

RIGA TECHNICAL UNIVERSITY
Faculty of Power and Electrical Engineering
Institute of Industrial Electronics and Electrical Engineering

Atis HERMANIS

Student of the Doctoral study program "Computer Control of Electrical Technologies"

**SHAPE SENSING BASED ON EMBEDDED
SENSORS FOR MOBILE CYBER-PHYSICAL
SYSTEMS**

Doctoral thesis

Academic supervisors
Dr.sc.comp., senior researcher
M. GREITĀNS
Dr.sc.eng., professor
O. KRIEVS

Riga 2016

**A DOCTORAL THESIS SUBMITTED TO RIGA TECHNICAL UNIVERSITY IN
FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF DOCTOR OF
SCIENCE IN ENGINEERING**

The public defense of the doctoral thesis for promotion to the doctor's degree in engineering will take place on 25th of November, 2016, at 13.00 in Room 212, Institute of Industrial Electronics and Electrical Engineering, Azenes 12/1, Riga.

OFFICIAL REVIEWERS

Professor, Dr.sc.ing. Ilja Galkins
Riga Technical University, Latvia

Professor, Dr.sc.ing. Aleksandrs Grakovskis
Transport and Telecommunication Institute, Latvia

Senior research scientist, Dr. sc. ing. Yannick Le Moullec
Tallinn University of Technology, Estonia

CONFIRMATION

I confirm that this doctoral thesis, submitted for a degree in engineering at the Riga Technical University, is my own work. The doctoral thesis has not been submitted for a degree in any other university.

Atis Hermanis (Signature)

Date:

The doctoral thesis is written in English, contains an introduction, 5 chapters, conclusions, references, 7 appendices, an index, 46 figures and one table, 118 pages in total. The list of references consists of 96 titles.

ABSTRACT

This work is dedicated for research on new methods for three dimensional shape data acquisition applicable in mobile embedded systems with limited resources. The aim is to facilitate development of new cyber-physical system applications in fields such as flexible electronics, robotics, smart textile and wearable electronics. A particular attention is paid to application of miniature low-power, low-cost hardware, efficient data acquisition and processing that can provide real-time operation. Also, the work focuses on methods that enable ubiquitous operation and does not rely on external infrastructure. In this work a novel method for shape sensing is proposed that is based on 3-axis acceleration and magnetic sensor nodes that are embedded into the material and can measure local orientation data. The proposed algorithm for global shape reconstruction from local orientation measurements ensures fast computations utilizing data from large number of sensors. Detailed simulations are provided to demonstrate feasibility of the proposed method. Additional contribution is related to new network architecture that allows efficient data acquisition from more than 200 low-power sensors that is required for proposed shape sensing algorithm. Also, practical implementation of proposed methods is demonstrated and shape reconstruction performance is evaluated in experimental tests by comparison to commercial Kinect v2 sensor that can act as three dimensional scanner. In the end of the work approbation of proposed methods in mobile cyber-physical system for medical applications is demonstrated. Real-time posture monitoring and feedback system is developed using proposed methods and applied for medical studies in collaboration with medical specialists, demonstrating the applicability of the method in practical mobile embedded systems.

ANOTĀCIJA

Šis darbs ir veltīts jaunu trīs dimensiju formas datu ieguves metožu izpētei, kas izmantojama mobilās iegultās sistēmās ar ierobežotiem resursiem. Darba mērķis ir veicināt attīstību jaunu kiberfizikālo sistēmu pielietojumos tādās jomās, kā viedie audumi, valkājamas sistēmas, robotika un lokanā elektronika. Darbā tiek apskatīta miniatūru zemu izmaksu un neliela enerģijas patēriņa ierīču izmantošana, kā arī efektīva datu savākšana un apstrāde, kas var nodrošināt darbību reālā laikā. Papildus uzmanība tiek vērsta uz pieeju, kas nodrošina mobilu darbību un nav atkarīga no ārējas infrastruktūras. Šajā darbā tiek piedāvāta jauna metode formas mērīšanai, kas ir balstīta uz trīs asu paātrinājuma un magnētiskā lauka sensoru mezgliem, kas tiek iestrādāti materiālā un var iegūt datus par tā orientāciju dažādās vietās. Piedāvātais algoritms kopējās formas atjaunošanai no lokālo orientāciju datiem nodrošina ātrus aprēķinus, vienlaikus izmantojot datus no liela daudzuma sensoru mezglu. Detalizētās simulācijās ir pierādīta piedāvātās metodes darbība un novērtētas kļūdas. Papildus ieguldījums ir saistīts ar jaunas sensoru tīkla arhitektūras izveidi, kas var nodrošināt efektīvu datu savākšanu no vairāk kā 200 zema jaudas patēriņa sensoriem, kas var būt nepieciešami virsmas atjaunošanas algoritmam. Tāpat ir demonstrēta metodes praktiska realizācija, kā arī virsmas atjaunošanas darbības kvalitāte ir novērtēta eksperimentālos testos, salīdzinot rezultātus ar komerciālo Kinect v2 sensoru, kas var darboties kā trīs dimensiju skeneris. Darba beigās ir demonstrēta arī piedāvāto metožu aprobācija mobilā kiberfizikālā sistēmā medicīnas pielietojumiem. Ir izstrādāta reāla laika stājas monitoringa un atgriezeniskās saites sistēma, un tā pielietota medicīniskos pētījumos sadarbībā ar medicīnas speciālistiem, demonstrējot metožu pielietojamību praktiskās mobilās iegultās sistēmās.

ACKNOWLEDGMENTS

First of all, I would like to thank all of my current and past colleagues from the Institute of Electronics and Computer Science (EDI), Riga, Latvia. All of my research work has been done during the period of around six years working there as an electrical technician, a scientific assistant and a researcher. I have gained a huge amount of valuable professional knowledge which has been essential in my research work and for writing this thesis. I also have been able to join fruitful and fun discussions related to science and other topics during tea breaks at work, business trips and in free time. Special thanks goes to residents of room number 201 Kaspars Ozols and Roberts Kadiķis who have been my roommates throughout all of the years in EDI and have always provided pleasant ambiance and valuable advices. Also big thanks is dedicated to Ričards Cacurs and Krišjānis Nesenbergs with both of whom at times I have done large amount of joint work.

I am very grateful and want to give the biggest acknowledgment to my main PhD supervisor Modris Greitāns even thou during the supervision of my Bachelor's and Master's thesis works he used to say that you don't need to acknowledge the supervisor as it is his job. He provided the main encouragement and motivation to start my PhD studies, which most probably would not had happened without him. Also, thanks to professor Dr.sc.ing. Oskars Krievs for providing quick and understanding aid for cooperation with Riga Technical University.

I would also like to thank my family especially parents for supporting me and always trusting in the things I do. Additional acknowledgment goes to my grandfather Dr.habil.sc.comp. Ēvalds Hermanis for quite hesitant but still worthwhile introduction to the world of science research.

Last but not least, I thank all my friends for getting by with me and supporting in the times of need. Special thanks goes to I.A. who put a considerable effort up in proofreading this *piece*, all criticism relating this goes to him.

Atis Hermanis,

Riga 2016

TABLE OF CONTENTS

ABBREVIATIONS	10
NOMENCLATURE	11
INTRODUCTION	12
1. METHODS FOR SURFACE SHAPE SENSING WITH EMBEDDED EQUIPMENT.	16
1.1 Shape sensing by measuring bending of the material	16
1.2 Shape sensing by measuring orientation of object segments.	18
1.3 Conclusions	23
2. SHAPE SENSING BASED ON INERTIAL/MAGNETIC SENSOR MODULES	25
2.1 Shape sensing based on accelerometer modules	25
2.1.1 Orientation estimation using local gravity field vector	25
2.1.2 Shape reconstruction from discrete inclination values	27
2.2 Shape sensing based on accelerometer and magnetometer modules.	32
2.2.1 Orientation estimation using local gravity and magnetic field vectors	33
2.2.2 Shape reconstruction from sensor orientation data	35
2.2.3 Proposed method	36
2.2.4 Simulations	39
2.3 Conclusions	42
3. HARDWARE ARCHITECTURE FOR SHAPE SENSING SENSOR NETWORK	44
3.1 State of the art	44
3.2 Enhanced daisy-chained SPI	46
3.3 Conclusions	49
4. EXPERIMENTAL SYSTEMS	51
4.1 Surface reconstruction with accelerometer network	51
4.2 Surface reconstruction with accelerometer and magnetometer network	56
4.2.1 Orientation estimation	56
4.2.2 Sensor network	57
4.2.3 Surface reconstruction	62

4.3	Conclusions	69
5.	SHAPE SENSING APPLICATION IN MOBILE CYBER-PHYSICAL SYSTEMS. . .	72
5.1	Posture monitoring and feedback	72
5.2	Approbation and pilots for medical applications	77
5.3	Conclusions	81
	DISCUSSION AND CONCLUSIONS	83
	APPENDICES	88
	Appendix 1 <i>Schematics of LSM303DLHC sensor board</i>	89
	Appendix 2 <i>Schematics of master data acquisition board with Bluetooth</i>	90
	Appendix 3 <i>C code for MSP430g2553 of the sensor node</i>	91
	Appendix 4 <i>C code for MSP430g2553 of the master device</i>	98
	Appendix 5 <i>Main Matlab code of shape reconstruction algorithm</i>	105
	Appendix 6 <i>"Unihaus" ltd. approbation review</i>	111
	Appendix 7 <i>"Mēs esam līdzās" approbation review</i>	112
	BIBLIOGRAPHY	113

LIST OF FIGURES

2.1	Visual representation of accelerometer data vector.	26
2.2	Structure of nominal surface model.	27
2.3	Structure of reconstructed surface model.	28
2.4	Structure of segment representation.	29
2.5	Structure of a work-set.	29
2.6	The flowchart of the brute-force algorithm.	31
2.7	Surface segment structure.	37
2.8	Structure of control point connections.	39
2.9	Comparison of reconstruction methods.	40
2.10	Synthesized 3D curve.	41
2.11	Impact of various error sources.	42
3.1	Daisy-chained SPI structure.	46
3.2	Structure of enhanced daisy-chained SPI network.	47
3.3	Structure of enhanced daisy-chained SPI slave.	47
3.4	Clock pulse width variance with different comparator reference signals.	48
4.1	Accelerometer LIS331DLH on custom circuit board.	51
4.2	Structure of accelerometer network.	52
4.3	Structure of the experimental accelerometer system.	52
4.4	Approximated model form accelerometer sensor grid data.	54
4.5	Average inclination of model vectors from vertical position and resulting Δ_{avg}	54
4.6	Human back posture model approximation.	55
4.7	Experimental test setup of accelerometer network.	55
4.8	Comparison of orientation angle measurements with "Xsens" module and sensor from acceleration/magnetic grid.	57
4.9	Experimental sensor chain with 60 sensor nodes.	58
4.10	Structure of the sensor node (physical dimensions $13 \times 13 \times 2$ mm).	59
4.11	Relation between number of sensors in the network and last sensor supply voltage.	62

4.12	Sensor network for proof-of-concept setup. Blue lines illustrate sensor mutual grid structure.	63
4.13	Structure of data acquisition system.	63
4.14	Master data acquisition module.	64
4.15	Fabric with embedded sensor layer and corresponding reconstructed shape models in Android application.	64
4.16	Experimental setup diagram.	65
4.17	Reconstruction accuracy comparison with reference object.	66
4.18	Difference maps of shape sensing sensor array in comparison to Kinect reconstruction.	67
4.19	Difference distribution between points from proposed method and Kinect sensor. Reference in the center.	67
4.20	Difference distribution between points from proposed method and Kinect sensor. Reference in the corner.	67
4.21	Posture monitoring experimental setup.	68
4.22	Difference distribution between points from proposed method and Kinect sensor during dynamic movement.	69
5.1	Structure of posture monitoring system.	72
5.2	Construction of approximated 3D posture model.	73
5.3	Flowchart of feedback control algorithm in Android application.	75
5.4	Android application screens and sensor device.	76
5.5	Four test subject average posture statistics with different feedback threshold values.	77
5.6	Smart wearable device for posture monitoring.	78
5.7	Pilot study of SWD application in rehabilitation center "MEL".	80
5.8	Feedback interface for human head position feedback.	80
5.9	Pilot study of SWD with head position feedback in rehabilitation center "MEL".	81

ABBREVIATIONS

2D – *Two Dimensions*

3D – *Three Dimensions*

CAN – *Controller Area Network*

CLK – *Clock signal*

CPS – *Cyber-Physical Systems*

I2C – *Inter-Integrated Circuit*

ICP – *Iterative Closest Point*

MEMS – *Micro Electro Mechanical Systems*

MISO – *Master In Slave Out*

MOSI – *Master Out Slave In*

LiDAR – *Light Detection And Ranging*

PC – *Personal Computer*

PVDF – *PolyVinylidene DiFluoride*

RGB – *Red Green Blue*

RMS – *Root Mean Square*

SAW – *Surface Acoustic Wave*

SDK – *Software Development Kit*

SIMO – *Slave In Master Out*

SOMI – *Slave On Master In*

SPI – *Serial Peripheral Interface*

SPP – *Serial Port Profile*

SS – *Source Select*

SWD – *Smart Wearable Device*

TRIAD – *Three Axis Attitude Determination*

UART – *Universal Asynchronous Receiver Transmitter*

USB – *Universal Serial Bus*

NOMENCLATURE

d – distance between sensors

\vec{v} – vector

\vec{n} – rotation axis

\vec{v}' – transformed vector

θ – rotation angle [°]

R – rotation matrix

q – rotation quaternion

n – number of rows

m – number of columns

α – rotation angle around vertical axis

Δ_d – matching error

F – flatness coefficient

g – Earth gravity constant ($9.8 \frac{m}{s^2}$)

E_g – Earth gravity field vector

E_m – Earth magnetic field vector

S_g – Earth gravity field vector measurement

S_m – Earth magnetic field vector measurement

V_{ref} – reference voltage [V]

T_p – clock signal pulse width [s]

T_{ref} – buffered clock signal pulse width [s]

Δ_V – voltage drop [V]

I_{cc} – supply current [A]

V_{cc} – supply voltage [V]

R_w – wire resistance [Ω]

T_{sample} – sampling period [s]

f_{CLK} – clock signal frequency [Hz]

INTRODUCTION

Ability to measure 3D geometric properties of an object can provide data about shape of the object. If the object is measured in real-time, continuous deformations can also be monitored. This data is valuable for various new emerging applications. Shape sensing ability could be used in robotic systems to gather information about objects in outside environment [1] or to obtain precise feedback of robot manipulator position [2]. In the new emerging field of flexible electronics [3] shape sensing could provide direct feedback of device configuration and allow new ways of interaction with these devices. Additional new fields that are growing very fast and could benefit from shape sensing technologies are wearable computing and smart textiles [4]. Here the data about textile shape could be used to provide real-time information of wearers posture and movement. The clear potential applications of shape sensing combined with advancements in new sensing technologies, for example Micro Electro Mechanical Systems (MEMS) [5], attract more researchers to focus on this field.

The above mentioned applications usually are controlled by specific group of computers. People usually pay attention only to computers and software that are used for information processing for human consumption and interaction such as text editing, browsing web, emailing, etc., which is the most visible group. However, a vast majority of computers in use belong to a much less visible group. These computers do the essential work of controlling different appliances and processes such as car engines, automotive subsystems, radio transceivers of mobile phones and other wireless devices, washing machines, refrigerators, dishwashers and other home appliances, robots and conveyor systems in manufacturing facilities and just about any other appliance imaginable. This much less visible group of computers is called embedded systems [6].

Historically embedded systems were treated as small computers. The engineering problem was defined as copying computer functionality with limited resources - limited processing power, limited energy, limited amount of memory, etc.; therefore it was not much distinguished from computer science. As the desire to control more and more complex processes increased, researchers started to notice that the main challenges in embedded systems often arise from interaction with physical world rather than just limited resources. Because of this a new sub-field called cyber-physical systems (CPS) was introduced. A CPS can be seen as an extension

of embedded systems where the particular focus is on integration of computation with physical processes. "When studying CPS, certain key problems emerge that are rare in so-called general-purpose computing. For example, in general-purpose software, the time it takes to perform a task is an issue of performance, not correctness. It is not incorrect to take longer to perform a task. It is merely less convenient and therefore less valuable. In CPS, the time it takes to perform a task may be critical to correct functioning of the system" [7]. For example, a car airbag system has to be deployed in exact time and at exact circumstances in physical world to be useful. Any deviation from it means a critical error for the system.

To provide a close interaction between computers and physical world a system with specific structure has to be designed. Embedded computers and networks monitor and control the physical processes, usually feedback loops are implemented where physical processes affect computations and vice versa. To provide this functionality areas such as embedded computing, computer networks and sensor systems form the core of CPS research. Operation of CPS is closely dependent on physical environment, therefore novel sensing technologies and sensor systems have a huge role in facilitation of development of new and more functional CPS. In addition, a number of CPS applications such as wearable devices, portable electronics, robotics, etc. often require mobile operation, thus when developing sensor systems particular attention has to be paid not only to ensure system operation with limited resources but also to provide ability to operate in a portable system in different environments with limited supporting infrastructure.

In relation to CPS, all techniques for 3D shape data acquisition can be divided in two major categories. The first category performs some form of remote sensing utilizing either active or passive external equipment such as single and stereo camera setups, time of flight cameras, LiDARs, touch probes, and other solutions based on various principles - laser, light, sound, physical contact [8]. The second category uses sensors that are placed on or even embedded into the measured object itself. Most of the previous studies that are related to measurement of object 3D geometric properties fall under the category that utilizes external equipment [9]. These approaches can often provide object models with high accuracy; however, there are two major problems that are inherent in system architecture with external equipment. One is that it requires specially equipped surroundings, which considerably limits the range of operation and portability. The other significant problem is occlusions. External sensors have to have a clear line of sight to the object which is not always practical. In addition, an object with more compli-

cated shape can sometimes itself occlude some of its details. Both of these limitations prevent methods with external equipment from being used for mobile CPS applications such as smart shape aware textile, flexible electronics and other applications that require mobile measurement of relatively complex 3D shapes.

The aim of this work is to facilitate the development of new CPS applications by exploring new 3D free-form shape sensing methods. The focus is on the research of methods utilizing equipment that enables ubiquitous operation of mobile CPS. Also, attention is paid to the employment of miniature low-cost, low-power hardware, the development of efficient data processing algorithms suitable for real-time CPS and the development of methods for efficient data acquisition to gather data from a large number of sensors to provide high resolution measurements.

At the beginning of the work the following tasks were defined:

- review the literature related to object shape measurement with sensors embedded in the measured object and identify the most suitable approach for mobile CPS applications;
- develop a shape reconstruction method based on embedded sensors that can obtain real-time 3D shape information and can be implemented in an embedded system;
- develop a method for data acquisition that can effectively gather data from a large number of sensors required by the shape reconstruction algorithm;
- experimentally validate the feasibility of the proposed methods and practically test the shape reconstruction performance;
- implement and approbate the proposed methods in a mobile CPS application.

During the development of this work the following thesis statements have been promoted and proven:

1. Low-cost inertial and magnetic sensor network can be used to reconstruct the shape of 3D surface with an average error less than 6 % relative to the smallest surface dimension.
2. It is possible to acquire the necessary information about 3D orientation from more than 200 sensors with more than a 50 Hz sampling rate by utilizing the proposed enhanced-daisy chained 4 wire architecture and off-the-shelf hardware.

3. The proposed fixed-length vector algorithm allows reconstructing the object shape more than 40 times faster on the same hardware with negligible (less than 1 %) reduction in accuracy compared to state-of-the-art integrating algorithm.

The rest of the work is organized as follows. In Section 1, a detailed literature review of various shape sensing methods that are based on embedded equipment is provided. The most suitable approach for mobile CPS is identified. In Section 2, methods for 3D shape reconstruction based on local orientation data are proposed. First a case with application of acceleration sensor network together with its limitations is described. Then a method utilizing acceleration and magnetic field sensor network is proposed. Detailed simulations are provided to validate the feasibility of this method. In Section 3, various communication interfaces available for integral circuits are reviewed and a new method is proposed for an effective data acquisition from a large number of sensors suitable for shape sensing applications. Section 4 is dedicated to the description of development and testing of experimental systems for shape sensing sensor network. Detailed shape reconstruction performance evaluation of the proposed method is provided here. In final Section 5, the implementation and approbation of proposed methods in mobile CPS is shown. A mobile posture monitoring device for medical applications that is based on the new methods proposed in this work is demonstrated. An approbation in medical institutions for patients with posture and movement dysfunction is also described. At the end of the work overall discussion and conclusions are given. Conclusions are supplemented with the list of author's published papers, conference abstracts, demos, visited conferences and projects relevant for this work.

This Doctoral thesis is based on the papers **[10, 11, 12, 13, 14, 15]** by the author of this Thesis and paper co-authors Ricards Cacurs, Modris Greitans, and Krisjanis Nesenbergs. Some additional information found in this work is previously published in a master's thesis work **[16]** by the author of this work and in the conference abstracts and demos **[17, 18, 19, 20, 21]** with additional co-authors Armands Ancans, Emil Syndykov, Leo Selavo, Santa Geidane, and Andra Greitane. Here and further in the text, the references with authors own contribution are highlighted in bold and underlined.

1. METHODS FOR SURFACE SHAPE SENSING WITH EMBEDDED EQUIPMENT

In the literature a number of studies try to acquire a 3D shape of object surface by utilizing sensors that can be placed on the object or even embedded into the object. These are often referred as self sensing devices as they include the required sensors themselves. Usually, a network of sensors is used to provide some local information from multiple locations on the object, which can then be used together to obtain global shape characteristics. These studies can be categorized in two major subgroups. In one subgroup, the information about object shape is obtained by measuring the bending of the material based on various physical principles that can be sensed by an electronic equipment. In the other subgroup, the shape of the object is estimated by measuring the orientation of different segments of the shape. The global shape characteristics are then reconstructed from the local orientation information basing on known interconnection model between the segments.

1.1. Shape sensing by measuring bending of the material

Different complicated shapes can be seen as deformations of some more simple reference state, for example a flat surface. This implies that sensing of object shape in a straightforward way can be done by measuring the bending of the material of the object in different locations along the object. A number of physical principles can be used to produce bend sensor [22]. Multiple such sensors can be combined to obtain bending information along different degrees of freedom, as well as to measure multiple different deformations along the object.

Authors in [23] propose a surface profile sensing mechanism that is based on surface acoustic wave (SAW) sensors. The sensors were fabricated from Polyvinylidene difluoride (PVDF) that ensures flexibility. In an experimental setup it was demonstrated that the produced sensor changes the amplitude and the phase of the input signal in correlation with bending radius of the sensor, however, it was also stated that sensor output signal is extremely small and noisy relative to the input signal, therefore significant attention has to be paid to signal filtering and amplification. Also a system setup that allows surface shape sensing using two independent networks of 2D SAW sensor grids was superficially described and no simulation or experimental results for surface shape sensing performance was provided.

Authors in [24, 25] use ShapeTape sensor, which is a thin array of fiber optic curvature sensors laminated in a rubber band that can sense its bend and twist. Bend and twist are measured at intervals by two fiber optic bend sensors. "Resolution is limited by the spacing of these sensors. By summing the bends and twists of the sensors along the tape, the shape of the tape relative to the first sensor can be reconstructed." [25] The sensors are used to reconstruct curves, also, the possibility to reconstruct the shape of surfaces is stated. In [24] the main focus is on sensor application for various human-computer interaction techniques. The precision of reconstruction is not evaluated in detail, but the authors state that the method is less suitable for technical modeling due to the lack of sensor precision. Problems such as hysteresis and drift are reported in [25]. Also, in both sources the surface reconstruction using multiple sensor bands to obtain deformations along whole surface is not tested.

A number of studies use sensors that are based on piezoelectric effect and can sense bending. In [26], a PVDF material sensors with piezoelectric properties are used for shape measurement. The correlation between the output voltage and direction as well as the amplitude of bending is demonstrated. Basic concepts for shape reconstruction are also shown with limited resolution setup that utilizes only four sensors. In [27], similar approach is used. An array of 4×4 piezoelectric sensors are placed on thin film. Sensor data are used to recognize different discrete shapes limiting the applications for free-form shape measurement. In both studies no detailed shape reconstruction accuracy evaluation is provided.

In [28], a transparent sensing surface for 3D surface shape detection is proposed. A network of PyzoFlex [29, 30] bend sensors are used in order to profile the surface and reconstruct its shape. The sensors can be printed on thin transparent substrate layer. A specific layout of the sensor network ensures the ability to profile different free-form deformations of the material layer. Two reconstruction algorithms were proposed - weighted linear interpolation and learning based continuous regression. Detailed evaluation of accuracy was performed using custom multi video camera setup as ground truth, that can detect different marker locations on the surface and reconstruct its shape. The reported average error over 10 000 bending poses was $15 \pm 7mm$ for learning based algorithm and $18 \pm 9mm$ for linear interpolation algorithm with increased error in the corners of the surface where most of the deformations occurred. In addition, several application scenarios were proposed. Main application example included the use of the sensor device as a transparent tablet display cover that can allow intuitive multilayer

interaction due to the transparent properties of the sensor. For example, the transparent sensor cover allows a variety of paper like interactions with digital display, where the sensor layer can be physically lifted up as a page of the book to peek what is underneath the top layer in user interface of applications such as image and animation editors or games. Other applications relating to human-computer interaction also were proposed. Several problems were reported including sensor deterioration over prolonged periods, folding limitations, sensor drift and challenges in processing algorithms especially for larger deformations. Also the physical implementation of printed sensors on composite substrate limits their integration in several environments, for example, smart garment applications.

In [31] a fabric with an embedded layer of conducting polymer that has piezoresistive properties is used in order to obtain shape of human body parts. A mesh type array of sensors is embedded into tight garments to sense surface strain fields. It is assumed that strain fields change their intensity around deformation triggering points, for example, joints on human body, allowing to estimate the bending angle of the deformation. Problems such as hysteresis and aging of the sensors as well as the difficulty to represent high-dimensional deformations were among the reported limiting the accuracy of the method and its application for surface shape sensing in general case.

Most of the surface shape sensing methods that are based on measurement of material bending have similar properties. The most frequently mentioned problems include hysteresis of sensors, aging, limited resolution, complex hardware and data processing algorithms to measure deformations with high degree of freedom on larger surface areas. Also a lot of focus is on recognition of discrete deformations rather than on universal full free form 3D shape reconstruction. This is partly because of the complexity of mapping the raw sensor readings to precise continuous deformations due to various sensor errors and other effects described earlier.

1.2. Shape sensing by measuring orientation of object segments

Recent advances in MEMS technologies allow to manufacture miniature low-cost sensors [5] that can be embedded into materials to provide local information which can be used to obtain global shape characteristics. Utilizing orientation information instead of acquiring displacement from acceleration data allow to remove the sensor drift that is often associated with inertial sensors. Magnetic sensors are often used in combination with accelerometers in order to obtain

the full three degrees of freedom orientation measurement. The measured orientations of different object segments can be applied to the model of segment interconnections to reconstruct the shape of the object.

Most of shape sensing methods that are based on orientation measurement with inertial sensors are used for specific non-general purposes. A complex object that can change its shape with high degree of freedom deformations usually is monitored by dividing it in a number of rigid segments and applying the orientation sensor to each segment. The measured orientations are then applied to the model of the in order to retrieve specific parameters or reconstruct the shape of the object relative to some nominal state. Most popular examples of such applications include wearable systems that allow monitoring of different poses and movements of the human body. In this case the sensors are attached to various human body parts and the pose of the body is reconstructed using biomechanical models. It must be stated that movements can be interpreted as a sequence of different subsequent poses similarly as video in essence is a sequence of static images.

In [32] the validity and reliability of accelerometer application for general human posture and movement monitoring is studied. It was showed that accelerometers can provide valuable information despite measurement errors which mainly are the result of dynamic accelerations. In [33] human inclination is measured using three-axis accelerometer in order to characterize human pose and movement. In [34] six 2-axis accelerometers are used to measure human back posture in sagittal plane and in [35] three 3-axis sensor modules are used to measure it both in sagittal and coronal plane. In [36] 3-axis acceleration, magnetic and gyroscope sensor nodes are used to measure the angle of the ankle joint. The method is applicable to measurements of other joints in general in order to reconstruct articulated object shape.

A number of studies measure poses and also movements of the whole human body. In [37] an accelerometer and gyroscope sensor network is proposed as inexpensive alternative to optical motion capture systems. The measurements of human body shape are done in sagittal plane only. In [38] a system based on 3-axis acceleration, magnetic and gyroscope sensor nodes is used in order to reconstruct human body pose and motion. Similar approach is used in [39] where a wearable motion capture system is described. The system is usable in different conditions including dynamic motions and can accurately reconstruct the shape of the human body. Similar approach is described in [40] and [41]. In [40] particular attention is paid to magnetic sensor

disturbance compensation and studies of different sensor fusion algorithms including inertial sensor fusion with optical sensors. In [41] the author has a particular focus on inertial sensor signal processing to minimize drift.

In contrast only a few researchers have tackled the problem of reconstructing free form shape from inertial/magnetic sensors in general case. In [42] authors propose an accelerometer network applied to a cloth-like sheet that can measure 3D shape of its own configuration. The proposed solution consists of rigid lattice structure where a 3-axis accelerometer is attached to each link of the structure. A roll and pitch angle of each link is measured directly with accelerometer. A specific algorithm in combination with deformations with limited degrees of freedom enforced by the physical lattice is proposed to estimate the third rotation degree of freedom (yaw - rotation around vertical axis) for each link and thus estimate the shape of the sensor structure. The feasibility of simple shape reconstruction is demonstrated with an experimental setup using 12 sensors, but no detailed shape sensing performance analysis is provided. In [43] a tactile sensing application of the same method is proposed. The tactile sensing is planned to be achieved by wrapping a compressible material in the shape measuring sheet. The tactile information could be obtained by knowing the stiffness coefficient of the material and measuring the place and amplitude of deformation. In [44] authors propose to also add magnetic sensors to each link of the lattice in order to provide a direct method for estimation of the third rotation degree of freedom (yaw - rotation around vertical axis). In [45, 46] the same authors introduce larger test setup with 24 links equipped with 3-axis accelerometers and magnetometers. Simulations to determine the sensor noise effects are showed, however, still no detailed shape reconstruction performance evaluation is provided. Problems such as limited resolution, sensor wiring difficulties and limited accuracy are reported.

In [47] authors propose an artificial robotic skin architecture. The architecture is developed to provide rich and direct feedback about robot interaction with the outside world. It is based on flexible hexagonal cells that can uniformly cover various surfaces and provide cell to cell connections for data transfer. Basing on this architecture in [48] authors demonstrate an artificial skin that can reconstruct the 3D surface of robotic body parts. The shape reconstruction method uses an accelerometer network that is embedded into a flexible material with the proposed cell type architecture, where each cell is equipped with one 3-axis acceleration sensor. The relative orientation of each segment is determined by a corresponding accelerometer and the shape is

reconstructed using some a priori knowledge and a number of assumptions about the model of the skin cell structure and its properties. An experimental prototype is developed and its performance is validated by reconstructing the shape of two objects with known dimensions. The reconstruction procedure requires putting the measured object in two different poses. The reported error for these two experiments were up to 5%, but a statistically insignificant number of tests were performed. Also, problems such as sensor noise, limited resolution due to the small amount of sensors for curvature radius of tested surface were reported.

Authors in [49, 50] describe their work on 3D curve reconstruction from orientation measurements. A ribbon of acceleration and magnetic sensor modules is proposed which is capable of measuring orientations at different places along the curve. These orientations are used to obtain tangents of the curve from which the shape of the curve is reconstructed. A prototype device was built providing reconstruction of single 3D curve with 5 frames per second. In [51] multiple such ribbons are applied for 3D surface shape reconstruction. Multiple curves along the surface are reconstructed and then the surface area between the curves is interpolated in order to reconstruct the surface shape. No practical evaluation of the method is provided, but simulations show a maximum error of 8.23% for more complicated shapes. In [52] the same method is applied to measure animated surfaces with variable shape in time. A mechanical device for validation of system performance is proposed, but not yet tested. However, simulation results suggest feasibility of the proposed system. Finally, in [53] the authors propose a smart textile capable of measuring its own shape. 3-axis acceleration and magnetic sensor nodes are embedded into the textile in a grid like pattern to provide local orientation information. Then similar algorithm as in [51] is used to reconstruct the shape of the textile. Real time surface reconstruction is demonstrated with 15 Hz frame rate using 3 by 3 sensor grid embedded into the fabric. No performance evaluation in terms of accuracy is provided.

