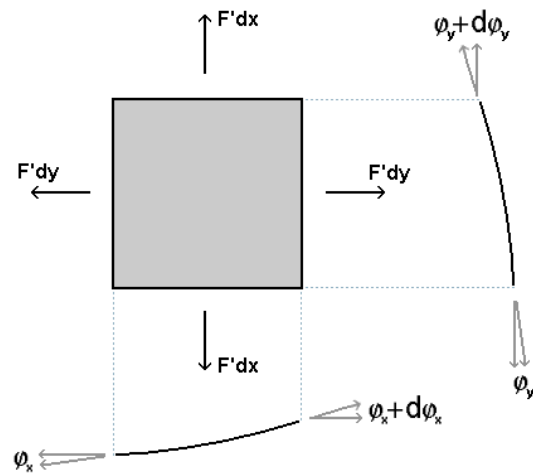
## 2.2. Vibracije pravougaonih membrana

Membrane možemo posmatrati kao dvodimenzioni niz zategnutih žica. U cilju opisivanja transverznog oscilovanja delića membrane, aproksimira se da se delići membrane pomeraju samo u  $z$  pravcu, tako da je pomeraj  $z$  zanemarljiv u odnosu na dimenzije membrane, kao i da je membrana tako zategnuta da je ugao nagiba tangentne ravni u tački koju posmatramo u odnosu na osnovnu  $x$ - $y$  ravan membrane mali. Ove aproksimacije su potpuno analogne onima opisanim u slučaju žice:

$$\varphi_i \approx \sin(\varphi_i) \approx \operatorname{tg}(\varphi_i) \approx \frac{dz}{di} \quad (2.27)$$

$$\cos(\varphi_i) \approx \cos(\varphi_i + d\varphi_i) \approx 1 \quad (2.28)$$

gde indeks  $i$  označava osu  $x$  ili  $y$ , a  $di$  u (2.27) označava element dužine po toj osi (sl. 2.4). Ovo su ustvari jednačine (2.10) i (2.11) napisane za slučaj dve dimenzije. Jednačina (2.12), napisana sa indeksima  $i$  i  $j$  koji označavaju ose u ravni membrane, glasi:



**Slika 2.4** Delić membrane dimenzija  $dxdy$  izložen transverzalnom istezanju

$$d^2 F_{ii} = F' dj(\sin(\varphi_i + d\varphi_i) - \sin(\varphi_i)) = F' dj d(\sin(\varphi_i)) = F' \frac{d\varphi_i}{di} didj = F' \frac{\partial^2 z}{\partial i^2} didj \quad (2.29)$$

$F'$  u jednačini (2.29) predstavlja silu po jedinici dužine oboda membrane.  $d^2 F_{ii}$  predstavlja razliku transverzalne ( $z$ ) sile zatezanja u koordinatama  $i$  i  $i+di$ , posmatrano na elementu površine čija je dimenzija po drugoj osi jednaka  $dj$ . Za razliku od žice gde je masa raspoređena po jednoj dimenziji, ona je kod membrane koncentrisana u dve dimenzije (membrana se opisuje površinskom gustinom  $\sigma$ ). Ovo znači da je delić membrane, duplo diferencijalno malih dimenzija  $dxdy$ , izložen dejstvu ne konačnih sila zatezanja  $F_x$  i  $F_y$  u pravcima  $x$  i  $y$ , kao u slučaju žice, već dejstvu jednostruko beskonačno malih sila zatezanja  $dF_x$  i  $dF_y$ , koje su jednake proizvodu podužne sile zatezanja membrane i bočne diferencijalno male dimenzije delića membrane ( $dy$  i  $dx$  respektivno). Iz jednačine (2.29) za dve ose  $i$  na osnovu drugog Njutnovog zakona:

$$d^2 F_t = d^2 F_x + d^2 F_y = F' \left( \frac{\partial^2 z}{\partial x^2} + \frac{\partial^2 z}{\partial y^2} \right) dxdy = \sigma dxdy \frac{\partial^2 z}{\partial t^2} \quad (2.30)$$

Konačno, ako zamenimo  $F'/\sigma$  sa  $c^2$  (gde je  $c$  brzina prostiranja transverznog mehaničkog talasa na membrani), dobijamo uobičajeni oblik jednačine oscilovanja membrane:

$$\nabla^2 z = \frac{1}{c^2} \frac{\partial^2 z}{\partial t^2} \quad (2.31)$$

gde  $\nabla^2$  predstavlja Laplasov operator  $(\sum_{i=x,y,z} \frac{\partial^2}{\partial i^2})$ . Analogno jednačini (2.14),

rešavanje se vrši metodom razdvajanja. U slučaju pravougaone membrane, smatramo da funkcija  $z(x,y,t)$  može da se izrazi proizvodom funkcija zavisnih isključivo od  $x$ ,  $y$  i  $t$ , koje obeležavamo sa  $X$ ,  $Y$  i  $T$ . Diferencijalna jednačina (2.30) onda postaje:

$$\left(\frac{1}{X} \frac{\partial^2 X}{\partial x^2} + \frac{1}{Y} \frac{\partial^2 Y}{\partial y^2}\right)c^2 = \frac{1}{T} \frac{\partial^2 T}{\partial t^2} \quad (2.32)$$

Leva strana ne zavisi od desne, te da bi jednačina (2.32) bila zadovoljena obe strane moraju biti jednake nekoj konstanti. Ovu konstantu ćemo kao i u slučaju žice obeležiti sa  $-\omega^2$  (očekujući negativnu vrednost u slučaju oscilacija). Ekvivalentno jednačinama (2.16) i (2.17) za žicu, sledi:

$$\left(\frac{1}{X} \frac{\partial^2 X}{\partial x^2} + \frac{1}{Y} \frac{\partial^2 Y}{\partial y^2}\right)c^2 = -\omega^2 \quad \Rightarrow \quad \frac{1}{Y} \frac{d^2 Y}{dy^2} + \frac{\omega^2}{c^2} = -\frac{1}{X} \frac{d^2 X}{dx^2} \quad (2.33)$$

$$\frac{1}{T} \frac{\partial^2 T}{\partial t^2} = -\omega^2 \quad \Rightarrow \quad \frac{d^2 T}{dt^2} + \omega^2 T = 0 \quad (2.34)$$

Jednačina (2.33) takođe može biti predmet primene metoda razdvajanja. Leva i desna strana zavise od nekorelisanih varijabli, pa da bi bila zadovoljena, obe strane moraju biti konstantne. Ovu konstantu ćemo obeležiti sa  $k_x^2$ , pa sledi:

$$\frac{1}{X} \frac{\partial^2 X}{\partial x^2} + k_x^2 = 0 \quad (2.35)$$

$$\frac{1}{Y} \frac{\partial^2 Y}{\partial y^2} + k_y^2 = 0 \quad (2.36)$$

gde je  $k_y^2 = \frac{\omega^2}{c^2} - k_x^2$ . Problem rešavanja jednačine (2.31) se svodi na problem rešavanja prostih diferencijalnih jednačina (2.34), (2.35) i (2.36). Njihova opšta rešenja su:

$$X = A \sin(k_x x) + B \cos(k_x x) \quad (2.37)$$

$$Y = C \sin(k_y y) + D \cos(k_y y) \quad (2.38)$$

$$T = E \sin(\omega t) + F \cos(\omega t) \quad (2.39)$$

Za membranu dimenzija  $a$  puta  $b$ , granični prostorni uslovi (učvršćenost membrane na krajevima) uslovljavaju:

$$X(0,t) = X(a,t) = Y(0,t) = Y(b,t) = 0 \quad (2.40)$$

što implicira da članovi B i D moraju biti jednaki nuli. Takođe, pošto funkcija  $T$  predstavlja prostu harmonijsku oscilaciju, proizvoljnim izborom početnog trenutka  $t$  jednačina (2.39) se može svesti na sinusni ili kosinusni član (alternativno može biti izražena i sa početnom fazom). Konačno rešenje je oblika:

$$z(x, y, t) = Z_0 \sin(k_x x) \sin(k_y y) \sin(\omega t) \quad (2.41)$$

gde je  $Z_0$  konstanta. Iz (2.35) i (2.36) dobijamo:

$$\omega = c \sqrt{k_x^2 + k_y^2} \quad (2.42)$$

a na osnovu graničnih uslova (2.40) sledi:

$$k_x a = n\pi \quad (2.43)$$

$$k_y a = m\pi \quad (2.44)$$



Slika 2.5 Modovi oscilovanja pravougaone membrane

Sopstvene frekvencije se mogu izraziti iz (2.42), (2.43) i (2.44) kao:

$$\omega_{nm} = c\pi \sqrt{\frac{n^2}{a^2} + \frac{m^2}{b^2}} \quad (2.45)$$

što se može prikazati i u obliku:

$$f_{nm} = \frac{1}{2} \sqrt{\frac{F'}{\sigma} \left( \frac{n^2}{a^2} + \frac{m^2}{b^2} \right)} \quad (2.46)$$

Mod se identifikuje sa dva cela broja, za razliku od jednodimenzionog slučaja. Neki modovi su ilustrovani na sl. 2.5. Oblik se matematički predstavlja prostornim delom jednačine (2.41):

$$Z(x, y) = Z_0 \sin\left(\frac{n\pi}{a} x\right) \sin\left(\frac{m\pi}{b} y\right) \quad (2.47)$$

### 2.3. Vibracije kružnih membrana

U slučaju membrana kružnog oblika, koristi se cilindrični koordinatni sistem. Jednačina (2.30), odnosno (2.31), prelazi u oblik:

$$\frac{\partial^2 z}{\partial r^2} + \frac{1}{r} \frac{\partial z}{\partial r} + \frac{1}{r^2} \frac{\partial^2 z}{\partial \theta^2} = \frac{1}{c^2} \frac{\partial^2 z}{\partial t^2} \quad (2.48)$$

Metoda razdvajanja se primenjuje tako što se pretpostavi da rešenje zavisi od radijalnog člana ( $R$ ), zavisnog od  $r$ , azimutskog člana ( $F$ ), zavisnog od  $\theta$ , i vremenskog člana ( $T$ ), koji je zavisan od  $t$ :

$$\left( \frac{\partial^2 R}{R \partial r^2} + \frac{1}{Rr} \frac{\partial R}{\partial r} + \frac{1}{Fr^2} \frac{\partial^2 F}{\partial \theta^2} \right) c^2 = \frac{1}{T} \frac{\partial^2 T}{\partial t^2} \quad (2.49)$$

Očekivano negativno rešenje (zbog suprotnih znakova koordinate i ubrzanja kod oscilatornog kretanja), jednako za obe strane, obeležićemo sa  $-\omega^2$ . Opšte rešenje vremenskog dela je isto kao u prethodnim slučajevima, i može se svesti na jedan sinusni ili kosinusni član izborom odgovarajućeg početnog momenta:

$$T(t) = A \sin(\omega t) \quad (2.50)$$

Za prostorne članove sledi:

$$r^2 \frac{\partial^2 R}{R \partial r^2} + \frac{r}{R} \frac{\partial R}{\partial r} + \frac{1}{F} \frac{\partial^2 F}{\partial \theta^2} = -\frac{r^2 \omega^2}{c^2} \quad (2.51)$$

što je podložno primeni metode razdvajanja kao i u slučaju sa pravougaonom membranom (analogno jednačini (2.33)). Rešenje obeležimo sa  $k^2$ :

$$r^2 \frac{\partial^2 R}{R \partial r^2} + \frac{r}{R} \frac{\partial R}{\partial r} + \frac{r^2 \omega^2}{c^2} = k^2 \quad (2.52)$$

$$-\frac{1}{F} \frac{\partial^2 F}{\partial \theta^2} = k^2 \quad (2.53)$$

Opšte rešenje jednačine (2.53) je sinusnog oblika, a granični uslovi ( $\theta$  predstavlja azimutalni ugao u opsegu od 0 do  $2\pi$ ) impliciraju da mora biti periodično sa periodom od  $2\pi$ , tj.  $F(\theta) = F(\theta + 2\pi)$ . Zaključak je:

$$F(\theta) = F_0 \sin(k\theta) + F_1 \cos(k\theta) \quad (2.54)$$

gde je  $k$  proizvoljan ceo broj,  $F_0$  i  $F_1$  konstante, a referentna tačka  $\theta$  se može izabrati tako da se kosinusni član ne pojavljuje, tj. tako da je  $F_1 = 0$ . Opšte rešenje jednačine (2.52) je dato Beselovim funkcijama:

$$R(r) = c_1 J_k(\xi r) + c_2 Y_k(\xi r) \quad (2.55)$$

Funkcija  $Y_k$  (Beselova funkcija druge vrste  $k$ -tog reda) teži beskonačnosti u nuli

tako da realna fizička priroda problema implicira  $c_2$  jednako nuli. Granični uslov u slučaju membrane je da je  $J_0(k_1 a) = 0$ , gde je  $a$  poluprečnik membrane, dakle:

$$\xi = \frac{\alpha_{kn}}{a} \quad (2.56)$$

gde  $\alpha_{kn}$  predstavlja  $n$ -ti koren Beselove funkcije prve vrste  $k$ -tog reda. Konačno rešenje glasi:

$$z(r, \theta, t) = Z_0 J_k(\xi r) \sin(k\theta) \sin(\omega t) \quad (2.57)$$

gde je  $Z_0$  konstanta,  $k$  ceo broj, a rezonantna učestanost  $\omega$  je data izrazom:

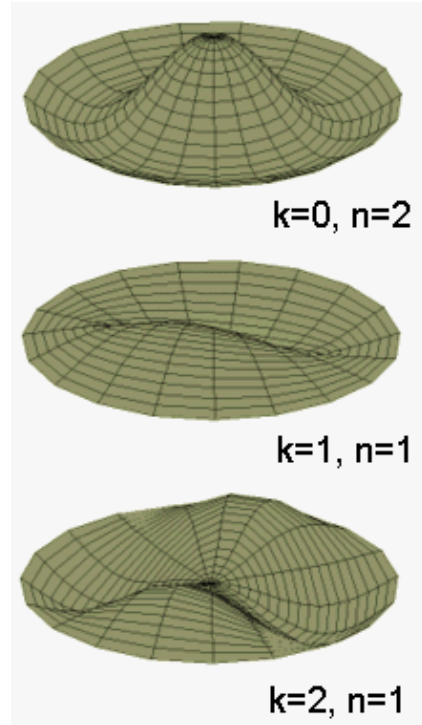
$$\omega_{kn} = c \frac{\alpha_{kn}}{a} \quad (2.58)$$

Alternativno, frekvencija moda određenog sa  $k$  i  $n$  se izražava kao:

$$f_{kn} = \frac{1}{2\pi} \frac{\alpha_{kn}}{a} \sqrt{\frac{F'}{\sigma}} \quad (2.59)$$

Neki oblici modova oscilovanja kružne membrane poluprečnika  $a$  su ilustrovani na sl. 2.6, a opisuje ih prostorni deo jednačine (2.57):

$$Z(r, \theta) = Z_0 J_k\left(\frac{\alpha_{kn} r}{a}\right) \sin(k\theta) \quad (2.60)$$



**Slika 2.6** Modovi oscilovanja kružne membrane

## 2.4. Longitudinalne vibracije štapova

U ovom slučaju, čestice od kojih se štap sastoji osciluju oko svojih ravnotežnih položaja, u pravcu ose štapa. Kako se poremećaj položaja ne prenosi trenutno kroz štap, postoji gradijent normalnog napona duž štapa, ekvivalentno varijacijama pritiska kod prenosa zvuka kroz gas. Ako posmatramo presek štapa diferencijalno male debljine  $dx$ , iz Hukovog zakona sledi:

$$dF = F(x + dx) - F(x) = ES(\delta(x + dx) - \delta(x)) = ES \frac{\partial \delta}{\partial x} \quad (2.61)$$

gde je  $E$  Jangov moduo elastičnosti,  $S$  površina poprečnog preseka, a  $\delta$  relativno izduženje. Ako koordinatu čestice (ili delića štapa) obeležimo sa  $y$  (iako je pomeraj longitudinalan, tj. u smeru  $x$  ose), možemo reći da je:

$$\delta = \frac{\partial y}{\partial x} \quad (2.62)$$

$dF$  u (2.61) predstavlja razliku sila sa dve strane malog isečka štapa, koja teži da ga isteže po osi. Po drugom Njutnovom zakonu:

$$ES \frac{\partial \delta}{\partial x} = ES \frac{\partial^2 y}{\partial x^2} = \rho S dx \frac{\partial^2 y}{\partial t^2} \quad (2.63)$$

Upoređivanjem jednačine sa jednačinom progresivnog talasa:

$$\frac{\partial^2 y}{\partial t^2} = c^2 \frac{\partial^2 y}{\partial x^2} \quad (2.64)$$

vidimo da  $c$  predstavlja brzinu prostiranja zvučnog talasa u čvrstom telu, koja iznosi  $\sqrt{E/\rho}$ . Metodom razdvajanja promenljivih, gornja jednačina se rešava ekvivalentno jednačinama (2.16) i (2.17). Vremenski deo ima rešenje sinusnog oblika (izborom pogodnog početnog trenutka kosinusni član nestaje). Prostorni deo se može pisati kao sinus ili kosinus (ili kombinacija), u zavisnosti od graničnih uslova. Opšte rešenje je oblika:

$$y = (y_0 \sin(kx) + y_1 \cos(kx)) \sin(\omega t) \quad (2.65)$$

gde su  $y_0$  i  $y_1$  konstante, i  $k = \omega/c$ . U slučaju štapa koji je uklješten na jednom kraju, u tački  $x=0$ , i slobodan na drugom kraju, u  $x=L$ , granični uslovi su:

$$y(0, t) = 0 \quad (2.66)$$

$$y(L) = \max \Rightarrow \frac{\partial y(L, t)}{\partial x} = 0 \quad (2.67)$$

Sledi da  $y_1$  mora biti nula, a da je  $k$  dato sa:

$$\cos \frac{\omega}{c} L = 0 \Rightarrow k = (2z-1) \frac{\pi}{2L} \quad (2.68)$$

dok su sopstvene kružne frekvencije date sa:

$$\omega_z = (2z-1) \frac{\pi c}{2L} \quad (2.69)$$

odnosno:

$$f_z = \frac{(2z-1)}{4L} \sqrt{\frac{E}{\rho}} \quad (2.70)$$

Zaključujemo da je jednačina oscilovanja data sa:

$$y(x, t) = Y_0 \sin(kx) \sin(\omega t) \quad (2.71)$$

gde prostorni deo  $Y$  predstavlja oblik modova oscilovanja, prikazan na sl. 2.7.

$$Y(x) = Y_0 \sin \frac{(2z-1)\pi x}{2L} \quad (2.72)$$

U slučaju štapa koji nije učvršćen, maksimalne amplitude javljaju se na oba kraja. Matematički formulisani granični uslovi, sopstvene kružne frekvencije, jednačina oscilovanja, i oblik modova dati su u ovom slučaju jednačinama:

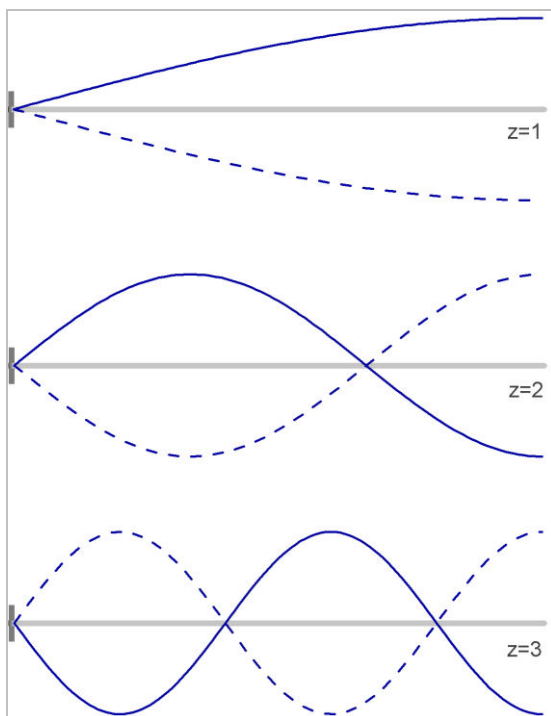
$$\frac{\partial y(0,t)}{\partial x} = \frac{\partial y(L,t)}{\partial x} = 0 \quad (2.73)$$

$$\omega_z = z\pi \frac{c}{L} \quad (2.74)$$

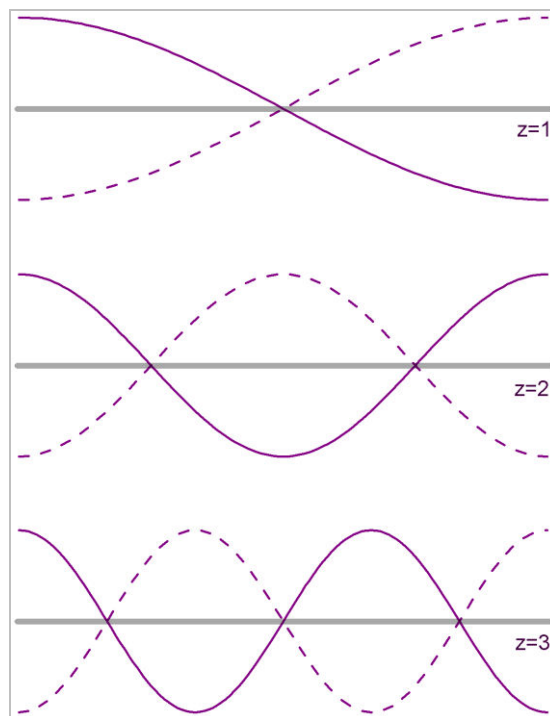
$$y(x,t) = Y_0 \cos(kx) \sin(\omega t) \quad (2.75)$$

$$Y(x) = Y_0 \cos \frac{z\pi x}{L} \quad (2.76)$$

koje se izvode slično kao (2.68)-(2.72). Oblici modova su dati na sl. 2.8.



**Slika 2.7** Modovi oscilovanja štapa učvršćenog na levom kraju



**Slika 2.8** Modovi oscilovanja štapa slobodnog na oba kraja

U slučaju štapa učvršćenog na oba kraja, amplitude oscilacija na krajevima su nula, pa ovaj set jednačina postaje:

$$y(0,t) = y(L,t) = 0 \quad (2.77)$$



$$\omega_z = z\pi \frac{c}{L} \quad (2.78)$$

$$y(x, t) = Y_0 \sin(kx) \sin(\omega t) \quad (2.79)$$

$$Y(x) = Y_0 \sin \frac{z\pi x}{L} \quad (2.80)$$

što matematički odgovara zategnutoj žici. Oblici modova su identični onima na sl. 2.3.

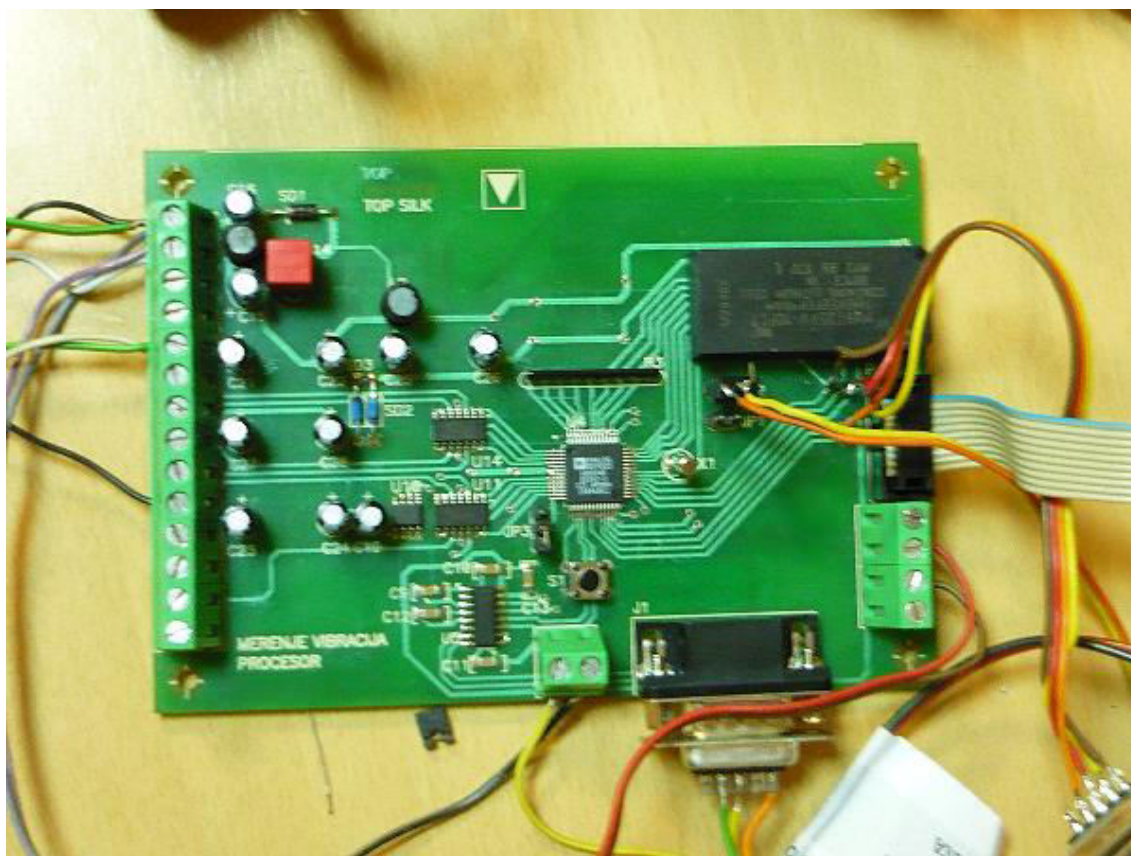
### 3. DIZAJN SENZORSKOG SISTEMA

Razvijena mreža bežičnih senzora sastoji se od više senzorskih uređaja (tri kompletna i tri nekompletna uređaja su proizvedena) i centralne stanice, koju čini personalni računar (PC) sa radio modemom i odgovarajućim softverom (koji je razvijen za ovu namenu). Centralna stanica poseduje praktično neograničene resurse: ne mora se voditi računa o potrošnji energije, procesorske sposobnosti su velike (brzina softverske obrade podataka), i prostor za skladištenje podataka (memorija i hard diskovi) je za više redova veličine veći nego što je neophodno. Svo procesiranje rezultata, osim kompresije prilikom prenosa podataka, vrši se na centralnoj stanici.

#### 3.1. Hardver senzorskog uređaja

Osnovni delovi svakog senzorskog uređaja su:

- glavna ploča, na kojoj se nalazi procesor sa pratećom elektronikom
- ploča akcelerometra, koja se posebno montira, a na kojoj se nalazi integrisani



Slika 3.1 Glavna ploča senzorskog uređaja

(MEMS) troosni akcelerometar, čija je prvenstvena funkcija merenje vibracija

- radio modem, kojim se vrši komunikacija sa centralnom stanicom
- displej, koji predstavlja poseban, za ovu priliku razvijen procesorski uređaj, i

čije je montiranje opciono

- baterijsko napajanje
- sistem za dopunjavanje energije (solarni panel), čija je upotreba opciona
- kućište u koje se smeštaju svi delovi osim solarnog panela, antene i displeja

Arhitektura je otvorena za dodavanje drugih senzora sa zajedničkim ili odvojenim napajanjem.

Izgled glavne ploče prikazan je na sl. 3.1, izgled procesora na sl. 3.2, a blok šema procesora na sl. 3.3. U prilogu je data električna šema svih delova uređaja.

Osim procesora koji će biti detaljnije opisan, na ploči se nalaze i sledeće komponente:

- LT1512 punjač baterija, namenjen za dopunjavanje NiMH baterija eksternim izvorom (kao što su upotrebljene solarne ćelije)

- četiri LP2986 stabilizatora napona predviđena za niske snage (*micropower voltage regulator*) koji stabilišu napon baterija koji je normalno u opsegu od 3,5 do 4,1 V na radni napon uređaja od 3,3 V; jedino se napon napajanja radio modema dovodi direktno sa baterija, pošto on treba da bude u opsegu od 3,4 do 5 V

- ADR291 izvor referentnog napona neophodan za A/D konverziju

- MAX3232 drajver-rišiver za serijsku komunikaciju

- M48T35AV eksterni RAM od 32 KB sa satom realnog vremena i sa sopstvenim (unutrašnjim) baterijskim napajanjem (podaci se čuvaju po isključenju spoljašnjeg napajanja)

- 74HC573 leč kolo za kontrolu RAM-a

- veći broj osnovnih komponenti, kao što su operacioni pojačavači, diode, kondenzatori, otpornici, i dr.

Procesor sadrži ugrađene A/D konvertore za merenje napona baterije (i eventualno izlaza dodatnih senzora) i merenje temperature.

Glavna ploča se može vezati sa pločom displeja čijim radom upravlja preko I2C interfejsa. Displej se priključuje na uređaje u toku razvoja softvera, kao i pri procedurama kalibracije parametara i dijagnostike stanja uređaja.

Na glavnu ploču priključuju se i baterijsko napajanje, ploča akcelerometra, čijim se radom upravlja preko SPI interfejsa, i radio modem, čijim radom se upravlja preko RS232 interfejsa, a moguće je i dodavanje sistema za energetska dopunjavanje, kao i još jednog analognog senzora sa zajedničkim ili odvojenim napajanjem, čiji bi se izlaz merio internim A/D konvertorom procesora. Uz manje prepravke moguće je dodavanje više digitalnih senzora umesto jednog koji se trenutno koristi.

Topologija bežične mreže realizovane pomoću ovih uređaja je zvezdasta. Poruke unutar mreže mogu biti i tzv. *multicast*, što znači da više uređaja prima poruku od jednog pošiljaoca, i *unicast*, što znači da se poruke razmenjuju isključivo između dva uređaja. Sinhronizacija uređaja, od kritičnog značaja za merenja modova oscilacija kapitalnih građevinskih objekata, se vrši prozivnim (tzv. *beacon*) signalom.

### 3.1.1. Procesor

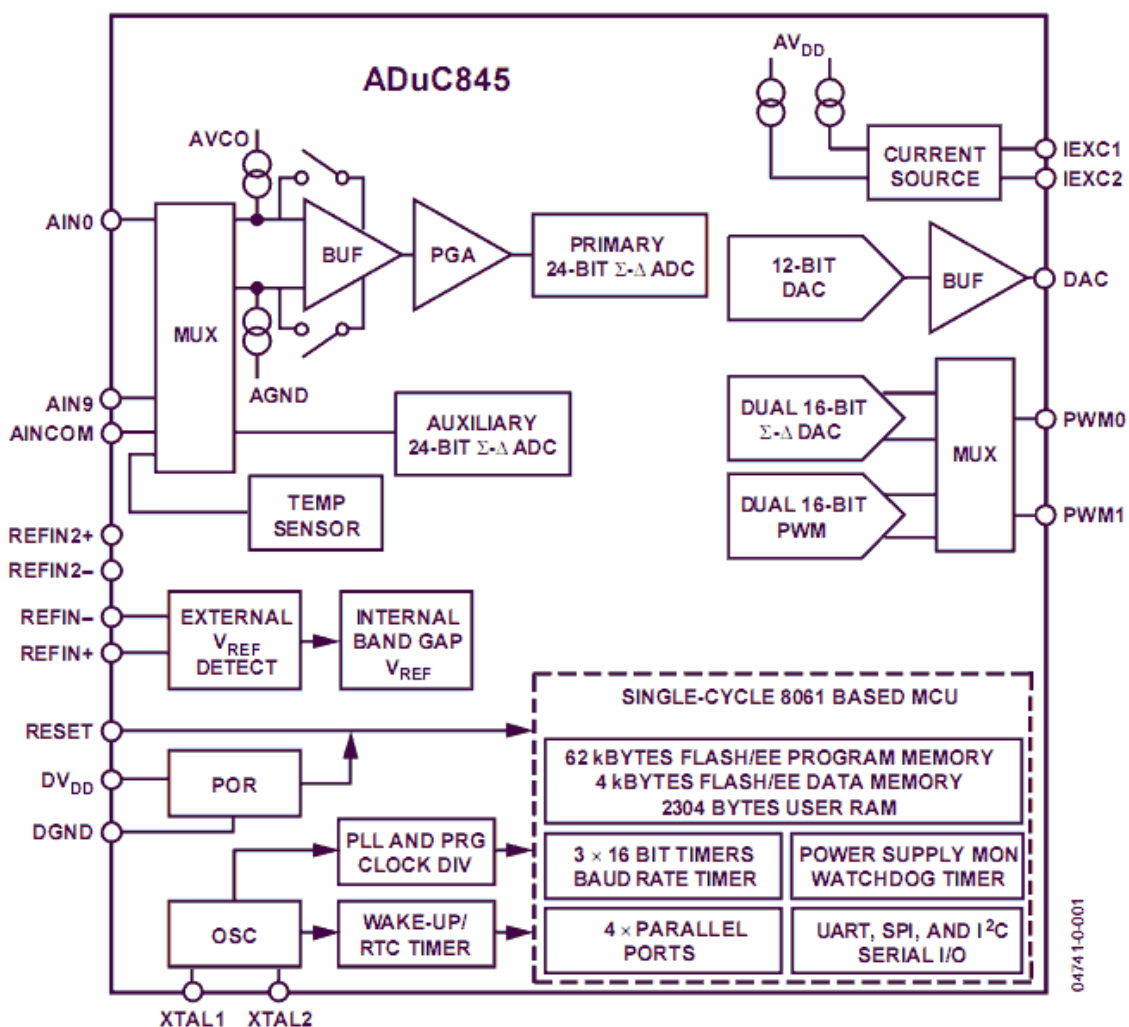
Upotrebljen je procesor (mikrokontroler) ADuC845 firme Analog Devices [35]. ADuC845 spada u 8051 (koristi se i izraz MCS-51) familiju procesora. Ovi procesori imaju dugu istoriju upotrebe u industrijskim aplikacijama i smatraju se vrlo pouzdanim. Iako je do nedavno prevladavalo mišljenje da je ova familija procesora osuđena na odumiranje, podaci govore drugačije.



Slika 3.2 Mikrokontroler ADuC845

Pre nekoliko godina, ovi procesori su imali skoro 50% udela na svetskom tržištu [36]. Ono što ih čini privlačnim je činjenica da postoji velika baza znanja (i u smislu velikog broja programera koji su obučeni za rad sa njima, i u smislu postojanja različitih biblioteka rutina na internetu) i jeftinih softverskih oruđa.

ADuC845 spada u noviju generaciju 8051 procesora. Karakteristika koja ga čini pogodnim za upotrebu u bežičnoj aplikaciji je niska potrošnja, odnosno mogućnost ulaska u *sleep* mod sa potrošnjom od oko 70  $\mu$ W. Za razliku od klasičnih 8051 procesora, novi rade sa programabilnim taktom, koji kod ADuC845 ide do 6,29 MHz za procesore sa napajanjem od 3 V koji su korišćeni u ovoj aplikaciji, odnosno do 12,58 MHz za procesore sa napajanjem od 5 V. Ugrađena su dva desetokanalna 24-bitna sigma-delta A/D konvertora, efektivne rezolucije 19,5 do 22 bita, sa frekvencijom



Slika 3.3 Blok šema procesora [35]

odabiranja do oko 1 kHz, koja premašuje potrebe građevinskih merenja i mogućnosti (u smislu dinamičkih karakteristika tj. propusne moći) senzora koji će se eventualno vezivati na njih. Ugrađeni termometar omogućuje merenje temperature (sa rezolucijom od 1 °C). Dva 16-bitna D/A konvertora nisu iskorišćena u ovoj aplikaciji. Osim kodne memorije od 62 KB (CODE) i 256 bajta interne memorije (DATA/IDATA, što je standard), ADuC845 sadrži i 2 KB RAM (XDATA) i 4 KB *flash* EEPROM. Procesor se napaja naponom od 3 do 3,6 V. Radni opseg temperatura iznosi od -40 do +85 °C, što je bolje od tolerancije većine ostalih komponenti. Maksimalna potrošnja struje iznosi 4,8 mA (što odgovara snazi od oko 16 mW). RS232, I2C i SPI interfejsi služe za povezivanje procesora sa svim perifernim uređajima predviđenim u mogućim aplikacijama. Za programiranje procesora koristi se RS232 veza (procesor ima režim

regularnog rada i režim *download*-a programa), a kao programska oruđa Keil C-51 i assembler.

Procesor ima tri nezavisna tajmera opšte namene, tajmer za uključenje, i watchdog tajmer. Tajmeri opšte namene precizno odmeravaju vreme sa rezolucijom od 0,16  $\mu$ s. Prvi tajmer služi uglavnom za vremensku sinhronizaciju mernih podataka. Drugi tajmer se povremeno aktivira i ima nekoliko namena od kojih su najvažnije pravovremeno isključenje (prebacivanje u tzv. *sleep* mod, odnosno režim smanjene potrošnje) uređaja ako nema korisničkog inputa u proceduri kalibracije, i reset lokalnog komunikacionog bafera prilikom prijema neispravnih poruka u serijskoj komunikaciji u režimu prenosa podataka. Treći tajmer se koristi za davanje takta za serijsku komunikaciju, koja se na internoj RS232 liniji vrši sa modemom bodnom brzinom od 38400 bps (bita u sekundi). *Watchdog* tajmer služi za havarijski reset uređaja (da ovaj ne bi ušao u nedefinisano stanje pri eventualnom kolebanju napajanja ili drugim nepoznatim smetnjama, i nastavio da nekontrolisano radi, i pri tom troši bateriju). Tokom razvoja operativnog sistema bio je korišćen (podešen na period od jedne sekunde), a finalna stabilna verzija softvera ga ne koristi. Konačno, tajmer za uključenje se koristi da se zada period buđenja uređaja (vraćanja iz *sleep* moda, u kome ovaj provodi većinu vremena radi uštede energije).

Procesor sadrži i sat realnog vremena. Sat realnog vremena postoji i u RAM-u sa internim baterijskim napajanjem. U principu je nebitno koji će se sat koristiti kao izvor „zvaničnog“ vremena uređaja. Embedded RTOS koristi RAM, a ne procesor, u tu svrhu, između ostalog zato što je ovaj sat podložen kalibraciji brzine. Realizovane su procedure za rad sa oba sata, kako iz manuelne kalibracije putem displeja, tako i automatski prilikom radio komunikacije sa glavnom stanicom (preuzimanje zajedničkog vremena iz PC računara).

Jedan kanal A/D konvertora koristi se za merenje napona baterije (važnog parametra kod svakog bežičnog uređaja, koji pokazuje koliko je ovaj još u stanju da funkcioniše bez dopunjavanja), a drugi je rezervisan za merenje izlaza analognog senzora koji se opciono može dodati na uređaj. Postoji dodatni „pomoćni“ kanal koji služi za očitavanje ugrađenog termometra. Ovi parametri (temperatura i napon baterije) se prenose na centralnu stanicu sa rezultatima merenja, a mogu se pogledati i na osnovnom ekranu prilikom povezivanja displeja, pre ulaska u proceduru kalibracije.

ADuC845 sadrži dualni data pointer registar (DPTR) sa mogućnošću automatskog svičovanja i automatskog inkrementa ili dekrementa, što pojednostavljuje i značajno ubrzava operacije u kojima se procesiraju dva memorijska bloka.

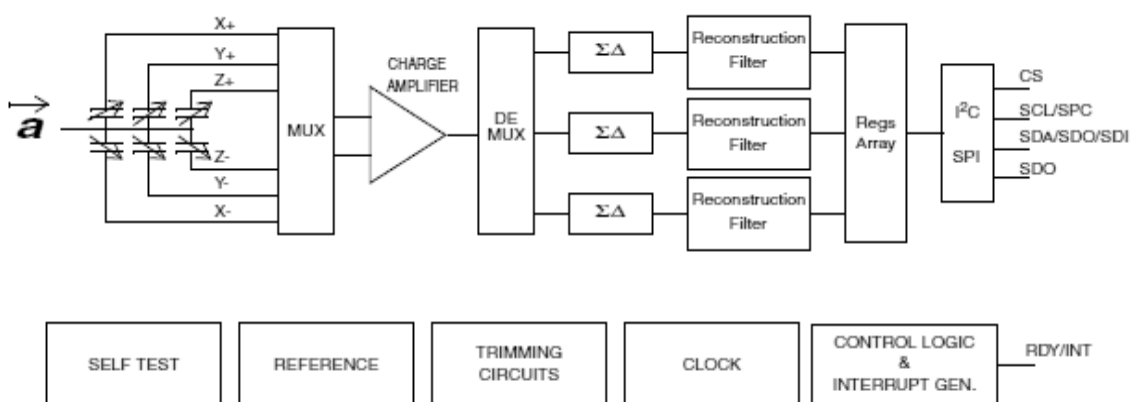
### 3.1.2. Akcelerometar



Slika 3.4 Ploča sa akcelerometrom

Glavni senzor uređaja je integrisani mikroelektromehanički (MEMS) troosni akcelerometar LIS3LV02DL [37], 12-bitne rezolucije, opsega  $\pm 2g$  (ili  $\pm 6g$ , koji se ne koristi u ovoj aplikaciji), koji komunicira sa procesorom glavne ploče putem četvorožičnog SPI interfejsa. Akcelerometar je postavljen na posebnu ploču koja se napaja iz zajedničkog izvora (baterije), da bi se mogao nezavisno montirati u fazama ispitivanja senzorskog uređaja. Izgled ove ploče prikazan je na sl.

3.4. Akcelerometar može da vrši odabiranje frekvencijama od 40, 160, 640 i 2560 Hz. 160 Hz je frekvencija primerena merenjima modova oscilacija kapitalnih građevinskih objekata, kao što su mostovi i brane, da bi se ostvarilo željeno oversamplinganje za frekvencije od značaja, koje su do 25 Hz.



Slika 3.5 Blok šema LIS3LV02DL [37]

LIS3LV02DL je prvenstveno dizajniran za merenja vibracija i detekciju slobodnog pada. Njegov merni element se sastoji od tri kapacitivna polumosta.

Seizmičke mase se pomeraju pod uticajem inercijalnih sila i izazivaju varijacije u ekvivalentnim kapacitetima. Nastali poremećaji se mere naponskom integracijom u odzivu na naponski impuls. Kapacitivni pojačavač sa niskim šumom pretvara ovo u analogni napon, koji je samplovan od strane trokanalnog A/D konvertora. Signal je dalje procesiran od strane logičkog kola na čipu, pre nego što stigne na izlaz sa koga se može pročitati u digitalnoj formi putem I2C ili SPI interfejsa [37,38]. Ovo je šematski prikazano na sl. 3.5.

Radi uštede na ograničenom broju digitalnih ulaza procesora, nema linije *data-ready* interapta, već se čip periodično proziva da se utvrdi da li se prethodno merenje završilo. Pošto je merenje glavni zadatak sistema, manjak procesorskog vremena za druge procese tokom merenja, ovim izazvan, je nebitan. Takođe, kašnjenja reda veličine do par mikrosekundi, ovim izazvana, ne utiču bitno na sinhronizovanost podataka koji se odabiraju sa mnogo većim periodom.

Uključenje u napajanje i softverska inicijalizacija akcelerometra se vrše neposredno po uključenju uređaja da bi se izbegli bilo kakvi efekti prelaznog režima. Nažalost, u praksi se pokazalo da frekvencije merenja akcelerometara nisu identične, odnosno da variraju između različitih primeraka i preko 1%. Zato je važno vremensko označavanje, jer je cilj da sistem uređaja vrši poređenje oscilacija u različitim tačkama u prostoru. Kako je eksperimentalno ustanovljeno, frekvencija jednog pojedinačnog akcelerometra održava se u toku kraćeg vremenskog intervala (jednog merenja) sa zadovoljavajućom preciznošću.

Svaki akcelerometar se posebno ispituje na unakrsnu osnu osetljivost. Ova može iznositi do 3,5% za LIS3LV02DL po proizvođačkoj specifikaciji (što je eksperimentalno potvrđeno). Procedura za prikaz usrednjenog pokazivanja na displeju sa ekstra bitovima tačnosti je realizovana i poziva se prilikom kalibracije. Ispitivanje se izvodi postavljanjem akcelerometra u različite ortogonalne pozicije i merenjem uticaja gravitacionog ubrzanja na sva tri izlaza.

### **3.1.3. Radio modem**

Radio modem PRM-4 je domaće proizvodnje [39] i vezan je putem RS232 interfejsa na procesor glavne ploče. Modem (prikazan na sl. 3.6) predstavlja nezavisan procesorski uređaj, koji sa jedne strane komunicira putem RS232 interfejsa, a sa druge



šalje i prima elektromagnetske signale preko antene koja može biti različitih oblika i dometa. Snaga izlaznog signala je programabilna što omogućava uštedu u slučaju da se komunicira na relativno malim odstojanjima. Maksimalna izlazna (radio) snaga modema je 19 mW. Ovo odgovara maksimalnom dometu u urbanim uslovima od oko 1 km. Tačan domet dosta zavisi od nivoa smetnji i topologije terena. Bodna



**Slika 3.6** Radio modem PRM-4 [39]

brzina komunikacije radio talasima iznosi 9600 bps a noseća frekvenciji je 863-867 MHz. Modem troši struju od nekoliko mA pri prijemu, i 40 mA pri slanju. Koristi se FSK modulacija.

Uobičajena frekvencija rada bežičnih modema je 2,4 GHz, ali je noseća frekvencija od 863 MHz izabrana zbog bolje difrakcije (lakše komunikacije u uslovima zatvorenih prostora i u prisustvu prepreka) i manje elektromagnetskih smetnji (veći broj postojećih bežičnih uređaja radi na 2,4 GHz nego na 863 MHz).

Modem ima sopstveni protokol u kome koristi sopstvene čeksume i adresiranje. Slične mere preduzima i softver na uređaju, tako da je mogućnost prenosa pogrešnih podataka sa dva sloja kontrole praktično nula.

Radi se o takozvanom „paket modemu“, koji formira pakete podataka koje zatim transmituje pojedinačno, uz pomenute mere zaštite (protokol koji sadrži čeksume, adrese, itd).

Modem radi u dva režima, a to su komandni režim, u kome se zadaju parametri komunikacije, i radni režim, u kome se šalju i primaju podaci. Posebnu teškoću predstavlja činjenica da radio modem ne radi kao standardni žičani RS232 interfejs, jer se podaci ne šalju fizički po upisivanju u kontrolne registre procesorovog UART-a, već ih prvo obrađuje softver na modemu (koji je posebna procesorska naprava), a zatim emituju radio talasi. Takođe, prilikom prijema se ne može odrediti tačan momenat fizičkog prijema, već se primaju podaci obrađeni od strane modemovog procesora, koji su ranije stigli putem radio talasa. Ove činjenice onemogućavaju visoku preciznost pri

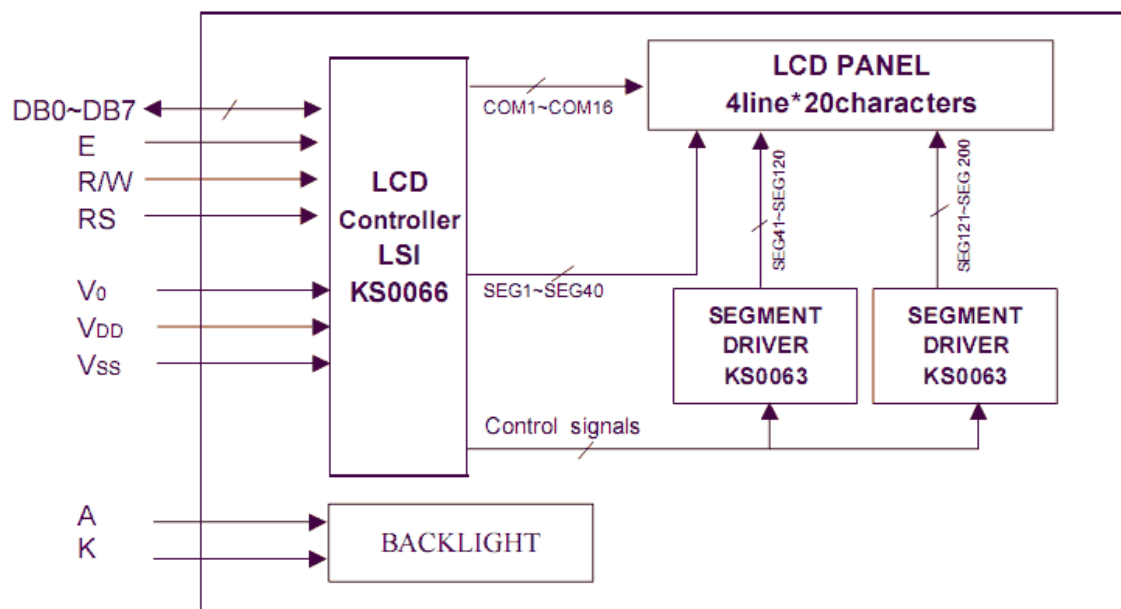
sinhronizaciji mernih rezultata sa različitih tačaka, ali, kao što će biti pokazano, uz različite korekcije mogu se dobiti rezultati više nego zadovoljavajući za merenja u građevinskoj struci.

U komandnom režimu, vrši se podešavanje parametara modema, a u režimu prenosa podataka se puni ili prazni bafer koji se prethodno prima ili nakadno šalje putem radio talasa. Režim prenosa podataka funkcioniše tako da kada dođe do pauze (programabilnog trajanja) na RS232 liniji u smeru procesor - modem, modem vrši kreiranje paketa i radio počinje sa fizičkim slanjem podataka u vrlo kratkom roku. Kada se komunicira u smeru modem - procesor, podaci stižu onog časa kada se primi ispravan paket. Jedan paket sadrži do 253 bajta neto podataka.

Softver na uređaju vrši inicijalizaciju procesorovog UART-a i serijsku komunikaciju u režimu prenosa podataka, kao i kalibraciju modema. Modem ima „masku adrese“, tj. može da prima poruke namenjene ili samo njemu ili grupi modema. Ovo se praktično ne koristi, pošto je dizajniran protokol takav da se komunikacija vrši ili *multicast* u smeru centralna stanica - senzorski uređaji, ili *unicast* dvosmerno između centralne stanice i određenog uređaja. To znači da je maska podešena tako da modem sve vreme prima sve poruke. Poruke čije odredište nije dati uređaj se ignorišu. Potencijalna prednost ovakvog načina rada jeste što je protokol otvoren za nadgradnju u smislu korišćenja tehnike „prisluškivanja“ (*eavesdropping*) [40]. Ova tehnika se koristi da bi senzorski uređaji izvukli određene zaključke iz komunikacije drugih grupa uređaja u okviru mreže. Najčešća primena prisluškivanja je u svrhe vremenske sinhronizacije. Prisluškivanje je, naravno, i vrsta napada na bežične mreže, kojoj su one po prirodi podložne [41].

