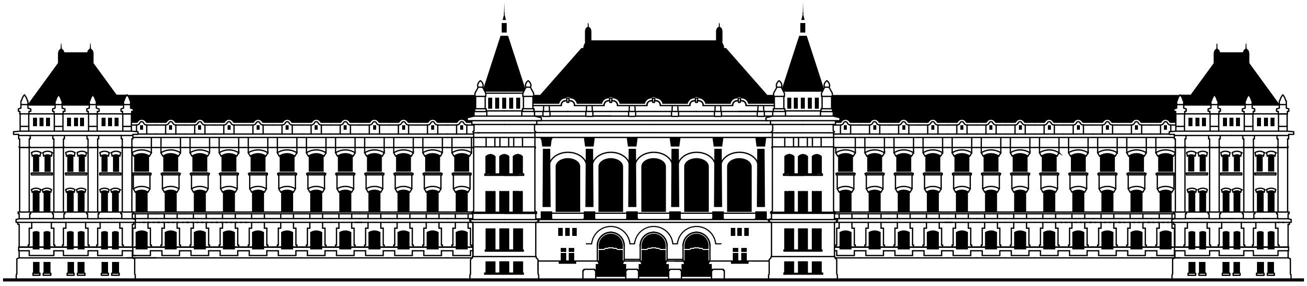
Instrumentation and Measurement & Engineering
in Medicine and Biology Joint Chapter

Proceedings
# of the
21$^{\text{st}}$ PhD Mini-Symposium

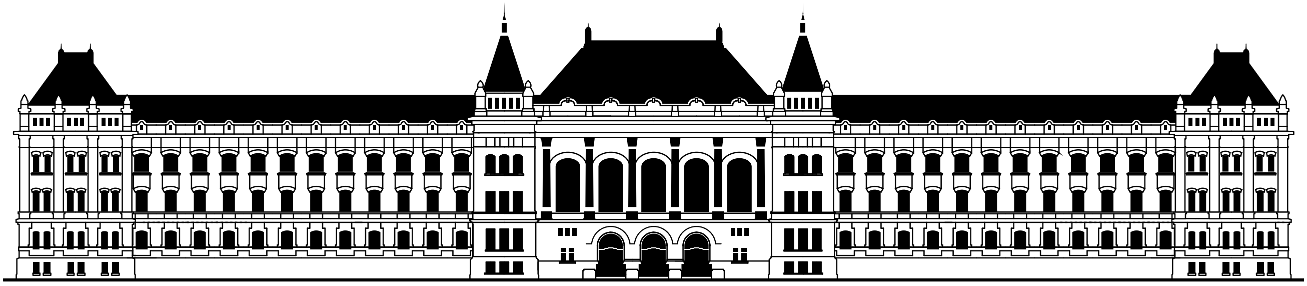
February 3, 2014



Budapest University of Technology and Economics
Department of Measurement and Information Systems

Proceedings of the 21st PhD Mini-Symposium

February 3, 2014
Budapest University of Technology and Economics
Building I



Budapest University of Technology and Economics
Department of Measurement and Information Systems

Head of the Department: Prof. Dr. Ákos JOBBÁGY

Conference chairman:
Béla PATAKI

Organizers:
Benedek IZSÓ
Ivor DÜLK
Bence BOLGÁR

Homepage of the Conference:
http://minisy.mit.bme.hu/

Sponsored by:

IEEE Instrumentation and Measurement & Engineering
in Medicine and Biology Joint Chapter

Schnell László Foundation

ISBN 978-963-313-104-6

FOREWORD

This proceedings is a collection of the lectures of the 21st PhD Mini-Symposium held at the Department of Measurement and Information Systems of the Budapest University of Technology and Economics. The main purpose of these symposiums is to give an opportunity to the PhD students of our department to present a summary of their work done in the preceding year. It is an interesting additional benefit, that the students get some experience: how to organize such events. Beyond this actual goal, it turned out that the proceedings of our symposiums give an interesting overview of the research and PhD education carried out in our department. The lectures reflect partly the scientific fields and work of the students, but we think that an insight into the research and development activity of the department is also given by these contributions. Traditionally our activity was focused on measurement and instrumentation. The area has slowly changed during the last few years. New areas mainly connected to embedded information systems, new aspects e.g. dependability and security are now in our scope of interest as well. Both theoretical and practical aspects are dealt with.

The proceedings will not be published in printed form, it has turned out that nowadays the web publication of symposium lectures is enough. This new form has some advantages, but it has some challenges as well. We hope that the advantages will dominate.

The papers of this proceedings could be sorted into two main groups: Model-based and Intelligent Systems; and Measurement, Signal Processing and Embedded Systems. The lectures are at different levels: some of them present the very first results of a research, others contain more new results. Some of the first year PhD students have been working on their fields only for half a year.

There are two types of papers in the proceedings. One is a short independent publication; the other is simply a summary of work. This second one is intended to give an overview of the work done during the last year; therefore it could contain shorter or longer parts of the PhD student's other publications; and it does not necessarily contain new results, which have not been published earlier. It is clearly indicated in each paper that which category it belongs to.

During this twenty-one-year period there have been shorter or longer cooperations between our department and some universities, research institutes, organizations and firms. Some PhD research works gained a lot from these connections. In the last year the cooperation was especially fruitful with the Vrije Universiteit Brussel Dienst ELEC, Brussels, Belgium; Innomed Medical Zrt., Budapest; ERICSSON Magyarország Kft., Budapest; Complex Systems and Computational Neuroscience Group at Wigner Research Center, Budapest; IEEE Instrumentation and Measurement Society & Engineering in Medicine and Biology Society Joint Chapter, IEEE Hungary Section.

We hope that similarly to the previous years, also this PhD Mini-Symposium will be useful for the lecturers, for the audience and for all who read the proceedings.

Budapest, January, 2014.

Béla Pataki
Chairman of the PhD
Mini-Symposium

List of Participants

Participant	Advisor	Starting Year of PhD Course
BOLGÁR, Bence	ANTAL, Péter	2012
CSERPÁN, Dorottya	HORVÁTH, Gábor	2011
DÜLK, Ivor	KOVÁCSHÁZY, Tamás	2012
FERENCZ, Bálint	KOVÁCSHÁZY, Tamás	2013
GYÖRKE, Péter	PATAKI, Béla	2011
HORVÁTH, Áron	HORVÁTH, Gábor	2011
IZSÓ, Benedek	RÁTH, István	2012
LÓSKA, Ádám	PATAKI, Béla	2013
MARX, Péter	ANTAL, Péter	2010
RENCZES, Balázs	KOLLÁR, István	2013
WACHA, Gábor	FEHÉR, Béla	2013

PROGRAM OF THE MINI-SYMPOSIUM

Model-Based and Intelligent Systems

Chair: JOBBÁGY, Ákos

Measurement, Signal Processing and Embedded Systems

Chair: KOLLÁR, István

Conference Schedule

Time	**February 3, 2014**
8:30	Opening speech
8:35	Model-Based and Intelligent Systems
10:20	Coffee break
10:40	Measurement, Signal Processing and Embedded Systems

ESTIMATION OF INPUT SIGNALS BASED ON MULTIELECTRODE ARRAY MEASUREMENTS

ANNUAL RESEARCH REPORT*

Dorottya CSERPÁN
Advisor: Gábor HORVÁTH

I. Introduction

The human brain contains 10^{11} neurons, each of them on average connecting to 10^4 others resulting a highly complex system, what scientists strive to understand. As the centuries are passing by, the equipment used in this quest are getting more sensitive and specific. Even though it's possible to reach sub-millisecond temporal and sub-micrometer spatial resolution or to see the whole brain's activation, our understanding is highly limited. Some of the most important and crucial questions are what the neural code in the brain is and how computation is done. One specific property of the neurons is, that they are not just able to change their membrane potential significantly within a few milliseconds, which mechanism is called action potential, but influence other neuron's membrane potential typically through the connections called synapses. There is a huge variability in the morphology of neurons in the different brain regions usually connected to their roles. As typically it is the dendrite-tree, where the inputs of other cells arrive modifying the membrane potential, if there were a tool to detect the caused change in the membrane potential and the location, it would lead to the possibility of the deeper understanding of the computation done by neurons. There are few methods aiming to estimate the current source density, which is the sum of currents entering or leaving a specific volume of a population [1] [2] [3] or a single neuron [4] based on the recordings of 1 or 2 dimensional multielectrode-arrays (MEA). These methods are using some special geometrical assumptions, and do not provide any information about the inputs on a detailed dendrite level. As a result of the rapid improvement in the MEA technology it is possible to make MEAs with 20 um spatial distribution and 30 kHz sampling frequency. The motivation of our my work was to develop a method for estimating the membrane currents of a neuron with respect to its morphology. Hereby I will present the theoretical background of this novel and its validation on simulated data.

II. The kernel Current Source Density method for single neurons (ksCSD)

This method builds on the results of two other methods to calculate the current source density distribution of single neurons [4] by using kernel methods[2]. The main advantages of this method is the more accurate spatial resolution, but for this the morphology of the cell is needed. Fluorescent dyes enable such reconstruction after the recording was done.

A. Theory

The CSD distribution at point **x** can be expressed as the sum of the M current sources:

$$C(\mathbf{x}) = \sum_{j=i}^{M} a_j \tilde{b}_j(\mathbf{x}) \tag{1}$$

[*] This paper is not an independent one, it is the summary of the PhD work of the last year presented in the minisymposium. It may contain parts of previously published documents or parts to be published later.

$\tilde{b}$ are the basis functions overlapping each other and are distributed evenly along the morphology of the cell,a_j is a multiplication constant.

The generated potential by the $\tilde{b_i}$ is denoted as b_i, and the connection is made by the A linear operator. While reconstructing the shape of the neuron, first we needed to identify the branching points and the branches. After that given based on the morphology information, each branch were estimated by a spline. The basis functions were distributed along these curves uniformly. The current source density distributions were estimated along these splines as well.

The curves in the 3D space can be parametrized with variable t.

$$\begin{aligned} x &= f_x(t) \\ y &= f_y(t) \\ z &= f_z(t) \end{aligned} \tag{2}$$

There is an infinite number of basis function we could use, we used Gaussian source function:

$$\tilde{b}_i(t') = e^{-\frac{(t'-t_i)^2}{R^2}} \tag{3}$$

where both $t, t_i \in [0, d]$ are parameters on the same branch which has a length of d. Here R is the double of the variance of the Gaussian function. The connection between the current source densities and potentials Φ is introduced by the A operator($A : \tilde{F} \rightarrow F$)

$$b_i(x, y, z) = A\tilde{b}_i(t') = \frac{1}{4\pi\sigma} \int \frac{\tilde{b}_i(t')}{\sqrt{(x - x'(t))^2 + (y - y'(t))^2 + (z - z'(t))^2}} dt' \tag{4}$$

$$\Phi(\mathbf{x}) = AC(\mathbf{x}) = \sum_i^M a_i b_i(\mathbf{x}) \tag{5}$$

where $b_i = A\tilde{(b)}_i$.

The detailed derivation of this method can be found in the reference [2], due to the length limitation it's not detailed here. To determine the CSD distribution in arbitrary positions(x), the following kernel functions were introduced:

$$K(\mathbf{x}_k, \mathbf{x}_l) = \sum_{i=1}^M b_i(\mathbf{x}_k) b_i(\mathbf{x}_l) \tag{6}$$

$$\tilde{K}(\mathbf{x}_k, \mathbf{y}_l) = \sum_{j=1}^M b_j(\mathbf{x}_k) \tilde{b}_j(\mathbf{y}_l) \tag{7}$$

Using the simulated or measured extracellular potentials (V) and assuming $\tilde{K}$ is invertible the solution for C is straightforward.

$$C(\mathbf{x}) = \tilde{\mathbf{K}}^T(\mathbf{x}) \tilde{\mathbf{K}}^{-1} \mathbf{V} \tag{8}$$

B. Testing the method on simulated data

As there are several parameters used in the simulations, like the width and number of used base functions, the number and arrangements of the electrodes, the cell-electrode distance and the morphology of the cell, it is impossible the study all cases. Therefore some special cases in terms of morphology was chosen for studying the behaviour of the method. As to remain closer to biological reality, I used the LFPy [5] simulational environment, which given a certain cell morphology and parameters describing the electrophysiological properties of the cell, simulates the membrane potential, membrane currents and extracellular potential both in space and time. On the simulated extracellular potential we

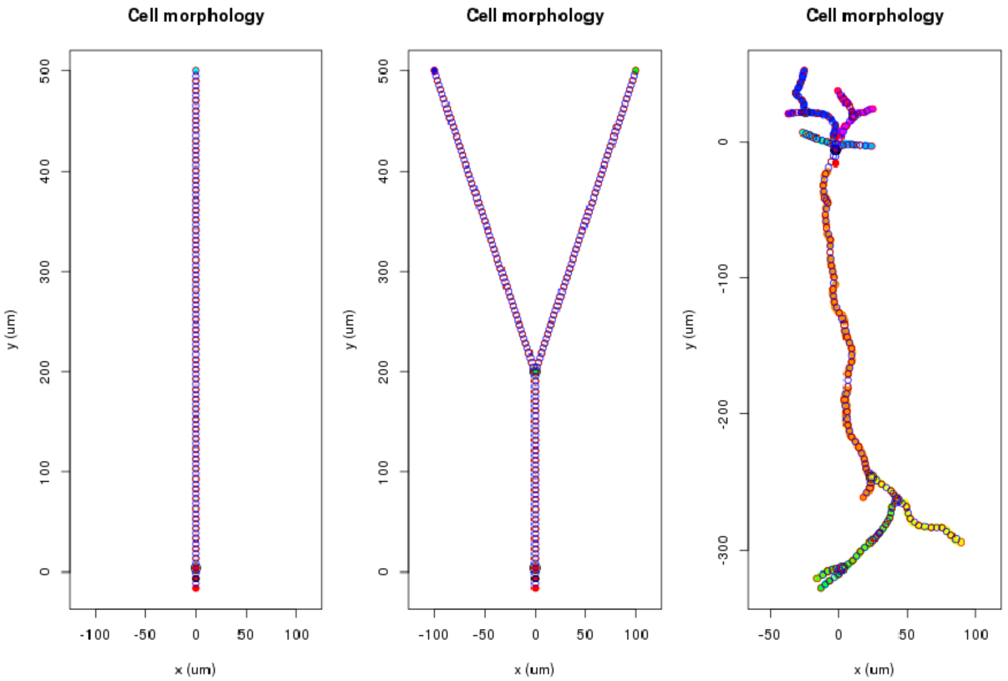


Figure 1: The ksCSD method was tested for 3 different morphologies: ballstick, Y-shaped and branching. The empty blue and red circles represent the beginning and the end of the segments used in the simulation of the extracellular potential.The filled circles with different colours belong to different branches identified based on the connection information provided in the morphology files. On the pictures the filled circles only occur at points, for which the coordinates were given.

use the ksCSD method to give an estimation for the membrane currents, as these are known from the simulation, it is possible to calculate the error of the method with the following error measure:

$$\epsilon = \frac{\sum_i \sum_t |C_{i,t} - C^0_{i,t}|}{\sum_i \sum_t |C^0_{i,t}|}, \tag{9}$$

where C^0 denotes the membrane currents calculated by LFPy, C stands for the ksCSD estimated current, the summation goes for runs all segments i and all time snaps t. One of the main results is the comparison of the estimated currents with the the original membrane currents by using the above mentioned measure, which is shown on Figure 2. As expected, applying the ksCSD method for cells with simpler morphologies and using more electrodes improves the performance. It seems, that there is an optimal width for the basis functions, but increasing the number of them does not decreases the error significantly in many cases.