The authors of aforementioned works [49, 51, 52, 53] report several problems for the proposed methods including difficulty to provide reference for shape reconstruction accuracy evaluation and complicated data acquisition process from sensor network to obtain real-time measurements with reasonable frame rate. Because of this, practical implementation and system performance is not tested in detail, especially for larger sensor networks, that could provide sufficient resolution for measuring detailed shape deformations with periods in the order of few centimeters and less. In addition, the methods rely on detailed data interpolation between sensor

nodes and are mathematically non-trivial requiring relatively high processing power for implementation on portable devices.

In many of the aforementioned works accelerometers are not the only sensor type used for orientation estimation. 3-axis accelerometers alone can only measure direction of single reference vector, providing incomplete sensor orientation estimate. In order to obtain full 3D surface model another reference direction has to be measured such as the Earth's magnetic field. Sensor orientation estimation from gravity and magnetic field measurements is discussed in Section 2.2.1.

In addition to accelerometers and magnetometers many of previous works especially those related to dynamic movement measurements also use gyroscopes. Gyroscopes often are added to the system to provide accurate orientation estimates in dynamic conditions and when mechanical vibrations occur [54]. This is due to the fact that accelerometers and gyroscopes can be seen to have complimentary properties because of their different noise parameters. Gyroscope based orientation measurements mainly have low frequency noise often referred as drift [55]. Drift occurs because gyroscopes directly output angular velocity instead of rotation angle. In order to obtain relative orientation angle the sensor output data has to be integrated over time. During integration various sensor errors accumulate over time producing a low frequency drift. In contrast accelerometer based orientation mainly suffers from high frequency noise that occurs due to the high dynamic accelerations and vibrations. Similar noise characteristics arising from noisy electromagnetic environment are seen in magnetic sensors that are often used in combination with accelerometers to provide full three degrees of freedom orientation measurements. To utilize this complimentary nature a number of filters have been proposed for accelerometer, magnetometer and gyroscope sensor data fusion. Some of the most widespread techniques are various modifications of the Kalman filter [56, 57, 58, 59]. Usually the rotation kinematic model based on gyroscope data is used in Kalman filter update step and then the data from acceleration and magnetic sensors are used to remove the gyroscope drift. Several other methods are proposed to limit required computing power and facilitate implementation in embedded systems. In [60] a gradient descent algorithm is proposed and in [61] complimentary filter is proposed for the same task. In both articles similar accuracy with reduced computing power is reported, however, due to the nonlinear nature of the systems it is hard to make exact comparison as different algorithms in various conditions can have particular advantages or drawbacks [62].

Despite the advantages that gyroscopes can provide there are also considerable drawbacks. Gyroscope application for orientation estimate requires integration of sensor output. This implies that the sensor data must be sampled and processed with a relatively high frequency and the orientation measurement is always dependent on previous measurements. This noticeably increases the required computing resources for sensor signal processing when compared to orientation estimation with accelerometers and magnetometers. In addition, in modern MEMS low-power systems gyroscopes can consume over 17 times more current than accelerometers and magnetometers combined [63].

1.3. Conclusions

There are a number of studies that are dedicated to 3D shape acquisition with embedded equipment, however, many of these just provide superficial method descriptions leaving some important details not yet resolved which limits validation and practical implementation. None of the previous works evaluate proposed solutions in detail for general context in terms of accuracy, required computing power, physical implementation and cost. Usually studies either lack detailed evaluation of practical implementation and real world performance or are suited for a limited scope of applications.

Measurement of material bending with different approaches provide potentially interesting solutions for the problem. The main advantage is that the information of bending angle and curvature radius is direct geometric data that can be used in the object shape model. Despite this, a bend sensor usually provides data regarding one degree of freedom of a single deformation. In order to obtain information of more complicated 3D deformations and curvatures in multiple places along the object, sensor networks with complicated architectures have to be designed. This also introduces challenges in the sensor signal processing where data from these sensor structures have to be interpreted as an object shape. The task of signal processing is cumbered even more due to various physical imperfections of these sensors such as hysteresis, nonlinearity and aging. These drawbacks provide difficult application of these methods in mobile CPS.

The shape sensing by measuring orientation with inertial/magnetic sensors of object segments provides an interesting alternative. The most successful applications of this approach are related for specific tasks such as human body pose and movement monitoring where a relatively small amount of sensors is required. Very limited previous work has been done to implement

and experimentally validate this approach for general case free form shape measurements such as acquisition of surface shape. This is mainly due to the problems related to design of sensor network for effective data acquisition from large number of low-power sensors and relatively complicated data processing algorithms that can require significant computing power. Despite this, high degree of freedom measurement within single sensor, good measurement repeatability, highly efficient and cheap production technique of MEMS and easy integration in embedded systems makes this approach very promising. Other advantages include the ability to provide not only relative shape of the object, but also information about object orientation relative to some general reference frame such as earth.

Due to the clearly identified problems, no directly foreseeable limitations from fundamental physics point of view and large number of potential applications the author further in this work chooses to study shape sensing based on orientation measurements from inertial/magnetic sensors. From inertial sensors only accelerometers are considered, because gyroscopes consume a significantly larger amount of current and have resource intensive signal processing requirements as stated in previous subsection, which would considerably reduce method applicability in mobile CPS. Also, a large number of solutions require shape measurements with small movement accelerations compared to gravity which even further reduce the requirement of gyroscopes.

2. SHAPE SENSING BASED ON INERTIAL/MAGNETIC SENSOR MODULES

2.1. Shape sensing based on accelerometer modules

Geometric properties of 3D object shape can be detected using multiple three-axis accelerometers arranged in a grid pattern along the object surface [15]. Using the phenomena, that in static conditions, acceleration measured by accelerometer is only due to the gravitational field of the Earth's, orientation of each accelerometer in 3D space relative to the ground can be obtained except for the rotation angle around vertical axis. This happens as each accelerometer in static states can be seen as an 2D inclinometer providing no heading information (rotation around vertical axis). If each accelerometer is applied to a particular segment of the surface, then sensor orientation data provide information about surface segment relative orientations. By comparing the orientation of each segment and applying the physical constraints of degrees of freedom enforced by the shape of the grid, it is possible to approximate the shape of the surface.

2.1.1. Orientation estimation using local gravity field vector

Any orientation of a rigid body can be represented as a rotation of the body reference system within the global reference system. The most popular way of representing 3D rotations are the Euler angle rotation sequences. However, this method suffers from multiple drawbacks such as gimbal lock and non-unique representation. Rotation representation with quaternions help to avoid these drawbacks. In short, quaternion representation defines one rotational axis \vec{n} and one rotation angle θ instead of sequence of three rotations represented by Euler angles.

In static conditions normalized 3-axis accelerometer measurement vector (a_x, a_y, a_z) can be defined as a vertical vector $(0, 0, 1)$, which has been rotated by an angle θ about an axis (a_x, a_y, a_z) relative to some global reference system (Fig. 2.1).

Angle θ can be calculated from trigonometry:

$$\theta = \arccos\left(\frac{a_z}{\sqrt{a_x^2 + a_y^2 + a_z^2}}\right), \quad (2.1)$$

which actually is the same as arc cosine from vector dot product:

$$\theta = \arccos((a_x, a_y, a_z) \cdot (0, 0, 1)). \quad (2.2)$$

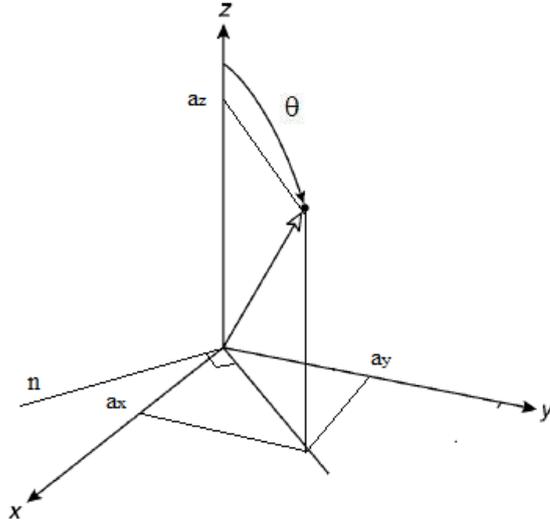


Figure 2.1. Visual representation of accelerometer data vector.

Rotation axis \vec{n} can be obtained as the vector cross product:

$$\vec{n} = (a_x, a_y, a_z) \times (0, 0, 1). \quad (2.3)$$

Components of rotation quaternion can be obtained as follow [64]:

$$\begin{aligned} q_0 &= a = \cos\left(\frac{\theta}{2}\right) \\ q_1 &= n_x b = n_x \sin\left(\frac{\theta}{2}\right) \\ q_2 &= n_y b = n_y \sin\left(\frac{\theta}{2}\right) \\ q_3 &= n_z b = n_z \sin\left(\frac{\theta}{2}\right). \end{aligned} \quad (2.4)$$

Rotation quaternion can be applied directly to transform 3D vectors, or used to construct a classic three by three rotation matrix:

$$R(q) = \begin{bmatrix} q_0^2 + q_1^2 - q_2^2 - q_3^2 & 2q_1q_2 - 2q_0q_3 & 2q_1q_3 + 2q_0q_2 \\ 2q_1q_2 + 2q_0q_3 & q_0^2 - q_1^2 + q_2^2 - q_3^2 & 2q_2q_3 - 2q_0q_1 \\ 2q_1q_3 - 2q_0q_2 & 2q_2q_3 + 2q_0q_1 & q_0^2 - q_1^2 - q_2^2 + q_3^2 \end{bmatrix}. \quad (2.5)$$

The matrix $R(q)$ represents a rotation that describes sensor orientation in relation to global reference frame. One must note that this rotation matrix includes only two degrees of freedom – roll and pitch, rotations along the third degree of freedom (often referred in literature as either direction, yaw or heading) can not be obtained from inclination measurements and is assumed to be zero in this model. Any vector describing surface segment can be simply transformed by multiplying it with $R(q)$:

$$\vec{v}' = R(q)\vec{v}, \quad (2.6)$$

where \vec{v} is the vector and \vec{v}' is the same vector transformed according to sensor orientation.

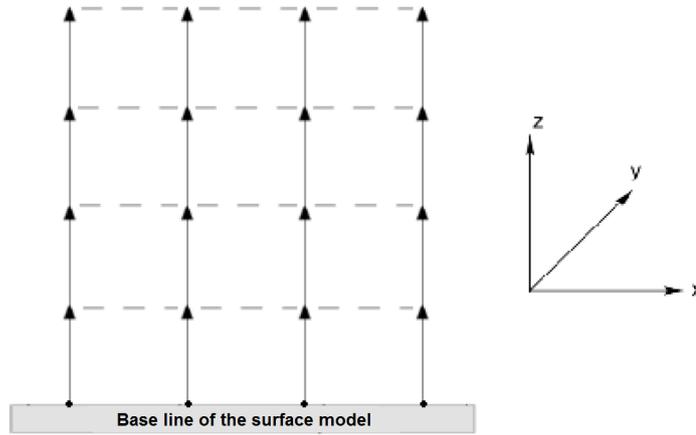


Figure 2.2. Structure of nominal surface model.

Surface segment vectors are in the nominal state without any rotation from reference position.

2.1.2. Shape reconstruction from discrete inclination values

A simplified surface shape model with one fixed side and without significant distortions such as stretching and folding along vertical axis is considered [15]. The model consists of multiple strips attached to the fixed base line of the surface model (Fig. 2.2). Every strip is constructed from several vectors representing surface segments, each starting from the end of the previous one. The structure of surface model corresponds to the sensor network arranged in a grid pattern, where the number of strips and the number of vectors on each strip correspond to the number of columns and rows in the sensor network. In the nominal state, when rotation matrix of each sensor from equation 2.5 is a unity matrix (no rotation), all segment vectors are in the reference position defined by vertical vector $(0; 0; l)$, where l is the distance between the sensors.

To reconstruct the shape of the actual sensor network the orientation of each sensor is calculated using equation 2.5. Then the obtained rotation matrices are used in equation 2.6 to transform the surface segments from the reference state to the actual orientation of the corresponding sensor. The reconstruction is started from the base line of the model that defines the origin of each strip of segment vectors. Then each vector is added to the end of the previous vector in the strip. The Fig. 2.3 depicts the structure of surface shape reconstruction model. The base line of the reconstructed model can be tied to one of the sensors. This provides that the base line is transformed according to sensor orientation allowing to dynamically change the position of the vector strip origins. This leads to a closer approximation of the original shape.

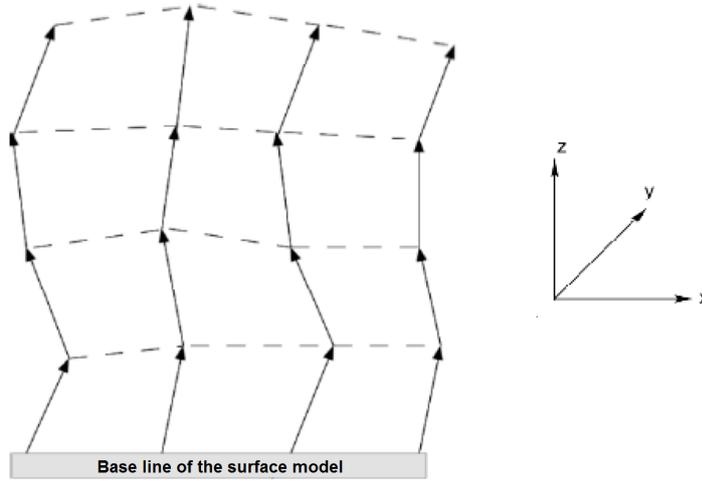


Figure 2.3. Structure of reconstructed surface model.

Surface segment vectors are rotated according to orientation of each corresponding sensor.

This approach has notable limitations arising from segment rotations around vertical axis. The orientation estimation that is based solely on one reference vector can provide information only regarding 2 degrees of freedom - rotation about two horizontal axes leaving the third degree of freedom undetermined. Because of this orientation of each sensor around z-axis is unknown and is assumed to be a fixed value when surface model is constructed. This simplification allows basic shape approximation, but can introduce significant errors in more complicated shapes.

To minimize this drawback, a method is proposed for minimization of segmented surface approximation error by selection of z-axis rotation angle for each segment if only inclination of each segment is known [14]. A surface consisting of I segments arranged in a $n \times m$ grid formation is assumed. This surface is continuous and bendable. It contains equal size segments arranged in an evenly spaced grid formation. The distance between segment rows and columns is denoted by d . Each segment in the model is represented as a cross-shaped object defined by four direction vectors: $\vec{N} = (0; 0; d/2)$, $\vec{E} = (d/2; 0; 0)$, $\vec{S} = (0; 0; -d/2)$, $\vec{W} = (-d/2; 0; 0)$ and center location vector \vec{C} (Fig. 2.4). These segments are used to construct approximated surface model.

Starting data $\vec{N}_i, \vec{E}_i, \vec{S}_i, \vec{W}_i$ for each $i \in [1..I]$ segments are calculated by rotating vectors $\vec{N}, \vec{E}, \vec{S}, \vec{W}$ to the inclination values of the corresponding segment. The actual segment z-axis rotation angle α_i and actual segment centre vector \vec{C}_i will be calculated by applying the method described below.

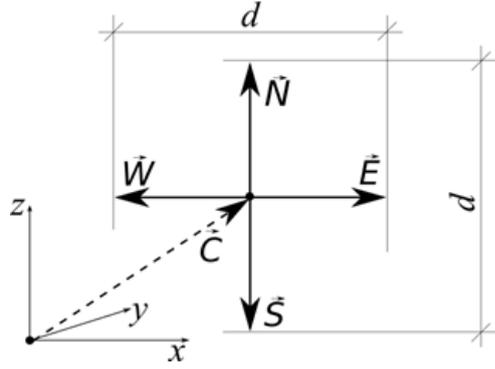


Figure 2.4. Structure of segment representation.

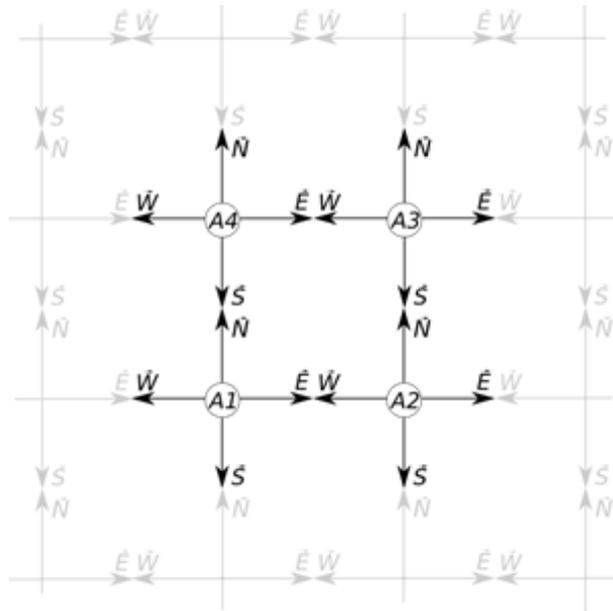


Figure 2.5. Structure of a work-set.

The feasibility of the proposed method is demonstrated with brute-force exhaustive search over all the possible rotation angles around z-axis for each of the segments. The regular grid is divided into $(n - 1) \times (m - 1)$ sub-problems hereinafter referred to as work-sets. Each of these work-sets consist of four segments $A1, A2, A3, A4$ which are connected in such a way, that without any initial rotation they would form a square as seen in Fig. 2.5. Each of the segments is represented in work-sets at least once. For the requirements of the algorithm a segment is considered processed when corresponding α_i and \vec{C}_i have been found. Initially for one of the segments these values are manually defined to serve as a reference for the rest of the model.

The algorithm consists of these main steps:

- Every work-set in the system is handled one at the time. The order of work-set selection is previously defined in such a way that every handled work-set contains at least one previously processed segment.
- While handling a work-set all the combinations of possible α_i values for non-processed segments within it are evaluated based on evaluation criteria explained below. The set of possible α_i values contains successive discrete angles from a certain interval. In this step \vec{C}_i values are calculated from selected α_i values together with values from previously processed segments.
- Those α_i and \vec{C}_i values that match the evaluation criteria the best are stored as the resulting values for corresponding segments. This means, that all segments in the current work-set are processed and algorithm moves to the next work-set until no more are left.

The structure of the algorithm is showed in Fig. 2.6.

To evaluate which of the combinations of possible z-axis rotation angles will improve the precision of the reconstructed model the most, constraints from the surface definition are applied during approximated model construction. The surface is defined as continuous, and because of that the corresponding sides of the segments should be joined together. Ideally there should be such a combination of α_i rotations for each segment, that all the segments can be connected as showed in Fig. 2.5 and their corresponding direction vector end points would coincide. Unfortunately this is not true because of the simplified grid model, however, the algorithm looks for the closest solution to ideal model, by introducing a continuity criterion working as follows:

- Within a work-set three of four pairs of segments are assumed to be jointed in “U” shaped pattern ($A4$ to $A1$, $A1$ to $A2$ and $A2$ to $A3$). Continuity criterion prefers solutions minimizing the distance Δ_d between the fourth pair of segments ($A3$ to $A4$).
- For each of four segments within a work-set and a given combination of α_i angles temporary endpoint vectors $\vec{N}'_i, \vec{E}'_i, \vec{S}'_i, \vec{W}'_i$ are calculated by rotating each of vectors $\vec{N}_i, \vec{E}_i, \vec{S}_i, \vec{W}_i$ by an angle α_i around z-axis.

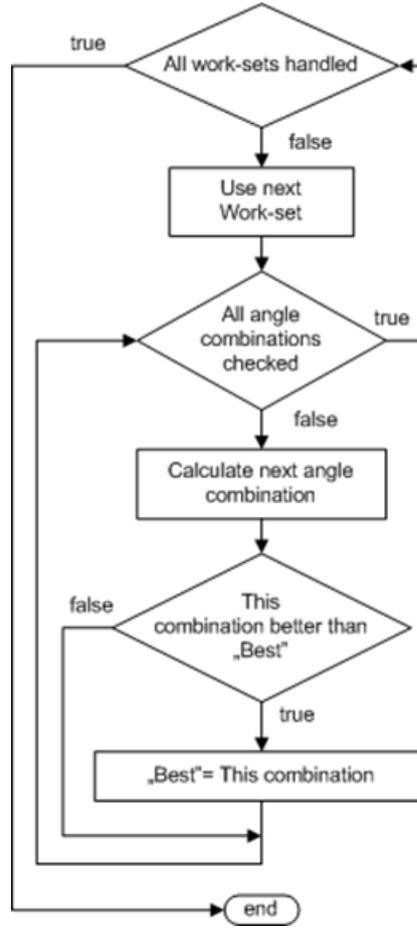


Figure 2.6. The flowchart of the brute-force algorithm.

- From these temporary points segment centers \vec{C}_i are calculated. For each work set at least one of the segments is already processed at most three of \vec{C}_i are unknown and can be solved from equation system:

$$\begin{cases} \vec{C}_{A1} + \vec{E}'_{A1} = \vec{C}_{A2} + \vec{W}'_{A2} \\ \vec{C}_{A1} + \vec{N}'_{A1} = \vec{C}_{A4} + \vec{S}'_{A4} \\ \vec{C}_{A2} + \vec{N}'_{A2} = \vec{C}_{A3} + \vec{S}'_{A3} \end{cases} \quad (2.7)$$

- For this combination the distance Δ_d is then calculated:

$$\Delta_d = \|(\vec{C}_{A3} + \vec{W}'_{A3}) - (\vec{C}_{A4} + \vec{E}'_{A4})\|. \quad (2.8)$$

- Δ_d represents the matching error between segments so the combination with the lowest Δ_d is considered the best.

After applying the continuity criterion several valid solutions are still possible because cases are possible with infinite number of solutions, for example, when two of the connection points

between segments are both on a line parallel to z-axis. Also, α_i angles are changed by a discrete step. Because of that, continuity criterion must be loosened by stating that all combinations with Δ_d that differ from the best Δ_d by an experimentally determined step ϵ_d are also considered as possible candidates for the best solution.

Because of this, another criterion must be introduced to select the most probable continuous surface configuration of the possible candidates. It is assumed, that the starting configuration of the surface is a flat surface and all deformations require additional energy. For the purposes of this work it is assumed that states which require less energy are more probable than those requiring more energy. In essence it means that the algorithm prefers flatter surfaces. This is provided by the second - flatness criterion. For all the solution candidates after continuity criterion is applied the distance between the segment centers is calculated as:

$$F = \|\vec{C}_{A1} - \vec{C}_{A2}\| + \|\vec{C}_{A2} - \vec{C}_{A3}\| + \|\vec{C}_{A3} - \vec{C}_{A4}\| + \|\vec{C}_{A4} - \vec{C}_{A1}\|. \quad (2.9)$$

The solution with largest the F value is selected as the most flat of all solution candidates.

These two criteria are sufficient to select the preferred solution. It must be stated that this proposed improvement has limitations in general case when inclination and restricted model does not provide enough information for selection of segment vertical rotations, for example, in case of vertically curved surface. Despite this, in several cases significant improvements can be provided. Experimental evaluation of the proposed methods are presented in Subsection 4.1.

2.2. Shape sensing based on accelerometer and magnetometer modules

In previous subsection a particular limitation of shape sensing system regarding surface segment orientation estimation using a single reference vector measurement was stated. To overcome this problem sensors that can measure another reference vector can be used in order to obtain full three degrees of freedom orientation estimates, thus leading to method for detailed free-form shape detection. This subsection is dedicated to detailed study of shape reconstruction methods that utilize measurements of two reference vectors. Two reference vectors are obtained by measuring Earth's gravity and magnetic field directions.

2.2.1. Orientation estimation using local gravity and magnetic field vectors

To recover a 3D shape with embedded 3-axis acceleration and magnetic sensor grid, first the relative surface segment orientations, which correspond to their sensor orientation, have to be obtained. Due to the reconstruction challenges discussed in Subsection 2.2.2, sensor layer in material can require a large number of acceleration/magnetic sensor nodes, requiring as much as several hundred orientation calculations for each reconstructed surface frame. This implies that it is necessary to use an effective algorithm to minimize the computational burden.

In general, the Euler's theorem states that any rigid body orientation can be described by a single rotation around fixed axis. This implies that any sensor orientation can be defined as a single rotation of sensor reference frame relative to some global reference frame and can be characterized with orthogonal rotation matrix R . The problem of finding R is discussed a lot in spacecraft attitude estimation relative to some general reference frame, where R can be found using observations of vectors which are known in general reference frame, such as direction of the Sun and stars [65, 66, 67, 68]. The same problem can be assigned for any rigid body orientation estimation, in this case determining orientation of the sensor node which can measure Earth's gravity and magnetic field vectors in its reference frame. It must be noted that acceleration sensor measurements are suitable for sensor orientation determination only in static or close to static conditions (dynamic acceleration $\ll g$). In Subsections 2.3 and 4.3 a few possibilities for dynamic performance enhancement are discussed.

There are a number of algorithms proposed for orientation determination. The general problem of orientation determination can be stated as Wahba's problem [69], which seeks the R as the solution for minimization of expression [70]:

$$\sum_{k=1}^K \|v_k^* - Rv_k\|^2, \quad (2.10)$$

where $\{v_1, v_2, \dots, v_K\}$ and $\{v_1^*, v_2^*, \dots, v_K^*\}$ are sets of K vector observations in object and general reference frames respectively. Acceleration/magnetic sensors provide only two vector observations at each system state, which is minimum for full orientation determination with deterministic approach. Because of this no real minimization problem can be defined. Several algorithms exist that can obtain orientation from two vector measurements. TRIAD [70] is one of the fastest, singularity free and computationally simple deterministic algorithms for orientation estimation. Authors in [71] propose an algorithm that obtains the same orientation data as TRIAD, but the

output is in quaternion form, however, the required computations are more complex. Due to simpler calculations and no direct requirement for quaternion orientation estimation the TRIAD algorithm was preferred.

TRIAD constructs two triads of orthonormal unit vectors, one triad is formed from two non-parallel normalized vectors in general reference frame, the other is formed from the same vector measurements in sensor reference frame. If we denote Earth's gravity field vector E_g , magnetic field vector E_m , sensor measurement of gravity field vector S_g and sensor measurement of magnetic field vector S_m (all vectors normalized), then the triad in Earth reference frame is constructed as follows:

$$e_1 = E_g; \quad (2.11)$$

$$e_2 = \frac{E_g \times E_m}{|E_g \times E_m|}; \quad (2.12)$$

$$e_3 = e_1 \times e_2, \quad (2.13)$$

and the triad in sensor reference frame:

$$s_1 = S_g; \quad (2.14)$$

$$s_2 = \frac{S_g \times S_m}{|S_g \times S_m|}; \quad (2.15)$$

$$s_3 = s_1 \times s_2. \quad (2.16)$$

These triads are then used to form matrix for global Earth reference:

$$M_e = [e_1 \quad e_2 \quad e_3], \quad (2.17)$$

and matrix for sensor measurements:

$$M_s = [s_1 \quad s_2 \quad s_3]. \quad (2.18)$$

Then the rotation matrix R which describes the sensor orientation relative to global reference frame can be calculated by:

$$R = M_e M_s^T. \quad (2.19)$$

The rotation matrix R now can be used to calculate surface segment orientation relative to initial position that corresponds to the sensor orientation relative to Earth reference frame. Any vector describing surface segment can be simply transformed by multiplying it with R :

$$\vec{v}' = R\vec{v}, \quad (2.20)$$

where \vec{v} is the vector and \vec{v}' is the same vector transformed according to sensor orientation.

2.2.2. Shape reconstruction from sensor orientation data

A few authors have discussed the problem of reconstruction of free-form 3D shape from full three degree of freedom local orientation data. In [51, 53, 52, 50] authors have approached the problem by treating orientation information as local tangent data of the surface. This tangent data is obtained from sensor ribbons which are able to reconstruct their own shape as a 3D curve [49]. Multiple such sensor ribbons are applied to the surface allowing to obtain geodesics of the surface and thus enabling reconstruction of the global surface geometry.

Ribbons are equipped with sensor nodes capable of obtaining their orientation relative to Earth's gravity and magnetic fields. Orientation data is used to obtain local tangent information which describes the derivative of the curve [49]. Therefore, the output of the sensor ribbon is discrete samples of the curve derivatives. The essence of the reconstruction problem is that the spatial locations of derivative samples are not known. Only the derivative samples and the curvilinear distances (arc lengths) between them are known, which are the distances between the sensor nodes on flat ribbon. If a curve $U(l)$ is parameterized by an arc length parameter l with known derivatives $U'(l)$ at points $l = p_k$ ($k = 1, \dots, n$), where n is the total number of sensor nodes, this relation can be defined with following equation:

$$L = \int_{p_k}^{p_{k+1}} \|U'(l)\| dl, \quad (2.21)$$

where $L = p_{k+1} - p_k$ is the distance between the sensor nodes assuming even sensor distribution. Preserving relation in (2.21), first the discrete derivative samples are interpolated with natural cubic splines for 2D curves or cubic splines on unit sphere for 3D curves to obtain continuous derivative function $U'(l)$. Then this function is numerically integrated to obtain the curve solution. Due to requirement for the interpolation and numerical integration, the method requires considerable computing power.

More detailed analytic solution for curve reconstruction from orientation data is proposed in [72]. Here the interpolation of discrete derivatives of the curve is done with spatial Pythagorean-Hodograph quintic splines. It is stated that the reconstruction is closer to the original curve, compared to natural cubic spline reconstruction. However, the method requires extensive iterative computations, and in practical implementation with 3 GHz PC the reconstruction time of a curve from 8 points (a ribbon with 8 sensor nodes) was reported to be 0.28 s with initialization of

interpolation coefficients, and 2.25 s without it (see details in [72]). This is a serious limitation for applications in real-time low-power systems.

In either case the methods have a limited resolution as they can not reconstruct distortions with a period less or equal to the distance between the sensors. Therefore, high sensor density can noticeably rise the performance of shape reconstruction, thus systems with large sensor quantity (potentially several hundred) should be considered to achieve high resolution for detailed shape measurements. This requires very efficient reconstruction algorithms applicable for data processing of large sensors networks. This is particularly important to potential applications in mobile cyber-physical systems such as wearable systems, flexible electronics, etc., that require real time performance with limited computing power.

2.2.3. Proposed method

Based on the previously described approach an algorithm was designed with the emphasis on application of as many sensors as possible, to reduce the limited resolution problem [10, 11]. Instead of detailed interpolation and integration, it is assumed that sensors are attached to a rigid mutually connected segments of the surface. This provides coarser approximated model compared to approach from Subsection 2.2.2, but noticeably decreases computing time. Also, the approach allows fast shape reconstruction from a large number of sensors. This can provide a higher sensor density thus increasing the resolution and overall performance of shape reconstruction.

Acceleration/magnetic sensor nodes are arranged in a regular grid along the surface. The model of the surface is divided in n rigid segments, where $n = i \cdot j$ is the total number of sensors used, so that the segment structure corresponds to the structure of the sensor grid (i and j denote row and column of sensor location in the grid). Each segment is described by four direction vectors, denoted by $\vec{N}[i, j]$, $\vec{E}[i, j]$, $\vec{S}[i, j]$ and $\vec{W}[i, j]$ and segment center point $C[i, j]$. The segment center points are the surface control points which will define the surface geometry. Initially all segments are aligned with the global reference system by assigning some base direction vector values such as:

$$\begin{aligned}
 \vec{N}_b &= [0; 0; \frac{L_1}{2}] \\
 \vec{E}_b &= [\frac{L_2}{2}; 0; 0] \\
 \vec{S}_b &= [0; 0; -\frac{L_1}{2}] = -\vec{N}_b \\
 \vec{W}_b &= [-\frac{L_2}{2}; 0; 0] = -\vec{E}_b,
 \end{aligned} \tag{2.22}$$

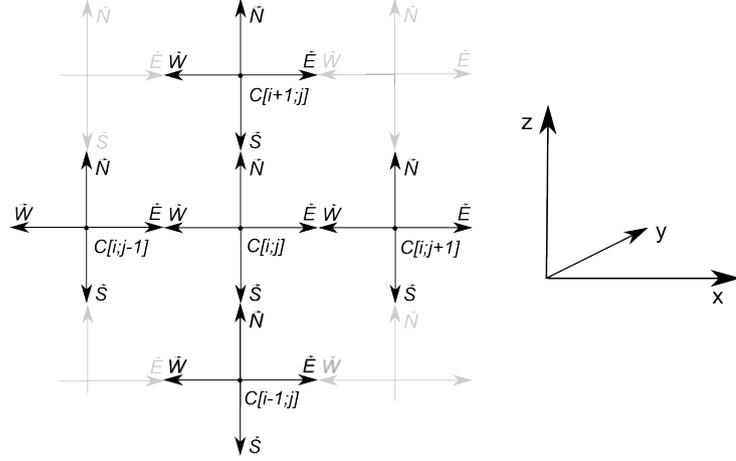


Figure 2.7. Surface segment structure.