### **3.1.4. Displej**

LCD displej DEM 20486 SYH-LY [42] je stavljen na posebnu integrisanu ploču sa procesorom AT89C4051 [43]. Ova ploča se nalazi u posebnom kućištu, koje se povezuje sa senzorskim uređajem opciono. Svrha priključenja displeja glavnom senzorskom uređaju je višestruka: olakšava razvoj softvera na glavnoj ploči, omogućava dijagnostiku stanja uređaja u realnim uslovima u slučaju da se ne odaziva na radio signale i služi za kalibraciju uređaja (upisivanje radnih parametara). Ploča sa displejom se napaja iz baterije senzorskog uređaja, jer I2C interfejs ne sadrži zajedničku masu. U



Slika 3.7 Blok šema displeja [42]

slučaju da je baterija prazna, displej ne funkcioniše.

Displej komunicira sa glavnom pločom putem I2C interfejsa (dvožična komunikacija). Procesor AT89C4051 nema ugrađen I2C protokol, tako da je svo programiranje izvršeno manuelno, u assembleru. LCD displej sadrži 4 reda po 20 karaktera, ima 8 tastera, i osvetljenje. Blok šema displeja je prikazana na sl. 3.7. Displej je prikazan na sl. 3.8, gde su objašnjene i funkcije tastera.



Slika 3.8 Displej izvan svog kućišta, sa funkcijama tastera (levi red: levo, gore, dole, Escape; desni red: desno, buđenje, svetlo, Enter)

### 3.1.5. Baterije i solarni panel

Napajanje uređaja vrši se pomoću tri NiMH baterije. Upotrebljavaju se klasične AA baterije kapaciteta 2400-2700 mAh. Alternativa je korišćenje *Low Self-Discharge* baterija (LSD). Sve baterije se prazne i u odsustvu potrošača, što se naziva samopražnjenje (*self-discharge*). Kod LSD baterija ovaj efekat je znatno manje izražen. Nažalost, LSD baterije koje se trenutno mogu naći na tržištu imaju veću cenu, manji kapacitet, i, presudno, manji broj ciklusa punjenje-pražnjenje, tako da uprkos tome što bi uređaj bez solarnih ćelija nešto duže radio sa njima, nisu atraktivna opcija.



Slika 3.9 Realizovani solarni panel

Opciono, na uređaj se priključuje solarni panel (prikazan na sl. 3.9) radi energetskeg dopunjavanja. Razvijeni solarni panel se sastoji od 6 polikristalnih Conrad solarnih ćelija (nominalnog napona 5 V, nominalne struje 81 mA, dimenzija 57×65 mm<sup>2</sup>). Uređene su u pravougaonik i spojene paralelno. LT1512 kolo za kontrolu punjenja je tipa konstantnog napona konstantne struje i može da puni

baterije bez obzira da li je napon na panelu veći ili manji nego napon baterija, u opsegu od 2,7 do 25 V. Gornji limit napona punjenja baterija je 4,1 V (podešava se otporničkim razdelnikom), što odgovara energetskeg nivou baterija od oko 95% [44].

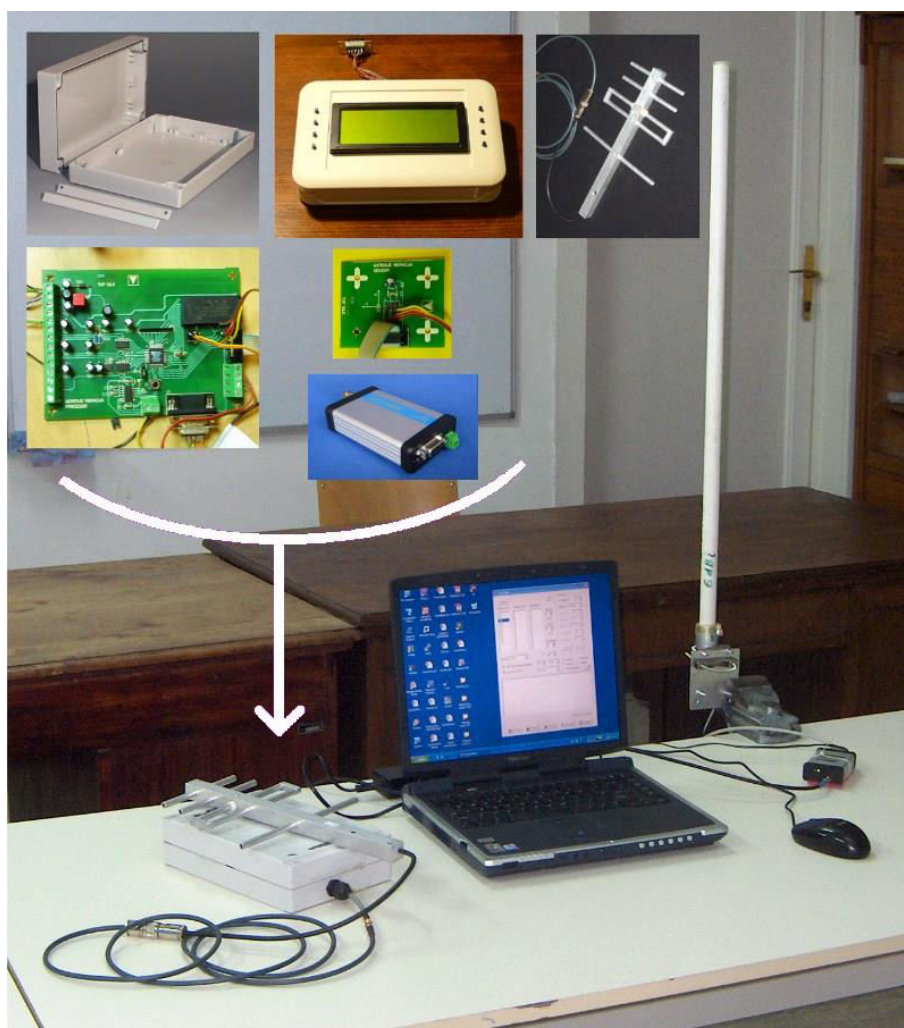
### 3.1.6. Kućište

Radi zaštite od nepovoljnih atmosferskih uslova štampana ploča, ploča akcelerometra, baterije i modem se pakuju u kućište OKW C2016241, koje pruža visoku zaštitu od vlage, a prikazano je na sl. 3.10. Kritičan momenat je izvođenje kontakata za antenu i solarni panel, što kompromituje IP rejting kućišta, ali je neophodno. Sistem sastavljen od



Slika 3.10 Kućište OKW C2016241

jednog senzorskog uređaja i centralne stanice, sa antenama, prikazan je na sl. 3.11.



Slika 3.11 Senzorski uređaj sa centralnom stanicom i antenama

## 3.2. Algoritam rada sistema

Senzorski uređaji rade impulsno, što znači da provode većinu vremena u režimu niske potrošnje (*sleep* režimu), u kome radi samo procesorov tajmer za aktiviranje, i aktiviraju se na nekoliko sekundi u dužim vremenskim intervalima. Potrošnja uređaja u režimu niske potrošnje je oko 0,4 mA (oko 1,3 mW), u slučaju da se ne koristi uređaj za dopunjavanje (čiji kontroler troši još oko 0,1 mA i ako nema prinosa energije). Da bi se izbeglo trošenje struje od strane kontrolera punjenja, koristi se džemper kojim se ovome ukida napajanje, u slučaju da upotreba solarnog panela nije predviđena. Većinu struje od

0,4 mA konzumiraju kola koja omogućavaju da procesor ostane uključen, a ovo se uglavnom odnosi na glavni stabilizator napona. Odvojeni stabilizatori se koriste za delove periferije, što doprinosi tačnijoj A/D konverziji u slučaju potrebe za merenjem eksternog naponskog signala. Oni su, kao i modem (najveći potrošač energije) i sva periferija, isključeni i ne konzumiraju energiju u *sleep* modu.

Period aktiviranja procesora je programabilan i predviđen da bude u opsegu od desetak sekundi do više sati. Pošto uređaj ima ugrađen sat realnog vremena, moguće je zabraniti noćna buđenja („noć“ je, naravno, proizvoljan period dana podložan deklaraciji). Softverski je moguće izabrati osmobarbitni broj (vrednost u opsegu 1-255) i vremensku jedinicu (sekund, minut ili sat).

Po aktiviranju, procesor uključuje radio modem, koji detektuje postoji li signal prozivanja od strane centralne stanice, i u slučaju da je on detektovan u programabilnom roku (oko 3 sekunde), započinje sa radom, a u suprotnom se vraća u *sleep* režim. Centralna stanica, po zahtevu korisnika, počinje da periodično šalje *beacon* signal, kojim se vrši prozivanje i sinhronizacija senzorskih uređaja. Ne postoje garancije da su trenuci aktiviranja uređaja međusobno usaglašeni, mada je to u praksi slučaj kada se sistem aktivno koristi. Kada prođe maksimalno predviđeno vreme (poželjno je da se setuje da bude malo veće od očekivane periode aktiviranja uređaja) ili se svi predviđeni uređaji jave, centralna stanica šalje komandu za rad. Svi uređaji do ovog momenta ostaju u režimu očekivanja naredne komande (*stand-by* režim). To znači da ostaju budni prolongiran vremenski period u očekivanju sledeće naredbe. Ovo je takođe proizvoljan programabilan period, uobičajeno jedan minut. Po izvršenju datog zadatka ponovo se vrši komunikacija sa centralnom stanicom, tj. vrši se prenos mernih podataka centralnoj stanici i uređaji odlaze nazad u *sleep* režim.

Kada se procesor uređaja uključi, on ispituje koji je razlog uključivanja da bi odredio kojim putem izvršenja da krene. Izvori uključivanja procesora su:

- regularno buđenje, tj. zakazano buđenje koje je inicirao tajmer za buđenje
- dolazak napona; dešava se kada se baterije postave u uređaj ili kada se ispražnjene baterije dovoljno napune
- priključivanje displeja, pri čemu buđenje vrši interapt od strane I2C interfejsa (izaziva se pritiskom na taster za buđenje)
- watchdog timer je resetovao uređaj (do ovoga je dolazilo samo prilikom

testiranja operativnog sistema, a u finalnoj verziji watchdog je isključen)

U slučaju regularnog buđenja ili dolaska napona ili watchdog reseta, uređaj ulazi u glavnu petlju, osluškuje postoji li prozivni signal od strane centralne stanice, i vraća se u *sleep* ako ga ne detektuje za par sekundi. U slučaju priključenja displeja i pritiska na taster za buđenje, smatra se da korisnik želi da izvrši dijagnostiku i kalibraciju, te uređaj prikazuje osnovne podatke na ekranu. Ovi podaci uključuju i uzrok prethodnog buđenja, koji se pamti. Čeka se korisnikov pritisak na taster da bi uređaj ušao u proceduru za kalibraciju, u kojoj se mogu menjati mnogi parametri rada, kao što su: vreme i datum, period buđenja, snaga transmisije radio modema, frekvencija samplovanja akcelerometra, i adresa stanice (redni broj uređaja). Može se vršiti i kalibracija A/D konvertora za merenje napona baterije, posmatranje usrednjenog izlaza akcelerometra, itd. Većina parametara koji se mogu editovati tastaturom, takođe se mogu promeniti i prilikom radio komunikacije između centralne stanice i senzorskih uređaja.

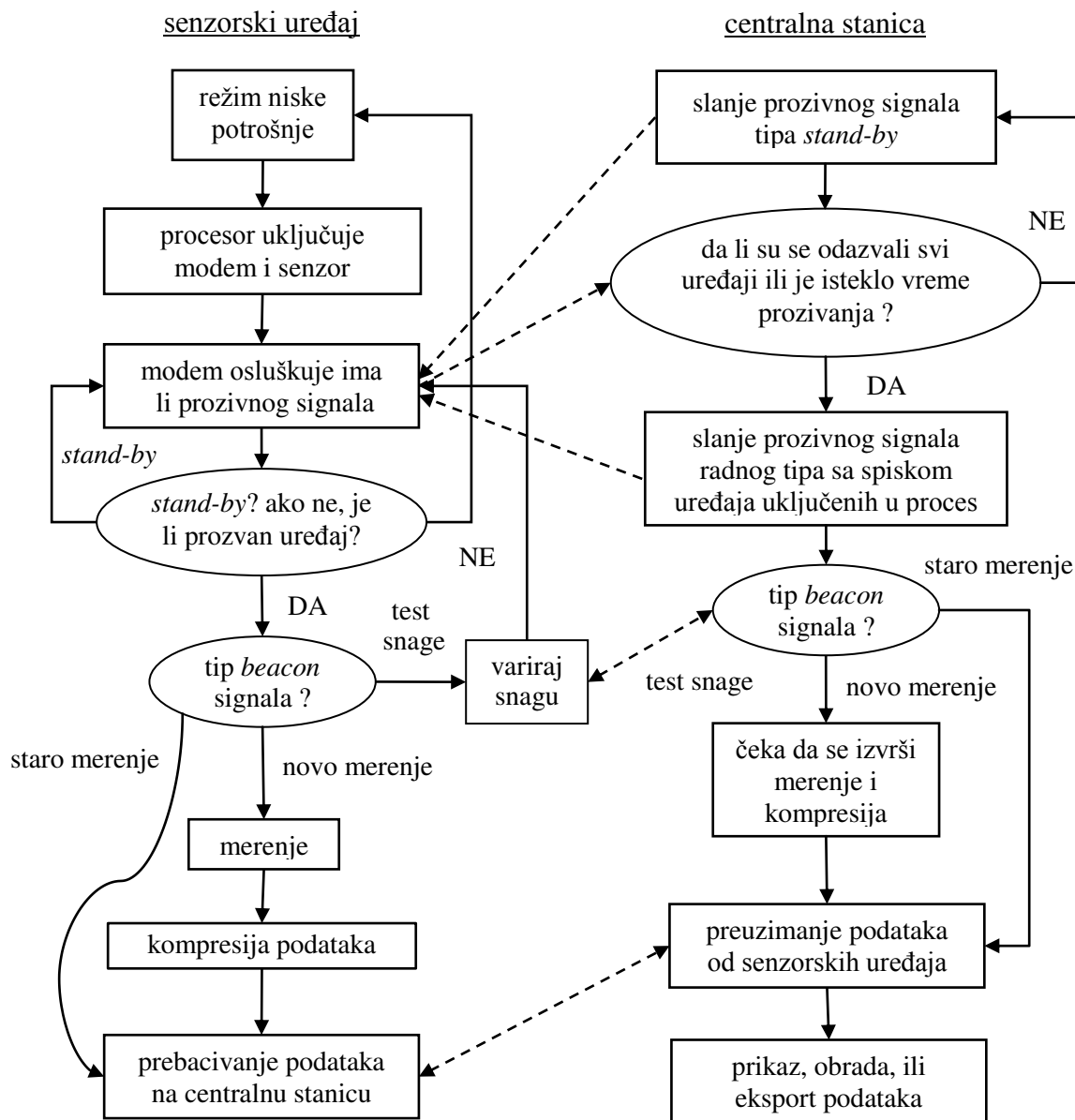
Pošto se komunikacija između centralne stanice i svih senzorskih uređaja vrši na istom kanalu (frekvenciji), algoritam komunikacije prilikom slanja prozivnog signala je takav da stanice odgovaraju sa varijabilnim kašnjenjem koje je funkcija adrese uređaja (adresa puta konstanta). Na ovaj način modem centralne stanice prima sve odgovore sekvencijalno i program analizira niz bajtova poslatih i primljenih hronološkim redosledom. Eksperimentalno je potvrđeno da ovaj metod može podržavati desetak senzorskih uređaja pri frekvenciji emitovanja prozivnog signala od 1 Hz.

Jedan od važnih ciljeva merenja u građevini je dobra vremenska sinhronizacija podataka dobijenih iz različitih tačaka. Ovo je bitno radi preciznog upoređivanja faza harmonika mehaničkih oscilacija u različitim mestima (radi modalne analize vibracija kapitalnih građevinskih objekata). Ovaj cilj ima prioritet nad štednjom energije pa napravljena bežična mreža ima topologiju zvezde, koja kao jednostavna omogućava najbolju vremensku sinhronizaciju, jer je centralna stanica, koja je izvor zajedničkog vremena, spojena sa svim senzorskim uređajima.

Program razvijen za MS Windows kontroliše rad centralne stanice (PC računara). Osnovne funkcije koje ovaj program može da obavi su prozivanje stanica (svaka prozvana stanica ostaje u stand-by režimu neko vreme), test snage (vrše se varijacije izlazne snage senzorskih uređaja, i snima se procenat uspešno prenesenih paketa u funkciji snage), novo merenje (šalje se komanda za merenje, čeka se,

prikupljaju se rezultati), i staro merenje (prikupljaju se postojeći rezultati iz memorije).

Algoritam rada sistema je šematski prikazan na sl. 3.12.



Slika 3.12 Algoritam rada sistema

### 3.3. Operativni sistem senzorskog uređaja

*Embedded real-time operating system* ili *embedded RTOS* je naziv za softver koji se izvršava na glavnoj ploči. Za razliku od na primer PC računara, na kome postoje različiti „slojevi“ softvera, kao što su onaj upisan u matičnu ploču (BIOS), operativni



sistem, i aplikacija, eventualno i skript ili makro unutar aplikacije, na uređaju ovih dimenzija softver je statičniji, tj. ne postoji mogućnost učitavanja različitih modula za vreme regularnog rada, te se ceo ovaj skup, od rutina za čitanje tastature i prikaz tačaka na displeju, do najkompleksnijih rutina za kompresiju podataka, naziva *embedded RTOS*. Razvoj ovog softvera predstavlja najveću stavku u realizaciji ove vrste sistema, ne samo zbog složenosti zadatka, već i zbog nedostatka skupog hardvera i softvera koji omogućava da se proces prati u detalje i ima uvid u fazu izvršenja programa u svakom trenutku.

Osim kontrole toka radi ostvarenja zamišljenog algoritma rada sistema, operativni sistem vrši određen nivo multitaskinga. On vodi računa o interaptima od strane različitih tajmera, I2C interfejsa i serijskog (RS232) interfejsa.

Kada je uređaj isključen, tj. u *sleep* modu, eksterna memorija čuva rezultate merenja, pošto je ugrađen RAM sa baterijom (čiji je rok trajanja nominalno 5 godina, a u ranijoj praksi se pokazalo da slični modeli traju i znatno duže). Na ovaj način može se, recimo, posle gašenja uređaja ponovo zahtevati slanje mernih podataka, ako je došlo do greške u prenosu (ostaju zapamćeni i ostali parametri merenja, a to su datum i vreme, radna temperatura, napon baterije, i tačna vremena uzimanja uzoraka, relativno u odnosu na prijem prozivnog signala).

Radi uštede energije pri slanju rezultata merenja, potrebno je izvršiti kompresiju rezultata merenja. U tom cilju napisan je skup procedura kojima se vrši Hafmanovo kodiranje [45,46] signala koji je prethodno diferencijalno kodiran (DPCM modulacija [47]). Diferencijalno kodiranje se izvršava posle analize periodičnosti signala. Ako signal pokazuje periodičnost, pristupa se kodiranju razlika između podataka koji su jedan detektovani period udaljeni, a u suprotnom se pristupa klasičnom DPCM kodiranju (kodiraju se razlike između susednih odbiraka). Ovako izvedena kompresija u praksi, na realnim signalima, daje rezultate u opsegu od 30% do 70%. Ovaj stepen je dat kao odnos veličina izlaznog i ulaznog seta podataka, pri čemu se pod ulaznim podacima podrazumeva signal koji je DPCM kodiran i ne sadrži neiskorišćene bitove. Odnos kompresije zavisi od intenziteta i pravilnosti vibracija. Poseban izazov u primeni kompresionih tehnika predstavlja činjenica da je radna memorija uređaja samo 32 KB. Preko pola ovog prostora je zauzeto neobrađenim rezultatima merenja i drugim radnim varijablama, tako da kompresija zahteva racionalno korišćenje i preklapanje podataka

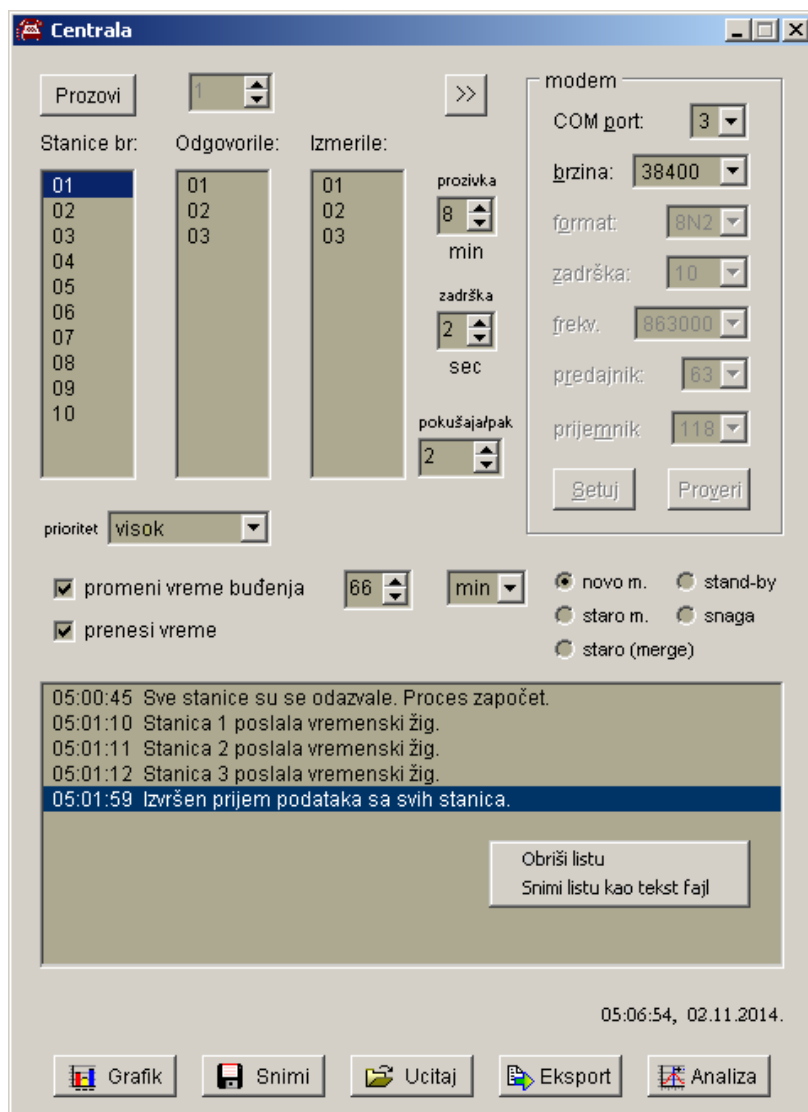
(tzv. *overlay* tehniku, što znači korišćenje istog memorijskog prostora za različite varijable koje se ne koriste istovremeno). Efikasno assembler programiranje je neophodno radi postizanja zadovoljavajuće brzine rada. Analiza periodičnosti (koja se može preskočiti, opciono) traje oko 3 sekunde, a Hafmanovo kodiranje se izvršava za 1-20 sekundi (u zavisnosti od entropije podataka, koja je, kao što je pomenuto, funkcija intenziteta i pravilnosti vibracija). Ovo je u većini slučajeva opravdano vreme (sa stanovišta energetske bilansa) zato što je radio modem u režimu slanja mnogo veći potrošač energije od procesora.

### **3.4. Softver na centralnoj stanici**

Na PC računaru napisan je program za upravljanje sistemom bežičnih senzora. Ovaj program vrši kontrolu celog procesa preko radio modema vezanog na PC serijski port. Program je razvijen u programskom jeziku delphi (pascal).

Pre pokretanja procesa, korisnik selektuje uređaje koje želi da prozove sa liste. Zatim vrši funkciju „prozivanje uređaja“. PC setuje svoj radio modem i počinje sa slanjem prozivnog signala tipa *stand-by*. Stanice koje se uključuju i registruju ovaj signal, odgovaraju, kako prvi tako i svaki sledeći put (signal se šalje učestanošću od oko 1 Hz), sve dok PC ne prestane da ga šalje. On će to učiniti po isteku vremena prozivanja, ili ako sve obeležene stanice odgovore. U momentu kada je neki od ovih uslova ispunjen, prelazi se na slanje drugog prozivnog signala koji sadrži traženu funkciju (novo merenje, staro merenje, ili test snage) i spisak stanica koje treba da je izvrše. Stanice koje nisu uključene u predviđenu funkciju će se ugasiti po prijemu drugog *beacon* signala.

U slučaju funkcije „novo merenje“, PC obustavlja aktivnosti za tridesetak sekundi koliko traju merenje, tipična kompresija, i razne pauze, a onda pokušava da pojedinačno prozove stanice. Svaka koja se javi potom šalje bafer sa rezultatima merenja, koji se tipično sastoji od dvadesetak paketa (oko 5 KB kompresovanih podataka). Svi događaji u procesu ispisuju se sa vremenskim žigom u listi koja se može kasnije skrolovati (log), kao i snimiti u tekst fajl. U slučaju funkcije „staro merenje“, samo se preskače se proces merenja i kompresije, i preuzima se sadržaj memorije sa stanica. Preko funkcije „test snage“, PC zahteva od senzorskog uređaja da šalje pakete izlaznom snagom koja varira od maksimuma do minimuma. Paketi sadrže različite



**Slika 3.13** Nadzorni program na PC računaru, glavni ekran

bajtove (kao što je u stvarnosti tipično slučaj sa kompresovanim podacima). Pamti se koliki je procenat primljenih paketa na centralnoj stanici u funkciji od izlazne snage senzorskog uređaja. Predajna snaga centralne stanice je sve vreme maksimalna.

U slučaju funkcije „stand-by“, PC prestaje sa komunikacijom (posle odaziva svih prozvanih stanica ili isteka vremena prozivanja), a stanice ostaju uključene oko minut i spremne za prijem novih komandi.

Kod svih funkcija, moguće je preneti vreme i datum sa PC računara na senzorske uređaje, kao i podesiti njihovu predajnu snagu i period buđenja.

Program ima funkciju prelaska u komandni mod modema, pri čemu se setuju interni (na liniji PC - modem) i eksterni (modem - bežična mreža) parametri

komunikacije. Interni parametri komunikacije su, jednostavnosti radi, uglavnom fiksirani na senzorskim uređajima. Radi se o bodnoj brzini prenosa, formatu bajtova, vremenu paketizacije, nosećoj frekvenciju (koja se može menjati u finim koracima, a usvojena je fiksna vrednost za celu mrežu), i emisionoj snazi (koja se može smanjiti radi uštede energije u slučaju terenskih merenja na manjoj udaljenosti). Dodatno, na PC-ju postoji i parametar COM port, pošto se modem može priključiti na različite portove.

Još nekoliko parametara je vidljivo na sl. 3.13 koja prikazuje izgled glavnog ekrana programa. Vreme prozivanja, u sekundama, predstavlja vreme trajanja emisije prozivnog signala, a koje treba postaviti tako da ne bude manje od perioda buđenja uređaja. Vreme zadržke, u sekundama, predstavlja vreme između prvog radnog *beacon* signala i trenutka kada senzorski uređaji počinju da mere. Ovo vreme se šalje senzorskim uređajima uz komandu za merenje. Ako je zadržka dovoljno velika, PC će emitovati više prozivnih signala radnog tipa (koji naređuju početak merenja). Svrha je da se u slučaju nesavršene komunikacije povećaju šanse da sve stanice stvarno prime naredbu za početak merenja (trigger). Vreme do početka stvarnog samplovanja izlaza akcelerometra se smanjuje sa svakim sledećim prozivnim signalom. Ovime se ne postiže sinhronizacija reda veličine milisekunde ili manje, zbog neodređenosti izlaznog vremena paketizacije modema i neodređenosti vezanih za multitasking operativni sistem na PC računaru, zbog čega je potrebno primeniti i post-procesiranje u cilju bolje sinhronizacije mernih podataka.

Broj pokušaja po paketu predstavlja broj puta koliko stanica treba da pošalje PC računaru paket sa podacima pre nego što se u slučaju neuspeha odustaje od transfera bafera. Teoretski, ovaj broj bi mogao da bude veliki, ali u realnom radu nema smisla dići ga preko 3 ili 4, jer se u slučaju vrlo loše komunikacije ne postiže mnogo. Gubi se na energetske rezervama jer ostale stanice u redu (koje još nisu prozване radi prikupljanja rezultata) čekaju, čak je moguće da vreme čekanja istekne i da one odu u *sleep* mod, posle čega treba čekati još jedan period buđenja da bi se podaci pokupili. Naravno, u stvarnom radu na terenu, poželjno je da se uređaji po prvom prozivanju prebace u režim čestog buđenja (smanji se period buđenja na minut ili slično), da bi se izbegli problemi. Takođe, postupak preuzimanja bafera sa rezultatima merenja se može ponoviti korišćenjem funkcije „staro merenje“ i to za određenu stanicu samo, pa veliki broj pokušaja gubi smisao, osim kada se komunicira samo sa jednim senzorskim

uređajem, kada broj može biti proizvoljan.

Prioritet procesa unutar operativnog sistema MS Windows je moguće varirati.

Rezultati merenja se mogu pregledati na grafiku, snimiti na disk, ili učitati sa diska iz binarnog fajla. Eksportovanje za upotrebu u drugim programima vrši se snimanjem u tekst fajl. Određene vrste analiza se mogu uraditi u programu, kao što je korelacija signala sa dva senzorska uređaja ili Furijeova analiza. Za većinu kompleksnijih analiza eksportovani fajlovi se učitavaju u programe za matematičku analizu kao što su Origin ili Matlab.

## 4. KOMPRESIJA PODATAKA

Kompresija podataka se definiše kao promena načina kodiranja da bi se smanjila ukupna veličina. Očigledne pogodnosti ovakve procedure su smanjene potrebe za prostorom za skladištenje, u slučaju da se fajl čuva, odnosno transmisionim kapacitetima (u smislu bilo vremena bilo propusne moći), u slučaju da se fajl transmituje. Naravno, sve ima cenu, a u ovom slučaju su to resursi koji se troše na kodiranje i dekodiranje podataka. Ovo može postati i kritično ako se radi o podacima koji se prenose u stvarnom vremenu a sistem ne poseduje zadovoljavajuće sposobnost procesiranja.

Kompresija podataka je vrlo poželjna u bežičnim mrežama jer se štedi na energiji koja je potrebna za transmisiju podataka [48]. Tipično, bežični uređaj troši najviše energije kada je modem ili transsiver u režimu slanja, odnosno kada emituje elektromagnetske talase.

### 4.1. Algoritmi kompresije

Kompresioni algoritmi se dele na one sa gubicima i bez gubitaka. Kod kompresije i dekompresije bez gubitaka (*lossless*) signal se rekonstruiše do zadnjeg bita, bez ikakve distorzije. Tehnike sa gubicima koriste činjenicu da su neki elementi podataka manje bitni od drugih i menjaju ih do odgovarajuće granice, tako da informacije koje su suštinske ostanu jasno definisane. Prirodno, algoritmi sa gubicima proizvode mnogo bolji odnos kompresije (odnos veličine izlaznih i ulaznih podataka). Najčešća oblast primene algoritama sa gubicima su multimedijalni fajlovi (filmovi, muzika, slike). Na primer, kompresije ispod 1% mogu se ostvariti kod filmova, a da se zadrži korektan subjektivni nivo kvaliteta.

Da bi se izvela kompresija bez gubitaka, obično se primenjuje procedura u dva koraka. U prvom koraku (preprocesiranje), ulazni (neobrađeni) podaci se transformišu u grupe bitova (kao što su bajtovi ili *word*-ovi) tako da se entropija smanjuje. Smanjenje entropije znači da treba da bude što manje grupa (ili simbola), i da se one ponavljaju što je češće moguće (da se smanji raznovrsnost). U drugom koraku, poznatom kao entropijsko kodiranje (*entropy coding*), ove grupe se zamenjuju kodovima manje dužine. Što je veća frekvencije neke grupe (bajta, simbola, ili sl), manji broj bita se

dodeljuje njenoj prezentaciji u kompresovanom setu.

Jedan od uobičajenih preprocesora entropijskom kodiranju jeste difrencijalna impulsno kodna modulacija ili DPCM [47,49]. Ako se svaki odmerak signala kodira kao razlika između njega i prethodnog odmerka, ukupan broj simbola koji se koristi da bi se kodirao signal će se verovatno smanjiti. Dva su jasna razloga za ovu pojavu. Prvo, varijacije između susednih odmeraka obično pokrivaju manji prostor nego ceo signal, i drugo, različiti kanali sa sličnim dinamičkim karakteristikama mogu biti kodirani na ovaj način bez potrebe za stalnom transmisijom informacije o jednosmernom ofsetu.

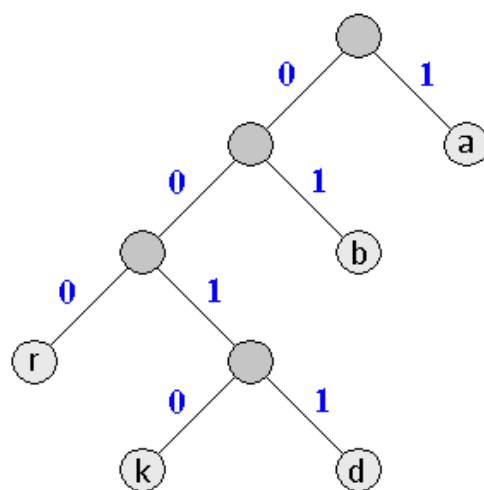
Razlika se može računati ne samo u odnosu na prethodni podatak u setu, već i u odnosu na vrednost koju proizvoljan algoritam predviđa (ovo predviđanje mora biti bazirano samo na prošlim podacima, tako da dekođer može da koristi istu informaciju za rekonstrukciju signala). Izbor pravog predikcionog algoritma za dati slučaj je od suštinskog značaja za dobru kompresiju. Kreacija predikcionog algoritma zahteva da su neke osobine ulaznog signala poznate. Posebna vrsta DPCM kodiranja je predložena u ovom radu, takva da se razlika računa u odnosu na vrednost od pre jednog perioda signala, ako se prethodno utvrdi da signal pokazuje odgovarajući nivo periodičnosti.

Entropijsko kodiranje, koje sledi posle DPCM ili drugog preprocesora, se obično izvodi koristeći jedan od dva algoritma poznata pod nazivima Hafmanovo kodiranje i aritmetičko kodiranje [50,51]. Hafmanovo kodiranje dodeljuje fiksne sekvence bita simbolima iz ulaznog seta, dok kod aritmetičkog kodiranja ne postoje nikakvi fiksni kodovi. Stvarno aritmetičko kodiranje je optimalan način entropijskog kodiranja, mada njegova praktična implementacija uvodi određene gubitke, pošto se u proračunu moraju koristiti digitalni registri i memorijske lokacije konačnog kapaciteta. Ipak, tzv. binarno (digitalno) aritmetičko kodiranje je superiorno u odnosu na Hafmanovo u pogledu kompresionog odnosa, tipično za nekoliko procenata. Problem je u brzini, te se aritmetičko kodiranje izbegava na uređajima niskog nivoa. Aritmetičko kodiranje je dugo bilo nepopularno ne samo iz razloga sporosti (PC računari su postali dovoljno dobri da praktično podrže aritmetičko kodiranje tek kada smo zašli u dvadesetprvi vek), već i zato što su algoritmi za aritmetičko kodiranje bili pokriveni velikim brojem patenata, koji su počeli da ističu oko 2010.

Oba entropijska kodiranja počinju određivanjem broja pojavljivanja simbola u ulaznom setu i kreiranjem tzv. tablice frekvencija. Ovaj proces se može uraditi na celom

ulaznom setu ili se ovaj može podeliti na podskupove. Izbor da li da se podela vrši ili ne je veoma važan, jer dobitak od korišćenja različitih tablica mora biti veći od utroška za transmisiju ovih tablica. U nekim slučajevima, nikakve tablice se ne transmituju. To znači da se podrazumeva inicijalna tablica frekvencija. Primer je kompresija teksta na poznatom jeziku. Frekvencije slova po jezicima su dobro poznate i mogu da posluže kao inicijalna tablica. Tablica onda može ostati nepromenljiva tokom celog procesa kodiranja i dekodiranja, a može se i menjati bazirano na proizvoljnom broju zadnje obrađenih simbola, što se naziva adaptivno kodiranje [46,47,52].

Hafmanov algoritam traži parove simbola sa najmanjim frekvencijama (brojem pojavljivanja) i zamenjuje ih novim simbolom, označavajući zamenjene simbole bitovima 0 i 1. Kasnije, prilikom upisa u izlazni fajl, piše se kod za novi simbol, praćen nulom ili jedinicom. Rekurzivni algoritam ponavlja proceduru zamene najmanje iskorišćenih simbola do momenta kada je ulazni skup prazan. Pri tome se stvara struktura koja se naziva Hafmanovo drvo. Svaki interni čvor ovog grafa ima dve grane potomke označene sa 0 i 1. Krećući se od korena (grafički prikaz je obično obrnut, te je ovo vrh na sl. 4.1) do listova (čvorova na dnu slike), koji predstavljaju simbole koji treba da se kodiraju, svi bitovi na koje nailazimo se uzimaju redom da se formira novi kod (od najznačajnijeg, ili MSB, do najmanje značajnog bita, ili LSB). Manje frekventni simboli, koje predstavljaju lisni čvorovi daleko od korena, okupiraju više bitova, nego više frekventni simboli, koji su predstavljeni lisnim čvorovima „unutar“ drveta. Tablica kodova, ne tablica frekvencija, se obično transmituje, jer stvarne frekvencije nemaju nikakvog značaja za dekođer. Moguće je, naravno, transmitovati i tablicu frekvencija i pustiti dekođer da po istom algoritmu kao i koder napravi svoju tablicu kodova.



**abrakadabra**

1-01-000-1-0010-1-0011-1-01-000-1

**Slika 4.1** Primer Hafmanovog drveta

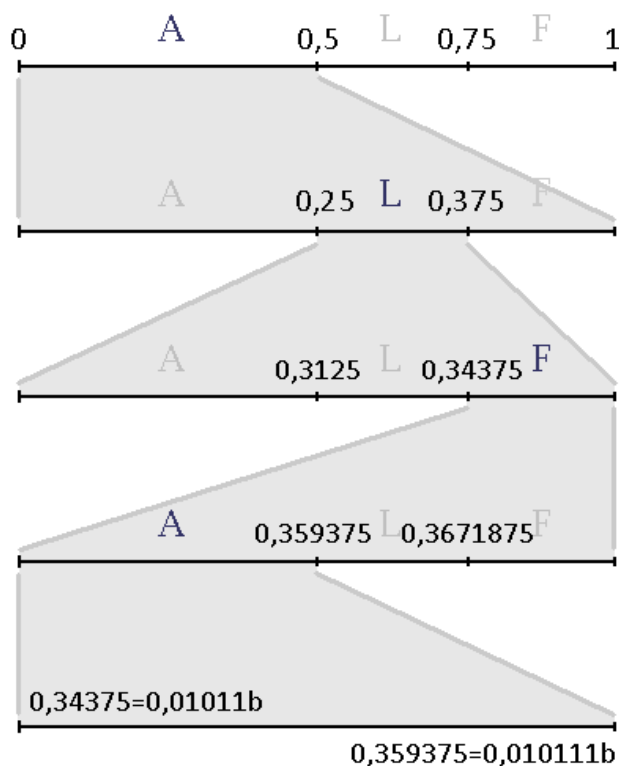
Leđa strana Hafmanovog kodiranja se činjenica da dodeljivanje fiksnih kodova nije optimalan način za redukciju entropije, što se najbolje da razumeti na sledećem primeru. Pretpostavimo da imamo 3 simbola istih verovatnoća (1/3) u ulaznom setu.

Loša strana Hafmanovog kodiranja se činjenica da dodeljivanje fiksnih kodova nije optimalan način za redukciju entropije, što se najbolje da razumeti na sledećem primeru. Pretpostavimo da imamo 3 simbola istih verovatnoća (1/3) u ulaznom setu.



Hafmanov algoritam će ih kodirati kao ili {0,10,11} ili {1,00,01} a to je 5 bita za 3 simbola u proseku. Samo 27 kombinacija zaista postoji za sekvencu 3 simbola, dok 32 jedinstvena binarna broja mogu da se kodiraju sa 5 bitova, tako da je očigledno da još podataka može da se „utisne“ u ovaj broj bitova (postoji 5 neiskorišćenih kombinacija koje se ne javljaju u izlaznom setu).

Aritmetičko kodiranje je odgovor na ovaj problem. Ono ne dodeljuje fiksne sekvence bitova ulaznim simbolima, već operiše sa racionalnim numeričkim vrednostima. Svakom simbolu se dodeljuje *start\_kaunt* i *end\_kaunt* u tablici frekvencija u aritmetičkom stilu, tako da razlika između ova dva bude jednaka broju pojavljivanja simbola. Ove varijable se takođe nazivaju i kumulativnim frekvencijama pošto svaki sledeći simbol koji se procesira ima *start\_kaunt* jednak *end\_kaunt*-u prethodnog simbola, što znači da opisuje ukupan broj pojavljivanja svih simbola do tada procesiranih (*start* ekskluzivno odnosno *end* inkluzivno). Vrednosti *start\_kaunt/total\_kaunt* (*total* opisuje ukupan broj simbola u ulaznom fajlu) i *end\_kaunt/total\_kaunt* za prvi simbol na koji nailazimo u ulaznom setu se dodeljuju



**ALFA = 01011**

Slika 4.2 Primer aritmetičkog kodiranja

varijablama nazvanim *donja\_granica* i *gornja\_granica*. Sukcesivni simboli se kodiraju „insertujući“ njihove vrednosti u opseg između donje i gornje granice (koje se tretiraju kao racionalni brojevi između 0 i 1). Donja granica se povećava za normalizovani *start\_kaunt* sledećeg simbola, a gornja se postavlja na donju plus normalizovani *end\_kaunt* sledećeg simbola. Normalizacija znači množenje sa tekućim opsegom (razlika granica) podeljenim sa *total\_kaunt*-om. Izlaz aritmetičkog kodiranja se obično (mada ne obavezno) vizuelizuje kao broj između 0 i 1, sa preciznošću koja

raste kako se kodiranje nastavlja. Na kraju, proizvoljna vrednost u opsegu između varijabli *donja\_granica* i *gornja\_granica* se izabira, tako da se što manje bita upotrebi za njeno kodiranje. Vizuelni prikaz algoritma aritmetičkog kodiranja dat je na sl. 4.2.

Prilikom izvođenja ponovljenih množenja i deljenja, broj bita potreban da se tačan rezultat zapiše brzo prelazi vrednost sa kojom se može praktično operisati. Različiti algoritmi su smišljeni da se izbore sa ovim problemom i vrše zaokruživanje granica odsecajući određen broj bitova (pri čemu je potrebno voditi računa da enkoder i dekoder ostanu na „istom koloseku“). Cena je malo smanjenje efikasnosti. Npr, povećanje preciznosti registara sa 47 na 63 bita obično proizvodi ispod 1%o dobitka u odnosu kompresije.

Postoji i veliki broj drugih tehnika koje se primenjuju u kompresiji bez gubitaka. Dve češće korišćene su *run length encoding* (RLE), kod koje se nizovi istih simbola zamenjuju kodom i brojem ponavljanja, i Lempel-Ziv algoritmi, kod kojih se frekventni stringovi (nizovi simbola) zamenjuju kodovima [51,53].

## 4.2. Detekcija periodičnosti signala

### 4.2.1. Kros-korelaciona i autokorelaciona funkcija

Korelacija predstavlja meru sličnosti. Kros-korelaciona funkcija između dva signala najčešće se definiše kao [54]:

$$(f * g)(\tau) = \lim_{T \rightarrow \infty} \frac{1}{2T} \int_{-T}^T f(t)g(t + \tau)dt \quad (4.1)$$

gde se  $\tau$  naziva vremenski pomak ili šift. Normalizacija vremenom nije obavezna, i nekad se podrazumeva da je barem jedan signal ograničen na konačan vremenski period (ima vrednost nula izvan njega). U tom slučaju može se koristiti formula:

$$(f * g)(\tau) = \int_{-\infty}^{+\infty} f(t)g(t + \tau)dt \quad (4.2)$$

što se svodi na integral od  $t_1$  do  $t_2$ , gde su  $t_1$  i  $t_2$  granice u kojima su obe funkcije definisane.

Kros-korelacija se može primeniti na bilo kakve signale, ali je uobičajeno da se vrši na naizmeničnim signalima čije su jednosmerne komponente (ofseti) uklonjeni. Na ovaj način, ako ne postoji sličnost između signala, dobijaće se vrednosti oko nule za sve

vremenske pomeraje (šiftove)  $\tau$ . Ako sličnost postoji, vrednost integrala će se periodično menjati, i biti pozitivna ako se faze signala poklapaju, odnosno negativna ako su suprotne.

U kompleksnoj analizi jedna funkcija se konjuguje, da bi kompleksni delovi funkcije iste faze pri množenju dali pozitivan rezultat.

$$(z_1 * z_2)(\tau) = \int_{-\infty}^{+\infty} \bar{z}_1(t) z_2(t + \tau) dt \quad (4.3)$$

Autokorelacija podrazumeva korelaciju funkcije same sa sobom. Autokorelaciona funkcija je obično definisana kao:

$$R_{ff}(\tau) = \lim_{T \rightarrow \infty} \frac{1}{2T} \int_{-T}^T f(t) f(t + \tau) dt \quad (4.4)$$

Isto kao i kros-korelaciona funkcija, i ona se može drugačije normirati, i koristi se konjugacija u slučaju operacije sa kompleksnim funkcijama. U ovom radu je korišćena normalizovana autokorelacija u obliku:

$$R_{ff}^{norm}(\tau) = \frac{R_{ff}(\tau)}{R_{ff}(0)} \quad (4.5)$$

Ukoliko se koristi ovaj oblik normalizacije, uz korišćenje ograničenog vremenskog prozora (korišćenje samo segmenta funkcije u ograničenom vremenskom intervalu), nepotrebno je vršiti normalizaciju vremenom, te je:

$$R_{ff}^{norm}(\tau) = \frac{\int_{-\infty}^{+\infty} f(t) f(t + \tau) dt}{\int_{-\infty}^{+\infty} f^2(t) dt} \quad (4.6)$$

Ako se određeni obrasci u signalu ponavljaju, ova funkcija će takođe biti periodična, te se može iskoristiti i za detekciju perioda i za kvantitativan opis periodičnosti signala.

## 4.2.2. Estimacija kašnjenja

Estimacija kašnjenja (*Time Delay Estimation* odnosno TDE) je procedura identifikacije vremenskog pomeraja (šifta) signala pri kome se dobija maksimum kros-korelacione funkcije sa referentnim signalom [55]. TDE se koristi u aktivnim sistemima kao što su radari i sonari da bi se detektovalo poklapanje emitovanog i odbijenog

signala, i time izračunalo odstojanje do objekta, i u pasivnim sistemima (gde ne postoji veštački izvor signala) kao što su razne matrice senzora (npr. *microphone arrays* ili nizovi seizmičkih senzora) da bi se locirao izvor signala [56]. Slični metodi se koriste u automatici za određivanje mrtvog vremena sistema [57].

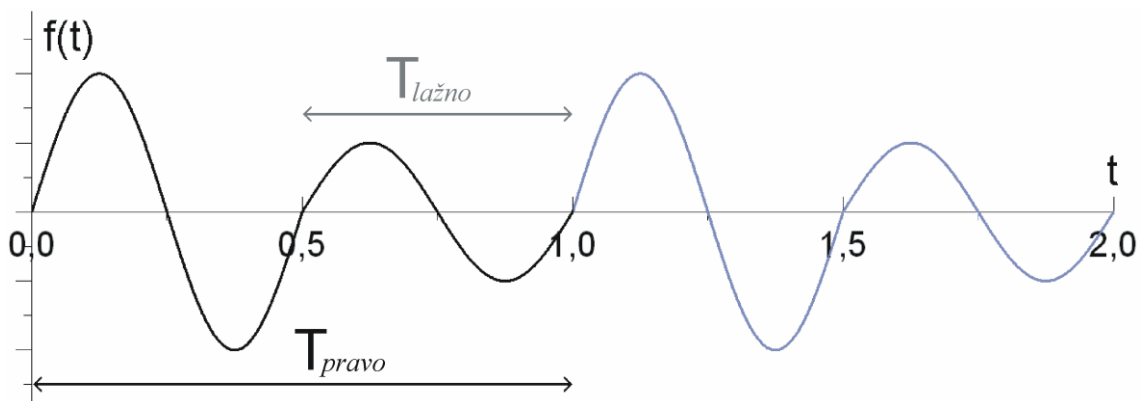
Ako se posmatra autokorelacija jednog signala, možemo da zaključimo da li postoji periodičnost ili kvaziperiodičnost u njemu. Ako je nema, autokorelaciona funkcija ima samo jedan značajan maksimum pri nultom vremenskom šiftu, dok ako posmatramo pravilnu periodičnu funkciju (npr. sinusnu), maksimumi jednakog intenziteta se ponavljaju bez prestanka. Da bi se proizvoljan signal okarakterisao kao periodičan ili kvaziperiodičan, tj. da bismo zaključili da postoji određen obrazac koji se ponavlja, autokorelaciona funkcija mora da ima lokalni maksimum takav da je njegova brojna vrednost u odnosu na brojnu vrednost za nulti pomeraj „značajna”. Tačna vrednost ovog odnosa (normalizovane autokorelacije) se ne može jednoznačno deklarirati. On zavisi uglavnom od toga gde postavimo marginu između aperiodičnog i kvaziperiodičnog, a takođe ima veze sa odnosom signal-šum i mnogim drugim faktorima.

Većina alternativnih tehnika za detekciju periodičnosti, kao što je Furijeova analiza [58], kosinusna transformacija [59,60], ili spektralna analiza metodom najmanjih kvadrata (*least squares spectral analysis* ili LSSA) [61,62], uključuju upotrebu trigonometrijskih funkcija i zato nisu pogodne za upotrebu na senzorskom uređaju niskog nivoa. Veliki broj relativno efikasnih algoritama za određivanje trigonometrijskih funkcija na embedded uređajima je razvijen [63], ali svi su oni relativno spori u poređenju sa osnovnim aritmetičkim operacijama (množenjem, deljenjem, sabiranjem i oduzimanjem).