III. Conclusion

The above presented method has the potential of providing information regarding to the inputs of the cell, which would deepen our knowledge in understanding the computational mechanisms of the neurons. I tested the ksCSD method for different simulational setup, in which cases the results were promising. Our future plans involve the the comparison of our method with the other existing ones, to extend the simulations to a biologically realistic tissue involving other cells. It is inevitable to try our method on experimental data, but currently there is none available.

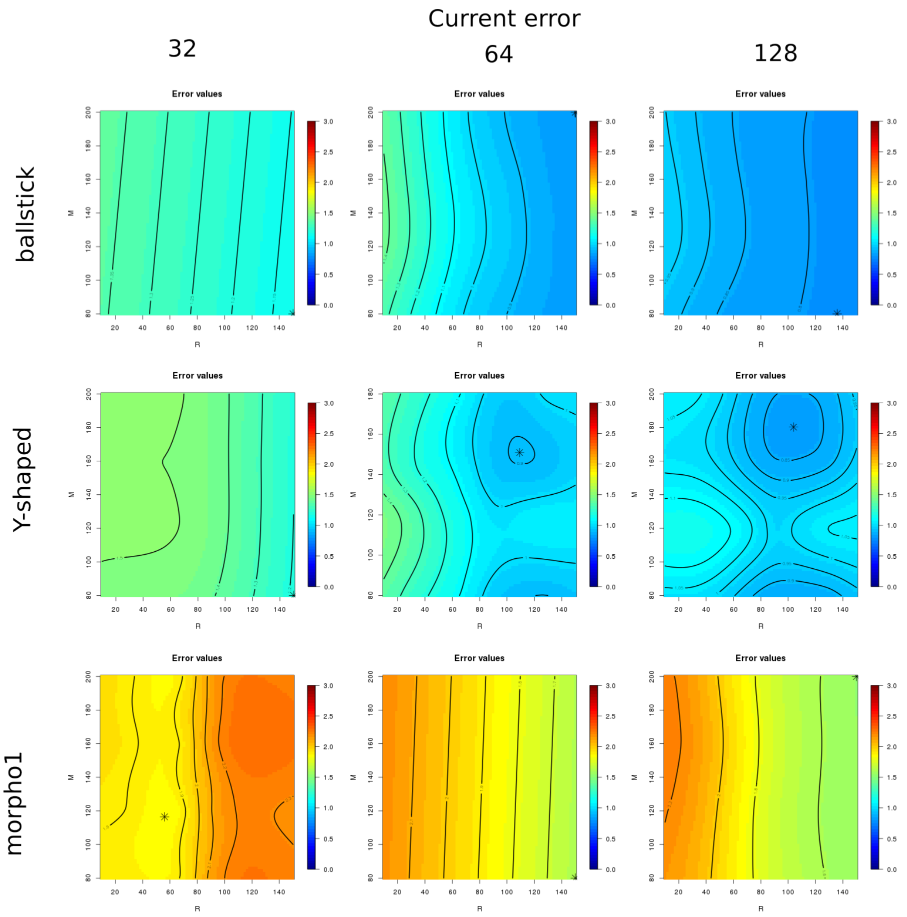


Figure 2: Colour map of the error of the ksCSD method. During the simulations, the above mentioned 3 morphologies and the following number of electrodes was used: 32, 64, 128, the number of the basis functions varied between 80 and 200, the R parameter took values between 10 and 150. Stars marked the smallest error value on the plot. The position of the cell compared to the electrode remained the same during the simulations, including the cell to electrode distance.

Acknowledgement

I am indebted first of all to my supervisor Zoltán Somogyvári and to my many colleagues from the Complex Systems and Computational Neuroscience Group at Wigner Research Center who supported me, furthermore to Daniel Wójcik who gave me the the opportunity to work in his Laboratory of Neuroinformatics at Nencki Institute.

References

[1] C. Nicholson and J. A. Freeman, “Theory of current source density analysis and determination of conductivity tensor for anuran crebellum,” *J. Neurophysiol.*, 38(2):356–368, 1975.

[2] J. Potworowski, W. Jakuczun, S. Leski, and D. Wójcik, “Kernel current source density method,” *Neural Comput.*, 24(2):541–575, Feb. 2012.

[3] S. Łeski, K. Pettersen, B. Tunstall, G. Einevoll, J. Gigg, and D. Wójcik, “Inverse current source density method in two dimensions: Inferring neural activation from multielectrode recordings,” *Neuroinformatics*, 9(4):401–425, 2011.

[4] Z. Somogyvári, D. Cserpán, I. Ulbert, and P. Erdi, “Localization of single-cell current sources based on extracellular potential patterns: the spike csd method.,” *Eur J Neurosci.*, 36(10):3299–3313, Nov. 2012.

[5] H. Linden, E. Hagen, S. Leski, E. Norheim, H. Pettersen, and G. Einevoll, “Lfpy: A tool for simulation of extracellular potentials.,” *Front Neuroinforn.*, (147), 2011.

MULTIPLE KERNEL LEARNING OF DISTANCE METRICS WITH PAIRWISE EQUIVALENCE AND INEQUIVALENCE CONSTRAINTS

Bence BOLGÁR
Advisor: Péter ANTAL

I. Introduction

Machine learning algorithms play an important role in intelligent data analysis, especially when dealing with high-dimensional data. In the typical setting, data samples are represented in a high-dimensional Euclidean space. However, very often, the underlying "reality" generating the data has a lower number of degrees of freedom, and therefore samples can be assumed to form a lower-dimensional manifold embedded into the input space. Determining and exploiting the structure of this manifold is the realm of manifold learning, which has drawn considerable interest in the past few years [1]. The first techniques which can be seen as members of this family date back to the early 1900s and have been developed to perform dimensionality reduction. Many of the dimensonality reduction methods approximate the manifold structure by searching for something "simple", in the spirit of Ockham's razor: *e.g.* linear subspaces, local structures, or simple features in a transformed (kernel) space. Manifold learning approaches also vary greatly in terms of input, *e.g.*:

- Equivalence classes ("inverse" clustering)
- Topology (Self-Organizing Maps)
- Prior constraints (Distance Metric Learning)

In this work, we focus on DML, which essentially concerns itself with learning an optimal transformation of the data in a given prediction task, also having deep connections to clustering, rank learning, recommendation systems and multitask learning. A large number of techniques have been developed, however, few of them addressed the problem of combining and extending multiple pre-specified metrics based on heterogeneous information sources. In this work, we propose a metric learning method which computes a combined metric, based on multiple prior metrics and equivalence classes (*i.e.* must-link and cannot-link constraints specified in the training set). This is achieved by learning a Mahalanobis distance metric subject to these constraints, and imposing a squared L_p-norm on the learned distance matrix weights (a similar approach was utilized by Wang *et al.* for image classification [2]; many DML algorithms are described in comprehensive reviews [3, 4]). We also introduce a tool called QDF^3 which implements the algorithm and apply it to multi-aspect drug repositioning [5]. This methodology extends an earlier version of our work, which utilized one-class multiple-kernel SVMs [6] to rank drugs [5, 7].

II. The multiple kernel metric learning framework

Due to their flexibility and good empirical performance, distance metric learning techniques using the Mahalanobis distance are probably the most popular approaches. The Mahalanobis distance is defined as $d_{\boldsymbol{M}}(\boldsymbol{x}, \boldsymbol{x'}) = \sqrt{(\boldsymbol{x} - \boldsymbol{x'})^T \boldsymbol{M} (\boldsymbol{x} - \boldsymbol{x'})}$ where $\boldsymbol{M}$ is a positive semidefinite matrix. This can be also seen as a standard Euclidean distance metric in a transformed space, since $\sqrt{(\boldsymbol{x} - \boldsymbol{x'})^T \boldsymbol{M} (\boldsymbol{x} - \boldsymbol{x'})} = \sqrt{(\boldsymbol{x} - \boldsymbol{x'})^T \boldsymbol{L}^T \boldsymbol{L} (\boldsymbol{x} - \boldsymbol{x'})} = \|\boldsymbol{L}\boldsymbol{x} - \boldsymbol{L}\boldsymbol{x'}\|_2$. In this work, we utilize multiple distance metrics parameterized by the Mahalanobis matrices $\boldsymbol{W}_k$, corresponding to k different data sources. Since we also want to utilize pairwise constraints, we write training instances as $\{(\boldsymbol{x}_{ik}, \boldsymbol{z}_{ik}, y_i)\}_{i=1}^{P}$ where $\boldsymbol{x}_{ik}, \boldsymbol{z}_{ik} \in \mathbb{R}^{D_k}$ are data samples in the kth data source and $y_i \in \{-1, +1\}$ denotes their relationship (-1 for cannot-link and $+1$ for must-link constraint).

The squared Mahalanobis distance between the entities of the ith training instance can be written as

$$d_i = \sum_k \operatorname{Tr}\left(\boldsymbol{W}_k \left(\boldsymbol{x}_{ik} - \boldsymbol{z}_{ik}\right) \left(\boldsymbol{x}_{ik} - \boldsymbol{z}_{ik}\right)^T\right),$$

where k indexes the data sources, $\boldsymbol{W}_k$ is the kth Mahalanobis matrix, and d_i denotes the squared distance between the two samples of the ith training instance. Note that the matrix weights m_k have been omitted at this stage as they can be incorporated in a more convenient way in the next step.

Adapting the work of Vishwanathan *et al.* [6], a Multiple Kernel Learning-like formulation of the metric learning problem is

$$\begin{aligned} \min_{\boldsymbol{W},\boldsymbol{m},\boldsymbol{\xi}} \quad & \frac{1}{2}\sum_k \frac{\left\|\boldsymbol{W}_k - \boldsymbol{W}_k^0\right\|_F^2}{m_k} + C\sum_i \xi_i + \frac{\lambda}{2}\left\|\boldsymbol{m}\right\|_p^2 \\ s.t. \quad & y_i(r - d_i) \geq \rho - \xi_i, \quad \xi_i \geq 0, \quad m_k > 0, \end{aligned}$$

where $\boldsymbol{W}_k^0$ is the kth input distance matrix, $\|\cdot\|_F$ is the Frobenius-norm, m_k is the weight of the kth data source, ξ_i's are the slack variables, C controls the model complexity, and the last term stands for the L_p-norm regularization of the weights. The first constraint can be decomposed as (omitting the slack variables for the moment)

$$\begin{aligned} (d_i, y_i = +1) \quad &\Rightarrow \quad d_i \leq r - \rho, \\ (d_i, y_i = -1) \quad &\Rightarrow \quad d_i \geq r + \rho, \end{aligned}$$

i.e. when the entities $\boldsymbol{x}_i$, $\boldsymbol{z}_i$ are of the same class (denoted by $y_i = 1$), the distance should be small, and for the opposite case ($y_i = -1$), the distance should be large. The constants r and ρ control the ratio of distance between "similar" and "dissimilar" entities – in fact, ρ acts something like a margin which separates the "class of distances between similar entities" and the "class of distances between dissimilar entities".

Following a similar argument to that of Vishwanathan *et al.* [6], the dual is derived to be

$$\begin{aligned} \max_{\boldsymbol{\alpha}} \quad & -\frac{1}{8\lambda}\left(\sum_k \left(\boldsymbol{\alpha}^T\boldsymbol{Q}_k\boldsymbol{\alpha}\right)^q\right)^{\frac{2}{q}} - \sum_i \alpha_i y_i \left(r - d_i^0\right) + \boldsymbol{\rho}^T\boldsymbol{\alpha} \\ s.t. \quad & 0 \leq \boldsymbol{\alpha} \leq C, \end{aligned}$$

where $\boldsymbol{Q}_{k_{ij}} = y_i y_j \operatorname{Tr}\left((\boldsymbol{x}_{ik} - \boldsymbol{z}_{ik})(\boldsymbol{x}_{ik} - \boldsymbol{z}_{ik})^T(\boldsymbol{x}_{jk} - \boldsymbol{z}_{jk})(\boldsymbol{x}_{jk} - \boldsymbol{z}_{jk})^T\right)$ and d_i^0 is the distance between $\boldsymbol{x}_i$ and $\boldsymbol{z}_i$ according to the prior metrics. Note that the dual is convex, differentiable and can be easily optimized using the projected gradient method. In particular, in view of earlier findings, we decided to focus on squared L^2-norm regularization. In this case, the gradient updates can be calculated analytically, *i.e.* we can avoid the computation of the Hessian. Moreover, it has been shown that L^2-norm regularization works rather well with biomedical data (*i.e.* a relatively small number of heterogeneous information sources, each of them containing relevant information about the entities). After the optimization is complete, the squared distance of any two samples $\boldsymbol{x}$ and $\boldsymbol{z}$ can be recovered from the optimal dual variables as

$$d(\boldsymbol{x}, \boldsymbol{z})^2 = \sum_k \left[d_{\boldsymbol{W}_0^k}(\boldsymbol{x}, \boldsymbol{z})^2 - m_k \sum_i \alpha_i y_i \operatorname{Tr}\left((\boldsymbol{x}_{ik} - \boldsymbol{z}_{ik})(\boldsymbol{x}_{ik} - \boldsymbol{z}_{ik})^T(\boldsymbol{x}_k - \boldsymbol{z}_k)(\boldsymbol{x}_k - \boldsymbol{z}_k)^T\right)\right].$$

It is a well-known fact that working in high-dimensional feature spaces can significantly improve accuracy. Here we utilize the KPCA trick [8] to learn a nonlinear distance metric, by learning a Mahalanobis distance metric in a feature space $\mathbb{F}$. Let $\phi : \mathbb{R}^d \rightarrow \mathbb{F}$ be the map from the input space to the feature space, and $k(\boldsymbol{x}_i, \boldsymbol{x}_j) = \phi(\boldsymbol{x}_i)^T\phi(\boldsymbol{x}_j)$. Recalling the definition of the Mahalanobis distance, we get $d_{\boldsymbol{M}}(\boldsymbol{x}, \boldsymbol{x'}) = \sqrt{(\phi(\boldsymbol{x}) - \phi(\boldsymbol{x'}))^T\boldsymbol{L}^T\boldsymbol{L}(\phi(\boldsymbol{x}) - \phi(\boldsymbol{x'}))}$, *i.e.* now we have the linear transformation in $\mathbb{F}$. Our goal is to express the distance solely in terms of inner products. To this end, we parameterize the linear transformation $\boldsymbol{L}$ to be $\boldsymbol{L} = \boldsymbol{F\Phi}$ where $\boldsymbol{\Phi}$ denotes the matrix $[\phi(\boldsymbol{x}_1)\ \phi(\boldsymbol{x}_2)\ \cdots\ \phi(\boldsymbol{x}_n)]^T$, and thus $d_{\boldsymbol{M}}(\boldsymbol{x}, \boldsymbol{x'}) = \sqrt{(\phi(\boldsymbol{x}) - \phi(\boldsymbol{x'}))^T\boldsymbol{\Phi}^T\boldsymbol{F}^T\boldsymbol{F\Phi}(\phi(\boldsymbol{x}) - \phi(\boldsymbol{x'}))}$. If we define $\tilde{\boldsymbol{x}}_i = \boldsymbol{\Phi}\phi(\boldsymbol{x}_i) = [k(\boldsymbol{x}_1, \boldsymbol{x}_i)\ \cdots\ k(\boldsymbol{x}_n, \boldsymbol{x}_i)]^T$, we can write the squared distance solely in terms of inner products:

$$d_i = (\tilde{\boldsymbol{x}}_i - \tilde{\boldsymbol{z}}_i)^T\boldsymbol{F}^T\boldsymbol{F}(\tilde{\boldsymbol{x}}_i - \tilde{\boldsymbol{z}}_i) = \operatorname{Tr}\Big(\boldsymbol{W}(\tilde{\boldsymbol{x}}_i - \tilde{\boldsymbol{z}}_i)(\tilde{\boldsymbol{x}}_i - \tilde{\boldsymbol{z}}_i)^T\Big)$$

where $\boldsymbol{W} := \boldsymbol{F}^T\boldsymbol{F}$, or similarly, in the multiple kernel setting,

$$d_i = \sum_k \operatorname{Tr}\Big(\boldsymbol{W_k}(\tilde{\boldsymbol{x}}_{ik} - \tilde{\boldsymbol{z}}_{ik})(\tilde{\boldsymbol{x}}_{ik} - \tilde{\boldsymbol{z}}_{ik})^T\Big)$$

using multiple kernel functions $k_k(\cdot, \cdot)$.