Each segment consists of center C and four direction vectors \vec{N} , \vec{E} , \vec{S} and \vec{W} .

where L_1 is the distance between sensors lengthwise and L_2 is the distance between sensors across in the actual grid. In Fig. 2.7 the structure of the surface model can be seen.

During shape reconstruction the base direction vectors of each segment are translated according to the corresponding sensor orientation. By inserting segment direction vectors in the equation (2.20), actual orientation of segment can be obtained:

$$\begin{aligned}\vec{N}[i, j] &= R_{ij}\vec{N}_b \\ \vec{E}[i, j] &= R_{ij}\vec{E}_b,\end{aligned}\quad (2.23)$$

where R_{ij} is a rotation matrix that describes the corresponding sensor orientation and is obtained as showed in Subsection 2.2.1. The rest of the direction vectors can be obtained simply as an inverse:

$$\begin{aligned}\vec{S}[i, j] &= -\vec{N}[i, j] \\ \vec{W}[i, j] &= -\vec{E}[i, j].\end{aligned}\quad (2.24)$$

From the segment structure (Fig. 2.7) it can be deduced, that if a single control point location is known, then any other control point on the same segment row or column can be calculated by adding and subtracting the corresponding segment direction vectors. If an arbitrary sensor in i_{ref} row and j_{ref} column is defined as reference by assigning some constant value to $C[i_{ref}; j_{ref}]$, then control points on the reference column ($j = j_{ref}$) can be obtained with expressions:

$$\begin{aligned}C[i; j_{ref}] &= C[i_{ref}; j_{ref}] + \\ &+ \sum_{k=i_{ref}}^{i-1} (\vec{N}[k, j_{ref}] - \vec{S}[k+1, j_{ref}])\end{aligned}\quad (2.25)$$

if ($i > i_{ref}$)

and

$$\begin{aligned}
C[i; j_{ref}] &= C[i_{ref}; j_{ref}] + \\
&+ \sum_{k=i}^{i_{ref}-1} (-\vec{N}[k, j_{ref}] + \vec{S}[k+1, j_{ref}]) \\
&\text{if } (i < i_{ref}).
\end{aligned} \tag{2.26}$$

Similarly it is possible to obtain control points on the reference row ($i = i_{ref}$):

$$\begin{aligned}
C[i_{ref}; j] &= C[i_{ref}; j_{ref}] + \\
&+ \sum_{k=j_{ref}}^{j-1} (\vec{E}[i_{ref}, k] - \vec{W}[i_{ref}, k+1]) \\
&\text{if } (j > j_{ref})
\end{aligned} \tag{2.27}$$

and

$$\begin{aligned}
C[i_{ref}; j] &= C[i_{ref}; j_{ref}] + \\
&+ \sum_{k=j}^{j_{ref}-1} (-\vec{E}[i_{ref}, k] + \vec{W}[i_{ref}, k+1]) \\
&\text{if } (j < j_{ref}).
\end{aligned} \tag{2.28}$$

$$\phi(s_i) = \sum_{k=s_0}^{s_i} (\phi'(k) * L) \tag{2.29}$$

After the first row and column are obtained, there is at least one control point with known coordinates on each other row and column in the grid. Any of the unknown control points now can be calculated using the known control point in either its row or column as reference. In theory, both cases should provide equal outcome, however, the results tend to differ depending on the chosen connection path, due to the finite number of sensing elements in the grid and the sensor measurement noise. To overcome this problem the control point recovery follows bilateral process. First, according to structure in Fig. 2.8 (a) each segment center coordinate is calculated from reference by finding one reference row with equations (2.27) and (2.28) and then connecting other segment direction vectors lengthwise using (2.25) and (2.26). Then similarly according to structure in Fig. 2.8 (b) all control points are obtained again by obtaining one reference column with equations (2.25) and (2.26) and then connecting segments across using (2.27) and (2.28). In the end averaged values from both results are used as a surface control point coordinates to form closed grid similarly as in [51].

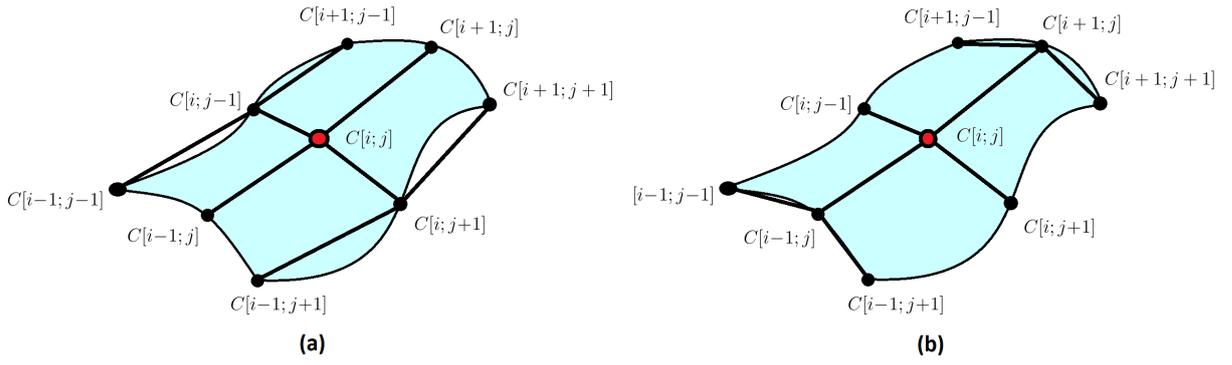


Figure 2.8. Structure of control point connections.

$C[i; j]$ - reference point. (a) single reference row is obtained, then all other points are calculated with column method. (b) single reference column is obtained, then all other points are calculated with row method.

The obtained control points can be seen as 3D point cloud defining the shape of the body similar to a 3D scanner. The full shape model is constructed by drawing a quad-mesh structure through control points that coarsely approximates the surface shape. If high sensor density is used, this method can provide reasonably smooth models. In addition, more advanced algorithms can be used providing smoother and more accurate shape fitting on control points, if limited computing power is of no concern [73].

2.2.4. Simulations

To validate the performance of the shape approximation with the modified approach, the proposed method was compared with the previous studies. For simplification purposes only 2D case is analyzed. In 2D proposed method is very similar to the 3D case. The curve is divided in segments according to the number of the sensors. Each segment now is described by only two mutually opposite direction vectors. Length of each vector is half the distance between sensors along the curve. The translation of the direction vectors according to corresponding orientation is done with 2D rotation matrix.

A complex curve similar as in [49] was synthesized. The curve is defined as:

$$\begin{aligned} x &= 10 \cos\left(2\pi t + \frac{\pi}{8}\right) \sin^2(2\pi t) \\ y &= 10 \cos^2(2\pi t) \sin\left(2\pi t + \frac{\pi}{8}\right), \end{aligned} \tag{2.30}$$

where $t \in [0; 1]$. The curve was reconstructed from 30 simulated sensor locations with the proposed method and the method described in [49]. The original and reconstructed curve as well

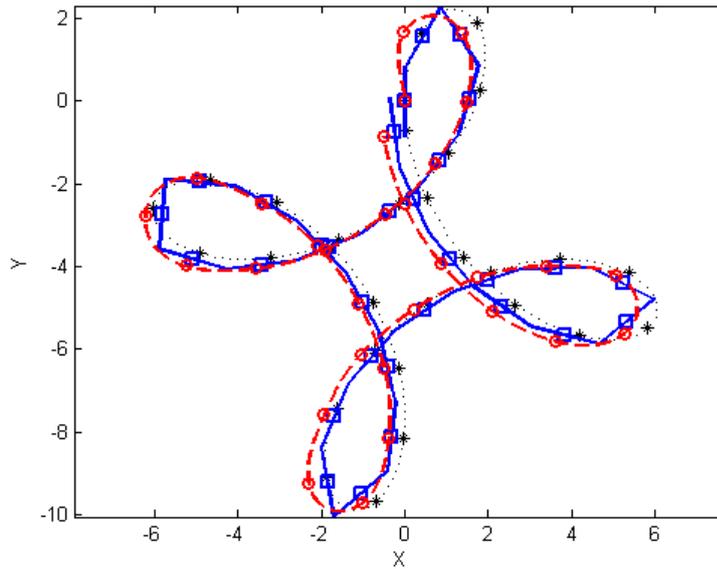


Figure 2.9. Comparison of reconstruction methods.

Synthesized curve is thin dotted line. Simulated sensor locations on the curve are denoted by star. Reconstruction with method from [49] is indicated with dashed line, sensor locations are denoted by circles. Reconstruction with our method is indicated by solid line, sensor locations are denoted by squares.

as sensor locations can be seen in Fig. 2.9. The average Euclidean distance between reference and reconstructed sensor locations relative to curve length with proposed method and method in [49] was 0.49% and 0.31% respectively. The simulation was done in MATLAB on 3.2 Mhz dual core PC and the calculation times were measured for the proposed method 0.72 ms and 30.05 ms for the method in [49]. It must be noted, for each method reconstruction accuracy can change depending on the measured shape, sensor density and phase of sensor distribution relative to the curvatures. In some cases the proposed method provided even better results.

In addition to approximation errors described previously, there are a number of different error sources that arise from measurement process. These errors fall in two major categories - sensor measurement errors and sensor mechanical mounting errors. The sensor measurement errors include calibration errors (scale and offset), magnetic perturbations, dynamic accelerations and sensor noise, which introduce errors in Earth's gravity and magnetic field vector component measurement. The sensor mechanical mounting errors include orientation errors, which introduce misalignment of sensor reference frames relative to measured object reference frame, and placement errors, which introduce differences in inter sensor distances leading to orientation measurement in incorrect place on the curve.

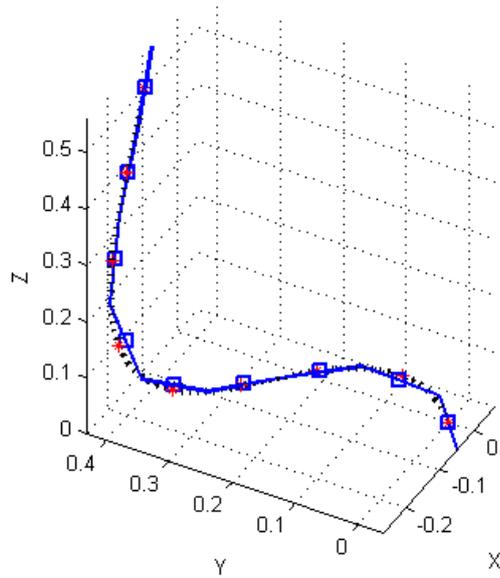


Figure 2.10. Synthesized 3D curve.

Synthesized curve is dotted line. Simulated sensor locations on the curve are denoted by stars. Reconstructed curve is solid line, sensor locations are denoted by squares.

To estimate the impact of each error source on shape reconstruction accuracy, nine sensor nodes evenly distributed along 3D curve were simulated, allowing to obtain theoretical acceleration and magnetic sensor readings (Fig. 2.10). Before reconstruction the theoretical acceleration and magnetic sensor data were deteriorated with different errors with increasing magnitudes to observe impact on reconstruction accuracy.

Sensor measurement errors were simulated by adding the white noise with zero mean and increasing standard deviations to x , y and z components of gravity and magnetic field vector measurements. The sensor mounting location errors were simulated by adding the white noise with zero mean and increasing standard deviations to previously set sensor locations along the curve. Sensor mounting orientation errors were simulated by rotating theoretical acceleration and magnetic sensor readings in random direction by rotation angle generated as white noise with zero mean and increasing standard deviations.

For each error source with each standard deviation 1000 reconstructions were simulated. Mean values of average reconstruction errors are depicted in Fig. 2.11. Experimental setup for validation of proposed method is described in Subsection 4.2. Further analysis and discussion of the errors is done in Subsection 2.3 and Subsection 4.3.

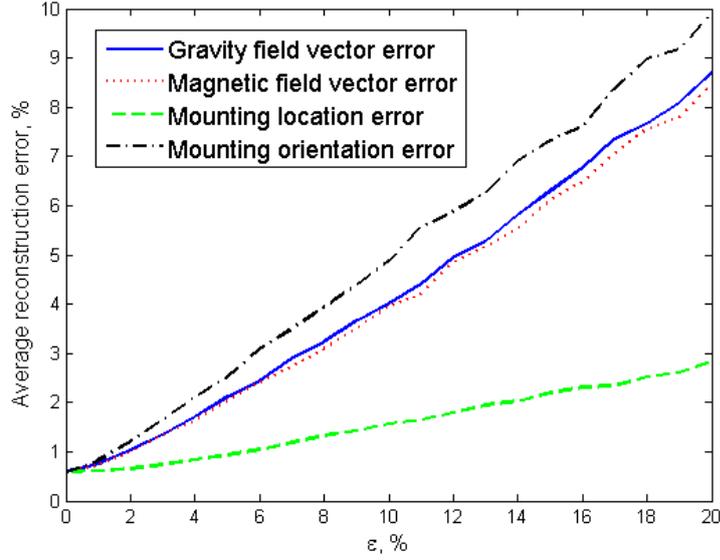


Figure 2.11. Impact of various error sources.

For each curve ε denotes value of simulated error standard deviation relative to: gravity field vector length (gravity field vector error); magnetic field vector length (magnetic field vector error), distance between sensors (mounting location error), π (mounting orientation error). The average curve reconstruction error using undeteriorated data (approximation error) was 0.59% relative to curve length.

2.3. Conclusions

In Subsection 2.1 theoretical description for shape sensing based on 3-axis acceleration sensors is provided. The main advantage of the method is the relatively simple sensor network structure that utilizes only one type of sensors. Also, the data processing algorithm that obtains two degrees of freedom orientation of each surface segment and applies it to simple surface model constructed of 3D vectors is quite simple and effective. This model is sufficient for basic shape monitoring, however, still a major source of shape reconstruction errors are the segment rotations around the vertical axis which are not measured.

This problem is minimized with the proposed method for selection of z-axis rotation angle for each segment if only inclination of each segment is known. Despite potentially providing more accurate model, the method also significantly increase complexity of data processing algorithm. Also, the method still does not provide a solution in general case as the ability to determine segment rotation along vertical axis is dependent on inclination. However, more simple shape acquisition is still feasible, for example, in applications such as posture monitoring, where a relatively small amount of deformations is expected.

In contrast, Subsection 2.2 provides description of shape sensing method that is based on full 3 degree of freedom orientation measurements obtained from accelerometer and magnetometer data in deterministic way. It is demonstrated that the shape approximation approach with finite length vectors compared to previously proposed detailed interpolation and integration of local derivatives provides slightly lower reconstruction accuracy when using same number of sensors, however, in Matlab simulation reconstruction is done more than 40 times faster, which is a noticeable difference and enables processing of more sensor data. Due to the inability of interpolation algorithms to estimate deformations with period smaller or equal to the distance between sensors, the density of the sensors has to be increased to obtain higher shape sensing resolution and accuracy. Therefore, it seems that for practical real-time shape sensing systems complicated reconstruction algorithms can be substituted with a simplified approximation in favor of faster processing time that allows higher sensor density. The clear limitation of the method is sensor spatial density which has to be as high as possible, because it directly contributes to the resolution of shape reconstruction.

3. HARDWARE ARCHITECTURE FOR SHAPE SENSING SENSOR NETWORK

Inertial/magnetic sensor network for shape sensing can potentially have several hundred sensors integrated in a compact environment to provide high resolution of shape measurements. While it may seem that wireless data transfer solutions are most convenient, as they reduce practical wiring problems, there are many drawbacks and open research issues, such as limited bandwidth, power supply, additional cost and space for radio hardware, security, etc. [74, 75]. For applications such as shape sensing the effectiveness of wireless networking is even more questionable because of the requirement of high sensor density and previously mentioned drawbacks. Due to this fact there are studies which discuss different wire based data acquisition and communication architectures for small scale sensor networks [76], however, they mainly consider a limited number of sensors.

Data acquisition from shape sensing sensor network is a challenging task as there is a large number of sensors and there are specific requirements for data transfer interface such as simple wiring structure to reduce amount of wires, limited power consumption, reasonable data transfer speed to provide required sampling rate and limited size of hardware implementation. Conventional data transfer interfaces available in current low-power microelectronics are not directly suitable for this task, also, existing solutions for large scale networks with advanced addressing and routing cannot be applied effectively due to limited resources. Because of this, an alternative solution has to be used. For data acquisition a custom network interface was developed. The method supports up to several hundred sensors and can be implemented using conventional low-cost hardware with simple four wire serial connection.

3.1. State of the art

There are several conventional communication interfaces available for data transfer between two or more IC. Each of them have different electrical and physical parameters. Even though, theoretically most of them provide the ability to create multi-device networks, in practice none of them are directly suitable for data acquisition from sensor network with a large number of nodes.

UART (Universal Asynchronous Receiver Transmitter) is a serial interface that provides full-duplex communication between two devices [77]. This interface provides only point-to-

point communication so it is unsuitable for application in larger networks. In addition, this interface is asynchronous, which has higher requirements for oscillator precision at faster data transfer rates and also makes sensor synchronization difficult.

I2C (Inter-Integrated Circuit) is a serial interface that supports half-duplex communication between devices. All devices are connected to the same bus and software addressing is used to determine the device to communicate with. In practice many devices have a single factory hardcoded address for I2C communication which makes impossible to connect multiple similar devices to the same bus without additional hardware [78]. Also, maximum number of devices on the bus is limited not only by the number of available addresses, but also by maximum bus capacitance.

CAN (Controller Area Network) is a serial interface that is mainly designed for reliable communications in error critical applications such as automotive [79]. This interface is very robust, however, data transfer has large overhead and is relatively complicated, therefore, it is not suitable for low-power applications.

SPI (Serial Peripheral Interface) is a serial interface that supports full-duplex communication in master/slave mode [80]. This interface supports baudrates up to several megahertz and requires no data overhead for device addressing, which provides relatively fast data transfer speeds. In the classic mode all devices are connected to a three line bus. Two lines of the bus are for data transfer to and from master respectively, third line is for data transfer synchronization with clock signal. In addition, each slave is connected to the master with an individual source select line that is used for slave addressing. Hence, for each additional device in the network an additional wire is required. This significantly complicates the implementation of network with a large number of devices.

In a “daisy-chained” SPI mode this drawback can be overcome by connecting devices and organizing data flow in a serial manner, thus removing the individual addressing (Fig. 3.1). Each slave is connected to a data line in series instead of connecting in parallel to a single bus. Data output port (SOMI) of each device is connected to data input port (SIMO) of the next device [80]. Clock signal line (CLK) is still shared as a single bus, and also all source select lines (SS) are connected to single line. With this wiring method SPI slaves can act as simple shift registers. Data is sent from one slave to next till master have acquired data from each slave device. This mode can only be used in applications where fast data acquisition from single individual slave

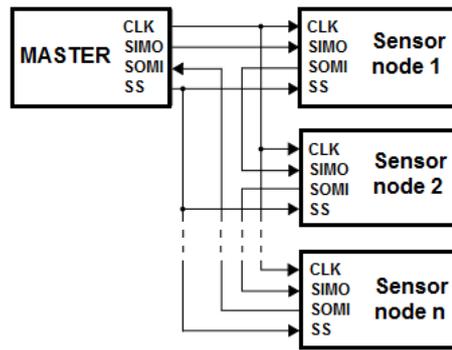


Figure 3.1. Daisy-chained SPI structure.

device is not necessary, however, it can be very valuable due to its simple structure, small amount of wires and reasonable data transfer speed.

A significant drawback for daisy-chained SPI is the clock signal line (CLK) which is shared by all sensor nodes in the network and thus must be as long as the sensor chain itself. Even if the sensors are placed only couple of centimeters apart, chain with 100 sensor nodes will be several meters long. This results in a serious clock signal distortion for higher frequencies as the resistance and parasitic capacitance of wire together with sensor node input port capacitances forms an RC circuit with significant time constant.

3.2. Enhanced daisy-chained SPI

As mentioned in subsection 3.1, SPI in daisy-chain mode provides a number of benefits. It is reasonably fast and is based on a relatively simple wiring scheme. The network can be easily expanded, as additional sensors can be simply connected at the end of the chain and require only some minor changes in system software setup. In addition, the number of wires can be reduced even further, by removing the source select line and determining the duration of data transfer by programming slave devices to do a certain amount of byte transfers before doing other tasks.

To overcome the problem with the clock signal distortion an enhanced daisy-chained SPI is proposed [12]. Instead of connecting each slave to the clock line in parallel, the clock signal is repeated in every slave (Fig. 3.2). Signal repeater acts as a buffer for each stage of the clock line, thus removing the effects from the rest of clock circuit. Fig. 3.3 illustrates structure of each daisy-chained SPI slave with a clock signal repeater. SIMO and SOMI are data input and output lines respectively, CLK is the clock signal coming from master or previous slave and

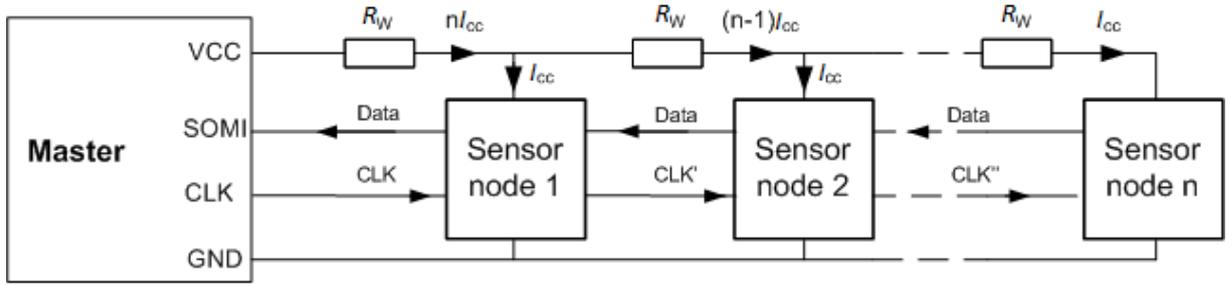


Figure 3.2. Structure of enhanced daisy-chained SPI network.

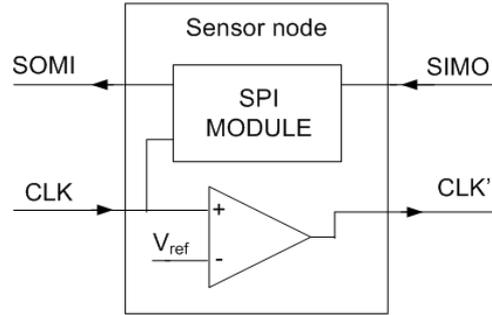


Figure 3.3. Structure of enhanced daisy-chained SPI slave.

CLK' is repeated clock signal going to next slave. This solution reduces the network to multiple relatively simple and independent links between two adjacent sensor nodes in chain. The whole network can be formed with only 4 wires in total: 2 communication lines which connect sensors in series and 2 parallel power supply lines.

The clock signal repetition can be easily achieved with an analogous comparator. In addition many low-power mixed signal microcontrollers provide integrated analogous signal comparators, thus the clock repetition can be provided in sensor node without any additional hardware components. However, as clock signal has non-ideal edges, the pulse width of the repeated signal can vary depending on comparator reference voltage V_{ref} . In Fig. 3.4 a clock signal positive pulse shape is illustrated. T_p is the incoming clock signal positive pulse width. If the comparator response time is neglected, then with comparator references V_{ref1} and V_{ref2} the repeated signal positive pulse widths will be T_{ref1} and T_{ref2} respectively. As it is visible in the Fig. 3.4, the incoming clock signal positive pulse width T_p will be changed depending, on the reference used:

$$T_{ref1} < T_p < T_{ref2} \quad (3.1)$$

After multiple signal repetitions the signal gets too distorted for SPI module to work properly. In theory it is possible to find such V_{ref} value, that the positive clock signal pulse width stays

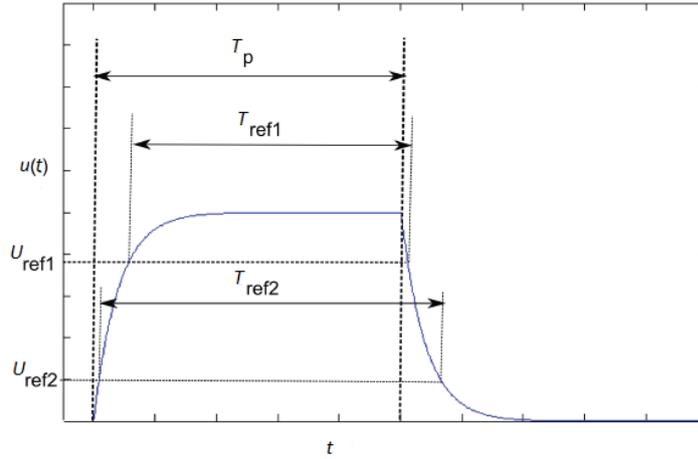


Figure 3.4. Clock pulse width variance with different comparator reference signals.

intact after repetition, although, it would require manual calibration for each sensor node. An alternative solution is to use two V_{ref} values - one which narrows and one which expands the pulse width of repeated clock signal. At this case an automatic calibration can be introduced which measures the incoming clock signal positive pulse width, and decides which reference to use depending whether signal positive pulse width needs to be expanded or narrowed.

Enhanced daisy-chained SPI data transfer does not require sensor addressing and the clock repetition method provides undistorted clock signal throughout the network. Based on this there is no limitation of maximum amount of sensor nodes from network topology point of view. However, in practice there are two other properties which limit the maximum number of sensor nodes – sensor supply voltage drop on power lines and the required sampling frequency. The most convenient way to supply each node is with common lines from the master (Fig. 3.2). The wire resistance and sensor node supply currents cause increasingly reduced voltage on power supply lines. If we assume that each sensor node consumes an equal amount of current and there is equal wire resistance between every two sensor nodes, then the supply voltage drop on last sensor in the chain can be calculated as:

$$V_{\Delta} = \sum_{k=1}^n k I_{cc} R_W, \quad (3.2)$$

where n – total number of sensor nodes in the network, I_{cc} – current consumption per sensor node [A], R_W – wire resistance between two sensor nodes [Ω].

Supply voltage on the last sensor can simply be calculated by:

$$V_n = V_{cc} - V_{\Delta} = V_{cc} - \sum_{k=1}^n k I_{cc} R_W, \quad (3.3)$$

where V_n – supply voltage on last sensor node [V], V_{cc} –master power supply voltage [V]. V_n must always be greater than minimum rated supply voltage for any active component in the sensor node, thus equation 3.3 can be used for estimation of maximum number of sensors for particular wire and sensor node parameters.

The other property that limits maximum number of sensors is the application required sampling rate of each sensor in network. The sampling rate depends on the total number of sensor nodes in the network, the total number of bytes to receive from each sensor node and the SPI clock frequency. If we denote the number of bytes to receive from each sensor by b , then theoretical minimum sampling period of whole sensor network using SPI devices with 8 bit shift registers can be calculated:

$$T_{sample} = \frac{8bn}{f_{CLK}}, \quad (3.4)$$

where T_{sample} – minimum sampling period [s], f_{CLK} – SPI clock signal frequency [Hz], n – number of sensor nodes in network. In practice there is some additional time interval for data preparation in sensor node, also, there is SPI hardware related delay between each consecutive byte transfer. Both of these have to be taken into account and can considerably reduce maximum sampling rate. Total sampling period can then be calculated:

$$T_{sample} = bn\left(\frac{8}{f_{CLK}} + T_{SPI}\right) + T_{data}, \quad (3.5)$$

where T_{data} – delay for data preparation in sensor node [s], T_{SPI} –SPI hardware related delay. The experimental validation of the performance of the proposed method is described in subsection 4.2.2.

3.3. Conclusions

The proposed method can provide efficient data acquisition of multi-sensor network by connecting devices in line topology. This allows to overcome addressing problems which occur in networks with large number of nodes. The method also reduces the number of wires and provides simple wiring structure for convenient integration into a garment or similar medium. The simple wire structure also enables this architecture to be conveniently used with new e-textiles

that substitute wires with conductive yarns [81]. At the same time method only requires conventional low-power hardware and provides reasonable data transfer speeds for real-time systems, which are essential aspects in number of applications.

There are few drawbacks of the proposed method. The inability to individually address specific sensor might introduce difficulties in some applications. Another considerable drawback is that each sensor node in the system is a single point of failure. If a sensor node goes out of order, the communication is also lost with all other nodes connected in the network after the damaged one. Also the addition of a microcontroller for each sensor noticeably increases the sensor network overall power consumption as the microcontroller can require more power than the sensor itself, however, with modern low-power devices this drawback can be endured.

Finally, it can be appended that the enclosed microcontrollers can not only enable convenient solution for data acquisition, but also provide some additional features. These computational components can be used to create systems with distributed computing architecture. Each sensor node instead of directly sending raw sensor data can perform some data processing to reduce the size of the data to be transferred and reduce the computing power requirements for the central processing unit in the system. This could provide significant benefits for real-time, low-power applications in mobile CPS.

4. EXPERIMENTAL SYSTEMS

A number of experimental systems were designed to demonstrate proof-of-concept as well as to evaluate the real world performance of the new methods proposed in this work. In Subsection 4.1, an experimental system and testing of 3-axis accelerometer network with proposed shape sensing algorithms is described. In Subsection 4.2 an experimental system and testing of 3-axis accelerometer and magnetometer system is described including the evaluation of orientation estimation performance with low-cost acceleration/magnetic sensor modules, hardware architecture of sensor network proposed in Section 3 and overall shape reconstruction performance.

4.1. Surface reconstruction with accelerometer network

To validate the methods of surface shape reconstruction from accelerometer sensor data described in Subsection 2.1, a prototype device was built consisting of 3-axis accelerometer network and data acquisition board [15, 16]. ST Microelectronics LIS331DLH [82] MEMS digital output 3-axis accelerometers were used because of their relatively small size ($3 \times 3 \times 1$ mm) providing 3-axis sensor in single package, high resolution (12 bit over $\pm 2g$), convenient data acquisition through digital SPI interface that reduces the impact of noise interference on analog signal transmission lines. Each accelerometer is mounted on a custom made circuit board with dimensions less than 2×2 cm (Fig. 4.1). A network of 16 accelerometers organized in a 4×4 -grid configuration was designed (Fig. 4.2). Sensors were attached to a piece of fabric with distances of 8 cm between the centers of each adjacent sensors.

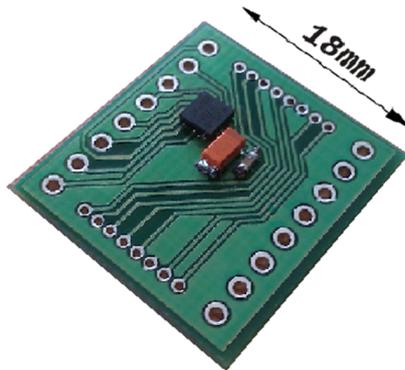


Figure 4.1. Accelerometer LIS331DLH on custom circuit board.

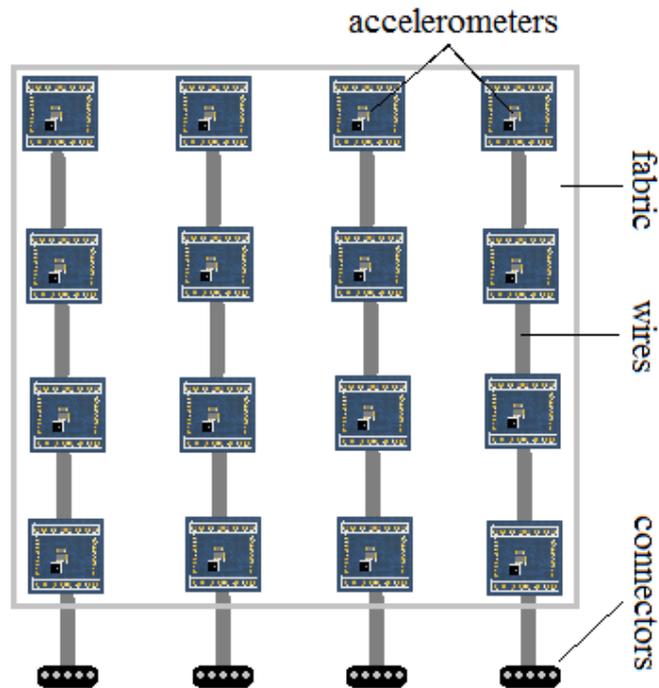


Figure 4.2. Structure of accelerometer network.