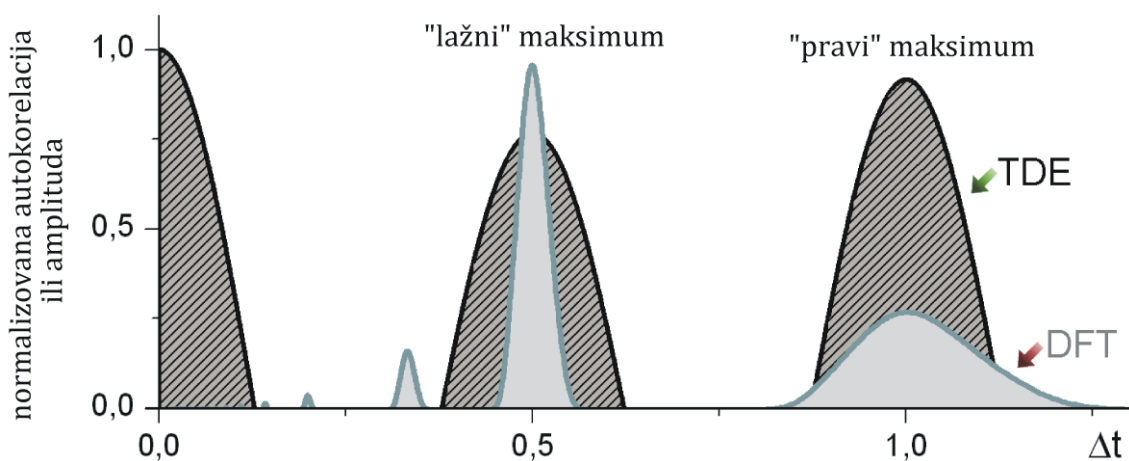
Autokorelaciona funkcija takođe ima veliku prednost nad drugim metodama u slučaju da je osnovni harmonik maskiran višim harmonicima. Primer je ilustrovan na sl. 4.3 (sinus sa promenljivom amplitudom). Pitanje koja vrednost bi trebalo da se uzme za period ovog signala je teorijsko. Za predikcione algoritme, u svrhu smanjenja entropije signala, očigledno je bolje da se porede sekcije signala koje su onoliko slične koliko mogu da budu, tako da ćemo deklarirati veliki period kao „pravi”, a mali period kao „lažni”. Rezultati autokorelacije i Furijeove transformacije su prikazani na sl. 4.4, potvrđujući da je TDE bolje oruđe za detekciju periodičnosti. Slični rezultati se dobijaju

i u drugim slučajevima kada se ponavljajući pod-periodi razlikuju ne samo po amplitudi, već i po trajanju, obliku, itd.

Alternativna metoda za detekciju periodičnosti, koja je primamljiva zbog svoje relativne matematičke jednostavnosti, jeste Enrajtov (Enright) periodogram [64,65], takođe poznat i kao  $\chi^2$  (*chi-square*) periodogram. On se bazira na usrednjavanju vrednosti odbiraka signala koji su udaljeni za različite vremenske šiftove (pomake) i analizi standardne devijacije dobijenih skupova. Ako se pomak poklopi sa stvarnim periodom signala (kada periodičnost postoji), standardna devijacija usrednjenog skupa se približava vrednosti standardne devijacije za ceo signal. U suprotnom, dolazi do usrednjavanja nekorelisanih vrednosti, i signal se pretvara u šum, a standardna devijacija opada. Iako se ovaj metod nije pokazao dobro u uslovima jakog šuma i u prisustvu viših harmonika, mogao bi da se ima u vidu kada se snimaju regularniji signali.



Slika 4.3 Sinusna funkcija promenljive amplitude



Slika 4.4 Poređenje TDE i DFT (diskretne Furijeove transformacije) primenjenih na signal sa sl. 4.3

### 4.3. Algoritam kompresije sa TDE-DPCM preprocesorom

Na senzorskom uređaju je upotrebljeno Hafmanovo kodiranje (entropijsko kodiranje) DPCM preprocesiranih mernih podataka. Razlika je računata u odnosu ne na prethodni odmerak već u odnosu na odmerak uzet pre jedne periode. Period signala je određen primenom autokorelacije (TDE metod u vremenskom domenu). Ovaj tip preprocesiranja nazvan je TDE-DPCM.

Da bi se izbeglo pogoršanje kompresionog odnosa primenom TDE-DPCM metode na neperiodičan signal, kao što je šum ili slučajni impulsi, prvo moraju biti definisani uslovi pod kojima se TDE-DPCM primenjuje, a u suprotnom se prelazi na kodiranje susednih samplova (klasičan DPCM). Dovoljna vrednost autokorelacionog integrala (normalizovana autokorelacija) za detektovani period (vremenski šift) se nameće kao očigledan parametar, pošto zavisi od regularnosti signala i odnosa signal-šum. Univerzalni prag (granična vrednost), koji bi odgovarao različitim mernim sistemima i različitim tipovima signala se ne može definisati. Praksa je pokazala da je prag od 25% (normalizovana autokorelacija za detektovani period) odgovarajući za razvijeni merni sistem i tipove signala koji se sreću prilikom vibriranja građevinskih struktura. Ova vrednost je tesno povezana sa osobinama primenjenog senzora i snimanih signala. Drugačije vrednosti bi mogle odgovarati drugim mernim sistemima. Takođe, ukoliko je signal niskog intenziteta ili ima odličan kompresioni odnos (ova dva uslova su vrlo tesno povezana i praktično se mogu svesti na jedan), normalizovanu autokorelaciju ne treba ispitivati, već se TDE-DPCM procesiranje otkazuje odmah.

Embedded RTOS računa srednju devijaciju signala po završetku merenja. Empirijski je određen prag od 3 kvanta merenja (ubrzanje od oko 3 mg) za sve ose. Prag za kompresioni odnos nije ispitivan prilikom odlučivanja da li da se vrši TDE-DPCM preprocesiranje, pošto je usko povezan sa srednjom devijacijom, a iznosi oko 30%.

TDE analiza može da se izvodi sa korakom vremenskog pomeraja manjim od intervala odabiranja, tako da se jedna instanca signala interpolira, bilo linearno, bilo različitim polinomima. Međutim, ovakva procedura nije opravdana na uređaju niskog nivoa, gde su sposobnosti procesiranja ograničene.

Izvođenje jednog ciklusa autokorelacione integracije zahteva značajan broj (stotine ili hiljade) dvanaestobitnih množenja sa znakom, na procesoru koji poseduje samo mašinsku instrukciju za množenje osmobitnih brojeva bez znaka. Integracija se

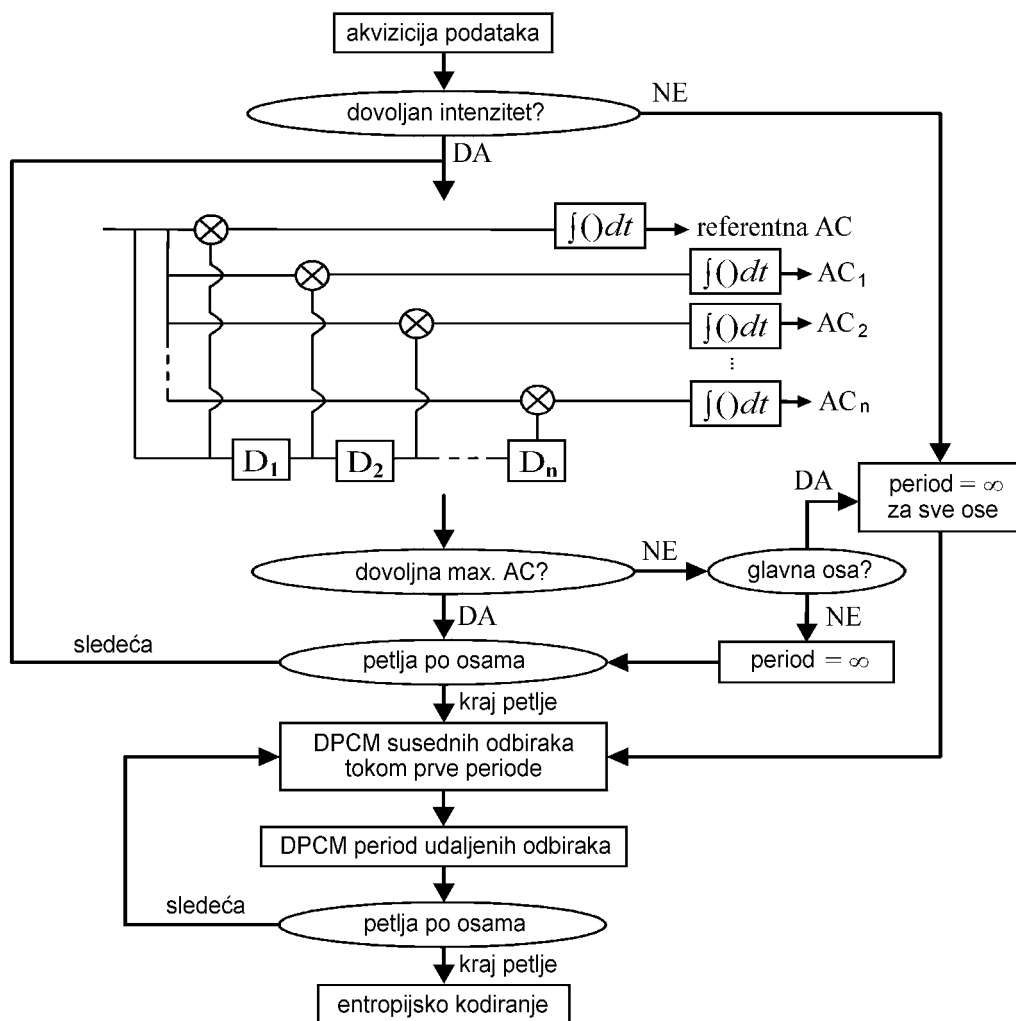
ponavlja za svaki novi vremenski pomeraj, tako da imamo dvodimenzionu petlju koja treba da se izvrši stotine hiljada ili milione puta. Optimizacija ovog procesa je izvedena primenom nekoliko tehnika. Prozor veličine 20% od ukupnog mernog bafera (ceo bafer sadrži po 3200 rezultata merenja za svaku osu), iz sredine istog, je izabran za proračun korelacije i pomeran je do 20% ukupnog trajanja merenja u jednom smeru. Ovaj broj je izabran empirijski jer daje relativno dobre rezultate uz razumno ubrzanje procesa. Najveći periodi u izvođenim eksperimentima su 5-10% trajanja merenja, a 20% je prihvatljiva vrednost koja je kompromis između finoće (rezolucije, odnosno kvaliteta) autokorelacije, šanse da se unutar prozora nađu relevantni, karakteristični elementi signala, i prihvatljivog vremena izvršenja. Na ovaj način se izvrši oko 400.000 instrukcija po osi. Ipak, puna kalkulacija integrala za svaki vremenski pomeraj nije neophodna. Ona može biti obustavljena ako određen broj odmeraka (izabrana vrednost je 1% ukupnog broja, tj. 5% prozora) koji su prvi procesirani daje negativan rezultat. Štaviše, najkraći periodi detektovani u stotinama eksperimenata na stvarnim objektima su 8 perioda samplovanja, tako da se po dobijanju prvog negativnog rezultata i obustavljanja daljeg računanja autokorelacije, preskaču sledeća 4 pomeraja. Uvođenjem svih ovih poboljšanja, vreme izvršenja se smanjuje na 2,5 sekundi po osi. U poređenju sa ovim, Hafmanovo kodiranje traje od sekunde za šum do 20 sekundi za veoma kompleksne signale. Dalji dobici se mogu postići ako se svi DPCM kodirani podaci sa apsolutnim vrednostima većim od 8 bita tretiraju kao maksimalan osmobitni ceo broj ( $\pm 255$ ), pa se sva množenja vrše korišćenjem jedne mašinske instrukcije (uz vođenje računa o znaku). Vibracije velikih građevinskih objekata ne mogu biti i velike frekvencije i velike amplitude tako da je ovo opravdano u stvarnom radu. Veće razlike susednih odmeraka od 8 bita su veoma retke a brzina se povećava za 25% primenom ove tehnike. Konačno, u slučajevima kad se očekuju periodi znatno manji od 20% trajanja merenja, glavni prozor od 20% se može dodatno smanjiti više puta (u zavisnosti od karakteristika očekivanog signala).

Osobina oscilacija velikih objekata je uzeta u obzir ovde. Oscilacije po jednoj, glavnoj osi, su ponekad velikih perioda, preko sekunde. To su npr. vertikalne oscilacije visećih mostova. Međutim, oscilacije po „sporednim” (sekundarnim) osama većine građevinskih struktura najčešće nemaju izražene harmonike niskih frekvencija. Zato, manji maksimalni vremenski šift može da se uzme da bi se ispitala korelacija po ovim

osama. Analiza po sekundarnim osama može biti čak potpuno izbegnuta pod uslovom da autokorelacija izvršena korišćenjem detektovanog pomeraja za glavnu osu daje zadovoljavajuće rezultate i za sekundarne ose. Fino podešavanje parametara algoritma je moguće u ovoj tački da bi se on optimizovao ili za brzinu ili za kompresiju. Trenutno korišćeni algoritam uzima isti 20% prozor ali ga pomera samo za 2% ukupnog vremena (umesto 20%) da bi odredio periode signala za sekundarne ose. Period dobijen za glavnu osu, ukoliko se nalazi izvan ovog opsega, uključuje se u poređenje (traženje maksimuma normalizovane autokorelacije). Na ovaj način TDE se izvršava za 3 sekunde ako se koriste puna dvanaestobitna množenja.

Algoritam kompresije prikazan je na sl. 4.5.

Sve petlje i rekurzivne procedure u TDE, DPCM, i Hafmanovom kodiranju su



Slika 4.5 Algoritam kompresije na senzorskom uređaju; AC označava autokorelaciju



napisane u assembleru da bi se povećala brzina. Ubrzanje u poređenju sa embedded C kodom, za identičan algoritam, a koji je već optimizovan za brzinu od strane kompajlera, iznosi 1,6 do 2,2 puta. Veća ubrzanja se dobijaju za kompleksne signale (visoke entropije), koji se kompresuju sporije. Tabela 3 pokazuje primere kompresionih odnosa i vremena kompresije za neke oblike signala.

**Tabela 3** Kompresioni odnosi i vremena kompresije u C-u i assembleru, za različite tipove signala;  $f_s$  označava frekvenciju samplinga (odabiranja)

tip signala	$f_s$ [Hz]	kompresija	$t_c$ [s]	$t_{asm}$ [s]	$t_{asm}/t_c$
šum	40	12 %	1,00	0,61	61 %
šum	160	15 %	1,19	0,72	61 %
šum	640	30 %	2,05	1,18	58 %
šum	2560	54 %	9,36	4,94	53 %
sinus $f/f_s=0,003$	2560	54 %	8,87	4,56	50 %
sinus $f/f_s=0,02$	160	63 %	15,63	7,73	49 %
sinus $f/f_s=0,19$	160	73 %	42,72	19,76	46 %
sinus $f/f_s=0,26$	2560	98 %	15,29	7,58	50 %
veštački generisan	-	99 %	45,22	21,19	47 %

#### 4.4. Hafmanove tablice i memorijska optimizacija

Primarna komponenta signala je, kao što je već spomenuto, obično prisutna samo na jednoj, glavnoj osi. LIS3LV02DL (upotrebljeni MEMS akcelerometar) takođe ima nejednake karakteristike po osama. Njegova z-osa, koja najčešće služi kao „primarna”, jer se primarne oscilacije struktura kao što su mostovi ili pruge vrše u vertikalnom smeru, ima veći šum kada se izloži inicijalnom ubrzanju od g (normalan položaj senzora). Ovo se objašnjava činjenicom da se čip ustvari sastoji od dva odvojena senzora, jednog dvoosnog akcelerometra za x i y ose, i jednog posebnog koji meri z ubrzanje. Njihove opruge i elektrode nisu identično građene i samo šum po z osi se povećava kada se z seizmička masa izvede iz ravnotežnog položaja [38]. Dakle, signali u x i y pravcima su obično i manje amplitude i manjeg šuma.

Zato su razmatrane tri varijante Hafmanovog kodiranja, a to su jedna tablica frekvencija i kodova za sve ose, dve tablice (jedna za primarnu i jedna za sekundarne ose), i po jedna tablica za svaku osu. Iz više stotina izvedenih eksperimenata, došlo se do zaključka da je verzija sa dve tablice najbolja u oko 60% slučajeva, zajednička

tablica je najbolja u oko 40% slučajeva, a tri odvojene tablice su skoro uvek najgore rešenje. Ovaj zaključak ipak ne može da se generalizuje i najbolja varijanta zavisi od karakteristika signala koji se meri.

Dok je merenje u toku, 12-bitni podaci se zapisuju u uobičajenoj 16-bitnoj formi (word). Proračun srednje vrednosti se izvodi za vreme merenja tako da se signalu ukida jednosmerna komponenta (oduzima zaokružena, tj. *integer* srednja vrednost) odmah po završetku mernog perioda, u petlji sa jednim prolazom, za sve ose. TDE analiza se zatim vrši da bi se odredila periodičnost signala. Ako su signali dovoljne amplitude i normalizovane autokorelacije zadovoljeni, DPCM počinje kodirajući razlike između susednih odmeraka tokom prvog perioda, a nastavlja kodirajući razlike između odmeraka jedan period udaljenih za ostatak signala. Ako uslovi nisu zadovoljeni, DPCM kodiranje susednih smplova se vrši za ceo bafer. Konačno, u cilju povećanja slobodne memorije za tablice frekvencija simbola i Hafmanove kodove istih, ceo bafer se pakuje - 4 neiskorišćena bita po wordu se eliminišu. Ovime se njegova veličina smanjuje sa 19200 na 14400 bajtova. Trinaesti bit je ispušten u ovoj proceduri. Teoretski, ovo može da uzrokuje gubitke, jer DPCM traži dodatni bit za znak, te ne može da kodira razlike sa apsolutnom vrednošću većom od pola pune skale koristeći originalni broj bita, ali ovo stanje nije nikada nastupilo u praksi (razlike susednih odmeraka nikad ne prelaze pola opsega akcelerometra u realnim situacijama). Pakovanje bafera povećava ukupno vreme izvršenja kompresije, ne zbog sopstvenog izvršenja, koje se meri milisekundama, već zato što je pristup pakovanim 12-bitnim varijablama kompleksniji i sporiji nego pristup normalno zapisanim dvobajtnim (16-bitnim) formama, ali oslobađanje memorije u uslovima uređaja ograničenog memorijskog kapaciteta je neophodno.

I dalje ne ostaje dovoljno mesta u 32KB RAM memoriji (deo zauzet baferom sa sirovim podacima i radnim varijablama) za kompletne tablice frekvencija i Hafmanove kodove. Zato, DPCM simboli veći od 9 bita (uključujući znak, dakle  $\pm 255$  kvanata merenja), dele zajedničko mesto u tablici kodova i frekvencija, čime se veličina tablice smanjuje na 512 zapisa (nula, negativni i pozitivni brojevi do 255, i jedno deljeno mesto za veće vrednosti). Pri pojavljivanju većeg simbola od 255, predviđeni kod se piše u izlazni strim praćen egzaktnom 12-bitnom vrednošću (sirovim podatkom). Ovo bi vrlo loše uticalo na kompresiju kada bi se dešavalo često. U većini eksperimenata, velike

razlike se uopšte ne javljaju. Jedino eksperimenti sa visokim frekvencijama na vibracionoj platformi (laboratorijski, ne terenski test) i prazni vozovi koji prelaze preko loše podbijenih pragova su proizveli ovo stanje. Uključivanje filtra propusnika niskih frekvencija na čipu, što je softverski moguće, bi dodatno smanjilo verovatnoću pojave velikih razlika.

Hafmanove tablice se pišu u izlazni bafer korišćenjem *pre-order traversala* [66], tako da je komprimovana veličina jednaka broju simbola koji se javljaju u ulaznom setu puta njihova veličina (9 bita) povećana za približno 1,5 bita (dakle 10,5 bita). U poređenju sa traversalima strukture tipa drveta u opštem slučaju, Hafmanovo drvo je posebno pogodno za aplikaciju ovog metoda pošto je potpuno binarno, što znači da svi interni čvorovi imaju tačno dve grane potomka. *Pre-order traversal* rešenje se smatra optimalnim u opštem slučaju, i mada nije uvek najbolje, nudi značajne uštede u slučajevima kompleksnih signala sa visokom entropijom (lošim odnosom kompresije), dok je gubitak mali na signalima sa niskom entropijom (dobrim odnosom kompresije).

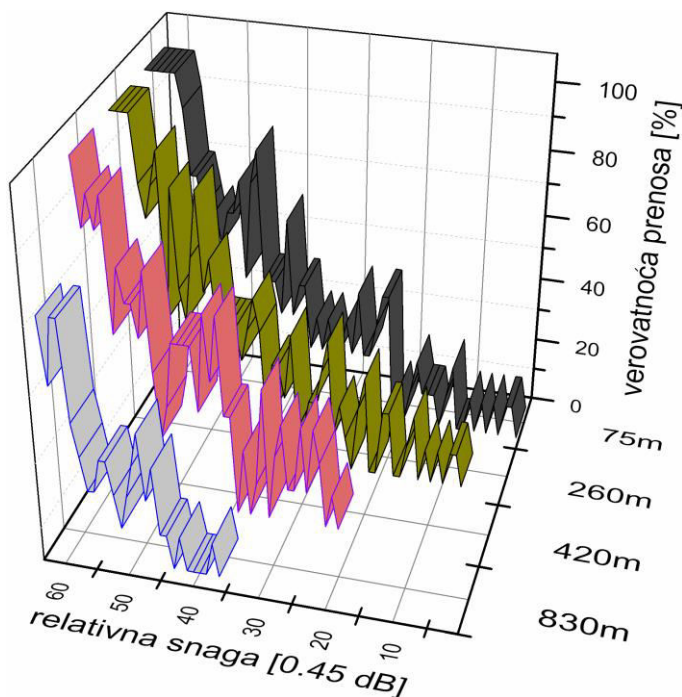
Radi uštede prostora, varijable koje se ne koriste istovremeno u procesu kompresije i kasnijeg prenosa podataka, koriste iste memorijske lokacije, što je poznato pod nazivom *data overlaying* (preklapanje). Izlazni bafer se nalazi 2 KB ispred bafera sa rezultatima merenja sa glavne ose, što je dovoljno da sadrži dve Hafmanove tablice maksimalne veličine. U režimu dve odvojene tablice, tablica kodova za glavnu osu se zapisuje u ovaj bafer, praćena kompresovanim podacima za glavnu osu, koji delimično zauzimaju mesto koje je bafer sirovih ulaznih podataka imao. Proces se nastavlja za sekundarne ose, koristeći isti prostor za Hafmanovu tablicu frekvencija i kodova, kao i druge varijable koje se koriste pri proračunu istih (brojanju i konstrukciji Hafmanovog drveta).

## 5. ISPITIVANJE KARAKTERISTIKA SISTEMA

### 5.1. Ispitivanje dometa radio modema

Dometa radio uređaja je pojam koji je nemoguće jednoznačno definisati. Verovatnoća prenosa određenog paketa podataka zavisi od njegove veličine, snage predajnika, udaljenosti prijemnika, prepreka, prirodnih i veštačkih elektromagnetskih smetnji, atmosferskih uslova, itd. Veći broj ponovnih pokušaja povećava šansu prenosa. U slučaju senzorskih uređaja sa ograničenim energetske izvorom potrebno je pronaći kompromis između šanse da se odgovarajući broj paketa (u slučaju razvijenog senzorskog sistema, ovaj broj iznosi od oko 10 za uobičajen šum do oko 40 za signale sa visokom entropijom) uspešno prenese, i činjenice da osim trošenja energije na datom uređaju, postoje i drugi senzorski uređaji uključeni u proces koji za to vreme čekaju da budu prozvani i troše svoje energetske rezerve. Program na centralnoj stanici dozvoljava korisniku da izabere maksimalan broj pokušaja da se paket prenese. Na grafiku prikazanom na sl. 5.1, date su verovatnoće da se paket pun različitih bajtova, dužine 250, prenese pri različitim predajnim snagama senzorskog uređaja u funkciji odstojanja. Za svaku snagu vršeno je po 10 prenosa (tako da su dobijene verovatnoće prenosa u desetinama procenata). Merenja su vršena na nekoliko desetina tačaka. Centralna stanica bila je fiksirana na najvišem spratu zgrade Građevinskog fakulteta, dok je senzorski uređaj pomeran duž Bulevara kralja Aleksandra u pravcu istoka. Maksimalna snaga predajnika iznosi 19 mW i odgovara softverskom parametru snage od 63, a svako smanjenje za 1 je ekvivalentno slabljenju od 0,45 dB (tako da relativna snaga 0 odgovara snazi od oko 30  $\mu$ W). Primećeno je da rezultati ispitivanja dometa ne zavise od napona napajanja baterija (u opsegu 3,55 do 4,1 V), ali da variraju u različitim periodima dana, što se pripisuje veštačkim smetnjama, jer su merenja vršena u uslovima intenzivnog saobraćaja na prometnom bulevaru u centru grada. Na grafiku je prikazano samo nekoliko, od više desetina setova merenja, radi preglednosti, pri čemu su izrazito loši rezultati (za koje se pretpostavlja da su posledica promenljivih veštačkih smetnji) u startu odbačeni.

Prilikom eksperimenata sa prenosom paketa različitim predajnim snagama, uočena je mana modema da šansa za uspešan prenos zavisi i od sadržaja paketa. Neki



**Slika 5.1** Verovatnoća prenosa paketa kao funkcija predajne snage i udaljenosti

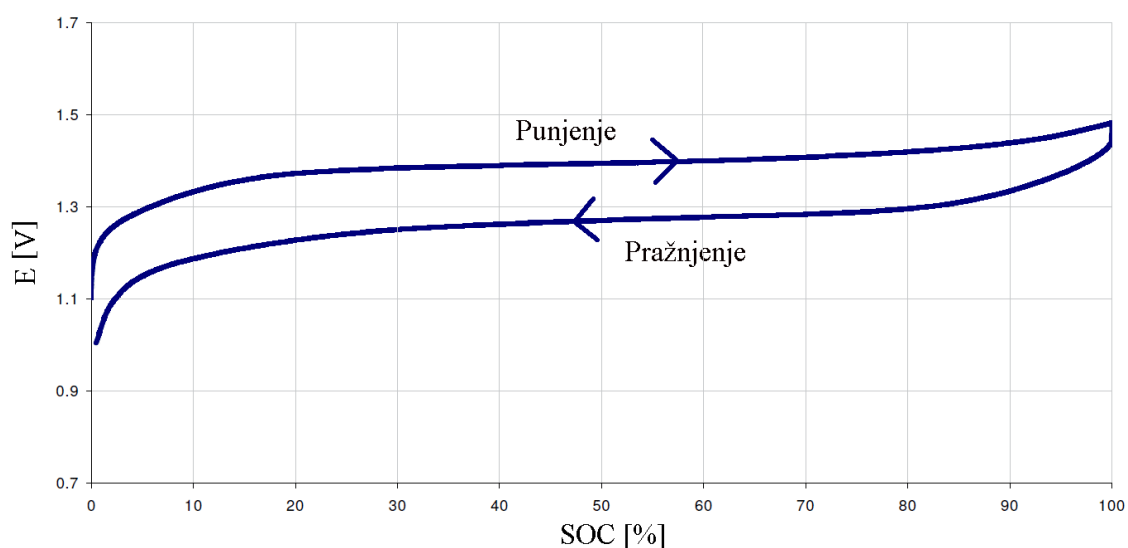
paketi konstantno imaju veću šansu da budu preneseni od drugih, pod uslovom da je modem u režimu smanjene snage. U cilju poboljšanja efikasnosti prenosa uvedeno je kodiranje, i dobijeni su rezultati iznad očekivanja. Prilikom drugog i narednih pokušaja da se paket prenese, u režimu smanjene snage, ako prenos nije uspeo iz prvog, vrši se kodiranje, tj. ekskluzivna ili (xor) operacija sa osmobicnim ključem (bajtom koji sadrži više različitih bitova, npr 10101010 ili 11001100) koji

se generiše na centralnoj stanici i šalje sa zahtevom za paketom, tako da se dekodiranje vrši odmah po prijemu (još jednom xor operacijom sa istim ključem). Ovakav metod bi se mogao sa promenljivim uspehom primeniti i u drugim slučajevima, kada mora da se koristi konstantna bodna brzina, a postoje različiti fizički izvori smetnji pri prenosu, jer se kodiranjem menja broj bit tranzicija a time i oblik, odnosno spektar signala, bez obzira na vrstu modulacije i medijum prenosa. Slične metode su poznate pod nazivom „beljenje podataka“ (*data whitening*) [67].

## 5.2. Ispitivanje ciklusa punjenja i pražnjenja solarnih ćelija

Potrošnja senzorskog uređaja u sleep modu iznosi oko 0,4 mA i dostiže maksimum od oko 40 mA prilikom transmisije od strane radio modema (u režimu pune snage). Ovo čini mogućim autonoman rad uređaja od oko mesec dana, ukoliko se ne priključi solarni panel, i ako se buđenje vrši na svakih nekoliko minuta (što je razuman period). Doplunjive baterije koje uređaj koristi mogu se povezati sa solarnim panelom koji omogućava dugoročno napajanje energijom. Solarni panel se sastoji od šest polikristalnih solarnih ćelija firme Conrad, nominalnog napona i struje od 5 V i 81 mA.

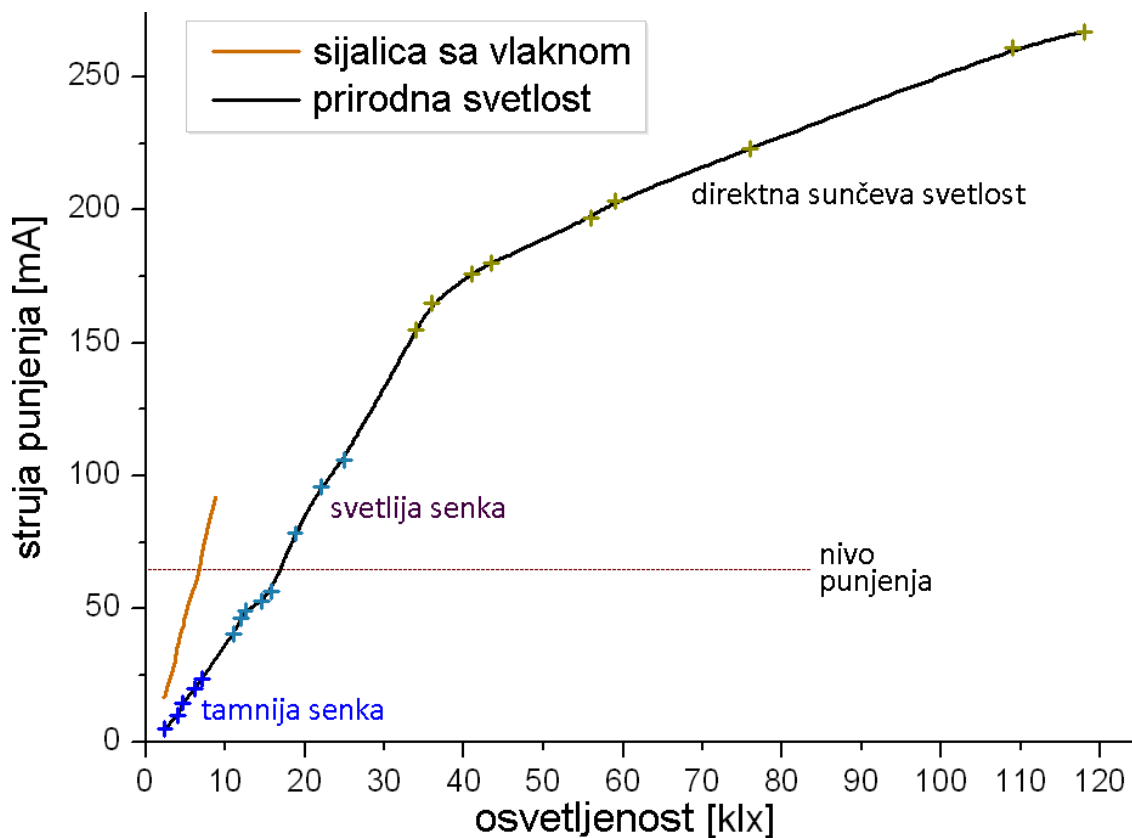
LT1512 kontroler punjenja može da puni baterija kada je panel osvetljen, bez obzira da li je napon na njemu manji od napona baterija, pod uslovom da se nalazi u opsegu od 2,7 do 25 V. Gornji limit punjenja baterija se podešava otpornim razdelnikom i iznosi oko 4,1 V, što odgovara 90-95% kapaciteta tri NiMH baterije. Punjenje i pražnjenje baterija podložno je histerezi, tako da jednokratno merenje napona ne može dati sasvim pouzdanu procenu energetskog stanja baterije. Tipičan histerezis prilikom dužih ciklusa pražnjenja i punjena baterija dat je na sl. 5.2.



**Slika 5.2** SOC u funkciji napona tipične NiMH baterije [44]

SOC označava stanje punjenja (*state of charge*) baterije, koji je uobičajeno kvantitativno merilo upotrebljivih energetskih rezervi. Histerezis koji se javlja u stvarnosti prilikom rada bežičnih senzorskih uređaja je manje predvidljiv, zato što ne postoje jasno definisani ciklusi punjenja i pražnjenja baterija, već se one sve vreme prazne (promenljivim intenzitetom, najmanje u sleep modu, više prilikom buđenja, a najviše prilikom radio komunikacije) a dopunjavaju se kada je solarni panel osvetljen dovoljnim intenzitetom svetlosti (što takođe varira).

Eksperimenti sa punjenjem uređaja solarnim panelom vršeni su tako što se prvo ispitivao njegov izlaz u funkciji osvetljenosti. Korišćen je Lux-UV-IR fotometar. Eksperimentalno ustanovljena zavisnost struje punjenja od osvetljenosti, za dnevnu i za svetlost sijalice, prikazana je na sl. 5.3. Rezultati merenja osvetljenosti dnevnim svetlom variraju od 2 klx u senci do 130 klx pod direktnom sunčevom svetlošću. Osvetljenost u područjima koja su solidno osvetljena prirodnom svetlošću, ali zaklonjena od direktne

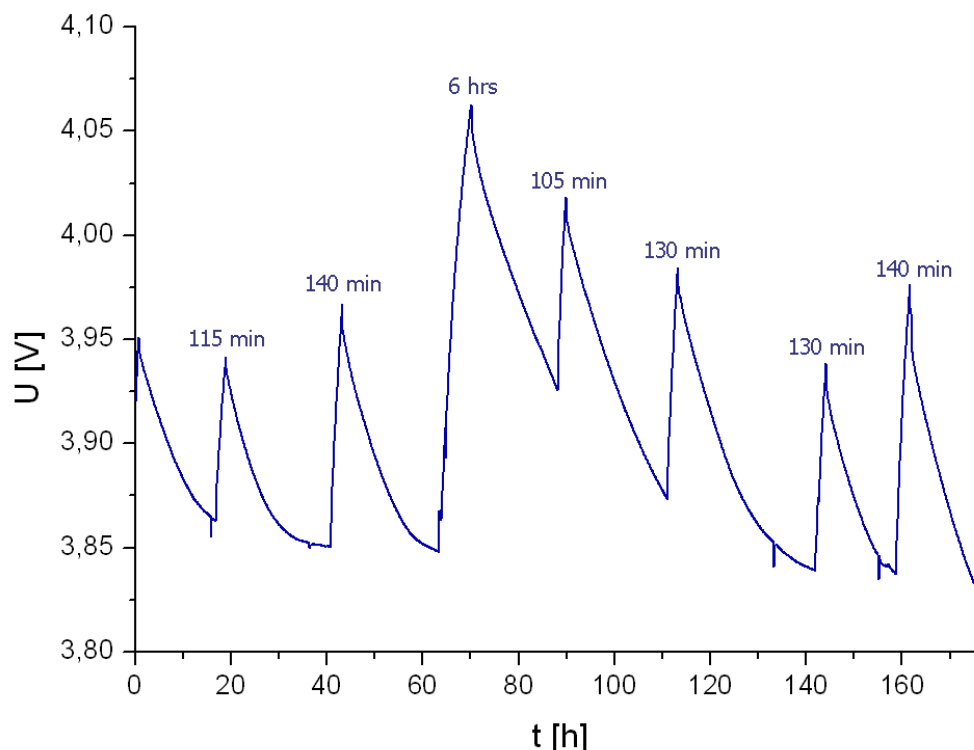


Slika 5.3 Struja punjenja panela kao funkcija osvetljenosti

sunčeve svetlosti, iznosi 10 do 30 klx tokom sredine dana. U cilju ispitivanja ciklusa punjenja baterija, panel je spojen sa senzorskim uređajem i osvetljiv svetlošću sijalice sa užarenim vlaknom unutar laboratorije. Sijalica sa užarenim vlaknom i dnevna svetlost nemaju isti spektar, što je uzeto u obzir. Panel se osvetljavao sa 5 klx, što je proizvodilo struju punjenja od 55 mA, što odgovara osvetljenosti od 14 klx dnevne svetlosti (relativno dobra osvetljenost u hladu). Period buđenja uređaja je podešen na 2,5 minuta. Oko dva merenja i transfera podataka dnevno je vršeno, uz oko dva sata rada solarnog panela. Dobijene varijacije napona baterija prikazane su na sl. 5.4 (vremena punjenja, koja odgovaraju usponima krive, su ispisana pored pikova). One pokazuju da oko 100 mAh dnevno, koliko je solarni panel doprinosa, jeste dovoljno da održi uređaj živim praktično neograničeno, ukoliko se on koristi na ovaj način (ne previše intenzivno). Predvidljivo je da bi i manji broj solarnih ćelija (manji panel) mogao da održava senzorski uređaj, u zavisnosti od intervala buđenja istog, i broja dnevnih merenja i transfera podataka.

Čak i u slučaju da napon baterija padne ispod minimuma koji je potreban za

regularan rad uređaja, punjenje je nezavisno i može ga vratiti u život, sa zanemarljivom verovatnoćom da se bilo kakav sistemski parametar ili rezultati prethodnog merenja obrišu. RAM sa sopstvenom ugrađenom baterijom čuva rezultate merenja, a važni parametri uređaja određeni kalibracijom čuvaju se u EEPROM-u.



**Slika 5.4** Ciklusi punjenja i pražnjenja baterija senzorskog uređaja pri upotrebi solarnog panela

Glavna ploča može da funkcioniše sa naponom napajanja od 3 V, a minimum modema je deklarisan na 3,4 V. Međutim, sa tri NiMH baterije, stvarni minimum za regularnu operaciju celog uređaja iznosi 3,5-3,55 V, što odgovara kolenu napon-SOC karakteristike za tri baterije [44,68]. Ovo koleno karakteristike je eksperimentalno potvrđeno za više brendova dopunjivih NiMH baterija kapaciteta 2400 do 2700 mAh. Kada napon padne niže, baterije ne mogu da održe vršnu struju potrebnu za emisiju od strane modema, te uređaj ne može regularno da komunicira (mada su određene operacije, kao što je kalibracija ili merenje pritiskom na reset taster, koji se nalazi na ploči, i dalje moguće).

Baterije se prazne i bez prisustva konzumirajućeg električnog kola [69]. Ovaj proces se naziva samopražnjenje (*self discharge*). Problem se može ublažiti korišćenjem *low self discharge* (LSD) baterija. Nažalost, u ovom trenutku ove baterije



su značajno skuplje, imaju nešto manji kapacitet, i, najvažnije, dvostruko manje garantovanih ciklusa punjenja i pražnjenja, tako da ne predstavljaju atraktivnu opciju za primenu u ovom bežičnom sistemu.

### 5.3. Ispitivanje poprečne osne osetljivosti akcelometara

Postoji nekoliko nesavršenosti koje utiču na prenosnu karakteristiku akcelometra. To su uglavnom nelinearnost (uključujući histerezis), poprečna ili unakrsna osna osetljivost, i dinamičke karakteristike. Dodatno, sve ove osobine su podložne temperaturnim varijacijama i starenju. Međutim, neki efekti se mogu eksperimentalno ispitati i modelovati tako da se softversko post-procesiranje može iskoristiti da se poboljša kvalitet mernih podataka.

Poprečna (ili unakrsna) osna osetljivost akcelometara je uglavnom izazvana nesavršenom orijentacijom njegovih delova. Stvari se pogoršavaju minijaturizacijom. Dok su novije generacije MEMS akcelometara mnogo praktičnije od klasičnih na mnogo načina, posebno u pogledu niske potrošnje energije, koja je ključna za sisteme sa baterijskim napajanjem, male seizmičke mase i elektrode su teže da se pravilno poravnaju. Zato komercijalni MEMS senzori ispoljavaju veću kros-aksijalnu osetljivost, tipično nekoliko procenata [70].

#### 5.3.1. Matematički model

Jednostavan metod za kalibraciju akcelometra, i za osetljivost na glavnoj osi, i za poprečnu osetljivost, koji se može primeniti u svakoj laboratoriji, jeste takozvani tilt test [71]. Akcelometar se stavlja u 6 međusobno normalnih položaja tako da gravitaciono ubrzanje deluje u njegovim pravcima +x, -x, +y, -y, +z i -z, i rezultati se analiziraju. Poprečna ili unakrsna osna osetljivost se kvantitativno definiše kao:

$$S_x = \frac{\sqrt{c_{xy}^2 + c_{xz}^2}}{c_{xx}} \quad (5.1a)$$

$$S_y = \frac{\sqrt{c_{yx}^2 + c_{yz}^2}}{c_{yy}} \quad (5.1b)$$

$$S_z = \frac{\sqrt{c_{zx}^2 + c_{zy}^2}}{c_{zz}} \quad (5.1c)$$

gde je  $S_i$  poprečna osetljivost za osu  $i$ , a  $c_{ij}$  su koeficijenti iz jednačina:

$$f_x = c_{xx}a_x + c_{xy}a_y + c_{xz}a_z + k_x \quad (5.2a)$$

$$f_y = c_{yx}a_x + c_{yy}a_y + c_{yz}a_z + k_y \quad (5.2b)$$

$$f_z = c_{zx}a_x + c_{zy}a_y + c_{zz}a_z + k_z \quad (5.2c)$$

koji opisuju uticaj eksitacije po osi  $j$  na izlaz  $f_i$  ose  $i$ . Koeficijenti obeleženi sa  $k_i$  označavaju nulte ofsete (izlaz akcelerometra po osi  $i$  kada nije izložen nikakvoj pobudi). Sama poprečna osetljivost je parametar koji opisuje kvalitet senzora ali se ne može iskoristiti za softversku obradu signala. U tu svrhu je potrebno odrediti sve koeficijente iz jednačina (5.2).

Izlazni signal senzora sa poznatim koeficijentima  $c_{ij}$  i  $k_i$  se može softverski korigovati. Sistem od tri jednačine sa tri nepoznate se rešava i inverzne jednačine glase:

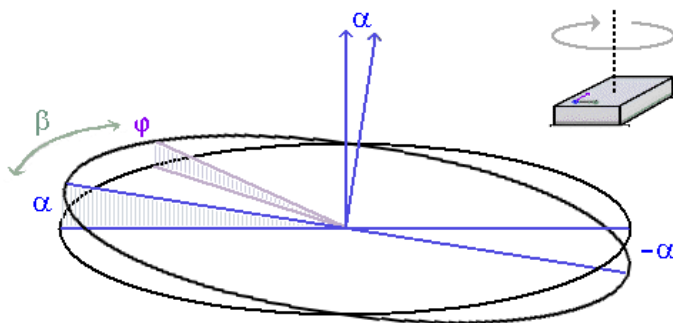
$$a_x = \frac{(f_x - k_x)(c_{yy}c_{zz} - c_{yz}c_{zy}) + (f_y - k_y)(c_{xz}c_{zy} - c_{xy}c_{zz}) + (f_z - k_z)(c_{xy}c_{yz} - c_{xz}c_{yy})}{c_{xx}(c_{yy}c_{zz} - c_{yz}c_{zy}) + c_{yx}(c_{xz}c_{zy} - c_{xy}c_{zz}) + c_{zx}(c_{xy}c_{yz} - c_{xz}c_{yy})} \quad (5.3a)$$

$$a_y = \frac{(f_x - k_x)(c_{yz}c_{zx} - c_{yx}c_{zz}) + (f_y - k_y)(c_{xx}c_{zz} - c_{xz}c_{zx}) + (f_z - k_z)(c_{xz}c_{yx} - c_{xx}c_{yz})}{c_{xy}(c_{yz}c_{zx} - c_{yx}c_{zz}) + c_{yy}(c_{xx}c_{zz} - c_{xz}c_{zx}) + c_{zy}(c_{xz}c_{yx} - c_{xx}c_{yz})} \quad (5.3b)$$

$$a_z = \frac{(f_x - k_x)(c_{yx}c_{zy} - c_{yy}c_{zx}) + (f_y - k_y)(c_{xy}c_{zx} - c_{xx}c_{zy}) + (f_z - k_z)(c_{xx}c_{yy} - c_{xy}c_{yx})}{c_{xz}(c_{yx}c_{zy} - c_{yy}c_{zx}) + c_{yz}(c_{xy}c_{zx} - c_{xx}c_{zy}) + c_{zz}(c_{xx}c_{yy} - c_{xy}c_{yx})} \quad (5.3c)$$

### 5.3.2. Tilt test u 24 pozicije

Ako postoji mali ugao  $\alpha$  između stvarne orijentacije gravitacionog ubrzanja i glavne ose troosnog akcelerometra, poprečne ose su izložene gravitacionim ubrzanjima reda veličine  $g \cdot \sin(\alpha)$ . Ova funkcija ima velike varijacije oko nule i ovo nas sprečava da



Slika 5.5 Rotacija akcelerometra

dobijemo pouzdane koeficijente poprečne osne osetljivosti običnim tilt testom. Posmatrajmo poprečnu osu normalnu na onu koja je okrenuta za mali ugao  $\alpha$  u odnosu na vertikalu, kao što je prikazano na sl. 5.5. Ugao

između ove ose i horizontalne ravni varira u opsegu od  $+\alpha$  do  $-\alpha$  zavisno od ugla rotacije oko glavne ose  $\beta$ , tako da je:

$$\operatorname{tg}\varphi = \operatorname{tg}\alpha \cos \beta \quad (5.4)$$

$$\varphi \approx \alpha \cos \beta \quad (5.5)$$

gde se  $\beta$  računa od pozicije sa najvećom visinom. Jednačina (5.4) se može aproksimirati jednačinom (5.5) pod uslovom da je  $\alpha$  malo. Uzimajući  $N$  proizvoljnih vrednosti  $\beta$  koje se razlikuju za  $\Delta\beta=2\pi/N$  (uniformno distribuirane tačke na krugu), odgovarajuća prosečna vrednost  $\sin(\varphi)$  i  $\cos(\varphi)$  postaje nula. Rotacija akcelometra dakle potire izlaz akcelometra koji potiče od glavne osne osetljivosti poprečnih osa na lateralnu komponentu gravitacije  $g \cdot \sin(\varphi)$ , ali ne potire unutrašnju kros-aksijalnu osetljivost čije postojanje dovodi do približno konstantnog doprinosa izlaznom signalu.

Jedan uobičajen metod kalibracije akcelometra na poprečnu osetljivost jeste postavljanje senzora u dva suprotna statična položaja u svakoj osnovnoj ravni (njih šest) [72]. Ovo je ekvivalentno opisanoj rotaciji sa  $N=2$ , odnosno  $\Delta\beta=\pi$ . Mana svakog statičnog tilt testa je nemogućnost da se integrisana ploča sa senzorom postavi u ponovljiv položaj sa dobrom ortogonalnošću u odnosu na druge merne položaje, usled elastičnosti elemenata kućišta u koje je postavljena, elastičnosti štampane ploče, i neravnina na radnoj površini i površini kućišta (koje je dodatno podložno neelastičnim deformacijama tokom vremena). Drugi metod je dinamički: kreiranje varijacija u izlazu jedne ose rotacijom akcelometra u normalnoj ravni, čime se izlaže inercijalnim silama u dva poprečna pravca, i snimanje rezultujućeg signala, koji se zatim analizira [73,74]. Mana dinamičkih metoda je potreba za kompleksnom opremom, koja ne postoji u svakoj laboratoriji.

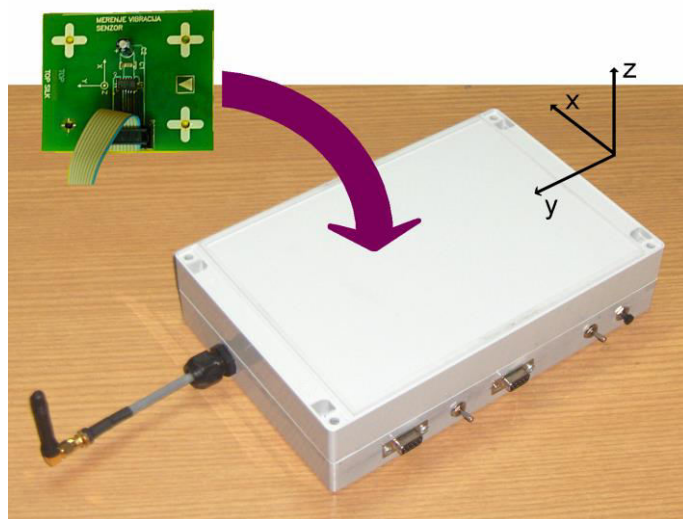
Tilt test u 24 pozicije izvodi praktično isto što i opisani test u 12 pozicija ( $N=2$ ,  $\Delta\beta=\pi$ ), ali, četiri tačke rotacije ( $N=4$ ,  $\Delta\beta=\pi/2$ ) nisu bolje samo zato što postoji više podataka za usrednjavanje. Operater može da prati vrednosti i ponovi eksperiment ako ove ne prate pravilo simetrije. Četiri proizvoljna ugla koja se razlikuju za  $90^\circ$  treba da proizvedu izlaz jednak redom:  $f_0+k \cdot \cos(\beta)$ ,  $f_0-k \cdot \sin(\beta)$ ,  $f_0-k \cdot \cos(\beta)$ , i  $f_0+k \cdot \sin(\beta)$ , gde je  $f_0$  srednja vrednost, a  $k$  konstanta koja zavisi od ugla  $\alpha$  i kros-aksijalne osetljivosti. Drugim rečima, srednja vrednost prve i treće pozicije mora da bude približno jednaka srednjoj vrednosti dobijenoj u drugoj i četvrtoj poziciji (do ostvarljivog nivoa preciznosti) ili merenje treba da se ponovi. Ova vrsta kontrole se ne može

implementirati ako se koristi test sa 12 pozicija.