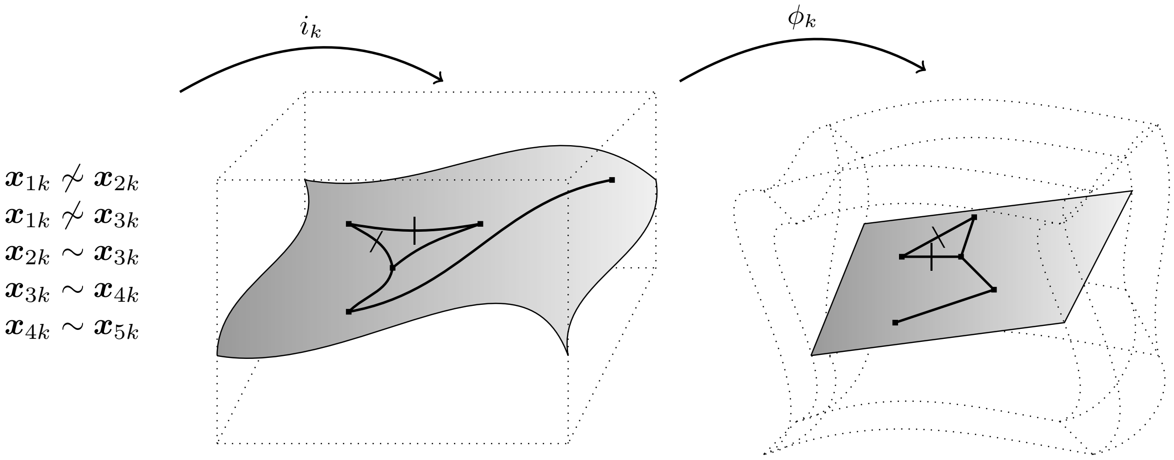

i_k
ϕ_k
$\boldsymbol{x}_{1k} \nsim \boldsymbol{x}_{2k}$
$\boldsymbol{x}_{1k} \nsim \boldsymbol{x}_{3k}$
$\boldsymbol{x}_{2k} \sim \boldsymbol{x}_{3k}$
$\boldsymbol{x}_{3k} \sim \boldsymbol{x}_{4k}$
$\boldsymbol{x}_{4k} \sim \boldsymbol{x}_{5k}$

Figure 1: Kernel space embeddings of the samples $\boldsymbol{x}_i$ according to the kth information source, its associated prior metric and kernel function. The algorithm learns an optimal metric which is as "close" as possible to the prior metrics, and satisfies the constraints specified by the user (in the form of equivalence classes). An optimal weighting of the information sources is also computed in the process.

III. Implementation (QDF^3)

We implemented the core algorithm in C++, using the Eigen library for fast expression template-based linear algebra. We also developed a QT-based user-friendly graphical interface, partly for educational purposes, which provides further functionalities such as

- On-the-fly kernel computation with many kernel functions.
- Various plotting options to visualize results and support interpretation.
- A graph viewer including several layout and clustering functions (OGDF).
- Statistical analyses, such as Enrichment Analysis using the SaddleSum algorithm.

It is also worth noting that both the optimization process and the computation of distances require a large number of matrix–vector multiplications, which can be greatly accelerated by using BLAS libraries or specialized hardware. An OpenCL-based implementation for GPUs is currently under development, from which we expect a speedup of at least two orders of magnitude.

IV. Results

Drug repositioning refers to the act of re-using an existing drug in a new indication, which has been a rapidly emerging paradigm in the past decade, and provides a relatively safe, cost-efficient and fast alternative to *de novo* drug discovery [9]. Here we present an evaluation of the method by applying the developed tool to drug repositioning. We utilized the following data sources:

- Chemical structural descriptors: MACCS, MolconnZ, pharmacophore-based 3D.
- Side-effect profiles obtained from drug package inserts by text mining and from the SIDER database [10].

We also employed the ATC classification, which is an international taxonomy of approved drugs. To maintain compatibility with our previous benchmarks [5], first we evaluated the method using 100-fold 70-30 cross-validation as follows:

1. For a given ATC class with at least 6 drugs, training and test sets were constructed by randomly selecting 70% and 30% of the drugs, respectively.
2. Metric learning was performed by connecting all possible pairs within the training set.
3. A ranking of drugs was computed by utilizing the approach of Willett *et al.* [11], where the query set consisted of drugs present in the training set.
4. The above steps were repeated 100 times and AUC values were computed. In ranking tasks, the AUC value can be interpreted as the probability of ranking a "positive"sample (*i.e.* an element of the test set) higher then a randomly selected "negative" one.

In accordance with our previous findings [5], we observed that the heterogeneity of the ATC class greatly affects the ranking. For example, the reported AUC value for the "umbrella" class V03AB (antidotes) was only 0.734294, while most reasonable classes scored around $0.9 - 0.95$. The AUC value for the widely used antiarrhythmic and antihypertensive beta-blockers was 0.960398, which is easily explained by their similar chemical structure and side-effects. Another example is the class of SSRIs (0.942902), which are highly popular antidepressive drugs.

Besides the positive prior examples, we also conducted a qualitative evaluation of using negative ones, through a case study in Parkinson's disease with the participation of pharmacological experts. Using a query of MAO-B inhibitors, the tool correctly identified further dopamine releasing, dopamine reuptake inhibiting and MAO-B inhibiting drugs, which can alleviate the symptoms of the disease. To eliminate reasonable, but somewhat trivial results (*e.g.* amphetamines), "negative" training pairs were used. The details of the evaluation will be published in a separate article, along with the details of the derivation, necessary proofs and implementation issues.

V. Future work

The outlined framework extends the scope of data and knowledge fusion methods by allowing the integration of multiple quantiative similarities (kernels) and qualitatively similar and dissimilar pairs. Besides the multiple kernel extension, the use of such pairwise priors allows the specification of prior constraints corresponding to incomplete or even inconsistent clustering, which is frequently the case with real-world priors. The proposed method thus provides a general framework for the fusion of heterogeneous quantitative and qualitative information from multiple sources. Promising directions for future research are as follows:

- Evaluation: full-fledged multiclass evaluation.
- Computational extensions: optimization of kernel space embedding, dimensionality reduction [8].
- Mathematical extensions: quantitative constraints, application for FSS and density estimation.
- Implementation: an OpenCL-based implementation for GPU clusters.

References

[1] A. J. Izenman, "Introduction to manifold learning," *Wiley Interdisciplinary Reviews: Computational Statistics*, 4(5):439–446, 2012.

[2] S. Wang, Q. Huang, S. Jiang, and Q. Tian, "Efficient lp-norm multiple feature metric learning for image categorization," in *Proceedings of the 20th ACM International Conference on Information and Knowledge Management*, CIKM '11, pp. 2077–2080, New York, NY, USA, 2011. ACM.

[3] L. Yang and R. Jin, "Distance Metric Learning: A Comprehensive Survey," Tech. Rep., Department of Computer Science and Engineering, Michigan State University, 2006.

[4] A. Bellet, A. Habrard, and M. Sebban, "A survey on metric learning for feature vectors and structured data," *CoRR*, abs/1306.6709, 2013.

[5] A. Arany, B. Bolgar, B. Balogh, P. Antal, and P. Matyus, "Multi-aspect Candidates for Repositioning: Data Fusion Methods Using Heterogeneous Information Sources," *Curr. Med. Chem.*, Nov 2012.

[6] S. V. N. Vishwanathan, Z. sun, N. Ampornpunt, and M. Varma, "Multiple kernel learning and the smo algorithm.," in *NIPS*, pp. 2361–2369. Curran Associates, Inc., 2010.

[7] B. Bolgar, A. Arany, G. Temesi, B. Balogh, P. Antal, and P. Matyus, "Drug repositioning for treatment of movement disorders: from serendipity to rational discovery strategies," *Curr Top Med Chem*, 13(18):2337–2363, 2013.

[8] R. Chatpatanasiri and B. Kijsirikul, "A unified semi-supervised dimensionality reduction framework for manifold learning," *Neurocomput.*, 73(10-12):1631–1640, June 2010.

[9] M. R. Hurle, L. Yang, Q. Xie, D. K. Rajpal, P. Sanseau, and P. Agarwal, "Computational drug repositioning: from data to therapeutics," *Clin. Pharmacol. Ther.*, 93(4):335–341, Apr 2013.

[10] M. Kuhn, M. Campillos, I. Letunic, L. J. Jensen, and P. Bork, "A side effect resource to capture phenotypic effects of drugs," *Mol. Syst. Biol.*, 6:343, 2010.

[11] J. Hert, P. W. 0002, D. J. Wilton, P. Acklin, K. Azzaoui, E. Jacoby, and A. Schuffenhauer, "Comparison of fingerprint-based methods for virtual screening using multiple bioactive reference structures.," *Journal of Chemical Information and Modeling*, 44(3):1177–1185, 2004.

Acknowledgements

This research was conducted by a joint research team of computer scientists and pharmacologists. The author thanks Ádám Arany, Dr. Balázs Balogh, Dr. Péter Antal and Prof. Dr. Péter Mátyus for their contribution.

KERNEL CONSTRUCTION AND EVALUATION IN MICRORNA-BASED PRIORITIZATION

ANNUAL RESEARCH REPORT*

Péter MARX
Advisor: Péter ANTAL

I. Introduction

During the last year the main focus of my research was the application of kernel methods in different fields of bioinformatics. In this report I will present the background of this work with the evaluation of the used methods in miRNA prioritization. This part of the research is submitted to the BIOSTEC Bioinformatics conference [1].

The growing number of available measurement sources and the enormous data produced by them created a huge need of different data analysis methods like prioritization. Since gene sequencing become cheaper and a routine in diagnostics gene prioritization emerged as a standalone task. Using gene prioritization and fusion of multiple information sources one can define new gene–phenotype pairs which can lead to the discovery new drug targets. Although plenty of gene prioritization tools exist to our knowledge nobody used microRNAs (miRNAs) as a separate data source to extend the todays gene centric prioritization methods [2, 3, 4, 5].

Genetic regulation is an extensively studied field in biology and biostatistics. The process under which a cell produces a protein is modulated at multiple levels. Basically the regularizer methods can be classified into two groups *cis* and *trans* regulation. *cis* regulatory elements like miRNAs acts after transcription while *trans* regulation (e.g. transcription factors, histone modifications) effects the gene transcription directly.

The main goal of my research was to find new ways of constructing graphs and graphs kernels for genetic regulators. Moreover I used these kernels e.g to find new candidates in breast cancer.

II. Graph kernels

The constraints on a kernel (symmetric, positive definite) make the application of kernel methods in non-Euclidean spaces more difficult. In this section I'll describe a method to transform a graph to a kernel.

A. *Diffusion kernels*

Diffusion kernels [6] are exponential kernels which are symmetric and positive definite by definition.

$$e^{\beta H} = \lim_{n \to \infty} \left(1 + \frac{\beta H}{n}\right)^n \tag{1}$$

The limit in Eq. (1) always exists and is equivalent with the summation below:

$$e^{\beta H} = I + H + \frac{1}{2!}H^2 + \frac{1}{3!}H^3 + ... \tag{2}$$

Replacing n with $2n$ and choosing H to be symmetric the matrix exponential above will results in a positive semidefinite symmetric matrix.

[*] This paper is not an independent one, it is the summary of the PhD work of the last year presented in the minisymposium. It may contain parts of previously published documents or parts to be published later.

$$H_{ij} = \begin{cases} 1 & \text{i} \sim \text{j} \\ -d_i & \text{i = j} \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

where d_i is the degree of the ith node is a good choice. This way we can define kernels to undirected graphs. With the β parameter we can modulate how wide should the heat spread on the graph. These kernels define similarity between two nodes. If $\beta = 0$ hence $K = I$ every node is connected or similar to itself only. With increasing β the heat will spread out more and more. If there is a path between two nodes i, j, K_{ij} could be higher than 0 depending on the length of the path.

III. Diffusion kernels on protein–protein interaction networks

Protein–protein interaction (PPIN) networks usually are highly connected networks and can contain isolated subnetworks. Diffusion kernels could be a good choice for the kernel characterization of these kind of networks with choosing appropriate β values. I wrote a MATLAB script to build diffusion kernels on the String PPIN and used them in an asthma prioritization task to smooth the gene expression array based kernel. I compared the results to a normalized Laplace based PPIN kernel. However the results were similar the normalized Laplace kernel performed better based on leave-one-out cross validation.

IV. Applying graph kernels for miRNA prioritization

However, the available *in silico* miRNA–target prediction tools extend the set of miRNA–target gene pairs it introduces a bias since an extensive evaluation of these methods is still necessary. For this reason I used miRTarBase [7] (downloaded 11/14/2013 Release: 4.5) and filtered out the non-human target genes. This way I had only experimentally supported miRNA–target pairs which were validated by different experiments. These experiments allow the classification of the validation as e.g. gene expression studies provide indirect evidence for miRNA regulation which results in a "weak" evidence. On the other hand reporter genes attached to the target gene give direct proof of the reaction as a "strong" evidence. The database contained 596 different miRNAs.

A. Kernel computation

Similarity between miRNAs is defined by the common targets so two miRNAs are more similar if they have more common target genes. The similarity matrix were squared which means that two miRNAs are more similar not only in that case if they share more target genes but also if they have common similar miRNAs. Squaring the symmetric similarity matrix also ensures positive definiteness which is required for a kernel. I tested two methods: First I created a kernel from the full set of miRNA–target pairs. Beside this I created four different kernels based on the weak or strong classes and the functional and nonfunctional miRNA target pairs. These kernels were used with Multiple Kernel Learning [8, 9, 10, 11].

B. Results

I used breast cancer as a test dataset as it is a well studied type of cancer. I used multiple queries based on the miRNAs connected to breast cancer by Samantarrai et. al [12]. These queries could be divided into two categories. The first type of miRNA training sets are subsets of BRCA subtype ductal carcinoma *in situ* upregulated miRNAs. With these queries I investigated whether the prioritization gives a lower rank to the left out miRNAs from the same subtype and regulation type. The other category contained queries with miRNAs connected to more BRCA subtype to build a whole BRCA kernel and examine the rank of other BRCA connected miRNAs. Query1 falls into the first category while Query3 in the second one.

microRNA	Query1		Query3	
	Full	MKL	Full	MKL
hsa-miR-106b	11	51	12	79
hsa-miR-141	51	27	375	386
hsa-miR-15b	26	11	47	30
hsa-miR-17	41	20	85	41
hsa-miR-19b	13	24	34	36
hsa-miR-200c	6*	5*	280	95
hsa-miR-20a	43	28	92	88
hsa-miR-21	2*	2*	14*	3*
hsa-miR-92a	16	31	3	18
hsa-miR-140	61	22	121	80
hsa-miR-182	3*	6*	16*	2*
hsa-miR-183	1*	1*	15*	1*
hsa-miR-186	76	18	68	34
hsa-miR-25	10	25	9	37
hsa-miR-320a	107	87	32	43

Table 1: The miRNAs ranked below 50 from the Jacobsen dataset. *The miRNA was part of the query. [1]

I validated the results with leave-one-out cross validation (results not shown here). With the MKL method the average rank fall below 10% while for the full kernel it was between 15% and 20%. The full kernel gave only in one case higher rank to the left out miRNA (hsa-miR-374a). On the other hand the MKL method performed better in all other cases and sometimes the difference is pretty high (hsa-miR-200c, hsa-miR-361-5p).