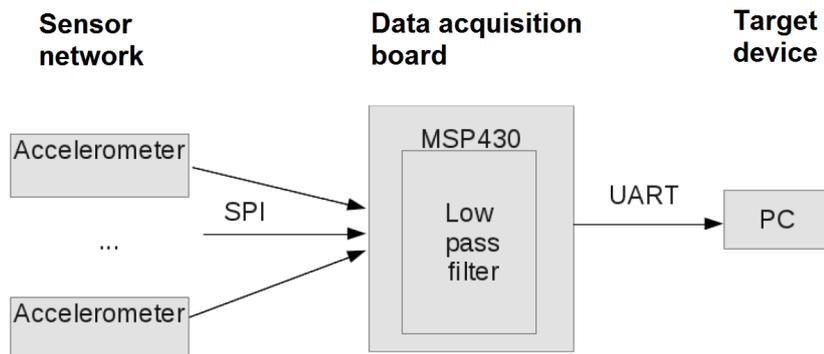


Figure 4.3. Structure of the experimental accelerometer system.

The data acquisition board was based on Texas Instruments MSP430 family microcontroller. Data were read from the sensors using SPI interface and then transmitted further using UART to the target device such as PC (Fig. 4.3). Each sensor was sampled with 50 Hz sampling rate, resulting readings were smoothed with a simple low pass filter before transmission to target device.

Before the testing of shape reconstruction performance sensor mounting errors have to be compensated for. Whenever sensors are mounted in the grid formation, perfect alignment cannot be guaranteed. For mounting error correction the method described in [83] was used. The grid is applied to a perfectly flat vertical surface. Using equation 2.5 from the Subsection 2.1

resulting base rotation matrix R_b for each sensor is calculated. This matrix contains the initial sensor orientations around both horizontal axis relative to the flat reference shape. The inverse of this matrix can be used for mounting error compensation. During shape reconstruction each new orientation estimate $R(q)$ is multiplied by the inverse of corresponding R_b . This results in modification of the segment transformation equation 2.6 from the Subsection 2.1 to the following:

$$\vec{v}' = R_b^{-1}R(q)\vec{v} \quad (4.1)$$

For the evaluation of the shape approximation precision, the distance between vectors from the reference model and obtained vectors from the experimental system were calculated for each vector in surface model. Parameters Δ_{max} and Δ_{avg} were introduced, which are maximal and average of these distances respectively. Several tests were done by applying the sensor grid to different surfaces with known geometry. Surface was approximated using the simplified approximation algorithm described in Subsection 2.1 according to Fig. 2.3. The shapes of the test surfaces were chosen to minimize the requirement of orientation detection around vertical axis, that is not obtainable solely with acceleration sensors. In Fig. 4.4 one of the approximated models is displayed. Overall resulting Δ_{avg} during the tests are plotted in Fig. 4.5 with respect to the average surface inclination from vertical axis. As this inclination angle increases, the uncalibrated mounting error around vertical axis has greater impact on Δ_{avg} . For measurements with average surface inclination from vertical axis below 60° the average error is 0.71 cm.

Due to the accelerometer inability to detect rotations around vertical axis, an assumption has to be made that the segment rotations around vertical axis are equal and fixed. For more complex shapes this is no longer valid and significantly larger errors can be introduced. This can be seen in Fig. 4.6 where the method is used for human back surface model reconstruction [14]. In positions with severe slouching, distances $d1$ and $d2$ are significantly different in the reconstructed model, although they are approximately equal in real life. This arises because the segments on the sides of of the actual surface have different orientation around vertical axis that is not taken in account during the reconstruction of the model.

In Subsection 2.1 a method that uses modified reconstruction algorithm that allows selection of rotation angle for third degree of freedom (rotations around vertical z-axis) basing only on inclination data was proposed for minimization of this drawback. The feasibility of this modified method was tested with the same 4×4 accelerometer grid prototype device. The sensors were

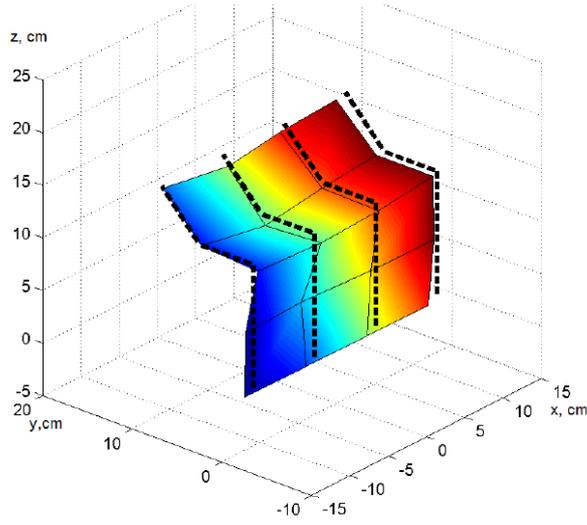


Figure 4.4. Approximated model form accelerometer sensor grid data.

Resulting surface with superimposed reference model vector positions (dashed lines). $\Delta_{max} = 1.26cm$ and $\Delta_{avg} = 0.67cm$.

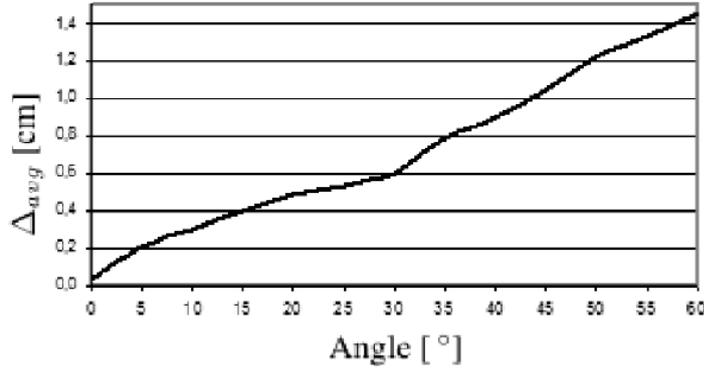


Figure 4.5. Average inclination of model vectors from vertical position and resulting Δ_{avg} .

fixed on an elastic wearable harness in an evenly spaced grid formation. Values \vec{N}_i , \vec{E}_i , \vec{S}_i , \vec{W}_i for each of $i \in [1..16]$ (see Subsection 2.1) were calculated from the corresponding sensor data using quaternion method described in Subsection 2.1.1. Segments were rotated to their corresponding sensor orientation according to their inclination measurement data.

A curved reference surface was selected to validate the shape reconstruction performance of the proposed improved method with experimental accelerometer network [14]. The shape of selected reference surface is a conical frustum with base radius of 273 mm, top radius of 236 mm and height of 333 mm. The sensor network was attached to this reference surface (Fig. 4.7, (a)) and the surface was placed 18° relative to the ground. The surface model was reconstructed from

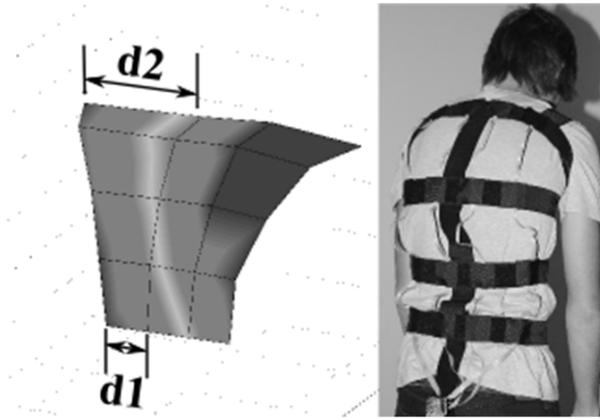


Figure 4.6. Human back posture model approximation.

Simplified approximation method introduces significant errors due to the different orientations around vertical axis (directions) of the segments in the actual surface.

the sensor data using methods from Subsection 2.1. Results were compared to the theoretical surface model. This comparison is described with an average error value Δe , which is obtained from reconstructed segment center values C_i and theoretical segment center values C'_i for all of $I = 16$ segments as:

$$\Delta e = \frac{\sum_{i=1}^I \|C_i - C'_i\|}{I} \quad (4.2)$$

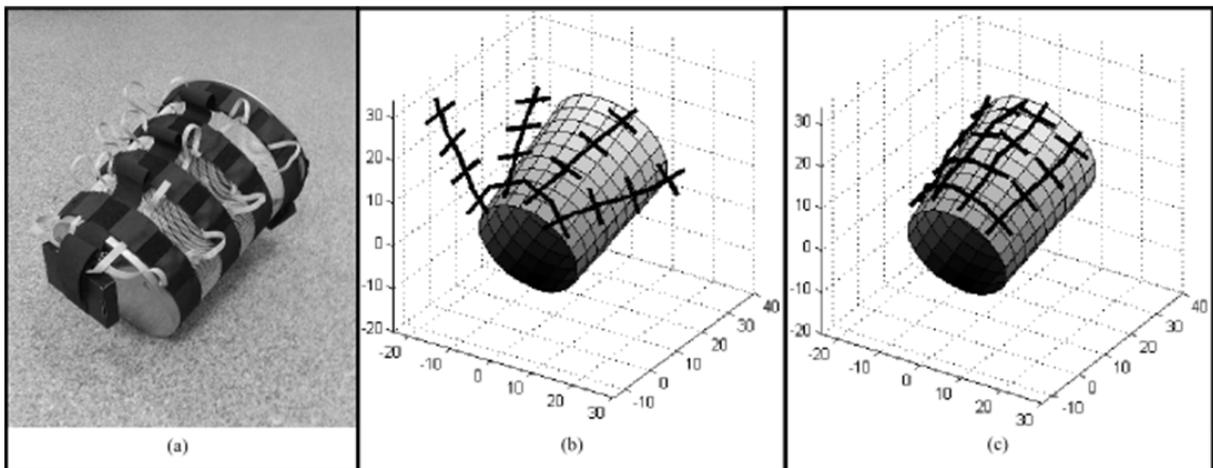


Figure 4.7. Experimental test setup of accelerometer network.

a) - experimental model; b) - superimposed reference object and reconstructed surface model assuming fixed segment orientations around vertical axis; c) - superimposed reference object and reconstructed surface model with estimation of orientations around vertical axis

Using constant and fixed segment rotations around vertical axis the average error between reconstructed surface and theoretical surface was $\Delta e = 79mm$ (Fig. 4.7, (b)). After applying

vertical axis rotation estimation algorithm the average error was reduced to $\Delta e = 12mm$ (Fig. 4.7, (c)) resulting in approximately 85% error reduction in surface reconstruction.

These calculations with a 10° step for α_i angle values within a range of $\alpha_i \in [-90^\circ..90^\circ]$ took approximately 1 second on a modern personal computer with algorithm implemented in Matlab environment.

4.2. Surface reconstruction with accelerometer and magnetometer network

As shown in previous subsection, shape data acquisition with an accelerometer network can provide basic shape data acquisition, however, it has noticeable limitations if more complex shapes have to be measured. To experimentally test free-form shape data acquisition the method based on accelerometer and magnetometer sensor network described in Subsection 2.2 was evaluated in detail.

4.2.1. Orientation estimation

For surface segment orientation estimation a sensor node based on low-power and low-cost hardware was designed. Earth gravity and magnetic field vector direction measurement was done with LSM303DLHC [84] chip with 3-axis accelerometer (set in $\pm 2g$ range and 12 bit resolution) and 3-axis magnetometer (set in ± 1.2 gauss range and 12 bit resolution). Performance of this low-cost setup in combination with algorithms from Subsection 2.2.1 was evaluated using commercial high precision inertial measurement unit Xsens MTi-G [85] as the ground truth. The data sheet specified accuracy of this module in static conditions is $< 0.5 \text{ deg}$ for pitch and roll angles, $< 1 \text{ deg}$ for heading. A sensor from acceleration/magnetic sensor grid was mounted on MTi-G sensor casing, several tests were done by slowly rotating the sensors in different orientations and the output data was compared. Prior to testing, the magnetometer was calibrated to compensate for hard and soft iron effects according to reference [86].

For output data comparison a residual matrix method was used [87]. The advantage of this method is that it decouples the sensor error with respect to Earth's gravity vector (attitude residual error: pitch and roll) from the sensor's error with respect to magnetic north (heading residual error). The residual matrix ΔR representing the difference between sensor orientation data is estimated as:

$$\Delta R = R_{grid} R_{Xs}^T, \quad (4.3)$$

where R_{grid} is the rotation matrix for the grid sensor and R_{X_s} is the rotation matrix for MTi-G. Using residual matrix, the individual Euler angle (pitch, roll and heading) errors by X-Y-Z rotation sequence convention can be extracted with these expressions:

$$\phi = atan2(\Delta R_{32}, \Delta R_{33}) \quad (4.4)$$

$$\theta = -asin(\Delta R_{31}) \quad (4.5)$$

$$\psi = atan2(\Delta R_{21}, \Delta R_{11}), \quad (4.6)$$

where $atan2$ is quadrant aware inverse tangent function and ΔR subscript numbers denote the element of matrix ΔR .

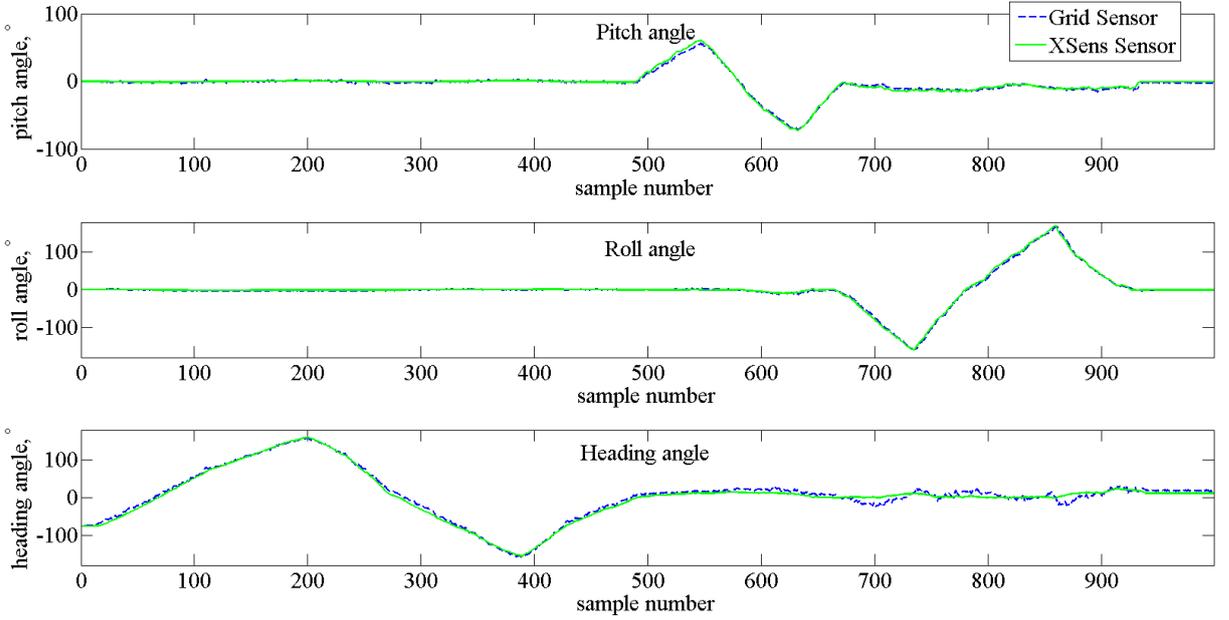


Figure 4.8. Comparison of orientation angle measurements with "Xsens" module and sensor from acceleration/magnetic grid.

In Fig. 4.8 superimposed orientation angles from Xsens MTi-G and sensor from acceleration/magnetic grid are shown. For 1000 sample measurement sequence determined RMS errors for pitch, roll, and heading angles was respectively 1.7, 1.8 and 5.4 degrees.

4.2.2. Sensor network

For validation of the enhanced daisy-chained SPI proposed in Subsection 3.2 an experimental proof-of-concept setup was developed with 60 sensor nodes (Fig. 4.9) [12]. All nodes are connected in a chain with point to point connections. Each connection requires only four wires,

two of which are for powering the devices and two for communications (see Fig. 3.2). A master data collection device is located at the beginning of the line, which gathers all the sensor data and then can either process the data locally or transmit it via wired or wireless connection (such as Bluetooth) to a more powerful device for processing.

Each of the sensor nodes were developed in such a way as to support most modern sensors. There is a huge variety of different sensors available with different specific interfaces. This problem can be overcome by adding a low cost microcontroller to each sensor node, giving the benefit of changing between most popular sensor interfaces with relative ease. Additionally the microcontroller ensures an easy to implement system providing the facilities required for the enhanced daisy-chained SPI network design. The resulting sensor node thus only has to contain two integrated circuits – one microcontroller and one sensor, making possible to implement it in a very small surface mount solution.

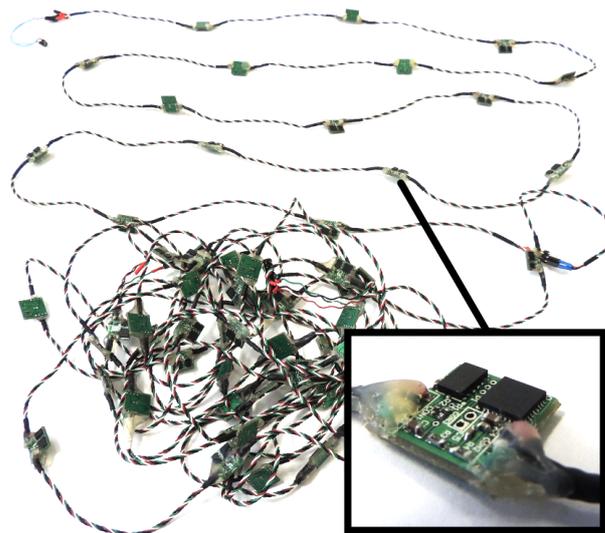


Figure 4.9. Experimental sensor chain with 60 sensor nodes.

In the proof-of-concept setup the MSP-EXP430FR5739 [88] development board was used as the master device. For the sensor nodes 3-axis acceleration and magnetic sensors LSM303DLHC [84] communicating over I2C line to MSP430g2553 [89] microcontroller were used. For sensor network architecture testing purposes only acceleration sensor measurements (6 bytes in total) were acquired. The structure of the sensor node is visible in Fig. 4.10. Full schematics of the sensor board are available in Appendix 1. The microcontroller samples the data from the LSM303DLHC sensor through I2C interface through SCL and SDA lines at a frequency set by an incoming clock line SCLK and also repeats the clock signal for the next sensor node in the

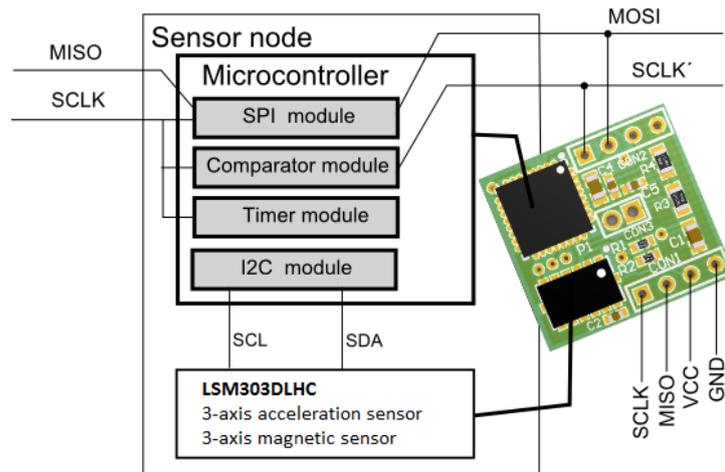


Figure 4.10. Structure of the sensor node (physical dimensions $13 \times 13 \times 2$ mm).

line over outgoing clock line SCLK'. When the data is sampled from sensors, each microcontroller transmits the data to the previous sensor node over the outgoing signal line MISO and receives data from the next sensor node over the incoming signal line MOSI. The received data is then sent further up the line until it reaches the incoming signal line MISO of the master data collection device.

The clock signal repetition on the sensor node is done by the comparator built in the microcontroller. When the incoming clock signal reaches a certain threshold the comparator circuit fires and sends out a repeated clock signal. As described in the Subsection 3.2, the repeated clock signal width is distorted in each repetition. This problem is solved by choosing between multiple available internal references of each sensor node comparator individually depending on the width of incoming clock pulses. The incoming pulses are first measured with a timer module and depending on the result a comparator reference value is set. If the incoming pulse width is narrower than the expected average pulse, the reference is set to a lower value making the repeated pulse wider, and vice versa. To achieve this each individual sensor has to work in two different modes: the first is a mode where the clock repeating circuit is calibrated, after which the sensor switches to data sampling and transfer mode. When the system is powered on, the clock repeat calibration mode is started. In this mode the master controller sends out a certain amount of dummy bytes to generate SPI clock signals for sensor node calibration. Each sensor node enables standard SPI communication and sets up the timer to capture the incoming clock pulse width. When a dummy byte is received the sensor node calculates the SPI clock pulse width and compares it to the expected for the specific running frequency. This allows to

decide which internal comparator reference voltage to use in order to maintain clock signal duty cycle as close to 50% as possible. After desired reference voltage is set, the microcontroller internal comparator is enabled for clock repetition to next sensor node. When all of the sensors are calibrated a signal to start data sampling and transfer mode is given over the data line by master data collection device.

In the beginning of the data sampling and transfer mode I2C communication lines are enabled for sensor configuration and data acquisition. Then the data is gathered as follows:

- A synchronization signal is sent over clock line by the master data collection device and repeated by each of the sensor nodes over the data line.
- Upon the signal each node synchronously samples the sensor x, y and z axis acceleration data (6 bytes in total) and stores it in the memory.
- When sensor data is sampled, nodes switch to SPI transfer mode. The master starts to generate SPI clock signal, on which each of the sensor nodes sends acquired data byte by byte over the data lines to the previous sensor in chain (first sensor node sends data to the master data collection device).
- At the same time the data that arrived from the next sensor in the chain is stored temporary in the place of the previously transmitted value thus after 6 byte transmissions the node has transferred all its previous data and stored different set of data in memory.
- This process is repeated until all sensor data has reached the master data collection device.
- When all data has been transmitted, sensor nodes disable SPI modules and wait for the next synchronization signal to start sample new data from the sensor.

After each previously described cycle the master data collection device has gathered data from all sensor nodes consecutively.

This setup makes sure that all sensor nodes sample sensor data synchronously and then the data is rapidly pumped to the master collecting device. Also each sensor node in the chain can be provided with identical software that considerably simplifies the design of the whole network.

An examples of C codes for sensor and master devices can be found in Appendix 3 and Appendix 4 respectively. Both of the code examples have only minor modifications relative to

the codes used in the experiments described in this subsection. In Appendix 3 a code version for sensor node modification when both acceleration and magnetic sensor data are acquired from LSM303DLHC is provided. In Appendix 4 a code version for master data acquisition device for implementation with MSP430g2553 microcontroller with acceleration and magnetic data reception from sensor network and retransmission through UART (see Subsection 4.2.3) is provided.

With master supply voltage (V_{cc}) 3.6 volts the measured supply current *RMS* value of each sensor node was 460 microamperes during sensor sampling and data transferring resulting in an overall power consumption of all 60 sensor network to 27.6 miliamperes. The wire resistance between two sensor nodes, taking into account both power supply and ground lines, was 0.15 ohms. The measured supply voltage for the last sensor node in 60 sensor network was 3.46 volts. Theoretical supply voltage for the last sensor node in the network with an arbitrary number of sensors nodes can be calculated according to the equation (3.3) form Subsection 3.2. The curve in Fig. 4.11 illustrates the relationship between the number of sensors in the network and the last sensor supply voltage in this network for setup where $I_{cc} = 460\mu A$, $R_W = 0.15\Omega$. The theoretical supply voltage on last sensor node for 60 sensor network is marked as 3.47 volts and approximately corresponds to the experimentally measured. Based on minimum power supply requirement for sensor LSM303DLHC which is 2.18 V, the maximum number of 201 sensor nodes is estimated for this setup.

The whole sensor network was sampled with 50 *Hz* sampling frequency receiving 6 bytes from each sensor. SPI data transfer clock frequency was 1 megahertz. The experimentally determined data preparation time T_{data} which includes sensor data sampling with microcontroller and preparation for transferring in the enhanced daisy chain network was 200 micro seconds. The measured SPI hardware delay was approximately 8 microseconds. Inserting these parameters into the equation (3.5) form Subsection 3.2, the minimum sampling period was calculated as 6 milliseconds, which corresponds to the maximum sampling rate of approximately 166 *Hz* for 60 sensor network with 6 bytes to receive from each sensor node. Based on the same calculations, a maximum sampling rate for maximum estimated number of sensors ($n_{max} = 201$) for this setup was obtained approximately 51 *Hz*.

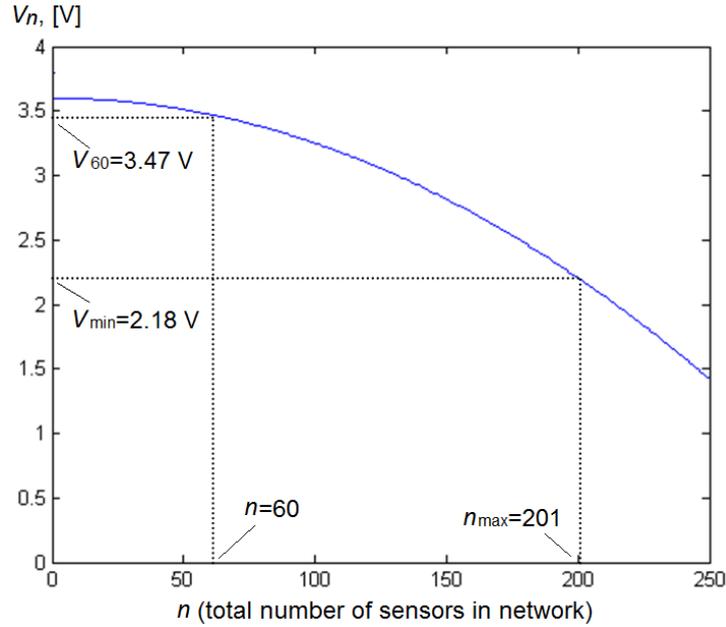


Figure 4.11. Relation between number of sensors in the network and last sensor supply voltage.

4.2.3. Surface reconstruction

To evaluate the performance of the surface shape reconstruction method proposed in Sub-section 2.2.3 a network of 63 3-axis acceleration and magnetic sensor nodes was experimentally tested [10]. Sensors were arranged in a 9×7 -grid formation and sewed between two layers of fabric with mutual distances 4.8 cm lengthwise and 3.5 cm across (Fig. 4.12). For convenient data acquisition the network was connected to a master data acquisition device with battery and Bluetooth transceiver to enable data transfer to any conventional Bluetooth enabled computing device such as PC, tablet or smartphone, where the data processing for shape reconstruction can be implemented (Fig. 4.13).

The master data acquisition module consists of a microcontroller MSP430g2553 connected to a Rayson BTM-222 [90] Bluetooth module through UART. The module also includes rechargeable 3.7 V, 1000 mAh lithium polymer battery and is supplemented with required circuitry for safe battery charging and voltage regulation. The module is depicted in Fig. 4.14 and full schematics are attached in Appendix 2. The average current consumption of data acquisition module during normal transmission is around 50 mA with the vast majority consumed by Bluetooth module. Total average current consumption of 63 sensor network and data acquisition module is around 80 mA providing more than 12 hours of operation from 1000 mAh battery.

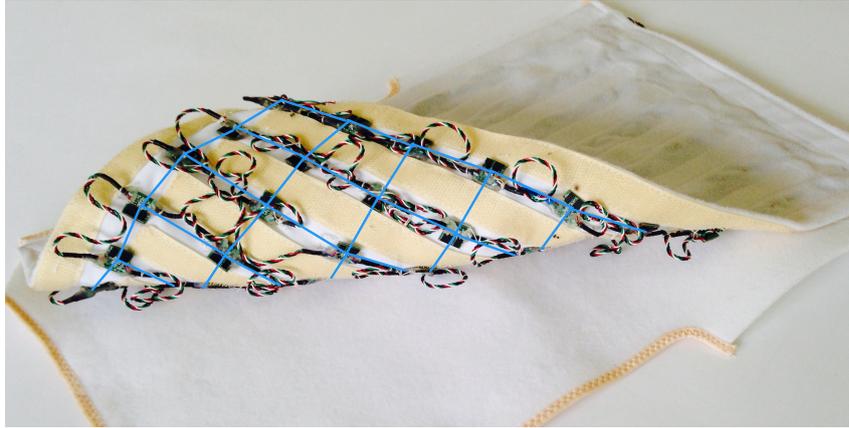


Figure 4.12. Sensor network for proof-of-concept setup. Blue lines illustrate sensor mutual grid structure.

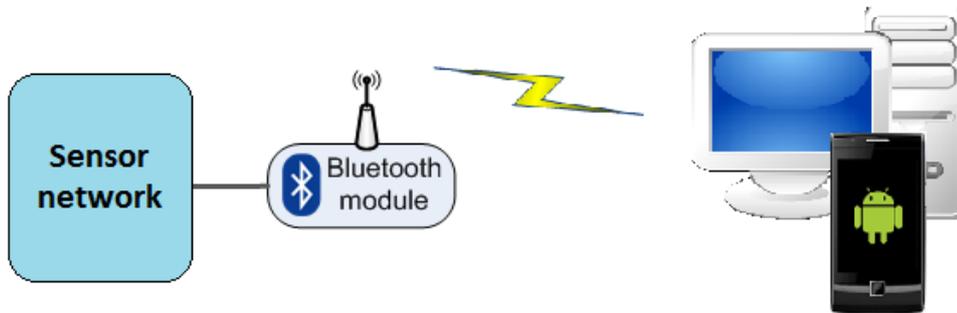


Figure 4.13. Structure of data acquisition system.

The processing of acquired data was implemented according to method described in Subsection 2.2 both in MatLab environment on a desktop computer and as an application on a mobile Android device to demonstrate fully portable system operation. In the reconstruction a simple lattice structure was drawn through obtained control points that define the surface geometry to display coarsely approximated surface model. The main MatLab code of the reconstruction algorithm for experimental system is attached in the Appendix 5 and video demonstration of real-time operation of the experimental setup is available in [91]. In Fig. 4.15 an example of reconstructed fabric shape in Android application is shown.

To evaluate the accuracy of proposed shape sensing method, the obtained models were compared to commercial Microsoft Kinect Sensor v2. Kinect v2 sensor device embeds three sensing units: RGB camera, structured light depth sensor and microphone array. For the experiments the former two were used. The accuracy assessment for Kinect v2 can be found in [92]. Depth error for range that was used in experiments (0.5-3m) is given to be less than 2mm.



Figure 4.14. Master data acquisition module.

Left - exposed components of module with MSP430g2553 microcontroller and BTM-222 Bluetooth transceiver.
 Right - components of module inside 3D printed enclosure with dimensions $5.5 \times 6 \times 1.3$ cm.

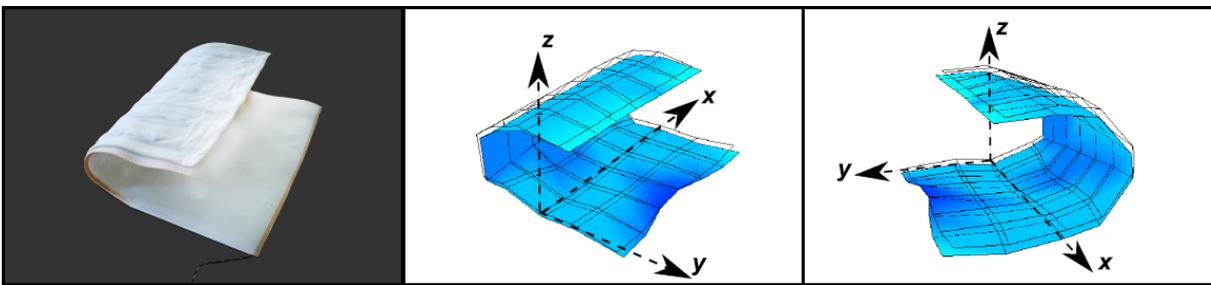


Figure 4.15. Fabric with embedded sensor layer and corresponding reconstructed shape models in Android application.