Tilt test se može izvoditi i sa dvoosnim, a ne samo troosnim akcelerometrima. Međutim, njihova manja kros-aksijalna osetljivost čini ga manje pogodnim. Jednoosni akcelerometri imaju previše nisku poprečnu osetljivost da bi ovaj test imao smisla u većini slučajeva.

### 5.3.3. Eksperimenti

Iskorišćen je opseg akcelerometra od  $\pm 2g$ , kao i obično, i najniža frekvencija samplovanja od 40 Hz, radi najnižeg nivoa šuma. Akcelerometar je postavljen u kućište kao što je prikazano na sl. 5.6. Tri senzorska uređaja su testirana pomeranjem celog kućišta manuelno po običnim površinama kao što su kancelarijski stolovi. Nikakva druga merna oprema nije korišćena.



Slika 5.6 Senzorski uređaj u plastičnom kućištu

Senzorski uređaji su programirani da izvode serije od više stotina merenja i zatim usrednje rezultat za svaki položaj. Podaci se mogu preneti radio modemom ili očitati sa displeja. Ono što je važno u eksperimentu ovog tipa je da se akcelerometru dozvoli minimalna sloboda kretanja u referentnom sistemu kućišta. Akcelerometarska pločica je pričvršćena zavrtnjima čija su ležišta usađena u kućište.

Preliminarni eksperimenti sa običnim tilt testom u šest pozicija su dali slabe rezultate. Javljale su se varijacije u izmerenim koeficijentima  $c_{ij}$  iz jednačine (5.2) koje odgovaraju 1% g. Rezultati tilt testa sa 24 pozicije su prikazani u tabeli 4, izraženi u sirovim kvantima (digitalni izlaz akcelerometra; 1024 kvanta nominalno odgovaraju ubrzanju od g). Ovi eksperimenti su izvedeni na metalnoj površini, drvenoj površini, i na blago nagnutoj drvenoj površini ( $1,5^\circ$ ) da bi se pokazalo da nagib ne utiče bitno na rezultate. Iz razloga jasnoće, rezultati iz različitih eksperimenata su prikazani u

različitim redovima unutar iste ćelije u tablici.  $\bar{f}(i)$  predstavlja srednji izlazni signal od izlaganja akcelerometra +g i -g pobudi u pravcu  $i$ . Ovo bi bio nulti ofset ako pretpostavimo da je funkcija prenosa linearna, dok  $\Delta f(i)$  predstavlja razliku između pokazivanja akcelerometra u dva suprotna položaja, što odgovara rasponu skale od 2g. Kolone predstavljaju izlaz, pokazujući osu i redni broj senzora. Znak je ispušten za glavne osne osetljivosti, koje su uvek pozitivne i imaju nominalnu vrednost od 2048. Najgori rezultat u redu (najveća devijacija od srednje vrednosti) je obeležen zvezdicom.

**Tabela 4** Rezultati eksperimenata na tri različite podloge

	X1	Y1	Z1	X2	Y2	Z2	X3	Y3	Z3
$\bar{f}(z)$	-2,12	+3,16	+0,59	+4,68	-0,12	-21,69	-19,57	-10,34	+18,02
	+1,05	-0,99*	+0,18	+1,92	-0,30	-21,86	-20,61	-10,15	+18,82
	-1,52	+4,30	+1,38	+1,34	+1,68	-21,44	-19,53	-9,28	+18,60
$\Delta f(z)$	<b>-32,16</b>	<b>-18,46</b>	2069,1	<b>+9,35</b>	<b>+4,20</b>	2067,7	<b>-9,69</b>	<b>+7,36*</b>	2071,4
	<b>-31,65</b>	<b>-15,85</b>	2070,1	<b>+12,64</b>	<b>+5,72</b>	2069,0	<b>-9,95</b>	<b>+3,75</b>	2071,6
	<b>-31,30</b>	<b>-15,20</b>	2069,4	<b>+10,20</b>	<b>+3,37</b>	2068,6	<b>-8,59</b>	<b>+1,38</b>	2071,4
$\bar{f}(y)$	-3,56	+4,66	+2,79	+0,16	+10,45	-18,54	-20,31	-17,40	+20,85
	+1,26	+4,71	+2,23	+7,01*	+9,66	-20,05	-19,30	-17,59	+24,46
	-8,48	+4,60	+3,67	-4,69	+10,19	-18,37	-26,39	-17,55	+20,79
$\Delta f(y)$	<b>+27,76</b>	2057,7	<b>+25,75</b>	<b>+64,44</b>	2043,8	<b>+5,11</b>	<b>+38,53</b>	2044,9	<b>+10,23</b>
	<b>+38,07</b>	2057,2	<b>+21,64</b>	<b>+69,99</b>	2043,8	<b>-3,05</b>	<b>+38,59</b>	2045,1	<b>-7,47*</b>
	<b>+30,44</b>	2057,2	<b>+10,35</b>	<b>+64,13</b>	2044,3	<b>-5,84</b>	<b>+49,52</b>	2045,0	<b>+2,63</b>
$\bar{f}(x)$	+4,36	-2,30	-5,37	-2,13	-5,54	-26,61	-18,91	-16,12	+12,50
	+4,45	-2,44	-6,13	-1,18	+0,18	-24,56	-19,06	-8,70*	+15,08
	+4,64	-1,66	-4,16	-1,80	-4,51	-24,30	-19,02	-15,43	+15,26
$\Delta f(x)$	2031,9	<b>+3,65</b>	<b>+3,36</b>	2029,7	<b>-30,31</b>	<b>-40,75</b>	2017,0	<b>-7,00</b>	<b>-17,98</b>
	2030,0	<b>+5,01</b>	<b>+3,83</b>	2031,2	<b>-16,01</b>	<b>-50,19</b>	2017,6	<b>+8,48</b>	<b>-22,59</b>
	2031,2	<b>+10,29</b>	<b>+18,86</b>	2030,5	<b>-28,56</b>	<b>-25,47*</b>	2016,5	<b>-3,55</b>	<b>-3,53</b>

Strane kućišta su veoma različite, kao što se vidi na sl. 5.6. Z strana (u x-y ravni), na koju je kućište položeno na slici, je relativno velika i ravna. Ovo čini da koeficijenti povezani sa z eksitacijom, iz jednačina (5.2), budu najpouzdaniji (ovi koeficijenti zapravo figurišu u x i y poprečnim osetljivostima). Antena na y strani može da se demontira kada se kućište okreće na tu stranu ili kućište može prosto da se ostavi da visi preko ivice površine, pošto se centar mase i dalje može postaviti iznad kontaktne površine, da ne pretil opasnost od pada. Ipak, totalna površina ove strane je manja što uzrokuje slabije ponovljive rezultate za koeficijente koji su u vezi sa eksitacijom po

ovoj osi (iste neravnine izazivaju veće razlike u položaju kućišta). Konačno, prekidači na prednjoj, x strani, čine da uređaj ne može da stoji na dva oslonca, a ovog puta centar mase se ne može postaviti iznad kontaktne površine, pa operater mora da drži kućište manuelno. Zato se najgori rezultati dobijaju za koeficijente u vezi sa x pobudom. Kućišta su prototipovi korišćeni intenzivno tokom par godina u momentu izvođenja eksperimenata, tako da ima razloga da se pretpostavi da bi bolji rezultati mogli da se dobiju korišćenjem novije opreme.

Vidimo da za z eksitacione (najbolju stranu kućišta) koeficijente, maksimalna devijacija od srednje vrednosti iz svih eksperimenata iznosi oko 3 kvanta (jedan odgovara približno  $10^{-3}$  g), ili 0,3%. Za koeficijente u vezi sa y eksitacijom, maksimalna devijacija iznosi oko 9 kvanata (0,9%), dok za najgori slučaj, koeficijente u vezi sa x eksitacijom, odstupanje je oko 13 kvanata (1,3%). Zaključak je da se ovim relativno jednostavnim eksperimentima i naknadnom softverskom korekcijom unakrsna osna osetljivost može svesti na značajno niži nivo od osnovne.

Najveća izmerena poprečna osetljivost definisana jednačinom (5.1) iznosi 3,3% za x osu senzora broj 2. Zaključak je da ovo odgovara specifikaciji proizvođača [37] (maksimalna kros-aksijalna osetljivost od 3,5%).

## **6. SIMULTANOST PODATAKA U BEŽIČNOJ MREŽI**

Jedan od važnih ciljeva u realizaciji bežičnog mernog sistema za merenja u građevinarstvu jeste postizanje odgovarajuće vremenske sinhronizacije između mernih podataka koji potiču iz različitih tačaka. Za praktične potrebe monitoringa kapitalnih građevinskih objekata potrebno je da usaglašenost podataka bude garantovana do nivoa zanemarljivog dela periode najviših harmonika koji se proučavaju, a oni su reda veličine ispod 100 Hz (maksimum frekvencija od interesa u slučaju brana i mostova tipično iznosi 20-50 Hz), što znači da bi praktično garantovana simultanost ispod jedne milisekunde bila zadovoljavajuća za ovu primenu.

### **6.1. Metode sinhronizacije u kompjuterskim i senzorskim mrežama**

#### **6.1.1. Izvori nepreciznosti satova**

Postoje dva glavna izvora greške vremenske sinhronizacije u distribuiranim sistemima (kao što su bežične senzorske mreže). Prvo, vreme propagacije poruka nije dovoljno determinisano, a pogotovo u slučajevima kada se koristi radio komunikacija između udaljenih uređaja. Drugo, stabilnosti tajmera (satova, koji su na uređajima niskog nivoa najčešće povezani sa sistemskim tajmerom, koji daje takt procesoru) su limitirane brojnim fizičkim faktorima, a čak i u slučaju da su njihove frekvencije perfektno kalibrisane u nekom trenutku, pomeranja (tzv. driftovi) moraju da se dogode i vremensko odstupanje (tzv. ofset) se akumulira vremenom, uslovljavajući potrebu za razmenom novih poruka da bi se uređaji međusobno ponovo sinhronizovali. Mrežni protokoli koji se bave ovim problemom izvode serije akcija koje su bazirane na distribuciji vremenskih žigova iz referentnih izvora, što se naziva ofset sinhronizacija, i štelovanju dinamičkih koeficijenata koji se koriste za kalibraciju satova, što se naziva sinhronizacija brzine [75,76].

#### **6.1.2. Spisak često korišćenih pojmova u metodama sinhronizacije**

Kontinualna ili implicitna sinhronizacija znači da se paketi podataka koji su

vezani za sinhronizaciju dodaju („kače”) na postojeće pakete koji se (kontinualno) šalju kroz mrežu (i nezavisno procesiraju, po potrebi). Ovo se u literaturi često označava i rečju *piggy-backed*, što u doslovnom prevodu znači „na krkače”. Suprotnost je sinhronizacija na zahtev, koja se takođe naziva eksplicitna, ili *event-triggered* (inicira se nekim događajem odnosno okidačem ili trigerom). Eksplicitna sinhronizacija zahteva dodatne komunikacione pakete koji služe isključivo (ili primarno) u ove svrhe. Ona je zato zahtevnija u smislu energetske potrebe, što je značajno kod bežičnih uređaja [77].

Eksterna sinhronizacija se bazira na jednom referentnom satu, bez obzira da li dolazi sa jednog uređaja unutar mreže (u kom slučaju reč eksterna ne treba bukvalno da se shvati) ili ne. Cilj eksterne sinhronizacije je da učini da svi satovi u mreži pokazuju koliko je moguće blizu referentnom satu. Suprotnost je interna sinhronizacija, kod koje ne postoji nikakvo referentno vreme. Njen cilj je da učini da pokazivanja svih satova unutar mreže budu što sličnija [78].

Unidirekciona sinhronizacija je procedura u kojoj se sat ciljnog uređaja podešava prema satu izvornog uređaja, dakle postoji jasan smer kretanja informacije. Tipično transmisiono vreme se oduzima od vrednosti po prijemu. Bidirekciona, često zvana i *pairwise* (parska), sinhronizacija je bazirana na proračunu takozvanog *round-trip* kašnjenja, odnosno vremena potrebnog da poruka otputuje sa izvornog uređaja do ciljnog i da se povratna poruka detektuje na izvornom uređaju (kružno putovanje). Prednost je što se meri stvarno fizičko vreme (naravno, ograničeno preciznošću lokalnog sata), što je nemoguće u dve odvojene tačke (kod unidirekcionih sinhronizacija). Iako se ne može odrediti tačna raspodela vremena i nesigurnosti vremena putovanja između smerova, za povratnu putanju može se tačno odrediti i srednja vrednost i standardno i maksimalno odstupanje vremena propagacije (*round-trip*). Pošto se obično radi sa uređajima istog tipa, za tipično vreme propagacije uzima se polovina *round-trip* vremena. I drugi uređaji u mreži mogu imati koristi od bidirekcionih sinhronizacionih poruka između dva čvora. Oni mogu da koriste razmenjene vremenske žigove da bi sinhronizovali svoje satove bez bilo kakve transmisije podataka, samo prijemom. Ovo se uobičajeno naziva *eavesdropping* (prisluškivanje) [79].

Transformacija vremenske skale spada u *backwards*, odnosno *post-facto* ili naknadne (unazad), sinhronizacione tehnike. Satovi nisu sinhronizovani za vreme



dogadjanja koje se monitoriše, ali se sinhronizacija skupljenih podataka vrši naknadno u zavisnosti od relevantnih parametara (ustanovljenih frekvencija tajmera i medjusobnih ofseta). Iako zahteva manje energije (manje dodatne komunikacije nego u slučaju sinhronizacije na zahtev), povećava potrebu za procesiranjem podataka, i najbolje je izvršiti je na sistemu visokog nivoa, kao što je PC. Post-facto sinhronizacija se nekada naziva i reaktivnom, što je suprotno od proaktivne sinhronizacije koja se vrši unapred [80].

*Reference Broadcast Synchronization* (RBS) [81,82] ili sinhronizacija putem referentnog ili prozivnog signala je termin koji se asocira isključivo sa bežičnim mrežama. Centralni server emituje referentni signal (*reference broadcast*, takođe poznat i pod nazivom *beacon*, odnosno svetionik, ili signal za navođenje) a prijemni uređaji resetuju svoje tajmere po prijemu. Ovaj tip poruke se naziva *multicast* (šire raspoređen), što je suprotno *unicast* porukama (koje imaju jednog pošiljaoca i jednog primaoca). Propagacione neodređenosti, vezane za transmisiju poruke (izlazno vreme kroz fizički sloj, koji se sastoji od interfejsa između procesora i komunikacionog modula, modema ili transsivera, i antene) se potiru jer se radi o jednom signalu. Neodređenosti vezane za prijem poruke su obično za red veličine manje i na ovaj način se ofset između čvorova praktično potire. Logičko vreme je na ovaj način nešto bolje sinhronizovano nego fizičko vreme. Logičko vreme predstavlja vreme unutar proizvoljnog vremenskog perioda, kao što je merni ciklus, dok fizičko vreme predstavlja stvarno vreme u spoljnom svetu. Sekundarne RBS procedure se mogu ponavljati u multihop mrežama (gde ne postoji direktna veza između izvora RBS-a i svih uređaja), gde sinhronizovani uređaji služe kao habovi (centralne stanice) lokalnih klastera (nižeg nivoa). Pouzdanost sinhronizacije opada sa brojem hopova (prenosa RBS-a).

Puls u sekundi (PPS) globalnog sistema za pozicioniranje (GPS) predstavlja odličan izvor vremenskih žigova visoke rezolucije i tačnosti [83]. Kako se očekuje da cene GPS risivera padaju u doglednoj budućnosti, ovaj metod će verovatno dobiti na popularnosti. Ipak, u ovom trenutku, osim potrebe da se jeftini bežični senzorski uređaji opreme dodatnim hardverom čija cena nije zanemarljiva, postoji problem dostupnosti signala unutar građevina, tako da se ovo inače efektivno rešenje uglavnom izbegava. Osnovni problem kod dostupnosti GPS signala predstavlja činjenica da se ovaj signal emituje iz satelita veoma malom snagom. Čak i malo gušće krošnje drveća mogu da

budu nepremostiva prepreka.

### 6.1.3. Uobičajeni mrežni algoritmi sinhronizacije

Rane kompjuterske mreže koristile su dve varijacije sinhronizacije putem centralizovanog servera, poznate kao Kristijanova [84] i Berkli [85] sinhronizacija. Hab sistema (centralna stanica) šalje vremenske žigove uređajima u mreži na zahtev. Očekivano vreme propagacije se oduzima prilikom prijema. Alternativno, višestruki paketi sa vremenskim žigovima se šalju u odgovoru na jedan zahtev, i najbrže pristigli paket se uzima u obzir (da bi se izbegla mogućnost povremenih nepredviđenih zastoja). U Kristijanovoj varijaciji algoritma, hab služi kao izvor vremena (njegov sat se uzima kao referentni eksterni izvor), dok se u Berkli varijaciji podaci skupljaju sa različitih čvorova, usrednjavaju (opciono sa težinskim faktorima), i usrednjena informacija se šalje nazad.

Jedna od najčešće korišćenih metoda u savremenim kompjuterskim mrežama jeste mrežni vremenski protokol ili NTP (*network time protocol*) [86]. Čvorovi mreže se dele u različite slojeve (*layers*), počevši od referentnih satova koji su na vrhu (takozvani „stratum 0” sloj). Nacionalne laboratorije su zadužene za kalibraciju ovih satova. Mrežni čvorovi nižeg nivoa sinhronizuju se na bazi poruka koje primaju sa uređaja istog ili višeg nivoa. Razmena poruka unutar mreže ne sadrži samo vremenske žigove, već i razne druge podatke, kao što su preciznost i kvalitet tajmera na uređaju, vreme njegove poslednje sinhronizacije, pouzdanost njegove poslednje sinhronizacije, i očekivano vreme putovanja poruke. Ciljni uređaj odlučuje da li da izvrši korekciju svog sata, i za koliko, ili ne, bazirano ne samo na zadnjoj primljenoj poruci već ispitujući set poruka pristiglih u većem vremenskom intervalu. NTP poruke se testiraju na „smislenost” i one koje izgledaju kao gruba greška se odbacuju u procesu interpolacije, tako da je algoritam robustan.

Precizni vremenski protokol ili PTP (*precision time protocol*), poznat i kao IEEE-1588 [87,88], je kompleksniji protokol baziran na istom principu. Veći broj tipova poruka se koristi, veći broj parametara se prosleđuje, i uređaji imaju različite funkcije, a ne samo klijent, server, ili klijent-i-server, što je slučaj u NTP-u. Ipak, glavna prednost ovog protokola nisu kompleksne softverske metode već „hardverska asistencija“. To znači da postoji specijalizovani hardver koji se instalira na mrežne kartice i druge

uređaje, koji određuje precizna (ispod mikrosekunde) vremena fizičke emisije poruka (tzv. prolaska kroz „fizički sloj“ mreže), koja bi inače bila nedostupna glavnim procesorima mrežnih uređaja. Radi poređenja, tipičan NTP klijent ostvaruje preciznost sinhronizacije reda veličine milisekunde, u poređenju sa PTP-om, kod koga su sinhronizacije reda veličine ispod mikrosekunde.

#### **6.1.4. Sinhronizacija u bežičnim mrežama**

Veliki broj protokola je osmišljen radi upotrebe isključivo u WSN.

Sinhronizacija putem vremenskog žiga ili vremenskog označavanja (*time-stamp synchronization* ili TSS) [89] je metod promene vremenskih žigova poruka koje prolaze kroz multihop mrežu, bazirano na vremenu prijema, procenjenom vremenu puta, i vremenu koje poruka provede u „redu“ (lokalnom baferu, odnosno steku, za prijem pristiglih paketa), odnosno *queue*-u, pre nego što se prosledi. Satovi nisu stvarno sinhronizovani tako da se ovde radi o formi *post-facto* odnosno naknadne sinhronizacije. Varijacija na ovu temu je RITS (*routing integrated time synchronization*) [90], kod koga se izvodi dodatna kompenzacija zbog razlike u brzinama rada različitih tajmera. Ova razlika se procenjuje na osnovu istorije razmenjenih poruka sa vremenskim žigovima iz bidirekciono komunikacije (gde se mogu uporediti *round-trip* vremena proračunata na različitim čvorovima).

Laka vremenska sinhronizacija (*lightweight time synchronization* ili LTS) [91] i *timing-sync* protokol za senzorske mreže (TPSN) [92] su bazirani na bidirekcionoj sinhronizaciji parova uređaja koji razmenjuju poruke unutar multihop mreže, gde jedna ili više referentnih tačaka postoje (čiji su tajmeri deklarirani kao referentni). Uređaji na nižim nivoima (više hopova udaljeni od referenci), upućuju zahteve za sinhronizacijom prema uređajima na višem nivou. Kod LTS-a, uređaji proveravaju da li drugi uređaji na istom nivou imaju aktuelne zahteve koji nisu izvršeni (*pending requests*), i sinhronizuju se sa njima umesto da šalju nove multihop zahteve. Kod TPSN-a, uređaji na nižem nivou prisluškuju (*eavesdrop*) sinhronizaciju uređaja na višem nivou, i zatim šalju zahtev za sinhronizacijom u pogodnom momentu.

Tiny-sync i Mini-sync [93] takođe rade na principu bidirekciono sinhronizacije. Oni koriste statistiku da poredi vremenske žigove koji se razmenjuju između parova uređaja. Vremenski žigovi odstupaju sve više tokom vremena, zbog razlika u brzinama

satova (koje su dodatno podložne varijacijama usled promene raznih fizičkih parametara), tako da se odnos frekvencija i inicijalni ofset mogu proceniti kreiranjem optimalne prave bazirane na većem broju parova vremenskih žigova sa oba uređaja (tzv. linearna regresija). Različiti algoritmi se koriste da procene najverovatniji odnos frekvencija i ofset.

Fleksibilna laka vremenska sinhronizacija (*scalable lightweight time-synchronization protocol*, ili SLTP) [94], koristi metod takozvanog pasivnog grupisanja (*passive clustering*) kao prvi korak, a zatim izvodi sinhronizaciju slično LTS-u. Na zahtev, kreira se topologija mreže (pasivno grupisanje, odnosno trasiranje), pri čemu se uređaji dele na grupe (klastere) i dodeljuju im se funkcije habova klastera (referentnih tačaka, koje dakle nisu fiksne kao kod originalnog LTS-a), članova klastera, ili *gateway*-a odnosno kapija između klastera. Upotrebljava se i linearna regresija, kao kod *tiny-synca* i *mini-synca*, za poređenje i usklađivanje satova prilikom bidirekcionе komunikacije.

Tsync protokol [95] koristi dve tehnike koje se nazivaju hijerarhijsko referenciranje (*hierarchy referencing time synchronization* ili HRTS) i individualni vremenski zahtevi (*individual-based time request* ili ITR). Glavni uređaj emituje RBS signal koji primaju uređaji najvišeg nivoa (lejera). Jedan od njih se slučajno izabira za bidirekcionu razmenu vremenskih žigova. Propagaciono kašnjenje se procenjuje na osnovu ovog procesa i novi RBS od strane centralnog uređaja se šalje dajući ciljnim uređajima referentnu vrednost vremena i pretpostavljeno vreme propagacije (smatra se da su uređaji hardverski identični te da se očekuje približno isto vreme propagacije). Kada se proces završi, uređaji iz prvog nivoa ponavljaju proceduru prema klasterima nižeg nivoa. Proces se rekurzivno ponavlja dok svi uređaji u mreži nisu sinhronizovani. ITR je komplementarni mehanizam u Tsync-u, koji se koristi za sinhronizaciju na zahtev, jednog uređaja ili grupe uređaja. Uređaji koji zahtevaju sinhronizaciju šalju zahtev koji se multihop prenosi do centralne stanice (glavnog uređaja) i referentni vremenski žig se prenosi nazad istim putem. HRTS i ITR u originalnom obliku koriste dva različita kanala za razmenu poruka, ali moguća je implementacija i na jednom kanalu, sa nešto lošijim rezultatima.

Intervalska sinhronizacija (*interval based synchronization* ili IBS) [96] koristi procenu granica brzine satova (njihovih minimalnih i maksimalnih vrednosti) da odredi

granice u kojima se vreme nalazi. Umesto razmene vremenskih žigova, čvorovi komuniciraju razmenjujući podatke o minimalnom i maksimalnom vremenu (tzv. *interval stamps* ili intervalski žigovi, bazirani na najgorem slučaju devijacije frekvencije sata). Ovo se naziva i intervalski model, koji je suprotan instant („tačkastom“) modelu. Iako neki autori tvrde da je kombinacija „garantovanih“ granica za lokalne satove optimalnija nego upotreba pojedinačnih vremenskih žigova uz estimaciju njihovog kvaliteta (procenu distribucije verovatnoće i standardne i maksimalne devijacije), suština oba metoda je ista, a u stvarnosti ne postoje stopostotne garancije za granice kod *bounded-drift* modela sata (modela limitarnog odstupanja odnosno limitiranog drifta frekvencije).

Asinhrona difuzija [97] je protokol kod koga svi uređaji periodično šalju signal susedima tražeći pokazivanja njihovih satova (vremenske žigove). Slično kao kod Berkli algoritma, vrši se usrednjavanje ovih vrednosti i one se putem RBS-a transmituju susedima (svim uređajima u dometu ili unutar klastera), koji vrše sinhronizaciju na novu vrednost. Proces se ponavlja širom mreže.

Protokol sinhronizacije putem preplavlivanja (*flooding time synchronization protocol* ili FTSP) [98] koristi tzv. preplavlivanje (*flooding*) mreže odnosno slanje velikog broja paketa (slično RBS-u) sa referentnog uređaja, u kombinaciji sa pamćenjem parova vremenskih žigova i linearnom regresijom koja se koristi kod *tiny-sync-a*. *PulseSync* [99] je sličan protokol, baziran na plavljenju mreže velikom brzinom, da se ne dozvoli akumulacija tzv. nagiba (*skew*). *Skew* označava akumuliranu grešku sata baziranu na driftu frekvencije tajmera (varijacijama tokom posmatranog vremena). I dok oba protokola imaju preciznost koja opada sa brojem hopova, *PulseSync* preciznost opada nešto sporije.

Kod protokola vremenske difuzije (*time diffusion protocol* ili TDP) [100], referentni uređaj šalje svoj vremenski žig, i prima odgovor od uređaja koji su u dometu na osnovu čega određuje vremena povratnih poruka (*round-trip*) i standardne devijacije. Procedura se zatim ponavlja kroz multihop mrežu dok svi uređaji nisu primili bar jednu poruku ovog tipa. Suština metode je da se uređaji koji su primili višestruke poruke sinhronizuju na srednju vrednost, pri čemu se koriste težinski koeficijenti. Drugi termin koji se koristi da se opišu slične tehnike je gradijentni protokol vremenske sinhronizacije (*gradient time synchronization protocol* ili GTSP) [101]. Ovi algoritmi su

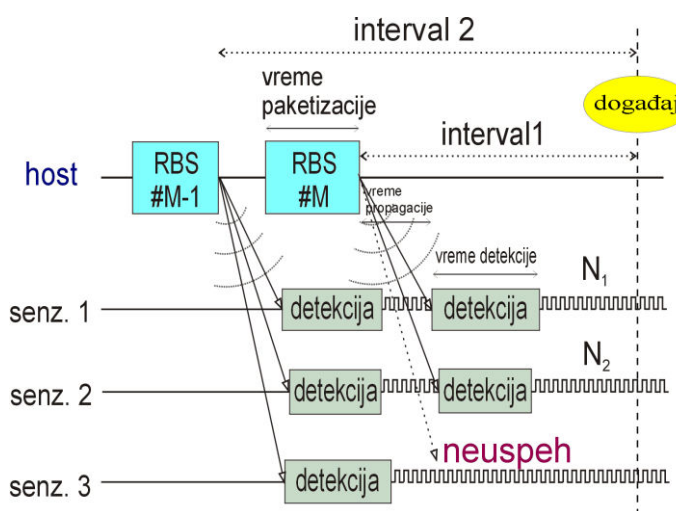
robusni jer ne zahtevaju jedan referentni čvor, definisanu topologiju, niti regularnu operaciju svih uređaja.

Konsenzus sinhronizacija (*consensus clock synchronization* ili CCS) [102] je interna sinhronizaciona metoda. Svi uređaji emituju unidirekzione poruke sa vremenskim žigovima i primajući uređaji pokušavaju da procene koliko iznosi vrednost „dogovorenog” vremena u mreži.

Kao što vidimo, sve opisane metode, i druge koje postoje, nisu strogo razdvojene, već se dosta preklapaju i teško je povući granicu između njih i razvrstati određen algoritam u samo jednu kategoriju.

## 6.2. Analiza greške u zvezdastoj mreži sa RBS signalom

Ako isključimo dinamičke karakteristike senzora (moguća kašnjenja unutar samih senzora) iz jednačine, greška označavanja vremenskih uzoraka (*timestamping-a*, ili „udaranja“ vremenskih žigova) u bežičnoj mreži topologije zvezde sa centralizovanim izvorom vremena potiče od neodređenosti propagacije signala i nestabilnosti frekvencija tajmera senzorskih uređaja. Uspešna i neuspešna RBS procedura su ilustrovane na Sl. 6.1. Ako RBS uspe (primer su uređaji 1 i 2 na dijagramu), onda interval 1 treba da se odredi, i fizičko vreme emisije prozivnog signala je nebitno. Napomena: precizno fizičko vreme (tačan sekund izvršenja) nije bitno za merenja u građevini. Ako RBS ne uspe (primer uređaja 3 na dijagramu), interval 2 treba



Slika 6.1 Vremenski dijagram uspešnog i neuspešnog RBS-a

da se izračuna, i tada je neodređenost transmisionog vremena bazne stanice (bilo kakvi slojevi softvera operativnog sistema i BIOS-a plus neodređenosti vezane za radio modem) bitna i utiče na označavanje vremena događaja.

Momenat u kome se neki događaj desio može se izračunati bazirano na informaciji sa uređaja  $n$  kao:

$$t_n = t_{out} + t_{p.n} + t_{d.n} + \frac{N_n}{f_n} \quad (6.1)$$

gde je  $t_{out}$  trenutak u kome je prozivni signal stvarno emitovan sa centralne stanice,  $t_p$  je vreme propagacije radio talasa,  $t_d$  je vreme potrebno da hardver reaguje na ovaj signal i startuje tajmer (kaunter),  $N$  je broj otkucaja kauntera do momenta detekcije događaja, a  $f$  je frekvencija tajmera uređaja. Indeks  $n$  se odnosi na  $n$ -ti uređaj.

Pod „događajem“ se u gornjoj jednačini smatra trenutak detekcije stvarnog događaja od strane uređaja. Ova dva vremena u stvarnosti nisu ista, mada je neodređenost reakcije uređaja na događaj, određena softverom (embedded RTOS-om) i hardverom (isključujući pomenute karakteristike samog senzora), najčešće relativno mala u odnosu na druge neodređenosti u bežičnoj mreži i biće privremeno zanemarena.

Ako je RBS procedura uspešna onda je vrednost  $t_{out}$  ista za sve uređaje, ne uzrokujući nikakvu neodređenost. Kao što je napomenuto, tačna fizička vrednost  $t_n$  nije bitna za merenja u građevini, već je bitno da devijacija ovog parametra na različitim senzorskim uređajima bude što manja. Maksimalna greška vremenskog žiga onda iznosi:

$$\Delta t_n = \Delta t_{p.n} + \Delta t_{d.n} + \Delta \left( \frac{N_n}{f_n} \right) \quad (6.2)$$

gde članovi sa desne strane predstavljaju maksimalne greške odgovarajućih članova.

Radio talasi se prenose brzinom od  $3 \cdot 10^8$  m/s i čak i u slučajevima kada su udaljenosti između centralne stanice i udaljenih senzorskih uređaja velike, one se mogu izmeriti (ili makar proceniti) sa dovoljnom sigurnošću da  $\Delta t_p$  padne daleko ispod mikrosekunde, jer jedan metar preciznosti odgovara 3 ns. Zato se ovaj član zanemaruje.

Zbog celobrojne (*integer*) prirode broja otkucaja tajmera, njegova maksimalna greška iznosi jedan, tako da možemo da zaključimo da je:

$$\Delta t_n = \Delta t_{d.n} + \frac{1}{f_n} + N_n \frac{\Delta f_n}{f_n^2} \quad (6.3)$$

Zamenjujući  $N/f$  član sa vremenom proteklim od starta tajmera do detekcije događaja  $t_e$ , i uvodeći maksimalnu relativnu grešku frekvencije tajmera  $\delta f$ , dobijamo:

$$\Delta t_n = \Delta t_{d.n} + \frac{1}{f_n} + \delta f_n \cdot t_{e.n} \approx \Delta t_{d.n} + \delta f_n \cdot t_{e.n} \quad (6.4)$$

Biće pokazano da je  $\Delta t_d$  oko 40 puta veće od mašinskog ciklusa  $1/f$ , tako da se ovaj zanemaruje.

Maksimalna devijacija između vremenskih žigova  $t_e$  sa dva različita uređaja  $n$  i  $m$  je suma maksimalnih devijacija na jednom uređaju (ovo ne uključuje različita vremena starta njihovih tajmera):

$$\Delta t_{e,n-m} = t_{e,n}(\delta f_n + \delta f_m \frac{t_{e,m}}{t_{e,n}}) \approx t_e(\delta f_n + \delta f_m) \quad (6.5)$$

Iako  $t_e$  ima unikatnu vrednost na svakom uređaju, zbog različitih momenata u kojima je brojanje počelo, možemo ga aproksimirati jednom vrednošću, zato što ako je  $t_e$  malo, onda je  $t_d$  dominantno u jednačini (6.4), pa su varijacije  $t_e$  nebitne, dok ako je  $t_e$  veliko, dve vrednosti za  $t_e$  teže jedna drugoj, pa član  $t_{e,m}/t_{e,n}$  u jednačini (6.5) teži jedinici. Zamenjujući sumu relativnih neodređenosti neodređenošću odnosa, dobijamo:

$$\Delta t_{n-m} \approx 2\Delta t_d + \delta\left(\frac{f_n}{f_m}\right) \cdot t_e \quad (6.6)$$

Smatra se da  $t_d$  ima neodređenost koja ne varira mnogo između različitih senzorskih uređaja već da zavisi od hardvera identičnog dizajna (uvedena je aproksimacija  $\Delta t_{d,n} + \Delta t_{d,m} = 2\Delta t_d$ ). Može se zaključiti da u slučaju uspešnog RBS signala, neodređenost relativnog vremenskog označavanja uglavnom zavisi od neodređenosti vezanih za vreme detekcije RBS-a i nestabilnost odnosa frekvencija tajmera.

Međutim, radio komunikacija u praktičnim uslovima je podložna gubicima i nije verovatno da veliki broj čvorova može da primi isti prozivni signal i sinhronizuje tajmere, izvodeći potpuno uspešnu RBS proceduru. Centralna stanica je iz ovog razloga programirana da emituje nekoliko prozivnih signala u kratkim intervalima, da maksimizira šanse da svi uređaji dobiju trigger za početak merenja. Svaki od prozivnih signala sadrži varijabilan broj koji označava koliko otkucaja tajmera uređaj treba da čeka da bi započeo merenje. Ovaj se računa na bazi pokazivanja sata centralne stanice (i smanjuje se sa svakim sledećim prozivnim signalom u sekvenci). Ako uređaj propusti finalni prozivni signal, ali detektuje neki od prethodnih, on i dalje može da učestvuje u merenju, ali postoji dodatni član u jednačini (6.4) i ona sada glasi:

$$\Delta t_n \approx \Delta t_b + \Delta t_{d,n} + \delta f_n \cdot t_{e,n} \quad (6.7)$$

jer  $t_{out}$  više nije zajedničko za sve uređaje.  $t_b$  predstavlja vreme emisije prozivnog signala (*beacon-a*).

Ako su standardne devijacije članova na desnoj strani poznate, možemo da zaključimo da je:



$$\sigma_n \approx \sqrt{(\sigma_b)^2 + (\sigma_{d,n})^2 + (\sigma(f_n) \cdot t_{e,n})^2} \quad (6.8)$$

gde je prvi član sa desne strane opcion, i važi u slučaju neuspešne RBS procedure. Za standardnu devijaciju razlike vremenskih žigova između uređaja  $n$  i  $m$ , važiće:

$$\sigma_{n-m} \approx \sqrt{2(\sigma_b)^2 + 2(\sigma_d)^2 + (\sigma(\frac{f_n}{f_m}) \cdot t_e)^2} \quad (6.9)$$

Merenje pojedinačnih  $t_b$  ili  $t_d$  je praktično nemoguće. Eksperimentalno je moguće odrediti razliku između vrednosti na različitim uređajima. Zato standardne devijacije nemerljivih veličina gube značaj i zamenićemo ih standardnim devijacijama koje se odnose na par uređaja:

$$\sigma_{n-m} \approx \sqrt{(\sigma_b)^2 + (\sigma_d)^2 + (\sigma(\frac{f_n}{f_m}) \cdot t_e)^2} \quad (6.10)$$

gde  $\tau_d$  predstavlja razliku vremena detekcije za par senzorskih uređaja a  $\tau_b$  opisuje razliku između dva različita prozivna signala (iz iste serije) koji su detektovani od strane različitih uređaja (ovaj član postoji samo u slučaju neuspešne RBS procedure).  $\sigma\tau_b$  zapravo predstavlja  $\sqrt{2}$  puta standardnu devijaciju takozvanog vremena paketizacije radio modema na centralnoj stanici.

### 6.3 Eliminacija neodređenosti vremena paketizacije modema

Vreme paketizacije modema je vreme između upisa zadnjeg bajta u modem od strane RS232 interfejsa i momenta početka fizičke transmisije radio signala. Ono ima neodređenost od 0,5 ms po proizvođačkoj specifikaciji, a eksperimenti su pokazali da je ova vrednost ispod 0,2 ms. To je još uvek prilično velika vrednost u poređenju sa ostalim neodređenostima u sistemu, koje su reda veličine mikrosekundi (izuzimajući karakteristike akcelerometara, koje se teško mogu izmeriti, a nisu dovoljno dobro specificirane od strane proizvođača).

Pomenuto je da centralna stanica sukcesivno emituje nekoliko prozivnih signala u kratkom vremenu (period, tj. razmak između dva susedna prozivna signala zavisi od broja senzorskih uređaja u mreži, jer je neophodno oko 0,1 s po uređaju za odgovor). Senzorski uređaji setuju tajmere sa svakim signalom koga prime (u skladu sa vremenom koje treba da protekne do početka merenja). Oni koji komuniciraju regularno, bez grešaka, sinhronizuju se sa zadnjim signalom. Oni uređaji koji prime neki prethodni, a ne prime zadnji prozivni signal, predstavljaju problem pri sinhronizaciji jer se  $t_b$  uvodi u

jednačinu. Operativni sistem (MS Windows) čini  $t_b$  potencijalno većim od vremena paketizacije modema. Šansa da dođe do interapta od strane operativnog sistema se minimizira trigerovanjem paralelnog procesa serijske transmisije neposredno posle izračunavanja informacije koja se transmituje kao kaunt tajmera uređaja (vreme do početka merenja izraženo u ciklusima uređaja), a koje se vrši odmah po direktnom čitanju kauntera PC procesora (ne softverskog sata tj. *tickcount* varijable operativnog sistema MS Windows). Program na PC računaru troši delić mikrosekunde procesorskog vremena da izvede ovu kalkulaciju. Interapt značajne dužine (u poređenju sa 0,2 ms) nikada nije detektovan u više stotina eksperimenata.

Dodavanje povremene nesigurnosti reda veličine do milisekunde je i dalje relativno prihvatljivo za većinu SHM aplikacija u građevini, ali ipak postoji način da se neodređenost svede na red veličine prijemne neodređenosti ( $t_d$ ). Svi senzorski uređaji dodeljuju lokalne vremenske žigove svakom prozivnom signalu (u seriji) koji prime sa centralne stanice, a ne samo zadnjem, u odnosu na koga vrše merenje. Podaci sa bilo kog „referentnog” uređaja koji je primio oba kritična prozivna signala (i stvarno zadnji i onaj koji je zadnji primljen na drugom uređaju koji nije dobro komunicirao) mogu poslužiti da se realno proceni vremenski razmak između ova dva signala i doda u jednačinu, eliminišući neodređenost vremena paketizacije, i zamenjujući je neodređenošću prijema. Zanemarena je neodređenost koja potiče od neodređenosti frekvencije referentnog uređaja tokom kratkog vremenskog intervala između dva prozivna signala (*skew*). Tako za neuspešan RBS imamo:

$$\sigma_{n-m} \approx \sqrt{(\sigma'_d)^2 + \left(\sigma \frac{f_n}{f_m} \cdot t_e\right)^2} \quad (6.11)$$

gde je:

$$\sigma'_d = \sqrt{2(\sigma_{d,n})^2 + (\sigma_{d,m})^2} = \sqrt{\frac{3}{2}} \sigma_d \quad (6.12)$$

u slučaju da RBS uspe na uređaju  $n$ , a ne uspe na uređaju  $m$ , ili:

$$\sigma'_d = \sqrt{2(\sigma_{d,n})^2 + 2(\sigma_{d,m})^2} = \sqrt{2} \sigma_d \quad (6.13)$$

u slučaju da RBS procedura nije uspela ni na jednom od ova dva uređaja, a *post-facto* sinhronizacija je izvršena koristeći informacije sa trećeg senzorskog uređaja.

„Nagib” (*skew*), tj. vremenska neodređenost koja potiče od neodređenosti frekvencije tokom kratkog vremenskog perioda, je zanemaren, pošto su tipične

neodređenosti frekvencije  $10^{-6}$  relativno, a period u pitanju je reda veličine do sekunde, pa se radi o neodređenosti reda veličine  $1 \mu\text{s}$ . Ovakva standardna devijacija, sabrana kao nekorelisana vrednost (kvadratni koren iz sume kvadrata) sa većom standardnom devijacijom prijema i detekcije signala, ne utiče bitno na rezultat. Efektivno se ovo može posmatrati i kao ekstenzija perioda  $t_e$  za jednu sekundu ili manje.

## 6.4. Metodologija

Eksperimentalno je ispitivano koliko su stabilne frekvencije kristala pod različitim uslovima i kolike su stvarne neodređenosti vremena detekcije prozivnog signala za parove uređaja i parove prozivnih signala ( $\tau_d$  i  $\tau_b$ ).

Jedna od metoda da se predvidi drift frekvencije jeste merenje temperature. Frekvencija u funkciji temperature je ispitivana za različite procesore da se odredi tačnost teorijskog modela i analiziraju efekti preciznosti na finalne rezultate sinhronizacije podataka.

Vremenski žigovi stvarnih događaja sa različitih uređaja su poređeni u slučaju električne eksitacije (zanemarljivo vreme detekcije), da se proverí da li je neodređenost razlike posledica efekata koji su analizirani u prethodnim poglavljima, ili postoji neki važan efekat koji je propušten. U slučaju mehaničke eksitacije, merenja su takođe vršena, ali ona više pokazuju kakvi su dometi primenjene mehaničke opreme i mehaničkih senzora, jer su vremenske konstante uključene u mehaničke procese daleko veće od onih za električne procese.

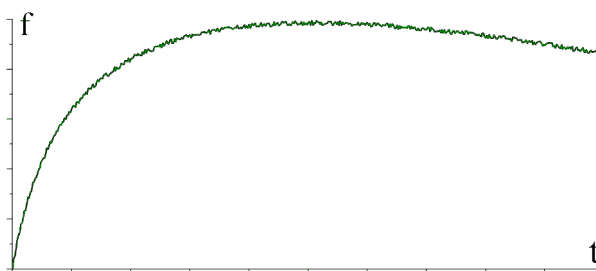
Konačno, metod brojanja frekvencije na licu mesta je izabran kao najbolji način za kompenzaciju dugoročnih i srednjoročnih nestabilnosti frekvencije tajmera, tako da je finalna postignuta simultanost mernih podataka (za merne periode ispod minuta) reda veličine neodređenosti vremena detekcije prozivnog signala. Ova neodređenost se ne može izbeći u sistemima bežičnog tipa.

## 7. EKSPERIMENTALNA PROVERA SIMULTANOSTI PODATAKA

### 7.1. Stabilnost i preciznost kvarcnih oscilatora

Kvarcni kristali, koji daju takt procesorima, rade na principu piezoelektričnog efekta. Rezonantna učestanost zavisi uglavnom od vrste, oblika i veličine kristala, načina sečenja, i primenjenih statičkih sila. Ova frekvencija je ne samo kratkoročno već i dugoročno promenljiva jer su kristali podložni starenju.

Od kratkoročnih varijacija najbitnije su one izazvane promenama temperature, a mnogobrojni fizički efekti doprinose stalnim skokovitim promenama i laganim „driftovima“ ove frekvencije [103]. Ne postoji jasna podela na kratkoročne i dugoročne promene frekvencije, a često se koristi i termin srednjoročne varijacije (ili varijacije srednjeg roka, odnosno *intermediate term*). Kratkoročne devijacije obično podrazumevaju one koje se dešavaju tokom sekundi ili minuta nekog eksperimenta. One se često označavaju i terminom „stabilnost“. Renomirani proizvođači određuju stabilnost frekvencije laboratorijski i predstavljaju rezultate logaritamskom krivom (stabilnost u periodima od milisekunde do stotinu ili hiljadu sekundi; za kvalitetnije kristale stabilnost može imati maksimum reda veličine čak do  $10^{-13}$  u relativnim jedinicama za periode reda veličine sekunde). Srednjoročne devijacije obično podrazumevaju one koje se dešavaju tokom različitih sati ili dana ili tokom različitih eksperimenata (nakon više isključenja i uključenja uređaja, promene napajanja, i sl, bez obzira da li je stvarno vreme reda veličine minuta ili dana). Spisak efekata koji utiču na kratkoročne i srednjoročne promene frekvencije je dugačak. Neki od njih su promene temperature (uključujući pojavu temperaturnog histerezisa), šokovi (fizički udarci), promena položaja, promena orijentacije gravitacionog polja, promena napona napajanja (uključujući histerezis prilikom uključenja i isključenja koji se naziva *retrace* efekat), i elektromagnetske



Slika 7.1 Tipična promena rezonantne frekvencije kvarcnih kristala usled starenja

smetnje (zračenje, npr. ono proizvedeno od strane radio modema). Dugoročne promene se odnose na duže periode koji su reda veličine meseci ili godina. Uzrok dugoročnih promena su kretanja molekula primesa unutar kristala, i fizičkih promena oblika samog kristala kao i delova sa kojima je doveden u dodir (postolja). Dugoročne promene su najčešće eksponencijalno opadajuće po intenzitetu (tokom godina), i što je kristal stariji smatra se stabilnijim. Tipična kriva promene frekvencije kvarcnog kristala usled starenja data je na sl. 7.1.

Preciznost kristala nije isto što i stabilnost. Pojam „preciznost“ označava razliku između srednje vrednosti rezonantne frekvencije kristala i njegove nominalne (deklarisane) vrednosti. Preciznost, prirodno, može da varira usled starenja.

Kvarcni kristali se koriste kao elementi u oscilatornim kolima (oscilatorima). Osnovna podela oscilatora je na tri kategorije: RTXO, TCXO i OCXO. Jednostavna kola koja ne vrše nikakvu kompenzaciju temperaturskih promena nazivaju se RTXO (*room temperature crystal oscillators*). TCXO (*temperature compensated crystal oscillators*) sadrže termometre, mere temperaturu, i uvode korekcije da bi se rezonantna frekvencija učinila stabilnijom. Kod OCXO (*oven controlled crystal oscillators*) se kolo održava na konstantnoj temperaturi, u posebnoj komori, tako da su temperaturski uticaji zanemarljivi. Bežični senzorski uređaji skoro isključivo koriste RTXO kola, ne samo zbog manje cene, već i zbog znatno niže potrošnje energije, što je važan faktor u projektovanju bežičnih uređaja. Vrlo je verovatno da će razvoj tehnologije digitalnih TCXO (kao što su MCXO - *microprocessor controlled crystal oscillators*) dovesti do pojeftinjenja, smanjenja snage, i šire upotrebe ove vrste oscilatora u bliskoj budućnosti [104].

## 7.2. Osnovna kalibracija procesora

Prvi korak u ispitivanju frekvencija procesora je bila „osnovna” kalibracija, izvedena u laboratorijskim uslovima, na sobnoj temperaturi, korišćenjem kauntera (frekvencmetra) visoke preciznosti Pendulum CNT-90, u Direkciji za mere i dragocene metale u Beogradu.

Procesor ADuC845 koristi eksterni *tuning fork* (u obliku zvučne viljuške) kvarcni kristal rezonantne frekvencije 32768 Hz, koji se preko PLL kola [105] za multiplikaciju frekvencije povećava do radnog takta procesora (maksimalno 6,29 MHz pri radnom

naponu od 3 V).

**Tabela 5** Rezultati bazične kalibracije procesora (p-p označava *peak-to-peak*)

uređaj br.	$f_0$ [MHz]	$\sigma f$ [Hz]	p-p $f$ [Hz]
1	6,2920338	0,355412	0,85419
2	6,2920401	0,118423	0,29495
3	6,2919372	0,018695	0,05906
4	6,2912398	0,016961	0,05743
5	6,2913025	0,099097	0,27562
6	6,2912446	0,045284	0,14080

Senzorski uređaji su programirani da isključivo menjaju stanje digitalnog izlaznog pina (svi interapti su bili zabranjeni), proizvodeći frekvenciju nekoliko puta manju od radne frekvencije tajmera. Na ovaj način, bilo kakva nepravilnost u radu multiplikatora frekvencije, koji predstavlja kolo ugrađeno u procesorski čip, se uključuje u rezultat. Merenja su izvršena na svih 6 procesorskih ploča. Elektronske komponente su nabavljene u dva navrata, tako da u tabeli 5 vidimo 3 procesora iz serije 1 i 3 procesora iz serije 2. Rezultati pokazuju relativne razlike reda veličine  $10^{-5}$  za procesore iz iste serije, odnosno  $10^{-4}$  između procesora iz različitih serija. Date su i standardne devijacije frekvencije i *peak-to-peak* razlike. Standardne devijacije su reda veličine  $10^{-8}$  za posmatrani period od oko 5 minuta.

Korišćenje ovih frekvencija umesto nominalnih ( $32768 \times 192$  Hz) pomaže da se dobiju bolji rezultati u vremenskoj sinhronizaciji, ali je ovakav metod korekcije ograničene vrednosti, zbog starenja kristala i nestabilnosti frekvencije. Jedan od glavnih uzroka varijacija frekvencije je temperatura. Sa ugrađenim termometrom u procesoru, moguće je izmeriti temperaturu (na maloj udaljenosti od kristala), i softverski izračunati (predvideti) rezonantnu frekvenciju. Svi senzorski uređaji informišu centralnu stanicu o temperaturi prilikom slanja rezultata merenja (što je standardna procedura), tako da se korekcija može obaviti na centralnoj stanici. Funkcija  $f(\theta)$ , gde je  $f$  frekvencija a  $\theta$  temperatura, mora biti ustanovljena, a vrlo je važna i procena njene pouzdanosti (tačnosti).