In prioritization tasks the measure of success is often computed based on the data used for the kernel computation which introduces a bias. To avoid this I used prospective evaluation as well. I compared the results computed based on Samantarrai et. al [12] from May 2013 to a paper published later in 2013 by Jacobsen et. al [13]. Results can be seen in Table 1.

The MKL method gives a weight to every used kernel. In my study the kernel weights was in all cases similar. The weak-functional kernel had ~0.4 and all the others ~0.2.

C. *Discussion and Conclusions*

The two validation method showed that in case of better studied diseases it is possible to find newer miRNAs which can be connected to the disease with laboratory experiments. Despite the promising results I have to deal with the following questions:

- There are other possible similarity definitions which will make the dataset truly heterogeneous;
- The validation of the *in silico* prediction methods can extend the dataset;
- Finally I should extend the available prioritization methods with this new source which can lead to a better descriptions of genetic regulatory networks.

The MKL method performed better in all cases with all queries. The distribution of the weights are possibly caused by the different number of connections available for the different kernels. Both methods ranked most of the genes from the Jacobsen paper to the top 10%. Beside the statistical importance of this it has biological relevance also. In a real study the researcher cannot validate all the miRNAs even with this small number because of budget limitations. Trying to find new candidates in the top 10% is a common method.

V. Future work

First I will define other miRNA–miRNA similarities based on sequence distance, miRNA family clusters and build miRNA kernels based on these definitions. Beside this I will extend the miRNA database with different information sources with experimentally supported new miRNA–target pairs. After I have a complete set of published miRNAs I will examine the possibility of using different kernel computation methods like diffusion kernels and compare the results. Later I am planning to create a whole kernel library for bioinformatics. In addition I will continue my Bayesian Feature Selection research to extend the kernel methods and make use of continuous variables.

References

[1] P. Marx, B. Bolgár, A. Gézsi, A. Gulyás-Kovács, and P. Antal, "Microrna prioritization based on target profile similarities," 2014.

[2] J. Freudenberg and P. Propping, "A similarity-based method for genome-wide prediction of disease-relevant human genes," *Bioinformatics*, 18:S110–S115, 2002.

[3] S. Aerts, D. Lambrechts, S. Maity, P. Van Loo, B. Coessens, F. De Smet, L. Tranchevent, B. De Moor, P. Marynen, B. Hassan, P. Carmeliet, and Y. Moreau, "Gene prioritization through genomic data fusion," *Nature Biotechnology*, 24(5):537–544, 2006.

[4] S. Kohler, S. Bauer, D. Horn, and P. Robinson, "Walking the interactome for prioritization of candidate disease genes," *American Journal of Human Genetics*, 82(4):949–958, 2008.

[5] Y. Moreau and L. Tranchevent, "Computational tools for prioritizing candidate genes: boosting disease gene discovery," *Nature Reviews Genetics*, 13(8):523–536, 2012.

[6] R. I. Kondor and J. Lafferty, "Diffusion kernels on graphs and other discrete structures," in *In Proceedings of the ICML*, pp. 315–322, 2002.

[7] S.-D. Hsu, F.-M. Lin, W.-Y. Wu, C. Liang, W.-C. Huang, W.-L. Chan, W.-T. Tsai, G.-Z. Chen, C.-J. Lee, C.-M. Chiu, et al., "mirtarbase: a database curates experimentally validated microrna–target interactions," *Nucleic acids research*, 39(suppl 1):D163–D169, 2011.

[8] B. Schölkopf, J. C. Platt, J. Shawe-Taylor, A. J. Smola, and R. C. Williamson, "Estimating the support of a high-dimensional distribution," *Neural computation*, 13(7):1443–1471, 2001.

[9] T. De Bie, L.-C. Tranchevent, L. M. Van Oeffelen, and Y. Moreau, "Kernel-based data fusion for gene prioritization," *Bioinformatics*, 23(13):i125–i132, 2007.

[10] Z. Sun, N. Ampornpunt, M. Varma, and S. Viswanathan, "Multiple kernel learning and the smo algorithm," in *Advances in neural information processing systems*, pp. 2361–2369, 2010.

[11] S. Yu, T. Falck, A. Daemen, L.-C. Tranchevent, J. A. Suykens, B. De Moor, and Y. Moreau, "L2-norm multiple kernel learning and its application to biomedical data fusion," *BMC bioinformatics*, 11(1):309, 2010.

[12] D. Samantarrai, S. Dash, B. Chhetri, and B. Mallick, "Genomic and epigenomic cross-talks in the regulatory landscape of mirnas in breast cancer," *Molecular Cancer Research*, 11(4):315–328, 2013.

[13] A. Jacobsen, J. Silber, G. Harinath, J. T. Huse, N. Schultz, and C. Sander, "Analysis of microrna-target interactions across diverse cancer types," *Nature structural & molecular biology*, 2013.

Application of Phase Correlation in Tomosynthesis

Annual research report[*]

Áron HORVÁTH
Advisor: Gábor HORVÁTH

I. Introduction

Phase correlation (PC) is a method of image registration and uses a fast frequency-domain approach to estimate the relative translative offset between two similar images. In this paper an application of this method will be presented which improves the precision of tomosynthesis reconstruction.

PC calculates the normalized cross-correlation (NCC) of two images by Fourier transform, and returns the translative offset as the location of the highest peak on the NCC.

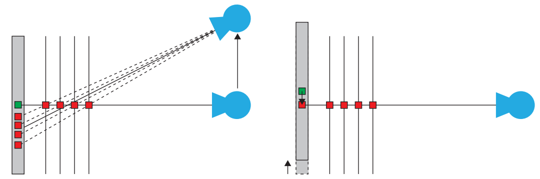
II. Dislocation correction

In a digital tomosynthesis system, the accuracy of the detector and source locations is essential for a clear reconstruction. If the mechanical system cannot provide this accuracy, it have to be provided either by measurement during exposition or by software postprocessing of the projected images.

The projected volume can be regarded as a set of slices parallel to the plane of the detector. Let us consider the case, where each slice contains only one dot in the middle. If the detector is displaced along the line of movement while the source stays steady, the projected image will also be a dot, but shifted in the opposite direction of the displacement as depicted on Fig. 1b. If the source is displaced the same way, the projected image will contain a line segment starting from the middle towards the direction of the displacement (see Fig. 1a).

If we consider a case where only one slice contains data (real data), the PC between two projections will give us the amount of dislocation. On its NCC image there will be a relatively sharp dot displaced from the middle by the amount of detector displacement. If we fill up the rest of the slices with data, as well, the NCC image will be the sum of the individual NCC images as the operation is linear. Thus we can get the same image as with the slices above with only one dot in the middle.

For the detector displacement case the PC directly gives the displacement. For the source displace-



(a) Effect of source displacement

(b) Effect of detector displacement

Figure 1: Illustration of where the center of the slices are projected onto the detector, if either the detector or the source is displaced. The green point represents the projections without displacement (all points are overlapped), the red ones denotes the projections after displacement.

[*] This paper is not an independent one, it is the summary of the PhD work of the last year presented in the minisymposium. It may contain parts of previously published documents or parts to be published later.

ment case it becomes a bit more difficult, because the NCC will not provide us a sharp dot. However the average offsets of the slices on the projections provide us clues to determine the amount of displacement of the source. The average offset can be obtained by simply calculating the center of gravity of the NCC image.

Normally both the source and the detector are moving simultanously, thus the NCC of each pair of projections will look like a blurred line-segment (see Fig. 3a). If we assume the source position to be accurate and want to calculate the real position of the detector from two adjacent projections (of which position is unreliable for some reason), we face with an NCC image like above, thus the calculation of the center of gravity is needed.

Not only for solving the above problem the calculation of the center of gravity is useful, it also brings subpixel precision, which could not be achieved by simple peak picking.

In real life one cannot know in advance where the displacement will occure, at the detector or at the source, generally it occures at both, but with different extent. Separation of the two is a bit elaborate, because both phenomena are realized on the NCC image as a displacement, however the source displacement involves a spreading in the direction of movement as well, of which extent is proportional to the displacement. This gave the line segment in Fig. 1a. In the remaining part of this paper, we will focus on solving the problem only for the detector displacement, with accurate source positions (This one is much more important than the one for source case, because inaccuracy of the detector position can cause up to 8 times more error during recostruction, because the object is much closer to the detector). But the presented method can be extended to the source dislocation, or even to dislocations in other directions, as well.

III. Detector dislocation correction algorithm

The algorithm is fairly simple. We apply median filtering on each projection image. (This is useful because PC is more sensitive to higher frequencies, and the image noise mostly manifests in the higher frequency domain). Then for each adjacent median filtered projections we calculate the I_{NCC} NCC image. The center part of an image like this is shown in Fig. 3a. It looks a bit cluttered, but the blurred linesegment is clearly noticable in the middle. This patch in the middle is the useful part, the rest is coming from noise, windowing effect, selfsimilarities or for the case of simulation it can even arise from interpolation errors. To keep only this central patch, we apply a thresholding on the $[0..1]$ normalized NCC image with threshold h:

$$f(x) = \begin{cases} 0 & \text{if } x \leq h \\ x - h & \text{otherwise} \end{cases} \tag{1}$$

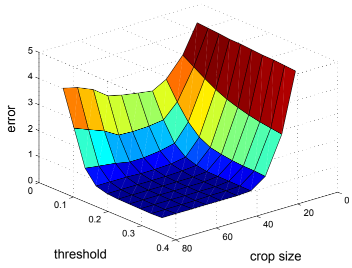
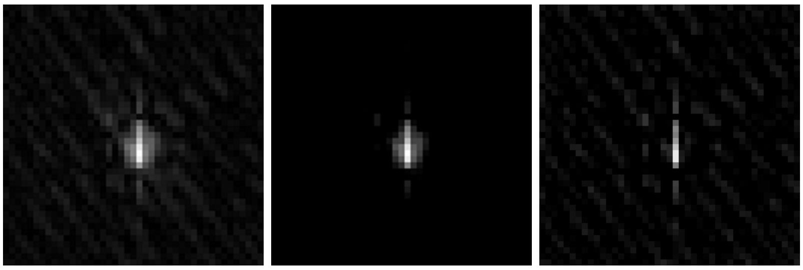


Figure 2: The measured PC error (pixel). For various thresholds and crop sizes.

However this operation is not linear, the summation of the effect of each slice will not hold further, but it is still a close approximation. This operation does not guarantee that far from the center everything is set to zero, therefore a cropping is applied as well. The size of the cropping and the threshold



(a) Original (b) After thresholding (c) After unsharp filter

Figure 3: PC images of two adjacent real life projections. Only the cropped part.

are intertwined, as it is depicted in Fig. 2. Smaller σ crop sizes slightly reduce the error, but too small ones deteriorate the whole method. The same stands for the threshold value. The optimal combination is around $h = 0.3$ and $\sigma = 45$, but it depends on the position noise level as well. Fig. 2 confirms that without thresholding (which means $h = 0$) there is no chance for decent accuracy.

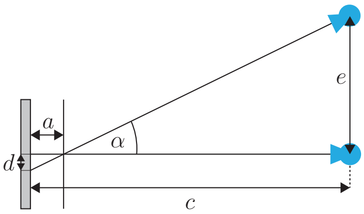


Figure 4: Calculating effective focal plane.

Estimating the detector positions from the center of gravities

If we know the exact source positions of the expositions, we can calculate the positions of the detector from the NCC images of the adjacent projections. (We deal only with the position along the axis of movement)

As shown in Fig. 4 one can calculate the distance from the detector to the effective focal plane a (this focal plane does not necesserily coincide with the center of the body, because homogenous parts of the body do not have effect on PC) from the distance between the detector and source planes $c = a + b$, if a displacement at the source e is measured on a steady detector as d:

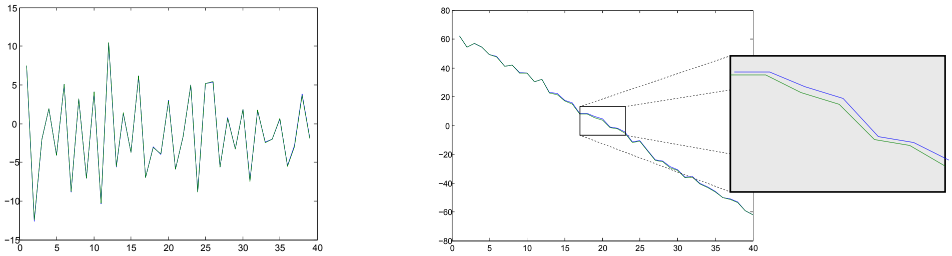
$$\tan\alpha = \frac{d}{a} = \frac{e}{b} \tag{2}$$

$$a = \frac{c\frac{d}{e}}{1 + \frac{d}{e}} \tag{3}$$

Let us denote the measured offset on the detector between the i^{th} and the $i + 1^{\text{th}}$ projections with t_i. Using the effective focal plane location a, the estimated detector position is:

$$\hat{d}_i = \frac{a}{b}\tilde{e}_i - \sum_{j=1}^{i-1} t_j + \frac{1}{2}\sum_{j=1}^{n-1} t_j \tag{4}$$

where $\tilde{e}_i$ denotes the i^{th} reliable source location. The complete derivation of this formula is omitted due to space restrictions.



(a) Interprojection offsets in mm.

(b) simulated and estimated positions

Figure 5: The estimated offsets and positions. Blue is the simulated, green is the estimated.

	Mean		RMS	
	position error	offset error	position error	offset error
with median filter	0.3518	0.0332	0.3854	0.0445
without median	1.1532	0.0988	1.2777	0.3150

Table 1: Results. (in mm)

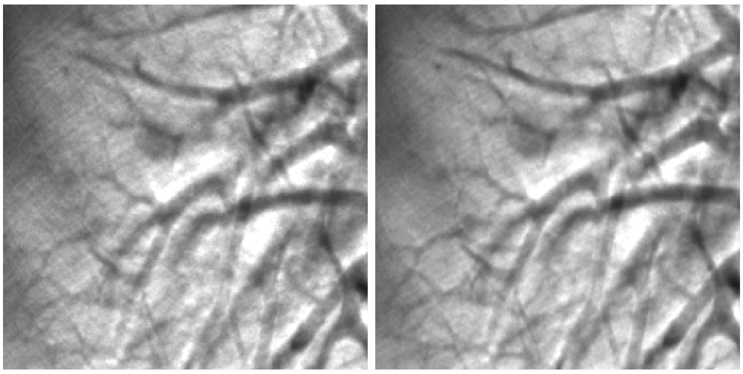
IV. Results

The method was tested on simulated tomosynthesis projections. The 1203 x 512 x 481 voxel volume data came from a high resolution CT image. The detector-source distance was set to 1.8m. The size of the detector was 423mm, and its resolution was 800x800. The length of the source and detector trajectory was 1184mm and 124mm respectively. The volume was placed 50 mm away from the detector and its largest extension was 600mm. The number of projections was 40.

A gaussian noise of 2mm standard deviation was added to the detector positions (except for the start and end positions) before projecting. The position estimator was run on this set of projections with and without median filtering, and the results are summerized in table 1. The simulated and the estimated offsets at the detector are compared in Fig. 5a, and the simulated and the estimated detector positions in Fig. 5b. The offset error denotes the errors of t_i values. The cropping size was set to 50 pixel, and the threshold was set to 0.27.