The hardware used in experimental system: shape sensing sensor array, Microsoft Kinect v2, desktop computer (Fig. 4.16). The sensor equipped fabric was filmed for several minutes with Kinect v2 sensor while performing various continuous deformations such as bending, twisting, etc., to obtain relatively complex and different test shapes. The data from Kinect and shape sensing sensor array were simultaneously sent to desktop computer using standard communication interfaces (USB 3.0 and Bluetooth) with a 10 Hz frame rate. Along each data frame time stamps were sent to ensure time synchronization of obtained data frames.

The interaction with Kinect sensor was done using Kinect for Windows SDK 2.0 that gives access to the raw data produced by Kinect sensor and also provides different utilities that allow generation of 3D point cloud and mapping between 2D RGB image and 3D point cloud.

Shape sensing array produces 3D coordinates for points in space that correspond to the sensor node locations on the grid. To calculate the difference between models produced with shape sensing sensor array and Kinect, we compared obtained sensor locations with both methods. The sensor locations on shape sensing sensor array were marked with colored markers as seen in Fig. 4.16. These markers were detected in 2D RGB image of Kinect sensor and pixels that cor-

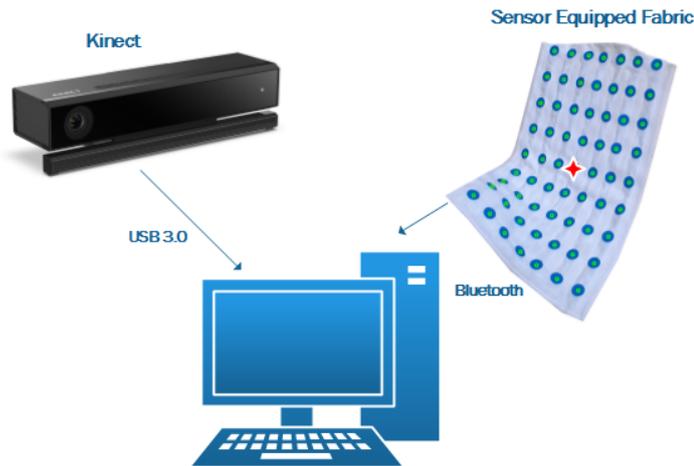


Figure 4.16. Experimental setup diagram.

On fabric sensor locations are marked with small circles. Reference sensor location is marked with a star.

respond to marker centers were mapped to corresponding 3D points of the point cloud obtained with the depth sensor, producing a Kinect representation of sensor locations.

In order to compare the models obtained with the Kinect sensor and proposed shape sensing sensor array, they had to be superimposed. This requires alignment of reference systems in terms of displacement and orientation offset. To remove displacement offset of reference systems, a reference sensor location $C[i_{ref}; j_{ref}]$ (see Subsection 2.2.3) was marked with different color as seen in Fig. 4.16. The displacement of reference systems were eliminated by equalizing the reference sensor locations in both models. To find the relative orientation offset between reference systems, ICP [93] algorithm was used. To ensure accurate alignment despite the sensor noise, a rotation matrix describing orientation offset was obtained for 100 frames and averaged using Matrix-Based mean algorithm described in [94].

In the first experiment the shape sensing sensor network was placed on an object with known geometry and errors between reconstructed model and mathematical model of object were determined. RMS errors between reconstructed and mathematical model points were 4.9 mm. Reconstructed model and reference model can be seen in Fig. 4.17 b.

In the second experiment Kinect point cloud data of reference object was compared to mathematical model of reference object. RMS errors between Kinect point cloud and reference model were 3.1 mm, that corresponds to [92]. The Kinect point cloud and reference model can be seen in Fig. 4.17 a.

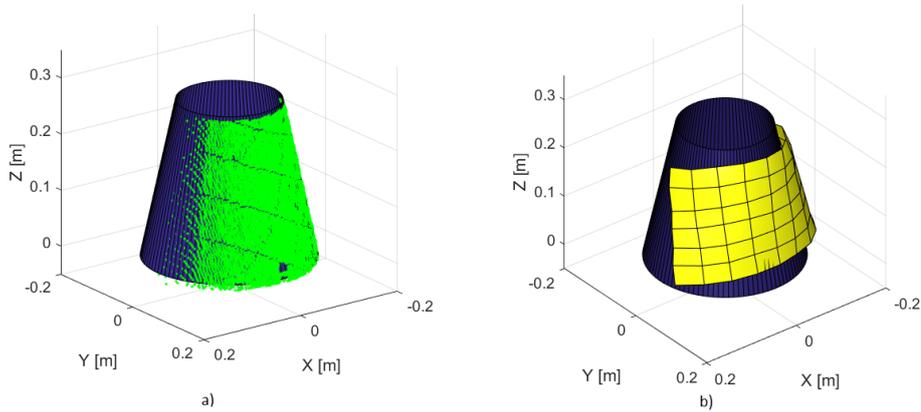


Figure 4.17. Reconstruction accuracy comparison with reference object.

a) Kinect point cloud of reference object. b) Reconstructed surface of reference object from shape sensing sensor network data.

To test larger amount of different shapes and performance in dynamic conditions, experiments were performed by comparing measured shape to the point cloud obtained by Kinect V2 sensor. The main metric for comparison of the models was the 3D Euclidian distance between sensor location coordinates obtained with Kinect sensor and proposed shape sensing sensor array. Three different experiments were done - two in static conditions (movement acceleration $\ll g$) and one in dynamic. In each experiment data for at least 1000 different test shapes was obtained.

In static conditions two tests were done: one with the reference sensor located in the center of the grid and other in the corner of the grid. This was done to observe error distribution in the reconstructed model, as reconstruction error was expected to increase for sensor locations further away from reference. In Fig. 4.18 the mean differences of all shapes for each sensor location are mapped on the sensor grid structure. The black dots indicate the locations of the sensors in the grid (points that are reconstructed). The data was interpolated between the sensor locations with bilinear interpolation. As expected, it can be seen that the mean differences are greater for points that are further from reference sensor location. The distribution of the mean differences for reconstruction are shown in Fig. 4.19 with reference sensor in the center and Fig. 4.20 with reference sensor in the corner. Differences and standard deviation along each axis is provided in Table 4.1.

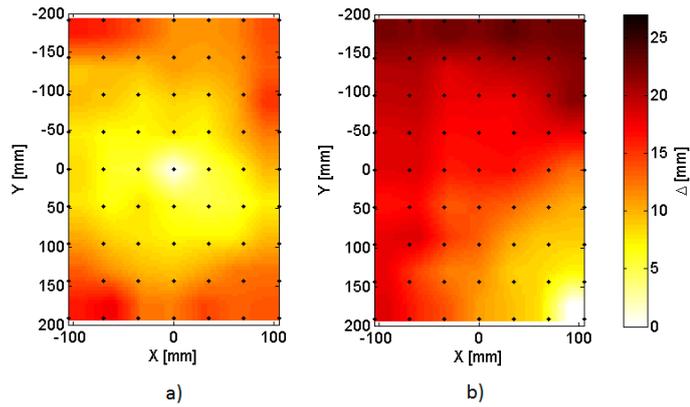


Figure 4.18. Difference maps of shape sensing sensor array in comparison to Kinect reconstruction.

a) reference sensor located in the center of the grid, b) reference sensor located in the corner of the grid. The black dots indicate the locations of the sensors in the grid (points that are reconstructed).

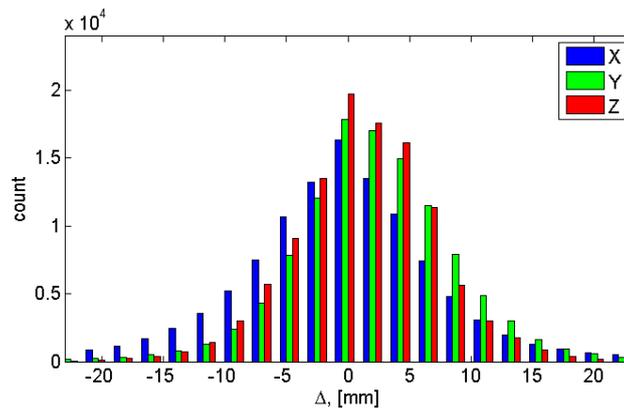


Figure 4.19. Difference distribution between points from proposed method and Kinect sensor. Reference in the center.

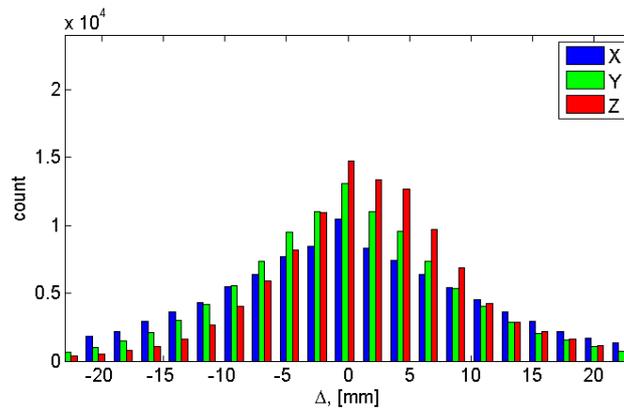


Figure 4.20. Difference distribution between points from proposed method and Kinect sensor. Reference in the corner.

Table 4.1

Distribution parameters of differences between Kinect and Shape sensing array points

	X	Y	Z
μ_{center} [mm]	-0.7	2.1	1.1
σ_{center} [mm]	8.9	6.7	5.8
μ_{corner} [mm]	0.1	-0.1	1.6
σ_{corner} [mm]	14.4	10.4	8.7
$\mu_{dynamic}$ [mm]	1.1	2.9	1,3
$\sigma_{dynamic}$ [mm]	12.1	10.4	8.2

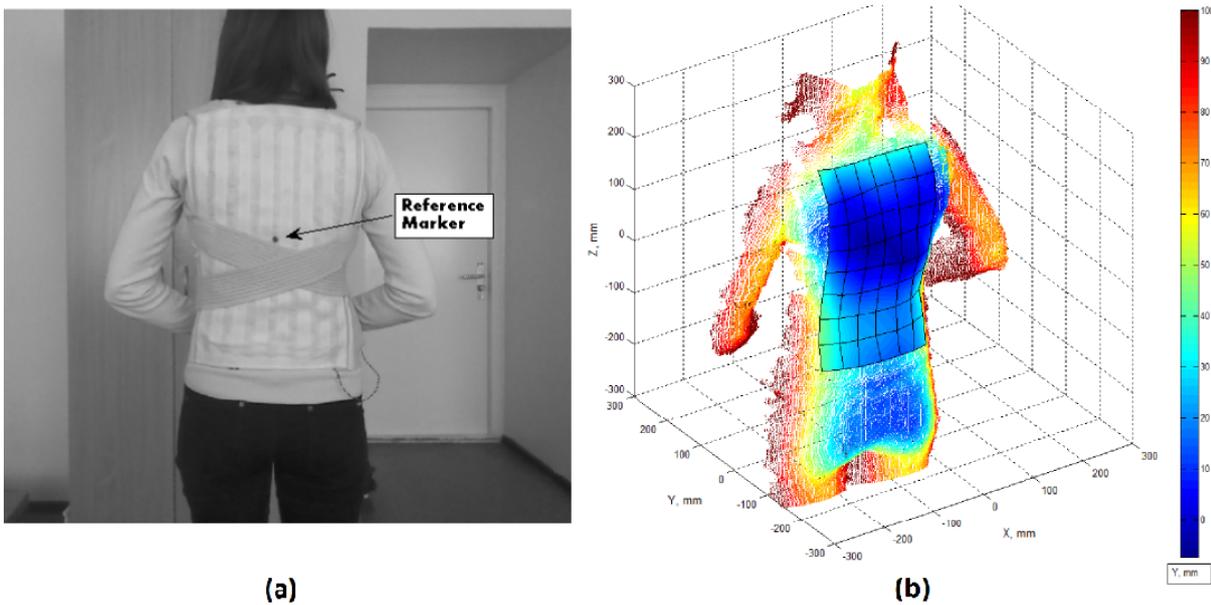


Figure 4.21. Posture monitoring experimental setup.

a) human wearing vest with an integrated shape sensing sensor array; b) superimposed models of Kinect sensor and author's proposed system. The color is mapped along y axis to demonstrate the depth profile.

As some of the potential application scenarios include dynamic motions (such as wearables), a specific test was conducted to see how surface reconstruction accuracy degrades in dynamic conditions. Motions with accelerations similar to human motion and exercises were tested. A wearable vest with the back side made from a fabric with attached shape sensing sensor array was designed (Fig. 4.21 a). A person was standing with the back side towards Kinect sensor. While performing movements such as bending and crouching in speeds typical to a light workout, the data was acquired similarly as in previous experiments. In addition to 50 Hz low-pass filter on the sensor integrated circuit, the acceleration and magnetic data were filtered with smoothing Kalman filter with empirically obtained coefficients to enhance measurement performance in

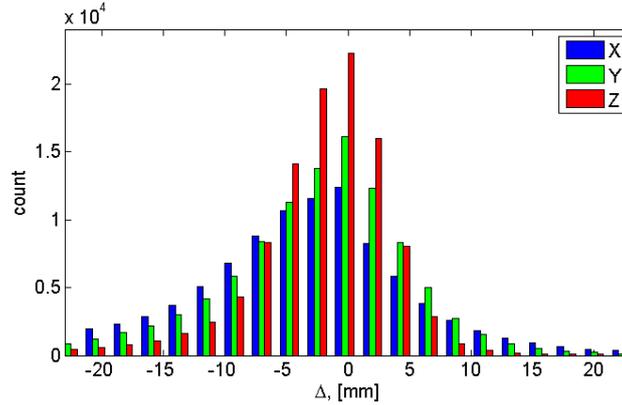


Figure 4.22. Difference distribution between points from proposed method and Kinect sensor during dynamic movement.

dynamic conditions. The distribution of differences for dynamic shape reconstruction can be seen in Fig. 4.22 and Table 4.1 contains statistic parameters of distribution.

4.3. Conclusions

In Subsection 4.1 the ability to reconstruct a basic surface shape with a 3-axis accelerometer network was demonstrated. Acceleration sensors can not directly measure surface segment orientations in all three degrees of freedom limiting the ability to measure free form 3D shapes. It was also shown that the proposed improved algorithm can provide measurable reduction of this problem by obtaining additional information from the segment grid structure and physical constraints that can not be directly acquired from sensors. This can be used in scenarios where physical constraints such as continuity are applicable, for example in human posture monitoring where sensors can be attached to the surface of clothing. Despite the increased precision the method still does not provide free form shape measurements. Also, it currently requires too much computational power to be practical for real-time embedded applications, especially when the number of sensors increases. Basing on this the author suggests that for mobile CPS applications if free from 3D shape measurement is required more suitable would be some direct method that can measure the third degree of freedom of orientation such as addition of magnetometers.

The proposed enhanced daisy-chained SPI architecture was tested experimentally in Subsection 4.2.2. Experimental setup showed capability of data acquisition from up to 200 sensor nodes while maintaining maximum sampling frequency of whole network around 50 hertz, which is reasonable for a number of real time applications.

The factor that limits the maximum number of sensors the most is the sensor node supply voltage drop on power lines. Equation (3.3) from Subsection 3.2 suggests that reducing either resistance of the conducting wires or the current consumption of sensor nodes could both significantly increase maximum number of sensors. For example, if the same hardware from our experimental setup is used and the distance between sensors is reduced from 10 cm to 1 cm, thus reducing wire resistance approximately 10 times, it would be possible to connect more than 600 sensor nodes in the network before reaching critical voltage drop for the active components on the last sensor node. Also from equation 3.5 we can obtain that with the same communication parameters a maximum of a 17-Hz sampling period could still be achieved for the network of 600 nodes. Also, the individual current consumption of each sensor node could be reduced, thus reducing the voltage drop on power supply lines and allowing the addition of more nodes in the network. The demonstrated ability to acquire data from a low-power, low-cost sensor network is essential for mobile CPS, especially the implementation of shape sensing methods which benefit from large amount of closely spaced sensors.

In Subsection 4.2 a method for 3D shape data acquisition with a 3-axis acceleration/magnetic sensor network is experimentally validated. The performance of the proposed system was evaluated in detail using Kinect V2 sensor as reference allowing fast and easy comparison of large number of different test shapes.

According to Table 4.1, in experiments where no significant dynamic acceleration is present, the proposed system provided sensor location reconstruction with standard deviation of error up to 9 mm relative to Kinect v2 sensor, when shape reconstruction reference point was in the center of the model. This is relatively high compared to approximation error in simulation results provided in Subsection 2.2. In dynamic conditions and when the reference point was in the corner average reconstruction accuracy reduced even more. These errors occur due to the various error sources analyzed in Subsection 2.2.4 and have significant impact on reconstruction accuracy.

The sensor measurement errors consist of calibration errors, sensor noise and dynamic accelerations (for accelerometers) as well as magnetic perturbations (for magnetometers). Empirically it was found that the sensor noise level of each sensor is around 0.5 % relative to the measured vector. Calibration errors after performing calibration procedure described in [86] are even smaller, therefore, both of these error sources introduce negligible impact on shape recon-

struction error according to Fig. 2.11. In practical experiments it was found that the magnetic sensors often need to be re-calibrated to maintain measurement accuracy.

In contrast, accelerations due to movement as well as magnetic perturbations that occur in the environment can introduce significant errors in measurement of Earth gravity and magnetic field vectors. The impacts on reconstruction accuracy of these disturbances can be determined from Fig. 2.11. This problem could be reduced by applying advanced data filtering methods or adding additional sensors such as gyroscopes to assist orientation estimation, however, increased data processing complexity due to gyroscope data integration for orientation estimation needs to be resolved for low-power systems with large number of sensor nodes.

Fig. 2.11 also shows the impact of mechanical mounting errors on shape reconstruction accuracy. It is reasonable to state that without sophisticated manufacturing process (Fig. 4.12) the sensors are mounted with up to a 10-% location error (3.5 *mm* error for inter sensor distances of 3.5 *cm*) and a 5-% orientation error ($\pi/20$ angle error) leading to an increase in reconstruction error of around 0.9 % and 2.0 % respectively. This implies that during the system design particular attention has to be paid to mechanical implementation. The mounting rotation errors can be calibrated by placing sensors on object with known geometry and alignment in general reference frame. Deviations between the expected and measured sensor orientations can be used for reference frame misalignment compensation. In practice, this approach can be quite challenging to perform without additional equipment.

Various error sources currently encumber practical implementation of shape sensing from local orientation data to reach the accuracy of commercial sensors utilizing external instrumentation (such as Kinect v2). However, it is demonstrated that the proposed method is portable and scalable, self-contained, occlusion free, it relies on low-cost sensors and required computations can be done on conventional portable computing devices such as smartphones. This makes the solution potentially attractive for applications in mobile CPS.

5. SHAPE SENSING APPLICATION IN MOBILE CYBER-PHYSICAL SYSTEMS

5.1. Posture monitoring and feedback

In this subsection the proposed shape sensing method is applied in mobile CPS for human posture monitoring. A complete system, that can be used during daily activities for posture monitoring and biofeedback generation is proposed [13]. The proposed system consists of two parts. The first part is the wearable sensor system for data acquisition and transfer, which includes a body mounted sensor network, a microcontroller and a Bluetooth module to provide wireless data transfer. The second part is a conventional smartphone with a custom made application for sensor data processing, visualization and feedback generation. Structure of the proposed posture monitoring and feedback system is visible in Fig. 5.1.

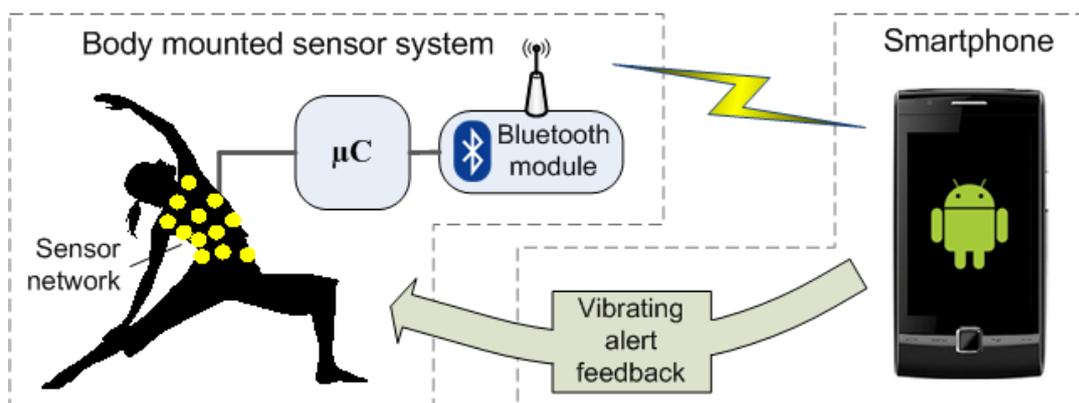


Figure 5.1. Structure of posture monitoring system.

In the wearable sensor system part a sensor network attached to human back is used for posture data acquisition. The sensors are evenly spread around back surface, providing detailed information about posture deformities. A microcontroller is used to sample data from sensors and send them to the Bluetooth module for wireless data transmission. Bluetooth transmission is chosen to provide convenient data transfer to smartphone application as most of available smartphones have integrated Bluetooth transceivers.

The sensor data processing is implemented in Smartphone application to provide system portability and reduce development complexity and cost. Smartphone application receives data from sensor network via Bluetooth. Processing is done to reconstruct the geometric model of sensor network alignment thus obtaining data about posture. By comparing this model to

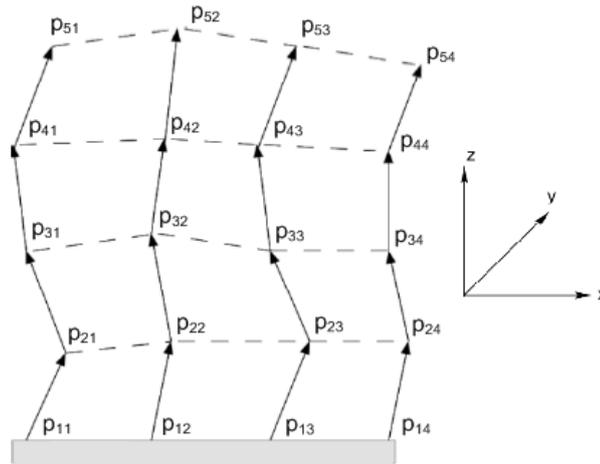


Figure 5.2. Construction of approximated 3D posture model.

the previously stored reference, decision is made whether alerting feedback must be generated. Feedback is provided using a speaker or vibrating motor integrated in the smartphone.

For human posture model acquisition the surface reconstruction method described in Subsection 2.1 is used. The three axis acceleration sensor network is mounted on human back to provide detailed information about human posture. Network of 16 interconnected sensors described in Subsection 4.1 arranged in four by four grid is used. Each sensor corresponds to a particular segment of back surface. A model of human back that corresponds to the actual posture is constructed using the surface shape reconstruction algorithm that uses simplified surface model and measures orientations with only two degrees of freedom proposed in Subsection 2.1.2. Approximated posture model is defined by a set of points p_{ik} (i – row, k – column) in 3D space which describe vector end points (Fig. 5.2). Each p_{ik} is a point in 3D space.

For data acquisition and wireless transmission a specially designed circuit board is used. The body mounted sensor network is wired to this board. MSP430 series microcontroller is used to sample data form sensors with a sampling rate of 50 Hz. Then a simple averaging of 5 samples is done to smooth the vibrations in the signal and reduce the amount of data to be transferred wirelessly. The averaged data are sent to BTM-112 Bluetooth module which uses SPP (Serial Port Profile) for wireless transmission to the smartphone. The system is powered with three 3 AAA rechargeable batteries and a voltage regulator is used to provide the required supply voltage.

An algorithm that compares the current state with the previously stored data of corrective posture is used for calculation whether to provide feedback. As previously stated, the human

posture model is defined by set of points in 3D space. At the beginning of use, initial calibration is required to acquire data about reference model of desired posture. Similar to the current state model this reference is defined by a set of 3D points ref_{ik} . This data is used to find parameter Δ_p which describes difference between current state and reference model. Δ_p is calculated as average distance in 3D space from each point in current data set to corresponding point in reference data set. If total number of sensor rows are defined with r and columns with c , then this average distance can be found:

$$\Delta_p = \frac{\sum_{i=1}^r \sum_{k=1}^c ||p_{ik} - ref_{ik}||}{rc}. \quad (5.1)$$

If Δ_p is grater than some experimentally determined threshold value Δ_{tresh} , then feedback is generated to warn about poor posture. At first vibrating motor in the smartphone is turned on for one second to warn about deviation from reference posture. If value of Δ_p does not drop below value of Δ_{tresh} during next five seconds, feedback signal is turned on until this condition is true.

A custom Android application was developed for user inteface, sensor data reception, processing, logging, approximated posture visualization and feedback generation. Basic flow chart of application operation is visible in Fig. 5.3. In the initial step the application establishes Bluetooth connection with the wearable sensor device. After smartphone and sensor device have been paired, initial calibration is required to obtain reference model of correct posture. With sensors mounted on body, subject must perform desired posture and store corresponding data ref_{ik} in the smartphone memory by pressing “Save state” button in the application. Also the approximate height of subject, and threshold value Δ_{tresh} must be set. Height of subject is used to estimate length of the vectors, from which approximated posture model is constructed. Δ_{tresh} define allowed rate of difference between current posture model and reference. Higher Δ_{tresh} means that subject can move more freely away from reference posture before feedback is generated.

After initialization, current posture monitoring can be performed. This is done in cyclic manner, where in each cycle a set of sensor data is received and processed in real time (faster than the arrival of the next data set). First, the calculation of posture model defining points p_{ik} and construction of graphical posture model is done using the method described in subsection 2.1. A graphical model provides visual reference of current posture that is monitored by the

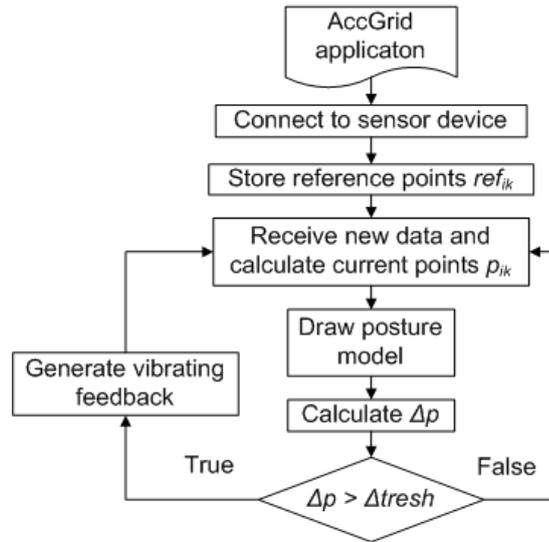


Figure 5.3. Flowchart of feedback control algorithm in Android application.

system. Then the parameter Δ_{tresh} is calculated using equation 5.1. If Δ_p is greater than Δ_{tresh} , feedback signal is generated.

Additionally, the Android application can log all posture data on phone memory for later analysis. Also in a given time period the application can register the amount of time when bad posture ($\Delta_p > \Delta_{tresh}$) was detected. This is provided as ratio in percent relative to the full time period of system operation. Both of these functions can be used for usage statistics collection and validation of feedback effectiveness.

As a result an experimental device for posture data acquisition with custom Android application for data processing and feedback generation was developed. The user interface of application is visible in Fig. 5.4 (a), (b), and (c). In Fig. 5.4 (a) interface screen of Bluetooth connection establishment is visible. Fig. 5.4 (b) shows the processing screen where user can save reference state, run the real-time processing and change parameters such as feedback threshold value, height, etc. Fig. 5.4 (c) contains a screenshot where the approximated posture model visualization is displayed in real-time. The constructed model corresponds to current posture visible in Fig. 5.4 (d), whereas the thin lines indicate the shape of previously stored reference posture.

To evaluate proposed systems operation performance and influence on users posture, a test study was done. The experimental device was tested on four healthy subjects. Sensors were mounted on subject back with elastic bands as visible in Fig. 5.4 (d). Data were logged while subjects were working at desk in sitting position without back support. Position when sitting

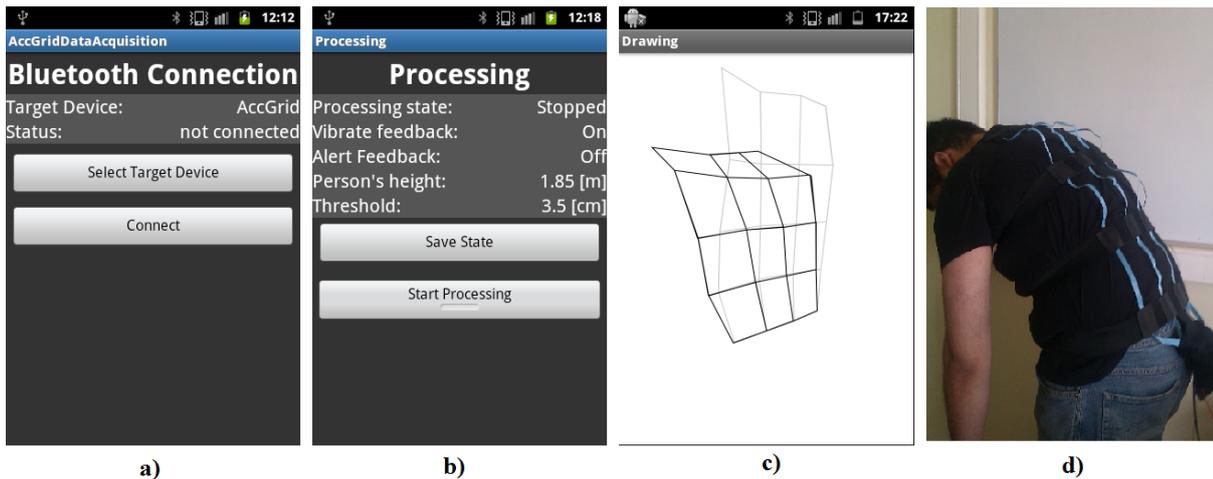


Figure 5.4. Android application screens and sensor device.

a) - screen of Bluetooth interface setup, b) - screen of processing state interface, c) - screen of stored and reconstructed posture models, d) - wearable device.

with straight back without slouching or bending sideways was calibrated as reference posture. Each subject logged their posture data with two different feedback threshold values: $\Delta_{thresh} = 3cm$ and $\Delta_{thresh} = 5cm$. Each threshold value was used for approximately half an hour. Obtained data from all subjects are summarized in Fig. 5.5. The histogram illustrates distribution of logged Δ_p values, which determines the offset from reference posture. Each set of bars corresponds to 0.5 cm wide interval, and show relative amount of time in % when posture with Δ_p in this interval was detected. The dashed lines indicate boundaries where in each threshold case feedback alert was switched on. Impact of feedback is clearly visible, as in both cases noticeably decreases the relative time spent in postures with Δ_p exceeding the threshold value. With the feedback threshold values $\Delta_{thresh} = 3cm$ and $\Delta_{thresh} = 5cm$, the time spent in “correct” posture ($\Delta_p < \Delta_{thresh}$), was respectively 89% and 90%. However, with the threshold $\Delta_{thresh} = 5cm$, only 49% of the time was spent with $\Delta_p < 3cm$, which shows feedback threshold value influence on subjects behavior.

Test results of the experimental setup proved the potential of the system to recognize different posture deformations and produce feedback if undesired posture is detected. Testing showed that around 90 % of time subjects spent with posture similar to the reference within the threshold interval. There is a noticeable reduction in distribution of Δ_p after threshold values, which indicate the presence of the feedback alert. This proves that biofeedback is an efficient way to control the users posture. As the sensor network is able to provide detailed data about different

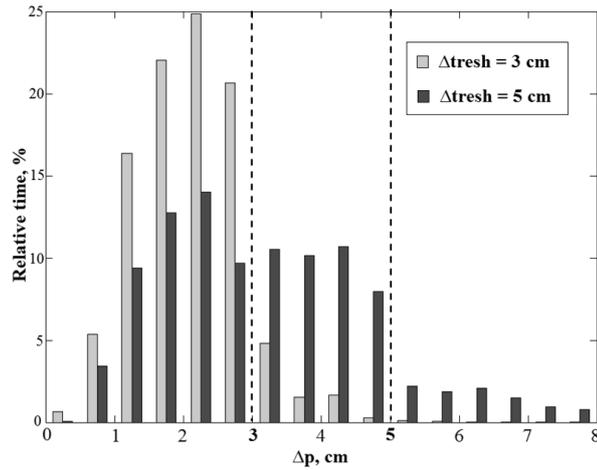


Figure 5.5. Four test subject average posture statistics with different feedback threshold values.

deformities, it is reasonable that system could detect scoliotic curves and provide feedback for patients who are diagnosed with scoliosis. This kind of system can help user to obtain specific required posture, thus it could serve as alternative to traditional hard material braces and help to reduce development of scoliosis. The sensors used in this method are small and lightweight which potentially allow them to be integrated into garment almost invisibly. Also, wires could be integrated into clothing which would allow unobtrusive use of the system during daily activities. This can provide a more convenient and visually appealing treatment method for scoliosis, which both are essential aspects for patients diagnosed with this condition.