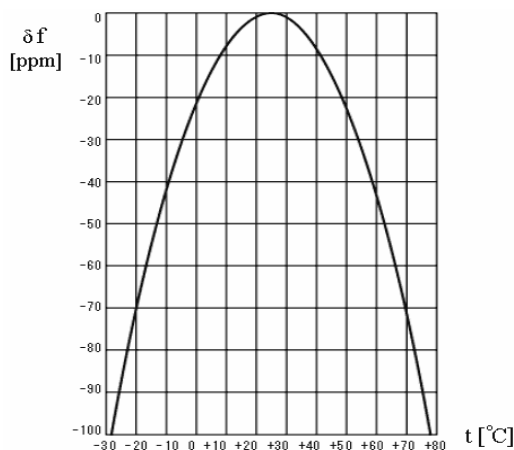
### 7.3. Temperaturska kalibracija

Teoretski, niskofrekventni kvarcni kristali tipa zvučne viljuške imaju paraboličnu funkciju  $f(\theta)$  [106,107], prikazanu na grafiku sa sl. 7.2:

$$f(\theta) = f_0(1 - c(\theta - \theta_0)^2) \quad (18)$$

**Tabela 6** Koeficijenti temperaturske zavisnosti frekvencije procesora

ured.br.	$c$ [ $^{\circ}\text{C}^{-2}$ ]	$\theta_0$ [ $^{\circ}\text{C}$ ]
1	$4,50 \cdot 10^{-8}$	24,5
2	$3,76 \cdot 10^{-8}$	22,5
3	$4,60 \cdot 10^{-8}$	20,0
4	$4,12 \cdot 10^{-8}$	24,0
5	$3,43 \cdot 10^{-8}$	23,0
6	$3,89 \cdot 10^{-8}$	23,0

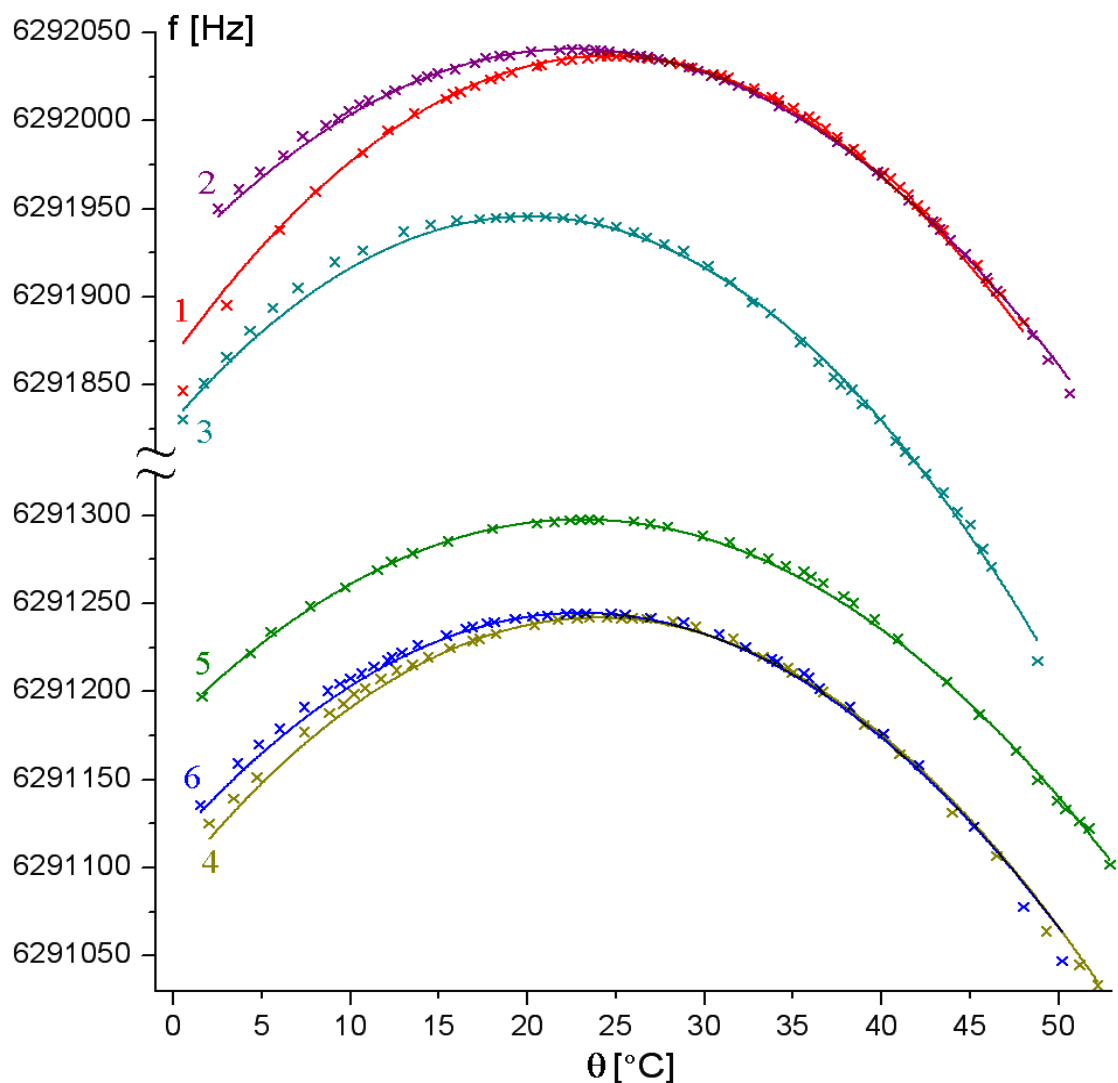


**Slika 7.2** Teorijska zavisnost  $f(\theta)$  za tuning fork kvarcne kristale

Temperaturska komora i frekvencometar su iskorišćeni da bi se ovo ispitalo. Temperature su se kretale u opsegu od oko  $0^{\circ}\text{C}$  do oko  $50^{\circ}\text{C}$ . Ispitivanje je izvršeno ne samo da bi se izmerili parametri za pojedinačne procesore, već i da se ispita pouzdanost matematičkog modela.

Rezultati merenja funkcije  $f(\theta)$  za različite procesore, predstavljeni na sl. 7.3, nisu sasvim ponovljivi iz velikog broja razloga. Glavni je nemogućnost da se tačno odredi stvarna temperatura kristala i da se održi konstantnom tokom značajnog vremenskog perioda. Efekti termalnog histerezisa i termalnog gradijenta [108], između ostalih, doprinose povećanju greške.

Parametri, dobijeni eksportovanjem podataka u Origin i korišćenjem funkcije fitovanja krive, su dati u tabeli 6. Koeficijenti  $c$  variraju između  $3,4 \cdot 10^{-8}$  i  $4,6 \cdot 10^{-8} \text{ }^{\circ}\text{C}^{-2}$ . Ovo nije u skladu sa proizvođačkom specifikacijom, po kojoj koeficijenti iznose  $3,4 \pm 0,6 \cdot 10^{-8} \text{ }^{\circ}\text{C}^{-2}$ . Dobijene temperature  $\theta_0$  se zaista nalaze u specificiranom opsegu,  $20\text{-}30^{\circ}\text{C}$ . Zaključak koji se može doneti na osnovu ovih eksperimenata ograničene tačnosti jeste da softverska korekcija frekvencije u zavisnosti od temperature pruža



Slika 7.3 Zavisnost frekvencije procesora od temperature u eksperimentu sa temperaturskom komorom

tačnost ispod ppm-a (milijonitog dela) ako se sistem koristi u zatvorenim prostorima, tj. u blizini sobne temperature, odnosno nekoliko ppm u slučaju da se od sobne temperature odstupa deset i više stepeni.

## 7.4. Određivanje neodređenosti vremena detekcije prozivnog signala

Serijski eksperimenti sa osciloskopom su izvedeni da se proceni postoji li sistemski razlik u vremenima reakcije na prozivni (*beacon* ili RBS) signal na različitim senzorskim uređajima, i kolika je standardna devijacija ove razlike ( $\tau_d$  u jednačini (6.10)).



Parovi senzorskih uređaja su programirani da setuju digitalni izlazni pin po prijemu prozivnog signala, i ovo je korišćeno kao triger storidž osciloskopa. Izvođeci 150 prozivnih signala za svaki od 3 para modema, dobijena su sistemska i slučajna odstupanja reda veličine mikrosekundi. Iznenadjuće, parovi modema 1-2 i 3-1 pokazuju potpuno isti sistemski ofset od 4,9  $\mu$ s (prvi modem naveden u paru je sporiji), dok se par 3-2 savršeno poklapa sa ofsetom od tačno 9,8  $\mu$ s. Ovo bi mogla da bude posledica nekog kalibracionog parametra modema koji je kvantifikovan a nije vidljiv korisniku. Važnija od sistemskog ofseta, koji se može lako softverski eliminisati, jeste distribucija rezultata i standardna devijacija. Svi parovi pokazuju standardne devijacije između 6,1 i 6,8  $\mu$ s, maksimalne devijacije između 17 i 21  $\mu$ s, i približno Gausovu distribuciju. Ovo je prikazano u tabeli 7 ( $\tau_d$ ) i na histogramima u levoj koloni sl. 7.4. Uzrok standardne devijacije je uglavnom nesavršenost modema, ali uključena je i neodređenost vezana za softver uređaja. Embedded RTOS izvršava petlju ispitujućih flegova iz serijskog (komunikacionog) i tajmer interapta, skrećući na ispitivanje poruke ili gašenje ako se ovi detektuju, tako da treba da izvede 3 instrukcije odnosno ukupno 11 mašinskih ciklusa, tako proizvođeci slučajan vremenski ofset od  $\pm 1,75 \mu$ s.

Eksperimenti su izvedeni i sa namernim promašajem finalnog prozivnog signala na jednom uređaju, da bi se odredila neodređenost vremena paketizacije ( $\tau_b$ ). Vrednost nikada nije prešla 0,2 ms.

## 7.5. Eksperimenti sa električnom eksitacijom

U eksperimentima sa električnom pobudom simuliran je stvaran događaj, a cilj je bio da se proveriti da li vremensko označavanje ovih „stvarnih” događaja odgovara kombinaciji neodređenosti koje su teorijski ustanovljene (zbiru neodređenosti vremena detekcije prozivnog signala i nestabilnosti odnosa frekvencija pomnoženog sa vremenom merenja). Nestabilnosti frekvencija su ustanovljene laboratorijski pod savršenim uslovima, gde su procesori bili u stabilnom režimu na konstantnoj temperaturi. U stvarnim uslovima, procesori se uključuju i isključuju, periferne komponente variraju potrošnju (izazivajući varijacije napajanja), radio talasi se emituju u neposrednoj blizini, i temperatura nije stabilna.

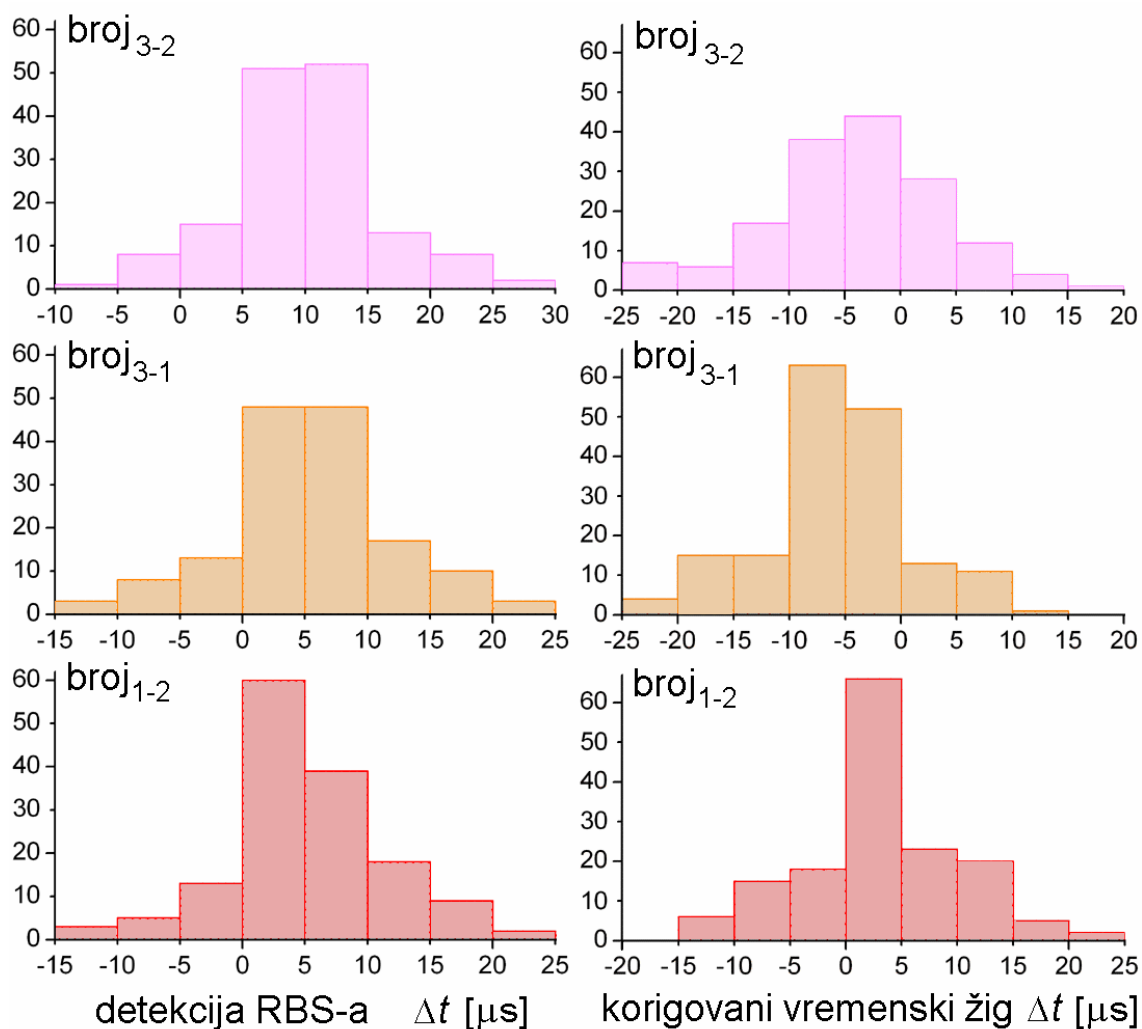
Signal generator je spojen na digitalne ulazne pinove para senzorskih uređaja. Ovi su spojeni paralelno. Generator je generisao pravougaoni signal niske frekvencije. Oblik

signala je posmatran na osciloskopu i ustanovljeno je da ima oko 100 ns uzlaznu i silaznu ivicu, što implicira zanemarljivo vreme neodređenosti detekcije (približno jedan mašinski ciklus je potreban da procesor primi hardverski interapt na ivicu ovog signala). Parovi uređaja su programirani da izvrše obično merenje, koje podrazumeva 20 sekundi, a zatim detektuju prvu ivicu signala i zapišu odgovarajući vremenski žig. Ustvari je prolazilo 22-24 sekunde do ovog „događaja”, zato što je zadato inicijalno čekanje od 2 sekunde, a ivice signala su nekorelisane sa uređajima, tako da je još neko slučajno vreme prolazilo do nailaska odgovarajuće ivice signala frekvencije ispod 1 Hz. Softverske korekcije su izvedene: brojevi otkucaja tajmera su podeljeni sa frekvencijama iz kalibracionog eksperimenta, i sistemski ofseti za parove modema su kompenzovani. Rezultati ( $t_2$  u tabeli 7 i histogrami na desnoj strani sl. 7.4) pokazuju da su standardne devijacije porasle za 0 do 1,8  $\mu$ s u odnosu na prvi eksperiment, što je očekivano, zbog još jedne softverske nesigurnosti (slučajan  $\pm 1$   $\mu$ s ofset za detekciju ivice ulaznog signala), i zbog nestabilnosti frekvencija tajmera. Sistemski ofseti koji se dobijaju ovom prilikom idu do 5,8  $\mu$ s i nisu konzistentni (ne slažu se međusobno), kao u eksperimentu prikazanom u prethodnom paragrafu. Da su frekvencije postigle nove vrednosti koje su stabilne tokom celog eksperimenta, ofseti bi se složili. Ovako možemo da zaključimo da frekvencije nisu stabilne tokom nešto manje od dva sata, koliko je ova serija merenja trajala.

Devijacije frekvencijskog odnosa  $f_n/f_m$  mogu se podeliti na kratkoročne, srednjeg roka, i dugoročne. Kratkoročne u tabeli 7 opisuju promene koje su nastale za vreme seta eksperimenata sa jednim parom uređaja (minuti, ne sati). Srednjoročne opisuju promene tokom celog eksperimenta (za sve parove uređaja). Fizičko vreme koje je prošlo nije jedini parametar u ovoj klasifikaciji, već i sve mehaničke i električne radnje koje su izvedene sa uređajima, a koje izazivaju devijacije. Dugoročne devijacije se odnose na vreme proteklo između osnovne kalibracije Pendulum kaunterom (frekvencmetrom) i ovog eksperimenta. Devijacije iz tabele 7 su procenjene na bazi srednjeg  $\tau_d$  i porasta  $\sigma_{t_{nm}}$  u odnosu na  $\sigma_{t_d}$ . Poredeći ove vrednosti sa onima iz bazične kalibracije, vidimo da su veće, ali porast od nekoliko puta ne iznenađuje ako imamo u vidu da su uslovi eksperimenta mnogo realniji i da su vremena o kojima se radi znatno duža.

**Tabela 7** Srednje vrednosti i standardne devijacije razlika u vremenima detekcija prozivnog signala i korigovanih vremenskih žigova, i proračunate devijacije odnosa frekvencija tajmera za parove uređaja

par	$\overline{\tau_d}$ [ $\mu\text{s}$ ]	$\sigma_{\tau_d}$ [ $\mu\text{s}$ ]	$\overline{\tau_{nm}}$ [ $\mu\text{s}$ ]	$\sigma_{\tau_{nm}}$ [ $\mu\text{s}$ ]	dugo+srednjoročno $\sigma(f_n/f_m)$	kratkoročno $\sigma(f_n/f_m)$
1-2	4,9	6,3	3,1	6,9	$1,3 \cdot 10^{-7}$	$1,2 \cdot 10^{-7}$
3-1	4,9	6,8	-5,8	6,8	$2,5 \cdot 10^{-7}$	$0,2 \cdot 10^{-7}$
3-2	9,8	6,1	0,8	7,9	$0,3 \cdot 10^{-7}$	$2,2 \cdot 10^{-7}$

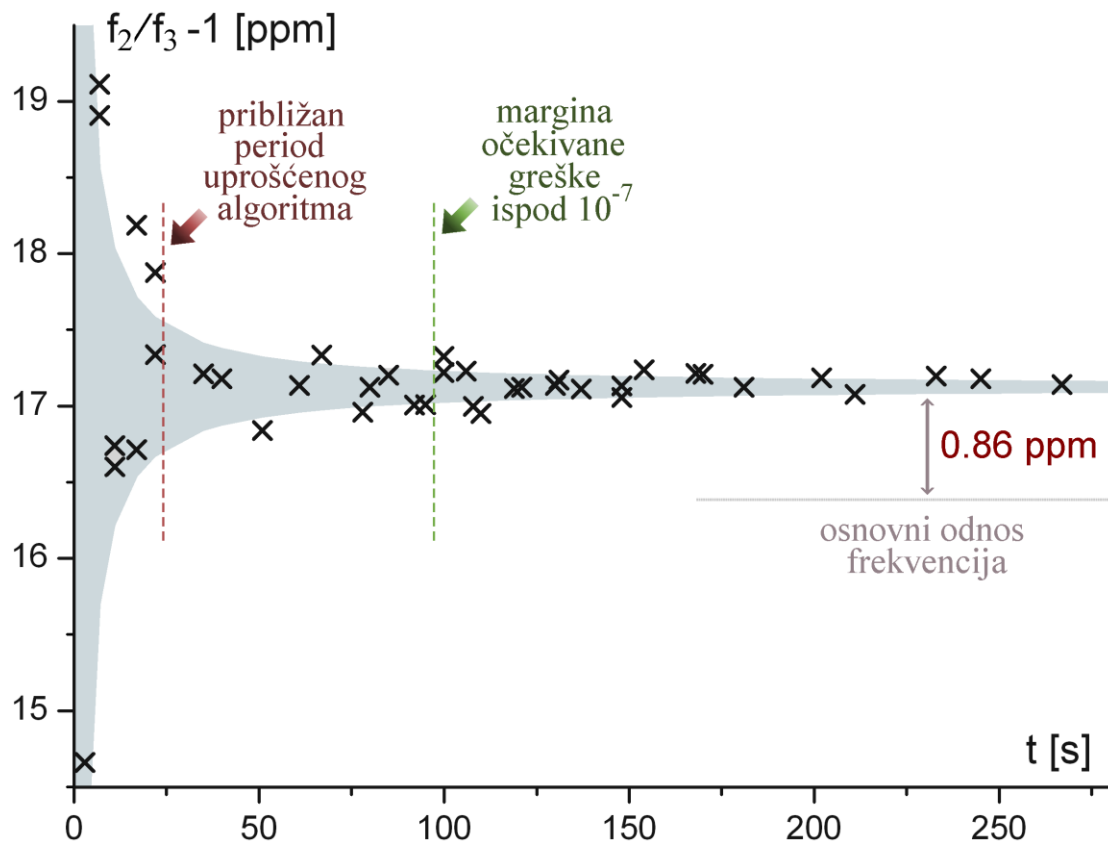


**Slika 7.4** Histogrami razlike u vremenu propagacije prozivnog signala (levo) i histogrami razlike u vremenskom označavanju događaja sa softverskom korekcijom (desno)

## 7.6. Metod brojanja na licu mesta

Jednostavna procedura koja traje nekoliko minuta može da pomogne da se odredi

odnos frekvencija sa zadovoljavajućom tačnošću, tako da se dobije greška sinhronizacije podataka manja od one koja se neizbežno čini usled neodređenosti prijema prozivnog signala. Ovime se izbegava i potreba za regularnim kalibracijama frekvencija procesora i potreba za softverskom korekcijom zbog temperature (iako ova korekcija može da se dodatno primeni u slučaju brzih promena temperature).



Slika 7.5 Rezultati kalibracije odnosa frekvencija dva uređaja metodom brojanja na licu mesta

Uređaji se prozovu jednostrukim prozivnim signalom i ostavljaju na stend-baju sve dok korisnik ne reši da pročitati njihove tajmere. Odnos broja otkucaja daje odnos frekvencija tajmera. Neodređenost potiče od neodređenosti prijema prozivnog signala. Što je trajanje eksperimenta duže, relativni udeo ove greške je manji, odnosno rezultat eksperimenta bolje konvergira ka tačnoj vrednosti odnosa frekvencija (onoliko koliko se tačan odnos uopšte može definisati, imajući u vidu neminovne nestabilnosti frekvencija). Tipični rezultati serije eksperimenata su prikazani na sl. 7.5. Siva površina označava odnos frekvencija unutar opsega koji je određen limitima standardne devijacije razlike prijema prozivnog signala na dva uređaja. Vidimo da su rezultati eksperimenta zadovoljavajući jer se većina tačaka nalazi unutar ove oblasti čime je

potvrđeno da ne postoje značajni dodatni izvori greške. Frekvencija tajmera jednog uređaja je namerno promenjena postavljanjem uređaja na desetak stepeni višu temperaturu od sobne (za 0,86 ppm-a, kao što je prikazano na sl. 7.5). Temperatura je održavana stabilnom za vreme trajanja eksperimenta.

Možemo da zaključimo da je 2-3 minuta prava margina dobijanja zadovoljavajućih rezultata metodom brojanja na licu mesta. Standardna devijacija odnosa frekvencija bazirana na neodređenosti (standardnoj devijaciji) prijema para prozivnih signala iznosi ispod  $10^{-7}$  kada je period eksperimenta veći od 1,5 minuta. Duži periodi (npr. 10 minuta) nemaju svrhe, jer se prevazilazi stvarna stabilnost frekvencije, koja je reda veličine ili nešto lošija od one određene eksperimentom sa Pendulum kaunterom (bazična kalibracija), dakle između  $10^{-8}$  i  $10^{-7}$ . Identičan eksperiment se može izvesti sa proizvoljnim brojem uređaja pri čemu se dobija međusobni odnos više frekvencija.

Kada je puna kalibracija ovom metodom nepraktična, bilo zbog nedostatka vremena operatera, bilo zbog malih energetske rezervi senzorskih uređaja, uprošćena verzija algoritma se može primeniti. Na kraju izvedenog merenja, centralna stanica može da pošalje još jedan prozivni signal (što normalno ne radi, već se obraća uređajima pojedinačno). Ovo dozvoljava senzorskim uređajima da „markiraju” još jednu tačku u vremenu. Sa dve referentne tačke u vremenu, odnos frekvencija se može izračunati na isti način kao u slučaju dužeg perioda, jedina razlika je što se u ovom slučaju vrši i merenje dok se čeka završni prozivni signal.

Metod korišćenja nekoliko uzastopnih prozivnih signala u slučaju loše komunikacije, i post-facto korekcija na PC računaru, je upotrebljiv i u ovom slučaju, samo što malo pogoršava pouzdanost rezultata. Preciznost ubrzanog testa je manja uglavnom zbog toga što je period kraći, a ne zbog post-facto korekcije.

Inače, uočeno je da su dugoročne (na mesečnom nivou) promene rezonantne frekvencije kristala nekoliko puta veće od proizvođačke specifikacije o starenju, koja predviđa varijacije od maksimalno  $10^{-7}$  mesečno. Pretpostavlja se da je glavni razlog za ovu pojavu činjenica da je rad bio vršen sa otvorenim kućištima, ili bez njih, svakodnevno, pri čemu su kristali, montirani na elastične pinove, bili često nenamerno iskrivljeni, i time izloženi deformacijama i promenama orijentacije g. Lista drugih potencijalnih uzroka je dugačka i uključuje: termalni histerezis, izloženost vibracijama,

kondenzaciju vode (koja se događala prilikom iznošenja uređaja iz friza u eksperimentu sa temperaturnom komorom), *retrace* efekat, radio talase, nestabilnost napona i struje napajanja, džiter PLL multiplikatora frekvencije, itd [106,108].

## 7.7. Zaključak o postignutoj simultanosti

Konačan zaključak je da je preciznost odnosa frekvencija reda veličine ppm-a ostvariva ako se pouzdamo u prethodnu kalibraciju frekvencimetrom i vršimo softversku korekciju u slučaju značajnih temperaturnih odstupanja od sobne temperature. Ali, korišćenjem metode brojanja na licu mesta, preciznost raste i do  $10^{-7}$ , čime se približavamo kratkoročnoj nestabilnosti frekvencije određene Pendulum kaunterom. Bitno je zapaziti da iako je simultanost podataka bitna za primene u građevinskoj struci, same apsolutne vrednosti frekvencija nisu toliko značajne, i apsolutna preciznost merenja frekvencije koju je potrebno postići je daleko od redova veličina koji se ovde spominju, tako da regularna kalibracija frekvencija pojedinačnih procesora nije uopšte potrebna.

U razvijenom senzorskom sistemu postignuta je simultanost podataka reda veličine od 7 do nekoliko desetina mikrosekundi, za uobičajene merne periode od 20 s. Simultanost je funkcija pouzdanosti komunikacije (jer je neodređenost ofseta ispod  $7 \mu\text{s}$  kod regularne komunikacije odnosno  $9\text{-}10 \mu\text{s}$  u slučaju gubitaka) i stabilnosti odnosa frekvencija (neodređenost iznosi oko  $10^{-6}$  u slučaju primene kalibracionih vrednosti i temperaturne korekcije u razumnom opsegu temperatura, odnosno oko  $10^{-7}$  u slučaju izvođenja punog brojanja na licu mesta; odgovarajući *skew* zavisi od trenutka unutar mernog perioda, a iznosi do oko  $20 \mu\text{s}$  u prvom slučaju odnosno do oko  $2 \mu\text{s}$  u drugom slučaju).

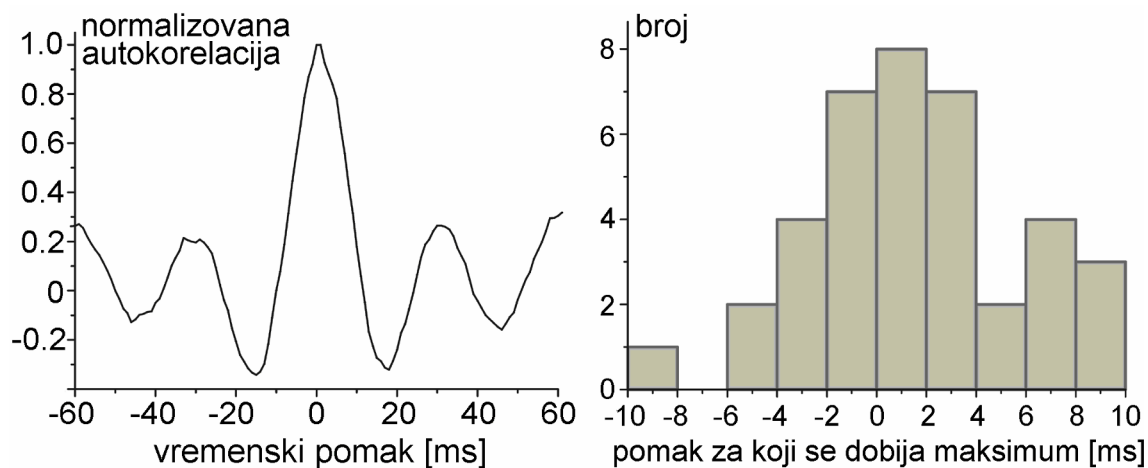
Ovi podaci su od značaja ako se vrši merenje brzih pojava brzim senzorima. Ako se koriste mehanički senzori sa periodima odabiranja reda veličine milisekunde, zaključak je da je postignuta simultanost koja daleko nadmašuje mogućnosti primenjenih senzora.

## 7.8. Mehanički testovi simultanosti

Eksperimenti sa mehaničkom eksitacijom vršeni su u dva navrata, da se ispita koliki nivo sinhronizacije može da se postigne merenjem mehaničkih veličina

mehaničkim senzorima. Pošto su vremenske konstante vezane za mehaničke procese mnogo veće od mikrosekunde, problemi koji postaju očigledni su uticaj dinamičkih karakteristika senzora (kašnjenja i propusne moći) i nemogućnost da se različiti senzori izlože identičnoj eksitaciji.

U prvoj seriji eksperimenata, dva akcelerometra su pričvršćena šrafovim na plastičnu pločicu koja je postavljena na sto i vibracije su manuelno izazivane na različite načine. U drugoj seriji eksperimenata, korišćena je Brabender DS1500 vibraciona platforma, sa mogućnošću podešavanja frekvencije i amplitude približno pravilnih sinusnih oscilacija. Tri senzora su vezana zajedno i fiksirana na metalnu rešetku koja je montirana na platformu. Oni su položeni na stranu, tako da je x osa izložena najjačim vibracijama, koje se vrše u vertikalnom pravcu. Glavni razlog za ovo je da se minimiziraju kašnjenja u propagaciji oscilacija, kada se senzori nalaze na identičnoj udaljenosti od platforme. Sporedni razlog je činjenica da x osa LIS3LV02DL ima nešto manji šum oko  $\pm g$  ofseta, dok se šum na z osi povećava kada je izložena  $\pm g$ .



**Slika 7.6** Normalizovana korelacija (levo) i histogram broja pojavljivanja maksimalne korelacije (desno), u funkciji vremenskog pomaka, za eksperimente na stolu

Kros-korelacija između parova signala je posmatrana (vršena je TDE analiza dva signala u vremenskom domenu). Pošto uzorci na različitim uređajima nisu uzimani istovremeno, jedan signal je interpoliran linearno tako da se uzorci poklope u vremenu sa onima sa drugog uređaja. Direktna komponenta signala je uklonjena i proizvod (korelacioni integral) je računat za različite vremenske pomake. U slučaju ispravnog vremenskog označavanja i identične pobude, signali će biti približno identični i maksimum korelacije treba da se dobije za nulti vremenski pomak. U suprotnom, TDE



**Slika 7.7** Senzorski uređaji na vibracionoj platformi

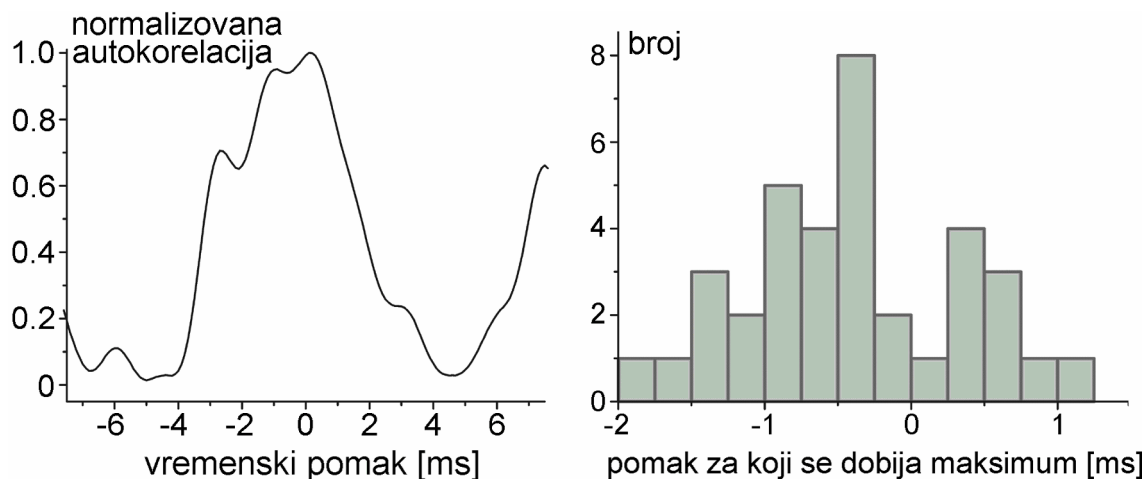
analiza pokazuje vremenski ofset između signala nalaženjem maksimuma korelacije. Rezultati prvog eksperimenta, na stolu, su prikazani na sl. 7.6 gde vidimo normalizovanu srednju korelaciju na levoj strani i histogram koji pokazuje broj maksimuma korelacije za odgovarajuće pomake. Proračun je izvođen sa korakom od 1 ms, a frekvencija odabiranja je bila uobičajenih 160 Hz (period od 6 ms). Statistička analiza pokazuje

sistemska ofset od 1 ms i standardnu devijaciju od 4 ms. Pošto su ovi parametri manji od perioda odabiranja možemo da zaključimo da se rezultati, iako nisu idealni, nalaze unutar očekivanih granica. Sekundarni maksimumi na dijagramu korelacije se objašnjavaju rezonantnim efektima na stolu.

U drugom eksperimentu, na vibracionoj platformi (eksperimentalna postavka je prikazana na sl. 7.7), frekvencije preko 200 Hz su izazivale jače sopstvene oscilacije rešetke, ali ovo nije mnogo uticalo na rezultate u pogledu korelacije u vremenu (mada je uočljivo da senzor u sredini, budući da se nalazi u čvoru stojećeg talasa, pokazuje veće amplitude ubrzanja). Jedan od važnih zaključaka iz ovog eksperimenta je da su uređaji dovoljno robusni jer su svi preživeli nekoliko sati intenzivnog izlaganja ubrzanjima od 0,3 do 1g u opsegu frekvencija od 5 do 700 Hz, bez oštećenja. Analiza korelacije je vršena primenom koraka od 0,1 ms, pošto je ovde korišćena frekvencija odabiranja od 2560 Hz (maksimum akcelerometra). Sasvim očekivano, najgori rezultati su dobijeni za par 1-3 senzorskih uređaja. Ovo su uređaji na krajevima (najdalje međusobno). Rezultati za ovaj par su prikazani na sl. 7.8. Oblik normalizovane autokorelacije je, prirodno, zavisano od mešavine frekvencija koje su se koristile u eksperimentu, a koja je proizvoljna. Srednji pomak za maksimalnu korelaciju (sistemska ofset) iznosi 0,42 ms a dobijena je standardna devijacija od 0,71 ms. Vidimo da su ovi rezultati bolji po



apsolutnoj vrednosti, ali lošiji kada se uporede sa periodom odabiranja od 0,4 ms. Iako su daleko od savršenih, oni pokazuju domete ove vrste opreme. Treba imati u vidu da su senzori međusobno udaljeni više santimetara i da su stavljeni na metalnu rešetku koja unosi dodatne modove vibracija.



**Slika 7.8** Normalizovana korelacija (levo) i histogram broja pojavljivanja maksimalne korelacije (desno), u funkciji vremenskog pomaka, za eksperimente na vibracionoj platformi

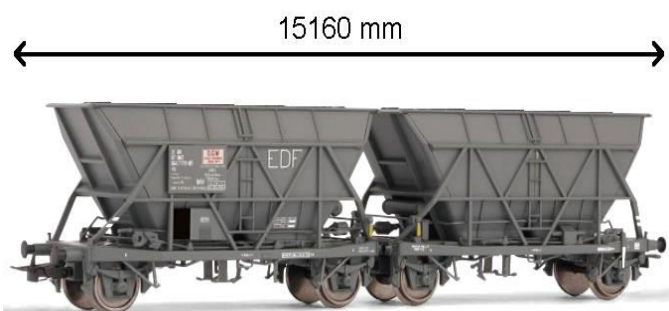
Izvršen je i određen broj eksperimenata sa frekvencijom samplovanja od 160 Hz. Statistički, rezultati su slični kao kod eksperimenata sa manuelnom pobudom. Sistemski ofset je 0 ms, a standardna devijacija je 4 ms (identična). Takođe, vršeni su i eksperimenti na 2560 Hz sa sensorima naslaganim vertikalno (jedan preko drugog). Uočava se vreme propagacije (sistemski ofset) od 0,8 ms po paru. Standardna devijacija je porasla na 1 ms, što se objašnjava činjenicom da ovog puta nisu svi senzori bili u kontaktu sa rigidnom metalnom podlogom, čime su dobili veću slobodu kretanja.

## 8. PRIMENA SISTEMA U ŽELEZNICI

Mali broj proizvedenih senzora ne omogućava da se oni upotrebe za efektanu modalnu analizu vibracija mostova i brana, ali jedna od mogućih primena sistema sa manjim brojem senzora jesu merenja na prugama. Ustanovljeno je da su senzorski uređaji pogodni za merenje brzine vozova ako je poznata geometrija vagona, sa dobrom



**Slika 8.1** Pruga za prevoz uglja u termoelektranu Nikola Tesla u Obrenovcu



**Slika 8.2** Par koševa koji čine teretni dvodelni EDF vagon za prevoz uglja

tačnošću, i za orijentacionu procenu dinamičkog ugiba pragova.

Merenja su vršena na pruzi nedaleko od termoelektrane Nikola Tesla u Obrenovcu (sl. 8.1). Ovom prugom se termoelektrana snabdeva ugljem iz površinskog kopa Kolubara. Snimljeno je po nekoliko punih i praznih vozova (koji se vraćaju istim putem iz termoelektrane). Za prevoz se koriste teretni dvodelni EDF vagoni, prikazani na sl. 8.2. Svaki vagon se sastoji od dva koša. Dužina vagona, odnosno para koševa, iznosi nešto preko 15 m. Masa im iznosi 25 t. Uobičajeno se tovari oko 60 t uglja, dok je maksimalna nosivost 100 t.

### 8.1. Značaj i tehnike merenja pomeraja pragova

Trajna oštećenja pragova i drugih elemenata pruga nastaju kao posledica niza uzroka. Prirodne pojave kao što su klizanje zemljišta i elementarne nepogode mogu doprineti ubrzanom starenju pruga. U toku regularne eksploatacije, oštećenja su

uglavnom posledica prevelikih opterećenja, više dinamičkih nego statičkih. Statička opterećenja podrazumevaju vozila suviše velike težine. Dinamička podrazumevaju kratka udarna opterećenja. Ova najčešće nastaju kao posledica defektnih točkova [109]. Točkovi nepravilnih dimenzija uzrokuju permanentna i potencijalno ozbiljna oštećenja, posebno na šinama. Ponovljena kratka udarna opterećenja, koja neravnine proizvode, pospešuju nastajanje i ubrzavaju rast pukotina u metalu [110], što skraćuje rok upotreba pruga i povećava šanse za katastrofalne nesreće. Drugi značajni uzroci oštećenja pruga prilikom eksploatacije su loš kvalitet osovina i kočnica vozila i prevelike bočne oscilacije kompozicija [111].

Merenje pomeranja pružnih pragova pod opterećenjem može pružiti relevantne informacije inženjerima u oblasti železnice pri proračunima krutosti zastora (ili balasta) i modula elastičnosti šinske podloge, kao i za proveru



**Slika 8.3** Laserski sistem za merenje statičkog ugiba šina

efikasnosti procedura održavanja pruga, kao što su čišćenje i podbijanje zastora. Krutost zastora ima značajan uticaj na njegovu brzinu sleganja, raspodelu opterećenja, dinamička opterećenja, i naprezanja određenih komponenti. Takođe, pragovi utiču i na stvaranje komunalne buke, čijem se smanjenju teži [112,113].

Praktična merenja pomeranja pragova su u stvarnosti povezana sa brojnim teškoćama. Najveća je obezbediti dobar referentni sistem, jer su svi delovi pružne konstrukcije podložni vibracijama. Vibracije se prenose i na okolno tlo na koje se eventualno postavljaju delovi mernog sistema. Neki često korišćeni senzori za merenje pomeraja nisu toliko efikasni u slučaju merenja pomeraja (ugiba) praga ili drugih elemenata. U ovu klasu spadaju induktivni, kapacitivni i otporni pretvarači, sistemi sa oprugama, i dr [114]. Da bi bili efikasni, ovi sistemi najčešće moraju da se ugrade duboko u tlo ispod pruge što predstavlja poteškoću. Ako se mere pomeranja pri visokofrekventnoj pobudi, situacija postaje još složenija usled ograničenja dinamičkih karakteristika ekonomskih senzora što uslovljava potrebu za skupljim mernim

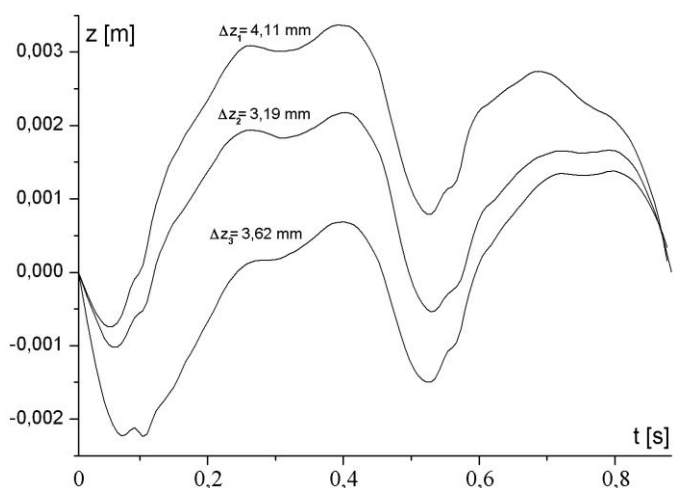
sistemima. Takođe, optoelektronski sistemi na bazi lasera često se koriste za merenje pomeraja [115]. I u njihovom slučaju javlja se problem referentne tačke. Ako je laser bliže mernoj tački, vibracije su jače, a ako je dalje, preciznost merenja opada. Primer ovakvog sistema je prikazan na sl. 8.3.

Ako se upotrebljavaju akcelerometri, prestaje potreba za spoljnom referencom jer oni mere inercijalno ubrzanje kome su izloženi. Brzina i položaj praga se mogu odrediti metodom jednostruke odnosno dvostruke integracije ubrzanja po vremenu. Mana ovog pristupa je što se šum akumulira i dolazi do slučajnog odstupanja koje raste sa vremenom, te su potrebne softverske metode korekcije [116]. Softverske korekcije nivoa se vrše po periodama, tj. iskorišćavaju informaciju da se pragovi (ili bilo koje druge oscilujuće strukture) vraćaju u početni položaj posle isteka periode, odnosno prolaska vagona ili kompozicije.

## 8.2. Mogućnosti merenja dinamičkog ugiba pragova postojećim sistemom

Dvostruka integracija signala da bi se dobilo pomeranje pragova u slučaju upotrebe razvijenog bežičnog sistema ne daje sasvim zadovoljavajuće rezultate. Ispostavilo se da je potrebna veća frekvencija odabiranja da bi se nagli udarci snimili dovoljnom rezolucijom za digitalnu integraciju. Takođe, postoji sumnja da je potrebno bolje pričvršćivanje akcelerometara za podlogu, jer se pri nekim udarcima javljaju kratka ubrzanja preko  $g$ , što znači da pri ubrzanjima praga na dole može da dođe do zaostajanja akcelerometra.

Ipak, orijentaciona procena amplitude pomeranja dela praga na kome je postavljen senzor (ubrzanja i ugib ne mogu biti identični duž cele površine praga) može se dati i analizom postojećih snimaka. Nije bitno koje tačke ćemo izabrati za



**Slika 8.4** Primeri određivanja vertikalnog pomeraja (dinamičkog ugiba) dvostrukom integracijom za slučaj relativno pravilnog signala

početak i kraj periode, ako je njena vrednost poznata. Usled nesavršenosti merenja, koje uključuje histerezis, nelinearnost, i razne druge greške, jednostrukom i dvostrukom integracijom u vremenu preko jedne periode se ne dobija tačno nula, iako bi brzina i pomeraj praga trebalo da se vrate na inicijalne vrednosti. Naravno da postoje i fizički razlozi za odstupanja, kao sporiji harmonici oscilacije kompozicije i strukture, i razlike između vagona. Iz ovih razloga uvode se pomenute korekcije, koje treba da pomeraju srednji nivo signala tako da se integracijom dobije isti položaj na početku i kraju periode. Metod koji je ovde korišćen jeste linearno pomeranje nultog nivoa ubrzanja u funkciji vremena (unutar jedne periode prolaska vagona). Varijacije u ovom postupku su moguće (npr. korišćenje fiksnog ofseta za svaku periodu). Otkrivanje najboljeg modela za korekciju predstavlja moguć budući pravac istraživanja.

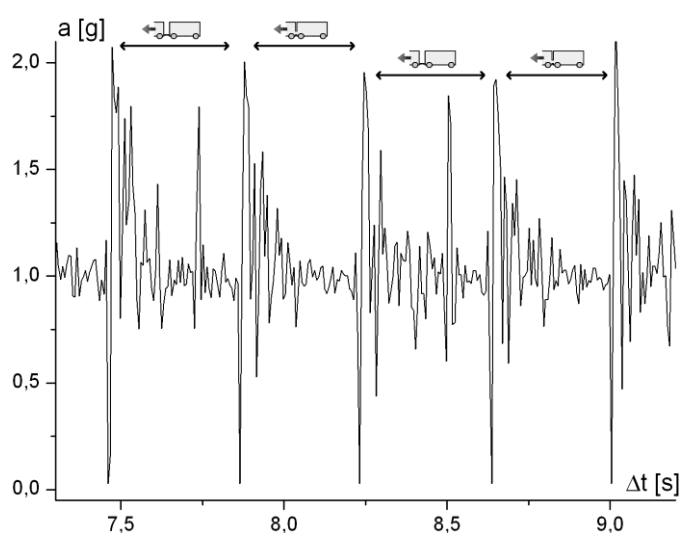
Na sl. 8.4 prikazani su primeri određivanja dinamičkog vertikalnog ugiba praga ( $z$ ) metodom dvostruke integracije sa korekcijom za tri različita perioda (vagona) na istom snimku punog voza (za koji je određena brzina od 62,52 km/h). Vrednosti na osama su relativne (referentna vrednost  $z$  nije određena, a vreme je brojano u odnosu na početak periode).  $\Delta z$  predstavlja odstojanje između dva krajnja položaja ( $z_{max}-z_{min}$ ). Iako su razlike između signala značajne, obrazac se jasno uočava, i devijacije amplituda pomeraja za slučaj punog voza su reda veličine 10%. Možemo da zaključimo da je amplituda pomeraja (polovina  $\Delta z$ ) oko 2 mm.

Najveće devijacije se dobijaju u slučaju praznog voza na pragu lošijeg kvaliteta (signal koji je prikazan na sl. 8.6). Ovde obrazac nije uočljiv a amplitude variraju i do 1:2,5 (što bi, obzirom na nepravilan vremenski dijagram, moglo biti i dobrim delom istinito). Možemo da zaključimo da je amplituda pomeraja u ovom slučaju maksimalna i iznosi oko 1 cm.

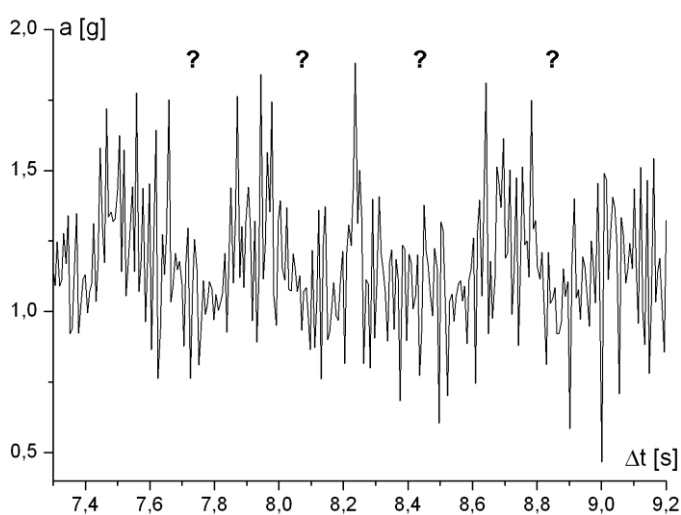
### **8.3. Određivanje brzine vozova**

Frekvencija odabiranja akcelerometara bila je 160 Hz a merni prozor 20 sekundi. Tri senzorska uređaja u svojim kućištima bila su zalepljena lepljivom trakom za pružne pragove, sa spoljne strane u odnosu na šine, na međusobnom odstojanju od po 3 m. Oblici signala variraju od praga do praga, i u zavisnosti od toga da li prolazi pun ili prazan voz. Za pune vozove oscilacije su manje amplitude, iako su statički ugibi u tom slučaju veći. Vertikalne oscilacije imaju amplitudu oko 0,5g u slučaju punih vozova

odnosno oko 1g u slučaju praznih. Tačne vršne vrednosti ubrzanja je teško precizno izmeriti zato što propusna moć akcelerometara nije bila dovoljna da obezbedi pouzdano beleženje kratkotrajnih impulsa velike sile. Ubrzanja u ostalim pravcima su tipično za red veličine manja nego vertikalna, i idu do 0,25g za slučaj praznih vozova. Dva parcijalna dijagrama vertikalnog ubrzanja u funkciji vremena, za prazne vozove (jače vibracije), data su na sl. 8.5 (pravilan dijagram dobijen na “boljem” pragu) i sl. 8.6 (nepravilan dijagram dobijen na “lošijem” pragu). Ustvari, fizički kvalitet drveta praga nije od presudnog značaja, već se pravilniji dijagrami dobijaju ako je prag dobro podbijen, tj. ako se težina pravilnije raspoređuje po površini.