The results show that using median filtering before PC significantly (∼3 times) improves precision. The reached subpixel accuracy of this PC is better than 0.1 pixel. However position error is more than 10 times larger than the offset error which implies, that the offset error can accumulate, which probably can be mitigated if PC was applied not just between adjacent projections.

Image 6 shows a real life application of the method.



(a) Without correction

(b) With correction

Figure 6: Same patch from a reconstructed slice with and without using displacement correction.

PREDICTING GRAPH QUERY PERFORMANCE BY METRICS*

ANNUAL RESEARCH REPORT†

Benedek IZSÓ
Advisor: István RÁTH

I. Introduction

Nowadays, model-driven software engineering (MDSE) plays an important role in the development processes of critical embedded systems. With the dramatic increase in complexity, modeling toolchains are facing scalability challenges as automated tool features become more sophisticated, which was a main motivation behind the MONDO project [2]. For tool engineers, *benchmarks* may provide guidance on picking the right technology for building a new tool architecture to fulfill increasing scalability requirements. Due to their central role in data-intensive applications, performance has been evaluated by previous benchmarks [3, 4, 5]. However, current benchmarks have their limitation: they use one data model [4], queries are ad-hoc, their features are described only verbally [3] or measure latest developments of one specific tool [5]. The generalizability of benchmark results is further limited by the *scarcity of relevant metrics* that could be used to assess an engineering problem and predict which technology would be best suited. Existing metrics emphasize a single aspect of the problem (typically model size), while internal metrics of an engine are either not documented well or are not accessible.

In our paper [1], we aimed at addressing these challenges by *assessing existing metrics, and also proposing new metrics for measuring difficulty*. We outline a benchmark in Sec. II. that uses model validation as its core scenario, thus focusing on model loading and model validation workloads. Then we collect metrics characterizing models, queries, and briefly describe how queries and instance models were generated. To identify which metrics provide reliable performance prediction for a given workload, in Sec. III. we *calculate the correlation with significance values* between execution times and metrics for three characteristically different graph query tools. Finally, we conclude our paper in Sec. IV.

II. Benchmarking and Metrics

Following the model-driven development approach, early design validation, transformation to mathematical models for formal verification or code generation can be used to improve software quality. These are all query-intensive operations, thus to measure a similar use-case, we opted for a batch model-validation scenario, where the underlying model is checked against a set of predefined constraints (formulated as queries). During the benchmark, the read time, checking time (query execution time) and memory consumption were measured. To identify sensitivity of tools for the changing of parameters of models and queries, we defined metrics to numerically describe the input.

A. Model validation scenario

In the *read phase* (see Fig. 1) a previously generated instance model is loaded from hard drive to memory, and constraints are loaded as model queries. The phase includes initialization of data structures used by the tool, possibly indexes or caches.

† The paper is based on article [1]. I would like to thank Zoltán Szatmári, Gábor Bergmann and Ákos Horváth for their work in this project.

* This work was supported by the MONDO (EU FP7 ICT-611125) project.

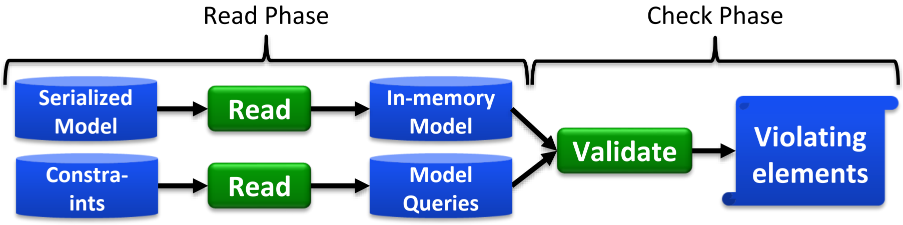


Figure 1: Phases of a model validation scenario

In the second, *check phase* the instance model is validated. This means that queries are executed on models, reporting constraint violating elements, which must be available in a list at the end. Theoretically, cache or index building and initial model traversal can be deferred from the read phase to the check phase, but it depends on the actual tool implementation.

B. Metrics

The performance of different tools depends on the model and the evaluated query, thus we defined metrics characterizing them. An instance model in this case is a graph, so general graph metrics can be used. The number of *nodes*, *edges*, or in total the number of *elements* may affect performance (e.g.: read time), as well as *maximum, minimum* and *average in-* or *out degree* of nodes (e.g.: during traversal). In a labeled graph the number of *edge* or *relation types* can affect processing time, especially, when indexes are built on them.

For graph pattern queries, metrics can be derived from language elements. Join performance can be affected by the *number of edge constraints*. *Node constraints* and *attribute checks* filter results, while *negations* test the speed of antijoin. *Variables* may affect performance as they must be bound to model elements, while *parameters* can not be projected away which may influence processing speed of (combined) queries.

Combined metrics describe both the query and the model. Such metric is the number of matches, and selectivity, which is defined as $\frac{matches}{elements}$. We propose two new query metrics for measuring difficulty of a query on a model. Let c be the ways a constraint can be satisfied, and let *absDifficulty* be defined as $ln \prod_c (1 + |c|)$. This gives (the logarithm of) a gross upper bound of the possible query evaluation cost. The *relDifficulty* calculates difficulty per matches, thus defined as $ln\frac{\prod_c (1+|c|)}{1+countMatches}$.

C. Generating models and queries

For benchmarking purposes 12 models were generated with different metrics (size, edge degree). As real-world models show different edge distribution (e.g.: scalefree for social networks, regular for software models), models were generated with regular, binomial, hypergeometric and scalefree distributions, although this property can not be directly described by a metric. 31 queries were constructed that can be divided into 5 categories, each scaling different metrics. The number of results ranged from zero to the number of all elements.

III. Evaluation

A. Measured technologies

An imperative *local search-based* approach was implemented in Java, operating on *Eclipse Modeling Framework (EMF)* [6] models. EMF is a model description language similar to UML class models. Queries are implemented as Java functions, traversing a model without any search plan optimization, but they cut unnecessary search branches at the earliest possibility.

The second tool is EMF-IncQuery, which provides a declarative graph pattern (GP) language [7] for describing queries. Generally, a result of a GP is a set of parameter bindings satisfying constraints of the query such as type, relation, attribute restrictions. Negations express the absence of specific patterns. The engine evaluates GPs *incrementally* based on the concepts of Rete nets over EMF models.

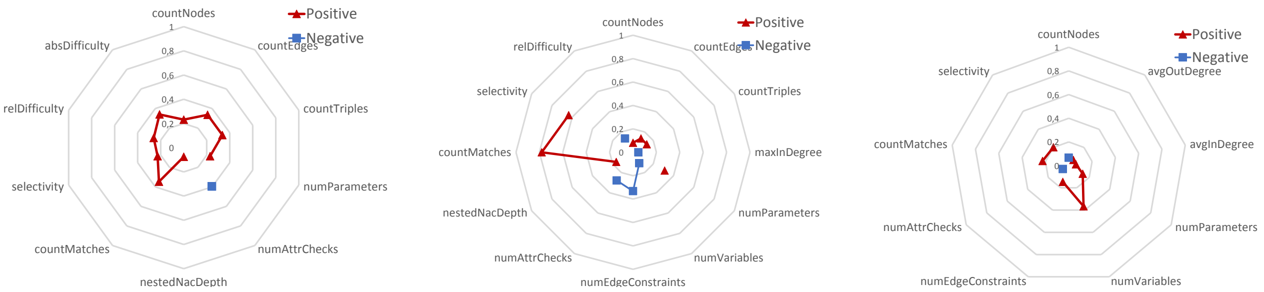
The third tool is a *black-box* execution engine as implemented in the Sesame framework. It uses the Resource Description Framework (RDF) [8] as model description language. RDF builds up the graph from {subject, predicate, object} triples, where the subject and object is connected by a relation, labeled with “predicate”. Sesame uses the SPARQL [9] GP based query language that is evaluated over RDF models.

B. *Method of Analysis*

We attempted to find out which model and query metrics are useful for predicting the performance of the various tools. A given metric can only be a useful predictor of a certain performance indicator of a certain tool if they are strongly correlated. Therefore our analysis used Kendall’s τ rank correlation coefficient, without assuming a linear model. Correlation coefficients may lead to incorrect conclusions if the sample of models and queries is too small. Therefore, whenever measuring an absolute value of τ, we additionally conducted the associated statistical test (τ-test), and considered only statistically significant ($p < 0.001$) correlation values. Note that any such statistical result is conditional to the uniform sampling of the selected queries and models.

C. *Results*

Consistently with previously published results, data show that model size is a strong predictor of both model loading time and memory consumption, regardless of the technology.



(a) Java Check time

(b) EMF-IncQuery Check time

(c) Sesame Check time

Figure 2: $|\tau|$ of correlating ($p < 0.001$) metrics for check time indicator

However, check times show a more diverse picture (see Fig. 2). The check times for the Java implementation (being a dominantly search-intensive approach) are additionally correlated with the query-on-model metrics as well, with the strongest correlation shown by the *absDifficulty* metric. Interestingly, the best Sesame check time predictor turned out to be the number of pattern variables, and there is no significant correlation with any direct model metrics. Check times of EMF-IncQuery are very strongly correlated with the number of matches – which is to be expected of an incremental tool whose check phase consists of just enumerating the cached match set. As the incremental indexes are constructed during the load time, the model-on-query metrics become correlated with EMF-IncQuery read time. It can also be highlighted that EMF-IncQuery seems not to be sensitive to the “difficulty” of the query (in any phase) or the model size (during the check phase) due to the very small correlations with corresponding metrics.

Overall, it can be said that model-only metrics are useful in predicting the performance of model persistence operations. However, query-based and combined metrics (such as our proposed *abs-* and *relDifficulty*) are necessary to provide a more thorough picture. Note that since only statistically significant correlations are included, a low magnitude correlation does not necessarily mean a measurement error. It is possible that there is a true link between the two variables, but the τ value is lowered by other metrics that strongly influence the performance indicator.

D. *Threats to Validity*

Regarding the technological foundations and methodology of our measurements, the most important threats to validity stem from *time measurement uncertainty* and distortions due to transient effects such as *garbage collection* in the JVM and *thrashing* due to heap size exhaustion. Such effects were mitigated by using the most accurate Java time measurement method (*System.nanoTime*), allocating as much heap as possible, and using a timeout mechanism to identify and exclude cases affected by thrashing from the results. Additionally, it is also possible that there is sampling bias in our choice of models and metrics; we believe that this is sufficiently mitigated by our systematic choice of model generation strategies and the design principles of the queries (see Sec. II.C. or [1] for details).

IV. Conclusion and Future Work

Based on an analysis of MDSE scalability issues and previously proposed model and query metrics, we devised a set of metrics that characterize the complexity of instance models and queries. We designed a benchmark to effectively differentiate various model representation and query evaluation approaches with respect to their scalability. To verify our approach with real-life MDSE tools, we conducted an initial experimental evaluation and found that our methodology is useful for distinguishing key tool characteristics. In addition to model size, we identified several additional metrics that correlate with query performance, which may be useful for the design and optimization of modeling applications.

From the subtasks of this joint work I implemented the instance model generator, its metrics calculator and the benchmark framework. I executed these generations, calculations and benchmarks, which provided artifacts that I could feed into my analyzer scripts.

As a primary direction for future work, we plan to extend the benchmark into a comprehensive evaluation. Real-life models and queries should be involved, the workload should support incremental modification and re-evaluation, and a large spectrum of tools should be measured. Using these results decision trees could be built, more sophisticated statistical analysis could be performed (also supporting correlating variables), aiding domain engineers to choose the right tools and languages for their task.

References

[1] B. Izsó, Z. Szatmári, G. Bergmann, Á. Horváth, and I. Ráth, "Towards precise metrics for predicting graph query performance," in *28th IEEE/ACM International Conference on Automated Software Engineering (ASE 2013)*, pp. 412–431, 11/2013 2013.

[2] MONDO, "Scalable modelling and model management on the cloud," `http://cordis.europa.eu/projects/rcn/110313_en.html`, Accessed: 2013. December 31.

[3] C. Bizer and A. Schultz, "The Berlin SPARQL Benchmark," *International Journal On Semantic Web and Information Systems*, 5(2), 2009.

[4] M. Schmidt, T. Hornung, G. Lausen, and C. Pinkel, "SP2Bench: A SPARQL performance benchmark," in *Proc. of the 25th International Conference on Data Engineering*, pp. 222–233, Shanghai, China, 2009. IEEE.

[5] O. Erling, "Benchmarks, redux," `http://www.openlinksw.com/weblog/oerling/?id=1658`, 2011, Accessed: 2014. January 1.

[6] The Eclipse Project, "Eclipse Modeling Framework," `http://www.eclipse.org/modeling/emf/`, Accessed: 2013. December 31.

[7] G. Bergmann, Z. Ujhelyi, I. Ráth, and D. Varró, "A Graph Query Language for EMF models," in *Theory and Practice of Model Transformations, ICMT 2011*, J. Cabot and E. Visser, Eds., vol. 6707 of *LNCS*, pp. 167–182. Springer, Springer, 2011.

[8] World Wide Web Consortium, "Resource Description Framework (RDF)," `http://www.w3.org/TR/2004/REC-rdf-primer-20040210/`, 2004, Accessed: 2013. December 31.

[9] World Wide Web Consortium, "SPARQL Query Language for RDF," `http://www.w3.org/TR/sparql11-overview/`, 2013, Accessed: 2013. December 31.

THE SENSORLESS PRINCIPLE IN SOLENOID ACTUATORS

ANNUAL RESEARCH REPORT[*]

Ivor DÜLK
Advisor: Tamás KOVÁCSHÁZY

I. Introduction

This paper summarizes my research contributions in the year of 2013. My research is dedicated to the modeling and sensorless control of solenoid actuators. Solenoid actuators are electromechanical converters most commonly used for flow control purposes in hydraulic and pneumatic systems. A survey in literature had shown that for solenoid actuators the thermal effects are not satisfactorily incorporated in the models that are used for the sensorless concept; therefore, changes in the external and internal thermal boundary conditions during normal operation may result in considerable estimation bias because e.g. the resistance of the coil changes (40% for 100°C). During the last year my research aimed at developing thermal models and methods for estimating or measuring the coil's resistance in order to compensate thermal bias in the sensorless principle.

From one point, the internal thermal state of the actuator (and the resistance) can be acquired from an in depth, dynamic thermal model. Major results are discussed in Sections II-III. On the other hand, I also studied the possibilities of directly "measuring" or "estimating" the coil's resistance from electrical signals (Section IV). Both approaches have advantages and disadvantages, and their usability depends on the actual field of application.

II. Thermal Modeling of Solenoids

A possible way for estimating the resistance and internal thermal state of solenoids is by constructing a dynamic thermal model that is evaluated continuously according to the thermal boundary conditions. In this approach, we preferred to create a physical thermal model for solenoid actuators and to derive an analytical solution to the thermal diffusion problem because:

- Analytical solutions to a physical model give full information about the thermal state e.g. hot-spots, average temperature;
- Give direct insight into the physical processes and parameter dependence; which are important for optimization and design,
- They are closed-form compact solutions that are computationally less exhaustive; therefore, suitable for embedded applications.

The solenoid is considered as a multi-layered cylindrical structure that is layered along the radial direction. The main modeling principles and the underlying problem are referred as the "multilayer problem" or heat diffusion in multi-layered medium in the related literature. The system is subject to transverse (in the direction perpendicular to the layers) and longitudinal (in the directions parallel to the layers) excitations (boundary conditions). For the governing PDE (Partial Differential Equation-heat diffusion) an analytical solution is sought.