5.2. Approbation and pilots for medical applications

To validate the proposed methods for application in mobile CPS operating in real-life conditions an approbation for medical applications was carried out. A number of prototype devices were built that utilize the proposed shape sensing method and their application piloted. Testing of prototypes with real patients was done in close collaboration and supervision of medical specialists from rehabilitation center "Mēs Esam Līdzās (MEL)" [95] and "Unihaus" Ltd. [96].

In "Unihaus Ltd." the posture monitoring system described in Subsection 5.1 was approbated in real life conditions for patients aged from 13 to 15 with severe back posture defects. The aim of approbation was to validate the system applicability in orthopedics and physiotherapy. During the approbation several suggestions for improvements of the prototype were received from medical specialists and implemented in the system. Suggested improvements included changes

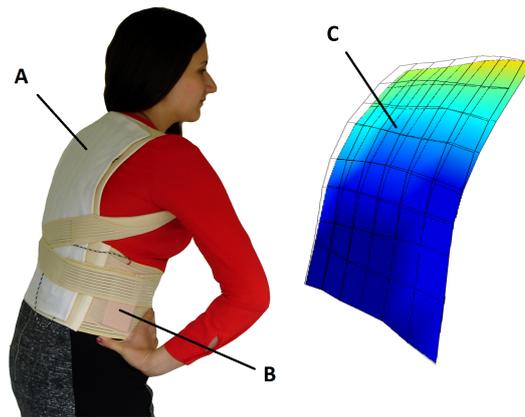


Figure 5.6. Smart wearable device for posture monitoring.

A - shape sensing fabric with embedded accelerometer/magnetometer network; B - data acquisition module with Bluetooth transceiver and rechargeable lithium battery; C - posture model rendered in Android application. The color indicates deviation from previously stored reference model indicated with transparent grid.

in displayed posture model for easier user interface and feature to store different posture models in the systems memory in order to compare obtained models before and after physiotherapy exercising session. Some practical suggestions also were received regarding the sensor integration into wearable garment for convenient use. It was suggested that the system has a potential application for biofeedback of posture during daily activities, specific therapy sessions as well home exercising. Also, a potential to substitute expensive 3D scanning technology that is currently used to obtain 3D shape of human posture in medical facilities was indicated. Full review from "Unihaus" Ltd. of the approbation is attached in Appendix 6.

In collaboration with the rehabilitation center "MEL" a prototype for mobile posture monitoring and feedback system was developed and piloted for application in rehabilitation of young cerebral palsy patients with upper body muscle dystonia. A smart wearable device (SWD) was developed basing on mobile system architecture described in Subsection 5.1 and shape sensing fabric described in Subsection 4.2.3 (see Fig. 5.6). The device is connected to a smartphone where a custom application provides reception of sensor data, processing, data storage, visualization and user interface. The posture model was obtained using algorithm described in Subsection 2.2.

During the pilot study the SWD was tested with cerebral palsy patients in close supervision of medical staff. It was used as technical tool to assist in rehabilitation exercises to help patients to control their upper body position and posture. During exercising the patient was wearing

SWD and had a tablet or smartphone nearby which was connected to the SWD via Bluetooth and was running data processing and system interface application. The application reconstructed the posture model in real-time, compared it to previously stored reference posture and produced feedback as vibrating alert and/or sound to warn the user about any deviation from desired posture set by the medical specialist. The system also provided usage statistics such as duration of training and relative time when poor posture is detected.

In Fig. 5.7 the process of system application in pilot study is shown. The study showed that the proposed posture monitoring and feedback system provided non obtrusive means for posture monitoring and feedback in mobile environment without specific infrastructure. The method for posture data acquisition proved to be convenient during daily exercising, implying that the electronics are of small size and can be fully embedded into garment, have a reasonable battery life providing a whole day of system operation on a single charge, and have a data processing application that can run smoothly in real time on conventional smartphones and tablets. It was demonstrated that the system is valuable for application in therapy sessions for children and helps to improve upper body control [20]. The review of approbation and pilot study from rehabilitation center "MEL" can be found in Appendix 7.

With recommendation from medical specialists of rehabilitation center "MEL" another prototype SWD was developed which, in addition to posture monitoring, can also monitor the position of human head relative to human spine. This additional data is valuable for rehabilitation of patients with severe movement and posture dysfunctions. The data of human head position can be used for visually easy to understand feedback generation that can facilitate training of head position fixation for patients with physical and often also mental disabilities. In the meantime the data of human back posture can help medical specialists to understand correlation between poorly fixed head position and back posture deformations.

The monitoring of head position was implemented by adding additional sensor node to the system, that is attached to human head and can measure its orientation. An extension for previous application was developed that can provide visually easy to understand feedback of head position. The relative orientation of users head was translated into displacement of object in the feedback application as depicted in Fig. 5.8.

During the pilot study the system was tested with patients who have dyskinetic and spastic form of cerebral palsy. The process of application is depicted in Fig. 5.9. The preliminary



Figure 5.7. Pilot study of SWD application in rehabilitation center "MEL".

(a) the SWD is being setup for application; (b) reference posture is being calibrated and stored in system memory; (c) the features of the system is introduced to the patient; (d) training process, the system is running in background of therapy session activities showing real time model of user posture and providing feedback if deviations from calibrated reference model are detected.

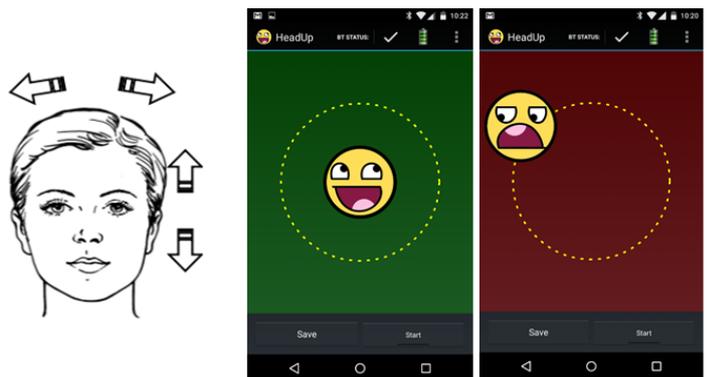


Figure 5.8. Feedback interface for human head position feedback.

The movements of human head in sagittal and coronal are translated into displacement along x and y coordinates of the object. The task of the user is to hold the object in application within the defined circle by holding his head in correct position defined by medical staff.

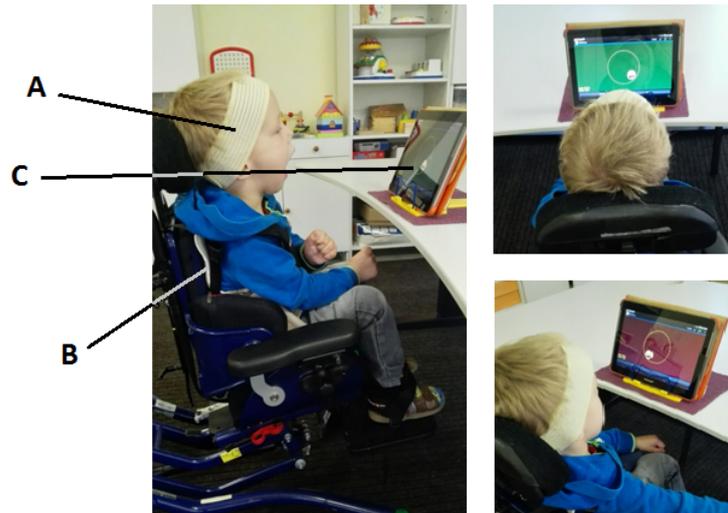


Figure 5.9. Pilot study of SWD with head position feedback in rehabilitation center "MEL".

A - head mounted sensor; B - shape sensing fabric for posture monitoring; C - tablet with data processing and feedback visualization.

studies provided that the proposed SWD could improve patient postural control [19].

5.3. Conclusions

The approbation and pilot tests proved that the proposed methods can be practically implemented in mobile CPS. It was demonstrated that the proposed shape sensing method that is based on orientation measurements from acceleration/magnetic sensor network indeed can be seamlessly integrated into smart garments providing data about the wearer in an unobtrusive way.

The pilot studies confirm that the proposed data processing algorithms can be implemented using conventional mobile computing devices such as smartphones or tablets. The custom made application running the proposed shape reconstruction algorithm in real-time was tested on multiple smartphones. Most tested units can be classified as mid-range devices with processor clock frequencies starting from 600 MHz and this computing power proved to be enough to implement real-time algorithms for multi sensor data processing and construction of a graphical model. This proves that the proposed algorithms are lightweight enough to be integrated into mobile CPS with limited resources.

Also, it was shown that embedded sensor system integration with modern portable computing devices provide excellent portable platform for mobile sensor data processing. This approach

also considerably reduces system development complexity and cost, as part of the hardware that is necessary for systems functionality is already provided. One drawback of the smartphone inclusion in system architecture is the necessity of Bluetooth data transfer, as it is the most convenient way to send data to smartphone. This solution has relatively high energy consumption (around 60% of total in SWD with 63 acceleration/magnetic sensor nodes), and it limits the battery life of both the sensor system and the smartphone.

The clear application potential of the proposed shape sensing methods are proved by interest from medical stuff which has taken a form of collaborative publications [\[19, 20\]](#) and presentations in medical conferences. In addition, while the practically tested applications in this work are related to medicine, the author feels that it is by no means the only suitable field of application for proposed methods. The ability to measure free form 3D shape in real-time with low-cost, low-power electronics can also be beneficial in other fields mentioned in introduction of this work such as robotics, flexible devices or high end sports equipment.

DISCUSSION AND CONCLUSIONS

The aim of this work was to facilitate the development of new CPS applications by exploring new 3D free form shape sensing methods. To achieve this aim a list of tasks were performed. In total six sections including final conclusions were written to describe activities and results of each task.

At the beginning of the work in Section 1 a literature review about previous work on shape sensing with embedded equipment was provided. In previous works two distinct approaches were found - one is based on measurement of bending of the material and other utilizes inertial/-magnetic sensors. The approach with inertial/magnetic sensors was chosen for further studies due to advancements in MEMS technology that allow production of small low-cost sensors in large quantities. In addition, these sensors often have digital output therefore they can be relatively easy integrated in various electronic systems. Because of these advantages MEMS sensors have become widely popular for various applications.

In Section 2 methods for shape sensing based on inertial/magnetic sensors were studied. First, it was demonstrated that the information of 3D object shape can be obtained with accelerometers. It was shown that this method provides solution to limited variations of object shapes and can introduce significant errors in more complicated cases. The main contribution of this work regarding shape sensing methods is in the proposed method for free-form 3D shape sensing with accelerometer/magnetometer network. It was identified that previously proposed solutions require considerable amount of computing power therefore are not well suited for mobile CPS. A novel method for shape reconstruction was proposed which requires noticeably lower computing resources while having negligible sacrifice in precision, enabling implementation of real-time shape sensing on embedded systems.

It was identified that for shape sensing with high resolution the use of large number of sensors is preferable, therefore methods for efficient data acquisition from multi sensor systems with limited resources were studied in Section 3. As none of the currently available data transfer interfaces were directly suitable for the task a new method for data transfer was proposed.

Section 4 was dedicated for the experimental validation of the practical implementation and performance of the proposed methods. First, a 4 by 4 3-axis accelerometer network was developed to test feasibility for its application of 3D shape data acquisition. Also, the proposed

improved data processing method that reduces the errors arising from limited degree of freedom orientation data was experimentally tested, but proved to be computationally inefficient and still not applicable in general case free-form shape measurements. Then, to provide experimental validation of shape sensing with acceleration/magnetic sensors, an experimental sensor network was developed with network architecture proposed in Section 3. The sensor network was embedded into a fabric and allowed to obtain the shape of the fabric in real-time using proposed shape reconstruction method. Also a specific experimental setup with Kinect V2 sensor was designed for detailed accuracy evaluation of proposed shape reconstruction algorithm. The setup allowed to simultaneously acquire data from proposed inertial/magnetic sensor network and reference shape data from Kinect V2 sensor. This provided error estimation with statistically reasonable data set.

Last but not least, in Section 5 the proposed methods were implemented in mobile CPS for medical application. Three smart wearable device prototypes were designed and piloted in two medical institutions. The devices were based on shape sensing methods proposed in this work and provided various information and feedback regarding user's body posture. In collaboration with medical specialists the devices were used for medical studies the results of which were published and presented in medical conferences. Positive feedback from medical specialists about systems capabilities and performance was received. Despite approbation only for medical applications the reported performance in terms that are essential for mobile CPS such as required computing power, precision, size, consumed current and robustness provides foundation for further application of proposed methods in other fields where mobile CPS are used.

In conclusion, in this work methods enabling real-time free-form 3D shape sensing with embedded equipment that is applicable in mobile CPS were proposed, experimentally tested and successfully piloted. All of the tasks defined in the beginning of this work are successfully completed and described throughout the sections of this document, therefore the author consider this work to be complete and finished.

During the development of this work following papers were published:

- A. Hermanis, R. Cacurs and M. Greitans, "Acceleration and Magnetic Sensor Network for Shape Sensing," in *IEEE Sensors Journal*, vol. 16, no. 5, pp. 1271-1280, March 1, 2016. (Impact Factor 1.762, IEEE, Scopus)

- A. Hermanis, R. Cacurs and M. Greitans, "Shape sensing based on acceleration and magnetic sensor system," 2015 IEEE International Symposium on Inertial Sensors and Systems (ISISS), Hapuna Beach, HI, 2015, pp. 1-2. (short paper, IEEE, Scopus)
- K. Nesenbergs, A. Hermanis, M. Greitans, "A Method for segment based surface reconstruction from discrete inclination values", (2014) Elektronika ir Elektrotechnika, 20 (2), pp. 32-35. (Scopus)
- A. Hermanis, R. Cacurs, K. Nesenbergs and M. Greitans, "Efficient real-time data acquisition of wired sensor network with line topology," 2013 IEEE Conference on Open Systems (ICOS), Kuching, 2013, pp. 133-138. (IEEE, Scopus)
- A. Hermanis, K. Nesenbergs, R. Cacurs, and M. Greitans, "Wearable Posture Monitoring System with Biofeedback via Smartphone", Journal of Medical and Bioengineering vol. 2, no. 1, pp.40-44, 2013. (WorldCat, CrossRef)
- A. Hermanis and K. Nesenbergs, "Grid shaped accelerometer network for surface shape recognition," 2012 13th Biennial Baltic Electronics Conference, Tallinn, 2012, pp. 203-206. (IEEE, Scopus)

Also, following conference abstracts, demos and posters were published:

- A. Hermanis, R. Cacurs, K. Nesenbergs, M. Greitans, E. Syundyukov, and L. Selavo. 2016. Demo: Wearable Sensor System for Human Biomechanics Monitoring. In Proceedings of the 2016 International Conference on Embedded Wireless Systems and Networks (EWSN '16), 15-17 February. Gratz, Austria. Junction Publishing, USA, 247-248. (Demo, abstract and poster, ACM)
- A. Hermanis, R. Cacurs, K. Nesenbergs, M. Greitans, E. Syundyukov, L. Selavo, "Wearable sensor grid architecture for body posture and surface detection and rehabilitation", (2015) IPSN 2015 - Proceedings of the 14th International Symposium on Information Processing in Sensor Networks (Part of CPS Week), pp. 414-415. (Demo, abstract and poster, ACM)
- A. Hermanis, A. Greitane, S. Geidane, A. Ancāns, R. Cacurs, M. Greitans, "Wearable Head and Back Posture Feedback System For Children With Cerebral Palsy", Abstract:

Journal of Rehabilitation Medicine (ISSN 1650-1977), 2015. (Abstract)

- K. Nesenbergs, A. Hermanis, A. Greitane, M. Greitans, “Virtual Reality Rehabilitation System for Children with Cerebral Palsy”, 25th European Academy of Childhood Disability, Newcastle, England, 10-12 oct, 2013. (Poster)
- A. Hermanis, K. Nesenbergs, ”Accelerometer network for human posture monitoring”, Riga Technical University 53rd International Scientific Conference, 10-12 October, 2012 Riga. (Abstract)

In addition, following international conferences and workshops were attended:

- ”4th Baltic and North Sea Conference on Physical and Rehabilitation Medicine”, 16–18 September 2015, Riga, Latvia;
- ”Cyber-physical systems week 2015”, 13–16 April 2015, Seattle, WA, USA;
- ”2015 IEEE International Symposium on Inertial Sensors and Systems (ISISS)”, 23–26 March 2015, Hapuna Beach, USA;
- ”2013 IEEE Conference on Open Systems (ICOS)”, 2–4 December 2013, Kuching, Malaysia;
- ”2013 2nd International Conference on Medical Information and Bioengineering (ICMIB 2013)”, 16–17 March 2013, Bali, Indonesia;
- ”International Symposium on Biomedical Engineering and Medical Physics”, 10–12 October 2012, Riga, Latvia;
- ”13th Biennial Baltic Electronics Conference”, 3–5 October 2012, Tallin, Estonia;
- ”5th International Doctoral School of Energy Conversion and Saving Technologies”, 27-30 May 2016, Ronisi, Latvia;
- ”4th International Doctoral School of Electrical Engineering and Power Electronics”, 29-30 May 2015, Ronisi, Latvia;
- ”3rd International Doctoral School of Electrical Engineering and Power Electronics”, 23-24 May 2014, Ronisi, Latvia;

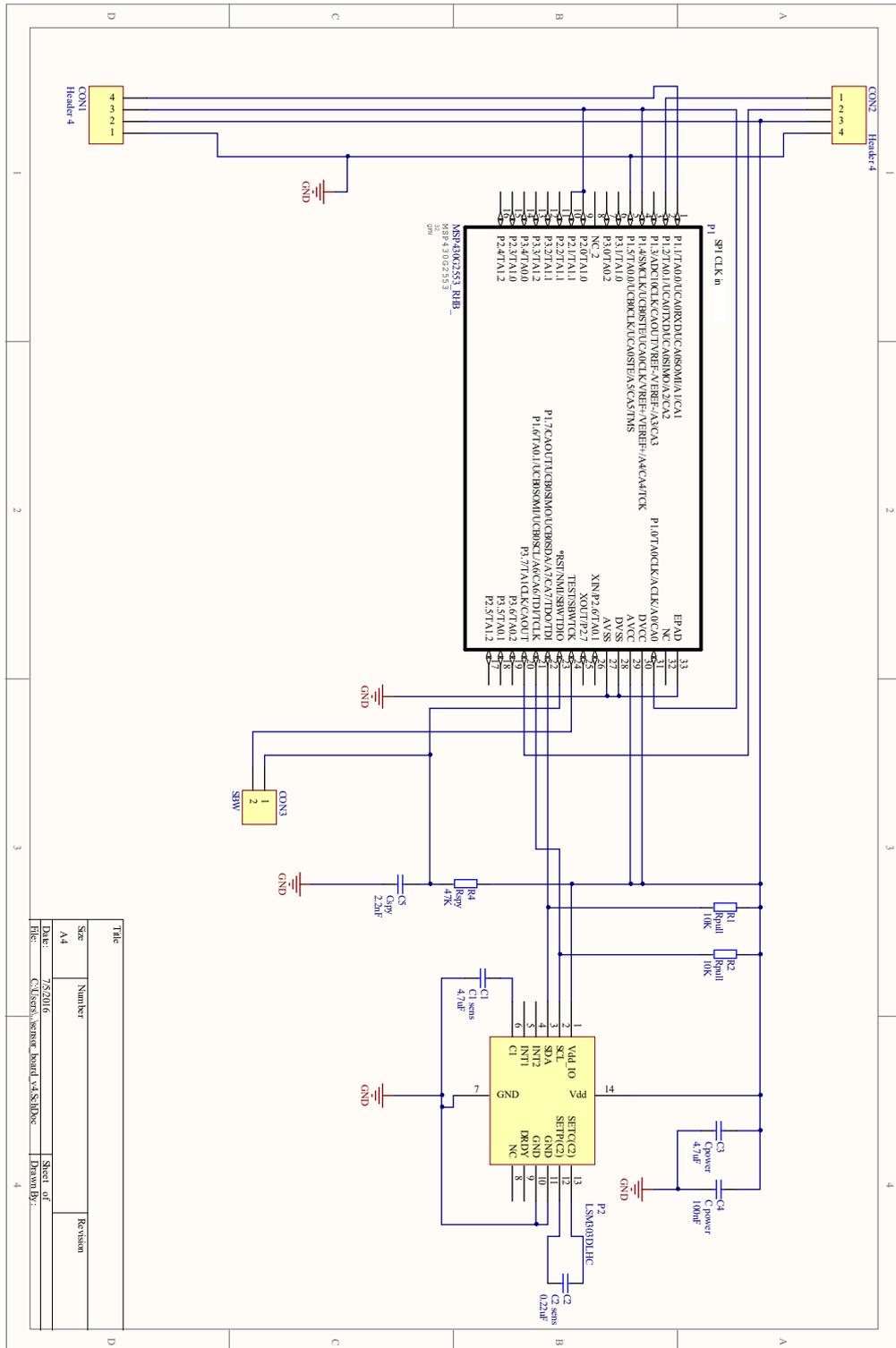
- "2nd International Doctoral School of Electrical Engineering and Power Electronics", 24-25 May 2013, Rõnisi, Latvia;
- "From exclusion to inclusion" Conference on novelties in Cerebral Palsy research, University of Latvia, May 8th 2014, Riga, Latvia;
- Latvijas Ergoterapeitu asociācijas vasaras konference 2014 "Ergoterapijas prakses kontekstu daudzveidība – mūsdienīgas metodes ergoterapijā", 14. jūnijs, 2014. g., rehab. centrs "Vaivari", Jūrmala, Latvija;
- Summer School "Smart Textiles for Healthcare", 25–28 August 2015, Riga, Latvia.

The work has been written in the Institute of Electronics and Computer Science and have been supported by following projects:

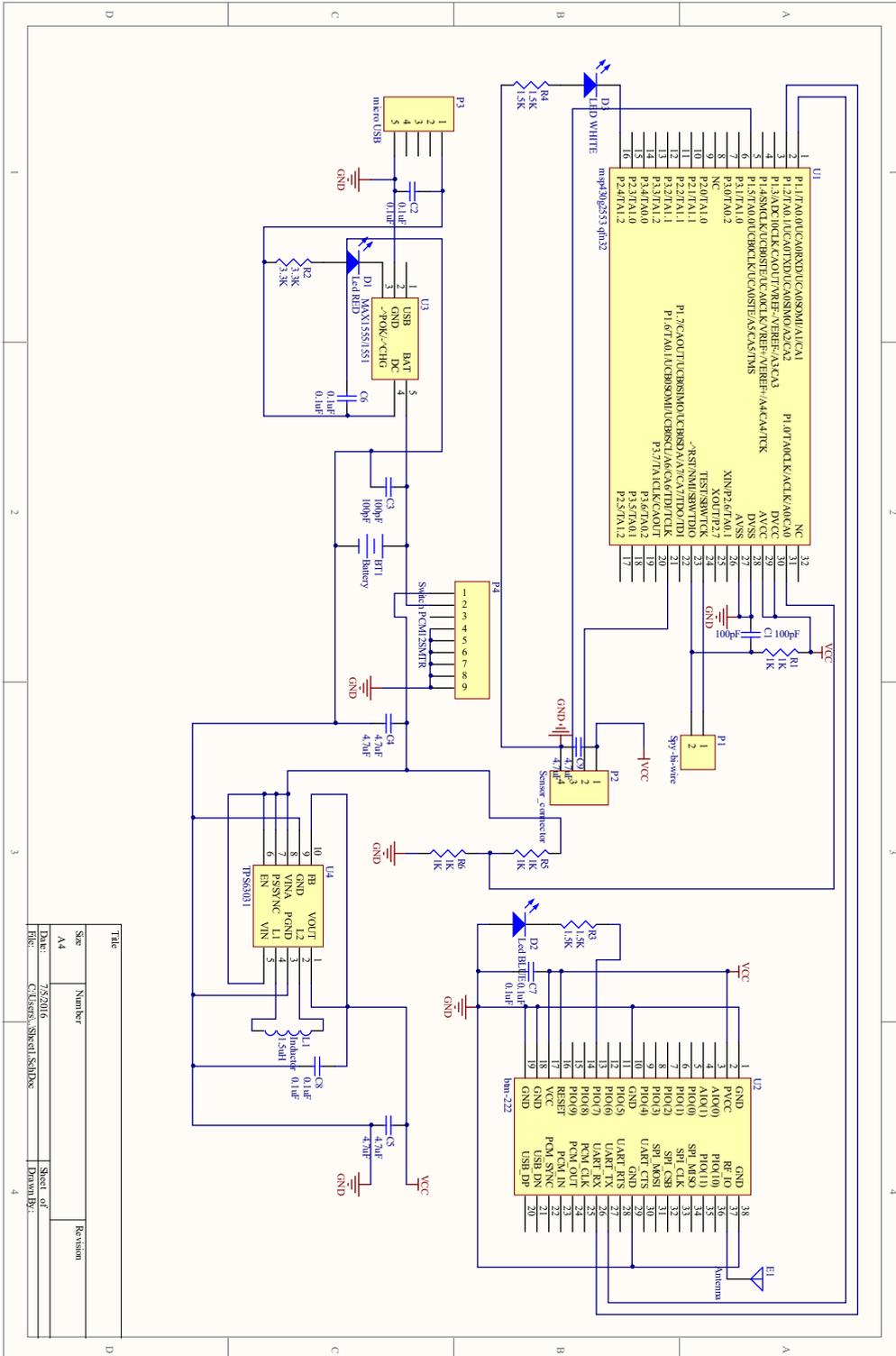
- National Research Program „Cyber-physical systems, ontologies and biophotonics for safe and smart city and society” (SOPHIS) project No. 1 „Development of technologies for cyber physical systems with applications in medicine and smart transport” (KiFiS);
- European Social Funds project "Smart City Technologies for Human Lives Improvements" (ViPTeh) No. 2013/0008/1DP/1.1.1.2.0/13/APIA/VIAA/016;
- State Research program "Development of innovative multi-functional material, signal processing and information technologies for competitive and research intensive products" project No.2 "Innovative signal processing technologies for smart and effective electronic system development" (IMIS).

APPENDICES

Schematics of LSM303DLHC sensor board



Schematics of master data acquisition board with Bluetooth



Title	Number	Revision
Sheet of		
Drawn By:		
File:	C:\Users\Shakil.Siddiqe	
Date:	7/5/2016	
Size	A4	

C code for MSP430g2553 of the sensor node

```

1 //SENSOR board code
2
3 //SPI slave daisy chain emulation for P1.3. First P1.4 defined as input. P1
4 // .4 interrupt (low-->high) initiates 6 byte data read from
5 // sensors LSM303DLHC through I2C. Data are stored in data_buffer. Then P1.4
6 // is switched to USCI_A CLK input and device
7 // acts as SPI shift register chain.
8
9 //includes
10 #include <msp430g2553.h>
11
12 //Function declarations*/
13 void init_all(); // initializes all modules needed before pulse
14 // measure
15 void init_receive_data(); // initializes all modules needed for receiving
16 // data, and sending
17 void P1_interrupt(void); // P1 interrupt procedure for sync
18 void i2c_write_to_register(char device_address, char address, char data);
19 // writes value to specified address to i2c device (sensor)
20 void i2c_receive_acc_data();// receives acceleration data from
21 // accelerometer
22 void i2c_receive_mag_data();// receive magnetometer data from sensor
23 void IO_port_to_CLK(void); //set IO Port to CLK (SPI CLK)
24 void CLK_to_IO_port(void); // set SPI CLK to IO port
25 void wait(volatile int a);
26
27 //void wait(volatile int); //delay function
28
29 //Defines
30 //acc setup
31 #define TURN_ON_ACC 0x27 // turn on all acc and enables all axes and
32 // 10 Hz ODR
33 #define ACC_ADDRESS 0x19 // accelerometer i2c device address
34 #define ACC_CTRL_REG1 0x20 //acc first control register address
35 #define ACC_FIRST_DATA_REGISTER 0x28 // acc first acc data register
36 // address
37 #define ACC_CTRL_REG4 0x23 // acc control register 4
38 #define ACC_HR 0x08 // accelerometer high resolution
39 // enable bit in ctrl reg4
40 #define ACC_TURN_ON 0x27 // 10 Hz ODR
41 #define ACC_AUTO_INC 0x80 // auto increment bit in subaddress
42
43 #define MAG_ADDRESS 0x1E // magnetometer i2c slave addr
44 #define MAG_FIRST_DATA_REGISTER 0x03 // mag first data register addr
45 #define MAG_CRA_REG_M 0x00 // mag set minimum data output rate
46 #define MAG_CRB_REG_M 0x01 // mag gain setting
47 #define MAG_MR_REG_M 0x02 // mag mode set to Continuous-conversion
48 // mode
49
50 #define BYTES_TO_TRANSFER 12 //Number of data bytes from each sensor

```

```

41 #define NUMBER_OF_SENSORS 63 //
42
43 // Global variables
44 unsigned int data_buffer[12]={0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0}; // here
    received data from accelerometers are stored.
45 int times_to_transfer_data=0; // indicator that shows how many times must
    transfer current sensor board controller
46
47 int sync=0; //synchronization indicator
48 unsigned int measured_interval=0;
49
50 //main.c
51 int main(void){
52     WDTCTL = WDIPW | WDTHOLD; // Stop watch dog timer
53
54     init_all();
55     wait(1000);
56     while (!(P1IN & BIT4)); // while clk line low wait until it sets high
57
58     P1SEL|=BIT4; // connect SPI module clk to P1.4
59     P1SEL2|=BIT4;
60
61     UCA0CTL1&=~UCSWRST; // set usci module in operational state
62
63     while (!(IFG2 & UCA0RXIFG)); // wait until we receive one full SPI byte
64     IFG2&=~UCA0RXIFG; // clear flag
65
66     // measure pulse
67     if(TA1CCR0>=TA1CCR1){
68         measured_interval=TA1CCR0-TA1CCR1;
69     } else {
70         measured_interval=0xFFFF-TA1CCR1+TA1CCR0;
71     }
72     // configure comparator
73
74     if(measured_interval>1){ // if measured pulse getting shorter
75         CACTL1=CAREL+CAREF_3+CAON; // diode reference
76         CACTL2=P2CA0;
77         CAPD=CAPD0;
78
79     } else {
80         CACTL1=CAREL+CAREF_2+CAON; // 1/2 Vcc
81         CACTL2=P2CA0;
82         CAPD=CAPD0;
83     }
84     P3DIR|=BIT7; // set comparator out to pin
85     P3SEL|=BIT7;
86     P3SEL2&=~BIT7;
87
88     while (!(P1IN & BIT1)); // waiting for pulse measure end indicator
89     P1OUT|=BIT2; // set SIMO in high state (passing pulse measure end
    indicator to other slaves)
90     while(P1IN & BIT1);
91     P1OUT&=~BIT2; // set SIM low state
92     // program main cycle
93