Slika 8.5 Primer 1: vertikalno ubrzanje praga



Slika 8.6 Primer 2: vertikalno ubrzanje praga

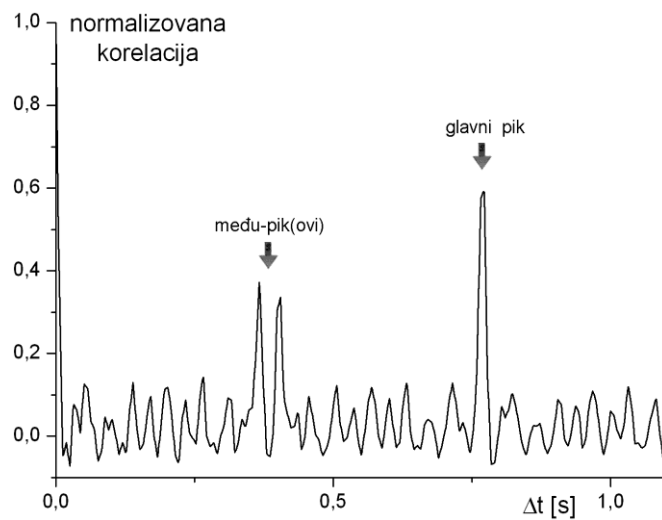
Vidimo da se na prvom primeru jasno uočava obrazac po kome se koševi i vagoni mogu prepoznati. Tako je moguće izračunati brzinu vozila deljenjem dužine vagona sa vremenom prolaska. Tačna dužina vagona sa kvačilom može malo varirati (u cm, ne dm) jer veza nije kruta tako da se procena brzine voza na osnovu poznavanja geometrije vagona može dati sa preciznošću reda veličine nekoliko promila. Pretpostavljeno je da ostale neodređenosti imaju za barem red veličine manji uticaj na ovaj proračun. Relativna neodređenost frekvencije sata senzorskog uređaja je u najgorem slučaju  $10^{-4}$  (pod uslovom da se ne koristi nijedna od opisanih korekcija, i da se koriste

procesori iz različitih setova) a uz kalibraciju i temperatursku softversku korekciju ili metodom brojanja na licu mesta pada i ispod  $10^{-6}$ . Neodređenost dimenzija usled sezonskih varijacija temperature i starenja je reda veličine  $10^{-4}$  relativno. Neodređenost izlaza digitalnog akcelerometra zbog

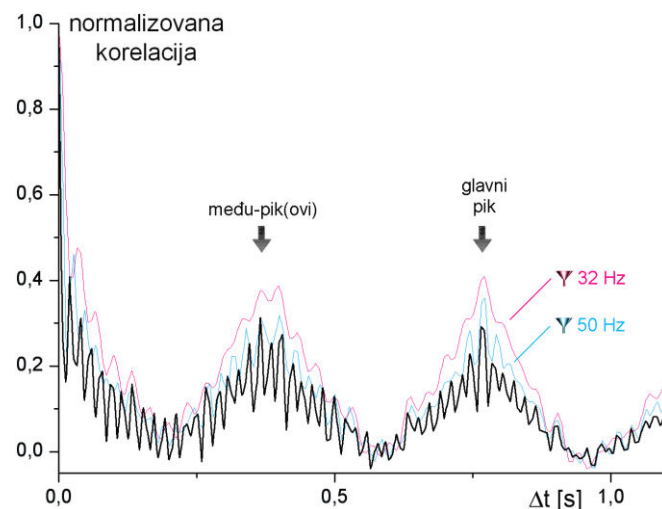
temperaturskih razlika ne utiče na proračun jer amplituda signala nije povezana sa određivanjem brzine (inače osetljivost može da varira do  $2,5 \cdot 10^{-4} \text{ } ^\circ\text{C}^{-1}$  pune skale, koja u ovom slučaju iznosi  $\pm 2\text{g}$ , prema proizvođačkoj specifikaciji). Iz prvog primera dobija se brzina voza od 70,69 km/h.

I u slučajevima kada signal nije vizuelno prepoznatljiv, kao na sl. 8.6 (drugi primer; prikazan je isti vremenski prozor kao na sl.

8.5, snimljen na drugom senzoru koji se nalazio na uslovno lošije podbijenom pragu), može se upotrebiti TDE tehnika za određivanje brzine voza. Na sl. 8.7 i sl. 8.8 vidimo normalizovanu korelaciju dobijenu TDE metodom za ova dva primera. U prvom slučaju, kada je signal pravilniji, vrhovi koji odgovaraju dvodelnim vagonima su jasni, i javljaju se za razmake koji se mogu lako uočiti na originalnom vremenskom dijagramu. Među-vrhovi su očekivano dvostruki i nižeg intenziteta, što savršeno odgovara geometriji sistema. Naime, usled različitih razmaka između osovin na spoju koševa nego na spoju vagona, situacija je slična kao da prolaze jedan nešto duži i jedan nešto kraći vagon. Delimično poklapanje signala se javlja kada se prednja ivica dužeg šiftuje

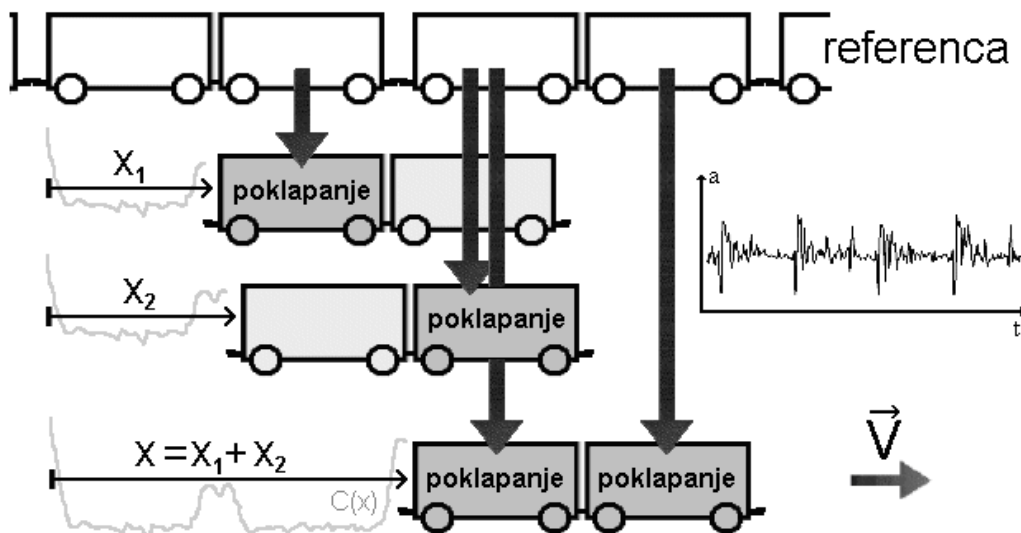


**Slika 8.7** Primena TDE metoda na signal iz primera 1



**Slika 8.8** Primena TDE metoda na signal iz primera 2

tako da se poklopi sa prednjom ivicom kraćeg, a zatim i kada se zadnja ivica prednjeg poklopi sa zadnjom ivicom kraćeg, što u slučaju male razlike u dužini rezultuje u dva bliska pika (ovo je skicirano na sl. 8.9, gde su koševi koji se poklapaju obeleženi strelicama). U drugom slučaju, kada je signal slabijeg kvaliteta, glavni i među-pikovi se



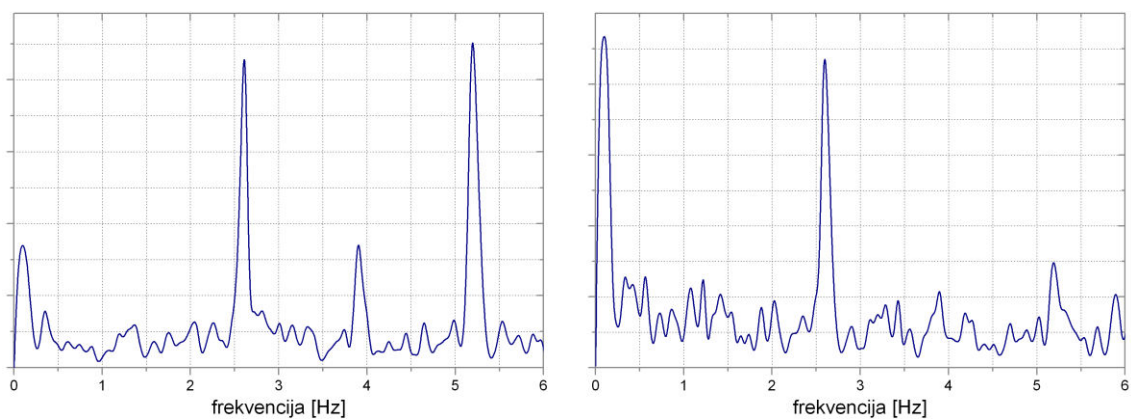
Slika 8.9 Vizuelni prikaz nastanka „među“ i glavnih pikova korelacije

ne mogu razlučiti po intenzitetu, ali nam saznanje da očekujemo među-pikove omogućuje da identifikujemo stvarne glavne pikove i dobijemo približno isti rezultat za brzinu kao i u prvom slučaju, a to je 70,97 km/h. Razlika u odnosu na prvu vrednost odgovara vremenskoj greški od jedne trećine perioda odabiranja akcelerometra (6 ms), odnosno 2 ms. Pomeranje signala u vremenu (korak TDE analize) je bio 1 ms.

Nešto uočljivija razlika između pravog i „među“ korelacionog maksimuma u slučaju signala slabijeg kvaliteta dobija se pri proučavanju slabijih komponenti ubrzanja koje su snimljene na x i y osama, ali mali intenzitet uz prisustvo viših harmonika može učiniti softversku detekciju značajnim problemom. Takođe, podaci se mogu digitalno filtrirati filtrom propusnikom niskih frekvencija, pri čemu dijagram sa sl. 8.8 neće imati značajno različit nivo među-vrhova i glavnih vrhova, ali će vrhovi postati prepoznatljiviji. U zavisnosti od primenjenog filtra, među-vrhovi, za razliku od glavnih, gube oštrinu i postaju dvostruki.

Na sl. 8.10 vidimo rezultate diskretne Furijeove transformacije (primenjene na signale iz primera 1 i 2, redom). Jasno se vidi efekat „maskiranja” osnovne frekvencije višim harmonicima, te da je analiza metodom autokorelacije (TDE) superiorna u





**Slika 8.10** Dijagrami amplitude u funkciji frekvencije, dobijeni DFT-om, za signale iz primera 1 i 2

ovakvim slučajevima. Frekvencija prelazaka vagona preko pragova (oko 1,3 Hz na sl. 8.10) ne uočava se ni na jednom tipu dijagrama ni u jednom slučaju (dobar ili loš prag, pun ili prazan voz).

Međusobna korelacija signala sa različitih pragova ne može da se upotrebi za određivanje brzine voza, kao ni za bilo kakvu drugu analizu, usled prevelikih razlika između snimljenih signala koje su posledica velikih fizičkih razlika između pragova (prvenstveno kvaliteta podbijanja).

## **9. PROVERA OPRAVDANOSTI TDE-DPCM PREPROCESIRANJA U REALNIM MERENJIMA**

U slučajevima kada su oscilacije monitorisane strukture pravilne, dobri rezultati se mogu postići primenom TDE-DPCM preprocesiranja. Najbolji primer regularnih vibracija su one proizvedene na vibracionoj platformi, opisane u poglavlju 7.7.

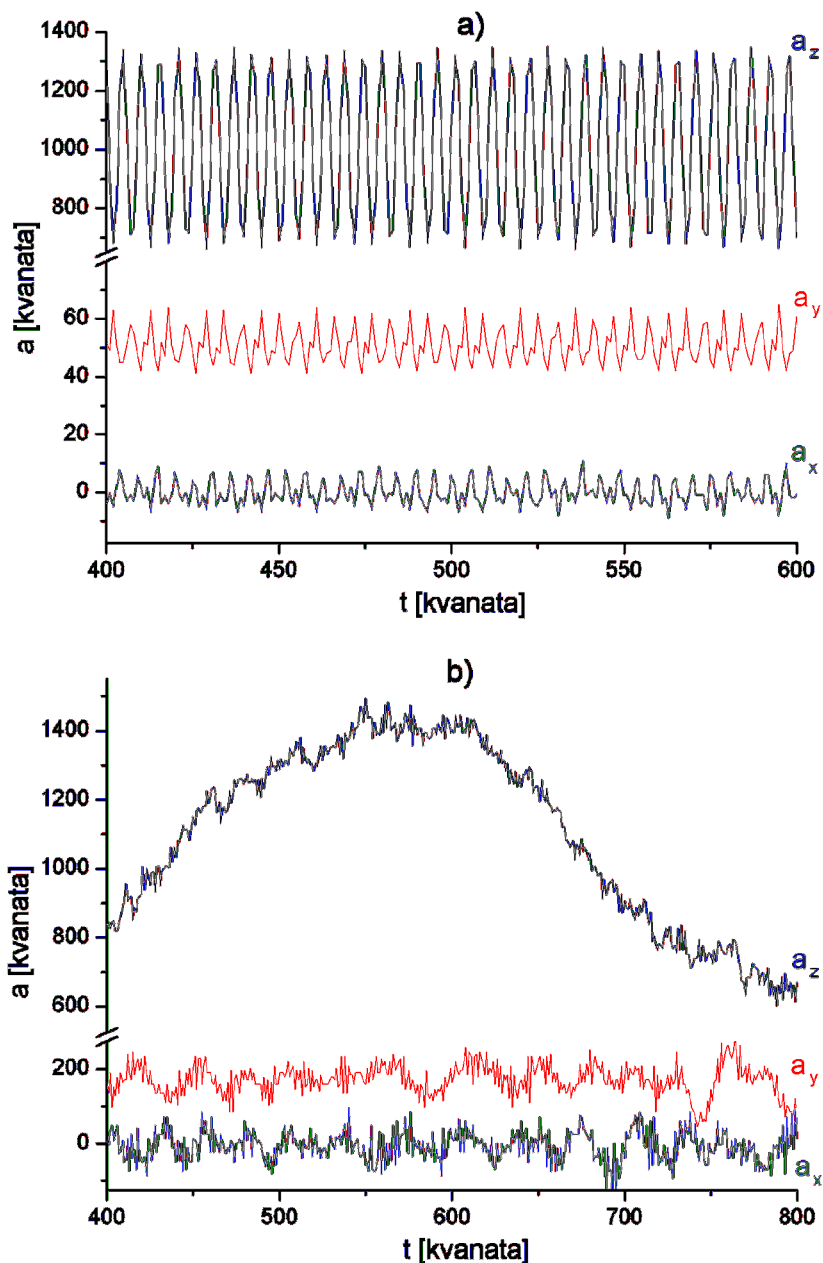
Ose dijagrama su obeležene u digitalnoj formi (u vremenskim i kvantima ubrzanja) jer njihove fizičke vrednosti nemaju nikakvog značaja za proces kompresije. Jedan od dva sekundarna signala (x ili y) je pomeren vertikalno da bi se poboljšala preglednost (oba u svim eksperimentima imaju srednju vrednost blizu nule). Odnosi kompresije su prikazani za slučaj odvojenih Hafmanovih tablica (jedna za glavnu osu i jedna za sekundarne ose).

Preko 50 eksperimenata je izvršeno na Brabender DS1500 vibracionoj platformi. Došlo je do značajnog poboljšanja odnosa kompresije primenom TDE-DPCM metode. U eksperimentima sa učestanošću odabiranja od 160 Hz (niži šum), kompresioni odnosi su smanjeni za 7-36% njihove originalne vrednosti, što znači da su u nekim slučajevima smanjeni više nego dva puta (npr. sa 71% na 35%). Šum raste sa frekvencijom sampionovanja tako da se i oblik signala menja, kao što je ilustrovano na sl. 9.1. Na donjem dijagramu (sl. 9.1.b), frekvencija od 2560 Hz je iskorišćena da bi se snimile vibracije relativno niske frekvencije. U takvim slučajevima, dolazi čak i do pogoršanja kompresionog odnosa, maksimalno za 1%. Rezultati se poboljšavaju povećanjem frekvencije vibracija i opet se dobijaju smanjenja odnosa kompresije do 32%.

Normalizovana autokorelacija za glavnu osu je dobra, kao što je očekivano, zbog relativne regularnosti signala. Ona ide od 78% do preko 100%. Moguće je dobiti vrednosti veće od 100% ako amplituda signala raste u smeru pomeranja dela signala (prozora) s kojim se vrši integraljenje u toku TDE procedure. Ova vrednost ne daje dobru indikaciju kompresionog odnosa. Međutim, ako se uzmu u obzir normalizovane autokorelacije za sekundarne ose, veza postaje očigledna. Dobri dobici kompresionog odnosa koincidiraju sa visokim autokorelacijama na sekundarnim osama.

Rezultati, prikazani u tabeli 8 i na sl. 9.1, su zavisni od mešavine frekvencija koja se koristi u eksperimentu. Za 1% bolje kompresije nego što su originalno dobijene, dobijaju se na PC računaru kada se fajlovi ponovo procesiraju, uz primenu identičnog

algoritma ali sa „punim” pomeranjem prozora za sekundarne ose (za 20% ukupnog vremena, umesto usvojenih „skraćenih” 2% radi ubrzanja).

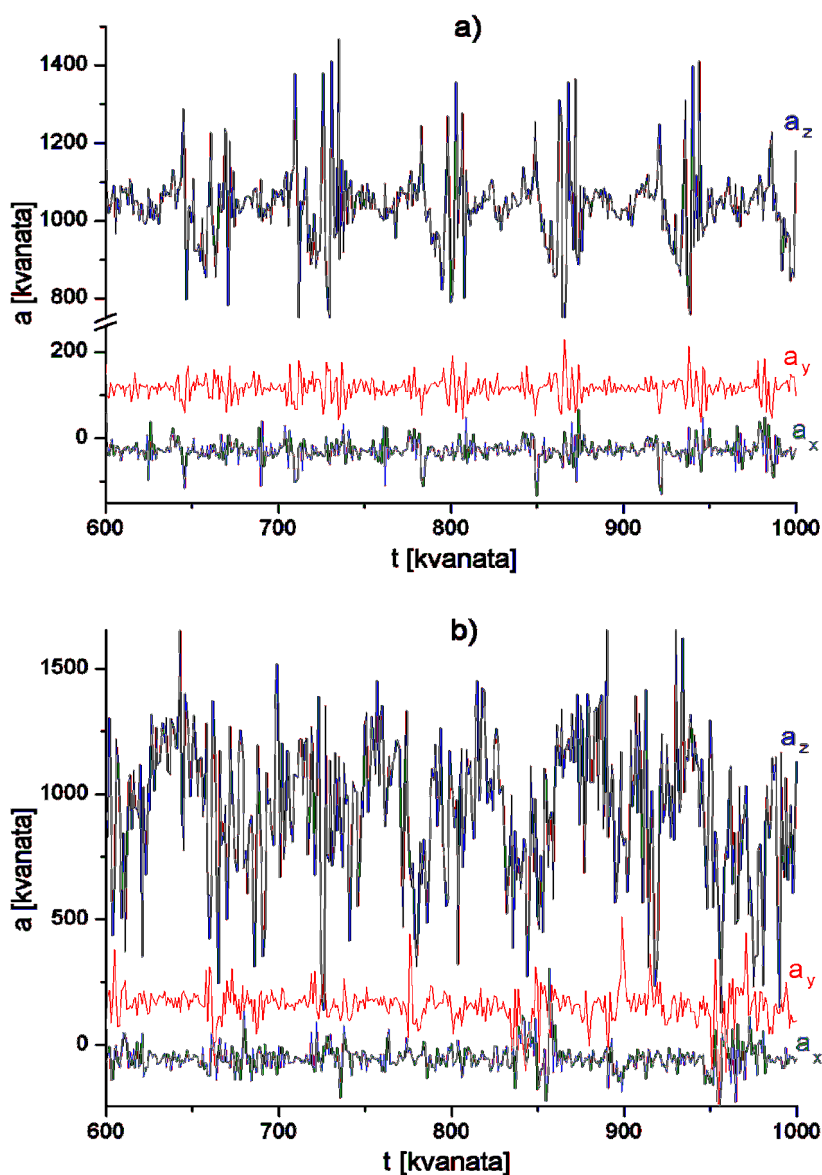


Slika 9.1 Signali sa vibracione platforme, snimani na 160 Hz (a) i na 2560 Hz (b)

Deset eksperimenata je izvršeno da se prouče vibracije drvenih pragova na pruzi za prevoz uglja do termoelektrane Nikola Tesla u Obrenovcu. Prazni vozovi proizvodili su intenzivnije vibracije, koje su imale dva neželjena efekta: akcelerometri su ulazili u zasićenje, i visoke razlike između susednih odmeraka (veće od 9 bita sa znakom) su se pojavljivale. Drugo je imalo za posledicu pogoršanje odnosa kompresije zbog

primenjenog algoritma redukcije Hafmanove tablice, opisanog u poglavlju 4.4.

Dobro podbijeni pragovi pokazuju pravilniji odziv na pobudu i može se vizuelno prepoznati obrazac prelaska vagona, za razliku od loše podbijenih pragova (sl 9.2). Ipak, TDE analiza daje tačne rezultate za period prolaska vagona u oba slučaja. Najveće poboljšanje odnosa kompresije primenom TDE-DPCM metode se dobija u slučaju da se TDE analiza izvrši samo po z osi, i rezultati primene na sve ose, ili ako se pun vremenski šift primeni za traženje perioda po svim osama. Nebitno je koja se varijanta upotrebi (prva je brža), jer se u oba slučaja korektni periodi detektuju za sve ose u većini slučajeva, a u preostalim slučajevima detektuju se multipli ove vrednosti, što



Slika 9.2 Vibracije pružnih pragova, dobro (a) i loše (b) podbijen prag

znači da se bolje poklapaju oscilacije udaljenih vagona nego susednih. Dobici za dobro podbijeni prag su 2-4% a za loše podbijeni prag 0-1%. Ako se algoritam optimizovan za većinu merenja primeni, svi rezultati se pogoršavaju za oko 0,5%, tako da je prosečan dobitak za loše podbijeni prag nula. Rezultati eksperimenata su prikazani u tabeli 8.

Normalizovana autokorelacija za glavnu osu je oko 75% za dobro podbijene pragove i 30-50% za loše podbijene pragove. Železnički inženjeri bi mogli da iskoriste ovu informaciju kao parametar pri utvrđivanju stanja pragova (u kombinaciji sa substrukturom). Ostale aplikacije merenja vibracija pragova uključuju procenu brzine voza i dinamičkog ugiba praga.

Zaključak iz eksperimenata sa vibracionom platformom i prugom je da u slučajevima kada je pobuda periodična, dobijeni period za glavnu osu treba da ima prioritet u odnosu na kraće periode koji se dobijaju TDE analizom na sekundarnim osama korišćenjem ubrzane procedure sa manjim pomeranjem integracionog prozora.

Tramvajski most (sl. 9.3) je jedini Beogradski most koji je preživio drugi svetski rat. Ima po jednu traku, sa tramvajskim šinama, u oba smera, koju koriste i automobili i tramvaji. Dužina mosta je 430 m a razmak između dva glavna betonska stuba koji su utisnuti u rečno dno iznosi 160 m.

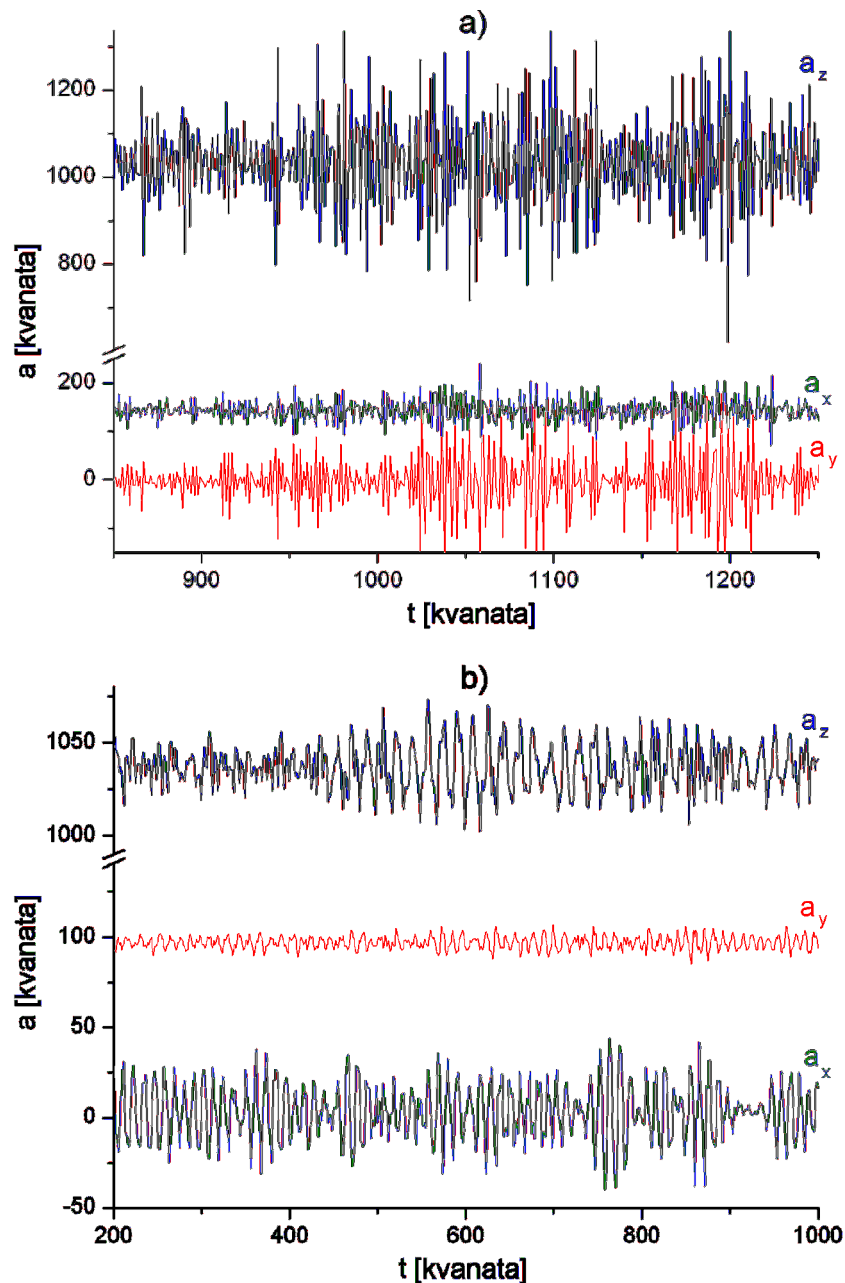


Slika 9.3 Tramvajski most u Beogradu

Vibracije koje proizvode tramvaji su veće amplitude i veće frekvencije nego kola, ali su regularnije, i TDE-DPCM metod daje poboljšanja 2-5% u svim

eksperimentima (po dvanaest je izvedeno za svaki tip eksitacije). Kola, međutim, unose varijabilne oscilacije. Samo četiri eksperimenta sa kolima koja su prolazila relativno uniformno tokom mernog perioda su uzeta u obzir i konstatovan je prosečan gubitak od 1%. TDE za sve ose je obavezan, pošto signali kao ovaj sa sl. 9.4 pokazuju različite rezonantne učestanosti po različitim osama. Eksperimenti nisu sprovedeni za vreme

špica, tako da je saobraćaj bio redak (većina vozača bira veće mostove bez šina) i koncentrisan (zbog semafora). Ovakva situacija nije pogodna za primenu TDE-DPCM metode jer signal nije kvaziperiodičan.



**Slika 9.4** Vibracije tramvajskog mosta prilikom tramvajске (a) i eksitacije od strane motornih vozila (b)

Normalizovana autokorelacija za sve ose se menja slučajno od 30 do 70% i ne vidi se veza sa dobtkom kompresionog odnosa. Razlog je što intenzitet vibracija pokazuje velike varijacije od eksperimenta do eksperimenta, pa je normalizovana autokorelacija više funkcija intenziteta nego regularnosti signala.



**Slika 9.5** Dilataciona sprava stuba mosta Gazela, izgled sa unutrašnje strane

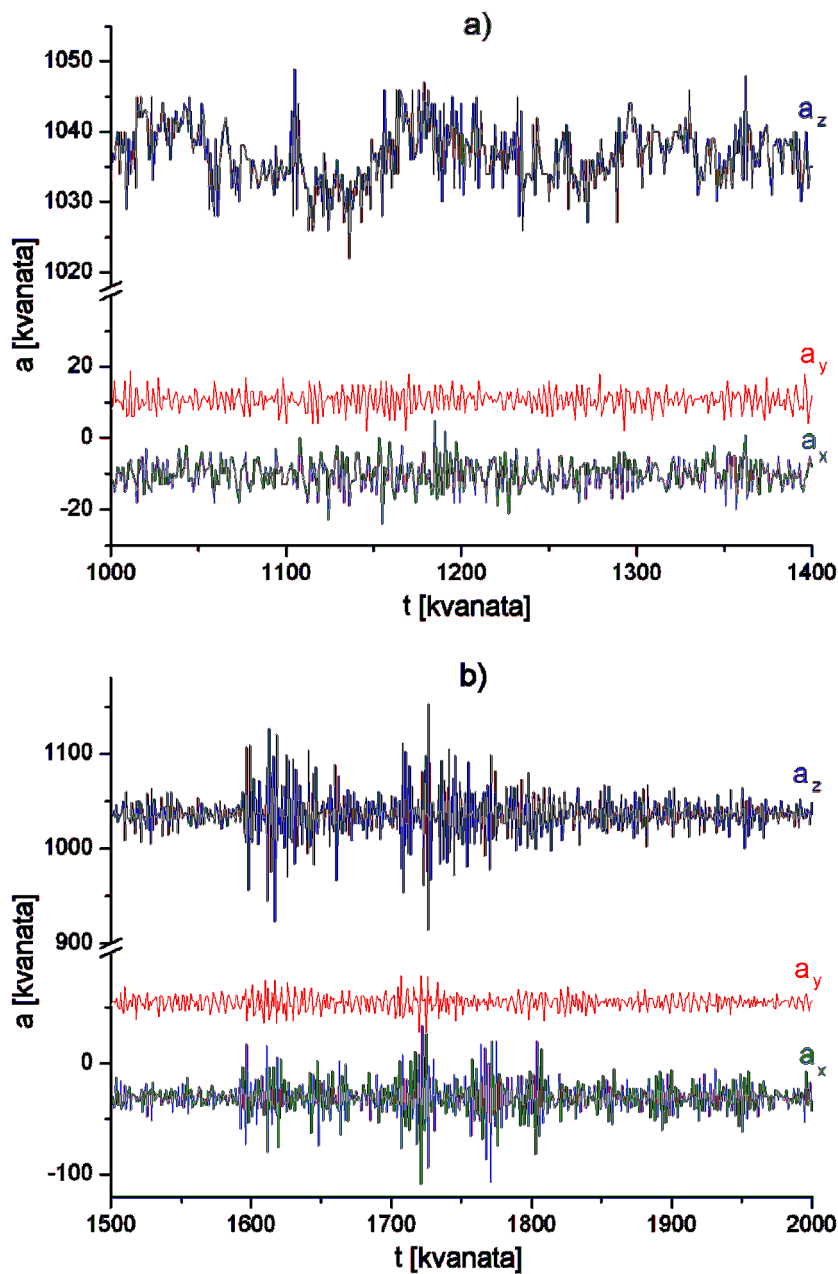
Most Gazela je dugačak 330 m sa glavnim rasponom između nosača od 250 m. Ima po tri trake auto-puta u oba smera (širina od 27,5 m). Sagrađen je krajem šezdesetih godina prošlog veka kao kombinacija grede i luka. U proseku, više od 160 hiljada vozila prođe dnevno preko njega, što znači da je vrlo preopterećen [117].

Vibracije su snimane u dva navrata, na sredini mosta i na i unutar nosećeg stuba. Primeri dijagrama vibracija su prikazani na sl. 9.6: jedan dijagram sa sredine

mosta (gore, sl. 9.6.a) i jedan dijagram sa stuba (dole, sl. 9.6.b). Rezultati su dati u tabeli 8. Prisustvo niskofrekventnog harmonika, koji je karakterističan za oscilacije celog mosta, se može uočiti na dijagramu sa sredine mosta. TDE nekada detektuje veliki period (postoji više harmonika frekvencije reda veličine oko Hz), ali češće prepoznaje više harmonike kao primarne. Empirijski postavljeni prag intenziteta autokorelacionog maksimuma od 25% za okidanje TDE-a na glavnoj osi se u slučaju sredine mosta pokazao kao nepodoban. Dvanaest eksperimenata sa dovoljnim intenzitetom signala pokazuju 1-4% poboljšanje odnosa kompresije, izuzev jednog sa pogoršanjem od samo 0,1%. Nažalost, polovina ima 15-25% normalizovanu autokorelaciju i fajlovi nisu procesirani na licu mesta već kasnije na PC računaru.

Na sl. 9.7 vidimo četiri tipična dijagrama dobijena Furijeovom analizom (amplituda, u relativnim jedinicama, u funkciji frekvencije). Osnovna frekvencija vertikalnih oscilacija centra mosta Gazela je oko 0,5 Hz, a izraženi su i harmonici na nešto preko 1 i 2 Hz. Harmonici na oko 3 i 4 Hz takođe mogu da se uoče.

Oscilacije na nosećem stubu su pak uglavnom odgovor mosta na udarce između kola i dilatacionih sprava (*expansion joints*) kojima se kolovoz mosta (između dva glavna noseća stuba) odvaja od ostatka kolovoza. Izgled ove naprave sa unutrašnje



Slika 9.6 Vibracije na sredini (a) i na stubu (b) mosta Gazela

strane prikazan je na sl. 9.5. Stub je čvor za oscilacije cele strukture tako da se niskofrekventni harmonici ne mogu detektovati. Kola koja pobuđuju stub ne uzrokuju sasvim identičan odgovor, pa postoje slučajevi kada se kompresija pogoršava, slično tramvajskom mostu pod uticajem kola, ali u proseku vidi se poboljšanje od 2%, iz preko 30 merenja izvršenih na raznim tačkama na i unutar stuba.

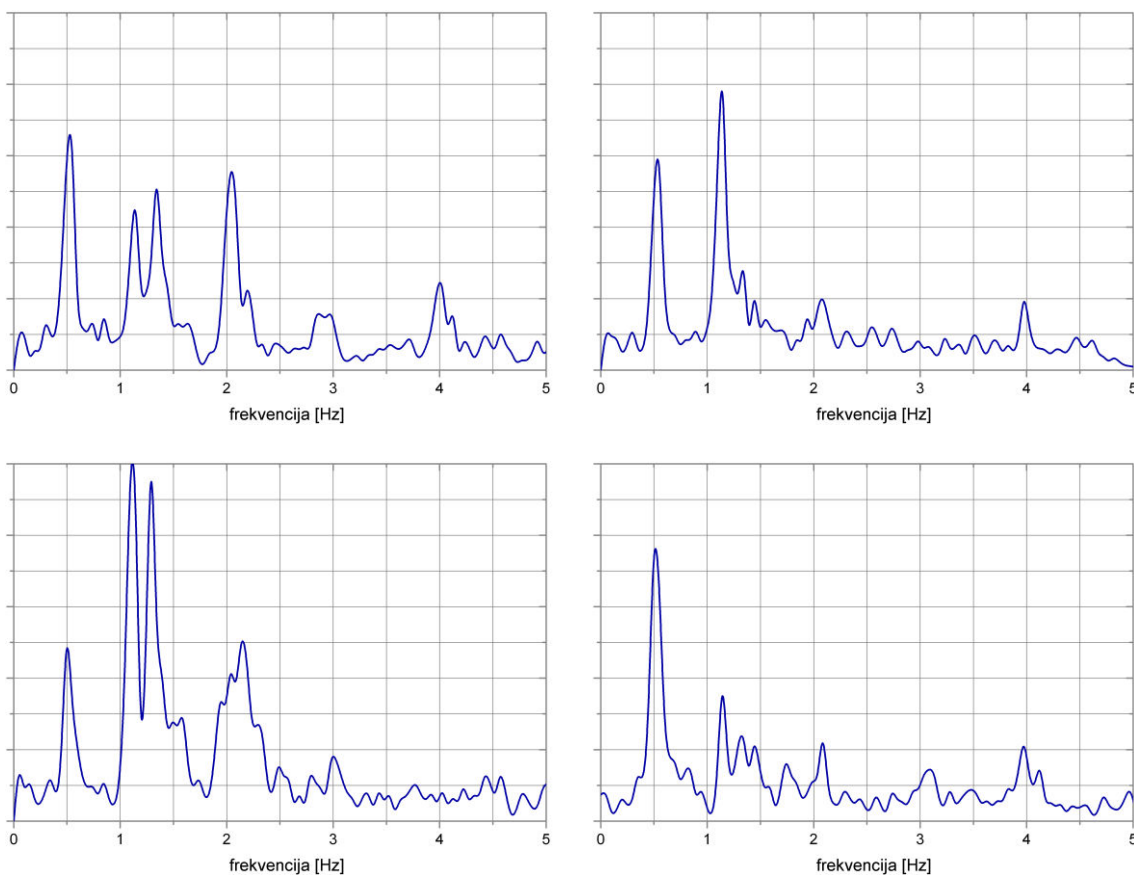
Prosečna normalizovana autokorelacija je oko 25% u oba seta eksperimenata na Gazeli. Neka merenja na sredini mosta nisu okinula TDE-DPCM preprocesiranje zbog



niskog intenziteta ovog parametra. Umesto ovoga, na PC računaru je korišćen samo usvojeni prag amplitude (srednja devijacija veća od tri kvanta merenja) da bi se ova merenja procesirala. Ovo ukazuje na potrebu da se vrše dodatna istraživanja uslova pod kojima je pogodno upotrebiti TDE-DPCM.

Prosečni odnosi kompresije i njihovi dobiti su grafički predstavljeni na sl. 9.8.

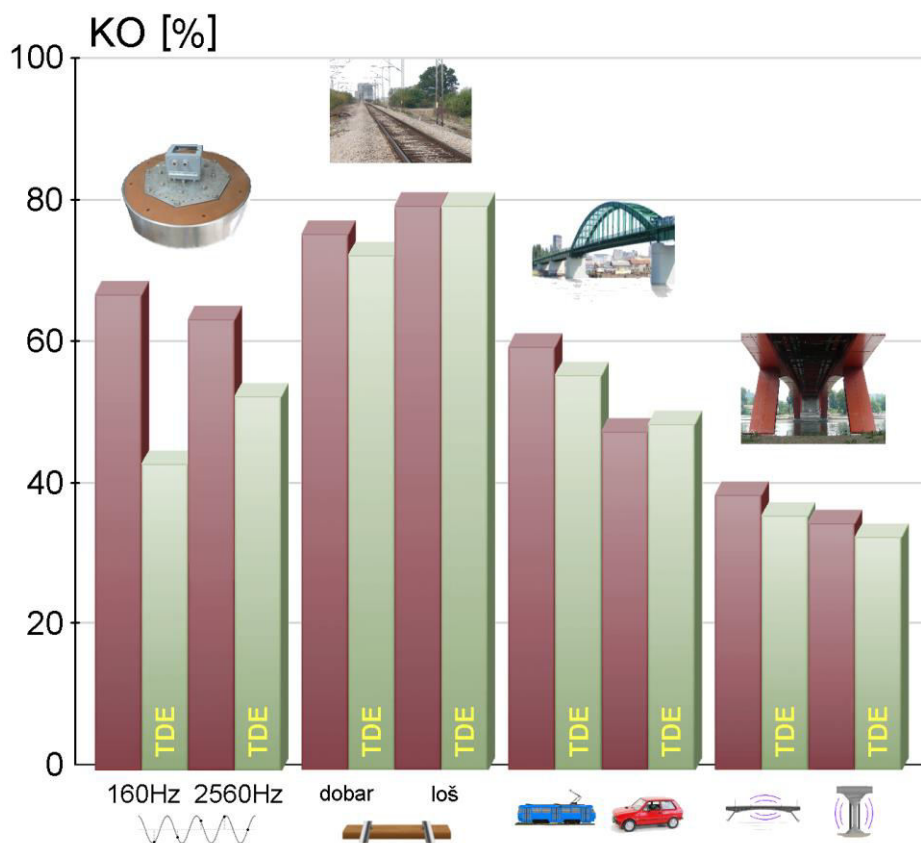
Pokušana je primena aritmetičkog kodiranja na senzorskom uređaju. Rezultati su nezadovoljavajući u smislu male brzine. Aritmetičko kodiranje je testirano na PC računaru, uz primenu istog TDE-DPCM preprocesiranja, i dobiti usled primene TDE-DPCM su praktično identični onima kod Hafmanovog kodiranja. Dobici upotrebom aritmetičkog umesto Hafmanovog kodiranja su, očekivano, nekoliko procenata.



**slika 9.7** Dijagrami amplitude u funkciji frekvencije, za signale snimljene na sredini mosta Gazela

**Tabela 8** Originalni kompresioni odnosi (KO) i dobici upotrebom TDE-DPCM preprocesiranja u odnosu na klasično DPCM preprocesiranje

tip signala	broj eksp.	originalni KO [%]		smanjenje KO primenom TDE-DPCM metode [%]		
		min.	maks.	minimalno	srednje	maksimalno
vibraciona platforma, 160 Hz	16	61	73	7	24	36
vibraciona platforma, 2560 Hz	40	48	79	-1	11	32
pun voz, dobar prag	4	71	79	1,5	2,5	3,5
prazan voz, loš prag	8	78	81	-0,5	0	0,5
tramvajski most, tramvaji	12	49	71	2	4	5
tramvajski most, kola	4	42	55	-6	-1	3
sredina mosta Gazela	12	36	40	0	3	4
stub mosta Gazela	32	29	58	-5	2	5



**Slika 9.8** Poređenje odnosa kompresije (KO) upotrebom DPCM susednih odmeraka i upotrebom TDE-DPCM

## 10. ZAKLJUČAK

U tezi je prikazan dizajn bežičnog senzorskog sistema za merenje vibracija u građevinarstvu, opisani su mnogobrojni eksperimenti kojima je izvršena karakterizacija sistema, i objašnjena su originalna rešenja i algoritmi koji se mogu koristiti u oblastima sinhronizacije merenja u udaljenim tačkama u bežičnoj mreži u realnim terenskim uslovima, kompresije podataka, i testiranja unakrsne osne osetljivosti MEMS akcelerometara.

Primarna svrha sistema je merenje vibracija kapitalnih građevinskih objekata kao što su mostovi, tornjevi, brane i tuneli. Ova merenja se vrše u okviru procene stanja konstrukcija, koja je bitna zbog povećanih potreba kako za osmatranjem zastarelih infrastrukturnih objekata tako i zbog tehnološki sve smelije građenih novih objekata. U cilju ove procene, najčešće se vrši detekcija karakterističnih osobina signala snimljenog akcelerometrima (mogu se koristiti i poluprovodničke merne trake ili razni drugi senzori), kao i upoređivanje amplituda i faza oscilacija pojedinih harmonika u različitim tačkama strukture. Bežični sistemi imaju velike prednosti u ovoj oblasti jer su fleksibilniji. Ipak, oni moraju da se nose sa različitim problemima, kao što je nedostatak zajedničkog takta (izvora vremena), ograničenje energetske rezervi, smetnje i sporost u komunikaciji, itd.

Senzorski uređaji, originalnog hardverskog dizajna, su optimizovani za malu potrošnju i mogućnost komunikacije u uslovima veštačkih smetnji, u prisustvu prepreka, i u zatvorenim prostorima. Energetsko dopunjavanje se može vršiti, u slučaju potrebe, panelom sastavljenim od varijabilnog broja solarnih ćelija. Razvijen je originalan softver (embedded RTOS) za senzorski uređaj, njegov displej (koji predstavlja nezavisnu napravu sa sopstvenim mikrokontrolerom), i softverski paket za centralnu stanicu (PC računar) koja kontroliše rad mreže senzora. Softver ima veliki broj parametara koji se mogu varirati, čime se postiže velika fleksibilnost sistema, i naravno otvoren je za izmene i dopune. Takođe, hardverski dizajn je u određenoj meri modularan, tako da je moguće dodavanje novih analognih ili digitalnih senzora.

Kompresija podataka je izvršena originalnom metodom TDE-DPCM, koja koristi DPCM preprocesor Hafmanovom entropijskom kodiranju, ali tako da kodira razlike između susednih perioda signala, pri čemu je obavezno detektovati koliki je

period signala i kolika je kvantitativna procena periodičnosti signala, i njegovog intenziteta, jer se u slučaju da uslovi nisu ispunjeni, vrši sekvencijalno DPCM preprocesiranje. Parametri ovog procesa su zavisni od vrste signala i dinamičkih karakteristika senzora, te mogu predstavljati predmet daljeg istraživanja. Brzina procedure i iskorišćenje ograničene memorije su poboljšani nizom tehnika koje uključuju *data overlaying*, programiranje isključivo u assembleru, i inteligentan izbor parametara integraljenja prilikom računanja autokorelacionog integrala.

Važna stavka u bežičnim sistemima koji služe za vršenje modalne analize jeste simultanost dobijenih podataka iz različitih tačaka. Da bi se garantovala simultanost reda veličina zanemarljivog dela periode najvišeg harmonika oscilacija od interesa, primenjeno je više metoda, koje osiguravaju eliminaciju neodređenosti izlaznog vremena (vremena transmisije, odnosno tzv. vremena paketizacije) radio modema, kao i određivanje odnosa frekvencija tajmera uređaja neposredno pre, ili za vreme izvršenja merenja, uz mogućnost dodatne softverske korekcije zbog kratkoročnih promena temperature. Stabilnosti frekvencija i različiti parametri koji utiču na nepreciznost ovih tajmera, zasnovanih na jeftinim kvarcnim kristalima, su detaljno analizirani.

Troosni akcelerometri iskorišćeni u sistemu imaju lošu osobinu visoke unakrsne (ili poprečne) osne osetljivosti, koja se može softverski komenzovati post-procesiranjem, pod uslovom da se izvrši tačna kalibracija parametara. Ovo je vrlo teško izvesti bez složene i skupe opreme, ali prosta metoda tilt-testa u 24 pozicije je primenjena kao najbolja alternativa koja ne zahteva nikakvu mernu opremu osim relativno ravne površine, i pokazano je da se efekti unakrsne osne osetljivosti mogu smanjiti više puta.

Osim merenja u oblasti vibracija velikih građevinskih objekata, sistem se može primenjivati i u mnogim drugim oblastima, a jedna od delimično istraženih jesu merenja na železnici. Pokazano je da se sistem može koristiti za merenje brzine vozova dobre preciznosti, i orijentacionu procenu dinamičkog ugiba (pomeraja) pragova. Pokazana je mogućnost da se TDE metoda primeni za određivanje stanja pragova (u kombinaciji sa pružnom strukturuom), ukoliko se primeni periodična, relativno uniformna pobuda, kao što je ona koja se javlja kada vagoni ravnomerno prelaze preko praga. Problem koji je specifičan za železnicu jeste ogroman broj elemenata koje treba ispitati, a koji su uz to i rasuti po velikom prostoru.

## 11. LITERATURA

- [1] Chang, P. C., Flatau, A., & Liu, S. C. (2003). Review paper: health monitoring of civil infrastructure. *Structural Health Monitoring*, 2(3), 257-267.
- [2] Brownjohn, J. M. (2007). Structural health monitoring of civil infrastructure. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 365(1851), 589-622.
- [3] Gavrilovska, L., Krco, S., Milutinović, V., Stojmenović, I., & Trobec, R. (2010). *Application and Multidisciplinary Aspects of Wireless Sensor Networks: Concepts, Integration, and Case Studies*. Springer London.
- [4] García-Hernández, C. F., Ibarguengoytia-Gonzalez, P. H., García-Hernández, J., & Pérez-Díaz, J. A. (2007). Wireless sensor networks and applications: a survey. *IJCSNS International Journal of Computer Science and Network Security*, 7(3), 264-273.
- [5] Ikegami, R., & Boller, C. (2009). History of SHM for commercial transport aircraft. In *Encyclopedia of Structural Health Monitoring*. Wiley online library.
- [6] Department of Civil and Environmental Engineering, University of Illinois (2009). Report NSEL-018. *Flexible smart sensor framework for autonomous full-scale structural health monitoring*. Urbana-Champaign, IL: Rice, J. A., & Spencer, B. F. Jr.
- [7] Lynch, J. P., & Loh, K. J. (2006). A summary review of wireless sensors and sensor networks for structural health monitoring. *Shock and Vibration Digest*, 38(2), 91-130.
- [8] Mukhopadhyay, S. C., & Ihara, I. (2011). Sensors and technologies for structural health monitoring: a review. In *New Developments in Sensing Technology for Structural Health Monitoring* (pp. 1-14). Springer Berlin Heidelberg.
- [9] Aygün, B., & Gungor, V. C. (2011). Wireless sensor networks for structure health monitoring: recent advances and future research directions. *Sensor Review*, 31(3), 261-276.
- [10] Chintalapudi, K., Fu, T., Paek, J., Kothari, N., Rangwala, S., Caffrey, J., Govindan, R., Johnson, E., & Masri, S. (2006). Monitoring civil structures with a wireless sensor network. *Internet Computing, IEEE*, 10(2), 26-34.