A popular method for deriving an analytical solution to such problems is the orthogonal expansion technique, in which the solution is expressed as the infinite set (sum) of orthogonal functions (eigenfunctions). For example, heat diffusion in 2D Cartesian coordinates with an exemplary solution can be written as (1). The eigenfunctions depend on the actual form of the PDE, and the

[*] This paper is not an independent one, it is the summary of the PhD work of the last year presented in the minisymposium. It may contain parts of previously published documents or parts to be published later.

orthogonality condition is ensured by the eigenvalues ("λ" in (1)). The infinite number of distinct eigenvalues is computed from the transcendental equations, which are constructed from the homogeneous boundary conditions on the opposite surfaces. In technical literature, this step is referred as the eigenvalue problem; and it can become quite complicated and computationally consuming with multiple layers, especially if longitudinal boundary conditions are non-homogeneous. The term non-homogenous refers to a nonzero source on a certain surface.

We observed that existing analytical solutions to multi-layered heat conduction considered non-homogeneous boundary conditions in the transverse direction, but non-homogeneous boundary conditions in the longitudinal directions could be included in the solution only under special considerations. As a major contribution, we developed a general method to include the effect of non-homogeneous longitudinal boundary conditions in the analytical solution. The presented method also simplified the associated eigenvalue problem thus reducing computational complexity of the solution; and some forms of internal heat generation could be also considered in the solution more comfortably. The corresponding article can be found in [1]. The main concept is that the multilayer structure is decomposed into single layers, which can be solved readily, and then the effects of the interfaces (junctions) are considered for the multilayer media in a far less complicated multilayer problem with homogeneous longitudinal boundary conditions.

III. Computation of the Analytical Solution of the Heat Equation in Multi-Layered Media

The computation of the analytical solution of the temperature distribution in multi-layered media can become numerically unstable if there are different longitudinal (i.e., the directions parallel to the layers) boundary conditions for each layer [3]. We developed a method, which has been submitted and being reviewed in [2], for resolving these computational difficulties. The illustration of the problem is as follows:

The analytical solution of the steady state temperature in the slabs is determined using separation of variables. The homogeneous longitudinal boundary conditions motivate the form of the thermal field in each slab (e.g., slab *#j*) given in Eq. (1). In Eq. (1), the term "T_j" denotes the 2D thermal field in slab "*#j*", and the coefficients "$a_{j,n}$", "$b_{j,n}$", "$c_{j,n}$", and "$d_{j,n}$" are the constants associated with the n^th eigenfunction of the "T_j" thermal field. The term "$\lambda_{j,n}$" is the eigenvalue associated with the n^th eigenfunction.

$$T_j(x_j, y) = \sum_{n=1}^{\infty} X_{j.n}(x) \cdot Y_{j,n}(y) = \sum_{n=1}^{\infty} \left(a_{j,n} \cdot e^{\lambda_{j,n} \cdot x_j} + b_{j,n} \cdot e^{-\lambda_{j,n} \cdot x_j}\right) \cdot \left(c_{j,n} \cdot \sin(\lambda_{j,n} \cdot y) + d_{j,n} \cdot \cos(\lambda_{j,n} \cdot y)\right) \quad (1)$$

The "$\lambda_{j,n}$" eigenvalues can be calculated from the corresponding transcendental equation, which is derived from the homogeneous longitudinal boundary conditions. To determine the analytical solution of the steady-state temperature in slab "*#j*", the four parameters in Eq. (1) for each eigenfunction must be found. The constants "$c_{j,n}$" and "$d_{j,n}$" of the trigonometric eigenfunctions can be simplified using the longitudinal boundary conditions (e.g., "$c_{j,n}$" or "$d_{j,n}$" are zero). In a single layer problem, the constants "a_n" and "b_n" can be computed straightforwardly from the transverse boundary conditions using the orthogonality of the longitudinal eigenfunctions (2).

$$\int_Y Y_{j,n}(y) \cdot Y_{j,m}(y) \cdot dy = \begin{cases} 0 & n \neq m \\ \neq 0 & n = m \end{cases} \quad (2)$$

If the kernel of the layers (longitudinal eigenfunctions) is the same for the layers; the "global" orthogonality of the longitudinal eigenfunctions (left side of (3)) can be exploited to compute the parameters of the hyperbolic eigenfunctions (i.e., the constants "$a_{j,n}$" and "$b_{j,n}$"). However, if the kernels are different the orthogonality condition becomes (right side of (3)) and every transverse eigenfunction of layer *"#j"* contributes to each transverse eigenfunction of the adjacent layer *"#k"*; resulting in an ill-conditioned system of linear equations for the hyperbolic parameters.

$$\int_Y Y_{j,n}(y)\cdot Y_{k,m}(y)\cdot dy = \begin{cases} 0 & n \neq m \\ \neq 0 & n = m \end{cases} \qquad \int_Y Y_{j,n}(y)\cdot Y_{k,m}(y)\cdot dy \neq 0 \tag{3}$$

In practice, all computational methods suffer from arithmetic errors, such as round-off and truncation error, which may cause the calculation process to become numerically unstable, in which case the "straightforward" approach (by inversion) to calculate the parameters of the analytical solution becomes unsuitable.

A The Proposed Computational Method

The computation of the analytical solution for a single layer problem is numerically stable. Therefore, we developed a computational method that is summarized as the following:

- First, a two-layered structure (layers A & B) is considered and the temperature at the junction is approximated step by step by computing the solution only in the single layers (Fig. 1));
- To speed up convergence, we propose a weighted successive overrelaxation and provide optimal weighting coefficients to use during the iterations;
- The two-layer method is generalized to an arbitrary number of layers on the basis of a hierarchical two-layer representation of the overall problem;
- A method to decompose the multi-layered media into a hierarchy of two-layered media is presented for further improving convergence speed and reducing computational load.

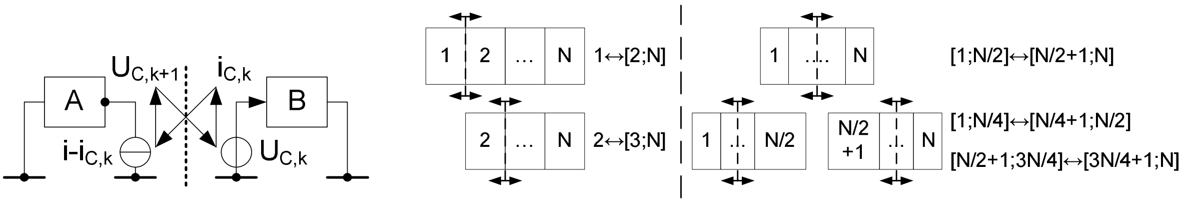


Figure 1: Concept of the computational method (left) and two layer decomposition (right).

IV. A Method for Estimating the Resistance of the Coil in Solenoid Actuators

In this Section an alternative approach is presented for estimating the resistance of a solenoidal actuator. Compared to Section II, where the goal was to construct a thermal model, here the resistance is estimated from electrical measurements. The electrical model consists of a temperature dependent resistance and of a current dependent inductance. If steady state is reached the resistance can be straightforwardly computed from the voltage and current signal irrespective of the actual magnetizing curve. However, in many real applications the system can undergo a long transient period i.e. the control signal changes; therefore, steady state is not reached although refreshed resistance data is necessary. We developed a method for estimating the resistance from the transient current step response considering a current dependent inductance, thus, a resistance estimate can be provided before reaching steady state. Furthermore, the inductance curve can be also identified with the proposed method. Considering the continuous case, the step response current signal can be expressed as (4) if the inductance depends on the current.

$$i(t) = \left(i_0 - \frac{U_S}{R}\right)\cdot \exp\left(-\int_0^t \frac{R}{L(i(t))}\cdot dt\right) + \frac{U_S}{R} \tag{4}$$

If the time constant function in (4) is identified, the steady state current can be calculated. For identifying the time constant function we make use of the following relationship (5).

$$\left[\frac{\partial^2 i(t)}{\partial t^2}\cdot\left(\frac{\partial i(t)}{\partial t}\right)^{-1}\right]\cdot H(t)+\frac{\partial H(t)}{\partial t}=1 \quad\rightarrow\quad H(t)=-\frac{L(i(t))}{R} \tag{5}$$

The inductance is considered as a polynomial function of the current. With a given polynomial order, the polynomial coefficients can be computed from a system of linear equations (6).

$$\sum_{k=0}^{K} h_k\cdot\left(\left[\frac{\partial^2 i(t)}{\partial t^2}\cdot\left(\frac{\partial i(t)}{\partial t}\right)^{-1}\right]\cdot i(t)^k + k\cdot i(t)^{k-1}\cdot\frac{\partial i(t)}{\partial t}\right)=1 \quad\rightarrow\quad H(i)=-\frac{L(i)}{R}=\sum_{k=0}^{K} h_k\cdot i^k \tag{6}$$

The resistance estimation formula (inductance identification) is susceptible to noise. Therefore, the following modifiactions are introduced for better robustness:

- The method is generalized to PWM considering the average current per PWM cycle waveform. In fact, solenoids are usually driven by PWM and calculating the average current on a PWM cycle greatly reduces noise;
- For the sampled average current waveform, a smooth polynomial function is fitted and (6) is computed from the continuous polynomial function. Thus, truncation error is also reduced;
- A moving average of subsequent resistance estimates is taken.

The proposed method for estimating resistance was applied to a real solenoid actuator; the inductance function to be identified was arbitrarily chosen to be a first order polynomial. The corresponding transient waveforms and resistance estimates are presented in Fig. 2, a resistance was estimated from each transient sections. The resistance was also measured in a static "DC" measurement and the mean error of the estimation was found to be 0.23%.

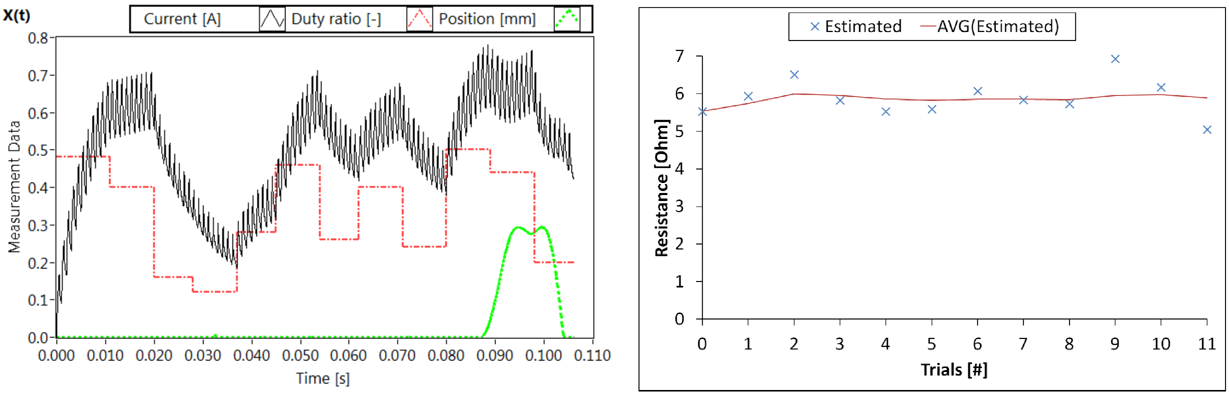


Figure 2: Experimental results of resistance estimation with a real valve actuator.

References

[1] Ivor Dülk, Tamás Kovácsházy, "Steady-state heat conduction in multilayer bodies: An analytical solution and simplification of the eigenvalue problem," *International Journal of Heat and Mass Transfer 67 (2013) 784–797*.

[2] Ivor Dülk, Tamás Kovácsházy, "A Method for Computing the Analytical Solution of the Heat Equation in Multi-Layered Media," *ASME Journal of Heat Transfer*, submitted (2013 august).

[3] F. de Monte, "Transverse eigenproblem of steady-state heat conduction for multi-dimensional two-layered slabs with automatic computation of eigenvalues," *Int. J. Heat Mass Transfer 47 (2004) 191–201*.

USING EXTERNAL VISUAL FEEDBACK TO IMPROVE P300 SPELLER PERFORMANCE

Ádám LÓSKA
Advisor: Béla PATAKI

I. Introduction

Brain-computer interface systems are getting more and more attention these days, mainly because of the improvements in performance and usability of these devices.

The most popular approaches for BCI systems are the sensorimotor rhythms (SMR [1]), the visually evoked potentials (VEP [2], SSVEP [3]), and the event related potentials (ERP, P300 [4]). In case of the VEPs and the ERPs, it is important to generate triggers (most often visual or auditory) for the user with reliable frequency and proper timing.

In case of the SSVEP based paradigms, visual stimuli are presented with a given frequency, and these frequencies are examined in the recorded EEG. In case of the popular P300 speller paradigm, letters are flashed at high speed, and a P300 artifact is looked for in the recorded EEG.

In this paper a P300 based paradigm, and the effect of jitter in display response time are studied. An external feedback system is implemented and examined, which uses a light sensor to detect the exact time when the stimuli are presented to the user.

II. Display delay jitter in P300

The P300 wave is an ERP component linked to decision making, specifically it reflects processes involved in stimulus evaluation and categorization. When recorded in EEG, it emerges as a positive deflection with a latency between the stimulus and the response of about 250 to 500 ms. [5] It is usually evoked using the oddball paradigm, in which low-probability target items are mixed with high-probability non-target (or “standard”) items. ERPs are typically studied by averaging multiple synchronized signals to enhance the quality and to increase the signal-to-noise ratio.

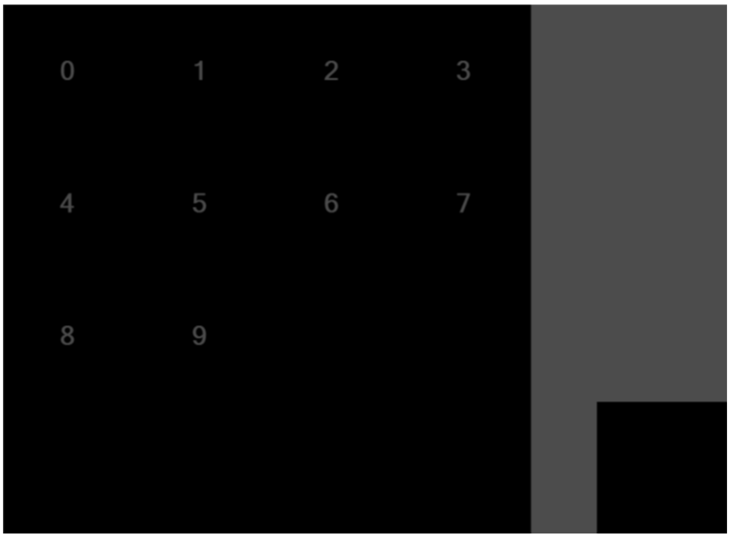


Figure 1: Layout of our P300 based application

Our P300 based application works as the following. The user can select from a set of icons – numbers from 0 to 9, arranged in a table layout (Figure 1). The icons flash in random order, each flash lasts 50 ms, and the inter-flash delay is 120 ms. The lower right area is used for synchronization as described later in the paper. The experiment is based on trials – one trial consisting of 10 flashes for each icon in random order, and a common target icon presented at the start of the trial. The EEG is recorded using an Emotiv EPOC Neuroheadset [6] using a band-pass filter of 0.1-30 Hz. The flashes are marked on the recorded EEG as triggers. The EEG is then broken down to epochs: time slices from 300 ms before to 800 ms after every trigger. We use a baseline-removal technique removing the mean of the pre-trigger signal, and epochs with values outside of a predetermined range are marked as noise and are discarded.