```

```

94 // configure spi, i2c and start receive data from accelerometers
95 // lower clk frequency
96 init_receive_data();
97
98 while(1){
99     int i;
100    int j;
101    CLK_to_IO_port(); //change P1.4 to input port.
102    while((P1IN & BIT4)); // wait for synchronization
103    while(!(P1IN & BIT4));
104    i2c_receive_acc_data(); //read sensor data
105    i2c_receive_mag_data(); // receive magnetometer data
106    IO_port_to_CLK();
107    for (j=0; j<NUMBER_OF_SENSORS;j++){
108        for (i=0; i<BYTES_TO_TRANSFER; i++){
109            while(!(IFG2 & UCA0TXIFG)); //check if TX buffer ready for new
data transmission
110            UCA0TXBUF=data_buffer[i]; // write TX data in buffer and wait for
master CLK to transmit
111            while(!(IFG2 & UCA0RXIFG)); //wait till input buffer ready (8 CLK
received from master)
112            data_buffer[i]=UCA0RXBUF; //read RX data from input buffer
113        }
114    }
115 }
116 }
117
118 void init_all(){
119     //init clk system
120     BCSCTL1=CALBC1_1MHZ;
121     DCOCTL=CALDCO_1MHZ;
122
123     // SETUP TIMER
124     TA1CTL=TASSEL_2+MC_2;
125     TA1CCTL0=CM_1+SCS+CAP; // rising edge, Binput, synchronize clk, capture
mode,
126     TA1CCTL1=CM_2+SCS+CAP; // falling edge,
127     P2DIR&=~(BIT0+BIT1);
128     P2SEL|=(BIT0+BIT1);
129     P2SEL2&=~(BIT0+BIT1);
130
131     // configure SPI
132     UCA0CTL1|=UCSWRST;
133     UCA0CTL0=UCCKPL+UCSYNC; // inactive state is high
134
135     // setup for pulse measure
136     P1DIR|=BIT2; // SIMO set as port output
137     P1OUT&=~BIT2; // simo output low
138     P3DIR&=(BIT7); //Comparator out clk' as output port
139     P3OUT&=~(BIT7); // output low
140 }
141
142 void init_receive_data(){
143     //Turn off timer
144     TA1CTL=MC_0; // STOP TIMER
145     TA1CCTL0=CM_0;

```

```

146 TA1CCTL1=CM_0;
147 //CLOCK SYSTEM
148 DCOCTL=CALDCO_1MHZ;
149 BCSCTL1=CALBC1_1MHZ;
150
151 //SPI USCIA0
152 UCA0CTL1|=UCSWRST;
153 P1SEL|=BIT1+BIT2;
154 P1SEL2|=BIT1+BIT2; //P1.1=SOMI, P1.2=SIMO, P1.4 = I/O (switch to CLK
later), P1.5=STE
155 UCA0CTL0 = UCCKPL + UCSYNC; //3 wire SPI slave. CLOCK inactive high
156 UCA0BR0=1;
157
158 //SYNC port config
159 P1SEL&=~ 0x10;
160 P1SEL2&=~ 0x10;
161 P1DIR &=~ 0x10; //10000 P1.4 input
162
163 //I2C USCIB0*/
164 UCB0CTL1|=UCSWRST; // put module in reset state
165 P1SEL|=BIT6+BIT7;
166 P1SEL2|=BIT6+BIT7; // P1.6=SCL (I2C) P1.7=SDA (I2C)
167 UCB0CTL1|=UCSSEL_3; // select SMCLK
168 UCB0CTL0|=UCMODE_3+UCMST; // set I2C MODE MASTER MODE
169 UCB0BR0=0x04; // set I2C clock prescaler 1MHZ/4 = 250 kHz
170
171 UCB0CTL1 &= ~UCSWRST; // start I2C
172 i2c_write_to_register(ACC_ADDRESS, ACC_CTRL_REG1, ACC_TURN_ON); // set
up accelerometer
173 i2c_write_to_register(ACC_ADDRESS, ACC_CTRL_REG4, ACC_HR); // turn on
high resolution
174
175 __delay_cycles(10000);
176 i2c_write_to_register(MAG_ADDRESS, MAG_CRA_REG_M, 0x10); // minimum
data output 15Hz (0x1C for max)
177 __delay_cycles(10000);
178 i2c_write_to_register(MAG_ADDRESS, MAG_CRB_REG_M, 0x20); // set
smallest gain
179 __delay_cycles(10000);
180 i2c_write_to_register(MAG_ADDRESS, MAG_MR_REG_M, 0x00); // set to
Continuous-conversion mode (turn on mag)
181
182 __bis_SR_register(GIE);
183 }
184
185 //I2C data transfer subroutines (no Richarda)*/
186 // sends one data byte to specified device
187 // arguments:
188 // device_address - target device slave address
189 // register_address - register address of sensor to which write data
190 // data - value that will be written to sensor register
191 void i2c_write_to_register(char device_address, char register_address, char
data){
192 UCB0I2CSA=device_address; //noradam ierices adresi ar
kuru komunicet

```

```

193 UCB0CTL1|=UCTR; //uzstadam I2C moduli raidisanas
    rezima
194 while(UCB0CTL1 & UCTXSTP); //gaidam, ja linija
    aiznemta
195 IFG2=~(UCB0TXIFG+UCB0RXIFG); //nodzesam sanemsana un
    nosutisanas karogus
196 UCB0CTL1|=UCTXSTT; //generejam STARTa nosacijumu
197 while((!(IFG2 & UCB0TXIFG))&&!(UCB0STAT & UCNACKIFG)); //gaidam kamer
    varam parraidit vai ir NACK
198 IFG2&=~UCB0TXIFG; //nodzesam parraidisanas karogu
199 if(!(UCB0STAT & UCNACKIFG)){ //ja nav bijis NACK
200     UCB0TXBUF=register_address; //nosutam akselerometra
    registra adresi,
201     //kura tiks sutiti dati
202     while((!(IFG2 & UCB0TXIFG))&&!(UCB0STAT & UCNACKIFG)); //gaidam kamer
    varam sutit, vai ir NACK
203     IFG2&=~UCB0TXIFG; //nodzesam sutisanas karogu
204     if(!(UCB0STAT & UCNACKIFG)){ //ja nav bijis NACK
205         UCB0TXBUF=data; //ierakstam nosutamo vertibu
    nositisanu buferi
206         while((!(IFG2 & UCB0TXIFG))&&!(UCB0STAT & UCNACKIFG)); //gaidam
    kamer sutisanas karogs pacelts
207         IFG2&=~(UCB0TXIFG); //nodzesam sanemsanas karogu
208     }
209     UCB0STAT&=~UCNACKIFG;
210     UCB0CTL1|=UCTXSTP; // beidzam I2C komunikaciju
211 }
212 }
213 void i2c_receive_acc_data() {
214     int i;
215     while(UCB0CTL1 & UCTXSTP); // gaidam STOP nosacijumu
216     UCB0I2CSA=ACC_ADDRESS; // uzstadam akselerometra i2c
    adresi
217     UCB0CTL1|=UCTR; // iestadam raidisanas rezima
218     IFG2&=~(UCB0TXIFG+UCB0RXIFG); // nodzesam visus
    partraukumu karogus
219     UCB0CTL1|=UCTXSTT; // generejam start nosacijumu
220     while((!(IFG2 & UCB0TXIFG))&&!(UCB0STAT & UCNACKIFG)); // gaidam kamer
    varam sutit datus vai ir NACK
221     IFG2&=~UCB0TXIFG; // nodzesam parraidisanas karogu
222     if(!(UCB0STAT & UCNACKIFG)){ // ja nav bijis NACK
223         UCB0TXBUF=(ACC_FIRST_DATA_REGISTER+ACC_AUTO_INC); // pirma registra
    adrese no kuras nolasit datus
224         while((!(IFG2 & UCB0TXIFG))&&!(UCB0STAT & UCNACKIFG)); //gaidam
    nosutisanas karogu vai NACK
225         if(!(UCB0STAT & UCNACKIFG)){ // ja nav bijis NACK
226             IFG2=~(UCB0TXIFG+UCB0RXIFG); // dzesam karogus
227             UCB0CTL1&=~UCTR; // uzstadam i2c moduli sanemsanas rezima
228             UCB0STAT&=~UCNACKIFG; // noddzesam NACK karogu
229             UCB0CTL1|=UCTXSTT; // generejam sakuma nosacijumu
230             while((!(IFG2 & UCB0RXIFG))&&!(UCB0STAT & UCNACKIFG)); // gaidam kad
    esam sanemusi datus vai NACK
231             if(!(UCB0STAT & UCNACKIFG)){ // parbaudam vai nav NACK
232                 //int i;
233                 for(i=0; i<6; i++){

```

```

234     while (!(IFG2 & UCB0RXIFG)); //gaidam kad dati sanemti sanemsanas
    buferi
235     IFG2&=~UCB0RXIFG;
236     data_buffer[i]=(UCB0RXBUF); //saglabajam sanemtus datus masiva
237     if(i==4){ // pec pirmspedeja baita uzstadam STOP nosacijumu
238         UCB0CTL1|=UCTXSTP; // i2c ipatnibas del
239     }
240     }
241     return; // izejam no funkcijas neipildot talako kodu
242 }
243 }
244 }
245
246 // ja bijis NACK
247 //int i;
248 for(i=0;i<6;i++){
249     data_buffer[i]=0; //nodzesam paatrinajuma datu buferi
250 }
251 UCB0STAT&=~UCNACKIFG; // nodzesam NACK karogu
252 UCB0CTL1|=UCTXSTP; // generejam STOP nosacijumu
253 }
254
255 void i2c_receive_mag_data() {
256     int i;
257     while(UCB0CTL1 & UCTXSTP); // gaidam STOP nosacijumu
258     UCB0I2CSA=MAG_ADDRESS; // uzstadam akselerometra i2c
        adresi
259     UCB0CTL1|=UCTR; // iestadam raidisanas rezima
260     IFG2&=~(UCB0TXIFG+UCB0RXIFG); // nodzesam visus
        partraukumu karogus
261     UCB0CTL1|=UCTXSTT; // generejam start nosacijumu
262     while (!(IFG2 & UCB0TXIFG) && !(UCB0STAT & UCNACKIFG)); // gaidam kamer
        varam sutit datus vai ir NACK
263     IFG2&=~UCB0TXIFG; // nodzesam parraidisanas karogu
264     if (!(UCB0STAT & UCNACKIFG)) { // ja nav bijis NACK
265         UCB0TXBUF=(MAG_FIRST_DATA_REGISTER); //+ACC_AUTO_INC (mag has
        auto increment)
266         while (!(IFG2 & UCB0TXIFG) && !(UCB0STAT & UCNACKIFG)); // gaidam
        nosutisanas karogu vai NACK
267         if (!(UCB0STAT & UCNACKIFG)) { // ja nav bijis NACK
268             IFG2&=~(UCB0TXIFG+UCB0RXIFG); // dzesam karogus
269             UCB0CTL1&=~UCTR; // uzstadam i2c moduli sanemsanas rezima
270             UCB0STAT&=~UCNACKIFG; // noddzesam NACK karogu
271             UCB0CTL1|=UCTXSTT; // generejam sakuma nosacijumu
272             while (!(IFG2 & UCB0RXIFG) && !(UCB0STAT & UCNACKIFG)); // gaidam kad
        esam sanemusi datus vai NACK
273             if (!(UCB0STAT & UCNACKIFG)) { // parbaudam vai nav NACK
274                 //int i;
275                 for(i=6; i<12; i++){
276                     while (!(IFG2 & UCB0RXIFG)); //gaidam kad dati sanemti sanemsanas
        buferi
277                     IFG2&=~UCB0RXIFG;
278                     data_buffer[i]=(UCB0RXBUF); //saglabajam sanemtus datus masiva
279                     if(i==10){ // pec pirmspedeja baita uzstadam STOP nosacijumu
280                         UCB0CTL1|=UCTXSTP; // i2c ipatnibas del
281                     }

```

```

282     }
283     return; // izejam no funkcijas neipildot talako kodu
284 }
285 }
286 }
287
288 // ja bijis NACK
289 for(i=6;i<12;i++){
290     data_buffer[i]=0;//nodzesam paatrinajuma datu buferi
291 }
292 UCB0STAT&=~UCNACKIFG;// nodzesam NACK karogu
293 UCB0CTL1|=UCTXSTP; // generejam STOP nosacijumu
294 }
295
296 void IO_port_to_CLK() {
297     P1SEL|= BIT4; //set P1.4 SEL registers for SPI CLK pin
298     P1SEL2|= BIT4;
299     UCA0CTL1 &= ~UCSWRST; //run USCI_A
300 }
301
302 void CLK_to_IO_port() {
303     UCA0CTL1 |= UCSWRST; //stop USCI_A
304     P1SEL&=~ BIT4; //clear P1.4 SEL registers of I/O pin
305     P1SEL2&=~ BIT4;
306 }
307 void wait(volatile int a){
308     while(a>0){
309         a--;
310     }
311 }

```

C code for MSP430g2553 of the master device

```

1 #include <msp430g2553.h>
2
3 void init_all();
4 void set_p22_as_output();
5 void wait(volatile unsigned int cycles);
6 void set_CLK_as_CLK();
7 void set_CLK_as_output();
8 void set_MISO_as_output();
9 void set_MISO_as_MISO();
10 void sync(volatile int cycles);
11 void send_spi_byte(char Data);
12 char receive_spi_byte();
13 char receive_spi_byte();
14 void SerialPacketStart(char type);
15 void SendByteToSerial(unsigned char bit);
16 void SerialPacketEnd();
17 void myTimer(void);
18 void SerialPacketPushInt(unsigned int content);
19 void SerialPacketPushByte(unsigned char content);
20
21 #define BYTES_TO_TRANSFER 12 //Number of bytes for master to receive (from
    one sensor)
22 #define NUMBER_OF_SENSORS 63 //Number of sensors in chain
23 #define BATTERY_LVL_PACKET 64 // Packet containing battery level
24
25 // Serial packets
26 #define PACKET_FRAME 0xFF
27 #define PACKET_ESCAPE 0xFE
28 #define LOW_BYTE 0x00
29 #define HIGH_BYTE 0x01
30 #define XX 0x00
31 #define YY 0x01
32 #define ZZ 0x02
33
34 // flags
35 int receive_data_flag=0;
36 int nr_of_timer_interrupts=0;
37 int nr_of_adc_interrupts=0;
38 char accelerometer_data[2][3];
39 char magnetometer_data[2][3];
40 unsigned int raw_battery_level=0xFFFF;
41 unsigned int low_battery_threshold=828; // 3.4V battery voltage
42 unsigned int adc_inrterupt_counter=0;
43
44 #pragma vector = TIMER1_A1_VECTOR
45 __interrupt void myTimer(void) {
46     switch(TA1IV) {
47         case 0x0A:
48             nr_of_timer_interrupts++;

```

```

49     if(nr_of_timer_interrupts==10){ //timer interrupt frequency 100 Hz,
receive_data_flag sets with frequency 100/10=10 Hz frequency;
50         receive_data_flag=1;
51         nr_of_timer_interrupts=0;
52     }
53     break;
54 default:
55     break;
56 }
57 }
58 // interrupt for ADC converter. sampling time 2*accelerometer data sampling
rate (ACC Data dample 100ms, ADC for battery level 200 ms)
59 #pragma vector = ADC10_VECTOR
60 __interrupt void myADC(void){
61     raw_battery_level=ADC10MEM & 0x3FF;
62     nr_of_adc_interrupts++;
63     if(nr_of_adc_interrupts==14){
64         nr_of_adc_interrupts=0;
65         if(raw_battery_level<low_battery_threshold){
66             P2OUT^=BIT4; // blink led if battery low
67         }
68     }
69 }
70
71 //main.c
72
73 int main(void) {
74     WDTCTL = WDIPW | WDIHOLD; // Stop watchdog timer
75     init_all();
76     P1OUT&=~(BIT6+BIT5); // set output low
77
78     set_CLK_as_output();
79     set_MISO_as_output(); //
80
81     wait(60000); // wait time to allow slaves to set up
82     set_CLK_as_CLK(); // set CLK pin as SPI CLK
83     int sensors_to_calibrate = NUMBER_OF_SENSORS; // how many sensors
must calibrate
84     wait(60000); // wait to ensure sensor ready
85     while(sensors_to_calibrate>0){
86         while(!(IFG2 & UCB0TXIFG)); // wait until we can send spi data
87             UCB0TXBUF = 0x00; // send empty byte to generate clk
88             sensors_to_calibrate--;
89             wait(60000); //wait to give slave controllers time for
impulse measurements and comparator configuration
90     }
91
92     P1OUT|=BIT6; // set MISO port in high state , to tell slaves that
measure mode ended
93     wait(60000); // wait to ensure that sensors detects this pulse
94     P1OUT&=~BIT6; // set MISO port low state
95     wait(60000);
96     // set SPI module
97     set_MISO_as_MISO();
98     TA1CTL|=MC_1; // run timer
99     TA0CTL|=MC_1; // run timer

```

```

100 ADC10CTL0|=ENC; // ENABLE ADC CONVERSION
101 // reconfiguring for sensor data acquisition
102 _enable_interrupts ();
103
104 while (1) {
105     P1OUT|=BIT5; // SPI CLK output high
106     set_CLK_as_output (); // set CLK pin as Output
107     while (receive_data_flag==0); // wait until timer sets flag
108     receive_data_flag=0; // clear receive data flag
109
110     sync (100);
111     set_CLK_as_CLK (); // set clk pin to SPI clk module
112     wait (2000); // 956 wait to allow sensors read data
113     int i, j;
114     for (i=0; i<NUMBER_OF_SENSORS; i++) { // for all sensor data
115         for (j=0; j<3; j++) {
116             IFG2&=~UCB0RXIFG; // clear receive flag because can be raised
117             from pulse measure
118             send_spi_byte (0x00);
119             accelerometer_data [LOW_BYTE][j]=receive_spi_byte ();
120             wait (69); // 69
121             send_spi_byte (0x00);
122             accelerometer_data [HIGH_BYTE][j]=receive_spi_byte ();
123             wait (69); // 69
124             }
125             for (j=0; j<3; j++) {
126                 raised from pulse measure
127                 send_spi_byte (0x00);
128                 magnetometer_data [HIGH_BYTE][j]=receive_spi_byte ();
129                 wait (69); // 69
130                 send_spi_byte (0x00);
131                 magnetometer_data [LOW_BYTE][j]=receive_spi_byte ();
132                 wait (69); // 69
133             }
134             SerialPacketStart (i);
135             SerialPacketPushByte (accelerometer_data [HIGH_BYTE][XX]); // X
136             data
137             SerialPacketPushByte (accelerometer_data [LOW_BYTE][XX]); // X
138             data
139             SerialPacketPushByte (accelerometer_data [HIGH_BYTE][YY]); // Y
140             data
141             SerialPacketPushByte (accelerometer_data [LOW_BYTE][YY]); // Y
142             data
143             SerialPacketPushByte (accelerometer_data [HIGH_BYTE][ZZ]); // Z
144             data
145             SerialPacketPushByte (accelerometer_data [LOW_BYTE][ZZ]); // Z
146             data
147             SerialPacketPushByte (magnetometer_data [HIGH_BYTE][XX]); // X
148             data
149             SerialPacketPushByte (magnetometer_data [LOW_BYTE][XX]); // X data
150             SerialPacketPushByte (magnetometer_data [HIGH_BYTE][ZZ]); // Y
151             data (SWAPPING X AND Y DATA because registers in sensors X Z Y )
152             SerialPacketPushByte (magnetometer_data [LOW_BYTE][ZZ]); // X data

```

```

145         SerialPacketPushByte(magnetometer_data[HIGH_BYTE][YY]); // Z
146     data
147         SerialPacketPushByte(magnetometer_data[LOW_BYTE][YY]); // +X
148     data
149         SerialPacketEnd();
150     }
151     // send battery status packet;
152     SerialPacketStart(BATTERY_LVL_PACKET);
153     SerialPacketPushByte((raw_battery_level & 0xFF00)>>8); //
154     battery level high byte
155     SerialPacketPushByte(raw_battery_level & 0x00FF); // battery
156     level low byte
157     SerialPacketPushByte(100);
158     SerialPacketPushByte(100);
159     SerialPacketPushByte(100);
160     SerialPacketPushByte(100);
161     SerialPacketPushByte(100);
162     SerialPacketPushByte(100);
163     SerialPacketPushByte(100);
164     SerialPacketPushByte(100);
165     SerialPacketEnd();
166 }
167 }
168
169 void init_all() {
170     // init clk module
171     P2DIR|=BIT4;
172     P2OUT|=BIT4;
173     DCOCTL=CALDCO_16MHZ; // SET DCO 16 MHZ and
174     BCSCTL1=CALBC1_16MHZ+XT2OFF; // XT2OFF, MCLK = 16 MHZ, SMCLK = 16 MHZ
175
176     // configure USCIB0 module SPI mode
177     UCBOCTL1=UCSWRST+UCSSEL_2; // stop usci module, SMCLK clk source
178     UCBOCTL0=UCCKPL+UCMST+UCSYNC; // CLK inactive high, LSB, 3-wire mode
179     UCBOBR0=0x25; // 125 kHz SPI CLK
180     UCBOBR1=0x00;
181
182     // init usci uart
183     UCA0CTL1=UCSSEL_2+UCSWRST;
184     // konfigurejam BAUD RATE GENERATOR
185     UCA0BR0=2;
186     UCA0MCTL=UCOS16+(2<<4)+(3<<1);
187     // port configuration
188     P1SEL|=(BIT1+BIT2);
189     P1SEL2|=(BIT1+BIT2);
190     UCA0CTL1&=~UCSWRST; // NODZESAM RESETU
191
192     // setup timer
193     TA1CTL=TASSEL_2+ID_3+TAIE;
194     TA1CCR0=20000; // timer TA1CCR0 interrupt frequency SMCLK/(8*20000) =
195     100 Hz

```

```

196 // CONFIGURE ADC
197 ADC10CTL0=SREF_1+ADC10SHT_3+REF2_5V+REFON+ADC10ON+ADC10IE; // !!! in
main() must enable conversion
198 ADC10CTL1=SHS_2+ADC10DIV_7+ADC10SSEL_3+CONSEQ_2;
199 ADC10AE0=BIT0; // ENABLE INPUT
200
201 // ADC timer for conversion control
202 TA0CTL=TASSEL_2+ID_3;
203 TA0CCR0=20000; // timer TA1CCR1
204 TA0CCTL0=OUTMOD_4; // toggle mode timer output for ADC conversion
timing
205 }
206 // sets P2.2 function to GPIO output
207 void set_CLK_as_output() {
208     P1DIR|=BIT5; // set P1.5 as output
209     P1OUT|=BIT5;
210     P1SEL&=~BIT5; // set P1.5 pin as I/O port
211     P1SEL2&=~BIT5; // set P1.5 pin as I/O port
212     UCB0CTL1|=UCSWRST;
213 }
214
215 // sets P1.5 pin function to SPI CLK
216 void set_CLK_as_CLK() {
217     UCB0CTL1&=~UCSWRST;
218     P1SEL|=BIT5;
219     P1SEL2|=BIT5;
220 }
221 // set P1.7 as MISO
222 void set_MISO_as_output() {
223     P1SEL&=~BIT6; // set P1.7 pin as GPIO
224     P1SEL2&=~BIT6; // -----//-----
225     P1DIR|=BIT6; // set P1.7 as output
226 }
227 void set_MISO_as_MISO() {
228     P1SEL|=BIT6;
229     P1SEL2|=BIT6;
230 }
231 void wait(volatile unsigned int cycles){
232     while(cycles>0){
233         cycles--;
234     }
235 }
236
237 // function drops P2.2 output to low for some time and sets back as high
238 // int cycles – determines how many while cycles hold the OUTPUT low.
239 void sync(volatile int cycles){
240     P1OUT&=~BIT5;
241     while(cycles>0){
242         cycles--;
243     }
244     P1OUT|=BIT5;
245 }
246
247 // sends one byte via spi
248 //char Data – is data byte to be transmitted
249 void send_spi_byte(char Data){

```

```

250 while (!(IFG2 & UCB0TXIFG)); // wait until transmit flag is raised
251 //UCB0IFG&=~UCTXIFG; // clear transmit interrupt flag
252 UCB0TXBUF=Data; // send byte through i2c
253 }
254 // receives one data byte from SPI
255 // returns received byte
256 char receive_spi_byte() {
257 while (!(IFG2 & UCB0RXIFG));
258 IFG2&=~UCB0RXIFG;
259 return UCB0RXBUF;
260 }
261
262 //-----serial communication module functions
263 void SendByteToSerial(unsigned char bit){
264 while (!(IFG2 & UCA0TXIFG));
265 IFG2&=~UCA0TXIFG;
266 UCA0TXBUF = bit;
267 }
268 // sends integer via serial
269 void SendToSerial(int num){
270 //Send as number:
271 int MSB, LSB;
272 MSB=(num>>8)&0xFF;
273 LSB=(num&0xFF);
274
275 SendByteToSerial(MSB); // send MSB
276 SendByteToSerial(LSB); // send MSB
277 }
278
279 void SerialPacketStart(char type){ // Start sending a serial packet with
280 given type (must not match FRAME or ESCAPE constants)
281 SendByteToSerial(PACKET_FRAME);
282 SendByteToSerial(type);
283 }
284
285 void SerialPacketPushInt(unsigned int content){ //Send contents of packet
286 to serial
287 SerialPacketPushByte((content & 0xFF00)>>8);
288 SerialPacketPushByte(content & 0x00FF);
289 }
290
291 //
292 //
293 //
294 //
295 //
296 //
297 //
298 //
299 //
300 //
301 //
302
303

```

```
304
305
306 void SerialPacketPushByte(unsigned char content){ //Send contents of packet
      to serial
307     if(content == PACKET_FRAME) {
308         SendByteToSerial(PACKET_ESCAPE);
309         SendByteToSerial(0x00);
310     } else if (content == PACKET_ESCAPE){
311         SendByteToSerial(PACKET_ESCAPE);
312         SendByteToSerial(0x01);
313     } else {
314         SendByteToSerial(content);
315     }
316 }
317
318 void SerialPacketEnd(){ //End sending a serial packet
319     SendByteToSerial(PACKET_FRAME);
320 }
```

Main Matlab code of shape reconstruction algorithm

```

1  %{
2  Main script of accelerometer and magnetometer sheet reconstruction
3  algorithm. Segment rotation using rotation matrix.
4  Atis Hermanis EDI
5  %{
6
7  clear all;
8  clc;
9
10 #####SETUP#####
11
12 %load_calibration_data
13 load calibration_data02 -Jun-2016_500_samples.mat
14
15 %Number of sensors and reference setup:
16 nr_of_rows=9;
17 nr_of_columns=7;
18 start_column=3;           %Reference sensor:
19 start_row=4;
20
21 %Declaration of segment class
22 segment(nr_of_rows,nr_of_columns)=segment_class;
23 init_cross_points=[0 0 1 ; 1 0 0; 0 0 -1 ; -1 0 0];
24
25 %Sheet model definition and construction
26 for rows=1:1:nr_of_rows
27     for columns=1:1:nr_of_columns
28         XX(rows, columns)=segment(rows,columns).center(1);
29         YY(rows, columns)=segment(rows,columns).center(2);
30         ZZ(rows,columns)=segment(rows,columns).center(3);
31         CC(rows,columns)=10;
32     end
33 end
34 model=surf(XX,YY,ZZ, CC);
35
36 set(model, 'FaceLighting', 'phong', 'FaceColor', 'interp', 'AmbientStrength',
37     ,0.5)
38 light('Position',[-1 -1 1], 'Style','infinite');
39 xlabel('X')
40 ylabel('Y')
41 zlabel('Z')
42 axis([-10 10 -10 10 -10 10])
43 axis vis3d
44 rotate3d on
45 grid on
46 hold on
47 %-----pushbutton to terminate application on graph corner-----
48 H = uicontrol('Style', 'PushButton', 'String', 'Break', 'Callback', 'delete(
    gcbf)');

```

```

49 %-----Kalman filter Setup-----
50 %Accelerometer:
51 delta_acc=1; %kalman DELTA
52     koeficient for accelerometer data
53 Q_acc=0.2; %kalman Q
54     koeficient for accelerometer data
55 P_xyz_acc=ones(nr_of_rows*nr_of_columns, 3); %Init acc data P
56     gain array
57 AccData_filter=zeros(nr_of_rows*nr_of_columns, 3); %Init acc data
58     filtered (preveous) value array
59
60 delta_magn=delta_acc; %kalman DELTA
61     koeficient for accelerometer data
62 Q_magn=Q_acc; %kalman Q
63     koeficient for accelerometer data
64 P_xyz_magn=ones(nr_of_rows*nr_of_columns, 3); %Init acc data P
65     gain array
66 MagnData_filter=zeros(nr_of_rows*nr_of_columns, 3); %Init acc data
67     filtered (preveous) value array
68
69 %-----SETUP PREALTERM SERVER-----
70 [hrealterm, fileID]=Realterm_open('3'); %this provides communication with
71 Bluetooth through virtual com port
72
73 %#####Main cycle#####
74 while (ishandle(H))
75
76     %GET DATA
77     [acc, magn]=getPureAccMagnData(fileID, 64); %read data Acc/Magn data
78     from incoming Bluetooth buffer
79
80     %CALIBRATE magnetometer DATA
81     magn=MagnCalibrate(magn, magnOffsets, linear);
82
83     %FILTER DATA
84     [acc, P_xyz_acc]=kalman_filter(acc, AccData_filter, P_xyz_acc, Q_acc,
85     delta_acc); %apply kalman filter for accelerometer data
86     AccData_filter=acc; %store filtered value as previous for the
87     next iteration
88
89     [magn, P_xyz_magn]=kalman_filter(magn, MagnData_filter, P_xyz_magn,
90     Q_magn, delta_magn); %apply kalman filter for magnetometer data
91     MagnData_filter=magn; %store filtered value as previous for the
92     next iteration
93
94     %ARRANGE AXIS (convert sensor reference system to model reference
95     system)
96     [AccData, MagnData]=Arrange_sensor_axis(acc, magn, nr_of_rows,
97     nr_of_columns);
98
99     %ROTATE SEGMENTS
100     for rows=1:1:nr_of_rows
101         for columns=1:1:nr_of_columns
102             Rrot=calculate_Rrot(squeeze(AccData(rows, columns, :))),
103             squeeze(MagnData(rows, columns, :))); %calculate the rotation data with

```

```

88     TRIAD algorithm
89         for k=1:1:2
90             segment(rows, columns).cross_points(k,:)=Rrot*
init_cross_points(k,:)';
91             segment(rows, columns).cross_points(k+2,:)=segment(rows,
columns).cross_points(k,:);
92         end
93     end
94 end
95 %—————SEGMENT LOCATION (CENTER) CALCULATION—————
96
97 %ALONG THE COLUMNS!!!!
98 %Reference row:
99 to_left=start_column;
100 while(to_left<nr_of_columns)
101     to_left=to_left+1;
102     segment(start_row, to_left).center_temp=segment(start_row, to_left
-1).center_temp+segment(start_row, to_left-1).cross_points(4,:)+segment(
start_row, to_left).cross_points(4,:);
103 end
104
105 to_right=start_column;
106 while(to_right>1)
107     to_right=to_right-1;
108     segment(start_row, to_right).center_temp=segment(start_row,
to_right+1).center_temp+segment(start_row, to_right+1).cross_points(2,:)+
segment(start_row, to_right).cross_points(2,:);
109 end
110
111 %Other segments:
112 for columns=1:1:nr_of_columns
113     to_bottom=start_row;
114     while(to_bottom>1)                                     %calculate each row one
by one in down direction
115         to_bottom=to_bottom-1;
116         segment(to_bottom, columns).center_temp=segment(to_bottom+1,
columns).center_temp+segment(to_bottom+1, columns).cross_points(3,:)+
segment(to_bottom, columns).cross_points(3,:);
117     end
118
119     to_top=start_row;
120     while(to_top<nr_of_rows)                               %calculate each
row one by one in up direction
121         to_top=to_top+1;
122         segment(to_top, columns).center_temp=segment(to_top-1, columns)
.center_temp+segment(to_top-1, columns).cross_points(1,:)+segment(to_top
, columns).cross_points(1,:);
123     end
124 end
125
126 %ALONG THE ROWS!!!!
127 %Reference column:
128 to_bottom=start_row;
129 while(to_bottom>1)                                       %calculate each column one
by one in down direction

```

```

130     to_bottom=to_bottom-1;
131     segment(to_bottom, start_column).center=segment(to_bottom+1,
start_column).center+segment(to_bottom+1, start_column).cross_points
(3,:)+segment(to_bottom, start_column).cross_points(3,:);
132     end
133
134     to_top=start_row;
135     while(to_top<nr_of_rows)                                %calculate each
column one by one in up direction
136         to_top=to_top+1;
137         segment(to_top, start_column).center=segment(to_top-1, start_column
).center+segment(to_top-1, start_column).cross_points(1,:)+segment(
to_top, start_column).cross_points(1,:);
138     end
139
140
141     %Other segments:
142     for rows=1:1:nr_of_rows
143         to_left=start_column;
144         while(to_left<nr_of_columns)
145             to_left=to_left+1;
146             segment(rows, to_left).center=segment(rows, to_left-1).center+
segment(rows, to_left-1).cross_points(4,:)+segment(rows, to_left).
cross_points(4,:);
147         end
148
149         to_right=start_column;
150         while(to_right>1)
151             to_right=to_right-1;
152             segment(rows, to_right).center=segment(rows, to_right+1).center
+segment(rows, to_right+1).cross_points(2,:)+segment(rows, to_right).
cross_points(2,:);
153         end
154
155     end
156
157     %Obtain center as the avarage of two methods
158     for rows=1:1:nr_of_rows
159         for columns=1:1:nr_of_columns
160             segment(rows, columns).center=(segment(rows, columns).center+
segment(rows, columns).center_temp)/2;
161         end
162     end
163
164     %Prepare data for model update
165     for rows=1:1:nr_of_rows
166         for columns=1:1:nr_of_columns
167             XX(rows, columns)=segment(rows, columns).center(1);
168             YY(rows, columns)=segment(rows, columns).center(2);
169             ZZ(rows, columns)=segment(rows, columns).center(3);
170         end
171     end
172
173     set(model, 'XData',XX);
174     set(model, 'YData',YY);
175     set(model, 'ZData',ZZ);

```

```

176
177     drawnow %refresh model
178 end
179 Realterm_close(hrealterm , fileID); %close realterm server
180
181 classdef segment_class
182     %segment_class encapsulates all data and properties of surface segments
183     % Detailed explanation goes here
184
185     properties
186         cross_points=[0 0 1 ; 1 0 0 ; 0 0 -1 ; -1 0 0]; %segmentu
187         apzimejoso krustu galapunkti %
188         1 %
189         4--+-2 %
190         3 %
191         center_temp=zeros(1,3);
192         center=zeros(1,3);
193     end
194 end
195
196 %Implementation of TRIAD algorithm
197 function [ Rrot ] = calculate_Rrot(Acc_data , Magn_data)
198     ba=Acc_data/norm(Acc_data);
199     bm=Magn_data/norm(Magn_data);
200
201     Eg=[0 0 1];
202     Em=[0.4472 0 -0.8944];
203
204     s1=Eg;
205     s2=cross(Eg, Em)/norm(cross(Eg, Em));
206     s3=cross(s1, s2);
207
208     r1=ba;
209     r2=cross(ba, bm)/norm(cross(ba, bm));
210     r3=cross(r1, r2);
211
212     Mmea=[r1', r2', r3'];
213     Mref=[s1', s2', s3'];
214
215     Rrot=Mref*transpose(Mmea);
216 end
217
218 function [ acc_data , magn_data ] = Arrange_sensor_axis(acc , magn , rows ,
219     columns)
220 %Arrange_sensor_axis used to virtually align sensor physical orientation
221 %with sensor orientation in 3D model
222 %returns 2d arrays where each element is vector with xyz sensor data.
223 for i=1:columns
224     for j=1:rows
225         if rem(i, 2)~=0
226             acc_index=(rows*(i-1)+j);
227             acc_data(j, i, 1)=acc(acc_index, 1); % X data

```

```

227     acc_data(j,i,2)=acc(acc_index,3); % Y data
228     acc_data(j,i,3)=-acc(acc_index,2); % Z data
229     magn_data(j,i,1)=magn(acc_index,1); % X data
230     magn_data(j,i,2)=magn(acc_index,3); % Y data
231     magn_data(j,i,3)=-magn(acc_index,2); % Z data
232 else
233     acc_index=rows*i-(j-1);
234     acc_data(j,i,1)=-acc(acc_index,1); % X data
235     acc_data(j,i,2)=acc(acc_index,3); % Y data
236     acc_data(j,i,3)=acc(acc_index,2); % z data
237     magn_data(j,i,1)=-magn(acc_index,1); % X data
238     magn_data(j,i,2)=magn(acc_index,3); % Y data
239     magn_data(j,i,3)=magn(acc_index,2); % Z data
240 end
241 end
242 end
243 end

```

"Unihaus" ltd. approbation review

SIA "UniHaus"
 Bruņinieku 5/25, Rīga, LV-1001
 Reģ. Nr. 40003931388
 A/S Hansabanka, kods HABALV22
 Konts: LV50HABA0551017155701

Atsauksme par sistēmas „AccGrid” eksperimentālā maketa aprobāciju

Esam aprobējuši Elektronikas un datorzinātņu institūta piedāvāto sensoru tīkla sistēmu „AccGrid” stājas monitorēšanas pielietojumam. Aprobācijas mērķis bija novērtēt iekārtas lietderīgumu ortopēdijas un fizioterapijas sfērā. Aprobācija tika veikta reālos apstākļos ārsta klātbūtnē, strādājot ar pusaudžu (13-15) vecuma pacientiem, kuriem ir izteikti stājas traucējumi.