- [11] Bischoff, R., Meyer, J., & Feltrin, G. (2009). Wireless sensor network platforms. In *Encyclopedia of structural health monitoring*. Wiley online library.
- [12] Andrzejczak, A., Łeczycki, P., Makowski, M., Pekosławski, B., Pietrzak, P., & Napieralski, A. (2013). Wireless Communication Solution for Distributed Structural Health Monitoring. *International Journal of Electronics and Telecommunications*, 59(2), 177-184.
- [13] Gilbert, J. M., & Balouchi, F. (2008). Comparison of energy harvesting systems for wireless sensor networks. *International Journal of Automation and Computing*, 5(4), 334-347.
- [14] Potdar, V., Sharif, A., & Chang, E. (2009, May). Wireless sensor networks: A survey. In *Advanced Information Networking and Applications Workshops, 2009. WAINA'09. International Conference on* (pp. 636-641). IEEE.
- [15] Peterson, L. L., & Davie, B. S. (2007). *Computer networks: a systems approach*. Elsevier San Francisco.
- [16] Royer, E. M., & Toh, C. K. (1999). A review of current routing protocols for ad hoc mobile wireless networks. *Personal Communications, IEEE*, 6(2), 46-55.
- [17] Warneke, B. A., & Pister, K. S. (2002). MEMS for distributed wireless sensor networks. In *Electronics, Circuits and Systems, 2002. 9<sup>th</sup> International Conference on* (Vol. 1, pp. 291-294). IEEE.
- [18] Bhushan, B. (Ed.). (2010). *Springer handbook of nanotechnology*. Springer.
- [19] Levis, P., & Gay, D. (2009). *TinyOS programming*. Cambridge University Press.
- [20] Bhatti, S., Carlson, J., Dai, H., Deng, J., Rose, J., Sheth, A., Shucker, B., Gruenwald, C., Torgerson, A., & Han, R. (2005). MANTIS OS: An embedded multithreaded operating system for wireless micro sensor platforms. *Mobile Networks and Applications*, 10(4), 563-579.
- [21] Han, C. C., Kumar, R., Shea, R., Kohler, E., & Srivastava, M. (2005, June). A dynamic operating system for sensor nodes. In *Proceedings of the 3<sup>rd</sup> international conference on Mobile systems, applications, and services* (pp. 163-176). ACM.
- [22] Wanner, L. F., Hoeller, A. S., de Oliveira, A. B., & Frohlich, A. A. (2006, September). Operating system support for data acquisition in sensor networks. In *Emerging Technologies and Factory Automation, 2006. ETFA'06. IEEE*

- Conference on (pp. 582-585). IEEE.
- [23] Dunkels, A., Schmidt, O., Finne, N., Eriksson, J., Osterlind, F., Tsiftes, N., & Durvy, M. (2011). *The Contiki OS: The Operating System for the Internet of Things*. Retrieved from <http://www.contiki-os.org/>
- [24] Lorincz, K., rong Chen, B., Waterman, J., Werner-Allen, G., & Welsh, M. (2008, June). Pixie: An operating system for resource-aware programming of embedded sensors. In *Fifth Workshop on Embedded Networked Sensors (HotEmNets' 08)*.
- [25] Cao, Q., Abdelzaher, T., Stankovic, J., & He, T. (2008, April). The liteos operating system: Towards unix-like abstractions for wireless sensor networks. In *Information Processing in Sensor Networks, IPSN'08. International Conference on* (pp. 233-244). IEEE.
- [26] Eswaran, A., Rowe, A., & Rajkumar, R. (2005, December). Nano-RK: an energy-aware resource-centric RTOS for sensor networks. In *Real-Time Systems Symposium, 2005. 26<sup>th</sup> IEEE International* (10 pages). IEEE.
- [27] Sorber, J., Kostadinov, A., Garber, M., Brennan, M., Corner, M. D., & Berger, E. D. (2007, November). Eon: a language and runtime system for perpetual systems. In *Proceedings of the 5<sup>th</sup> international conference on Embedded networked sensor systems* (pp. 161-174). ACM.
- [28] Stanković, D. (1997). *Fizičko-tehnička merenja*. Beograd: Naučna knjiga.
- [29] Green, D., & Unruh, W. G. (2006). The failure of the Tacoma Bridge: A physical model. *American Journal of Physics*, 74(8), 706-716.
- [30] Guillaume, P. (2007). Modal analysis. *Vrije University Brussels*, [Online] <http://mech.vub.ac.be/avrg/publications/ModalAnalysis.pdf>
- [31] Huebner, K. H., Dewhirst, D. L., Smith, D. E., & Byrom, T. G. (2008). *The finite element method for engineers*. Hoboken, NJ: John Wiley & Sons.
- [32] Petyt, M. (2010). *Introduction to finite element vibration analysis*. Cambridge university press.
- [33] Cunha, Á., & Caetano, E. (2006). Experimental modal analysis of civil engineering structures. *Sound and Vibration*, 40(6), 12-20.
- [34] Hall, J. F. (2006). Problems encountered from the use (or misuse) of Rayleigh damping. *Earthquake Engineering & Structural Dynamics*, 35(5), 525-545.
- [35] Analog Devices (2012). *Aduc845 MicroConverter datasheet*, Retrieved from

- [http://www.analog.com/static/imported-files/data\\_sheets/ADUC845\\_847\\_848.pdf](http://www.analog.com/static/imported-files/data_sheets/ADUC845_847_848.pdf)
- [36] Shelake, V. G., Kamat, R. K., Parab, J. S., & Naik, G. M. (2007). *Exploring C for microcontrollers*. Springer Dordrecht.
- [37] STMicroelectronics (2008). *LIS3LV02DL datasheet*, Retrieved from <http://www.st.com/web/en/resource/technical/document/datasheet/CD00091417.pdf>
- [38] Dixon-Warren, S. J. (2010). Motion Sensing in the iphone 4: MEMS Accelerometer. *MEMS Journal*, Retrieved from <http://www.memsjournal.com/2010/12/motion-sensing-in-the-iphone-4-mems-accelerometer.html>
- [39] Decode (2009). *PRM4 datasheet*, Retrieved from [http://www.decode.rs/documentation/PRM\\_4-datasheet.pdf](http://www.decode.rs/documentation/PRM_4-datasheet.pdf)
- [40] Römer, K., Blum, P., & Meier, L. (2005). Time synchronization and calibration in wireless sensor networks. In I. Stojmenović (Ed), *Handbook of sensor networks: Algorithms and architectures* (pp. 199-238). Hoboken, NJ: John Wiley & Sons.
- [41] Boukerche, A., & Turgut, D. (2007). Secure time synchronization protocols for wireless sensor networks. *Wireless Communications, IEEE, 14*(5), 64-69.
- [42] Display Elektronik GmbH (2003). *DEM 20486 SYH-LY datasheet*, Retrieved from <http://www.e-lab.de/diverse/DEM20486.pdf>
- [43] Atmel (2008). *AT89C4051 microcontroller datasheet*, Retrieved from <http://www.atmel.com/Images/doc1001.pdf>
- [44] Copera, J. J. C. (2004). *Inside the Nickel metal hydride battery*, Retrieved from [http://www.cobasys.com/pdf/tutorial/inside\\_nimh\\_battery\\_technology.pdf](http://www.cobasys.com/pdf/tutorial/inside_nimh_battery_technology.pdf)
- [45] Huffman, D. A. (1952). A method for the construction of minimum redundancy codes. *Proc. IRE, 40*(9), 1098-1101.
- [46] Pearlman, W. A., & Said, A. (2011). *Digital Signal Compression: principles and practice*. Cambridge University Press.
- [47] Sayood, K. (2000). *Introduction to data compression* (2<sup>nd</sup> ed.). San Francisco: Morgan Kaufmann Publishers.
- [48] Srisooksai, T., Keamarungsi, K., Lamsrichan, P., & Araki, K. (2012). Practical data compression in wireless sensor networks: A survey. *Journal of Network and Computer Applications, 35*(1), 37-59.
- [49] O'Neal Jr, J. B. (1976). Differential pulse-code modulation (PCM) with entropy coding. *Information Theory, IEEE Transactions on, 22*(2), 169-174.



- [50] Shahbahrami, A., Bahrampour, R., Rostami, M. S., & Mobarhan, M. A. (2011). Evaluation of Huffman and arithmetic algorithms for multimedia compression standards. *arXiv preprint arXiv:1109.0216*.
- [51] Youssef, A. S. (1998). *Parallel algorithms for entropy-coding techniques*. US Department of Commerce, Technology Administration, National Institute of Standards and Technology.
- [52] Storer, J. A. (1988). *Data compression: methods and theory*. New York, NY: Computer Science Press, Inc.
- [53] Salomon, D., & Motta, G. (2010). *Handbook of data compression* (4<sup>th</sup> ed.). London: Springer Verlag.
- [54] Vaseghi, S. V. (2007) *Multimedia signal processing: Theory and applications in speech, music and communications*. Chichester: John Wiley & Sons Ltd.
- [55] Kumar, A., & Bar-Shalom, Y. (1993). Time-domain analysis of cross correlation for time delay estimation with an autocorrelated signal. *Signal Processing, IEEE Transactions on*, 41(4), 1664-1668.
- [56] Kozick, R. J., & Sadler, B. M. (2004). Source localization with distributed sensor arrays and partial spatial coherence. *Signal Processing, IEEE Transactions on*, 52(3), 601-616.
- [57] Bjorklund, S., & Ljung, L. (2003, December). A review of time-delay estimation techniques. In *Decision and Control, 2003. Proceedings. 42<sup>nd</sup> IEEE Conference on* (Vol. 3, pp. 2502-2507). IEEE.
- [58] Terras, A. (1999). *Fourier analysis on finite groups and applications*. Cambridge University Press.
- [59] Strang, G. (1999). The discrete cosine transform. *SIAM review*, 41(1), 135-147.
- [60] Birvinskas, D., Jusas, V., Martišius, I., & Damaševičius, R. (2013). Data compression of EEG signals for artificial neural network classification. *Information Technology and Control*, 42(3), 238-241.
- [61] Hui, Y., & Pagiatakis, S. (2004). Least squares spectral analysis and its application to superconducting gravimeter data analysis. *Geo-Spatial Information Science*, 7(4), 279-283.
- [62] Puryear, C. I., Portniaguine, O. N., Cobos, C. M., & Castagna, J. P. (2012). Constrained least-squares spectral analysis: Application to seismic data.

*Geophysics*, 77(5), V143-V167.

- [63] Kirner, R., Grössing, M., & Puschner, P. (2006). Comparing WCET and resource demands of trigonometric functions implemented as iterative calculations vs. table-lookup. In *OASICS-OpenAccess Series in Informatics* (Vol. 4). Schloss Dagstuhl-Leibniz-Zentrum für Informatik.
- [64] Sokolove, P. G., & Bushell, W. N. (1978). The chi square periodogram: its utility for analysis of circadian rhythms. *Journal of Theoretical Biology*, 72(1), 131-160.
- [65] Zielinski, T., Moore, A. M., Troup, E., Halliday, K. J., & Millar, A. J. (2014). Strengths and limitations of period estimation methods for circadian data. *PLoS one*, 9(5), e96462.
- [66] Goel, I., Agarwal, S., & Prasad, R. (2012, March). A Linear Representation of Huffman Tree. In *Proceedings of the International MultiConference of Engineers and Computer Scientists* (Vol. 1). IAENG.
- [67] Texas Instruments (2010). Design Note DN509: *Data Whitening and Random TX Mode*. Dallas, TX: Christiansen, G.
- [68] Guoliang, W., Rengui, L., Chunbo, Z., & Chan, C. C. (2008, September). State of charge estimation for NiMH battery based on electromotive force method. In *Vehicle Power and Propulsion Conference, 2008*. (pp. 1-5). IEEE.
- [69] Zhu, W. H., Zhu, Y., Davis, Z., & Tatarchuk, B. J. (2013). Energy efficiency and capacity retention of Ni-MH batteries for storage applications. *Applied Energy*, 106, 307-313.
- [70] Сысоева, С. (2006). Автомобильные акселерометры. Часть 5. Перспективная элементная база поверхностных кремниевых емкостных MEMS акселерометров. *Компоненты и технологии*, (4), 28-39.
- [71] VTI Technologies (2005). *Cross-Axis Compensation*, Retrieved from [http://www.murataMEMS.fi/sites/default/files/documents/an32\\_cross-axis\\_compensation.pdf](http://www.murataMEMS.fi/sites/default/files/documents/an32_cross-axis_compensation.pdf)
- [72] Sill, R. D., & Seller, E. J. (2006, November). Accelerometer Transverse Sensitivity Measurement Using Planar Orbital Motion. In *77<sup>th</sup> Shock and Vibration Symposium*.
- [73] PCB Piezotronics (2000). Technical Note A-R 69: *Automated Testing of Accelerometer Transverse Sensitivity*. Depew, NY: Dosch, J. J., & Lally, D. M.
- [74] Veldman, C. S. (2012, September). Implementation of an accelerometer

- transverse sensitivity measurement system. In *Proceedings of test and measurement conference*. National Laboratory Association, South Africa.
- [75] Rahamatkar, S., Agarwal, A., & Kumar, N. (2010). Analysis and comparative study of clock synchronization schemes in wireless sensor networks. *International Journal on Computer Science and Engineering*, 2(3), 536-541.
- [76] Sundararaman, B., Buy, U., & Kshemkalyani, A. D. (2005). Clock synchronization for wireless sensor networks: a survey. *Ad Hoc Networks*, 3(3), 281-323.
- [77] Wu, Y. C., Chaudhari, Q., & Serpedin, E. (2011). Clock synchronization of wireless sensor networks. *Signal Processing Magazine, IEEE*, 28(1), 124-138.
- [78] Khediri, S., Nasri, N., Samet, M., Wei, A., & Kachouri, A. (2012). Analysis study of time synchronization protocols in wireless sensor networks. *International Journal of Distributed and Parallel Systems*, 3(3), 155-165.
- [79] Faizulkhakov, Y. R. (2007). Time synchronization methods for wireless sensor networks: A survey. *Programming and Computer Software*, 33(4), 214-226.
- [80] Elson, J., & Römer, K. (2003). Wireless sensor networks: A new regime for time synchronization. *ACM SIGCOMM Computer Communication Review*, 33(1), 149-154.
- [81] Elson, J., Girod, L., & Estrin, D. (2002). Fine-grained network time synchronization using reference broadcasts. *ACM SIGOPS Operating Systems Review*, 36(SI), 147-163.
- [82] Zhang, F., & Deng, G. Y. (2005, September). Probabilistic time synchronization in wireless sensor networks. In *Proceedings of 2005. International Conference on Wireless Communications, Networking and Mobile Computing* (Vol. 2, pp. 980-984). IEEE.
- [83] Kim, R., Nagayama, T., Jo, H., & Spencer Jr, B. F. (2012, April). Preliminary study of low-cost GPS receivers for time synchronization of wireless sensors. In *SPIE Smart Structures and Materials + Nondestructive Evaluation and Health Monitoring* (pp. 83451A). International Society for Optics and Photonics.
- [84] Cristian, F. (1989). Probabilistic clock synchronization. *Distributed computing*, 3(3), 146-158.
- [85] Gusella, R., & Zatti, S. (1989). The accuracy of the clock synchronization

achieved by TEMPO in Berkeley UNIX 4.3 BSD. *Software Engineering, IEEE Transactions on*, 15(7), 847-853.

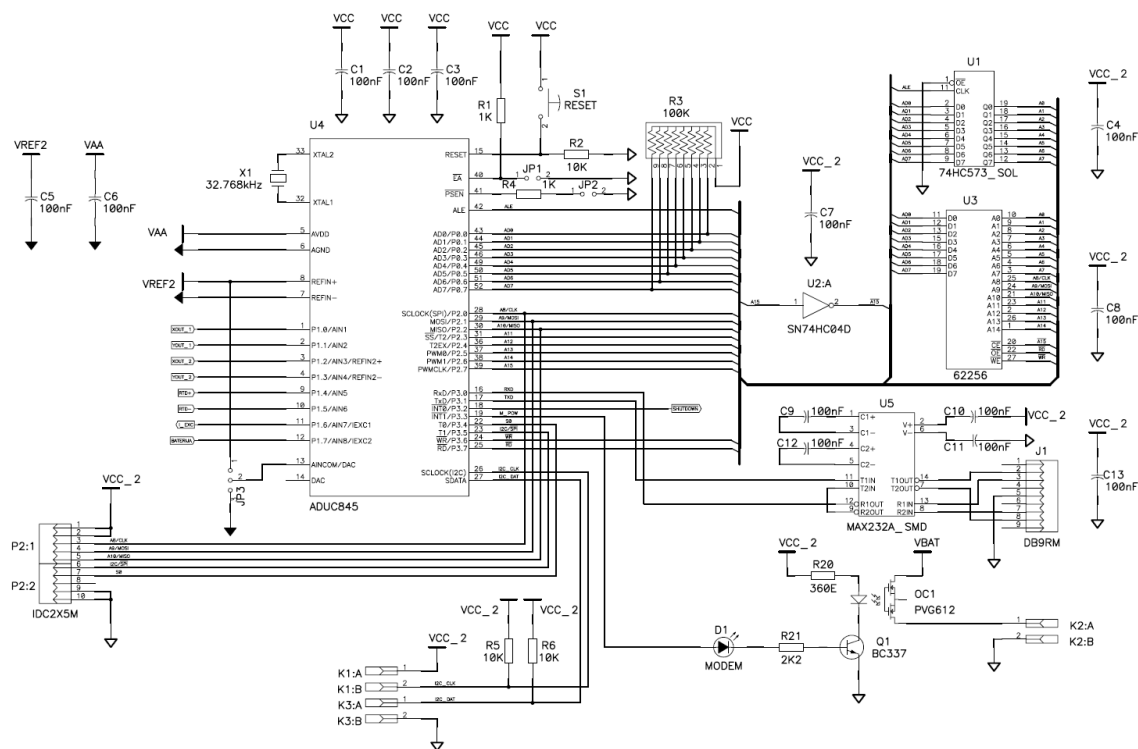
- [86] Mills, D. L. (1991). Internet time synchronization: the network time protocol. *Communications, IEEE Transactions on*, 39(10), 1482-1493.
- [87] Li, Z., Zhong, Z. L., Zhu, W. C., & Qin, B. Y. (2013). A Hardware Time Stamping Method for PTP Messages Based on Linux system. *Telkonnika - Indonesian Journal of Electrical Engineering*, 11(9), 5105-5111.
- [88] Exel, R., Gaderer, G., & Kerö, N. (2010, November). Physical Layer Ethernet Clock Synchronization. In *42<sup>nd</sup> Annual Precise Time and Time Interval (PTTI) Meeting* (pp. 77-87). ION.
- [89] Römer, K. (2001, October). Time synchronization in ad hoc networks. In *Proceedings of the 2<sup>nd</sup> ACM international symposium on Mobile ad hoc networking & computing* (pp. 173-182). ACM.
- [90] Sallai, J., Kusý, B., Lédeczi, Á., & Dutta, P. (2006). On the scalability of routing integrated time synchronization. In *Wireless Sensor Networks* (pp. 115-131). Springer Berlin Heidelberg.
- [91] Van Greunen, J., & Rabaey, J. (2003, September). Lightweight time synchronization for sensor networks. In *Proceedings of the 2<sup>nd</sup> ACM international conference on Wireless sensor networks and applications* (pp. 11-19). ACM.
- [92] Ganeriwal, S., Kumar, R., & Srivastava, M. B. (2003, November). Timing-sync protocol for sensor networks. In *Proceedings of the 1<sup>st</sup> international conference on Embedded networked sensor systems* (pp. 138-149). ACM.
- [93] Yoon, S., Veerarittiphan, C., & Sichitiu, M. L. (2007). Tiny-sync: Tight time synchronization for wireless sensor networks. *ACM Transactions on Sensor Networks (TOSN)*, 3(2), 8.
- [94] 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. In *Mobile Ad-Hoc and Sensor Networks* (pp. 536-547). Springer Berlin Heidelberg.
- [95] Dai, H., & Han, R. (2004). TSync: a lightweight bidirectional time synchronization service for wireless sensor networks. *ACM SIGMOBILE Mobile Computing and Communications Review*, 8(1), 125-139.

- [96] Meier, L. (2005, March). Interval-based Clock Synchronization for Ad-Hoc Sensor Networks. 4. *GI/ITG KuVS Fachgespräch „Drahtlose Sensornetze“*, 87.
- [97] Li, Q., & Rus, D. (2006). Global clock synchronization in sensor networks. *Computers, IEEE Transactions on*, 55(2), 214-226.
- [98] Gheorghe, L., Rughinis, R., & Tapus, N. (2010, March). Fault-tolerant flooding time synchronization protocol for wireless sensor networks. In *Networking and Services (ICNS), 6<sup>th</sup> International Conference on* (pp. 143-149). IEEE.
- [99] Lenzen, C., Sommer, P., & Wattenhofer, R. (2009, November). Optimal clock synchronization in networks. In *Proceedings of the 7<sup>th</sup> ACM Conference on Embedded Networked Sensor Systems* (pp. 225-238). ACM.
- [100] Su, W., & Akyildiz, I. F. (2005). Time-diffusion synchronization protocol for wireless sensor networks. *Networking, IEEE/ACM Transactions on*, 13(2), 384-397.
- [101] Sommer, P., & Wattenhofer, R. (2009, April). Gradient clock synchronization in wireless sensor networks. In *Proceedings of the 2009 International Conference on Information Processing in Sensor Networks* (pp. 37-48). IEEE Computer Society.
- [102] Maggs, M. K., O'Keefe, S. G., & Thiel, D. V. (2012). Consensus clock synchronization for wireless sensor networks. *Sensors Journal, IEEE*, 12(6), 2269-2277.
- [103] Riley, W. J. (2008). NIST special publication 1065. *Handbook of frequency stability analysis*. US Department of Commerce, National Institute of Standards and Technology.
- [104] Networked and Embedded Systems Lab, University of California (2009). Report. *On the interaction of clocks and power in embedded sensor nodes*. Los Angeles, CA: Schmid, T., Shea, R., Friedman, J., Charbiwala, Z., Srivastava, M. B., & Cho, Y. H.
- [105] Prasad, V., & Sharma, D. C. (2012). A Review of Phase Locked Loop. *International Journal of Emerging Technology and Advanced Engineering*, 2(6), 98-104.
- [106] US Army Communications-Electronics Research, Development & Engineering Center (2004). Report SLCET-TR-88-1, Rev. 8.5.2.2. *Quartz crystal resonators and oscillators - A tutorial for frequency and timing applications*. Fort

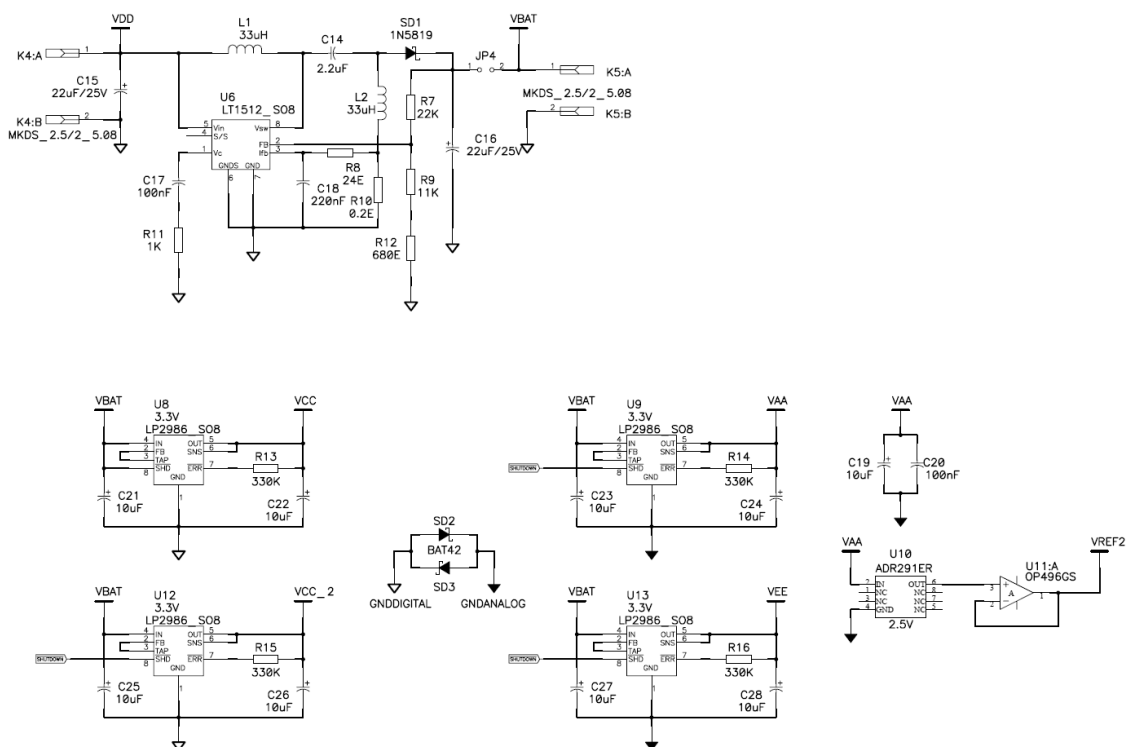
Monmouth, NJ: Vig, J. R.

- [107] Spassov, L. (1992). Piezoelectric quartz resonators as highly sensitive temperature sensors. *Sensors and Actuators A: Physical*, 30(1), 67-72.
- [108] Carleton University (2008). Report SCE-08-12. *Frequency accuracy & stability dependencies of crystal oscillators*. Ottawa, Canada: Zhou, H., Nicholls, C., Kunz, T., & Schwartz, H.
- [109] Nielsen, J. C., & Igeland, A. (1995). Vertical dynamic interaction between train and track influence of wheel and track imperfections. *Journal of Sound and Vibration*, 187(5), 825-839.
- [110] Cheng, Y., Chen, D., & Nogata, F. (1994). Fatigue behaviour of a rail steel under low and high loading rates. *Fatigue & Fracture of Engineering Materials & Structures*, 17(1), 113-118.
- [111] Barke, D., & Chiu, W. K. (2005). Structural health monitoring in the railway industry: a review. *Structural Health Monitoring*, 4(1), 81-93.
- [112] Thompson, D. (2008). *Railway noise and vibration: mechanisms, modelling and means of control*. Elsevier Oxford.
- [113] Fields, J. M., & Walker, J. G. (1982). Comparing the relationships between noise level and annoyance in different surveys: A railway noise vs. aircraft and road traffic comparison. *Journal of Sound and Vibration*, 81(1), 51-80.
- [114] Webster, J. G., & Eren, H. (Eds.). (2014). *Measurement, Instrumentation, and Sensors Handbook: Spatial, Mechanical, Thermal, and Radiation Measurement* (Vol. 1). CRC Press.
- [115] Pan, B., Qian, K., Xie, H., & Asundi, A. (2009). Two-dimensional digital image correlation for in-plane displacement and strain measurement: a review. *Measurement science and technology*, 20(6), 062001.
- [116] Cox, S. J. (1995). Deflection of sleeper in ballast. *Vehicle System Dynamics*, 24(sup1), 146-153.
- [117] Gučević, J., Delčev, S., Ogrizović, V., Pejić, M., Popović, J., & Pejović, M. (2014, april). Geodetic Works during the Estimation of the Vertical Displacement of a Bridge under a Load Test. In *Proceedings of 6<sup>th</sup> International Conference on Engineering Surveying*. (pp. 237-242). STU.

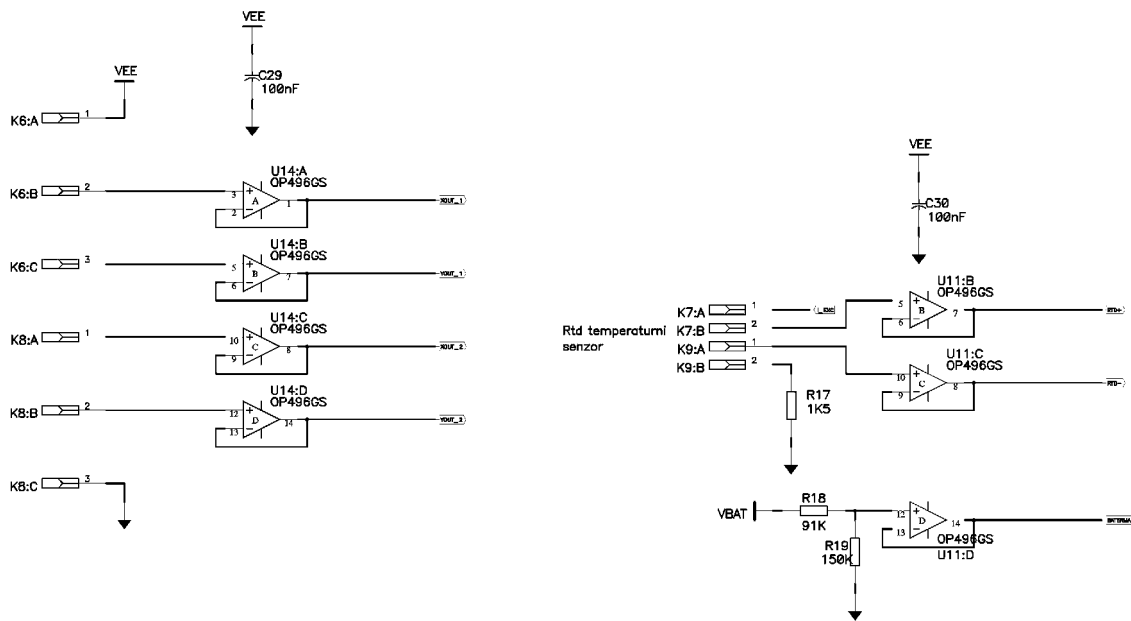
## 12. PRILOG 1: ELEKTRIČNE ŠEME



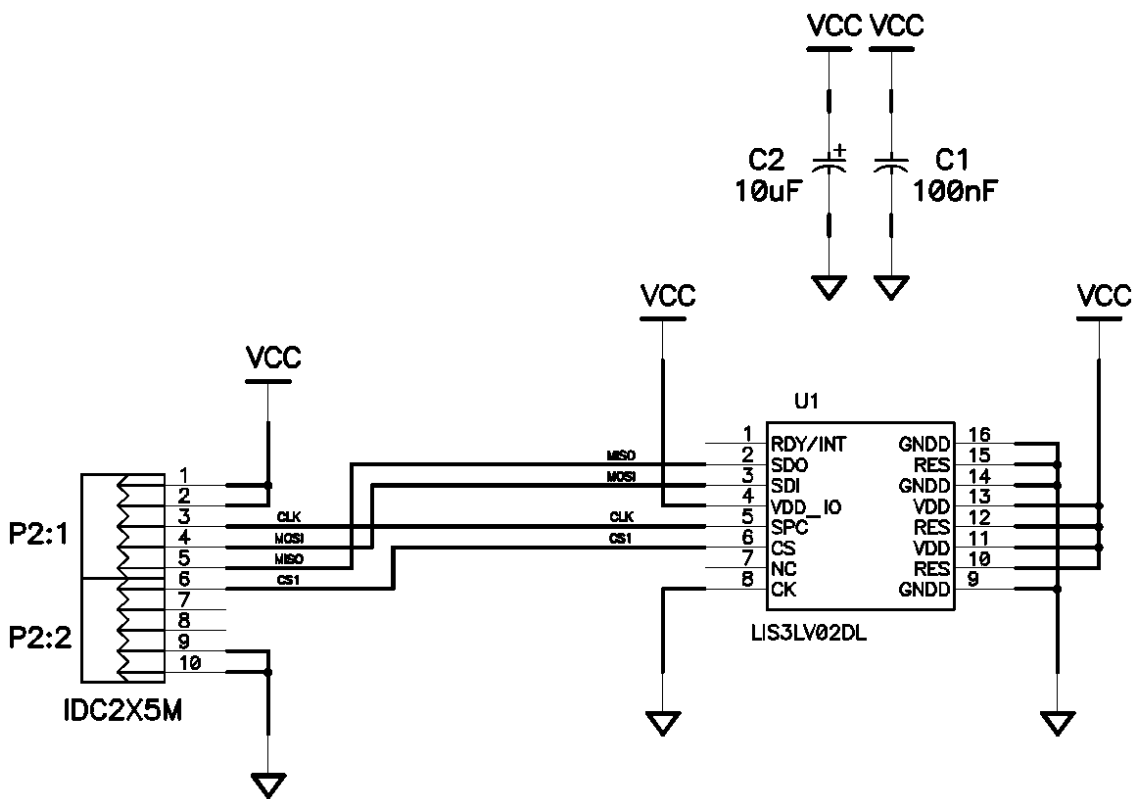
Slika 12.1 Procesor sa pratećom elektronikom



Slika 12.2 Stabilizatori i punjač

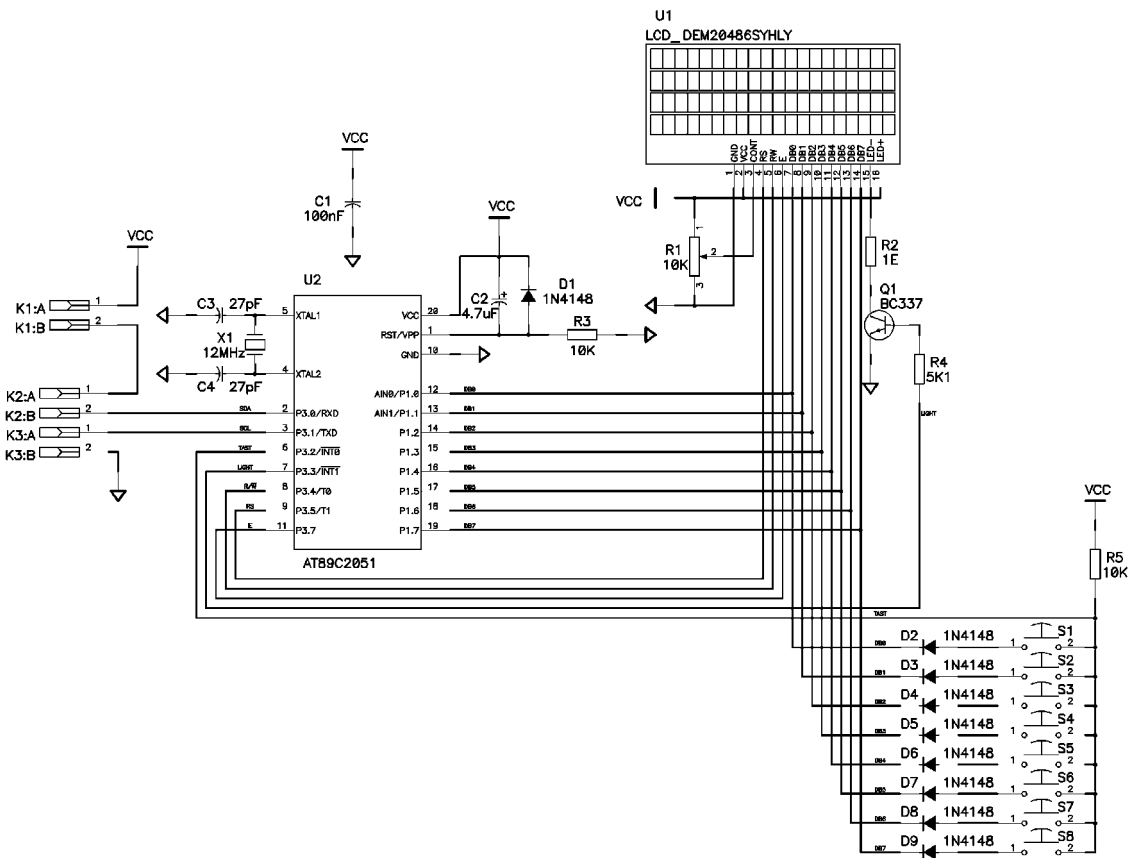


Slika 12.3 Analogni deo uređaja



Slika 12.4 Akcelerometar





Slika 12.5 Displej

## 13. PRILOG 2: IZVODI IZ PROGRAMA

### 13.1. Kompletan RTOS displeja

```
startup.asm
$DEBUG
NAME ?C_STARTUP
PUBLIC ?C_STARTUP, ytab, yu_char, cbuf, nbuf, smallpau, denpulse,
msgready, _transmb, IE, _DISPVAL, virtual, orient, x, y, _REDISP, CLS,
INITDISP, P3, timerbit, stopcon, paubig, yurom, home, cursor, pau3ms,
chksum, intrec, xorbyte, ichksumHi, I2C_DAT, I2C_CLK, KEYS, LIGHT, RW,
RS, DEN, broj_dec, mcnt, EX0, IE0, cleari2, blinkbit, inkey, tchang,
keyrelease, cursorbit, chksumhi, chksumlo, ichksum, p3func, mcntH,
mcntMH, mcntML, mcntL, _disprom, ichksumLo, inkeyupfunc, xx, _transm3,
upstatus, testcursor, tampon1, tampon2, display, cbuf1, firstsend
EXTRN CODE (?C_START)
CSEG AT 0
USING 0
?C_STARTUP:
    CLR A
    AJMP mreset
lab03H: SETB intrec          ; ext 0 interrupt, i2c
    RETI
    NOP
    NOP
    NOP
    NOP
    NOP
lab0Bh: MOV TL0,#050h       ; TIMER 0, ~ 3.33 KHz (regs: R0,R2,R3,R4)
    MOV TH0,#0FBh
    AJMP mytimer          ; TIMER 0 INTERRUPT
lab13h: RETI              ; EXTERNAL 1 INTERRUPT
smallpau:MOV R7,#0
sjp:    DJNZ R7,sjp
    RET
    NOP
    NOP
lab1Bh: RETI              ; TIMER 1 INTERRUPT (disabled)
denpulse:SETB DEN
    NOP
    CLR DEN
    RET
    NOP
lab23h: RETI              ; SERIAL INTERRUPT (disabled)
mreset: MOV IE,A
    MOV P3,#6
    MOV P1,A
    MOV PSW,A
    MOV R1,#16
chisty: MOV @R1,A
    INC R1
    CJNE R1,#MYSTACK,chisty
    MOV SP,R1
    ACALL paubig          ; 0.1 sec powerup
    ACALL INITDISP
    ACALL smallpau       ; 0.5 ms pause
    MOV TH0,#0F9h
```

```

        MOV IP,#1                ; setb TX0 (ext0 i2c reception)
        MOV TCON,#10101B        ; TR0 + falling edge exts + clr pend
        MOV IE,#131            ; setb EA, ET0, EX0
        LJMP ?C_START
; ##### KEYBOARD INTERRUPT #####
USING 1
mytimer: PUSH PSW                ; R0,R2,{R3},R4
        SETB RS0                ; RS1 remains low at all times
        MOV R3,A
        INC mcnt+3
        MOV A,mcnt+3
        JNZ lreti
        INC mcnt+2
        MOV A,mcnt+2
        JNZ lreti
        INC mcnt+1
        MOV A,mcnt+1
        JNZ lreti
        INC mcnt
lreti:  JB DEN,exila
        JB tchang,exila
        MOV A,mcnt+3
        ANL A,#15
        JNZ exila                ; read keys every 16th interrupt
        SETB timerbit
        MOV A,mcnt+3
        ANL A,#70h
        SWAP A
        INC A
        MOV R4,A                ; R4/R2 now holds value in range 1-8
        MOV R2,A
        ADD A,#chksumlo        ; EQU keyrelease-1
        MOV R0,A                ; address keyrelease[0-7]
        MOV A,#255
        CLR C
cabil:  RLC A
        DJNZ R4,cabil
        MOV P1,A
        SETB KEYS
        MOV C,KEYS
        CLR KEYS
        CPL C
        JNC nijeprit
        CLR A                    ; here if pressed, R0 range 1-8
cabily: RLC A
        DJNZ R2,cabily
        CJNE A,#4,nolight
        SETB LIGHT
        SJMP putkrl
nolight: ORL A,inkey
        MOV inkey,A
putkrl: MOV A,#8
        MOV @R0,A                ; keyrelease[key]
        MOV A,R3
        POP PSW                ; contains RS bits
        RETI
nijeprit:MOV A,@R0                ; here with JNC jump only
        JZ exila                ; already reset bit, just exit
        DEC A

```

```

        MOV @R0,A
        JNZ exila
        DEC A                ; here if not pressed, NC state
cabiln: RLC A
        DJNZ R2,cabiln
        CJNE A,#251,nodark
        CLR LIGHT
nodark:  ANL A,inkey
        MOV inkey,A
exila:   MOV A,R3
        POP PSW              ; contains RS bits
        RETI
; ##### I2C TRANSFERS #####
USING 0
intran:  MOV R6,#8
intra1:  CLR I2C_CLK
        RLC A                ; output set while CLK low
        MOV I2C_DAT,C
        NOP
        SETB I2C_CLK         ; clock out
        MOV R2,#15
trckcl0: MOV R4,#0
trckcl1: JB I2C_CLK,ltcont1  ; allow SCLK block by slave
        DJNZ R4,trckcl1     ; (prolong ~4000 cycles)
        DJNZ R2,trckcl0
        MOV R7,#254
        RET
ltcont1: DJNZ R6,intra1      ; loop 8 times for all data bits
        CLR I2C_CLK         ; wait ACK
        NOP
        SETB I2C_DAT
        MOV R4,#0
waitac1: JNB I2C_DAT,llcon1  ; got ACK
        DJNZ R4,waitac1
        MOV R7,#253
        RET
llcon1:  SETB I2C_CLK        ; pulse9 while ACK
        NOP
        CLR A
        MOV R7,A
        CLR I2C_CLK
        MOV R4,#48
waiten1: JB I2C_DAT,endla1   ; await slave releasing ACK (fast)
        DJNZ R4,waiten1
        MOV R7,#252
endla1:  RET
laextf:  MOV R7,#255
        AJMP laex
vspau:   MOV R7,#80
vspaul:  DJNZ R7,vspaul1
        RET
_transm3:CLR EA              ; Repsta: SDA then SCL go high
        JNB I2C_CLK,laextf
        JNB I2C_DAT,laextf  ; wrong state, attempt stop
        CLR I2C_DAT         ; [repeated] start condition

```

```

MOV A,R7
NOP
ACALL intran
MOV A,R7
JNZ laex
JBC firstsend,fsela
ACALL vspau
backt3: MOV A,R5
ACALL intran
MOV A,R7
JNZ laex
ACALL vspau
MOV A,R3
ACALL intran
MOV A,R7
JNZ laex
ACALL vspau
MOV A,xx
ACALL intran
MOV A,R7
JNZ laex ; SDAT is hi, SCLK is low
fsela: SETB I2C_CLK ; proper order of pulling up for RepSta later
MOV R2,#36 ; acts as vspau
lissu1: MOV R4,#0
lissue: JB I2C_CLK,eaex ; allow sufficient time for slave to process
DJNZ R4,lissue
DJNZ R2,lissu1
MOV R7,#251
SJMP laex
laex: ACALL stopcon
eaex: CLR msgready
CLR intrec
CLR IE0
ORL IE,#129
RET
_transmb:CLR EA ; high prior.
JNB I2C_CLK,laextf
JNB I2C_DAT,laextf
CLR I2C_DAT
MOV A,R7
NOP
ACALL intran
MOV A,R7
JNZ laex ; CLK line will be kept low transf.in progress
NOP
MOV R1,#25 ; counter of bytes to be received
MOV R0,#cbuf ; reception buffer pointer
recmsg: MOV R6,#8
waislac: SETB I2C_CLK
MOV R4,#96
waice0: MOV R2,#0
waicel: JB I2C_CLK,waicex
DJNZ R2,waicel

```

```

        DJNZ R4,waice0
lab154: MOV R7,#154           ; slave holds SCLK low on slav->mastr
        SJMP laex
waicex: MOV C,I2C_DAT
        CLR I2C_CLK
        RLC A
        NOP
        DJNZ R6,waislac
        CJNE R1,#1,nekra
        SETB I2C_DAT
        SJMP allkr
nekra:  CLR I2C_DAT
        NOP
allkr:  NOP
        NOP
        SETB I2C_CLK
        NOP
        NOP           ; state of data will depend on ack/nack
        NOP
        CLR I2C_CLK
        MOV @R0,A
        INC R0
        SETB I2C_DAT   ; recover from ack or prepare line
        DJNZ R1,recmsg
        NOP
        MOV R7,#0
        SETB I2C_CLK   ; prepare for RepSta
        SETB msgready
        CLR intrec
        CLR IE0
        ORL IE,#129
        RET
stopcon: CLR I2C_CLK
        CLR I2C_DAT
        ACALL nop6
        SETB I2C_CLK
        ACALL nop6
        SETB I2C_DAT
        RET
cleari2: CLR EA
        CLR msgready
        CLR intrec
        CLR IE0
        ORL IE,#129
        RET
; ##### DISPLAY SUBROUTINES #####
_dispval:MOV A,R4
        MOV C,ACC.7
        MOV F1,C       ; f1 saves sign
        JNC pozi
        CPL A          ; start of creation of two's complement
        MOV R4,A
        MOV A,R5

```

```

CPL A
MOV R5,A
MOV A,R6
CPL A
MOV R6,A
MOV A,R7
CPL A
INC A
MOV R7,A
JNZ pozi
INC R6
MOV A,R6
JNZ pozi
INC R5
MOV A,R5
JNZ pozi
INC R4
pozi: CLR A
      MOV R0,A
      MOV R1,A
      MOV R2,A
      MOV R3,A

repr4: MOV B,#8           ; 1st cycle: shift left highest byte
      MOV A,R4           ; result can reach no further than 2nd byte
      ADD A,R4
      MOV R4,A
      MOV A,R0           ; shift left result with carry in from R5 shift
      ADDC A,R0
      DA A               ; adjust to BCD after addition
      MOV R0,A
      MOV A,R1
      ADDC A,R1
      MOV R1,A           ; no DAA is performed because R1 is
in range 0-2
      DJNZ B,repr4
      MOV R4,#0          ; now clear R4 for decimal digit use
      MOV B,#8           ; 2nd highest byte
repr5: MOV A,R5
      ADD A,R5
      MOV R5,A
      ACALL inal
      DJNZ B,repr5
repr6: MOV B,#8           ; 3rd cycle: shift left lowest byte
      MOV A,R6
      ADD A,R6
      MOV R6,A
      ACALL inal
      DJNZ B,repr6
repr7: MOV B,#8
      MOV A,R7
      ADD A,R7
      MOV R7,A

```

```

ACALL inal
DJNZ B,repr7           ; at this point 10 digits are in R0-R4
CLR F0
MOV R5,AR0             ; now in R4-R3-R2-R1-R5 (h->l)
MOV R0,#nbuf          ; R0 will be used for addressing
MOV R6,#15
MOV R7,#48
MOV A,R5
ACALL maybedot
MOV A,R5
SWAP A
ACALL maybedot
MOV A,R1
ACALL maybedot
MOV A,R1
SWAP A
ACALL maybedot
MOV A,R2
ACALL maybedot
MOV A,R2
SWAP A
ACALL maybedot
MOV A,R3
ACALL maybedot
MOV A,R3
SWAP A
ACALL maybedot
MOV A,R4
ACALL maybedot
MOV A,R4
SWAP A
ACALL maybedot
MOV R5,#10
check0: DEC R0
MOV A,@R0
CJNE A,AR7,nije0
DJNZ R5,check0
SJMP nijetac
inal: MOV A,R0           ; internal dispval subroutine
      ADDC A,R0         ; carry out of input data shift is supplied
      DA A              ; shift left the output bits & add the new one
      MOV R0,A          ; adjust to BCD after each operation
      MOV A,R1
      ADDC A,R1
      DA A
      MOV R1,A
      MOV A,R2
      ADDC A,R2
      DA A
      MOV R2,A
      MOV A,R3
      ADDC A,R3
      DA A

```



```

        MOV R3,A
        MOV A,R4
        ADDC A,R4
        DA A
        MOV R4,A
exitlab: RET
nije0:  CJNE A,#'.',nijetac
        INC R0
nije0:  JNB F1,plusje
        INC R0
        MOV A,#'-'
        MOV @R0,A
plusje: MOV A,R0                ; R0 points to the last printable
        CLR C
        SUBB A,#nbuf
        MOV R7,A                ; R7=length of the message minus 1
        JB virtual,exitlab      ; in this case, only length is required
_redisp: ACALL takeyx           ; Add a,x produces NC state
        MOV R1,AR7              ; store to R1 now, R7 will be used
        JNB orient,narl        ; 0 = LEFT ALIGNED
        SUBB A,R1                ; NC
narl:   SETB tchang
        ORL A,#128               ; DDRAM-addressing flag
        MOV R4,A
        MOV A,R1
        ADD A,#nbuf
        MOV R0,A                ; left char on scr, right char in datbuf
        INC R1                   ; R1=total length of text (1-12)
        ACALL bfokwith
        MOV R6,#12               ; max length, rest to be blanked
        CLR RS
        CLR RW
        MOV A,R4                 ; set position (DDRAM addr)
        ACALL DEN37
        SETB RS
nobul:  MOV A,@R0
        ACALL DEN37
        DEC R0
        DEC R6
        DJNZ R1,nobul
        MOV A,R6
        JZ exitwith
nobus:  MOV A,#32
        ACALL DEN37
        DJNZ R6,nobus
exitwith:CLR tchang
        RET                       ; Ret.val.R7=0 kept from BFOK call
maybedot:ANL A,R6
        ADD A,R7
        MOV @R0,A
        INC R0
        JB F0,retdot
        MOV A,R0

```

```

        CLR C
        SUBB A,#nbuf
        CJNE A,broj_dec,retdot
        MOV A,#'.'
        MOV @R0,A
        INC R0
        SETB F0
retdot: RET
den37:  MOV P1,A
        ACALL denpulse
        MOV B,#156          ; more than 39-43 microsec
den37re: DJNZ B,den37re
        RET
waitBF: SETB tchang
        ACALL BFOK
        MOV A,R7
        JNZ waitBF
exitwi2: CLR tchang
        RET
BFOK:  CLR RS
        SETB RW
        SETB DEN
        MOV R7,#0
        MOV A,P1
        CLR DEN
        JNB ACC.7,retlas
        DEC R7
retlas: RET
BFOKWITH: CLR RS          ; strip return address on error (and clr tchang)
        SETB RW
        SETB DEN
        MOV R7,#0
        MOV A,P1
        CLR DEN
        JNB ACC.7,retlas
        DEC R7
        CLR tchang
        DEC SP
        DEC SP
        RET
CLS:   SETB tchang
        ACALL bfokwith
        CLR RS
        CLR RW
        MOV P1,#1
        NOP
        SETB DEN
        NOP
        CLR DEN
        ACALL pau3ms
exitwi4: CLR tchang
        RET
_disprom: SETB tchang