To measure the quality of the recorded signals, the target epochs (for triggers when the target icon was flashed) and non-target epochs have been averaged, thus further reducing the effect of noise, and extracting the P300 related time-synchronized artifacts.

The typical display delay jitter in a Windows Forms application has been measured, and it was revealed that the delay between when the instruction was executed from code and the actual display on screen can vary significantly.

Experiments were made using Windows Aero themes and Basic themes under light and heavy computer load on a typical general-use desktop PC. Table 1 shows that the average delay is significantly less using Basic themes, but the standard deviation in the measured data is actually greater.

Case	Mean (ms)	STD (ms)	5P (ms)	95P (ms)
Aero light load	60.4073	9.0871	44.6730	74.5829
Aero heavy load	63.1990	12.4302	45.1423	80.3857
Non-aero light load	23.3089	10.6208	7.7655	45.0570
Non-aero heavy load	27.2115	16.0421	6.1441	53.7185

Table 1: Display delay jitter

For the P300 paradigm, the delay of the visual display is not a concern, as it only effects the latency of the detected P300 component. However, the jitter in these latencies can cause the epochs to be averaged with a drift, therefore resulting in a worse quality ERP. Columns 5P and 95P shows the 5th and 95th percentile of the measured delays. Since the P300 component is usually described as a positive peak, averaging these epochs can significantly hinder the detection of P300 in the averaged ERP due to the hidden offset of more than 30 ms.

III. Improving synchronization

A solution to circumvent the display delay jitter was designed by using a light sensor to detect the exact moment the flash occurs on the display, which is the real target for the test person. With the implementation of our own P300 paradigm, we studied the results with and without external synchronization.

A Proposed architecture

The architecture is shown in Figure 2. The Paradigm Control maintains the flashing of the icons, and the light sensor is used to detect the actual display of the active icons. In standard architectures, the EEG is marked by triggers from the Paradigm Control: when the desired instruction is given to the display, the trigger is evoked. In the proposed solution, however, it only prepares the trigger, but waits until the light sensor detects the actual change in the display. To ensure that the light sensor does not disturb the user, a separate area is used on the lower right side of the display that is flashed together with every available icon. This way the jitter from the variable display delay is circumvented, and a more precise timing is achieved.

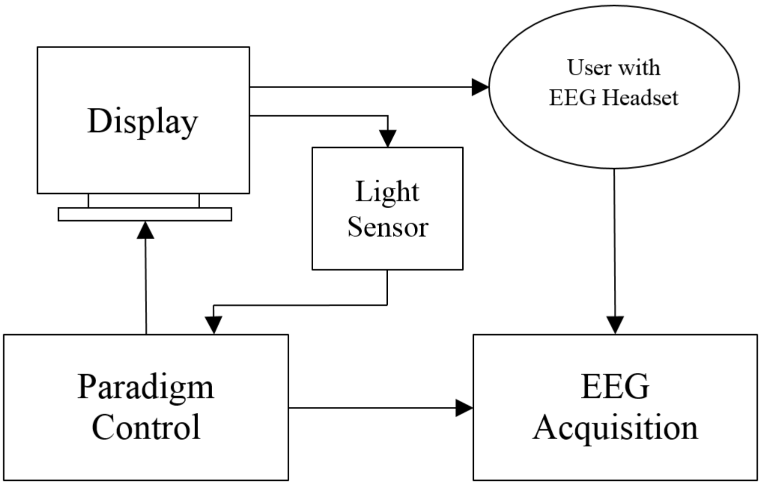


Figure 2: Improved architecture

B Comparison of the results

To compare the results of the synchronized and unsynchronized case several trials has been run for both methods. To keep the electrode placement intact, the headset was not moved between the experiments. The recorded signals were processed as described in Section II. The processing has been done for all 14 channels, but the results showed that a significant number of channels proved to be unusable in detecting P300 component.

We calculated the following signal features:

- Peak P300 amplitude of target ERPs (P3peak): the maximum voltage deflection in the range of 200-400 ms after the trigger.
- P300 latency on target ERPs (P3latency): the delay between the trigger and the peak of the P300
- P300 integral difference (P3int): the difference between the integral of the target and non-target ERPs in a 50 ms range calculated as the following:

$$\int_{P3start}^{P3end} tERP(t)dt - \int_{P3start}^{P3end} ntERP(t)dt$$

where

$$P3start = P3latency - 25ms$$

$$P3end = P3latency + 25ms$$

The greater the difference, the easier it is to discern target ERP from non-target.

Table 2 shows the comparison of the synchronized and unsynchronized case. In the last two columns (Synch P300 and Unsynch P300) one can see whether a clear P300 component can be detected visually in the channel. In the case of synchronized triggering, the Occipital and Posterior electrodes show clearer P300 components than with standard (unsynchronized) triggering.

In the 6 cases where both of the methods showed visible P300 component, 4 channels have greater P300 peak in the case of synchronized triggering. Even on channel F8, where the peak amplitude of the unsynchronized ERP is greater, it is greatly influenced by the error of the baseline removal as the mean of the pre-trigger ERP is significantly lower than the rest.

As shown in the table, the synchronized latencies are more stable between channels, while in the unsynchronized case significant difference can be seen between them.

Comparing the integrals of the P300 components (column P3int), there is no significant improvement on deterioration. Half of the comparable channels showed greater values with either the synchronized or unsynchronized methods, although the baseline removal error on Channel F8 affects this value as well.

Channel [7]	Target P3peak sync/unsynched (ms)	P3latency synch/unsynch (ms)	P3int synch/unsynch	Synch P300	Unsynch P300
AF3	3.6157 / 2.1192	309 / 332	20.1088 / 8.7226	+	+
F7	2.2157 / 2.0171	317 / 293	10.7289 / 11.7725	-	-
F3	3.4613 / 2.7682	317 / 340	21.1032 / 14.1902	+	+
FC5	1.2955 / 1.0557	325 / 200	3.1971 / 1.0187	+	-
T7	1.2793 / 1.8421	223 / 200	6.5533 / 2.6218	-	-
P7	0.8258 / 1.5861	215 / 200	5.6342 / 7.5153	-	-
O1	2.2391 / 1.6560	379 / 200	10.8502 / 6.9023	+	-
O2	3.8170 / 2.5854	379 / 200	19.2712 / 12.2197	+	-
P8	3.1285 / 1.5158	223 / 200	19.9971 / 3.3016	-	-
T8	1.4038 / 2.2632	293 / 301	-0.2907 / 8.0379	-	-
FC6	2.9215 / 3.2849	309 / 301	14.2979 / 16.1463	+	+
F4	4.2427 / 3.4907	309 / 309	26.7121 / 22.0978	+	+
F8	3.9668 / 5.6985	309 / 301	17.3450 / 32.7216	+	+
AF4	3.9850 / 3.7356	309 / 309	22.2211 / 22.8021	+	+

Table 2: Comparison of synchronized and unsynchronized ERPs

IV. Conclusion

As the experiments showed, circumventing the display delay trigger can improve some features of the recorded ERPs. The improvement, however, only occurs when the EEG is averaged across multiple epochs. The proposed method showed increased P300 peak amplitude, less variance in latencies of different channels, and no significant effect on calculated integral differences.

In future works, other methods should be studied, like synchronization to the display's vertical synch, to substitute for the unwieldy light sensor solution.

References

[1] A. Kübler and K. R. Müller, "An introduction to brain computer interfacing", Toward Brain-Computer Interfacing, pp. 1-25. MIT press, Cambridge, MA, 2007

[2] Bin, Guangyu, Xiaorong Gao, Yijun Wang, Bo Hong, and Shangkai Gao. "VEP-Based Brain-Computer Interfaces: Time, Frequency, and Code Modulations." *IEEE Computational Intelligence Magazine* 4, no. 4 (2009): 22–26.

[3] Diez, Pablo F., Vicente A. Mut, Enrique M. Avila Perona, and Eric Laciar Leber. "Asynchronous BCI Control Using High-Frequency SSVEP." *Journal of NeuroEngineering and Rehabilitation* 8, no. 1 (July 14, 2011): 39.

[4] Guan, Cuntai, M. Thulasidas, and Jiankang Wu. "High Performance P300 Speller for Brain-Computer Interface." In *2004 IEEE International Workshop on Biomedical Circuits and Systems*, S3/5/INV–S3/13–16, 2004.

[5] Duncan-Johnson, Connie C., and Emanuel Donchin. "The P300 Component of the Event-Related Brain Potential as an Index of Information Processing." *Biological Psychology* 14, no. 1–2 (February 1982): 1–52.

[6] Duvinage, Matthieu, Thierry Castermans, Thierry Dutoit, Mathieu Petieau, Thomas Hoellinger, Caty De Saedeleer, Karthik Seetharaman, and Guy Cheron. "A P300-Based Quantitative Comparison between the Emotiv Epoc Headset and a Medical EEG Device." ACTAPRESS, 2012.

[7] Homan, Richard W, John Herman, and Phillip Purdy. "Cerebral Location of International 10–20 System Electrode Placement." *Electroencephalography and Clinical Neurophysiology* 66, no. 4 (April 1987): 376–382.

NUMERICAL OPTIMIZATION OF MAXIMUM LIKELIHOOD PARAMETER ESTIMATION OF ADC PARAMETERS

ANNUAL RESEARCH REPORT[*]

Balázs RENCZES
Advisor: István KOLLÁR

I. Introduction

My work builds on results of the analog-digital converter tester group of the Department [1]. In this group the maximum likelihood (ML) method has been one of the main research topics for several years. It is shown to yield more precise results than the least squares (LS) method, which is commonly used in the sine wave test of A/D converters. As a result of this research, a MATLAB toolbox was developed which numerically optimizes the likelihood cost function [2], but its numerical methods were not yet thoroughly investigated.

II. The investigated issues

A Program speed

The first related problem was the relatively low speed of the toolbox: only 256 000 samples resulted in a 14 minutes iteration time, which is huge compared to the few seconds of the LS-estimator. However, a 16-bit ADC has about 64 000 bins, and in order to test all these we need at least four times more samples (about 256 000), hence the number of samples is usually large. In MATLAB,vectorizing the operations offered a major potential for optimization, since MATLAB caneffectively multiply a row and a column vector in a low-level (fast) routine, instead of using reliable, easy-to-understand but slow pairwise multiplications in a *for* cycle.

$$[a_1 a_2 \ \dots \ a_n] \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix} = \sum_{i=1}^{n} a_i \, b_i \tag{1}$$

After vectorizing these operations, the running time for 256000 samples decreased to about one and a half minute.

B Roundoff error

Increasing the speed allowed to increase the maximum iteration number of the algorithm. While testing with a sample set, an interesting phenomenon occurred: the logarithmic likelihood function became positive. By definition [3], the likelihood function is

$$L(p) = \prod_{i=1}^{M} P\big(Y(k) = y(k)\big), \tag{2}$$

where $P(Y(k) = n)$ is the probability, that the k^{th} value of Y has the digital code of *n*, *y(k)* is the k^{th} element of the (digital) sample set and *M* is the number of samples. The log-likelihood function is therefore

$$lnL(p) = \sum_{i=1}^{M} ln\big[P\big(Y(k) = y(k)\big)\big]. \tag{3}$$

[*] This paper is not an independent one, it is the summary of the PhD work of the last year presented in the minisymposium. It may contain parts of previously published documents or parts to be published later.

A probability is at most 1, the logarithm of which is 0. If all probabilities were equal to 1, the log-likelihood function would be 0. If only one of the probabilities is lower than 1, the log-likelihood function is negative. During the investigations it turned out that the strange value is caused by the cumulation of the roundoff errors, since during the summation the result is continuously growing, resulting that the next number to be added is relatively small compared to the sum. This causes high roundoff error values due to the floating point number representation.To illustrate the roundoff error we can calculate the following expression, having double precision floating point representation:

$$\big((10000+0{,}3)-10000\big)-0{,}3=-7{,}28\cdot 10^{-13} \qquad (4)$$

In the toolbox, the log-likelihood function was calculated as

$$lnL(p)=\sum_{i=1}^{M} ln\big[2\cdot P\big(Y(k)=y(k)\big)\big]-M\cdot \ln(2), \qquad (5)$$

which is mathematically the same as Eq. (3), and was thought to give the same result, as well. However, in this latter expression the absolute value of the sum grows much faster, if the probabilities are near to 1. In this case, instead of a small number, approximately $ln2=0.693$ is added to the sum in each summation step, resulting in a continuously growing sum and high roundoff error. When the log-likelihood function was evaluated according to Eq. (3), the log-likelihood function became non-negative.

The issue can be more generally solved, using the so-called Kahan summation method [4]. Let us calculate the sum $a+b+c$ so that $a>b>c$. The method calculates $a+b$, than first a, then b is subtracted. The result should be mathematically 0, but in reality the roundoff error is given, which the next term of the sum (c) can be compensated with, so the compensated term can be added to the sum. The pseudocode of the algorithm is the following:

```
sum = 0;                      %sum of elements of arg2
e = 0;                        %roundoff error
for i= 1 to M                 %Kahan compensation
     temp = sum;
     z = VectorToSum(i) + e;
     sum = temp + z;
     e = (temp-sum) + z;      %roundoff error
end
```

With the Kahan summation method practically the precision of the floating point representation is increased. On the other hand, the speed of the algorithm decreases this way.

C Result of different algorithms

The next question investigated was whether the optimum of the cost function has been reached indeed, when the program terminates. To investigate this, besides the Levenberg-Marquardt (LM) algorithm used in the toolbox, also the Differential Evolution (DE) genetic-type algorithm has been executed in order to be able to compare the results [5]. For a given sample set, the relative difference between the cost function values (for which we used the negative log-likelihood function) of the two algorithms was in the order of magnitude 10^{-13}, which seems to be negligible. On the other hand, this is much higher than the resolution of the number representation ($\approx 2\cdot 10^{-16}$), and the DE algorithm was closer to the optimum. The exact reason for this is yet to be investigated but we speculated that the gradient in the LM algorithm might be inaccurate due to roundoff errors, so we cannot further minimize the cost function with arbitrary accuracy. In Figures 1 and 2 the cost function has been investigated in the vicinity of the optimum of the two algorithms (a-e). Taking the two solution vectors, we selected one parameter, and keeping the other parameters constant, the values of the cost function (or rather, its deviation from the optimums) were plotted as a function of this.

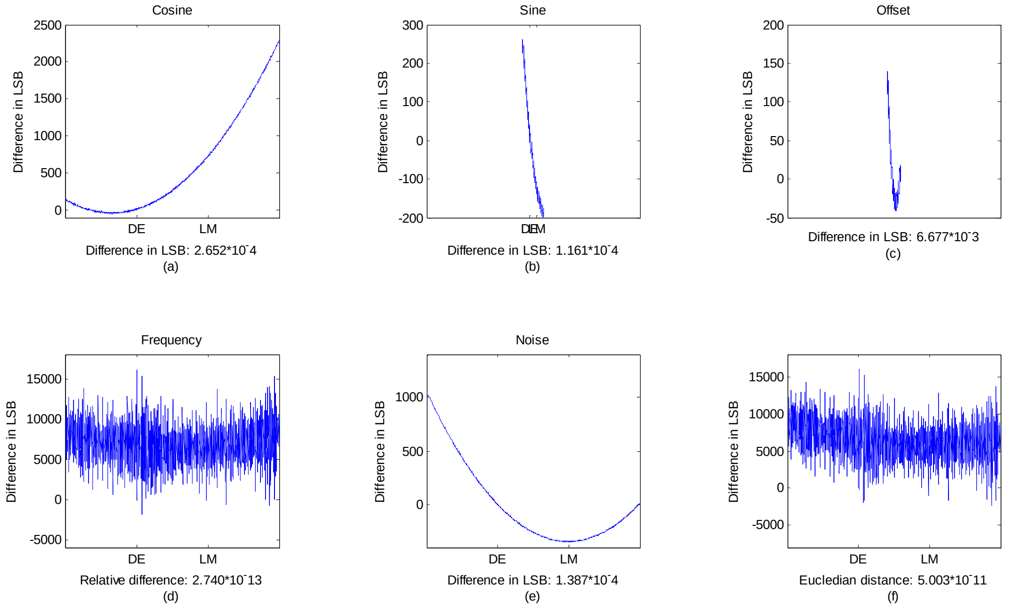


Figure 1: The value of the cost function in the vicinity of the DE optimum along the parameters, holding the others constant (a-e) and along the straight line between the two parameter vectors (f)

In addition, the cost function was evaluated along the straight line between the two resulting parameter vectors (f). The LSB on the horizontal axis is the LSB of the ADC, while on the vertical axis LSB is the resolution of the floating point number representation of the reference cost function value. Figures (a)-(c) were plotted so that the resolution of the horizontal axes are the same in order to be able to compare the results. In Fig. 1 (c) the difference is so small that it could not be plotted with this resolution.