Aprobācijas gaitā tika ieteikti virkne nepieciešamo uzlabojumu, kurus Elektronikas un datorzinātņu institūta darbinieki ņēma vērā, veicot modificēšanu. Piemēram, pēc SIA „UniHaus” speciālistu iniciatīvas sensoru tīkls tika iešūts sporta krekļā ar rāvējslēdzēju, lai vizuāli sensori būtu mazāk pamanāmi un neradītu lietotājam psiholoģisku diskomfortu, kā arī, lai nodrošinātu sensoru ciešāku kontaktu ar ķermeni kustību laikā. Lai arī Elektronikas un datorzinātņu institūta izstrādātā lietotne ir parocīga un saprotama, ņemot vērā mūsu komentārus, tika mainīts 3D modeļa noformējums tā, lai tas mūsu jomas lietotājam būtu saprotamāks.

Vēlamies atzīmēt, ka Institūta speciālists piedalījās arī fizioterapijas nodarbībās ar pacientiem, kura jau vairākus gadus valkā cieto korseti skoliozes korekcijai. Pēc fizioterapeites Alīnas Šilvānes ieteikumiem lietotnei tika pievienota jauna funkcija - iespēja fizioterapeitam saglabāt pacienta muguras formas attēlu (vai koordinātas) pirms un pēc nodarbības, tādā veidā salīdzinot izmaiņas, kas notiek tieši fizioterapijas nodarbības rezultātā.

Iekārtas potenciālais pielietojums var būt lietošana pacientam mājās patstāvīgi. To var izmantot pacienti, kuriem pēc cietās korsetes valkāšanas perioda ir pašiem, ar savu muskuļu spēku, jāiemācās noturēt mugurkauls pareizā pozīcijā. Kā arī pacientiem, kam muguras deformācija tiek ārstēta bez korsetes, bet uzdevums ir noturēt koriģētu muguras pozīciju.

Cita potenciālo lietotāju grupa ir bērni un jaunieši, kuriem ir novērojamas stājas deformācijas vai stājas vājums. Šī jauniešu grupa ar iekārtas palīdzību var iemācīties noturēt stāju pareizā pozīcijā.

Pasaules praksē pielieto ķermeņa 3D skenerus lai iegūtu muguras telpisku attēlu, tādēļ redzam, ka Elektronikas un datorzinātņu institūta piedāvātā sistēma var kļūt tiem par alternatīvu ar būtiski mazākām izmaksām. Minētā ierīce līdz ar to var tikt pielietota pacientu izmeklēšanā, lai iegūtu muguras formas attēlu/koordinātas, kā arī paveras iespēja pacientam to lietot mājās kā „vingrojumu komplektu”, lai kontrolētu vēlamo ķermeņa pozīciju. Iekārta ir inovatīva un tai ir potenciāls, lai nākotnē sniegtu piensumu Latvijas tautsaimniecībai.



Gundars Rusovs
 Ārsts, tehniskais ortopēds
 SIA UniHaus
 18.10.2013

SIA "UniHaus"
 Ārsts - tehniskais ortopēds
 Gundars Rusovs
 1-040168-12256

SIA "UniHaus"

Reg. Nr. 40003931388; A/S Hansabanka, kods: HABALV22; konts LV50HABA0551017155701
 Bruņinieku iela 5/29, Rīga, LV-1001, tālr.: +371 67295272, fakss: 67295294
 e-mail: unihaus@inbox.lv



”Mēs esam līdzās” approbation review



REHABILITĀCIJAS CENTRS
MĒS ESAM LĪDZĀS
REHABILITATION CENTER

Lēdmanes iela 2, Rīga, LV-1039, Latvija, telefons/fakss (+ 371) 675 53196, e-pasts: brc@eriga.lv, www.edi.lv/mel

Rīgā, 2013. gada 16. oktobrī

Atsauksme par stājas monitoringa iekārtas „SmartWear” prototipa aprobācijas rezultātiem Rehabilitācijas centrā „Mēs esam līdzās”

Pēc Elektronikas un datorzinātņu institūta lūguma Rehabilitācijas centrs „Mēs esam līdzās” veica Valsts pētījumu programmas projekta Nr. 2 „Inovatīvas signālapstrādes tehnoloģijas vielu un efektīvu elektronisko sistēmu radīšanai” pētījumu rezultātā izveidotās stājas monitoringa eksperimentālās sistēmas „SmartWear” aprobāciju.

Aprobāciju notika reālos apstākļos, strādājot ar Rehabilitācijas centra pacientiem. Iekārta ir universāla tāpēc, ka tā ir iestrādāta ortozē ar izmēra regulēšanas iespējām, kā arī tāpēc, ka var kalibrēt (saglabāt datus) iekārtu katram pacientam individuāli.

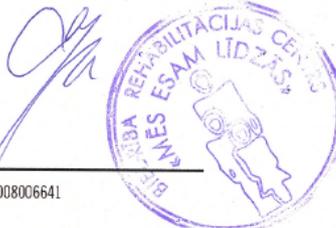
Aprobācijas laikā notika aktīva sadarbība ar Elektronikas un datorzinātņu institūta speciālistiem tehnoloģijas uzlabošanā. Balstoties uz mūsu komentāriem, sensori tika pāršūti no testējamā krekla, medicīniskā ortozē. Tika veiktas korekcijas mobilās iekārtas lietojumprogrammā, paplašinot sistēmas iespējas, piemēram, papildus iespējai mobilā telefonā saglabāt informāciju par to, cik ilgi pacients izmanto iekārtu un cik procentus no šī laika perioda viņš atrodas definētajā stājas pozīcijā. Aprobācijas periodā tika izveidots dalījums mazākos diapazonos, veicot procentu aprēķinu katrā laika diapazonā, ar mērķi novērot pacienta progresu.

Tā kā Rehabilitācijas centrā ārstējas pacienti ar Cerebrālo Trieku, kuriem ir nepieciešams attīstīt rumpja un augšējo ekstremitāšu stabilitāti un mobilitāti, viņiem tiek piemēroti vingrojumi, kuros ir iesaistītas roku kustības. Sākotnējā variantā sistēma bija pārāk jutīga un sūtīja atgriezenisko saiti, kad pacients kustināja rokas, tāpēc tika veiktas izmaiņas kļūdas aprēķināšanas algoritmā, kā rezultātā kļūdas aprēķinam netiek ņemti dati no plecu daļas sensoriem (pacients var brīvāk kustināt rokas), taču sensoru datu vizualizēšana notiek pilnā mērā. Tika veiktas uzlabojumi arī stājas 3D modeļa attēlošanai. 3D modeļa krāsu diapazons tika pielāgots stājas deformācijām – pacients redz, kura muguras daļa būtu jāiztaisno, jo tā attēlā izmaina krāsu uz sarkanu. Jaunajā lietojumprogrammā ir ērtāk rotēt un pārskatīt 3D modeli.

Aprobācijas laikā sistēma „SmartWear” ir parādījusi, ka tā ir noderīga darbam ar bērniem un jauniešiem, kuriem ir rumpja muskulatūras distonija vai ir novērojami stājas traucējumi. Šāda veida iekārtas potenciāli var tikt izmantotas rehabilitācijas medicīnā neiroloģisku un ortopēdisku saslimšanu gadījumos. Rehabilitācijas centra kolektīvs uzskata, ka ir mērķtiecīgi un lietderīgi turpināt šīs tehnoloģijas attīstību konkrētu produktu prototipu izveidē ar tālāku mērķi komerciāliem pielietojumiem, jo šobrīd medicīnas iekārtu tirgū iekārtai atbilstošā cenu kategorijā analogus nesaskatām.

Ņemot vērā aprobācijas rezultātus, uzskatām, ka stājas monitoringa sistēmai „SmartWear” ir potenciāls pielietojums rehabilitācijas medicīnā, un tās komercializācija dos ieguldījumu tautsaimniecības attīstībā.

Andra Greitāne
Rehabilitācijas centra „Mēs esam līdzās” valdes priekšsēdētāja
Sertificēts ārsts rehabilitologs



Biedrība "Rehabilitācijas centrs *Mēs esam līdzās*" Riekstu iela 12- 57, LV-1055, Reg. Nr. 40008006641

BIBLIOGRAPHY

- [1] R. A. Russell and S. Parkinson. “Sensing surface shape by touch”. In: *IEEE International Conference on Robotics and Automation*. Vol. 1. 1993, pp. 423–428.
- [2] D. Trivedi and C. Rahn. “Model-based shape estimation for soft robotic manipulators: The planar case”. In: *Journal of Mechanisms and Robotics*. Vol. 6. 2. 2014.
- [3] A. Nathan et al. “Flexible Electronics: The Next Ubiquitous Platform”. In: *Proceedings of the IEEE Special Centennial Issue*. Vol. 100. 2012, pp. 1486–1517.
- [4] J. Cheng et al. “Smart Textiles: From Niche to Mainstream”. In: *IEEE Pervasive Computing*. Vol. 12. 3. 2013, pp. 81–84.
- [5] F. Chollet and H. Liu. *A (not so) short introduction to Micro Electro Mechanical Systems*. This is an electronic document published under Creative Commons Attribution-NonCommercial 3.0 License. 2015. URL: <http://memscyclopedia.org/introMEMS.html>.
- [6] E. A. Lee and S. A. Seshia. *Introduction to Embedded Systems – A Cyber-Physical Systems Approach*. Second Edition, LeeSeshia.org, 2015.
- [7] P. Derler, E. A. Lee, and A. S. Vincentelli. “Modeling Cyber-Physical Systems”. In: *Proceedings of the IEEE*. Vol. 100. 1. 2012, pp. 13–28.
- [8] M. J. Holroyd. *Methods for the Synchronous Acquisition of 3D Shape and Material Appearance*. PhD Thesis, University of Virginia, 2011.
- [9] G. Sansoni, M. Trebeschi, and F. Docchio. “State-of-the-art and applications of 3D imaging sensors in industry, cultural heritage, medicine, and criminal investigation”. In: *Sensors*. Vol. 9. 1. 2009, pp. 568–601.
- [10] A. Hermanis, R. Cacurs, and M. Greitans. “Acceleration and Magnetic Sensor Network for Shape Sensing”. In: *IEEE Sensors Journal*. Vol. 16. 5. 2016, pp. 1271–1280.
- [11] A. Hermanis, R. Cacurs, and M. Greitans. “Shape sensing based on acceleration and magnetic sensor system”. In: *2nd IEEE International Symposium on Inertial Sensors and Systems (IEEE ISISS 2015)*. 2015.
- [12] A. Hermanis et al. “Efficient real-time data acquisition of wired sensor network with line topology”. In: *2013 IEEE Conference on Open Systems (ICOS 2013)*. 2013, pp. 133–138.
- [13] A. Hermanis et al. “Wearable Posture Monitoring System with Biofeedback via Smartphone”. In: *Journal of Medical and Bioengineering*. Vol. 2. 1. 2013, pp. 40–44.
- [14] K. Nesnebergs, A. Hermanis, and M. Greitans. “A Method for segment based surface reconstruction from discrete inclination values”. In: *Elektronika ir Elektrotechnika*. Vol. 20. 2. 2014, pp. 32–35.
- [15] A. Hermanis and K. Nesnebergs. “Grid shaped accelerometer network for surface shape recognition”. In: *Proceedings of the Biennial Baltic Electronics Conference, BEC*. 2012, pp. 203–206.
- [16] A. Hermanis. *Inerciālo sensoru tīkls virsmas formas noteikšanai*. Master’s thesis, RTU, 2012.

- [17] A. Hermanis et al. “Demo: Wearable Sensor System for Human Biomechanics Monitoring”. In: *Proceedings of the 2016 International Conference on Embedded Wireless Systems and Networks*. Graz, Austria, 2016, pp. 247–248.
- [18] A. Hermanis et al. “Wearable Sensor Grid Architecture for Body Posture and Surface Detection and Rehabilitation”. In: *Proceedings of the 14th International Conference on Information Processing in Sensor Networks*. Seattle, Washington, 2015, pp. 414–415.
- [19] A. Hermanis et al. “Wearable Head And Back Posture Feedback System For Children With Cerebral Palsy”. In: *Journal of Rehabilitation Medicine (ISSN 1650–1977)*. Vol. 47. 8. 2015, p. 777.
- [20] K. Nesenbergs et al. “Virtual Reality Rehabilitation System for Children with Cerebral Palsy”. In: *25th European Academy of Childhood Disability*. Newcastle, England, 10-12 oct, 2013.
- [21] A. Hermanis and K. Nesenbergs. “Accelerometer network for human posture monitoring”. In: *Riga Technical University 53rd International Scientific Conference*. Riga, Latvia, 2012.
- [22] E. Szelitzky et al. “Low Cost Angular Displacement Sensors for Biomechanical Applications – A Review”. In: *Journal of Biomedical Engineering and Technology*. Vol. 2. 2. 2014, pp. 21–28.
- [23] D. Preethichandra and K. Kaneto. “SAW sensor network fabricated on a polyvinylidene difluoride (PVDF) substrate for dynamic surface profile sensing”. In: *IEEE Sensors Journal*. Vol. 7. 5. 2007, pp. 646–649.
- [24] R. Balakrishnan et al. “Exploring Interactive Curve and Surface Manipulation Using a Bend and Twist Sensitive Input Strip”. In: *Proceedings of the 1999 Symposium on Interactive 3D Graphics*. I3D '99. Atlanta, Georgia, USA, 1999, pp. 111–118.
- [25] L. Danisch, K. Englehart, and A. Trivett. “Spatially continuous six degree of freedom position and orientation sensor”. In: *Sensor Review*. Vol. 19. 2. 1999, pp. 106–112.
- [26] T. Kato, A. Yamamoto, and T. Higuchi. “Shape recognition using piezoelectric thin films”. In: *Proceedings of the IEEE International Conference on Industrial Technology*. Vol. 1. 2003, pp. 112–116.
- [27] C.-H. Chuang, W.-B. Dong, and W.-B. Lo. “Flexible piezoelectric tactile sensor with structural electrodes array for shape recognition system”. In: *Proceedings of the 3rd International Conference on Sensing Technology, ICST 2008*. 2008, pp. 504–507.
- [28] C. Rendl et al. “FlexSense: A transparent self-sensing deformable surface”. In: *Proceedings of the 27th Annual ACM Symposium on User Interface Software and Technology*. 2014, pp. 129–138.
- [29] M. Zirkl et al. “An all-printed ferroelectric active matrix sensor network based on only five functional materials forming a touchless control interface”. In: *Advanced Materials*. Vol. 23. 18. 2011, pp. 2069–2074.
- [30] C. Rendl et al. “PyzoFlex: Printed piezoelectric pressure sensing foil”. In: *UIST'12 - Proceedings of the 25th Annual ACM Symposium on User Interface Software and Technology*. 2012, pp. 509–518.

- [31] F. Lorussi et al. “Wearable, redundant fabric-based sensor arrays for reconstruction of body segment posture”. In: *IEEE Sensors Journal*. Vol. 4. 6. 2004, pp. 807–818.
- [32] G. Hansson et al. “Validity and reliability of triaxial accelerometers for inclinometry in posture analysis”. In: *Medical and Biological Engineering and Computing*. Vol. 39. 4. 2001, pp. 405–413.
- [33] H. J. Luinge and P. H. Veltink. “Inclination measurement of human movement using a 3-D accelerometer with autocalibration”. In: *IEEE Transactions on Neural Systems and Rehabilitation Engineering*. Vol. 12. 1. 2004, pp. 112–121.
- [34] R. J. Nevins, N. G. Durdle, and V. J. Raso. “A posture monitoring system using accelerometers”. In: *Electrical and Computer Engineering, 2002. IEEE CCECE 2002. Canadian Conference on*. Vol. 2. 2002, 1087–1092 vol.2.
- [35] W. Wong and M. Wong. “Detecting spinal posture change in sitting positions with tri-axial accelerometers”. In: *Gait and Posture*. Vol. 27. 1. 2008, pp. 168–171.
- [36] “An inertial and magnetic sensor based technique for joint angle measurement”. In: *Journal of Biomechanics*. Vol. 40. 12. 2007, pp. 2604–2611.
- [37] R. E. Mayagoitia, A. V. Nene, and P. H. Veltink. “Accelerometer and rate gyroscope measurement of kinematics: an inexpensive alternative to optical motion analysis systems”. In: *Journal of Biomechanics*. Vol. 35. 4. 2002, pp. 537–542.
- [38] R. Zhu and Z. Zhou. “A real-time articulated human motion tracking using tri-axis inertial/magnetic sensors package”. In: *IEEE Transactions on Neural Systems and Rehabilitation Engineering*. Vol. 12. 2. 2004, pp. 295–302.
- [39] D. Roetenberg, H. Luinge, and P. Slycke. “Xsens MVN: Full 6DOF Human Motion Tracking Using Miniature Inertial Sensors”. In: *Xsens Technologies, White Paper*. 2013, pp. 1–9.
- [40] H. Luinge. *Inertial Sensing of Human Movement*. PhD Thesis, University of Twente, 2002.
- [41] D. Roetenberg. *Inertial and Magnetic Sensing of Human Motion*. PhD Thesis, University of Twente, 2006.
- [42] T. Hoshi, S. Ozaki, and H. Shinoda. “Three-dimensional shape capture sheet using distributed triaxial accelerometers”. In: *4th International Conference on Networked Sensing Systems*. 2007, pp. 207–212.
- [43] T. Hoshi and H. Shinoda. “Free-Form Tactile Sensor Using 3-Dimensional Shape Capture Sheet”. In: *Second Joint EuroHaptics Conference and Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems (WHC’07)*. 2007, pp. 403–408.
- [44] T. Hoshi and H. Shinoda. “Gravity-based 3D shape measuring sheet”. In: *SICE, 2007 Annual Conference*. 2007, pp. 2126–2131.
- [45] T. Hoshi and H. Shinoda. “Three-dimensional shape capture sheet using distributed six-axis sensors”. In: *5th International Conference on Networked Sensing Systems (INSS 2008)*. 2008, pp. 156–161.
- [46] T. Hoshi and H. Shinoda. “3D shape measuring sheet utilizing gravitational and geomagnetic fields”. In: *SICE Annual Conference, 2008*. 2008, pp. 915–920.

- [47] P. Mittendorfer and G. Cheng. “Uniform Cellular Design of Artificial Robotic Skin”. In: *Proceedings of ROBOTIK 2012: 7th German Conference on Robotics*. 2012, pp. 1–5.
- [48] P. Mittendorfer and G. Cheng. “3D surface reconstruction for robotic body parts with artificial skins”. In: *IEEE International Conference on Intelligent Robots and Systems*. 2012, pp. 4505–4510.
- [49] N. Sprynski et al. “Curve reconstruction via a ribbon of sensors”. In: *Proceedings of the IEEE International Conference on Electronics, Circuits, and Systems*. 2007, pp. 407–410.
- [50] D. David and N. Sprynski. “Method and device for acquisition of a geometric shape”. In: *US Patent 9,188,422*. 2005.
- [51] M. Huard et al. “Reconstruction of quasi developable surfaces from ribbon curves”. In: *Numerical Algorithms*. Vol. 63. 3. 2013, pp. 483–506.
- [52] N. Sprynski, B. Lacolle, and L. Biard. “Motion capture of an animated surface via sensors’ ribbons – Surface reconstruction via tangential measurements”. In: *1st International Conference on Pervasive and Embedded Computing and Communication Systems*. 2011, pp. 421–426.
- [53] N. Saguin-Sprynski et al. “Surfaces reconstruction via inertial sensors for monitoring”. In: *7th European Workshop on Structural Health Monitoring (EWSHM 2014) – 2nd European Conference of the Prognostics and Health Management (PHM) Society*. 2014, pp. 702–709.
- [54] J. Leavitt, A. Sideris, and J. E. Bobrow. “High bandwidth tilt measurement using low-cost sensors”. In: *IEEE/ASME Transactions on Mechatronics*. Vol. 11. 3. 2006, pp. 320–327.
- [55] X.-S. Ji and S.-R. Wang. “Research on the MEMS gyroscope random drift error”. In: *Yuhang Xuebao/Journal of Astronautics*. Vol. 27. 4. 2006, pp. 640–642.
- [56] D. Choukroun, I. Bar-Itzhack, and Y. Oshman. “Novel quaternion Kalman filter”. In: *IEEE Transactions on Aerospace and Electronic Systems*. Vol. 42. 1. 2006, pp. 174–190.
- [57] R. G. Valenti, I. Dryanovski, and J. Xiao. “A Linear Kalman Filter for MARG Orientation Estimation Using the Algebraic Quaternion Algorithm”. In: *IEEE Transactions on Instrumentation and Measurement*. Vol. 65. 2. 2016, pp. 467–481.
- [58] Z. Q. Zhang, X. L. Meng, and J. K. Wu. “Quaternion-Based Kalman Filter With Vector Selection for Accurate Orientation Tracking”. In: *IEEE Transactions on Instrumentation and Measurement*. Vol. 61. 10. 2012, pp. 2817–2824.
- [59] A. M. Sabatini. “Quaternion-based extended Kalman filter for determining orientation by inertial and magnetic sensing”. In: *IEEE Transactions on Biomedical Engineering*. Vol. 53. 7. 2006, pp. 1346–1356.
- [60] S. Madgwick, A. Harrison, and R. Vaidyanathan. “Estimation of IMU and MARG orientation using a gradient descent algorithm”. In: *IEEE International Conference on Rehabilitation Robotics*. 2011.
- [61] H. Fourati et al. “Complementary Observer for Body Segments Motion Capturing by Inertial and Magnetic Sensors”. In: *IEEE/ASME Transactions on Mechatronics*. Vol. 19. 1. 2014, pp. 149–157.

- [62] T. Brunner et al. “Evaluation of attitude estimation algorithms using absolute magnetic reference data: Methodology and results”. In: *2014 IEEE/ION Position, Location and Navigation Symposium - PLANS 2014*. 2014, pp. 212–218.
- [63] STMicroelectronics. *iNEMO inertial module: 3D accelerometer, 3D gyroscope, 3D magnetometer*. 2013, pp. 1–74.
- [64] A. Hanson. *Visualizing Quaternions*. The Morgan Kaufmann Series in Interactive 3D Technology, 2005.
- [65] F. Markley. “Attitude determination using vector observations and the singular value decomposition”. In: *Journal of the Astronautical Sciences*. Vol. 36. 3. 1988, pp. 245–258.
- [66] C. C. Liebe. “Star trackers for attitude determination”. In: *IEEE Aerospace and Electronic Systems Magazine*. Vol. 10. 6. 1995, pp. 10–16.
- [67] I. Bar-Itzhack and Y. Oshman. “Attitude Determination from Vector Observations: Quaternion Estimation”. In: *IEEE Transactions on Aerospace and Electronic Systems*. Vol. AES-21. 1. 1985, pp. 128–136.
- [68] F. Markley. “Attitude determination using vector observations: A fast optimal matrix algorithm”. In: *Journal of the Astronautical Sciences*. Vol. 41. 2. 1993, pp. 261–280.
- [69] G. Wahba. “A Least Squares Estimate of Satellite Attitude”. In: *SIAM Review*. Vol. 7. 3. 1965, pp. 409–409.
- [70] M. Shuster and S. Oh. “Three-axis attitude determination from vector observations”. In: *J Guid Control*. Vol. 4. 1. 1981, pp. 70–77.
- [71] X. Yun, E. R. Bachmann, and R. B. McGhee. “A Simplified Quaternion-Based Algorithm for Orientation Estimation From Earth Gravity and Magnetic Field Measurements”. In: *IEEE Transactions on Instrumentation and Measurement*. Vol. 57. 3. 2008, pp. 638–650.
- [72] M. Huard et al. “C2 interpolation of spatial data subject to arc-length constraints using Pythagorean-hodograph quintic splines”. In: *Graphical Models*. Vol. 76. 1. 2014, pp. 30–42.
- [73] N. Sprynski et al. “Surface reconstruction via geodesic interpolation”. In: *CAD Computer Aided Design*. Vol. 40. 4. 2008, pp. 480–492.
- [74] H. Cao et al. “Enabling technologies for wireless body area networks: A survey and outlook”. In: *IEEE Communications Magazine*. Vol. 47. 12. 2009, pp. 84–93.
- [75] S. Movassaghi et al. “Wireless Body Area Networks: A Survey”. In: *IEEE Communications Surveys Tutorials*. Vol. 16. 3. 2014, pp. 1658–1686.
- [76] P. Celka et al. “Wearable biosensing: signal processing and communication architectures issues”. In: *Journal of Telecommunications and Information Technology*. Vol. nr 4. 2005, pp. 90–104.
- [77] A. Osborne. “An Introduction to Microcomputers Volume 1: Basic Concepts”. In: *Osborne-McGraw Hill Berkeley California USA*. 1980, pp. 116–126.
- [78] S. B. J. M. Irazabal. “AN10216-01 I2C Manual”. In: *Application note, Philips Semiconductors, March 4*. 2003.
- [79] S. Corrigan. “Introduction to the Controller Area Network (CAN)”. In: *Texas Instruments, Application Report, July*. 2008.

- [80] H. Estl. “SPI interface and use in a daisy-chain bus configuration”. In: *Infineon Technologies AG, Application Note, Feb. 2002*.
- [81] H. Lee et al. “Wearable personal network based on fabric serial bus using electrically conductive yarn”. In: *ETRI Journal*. Vol. 32. 5. 2010, pp. 713–721.
- [82] STMicroelectronics. “LIS331DLH, MEMS digital output motion sensor ultra low-power high performance 3-axes ”nano” accelerometer, Data Sheet”. In: July. 2009, pp. 1–38.
- [83] S. Chen et al. “Enabling longitudinal assessment of ankle-foot orthosis efficacy for children with cerebral palsy”. In: *Proceedings - Wireless Health 2011, WH’11*. 2011.
- [84] STMicroelectronics. “LSM303DLHC, Ultra-compact high-performance eCompass module: 3D accelerometer and 3D magnetometer, Data Sheet”. In: November. 2013, pp. 1–42.
- [85] *Xsens MTi-G overview*. 25.05.2016. URL: <https://www.xsens.com/products/mti-g/>.
- [86] J. Vasconcelos et al. “Geometric approach to strapdown magnetometer calibration in sensor frame”. In: *IEEE Transactions on Aerospace and Electronic Systems*. Vol. 47. 2. 2011, pp. 1293–1306.
- [87] G. S. Faber et al. “A novel method for assessing the 3-D orientation accuracy of inertial/magnetic sensors”. In: *Journal of Biomechanics*. Vol. 46. 15. 2013, pp. 2745–2751.
- [88] *Texas Instruments, MSP-EXP430FR5739 Experimenter Board*. 25.05.2016. URL: <http://www.ti.com/tool/msp-exp430fr5739>.
- [89] TexasInstruments. “MSP430G2553, MIXED SIGNAL MICROCONTROLLER, Data Sheet”. In: April. 2013, pp. 1–72.
- [90] *Rayson BTM-222, Bluetooth module*. 25.05.2016. URL: http://www.tme.eu/lv/Document/b90358c53cd65c9b10d2914f55812666/btm222_datasheet.pdf.
- [91] A. Hermanis and R. Cacurs. *YouTube, EDI shape sensing fabric video demonstration*. 25.05.2016. URL: https://www.youtube.com/watch?v=YDG0ERF2_d8.
- [92] L. Yang et al. “Evaluating and improving the depth accuracy of Kinect for Windows v2”. In: *IEEE Sensors Journal*. Vol. 15. 8. 2015, pp. 4275–4285.
- [93] P. Besl and N. McKay. “A Method for Registration of 3-D Shapes”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence*. Vol. 14. 2. 1992, pp. 239–256.
- [94] G. Claus. “On Averaging Rotations”. In: *International Journal of Computer Vision*. Vol. 42. 1-2. 2001, pp. 7–16.
- [95] *Rehabilitācijas centrs ”Mēs Esam Līdzās”*. accessed 20.05.2016. URL: <http://www.mel.lv>.
- [96] *SIA „UniHaus”*. accessed 20.05.2016. URL: <http://www.unihaus.lv/lv/unihaus/paramums>.