```

```

PUSH AR7
ACALL bfokwith
ACALL takeyx          ; don't care about ORIENT bit here
SETB ACC.7           ; DDRAM addressing
CLR RS
CLR RW
ACALL DEN37          ; transmit position
SETB RS              ; RW is still low
MOV R1,x             ; not to overwrite memory
MOV DPH,R6
POP DPL
CLR F0
nobur:  MOV A,R1
        CLR C
        SUBB A,#20
        JNC exitwi2      ; x>=20 so don't print anything
        INC R1
        JNB F0,uzmir
        MOV A,#32
        SJMP nijro29
uzmir:  CLR A
        MOVC A,@A+DPTR    ; max of 20 chars or ending with 80h
        INC DPTR
        CJNE A,#128,nijro28
        SJMP exitwi2
nijro28: CJNE A,#129,nijro29
        SETB F0
        MOV A,#32
nijro29: ACALL DEN37
        SJMP nobur        ; no need to djnz cause x control
display: SETB tchang
        ACALL bfokwith
        ACALL takeyx      ; ORIENT=don't care, R7 unused
        SETB ACC.7        ; DDRAM addressing
        CLR RS
        CLR RW
        ACALL DEN37      ; transmit position (R7 unused)
        SETB RS          ; RW is still low
        CLR mubil
        MOV R2,#20
        MOV R1,x         ; control memory position
        MOV R0,#cbuf+3
        MOV tampon1,#0
        CLR F0
nobu2:  MOV A,R1
        CLR C
        SUBB A,#20      ; x>=20 so don't print anything, R7=0 from BFOK
        JC nastafi
        SETB mubil
nastafi: INC R1
        JNB F0,uzmil
        MOV A,#32
        SJMP nije129

```

```

uzmi1:  MOV A,@R0                ; max of 20 chars or ending with 80h
        INC tampon1
        INC R0
        CJNE A,#128,nije128
        SJMP exitwi4
nije128: CJNE A,#129,nije129
        SETB F0
        MOV A,#32
nije129: JB mubil,nobu3
        ACALL DEN37
nobu3:  DJNZ R2,nobu2
        AJMP exitwi4
takeyx: MOV A,y
        MOV DPTR,#ytab
        MOVC A,@A+DPTR
        ADD A,x
        RET
yurom:  SETB tchang
        ACALL bfokwith
        CLR RS
        CLR RW
        MOV A,#64
        ACALL den37
        SETB RS
        CLR RW
        MOV DPTR,#yu_char
        MOV R7,#64
yulu:  CLR A
        MOVC A,@A+DPTR
        INC DPTR
        ACALL den37
        DJNZ R7,yulu
        SJMP inhome
HOME:  SETB tchang
        ACALL bfokwith
inhome: CLR RS
        CLR RW
        MOV A,#128
        ACALL den37
        MOV A,#2
        ACALL den37                ; R7=0
        ACALL pau3ms
exitwi3: CLR tchang
        RET
INITDISP: SETB tchang
        CLR RS
        CLR RW
        MOV P1,#00111100B        ; N=2_LINES  Disp=ON
        ACALL denpulse
        ACALL smallpau           ; about 500 microsec
        MOV P1,#00001100B        ; Disp=ON Curs=Off Blink=OFF
        ACALL denpulse
        ACALL smallpau

```

```

        MOV P1,#1                ; CLS
        ACALL denpulse
        ACALL pau3ms
        MOV P1,#2                ; home
        ACALL denpulse
        ACALL pau3ms
        MOV P1,#00000110B       ; increment on, shift off
        ACALL denpulse         ; cont into pauza
        ACALL smallpau
        SJMP exitwi3
paubig: MOV R6,#192             ; roughly 0.1 sec
p111:   ACALL smallpau
        DJNZ R6,p111
        RET
pau3ms: MOV R6,#6
        SJMP p111
nop6:   NOP
        NOP
        NOP
        NOP
        RET
p3func: MOV A,P3
        ANL A,#15
        MOV R7,A
        RET
inkeyupfunc:
        MOV A,inkey
        MOV C,upstatus
        MOV ACC.2,C            ; replace 4 (LIGHT) with upstatus
        MOV C,ACC.7
        MOV ACC.4,C           ; replace 16 (WAKE) with ENTER and
        CLR ACC.7             ; clear ENTER to avoid creating address
        MOV R7,A
        XRL A,#0FDh
        JNB ACC.7,lutxor
        CLR ACC.7
        XRL ACC,#32
lutxor: MOV xorbyte,A
lxretla: RET
testcursor:
        MOV A,cbuf+2
        ANL A,#192
        MOV R7,A
        MOV C,cursorbit
        MOV ACC.7,C
        MOV C,blinkbit
        MOV ACC.6,C
        XRL A,R7
        JNZ docurs
        MOV A,cbuf+2
        ANL A,#31
        JNZ lxretla           ; will disp text later

```

```

docurs:  MOV A,R7
         RLC A
         MOV cursorbit,C
         RLC A
         MOV blinkbit,C
cursor:  SETB tchang
         ACALL bfokwith
         ACALL takeyx
         CLR RS
         CLR RW
         ORL A,#128
         ACALL den37
         MOV A,#12
         MOV C,cursorbit
         MOV ACC.1,C
         MOV C,blinkbit
         MOV ACC.0,C
         ACALL den37
         AJMP exitwi3
ytab:   DB 0,40h,14h,54h
yu_char: DB 0Ah,04h,0Eh,10h,10h,11h,0Eh,00h ; ch ; 0
         DB 02h,04h,0Eh,10h,10h,11h,0Eh,00h ; cc ; 1
         DB 0Ah,04h,0Eh,10h,0Eh,01h,1Eh,00h ; sh ; 2
         DB 0Ah,04h,1Fh,02h,04h,08h,1Fh,00h ; zz ; 3
         DB 01h,03h,0Dh,13h,11h,11h,0Fh,00h ; dj ; 4
         DB 00h,00h,02h,05h,02h,05h,12h,00h ; 1/2; 5
         DB 00h,04h,02h,1Fh,02h,04h,00h,00h ; -> ; 6
         DB 00h,04h,08h,1Fh,08h,04h,00h,00h ; <- ; 7

I2C_CLK EQU P3.1
I2C_DAT EQU P3.2
KEYS     EQU P3.0
LIGHT    EQU P3.3
RW       EQU P3.4
RS       EQU P3.5
DEN      EQU P3.7
GASEG SEGMENT DATA
RSEG GASEG
inkey:   DS 1
broj_dec: DS 1
GBSEG SEGMENT DATA
RSEG GBSEG
MCNT:    DS 4
mcntL   EQU mcnt+3
mcntML  EQU mcnt+2
mcntMH  EQU mcnt+1
mcntH   EQU mcnt
GXSEG SEGMENT DATA
RSEG GXSEG
chksumhi: DS 1
chksumlo: DS 1
keyrelease: DS 8
cbuf:     DS 25 ; MARKER,COMMAND,FLAGS,DATA[20],CRC[2]
tampon1:  DS 2

```

```

nbuf:      DS 12
tampon2:   DS 2
xorbyte:   DS 1
xx:        DS 1
ichksum    EQU cbuf+23
ichksumHi  EQU ichksum
ichksumLo  EQU cbuf+24
chksum     EQU chksumhi
cbuf1      EQU cbuf+1
GDDAT segment DATA
RSEG GDDAT
x:         DS 1
y:         DS 1
MBISEG SEGMENT BIT
RSEG MBISEG
cursorbit: DBIT 1
blinkbit:  DBIT 1
tchang:    DBIT 1
virtual:    DBIT 1
orient:    DBIT 1
msgready:  DBIT 1
intrec:    DBIT 1
timerbit:  DBIT 1
upstatus:  DBIT 1
mubil:     DBIT 1
firstsend: DBIT 1
F1 EQU PSW.1
DSEG AT 107
MYSTACK: DS 21
END

```

#### main.c

```

#pragma REGPARMS
#pragma SMALL
#pragma DEBUG
#pragma ROM(COMPACT)
#pragma OPTIMIZE(5,SPEED)
#include <mmdef.h>
#include <sfr845.h>
unsigned char i,j,rezim,errcnt; unsigned short xp;

void medipau(unsigned short ix) { while (ix--) {smallpau();}}

void keysproc() {
    EX0=0; IE0=0; intrec=0; msgready=0; i=inkeyupfunc();
    j=transm3(0xAA,i,xorbyte); if (j!=0) {
        errcnt++; if (errcnt==3) {cleari2();} smallpau();}
    else {errcnt=0;} cbuf[3]=128; }

void dispproc() {bit ordershut;
    EX0=0; IE0=0; intrec=0; msgready=0; ordershut=0; upstatus=0;
    j=transmb(0xAB); if (j!=0) {
        errcnt++; if (errcnt==3) {cleari2();} smallpau();} else {errcnt=0;}
    msgready=0; IE0=0; EX0=1;
    if ((cbuf[0]==0)&&(cbuf[1]==15)) {upstatus=0; goto lr8;}
    if (cbuf[0]!=0xB1) {lr7:upstatus=1; lr8:xx=0xF8; return;}
}

```

```

if (((cbuf[1]<128)&&(cbuf[1]!=15)) {goto lr7;}
upstatus=crcerror(); if (upstatus) {goto lr8;}
if (cbuf[0]==15) {upstatus=0; goto laex;}
y=cbuf[1]&3; x=(cbuf[1]&127)>>2; testcursor();
if ((cbuf[2]&32)==32) {if ((cbuf[2]&31)!=0) {goto lr7;} cls();
goto laex;}
ordershut=(((cbuf[2]&31)==4)&&(cbuf[3]==0x82)&&(cbuf[4]==0x83)&&
(cbuf[5]==0x85)&&(cbuf[6]==0x80));
if ((cbuf[2]&31)!=0) {display(); upstatus=((cbuf[2]&31)!=tampon1);}
if ((ordershut)&&(!upstatus)) {cbuf[3]=128; inkey=0; errcnt=127;
return;}
laex:if (upstatus) {xx=248;} else {xx=cbuf[23]+cbuf[24];
if ((xx==248)|| (xx==0xAA)) {xx--;} if (xx==0xAB) {xx++;} }

void main() {
yurom(); 100:orient=0; virtual=0; broj_dec=0; xx=0xF8; rezim=0;
cbuf[0]=244; cbuf[1]=244; cbuf[2]=244; cbuf[3]=244; 18:xp=1;
goto 19;
10:IE=0; cursorbit=0; blinkbit=0; cursor();
if (rezim!=1) {medipau(99);}
19:IE=0; timerbit=0; upstatus=0; msgready=0; intrec=0;
cbuf[3]=128; errcnt=0; IE0=0; IE=0x83;
if (rezim==1) {goto 11;} cls();
x=0; y=0; disprom(" device offline\x80");
y=2; disprom(" M.Malovi\x01 software\x80");
y++; disprom(" (c) GRF Beograd\x80");
11:if (rezim!=2) {
if (xp==0) {if (rezim==1) {goto 100;}}
cbuf[3]=128; upstatus=0; errcnt=0;
if (intrec) {rezim=2; timerbit=0; intrec=0; goto 12;}
if ((inkey&AWAKE)!=0)
{cls(); while (inkey!=0) {} firstsend=1; keyproc();
rezim=1; goto 18;}
xp++; goto 11; }
12:while (!timerbit) {} timerbit=0; while (!timerbit) {}
keyproc(); if (errcnt>6) {rezim=0; goto 10;}
while (!timerbit) {} timerbit=0; while (!timerbit) {}
dispproc(); if (errcnt>6) {rezim=0; goto 10;}
goto 11;
}

```

#### mndef.h

```

#pragma REGPARMS
#pragma SMALL
#pragma DEBUG
#pragma ROM(COMPACT)
#pragma OPTIMIZE(5,SIZE)
#define GORE 8
#define DOLE 32
#define LEVO 2
#define DESNO 1
#define CLEAR 128
#define ENTER 64
#define AWAKE 16
extern unsigned char dispval(long x); extern data unsigned long mcnt;
extern data unsigned char cbuf[25], x, y, broj_dec, inkey, screen[12],

```



```

ichksumLo, tampon1, ichksumHi, cbufl[22], xx, chksumHi, chksumLo,
mcntH, mcntMH, mcntML, mcntL, xorbyte;
extern bit orient, virtual, cursorbit, blinkbit, LIGHT, msgready,
intrec, EX0, IE0, timerbit, upstatus, firstsend;
extern data unsigned short ichksum, chksum;
extern unsigned char display(); extern void cleari2();
extern unsigned char disprom(unsigned char code* bx);
extern unsigned char home(); extern unsigned char yurom();
extern unsigned char cls(); extern unsigned char cursor();
extern void redisp(unsigned char prevlen); extern void pau3ms();
extern void initrl(); extern void stopcon(); extern unsigned char
inkeyupfunc(); extern unsigned char transmb(unsigned char x);
extern void testcursor(); extern void paubig(); extern void
smallpau(); extern unsigned char transm3(unsigned char x,y,z);
code char xlatCRCHi[] = {
0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0,
0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41,
0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0,
0x80, 0x41, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40,
0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1,
0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41,
0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1,
0x81, 0x40, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41,
0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0,
0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81, 0x40,
0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1,
0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40,
0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0,
0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41,
0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0,
0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41,
0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1,
0x81, 0x40, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41,
0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0,
0x80, 0x41, 0x00, 0xC1, 0x81, 0x40 };
code char xlatCRCLo[] = {
0x00, 0xC0, 0xC1, 0x01, 0xC3, 0x03, 0x02, 0xC2, 0xC6, 0x06,
0x07, 0xC7, 0x05, 0xC5, 0xC4, 0x04, 0xCC, 0x0C, 0x0D, 0xCD,
0x0F, 0xCF, 0xCE, 0x0E, 0x0A, 0xCA, 0xCB, 0x0B, 0xC9, 0x09,
0x08, 0xC8, 0xD8, 0x18, 0x19, 0xD9, 0x1B, 0xDB, 0xDA, 0x1A,
0x1E, 0xDE, 0xDF, 0x1F, 0xDD, 0x1D, 0x1C, 0xDC, 0x14, 0xD4,
0xD5, 0x15, 0xD7, 0x17, 0x16, 0xD6, 0xD2, 0x12, 0x13, 0xD3,
0x11, 0xD1, 0xD0, 0x10, 0xF0, 0x30, 0x31, 0xF1, 0x33, 0xF3,
0xF2, 0x32, 0x36, 0xF6, 0xF7, 0x37, 0xF5, 0x35, 0x34, 0xF4,
0x3C, 0xFC, 0xFD, 0x3D, 0xFF, 0x3F, 0x3E, 0xFE, 0xFA, 0x3A,
0x3B, 0xFB, 0x39, 0xF9, 0xF8, 0x38, 0x28, 0xE8, 0xE9, 0x29,
0xEB, 0x2B, 0x2A, 0xEA, 0xEE, 0x2E, 0x2F, 0xEF, 0x2D, 0xED,
0xEC, 0x2C, 0xE4, 0x24, 0x25, 0xE5, 0x27, 0xE7, 0xE6, 0x26,
0x22, 0xE2, 0xE3, 0x23, 0xE1, 0x21, 0x20, 0xE0, 0xA0, 0x60,
0x61, 0xA1, 0x63, 0xA3, 0xA2, 0x62, 0x66, 0xA6, 0xA7, 0x67,
0xA5, 0x65, 0x64, 0xA4, 0x6C, 0xAC, 0xAD, 0x6D, 0xAF, 0x6F,
0x6E, 0xAE, 0xAA, 0x6A, 0x6B, 0xAB, 0x69, 0xA9, 0xA8, 0x68,
0x78, 0xB8, 0xB9, 0x79, 0xBB, 0x7B, 0x7A, 0xBA, 0xBE, 0x7E,

```

```

0x7F, 0xBF, 0x7D, 0xBD, 0xBC, 0x7C, 0xB4, 0x74, 0x75, 0xB5,
0x77, 0xB7, 0xB6, 0x76, 0x72, 0xB2, 0xB3, 0x73, 0xB1, 0x71,
0x70, 0xB0, 0x50, 0x90, 0x91, 0x51, 0x93, 0x53, 0x52, 0x92,
0x96, 0x56, 0x57, 0x97, 0x55, 0x95, 0x94, 0x54, 0x9C, 0x5C,
0x5D, 0x9D, 0x5F, 0x9F, 0x9E, 0x5E, 0x5A, 0x9A, 0x9B, 0x5B,
0x99, 0x59, 0x58, 0x98, 0x88, 0x48, 0x49, 0x89, 0x4B, 0x8B,
0x8A, 0x4A, 0x4E, 0x8E, 0x8F, 0x4F, 0x8D, 0x4D, 0x4C, 0x8C,
0x44, 0x84, 0x85, 0x45, 0x87, 0x47, 0x46, 0x86, 0x82, 0x42,
0x43, 0x83, 0x41, 0x81, 0x80, 0x40 };
bit crcerror() {data unsigned char cx,dptr; chksum=65535;
  for (dptr=1;dptr<23;dptr++) {
    cx=chksumHi^cbuf[dptr];
    chksumHi=chksumLo^xlatCRCHi[cx];
    chksumLo=xlatCRCLo[cx];}
  return (chksum!=ichksum); }

```

## 13.2. Hafmanovo kodiranje na senzorskom uređaju

izvod iz asemblera:

```

_calcfreq:  MOV DPTR,#restek
            MOV R0,DPH
            MOV R1,DPL
            MOV DPCON,#4
            MOV A,R6
            MOVX @DPTR,A
            MOV A,R7
            MOVX @DPTR,A
            CLR EA
            MOV R4,SPH
            MOV R5,SP
            SETB EA
            ACALL _malcfreq
            MOV DPCON,#0
            RET
_malcfreq:  JB compressfail,malcerr2
            MOV A,R0
            CJNE A,#255,okmbphi
            MOV A,R1
            ADD A,#-240
            JNC okmbphi
malcerrex:  SETB compressfail
malcerr2:  MOV DPCON,#0
            CLR EA
            MOV SP,R5
            MOV SPH,R4
            SETB EA
            MOV R6,#0
            MOV R7,#1
            MOV DPTR,#errorkod
            MOV A,#103
            MOVX @DPTR,A
            RET
okmbphi:   MOV A,SPH
            CJNE A,#5,oksphi
            MOV A,#240
            CLR C
            SUBB A,SP

```

```

    JC malcerrex
oksphi:  MOV DPH,R0           ; MBP is BP-style pointer to restack area
        MOV DPL,R1           ; +0,1=parameter; +2,3=prvi sabirak;
        MOVX A,@DPTR        ; (drugi ide kroz registre)
        MOV R6,A
        MOVX A,@DPTR        ; taken param(ix) of this instance into
        MOV R7,A           ; r6/r7 but still remain in [mbp]
        MOV A,#1
        CLR C
        SUBB A,R6
        JC idimext         ; jump if param>511
        MOV A,R7           ; fetch dpcfreq[ix] word and 32767 -> & ret
        ADD A,R7
        MOV R7,A
        MOV A,R6
        ADDC A,R6
        MOV R6,A
        MOV A,#LOW(dpcfreq)
        ADD A,R7
        MOV DPL,A
        MOV A,#HIGH(dpcfreq)
        ADDC A,R6
        MOV DPH,A
        MOVX A,@DPTR
        ANL A,#127
        MOV R6,A
        MOVX A,@DPTR
        MOV R7,A
        MOV A,R1
        ADD A,#252
        MOV R1,A
        JC retlam
retlam:  DEC R0
        RET                 ; w/ result in r6/r7, and dec restack ptr
idimext: MOV A,R7
        ADD A,R7
        MOV R7,A
        MOV A,R6
        ADDC A,R6
        MOV R6,A
        MOV A,R7
        ADD A,R7
        MOV R7,A
        MOV A,R6
        ADDC A,R6
        MOV R6,A           ; param*4 byte offset for nodtabword[ix+ix]
        MOV A,#LOW(nodtabword)
        ADD A,R7
        MOV DPL,A
        MOV A,#HIGH(nodtabword)
        ADDC A,R6
        MOV DPH,A
        MOVX A,@DPTR        ; addressing nodtabword[ix<<1]
        ANL A,#127
        MOV R6,A
        MOVX A,@DPTR
        MOV R7,A           ; param for 1st recursive call is in r67 now
        MOV A,R1
        ADD A,#4

```

```

MOV R1,A
JNC nombpcarry
INC R0
nombpcarry: MOV DPH,R0
MOV DPL,R1      ; incremented pointer put var
MOV A,R6
MOVX @DPTR,A
MOV A,R7
MOVX @DPTR,A
ACALL _malcfreq ; call with incrementing restack
MOV DPH,R0      ; (will revert on exit)
MOV DPL,R1
INC DPTR
INC DPTR
MOV A,R6
MOVX @DPTR,A
MOV A,R7
MOVX @DPTR,A      ; save result in temp var (+2,+3)
MOV DPH,R0
MOV DPL,R1      ; already checked for higher than 511
MOVX A,@DPTR
MOV R6,A
MOVX A,@DPTR
ADD A,ACC
MOV R7,A
MOV A,R6
ADDC A,R6
MOV R6,A
MOV A,R7
ADD A,R7
MOV R7,A
MOV A,R6
ADDC A,R6
MOV R6,A
MOV A,#LOW(nodtabwordplus2)
ADD A,R7
MOV DPL,A
MOV A,#HIGH(nodtabwordplus2)
ADDC A,R6
MOV DPH,A
MOVX A,@DPTR
ANL A,#127
MOV R6,A
MOVX A,@DPTR
MOV R7,A      ; param for 2nd recursive call is r6/r7
MOV A,R1
ADD A,#4
MOV R1,A
JNC nombpkeri
INC R0
nombpkeri: MOV DPH,R0
MOV DPL,R1      ; incremented pointer put var
MOV A,R6
MOVX @DPTR,A
MOV A,R7
MOVX @DPTR,A
ACALL _malcfreq ; will return result in r6/r7
MOV DPH,R0
MOV DPL,R1

```

```

        INC DPTR
        INC DPTR
        MOVX A,@DPTR      ; address 1st addend
        MOV R2,A
        MOVX A,@DPTR
        ADD A,R7
        MOV R7,A
        MOV A,R2
        ADDC A,R6
        MOV R6,A          ; add to form result
        MOV A,R1          ; 2nd exit frame identical to 1st
        ADD A,#252
        MOV R1,A
        JC retlma
        DEC R0
retlma:  RET              ; result r6/r7, decrement restack ptr
_markfreq: MOV DPTR,#restek
        MOV R0,DPH
        MOV R1,DPL
        MOV DPCON,#4
        MOV A,R6
        MOVX @DPTR,A
        MOV A,R7
        MOVX @DPTR,A
        CLR EA
        MOV R4,SPH
        MOV R5,SP
        SETB EA
        ACALL _markmreq
        MOV DPCON,#0
        RET
_markmreq: JNB compressfail,labmok1
mretonly: MOV DPCON,#0
        CLR EA
        MOV SP,R5
        MOV SPH,R4
        SETB EA
        RET
labmok1: MOV A,R0
        CJNE A,#255,okmbpmi
        MOV A,R1
        ADD A,#-240
        JNC okmbpmi
err101:  MOV A,#101
malmerc: SETB compressfail
        MOV DPTR,#errorkod
        MOVX @DPTR,A
        SJMP mretonly
okmbpmi: MOV A,SPH
        CJNE A,#5,oksmhi
        MOV A,#240
        CLR C
        SUBB A,SP
        JC err101
oksmhi:  MOV A,tempbits
        ANL A,#192
        JZ oktempbi      ; maxdub can be up to 63
        MOV A,#102
        SJMP malmerc

```

```

oktempbi:  MOV DPH,R0                ; R01 is BP pointer to restack area
           MOV DPL,R1
           MOVX A,@DPTR
           MOV R6,A
           MOVX A,@DPTR
           MOV R7,A
           MOV A,#1
           CLR C
           SUBB A,R6
           JC mvece512
           MOV A,#LOW(dubina)
           ADD A,R7
           MOV DPL,A
           MOV A,#HIGH(dubina)
           ADDC A,R6
           MOV DPH,A                ; if ix<512, address dubina[ix]
           MOV A,tempbits
           MOVX @DPTR,A            ; dubina[ix]=tempbits
           SUBB A,maxdub           ; C must be clear (from dubina addressing)
           JC nomaxdu
           MOV maxdub,tempbits    ; if dubina>=maxdub then maxdub=dubina
nomaxdu:   MOV A,R7
           ADD A,R7
           MOV R7,A
           MOV A,R6
           ADDC A,R6
           MOV R6,A                ; IX*2
           MOV A,R7
           ADD A,R7
           MOV R7,A
           MOV A,R6
           ADDC A,R6
           MOV R6,A                ; IX*4
           MOV A,R7
           ADD A,R7
           MOV R7,A
           MOV A,R6
           ADDC A,R6
           MOV R6,A                ; IX*8 (SHL 3)
           DEC tempbits
           MOV A,tempbits
           RR A
           RR A
           RR A
           ANL A,#1Fh
           INC A
           MOV R2,A                ; will loop once for A=0, etc.
(inclusive index of last tempkod)
           MOV A,#LOW(kodiran)
           ADD A,R7
           MOV DPL,A
           MOV A,#HIGH(kodiran)
           ADDC A,R6
           MOV DPH,A                ; main now toggle
           MOV DPCON,#55h          ; toggle, all inc
           MOV DPTR,#tempkod       ; tempkod[0] addressed with shadow
tbgilo:   MOVX A,@DPTR            ; (kodiran[ix*8] with main)
           MOVX @DPTR,A
           DJNZ R2,tbgilo

```

```

        MOV DPCON,#4
        MOV A,R1
        ADD A,#254
        MOV R1,A
        JC retlmz
        DEC R0
retlmz:  RET                ; dec bp and exit
mvece512: MOV DPH,#HIGH(tempkod)
        MOV A,tempbits
        RR A
        RR A
        RR A
        ANL A,#1Fh
        ADD A,#LOW(tempkod)
        MOV DPL,A
        JNC nohiti
        INC DPH
nohiti:  MOV R2,DPH
        MOV R3,DPL
        MOVX A,@DPTR        ; address tempkod[tempbits>>3] and push using r23
        MOV B,A
        MOV A,tempbits
        ANL A,#7
        MOV DPTR,#antist2
        MOVC A,@A+DPTR
        ANL A,B
        MOV DPL,R3
        MOV DPH,R2
        MOVX @DPTR,A        ; and this tempkod with opposite mask
        INC tempbits
        MOV A,R1            ; prepare next instance call
        ADD A,#2
        MOV R1,A
        JNC nor0im
        INC R0
nor0im:  MOV A,R7            ; here, R67 is preserved from initial fetch
        ADD A,R7
        MOV R7,A
        MOV A,R6
        ADDC A,R6
        MOV R6,A            ; IX<<1
        MOV A,R7
        ADD A,R7
        MOV R7,A
        MOV A,R6
        ADDC A,R6
        MOV R6,A            ; another shift is needed to get byte
        MOV A,#LOW(nodtabword) ; address (word[ix<<1] equ byte[ix<<2])
        ADD A,R7
        MOV DPL,A
        MOV A,#HIGH(nodtabword)
        ADDC A,R6
        MOV DPH,A
        MOVX A,@DPTR        ; nodtabword[ix<<1]
        ANL A,#127            ; &32767
        MOV R6,A
        MOVX A,@DPTR
        XCH A,R6            ; can do, R01 is prepared
        MOV DPL,R1

```

```

MOV DPH,R0
MOVX @DPTR,A
MOV A,R6                ; low in this sequence
MOVX @DPTR,A
LCALL _markmreq
MOV DPH,#HIGH(tempkod)
MOV A,tempbits         ; repeat with OR and call[+1]
RR A
RR A
RR A
ANL A,#1Fh
ADD A,#LOW(tempkod)
MOV DPL,A
JNC notihi
INC DPH
notihi: MOV R2,DPH
MOV R3,DPL
MOVX A,@DPTR           ; address tempkod[tempbits>>3], push u/ r23
MOV B,A
MOV A,tempbits
ANL A,#7
MOV DPTR,#stepen2
MOVC A,@A+DPTR
ORL A,B
MOV DPL,R3
MOV DPH,R2
MOVX @DPTR,A          ; or this tempkod with mask
INC tempbits
MOV DPH,R0            ; here R67 needs to be refetched
MOV DPL,R1
MOVX A,@DPTR
MOV R6,A
MOVX A,@DPTR
MOV R7,A
MOV R0,DPH
MOV R1,DPL           ; in preparation for next instance
MOV A,R7
ADD A,R7
MOV R7,A
MOV A,R6
ADDC A,R6
MOV R6,A             ; ix<<1
MOV A,R7
ADD A,R7
MOV R7,A
MOV A,R6
ADDC A,R6
MOV R6,A             ; another shift is needed to get byte
MOV A,#LOW(nodtabwordplus2); addr. (word[ix<<1] equ byte[ix<<2])
ADD A,R7
MOV DPL,A
MOV A,#HIGH(nodtabwordplus2)
ADDC A,R6
MOV DPH,A
MOVX A,@DPTR         ; nodtabword[succ(ix<<1)]
ANL A,#127           ; &32767
MOV R6,A
MOVX A,@DPTR
XCH A,R6

```



```

        MOV DPL,R1
        MOV DPH,R0
        MOVX @DPTR,A
        MOV A,R6                ; this is actually low
        MOVX @DPTR,A
        LCALL _markmreq
        DEC tempbits
        MOV A,R1                ; exit block (BP)
        ADD A,#254
        MOV R1,A
        JC retlaf
        DEC R0
retlaf:  RET
_streambyte:MOV R6,#0
        MOV DPTR,#bitctr
        MOVX A,@DPTR
        MOV R5,A
        MOV R4,A
        MOV R1,A
        JZ noshiftsb
bitcsh:  CLR C
        RLC A
        XCH A,R6
        RLC A
        XCH A,R6
        DJNZ R5,bitcsh
        MOV R7,A
noshiftsb:MOV DPTR,#myptr
        MOVX A,@DPTR
        MOV R5,A
        INC DPTR
        MOVX A,@DPTR
        MOV DPL,A
        MOV R3,A
        MOV DPH,R5
        MOVX A,@DPTR
        XCH A,R4                ; R4=*myptr; A=bitctr; myptr=R53
        ANL A,#7
        MOV DPTR,#poww
        MOVC A,@A+DPTR         ; a=poww[bitctr and 7]
        ANL A,R4                ; and with *myptr
        ORL A,R7                ; or with lo
        MOV DPL,R3
        MOV DPH,R5
        MOVX @DPTR,A           ; returned *myptr to memory
        MOV R2,bits
        MOV A,R2
        ADD A,#-9
        JNC manjeod9
manjeod9:MOV R2,#8
        MOV A,R2
        ADD A,R1
        MOV DPTR,#bitctr
        MOVX @DPTR,A
        ANL A,#0F8h
        JZ retlaf                ; return if not bitctr>7
        MOVX A,@DPTR
        ANL A,#7

```

```

MOVX @DPTR,A
MOV DPTR,#myptr
MOVX A,@DPTR
MOV R4,A
INC DPTR
MOVX A,@DPTR
INC A
MOV R5,A
MOVX @DPTR,A
JNZ never4
INC R4
MOV A,R4
MOV DPTR,#myptr
MOVX @DPTR,A
never4:  MOV DPH,R4
        MOV DPL,R5
        MOV A,R6
        MOVX @DPTR,A
        RET
_streamarray:
        MOV A,R7
        CLR C
        RLC A
        XCH A,R6
        RLC A
        XCH A,R6
        CLR C
        RLC A
        XCH A,R6
        RLC A
        XCH A,R6
        CLR C
        RLC A
        XCH A,R6
        RLC A
        XCH A,R6
        ADD A,#LOW(kodiran)
        MOV DPL,A
        MOV A,#HIGH(kodiran)
        ADDC A,R6
        MOV DPH,A
sar11:  MOVX A,@DPTR
        MOV R7,A
        MOV R0,DPH
        MOV B,DPL
        ACALL _streambyte
        MOV DPL,B
        MOV DPH,R0
        MOV A,bits
        ADD A,#-9
        JNC retlastr
        INC A
        MOV bits,A
        INC DPTR
        SJMP sar11
retlastr:  RET
uvodstream:  MOV A,maxdub
            MOV DPTR,#xlatMDUB
            MOVC A,@A+DPTR

```

```

MOV R7,A
MOV k,A
MOV bits,#3
ACALL _streambyte
MOV R7,#0
MOV R6,#0
MOV R5,#0
LCALL _getacc
MOV bits,#8
MOV B,R6
ACALL _streambyte
MOV R7,B
MOV bits,#4
ACALL _streambyte
MOV R7,#0
MOV R6,#0
MOV R5,#1
LCALL _getacc
MOV bits,#8
MOV B,R6
ACALL _streambyte
MOV R7,B
MOV bits,#4
ACALL _streambyte
MOV R7,#0
MOV R6,#0
MOV R5,#2
LCALL _getacc
MOV bits,#8
MOV B,R6
ACALL _streambyte
MOV R7,B
MOV bits,#4
ACALL _streambyte
MOV R7,#0
MOV R6,#0
ix512loop: MOV curpage,R6
ix513loop: MOV curpos,R7
MOV A,R7
ADD A,R7
MOV R7,A
MOV A,R6
ADDC A,R6
MOV R6,A
MOV A,#LOW(dpcfreq)
ADD A,R7
MOV DPL,A
MOV A,#HIGH(dpcfreq)
ADDC A,R6
MOV DPH,A
MOVX A,@DPTR
MOV R4,A
INC DPTR
MOVX A,@DPTR
MOV R5,A
ORL A,R4
JNZ ndpnul
MOV bits,#1
MOV R7,#0

```

```

                LCALL _streambyte
                SJMP ex512loop
ndpnul:        MOV bits,#1
                MOV R7,#255
                ACALL _streambyte
                MOV bits,k
                MOV A,#LOW(dubina)
                ADD A,curpos
                MOV DPL,A
                MOV A,#HIGH(dubina)
                ADDC A,curpage
                MOV DPH,A
                MOVX A,@DPTR
                PUSH ACC
                DEC A
                MOV R7,A
                ACALL _streambyte
                POP bits
                MOV R7,curpos
                MOV R6,curpage
                ACALL _streamarray
ex512loop:    MOV R7,curpos
                MOV R6,curpage
                INC R7
                MOV A,R7
                JNZ ix513loop
                MOV A,R6
                INC R6
                JZ ix512loop
                MOV i,#0
i32loop:     MOV R6,#0
                MOV R7,#1
ix32koloo:  MOV curpage,R6
ix32kiloo:  MOV curpos,R7
                MOV R5,i
                LCALL _getaccwith
                MOV axx,R7
                MOV x,R6
                CLR C
                MOV DPTR,#umptr+1
                MOVX A,@DPTR
                SUBB A,udy+1
                MOV DPTR,#umptr
                MOVX A,@DPTR
                SUBB A,udy
                JC oknot107
                MOV R7,#107
                RET
oknot107:   CLR dummyuse
                MOV A,R7
                DEC R7
                JZ lx1388
                INC R6
lx1388:     MOV A,R6
                JB ACC.7,outdlr           ; if R67 operand is negative
                JZ indlr                 ; if <256 then OK
                DEC A
                JNZ outdlr               ; if >512 then out, proceed if 256-511
                CJNE R7,#255,indlr      ; if low!=255 then OK

```

```

outdlr:    SJMP justdu
           MOV R7,#255
           MOV R6,#1           ; dl default 511
justdu:    SETB dummyuse
indlr:    MOV A,#LOW(dubina)
           ADD A,R7
           MOV DPL,A
           MOV A,#HIGH(dubina)
           ADDC A,R6
           MOV DPH,A           ; dptr=dubina[d1]
           MOVX A,@DPTR
           MOV bits,A
           ACALL _streamarray
           JNB dummyuse,ex32kloop
           MOV bits,#8
           MOV R7,axx
           ACALL _streambyte
           MOV bits,#4
           MOV R7,x
           ACALL _streambyte
ex32kloop: MOV R7,curpos
           MOV R6,curpage
           CJNE R7,#7Fh,nije3199
           CJNE R6,#0Ch,nije3199
           SJMP iouter
nije3199:  INC R7
           CJNE R7,#0,ix32kloop
           INC R6
           SJMP ix32kloop
iouter:   INC i
           MOV A,i
           CJNE A,#3,i32loop
           MOV R7,#0
           RET
_used:    MOV A,R6
           ADD A,#254
           JC ixveliko
           MOV A,R7
           ADD A,R7
           XCH A,R6
           RLC A
           XCH A,R6           ; [R6,A] holds ix*2
           ADD A,#LOW(dpcfreq)
           MOV DPL,A
           MOV A,#HIGH(dpcfreq)
           ADDC A,R6
           MOV DPH,A
           MOVX A,@DPTR
           JB ACC.7,retfals
           MOV R4,A
           INC DPTR
           MOVX A,@DPTR       ; fetched dpcfreq[ix] short
           ORL A,R4
           JZ retfals
           SETB C
           RET
retfals:  CLR C
           RET
ixveliko: MOV A,R7

```

```

        ADD A,R7
        XCH A,R6
        RLC A
        XCH A,R6           ; [R6,A] = 2*R67
        ADD A,ACC
        XCH A,R6
        RLC A
        XCH A,R6           ; [R6,A] = 4*ix
        ADD A,#LOW(nodtabbyte)
        MOV DPL,A
        MOV A,#HIGH(nodtabbyte)
        ADDC A,R6
        MOV DPH,A
        MOVX A,@DPTR
        MOV C,ACC.7
        CPL C
        RET
miniproc: MOV R7,#0
        MOV R6,#0
        MOV x,#255
        MOV axx,#127       ; mini initial is 32767 (axx-x)
mpoloo:  MOV curpage,R6
        MOV curpos,R7
        ACALL _used
        JNC lminext
        MOV R7,curpos
        MOV R6,curpage
        LCALL _calcfreq
        SETB C
        MOV A,x
        SUBB A,R7
        MOV A,axx
        SUBB A,R6
        JC lminext
        MOV x,R7
        MOV axx,R6         ; location of mini
        MOV im1,curpage
        MOV im1+1,curpos
        CJNE R7,#1,lminext
        CJNE R6,#0,lminext
        RET
lminext: MOV R7,curpos
        MOV R6,curpage
        INC R7
        CJNE R7,#0,mp2013
        INC R6
mp2013:  MOV DPTR,#(curdepth+1)
        MOVX A,@DPTR       ; hi(curdepth)
        SETB C
        SUBB A,R7
        MOV DPTR,#curdepth
        MOVX A,@DPTR
        SUBB A,R6
        JNC mpoloo
        RET
mini2proc: MOV R7,#0
        MOV R6,#0
        MOV mxdata+1,#255
        MOV mxdata,#127   ; mini2 initial is 32767

```

```

mp21oo:    MOV curpage,R6
           MOV curpos,R7
           ACALL _used
           JNC lm2next
           MOV A,im1
           CJNE A,curpage,doinblock
           MOV A,im1+1
           CJNE A,curpos,doinblock
lm2next:   MOV R7,curpos
           MOV R6,curpage
           INC R7
           CJNE R7,#0,mp2014
           INC R6
mp2014:   MOV DPTR,#(curdepth+1)
           MOVX A,@DPTR                ; hi(curdepth)
           SETB C
           SUBB A,R7
           MOV DPTR,#curdepth
           MOVX A,@DPTR
           SUBB A,R6
           JNC mp21oo
           RET
doinblock: SETB hx
           MOV R7,curpos
           MOV R6,curpage
           LCALL _calcfreq
           SETB C
           MOV A,mxdata+1
           SUBB A,R7
           MOV A,mxdata
           SUBB A,R6
           JC lm2next
           MOV mxdata+1,R7
           MOV mxdata,R6                ; mini2=pxdata
           MOV im2,curpage
           MOV im2+1,curpos
           MOV A,R7
           CJNE A,x,lm2next
           MOV A,R6
           CJNE A,axx,lm2next
           RET

```

izvod iz embedded C-a:

```

curdepth=512; im1=32767; im2=32767; pim1=32767; pim2=32767;
lwhilenode:
miniproc(); hx=0; if (compressfail) {goto cfr;}
mini2proc(); if (!hx) {goto lout3;}
if (im1hi<2) {dpcfreq[im1]=32768;} else
  {nodtabbyte[im1<<2]=128;}
if (im2hi<2) {dpcfreq[im2]=32768;} else
  {nodtabbyte[im2<<2]=128;}
nodtabword[curdepth<<1]=im1; pim1=im1;
nodtabwordplus2[curdepth<<1]=im2; pim2=im2;
curdepth++; if (curdepth>=1536) {errorkod=105; goto cfr;}
goto lwhilenode;
lout3:
if (((pim1&32767)==32767) || ((pim2&32767)==32767))

```

```

    {errorkod=106; goto cfr;}
if ((pim1+pim2)>=3072) {errorkod=106; goto cfr;}
bitctr=0; maxdub=2; umptr=&huftab;
tempkod[0]=0; tempkod[1]=0; tempkod[2]=0; tempkod[3]=0;
tempkod[4]=0; tempkod[5]=0; tempkod[6]=0; tempkod[7]=0;
tempbits=1; markfreq(pim1); if (compressfail) {goto cfr;}
tempkod[0]=1; tempkod[1]=0; tempkod[2]=0; tempkod[3]=0;
tempkod[4]=0; tempkod[5]=0; tempkod[6]=0; tempkod[7]=0;
tempbits=1; markfreq(pim2); if (compressfail) {goto cfr;}
if (errorkod=uvodstream()) {goto cfr;}
if (umptr>=udy) {errorkod=104; cfr:compressfail=1;}

```

### 13.3. Hafmanovo dekodiranje na PC računaru

```

procedure UNZIP; var i,k,xits:byte; ix,iy:word; delta:smallint;
ll:longint; lb:array[0..3] of byte absolute ll; x1:array[0..1]
of byte absolute delta; label l0;
begin with mainform do begin
  rptr[node]:=0; bitctr:=0; bits:=8;
  for i:=3 downto 0 do lb[i]:=getstreambyte; tsvar[node]:=ll;
  for i:=3 downto 0 do lb[i]:=getstreambyte; tevar[node]:=ll;
  for i:=0 to 3 do lb[i]:=getstreambyte;
  gothth[node]:=getstreambyte;
  gotyear[node]:=10+((lb[0] shr 2) and 63);
  gotmonth[node]:=((lb[0] and 3) shl 2)+(lb[1] shr 6);
  gotdate[node]:=((lb[1] shr 1) and 31);
  gothour[node]:=((lb[2] shr 4)+(lb[1] and 1) shl 4);
  gotmin[node]:=((lb[2] and 15) shl 2)+(lb[3] shr 6);
  gotsec[node]:=lb[3] and 63;
  if gothth[node]>99 then gothth[node]:=0;
  temperature[node]:=getstreambyte; voltage[node]:=getstreambyte;
  logwrite(node); bits:=3; k:=getstreambyte;
  for i:=0 to 2 do begin
    bits:=8; ay3[0]:=getstreambyte;
    bits:=4; ay3[1]:=getstreambyte;
    if (ay3[1] and 8)=8 then ay3[1]:=ay3[1] or 240;
    accbuf[0,i,node]:=abz end;
  fillchar(kodiran,sizeof(kodiran),0);
  for ix:=0 to 511 do begin
    bits:=1; if getstreambyte=0 then dubina[ix]:=0 else begin
      bits:=k; dubina[ix]:=succ(getstreambyte); bits:=dubina[ix];
    getstreamarray(ix);
  end end;
  for i:=0 to 2 do for ix:=1 to 3199 do begin
    for xits:=1 to 127 do begin
      bits:=xits; fillchar(tempkod,sizeof(tempkod),0);
      get2array; bits:=xits;
      for iy:=0 to 511 do if equcode(iy) then begin
        if iy=511 then begin
          bits:=dubina[511]; getstreamarray(512);
          bits:=8; x1[0]:=getstreambyte; bits:=4;
          x1[1]:=getstreambyte and 15;

```



```

        if (x1[1] and 8)=8 then x1[1]:=x1[1] or 240;
        bits:=xits end
    else begin
        delta:=iy-255; getstreamarray(512)
    end;
    accbuf[ix,i,node]:=accbuf[pred(ix),i,node]+delta;
    if accbuf[ix,i,node]>2047 then
        dec(accbuf[ix,i,node],4096)
    else if accbuf[ix,i,node]<-2048 then
        inc(accbuf[ix,i,node],4096);
    goto 10 end end;
    addlog('Neuspela dekompresija podataka sa stanice '
    +inttostr(succ(node))+'.'); exit;
10:end end end;

function EQUCODE(z:word):boolean; var i,j:byte; begin
    result:=false; if dubina[z]<>bits then exit; if bits=0 then
    exit; j:=pred(bits) shr 3;
    if j<>0 then for i:=0 to pred(j) do if tempkod[i]<>kodiran[i+(z
    shl 4)] then exit;
    i:=pred(bits) and 7;
    if (tempkod[j] and bmask[i])<>(kodiran[j+(z shl 4)] and
    bmask[i]) then exit;
    result:=true end;

function GETSTREAMBYTE:byte; var pb,bp:byte; begin
    pb:=bits; if bits>8 then bits:=8;
    if bits=0 then begin result:=0; exit end;
    ab2[0]:=inbuf[node,rptr[node]];
    ab2[1]:=inbuf[node,succ(rptr[node])];
    asm MOV CL,bitctr; MOV AX,abx; TEST CL,CL; JZ @0;
        SHR AX,CL; @0:MOV result,AL end;
    result:=result and poww[bits]; bp:=bits;
    if bp>8 then bp:=8; inc(bitctr,bp);
    if bitctr>7 then begin asm AND bitctr,7 end;
        inc(rptr[node]) end; bits:=pb end;

procedure GETSTREAMARRAY(ix:word); label l1; begin
    asm SHL ix,4 end;
l1:kodiran[ix]:=getstreambyte; if bits<9 then exit;
    dec(bits,8); inc(ix); goto l1 end;

```

## BIOGRAFIJA

Mr Miodrag Z. Malović, dipl. ing. elektrotehnike, rođen je 1967. u Beogradu, gde je završio osnovnu i srednju školu. Diplomirao je na Elektrotehničkom fakultetu Univerziteta u Beogradu 1995. godine na odseku Fizička elektronika, smer optoelektronika i laserska tehnika. Magistrirao je na Elektrotehničkom fakultetu Univerziteta u Beogradu 2003. godine, smer merenja u elektrotehnici, magistarskom tezom pod nazivom „Određivanje dinamičkih parametara pretvarača neelektričnih veličina sa primenama“.

Od 1994. do danas bavi se razvojem industrijskog softvera niskog nivoa u svrhu automatskog upravljanja proizvodnim i drugim procesima putem elektronskih vaga, programabilnih kontrolera, i drugih kompjuterizovanih senzora. Njegovi programi su instalirani u preko stotinu fabrika i drugih objekata (tehnički pregledi vozila, silosi, pruge, površinski kopovi, itd) širom Srbije i zemalja u okruženju.

Od oktobra 1995. do oktobra 2014. bio je zaposlen na Građevinskom fakultetu Univerziteta u Beogradu kao asistent na grupi predmeta iz oblasti tehnička fizika, građevinska fizika i fizička elektronika. Od aprila 2015. zaposlen je kao stručni saradnik u Inovacionom centru Tehnološko-metalurškog fakulteta Univerziteta u Beogradu.

Učestvovao je na projektu „Istraživanje kratkotrajnog i dugotrajnog monitoringa konstrukcija“ (TP-16023) u periodu od 2006. do 2009. godine, projektu „Istraživanje stanja i metoda unapređenja građevinskih konstrukcija sa aspekta upotrebljivosti, nosivosti, ekonomičnosti i održavanja“ (TR-36048) u periodu od 2009. do 2014. godine, i od 2009. je uključen u projekt Ministarstva prosvete, nauke i tehnološkog razvoja Srbije: „Astroinformatika: Primena IT u astronomiji i srodnim disciplinama“ (III-44002).

Koautor je nekoliko priručnika za laboratorijske vežbe iz grupe predmeta iz tehničke fizike, građevinske fizike i elektronike, kao i zbirke zadataka iz tehničke fizike („Tehnička Fizika: Zbirka rešenih zadataka sa ispitnih rokova“, Beograd, 2004). Koautor je: 2 tehnička rešenja, 3 rada u međunarodnim časopisima, 5 radova u domaćim časopisima, 5 radova na međunarodnim konferencijama, i 12 radova na domaćim konferencijama (od toga 11 štampanih u celini).

Prilog 1.

## Izjava o autorstvu

Potpisani Miodrag Malović

broj upisa  /

### Izjavljujem

da je doktorska disertacija pod naslovom

Razvoj i analiza senzorskog sistema za merenje vibracija građevinskih objekata

- rezultat sopstvenog istraživačkog rada,
- da predložena disertacija u celini ni u delovima nije bila predložena za dobijanje bilo koje diplome prema studijskim programima drugih visokoškolskih ustanova,
- da su rezultati korektno navedeni i
- da nisam kršio autorska prava i koristio intelektualnu svojinu drugih lica.

**Potpis doktoranda**

U Beogradu, 3. 9. 2015.



Miodrag Malović

Prilog 2.

## Izjava o istovetnosti štampane i elektronske verzije doktorskog rada

Ime i prezime autora Miodrag Malović

Broj upisa  /

Studijski program \_\_\_\_\_

Naslov rada Razvoj i analiza senzorskog sistema za merenje vibracija građevinskih objekata

Mentor prof. dr Tomislav Šekara

Potpisani Miodrag Malović

izjavljujem da je štampana verzija mog doktorskog rada istovetna elektronskoj verziji koju sam predao za objavljivanje na portalu **Digitalnog repozitorijuma Univerziteta u Beogradu**.

Dozvoljavam da se objave moji lični podaci vezani za dobijanje akademskog zvanja doktora nauka, kao što su ime i prezime, godina i mesto rođenja i datum odbrane rada. Ovi lični podaci mogu se objaviti na mrežnim stranicama digitalne biblioteke, u elektronskom katalogu i u publikacijama Univerziteta u Beogradu.

**Potpis doktoranda**

U Beogradu, 3.9.2015.



**Prilog 3.**

## Izjava o korišćenju

Ovlašćujem Univerzitetsku biblioteku „Svetozar Marković“ da u Digitalni repozitorijum Univerziteta u Beogradu unese moju doktorsku disertaciju pod naslovom:

Razvoj i analiza senzorskog sistema za merenje vibracija građevinskih objekata

koja je moje autorsko delo.

Disertaciju sa svim priložima predao sam u elektronskom formatu pogodnom za trajno arhiviranje.

Moju doktorsku disertaciju pohranjenu u Digitalni repozitorijum Univerziteta u Beogradu mogu da koriste svi koji poštuju odredbe sadržane u odabranom tipu licence Kreativne zajednice (Creative Commons) za koju sam se odlučio.

1. Autorstvo
2. Autorstvo - nekomercijalno
3. Autorstvo – nekomercijalno – bez prerade
4. Autorstvo – nekomercijalno – deliti pod istim uslovima
5. Autorstvo – bez prerade
6. Autorstvo – deliti pod istim uslovima

**Potpis doktoranda**

U Beogradu, 3.9.2015.