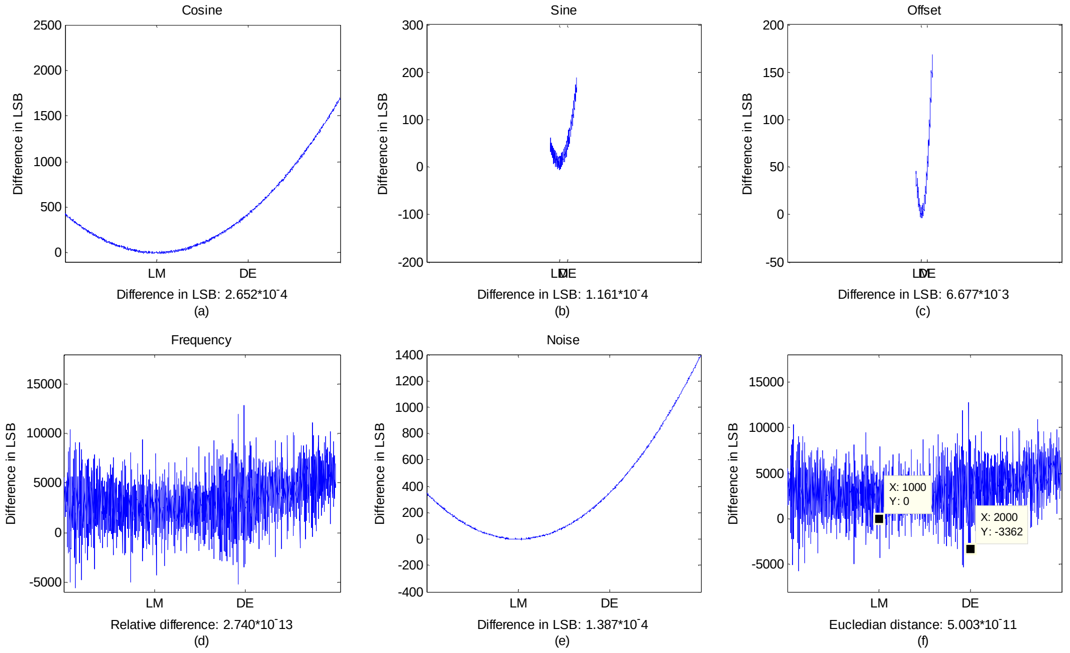


Figure 2: The value of the cost function in the vicinity of the LM optimum along the parameters, holding the others constant (a-e) and along the straight line between the two parameter vectors (f)

Figures 1 and 2 show that among the first three parameters the coefficient of the cosine has the most important effect on the difference. It can be clearly seen from the figures that although for more parameters (sine, noise) the LM-algorithm has lower cost function values, globally the DE-algorithm supplies better result, but it can also be observed that the value of the cost function is fairly “noisy” (we do not know yet if this is true unevenness, or just roundoff noise what we see).

D Precision of minimizing parameters

It should be noted that the algorithms might not have found the optimal solution. The differences between the algorithms may seem negligible, since 1000 LSB with $\approx 2 \cdot 10^{-16}$ resolution causes only about $2 \cdot 10^{-13}$ relative difference. However, it might be worthy to investigate even so small differences, considering the followings. For simplicity, let us try to minimize a second order cost function with one parameter [6]. In the vicinity of the optimum it can be calculated as

$$CF\left(x_{opt} + dx\right) = CF_{opt} + k \cdot dx^2. \tag{6}$$

Let us assume that we can determine the optimum with relative error *‘err’*. In this case

$$\frac{kdx^2}{CF_{opt}} = err. \tag{7}$$

$$\frac{dx}{x_{opt}} = \frac{1}{x_{opt}}\sqrt{\frac{CF_{opt}}{k}}\sqrt{err}. \tag{8}$$

This means that the uncertainty of the parameter is much higher than that of the cost function. For this reason, if the relative error in the cost function evaluation increases to the ten times of it, the uncertainty of the parameter increases to be the hundred times of the original value.

III. Conclusions

It has been shown that the maximum likelihood parameter estimation toolbox could be enhanced from several aspects. On the one hand, the running speed has been significantly increased. On the other hand, the evaluation of the cost function has also been improved. The numerical optimization has been executed with two different algorithms, yielding slightly different results. However, these results also showed that the evaluated cost function near the optimum values of the algorithms is fairly uneven, and therefore the minimum value cannot be determined with arbitrarily high accuracy. In other words, in a certain band the solutions are equivalent and this “additive noise” makes the only difference. To find the conditions to determine the bounds of these equivalent solutions is certainly not trivial and is a topic of further investigations.

References

[1] I. Kollár, T. Virosztek, V. Pálfi, B. Renczes, “*ADCTest Project*”
URL: http://www.mit.bme.hu/projects/adctest, Jan. 13, 2014.

[2] T. Virosztek, I. Kollár, “User-Friendly Matlab Tool for Easy ADC Testing”, 19th IMEKO TC4 Symposium, Barcelona, Spain, July 18-19, 2013, paper 133

[3] T. Virosztek, *“ADC testing in practice, using maximum likelihood estimation”*, TDK Report, 2014

[4] W. Kahan, *“Further remarks on reducing truncation errors”*, Communications of the ACM, Volume 8, p. 40, 1965

[5] K. Price, R. Storn, *“Differential Evolution (DE)”*
URL:http://www1.icsi.berkeley.edu/~storn/code.html, Jan. 13, 2014.

[6] B.Widrow, I. Kollár, *Quantization Noise: Roundoff Error in Digital Computation, Signal Processing, Control, and Communications,* Cambridge University Press, 2008.

ONE WAY NETWORK DELAY MEASUREMENT IN LOCAL AREA NETWORKS

Bálint FERENCZ
Advisor: Tamás KOVÁCSHÁZY

I. Introduction

Synchronizing the clocks of the nodes in a computer network is necessary to maintain the correct order of event processing or capturing an accurate timestamp of an event in an arbitrary timescale. This paper presents the foundations of message based network time synchronization in wired LAN (mainly Ethernet and IEEE 1588) environments. It also highlights the subtask of message propagation time estimation of the synchronization protocols and proposes alternatives to implement the measurement of these quantities.

A. *Terminology*

In a clock synchronization network the following metrics are used to characterize the performance of the network ([1] p. 40):

- Accuracy – the maximum of time differences between an arbitrary node and the reference clock.
- Precision – the smallest quantum of time effectively representable in the system.
- Jitter – the root mean square of differences in a series of a clock readout measurements.
- Wander – the root mean square of differences in a series of a clock frequency measurements.
- Delay – the elapsed time between the logical issuing and reception of a clock synchronization message.

The algorithm desribed in the IEEE 1588:2008 standard is commonly referred as the Precision Time Protocol (PTP), both nomenclatures are used in this paper. I refer to the widely available commercial equipments as COTS.

II. Overview of clock synchronization in LAN environment

Message based synchronization over a shared communication medium is a flexible way to implement clock discipline applications. These algorithms are based on the idea that the software can estimate the error of the clocks from observing the timestamps of the exchanged messages. In general the performance is primarily affected by the following factors:

- the quality of message (event) timestamps,
- the quality of message propagation time estimation,
- the quality of the clock discipline algorithms.

A. *Generation of message timestamps*

The calculation of ingress/egress event timestamps evolved from the software-only based estimation to the autonomous on-the-fly capture of the packets based on their signature on a broad range of platforms. The horizontal arrows in Figure 1 represent the most common timestamp capturing methods in the nodes.

The purely user-space software defined timestamp estimation method has the lowest precision and highest jitter, typically hundreds of microseconds in multiprogrammed environment. The method usually relies on the polling of a software counter which usually resides in the RAM. The readouts are encumbered by the jitter caused by the non–deterministic access time of the system memory, and the interruptions made by the task scheduler.

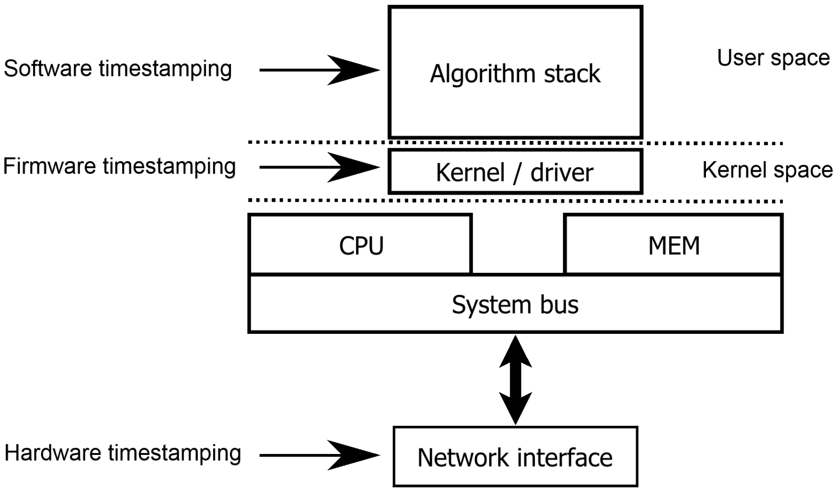


Figure 1: The available timestamping points in a node

The firmware based timestamp generation may reduce the average jitter and improve the precision typically in the order of tens of microseconds. The kernel mode code is scheduled with higher priority in Linux, and the available interrupt disabling opportunities might help to atomize the operation. Depending on the clock source, the non-deterministic memory or system bus access times lower the precision, and other non-compensated transmit/receive software paths introduce errors in the delay measurement.

The hardware based autonomous timestamp capturing is the only viable way to attain the necessary precision and jitter, up to tens of nanoseconds; its exact implementation exists on platforms such as FPGAs or ASICs (Ethernet MAC, PHY). Usually the implementation consists of the following elements:

- a numerically controlled oscillator (NCO),
- a clock register capture unit,
- and a packet filter (optional).

The NCO serves as the local clock of the board; the width of the NCO may be large enough to properly represent the used timescale, otherwise some kind of software translation is needed. The clock register capture unit executes the timestamping, which is controlled by the packet filter unit. If the packet filter functionality is absent, every sent and received packet is timestamped. However it may cause performance hit on the hosting system as the distribution of the ancillary information (the timestamps) to the user-space might be resource-intensive.

The high precision message timestamping has several robust implementations nowadays, there are plenty of high quality hardwares and IPs in the market.

B. *Message delay estimation*

The message propagation delay depends on the selected timestamping method. The exact quantity depends on the following components:

- the delay between the send state of the sender node and the timestamp generation
- the delay from the timestamp generation to the transmitter,
- the delay of the passive connections (wire delay),
- the delay of the active components (switching delay),
- the delay from the receiver to the timestamp generation,

- and the delay between the timestamp generation and application in the receiving node.

In a local area network several assumptions can be made about the components. The first and last elements are never exactly known in multiprogrammed systems, the delay introduced here causes dead-beat time in the servo. The hardware timestamping nodes effectively nullify the error caused by second and next-to-last components but in extremely accurate cases a hard-realtime (RTOS or FPGA) realisation must be used to provide the desired wander and jitter of the nodes. In general the wire jitter is negligible but in larger topologies hundred meters of cabling can cause microseconds of static delays; but this delay may be corrected with preentered correction factors on the nodes. The delay introduced by the active components (switches, routers) is responsible for the majority of the estimation errors as its jitter varies with respect to the current network parameters. The most common Ethernet switches use the store-and-forward principle, and the queueing model [2] may describe and forecast their behavior.

This subtask is in the focus of my future plans.

C. *Clock discipline algorithms*

The clock discipline algorithms are responsible for filtering of the incoming messages — based on their plausibility derived from runtime or apriori statistics — making statistics from the incoming messages for future use and applying the selected timestamps to the input of the clock servo. The clock servo might be implemented as a PI controller [3] but some in-research algorithms exist too [4]. The generated statistics are encumbered by the errors of the timestamping and one-way-delay measurement methods and by the method of implementation also (quantization or round-off errors).

The PTP algorithm and the most easily available implementations (linuxptp, ptpd) state several assumptions about the underlying carrier which may not true in multi-purpose network (such as the symmetry of the path). Figure 2 shows an implementation of the clock discipline task in a popular PTP implementation.

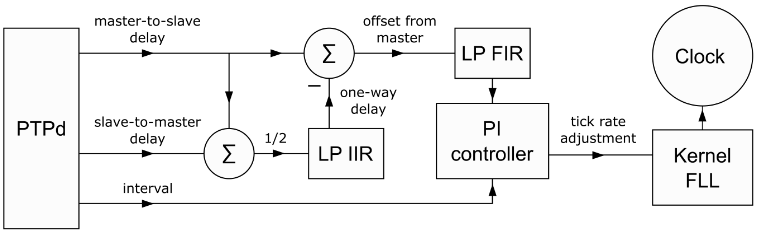


Figure 2: An example for a servo implementation [3]

III. Measurement and correction of message delay

Currently two main ways are available to measure the delay in the network, the estimation based only on protocol message exchanging and the estimation augmented with hardware (usage of transparent clocks) [5]. Transparent clocks can measure the residence time for selected packets between their ingress and egress ports and on-the-fly update them with that information. With the usage of this component the estimation error may be minimized but in legacy networks the substitution of the existing switches with transparent clocks are limited by financial and other constraints.

We propose to augment the delay estimation with software statistics from other sources, such as the traffic data of the switches. To identify the effects of other parameters of the network on the message propagation, a characterization measurement setup must be constructed to evaluate the effects. The characterization may reveal some quantitative information of the asymmetries in a real-world network.

A. *Measurement setup*

The measurement setup consists of the following components:

- A switch hardware which can measure the residence time of the packets,
- and a statistics aggregator which extracts the residence times from the switches and converts them to some standardized format which could be processed later on.

The choice of the switch hardware depends on the requirements of the medium and management capabilities. The modification of a simple COTS switch does not meet the timing criteria as the usage of software timestamps is unwelcome in the characterization process. The usage of a COTS transparent clock in the initial measurements may show the potential of the idea (see Fig. 3). The data collection in this phase relies on the residence time property of the PTP packets which is filled in by the transparent clock. If the available transparent clocks are not equipped with the necessary interfaces or their manageability is limited, the construction of a custom measurement setup is needed. The FPGA based solution is a viable way [2] of the implementation, but we chose a Linux and PTP capable NIC based setup due to its rapid development capability. The proposed transparent clock might be a four port Intel i350 adapter with its internal clocks synced together. With this setup the measurement of more complex switching rules will be attainable.

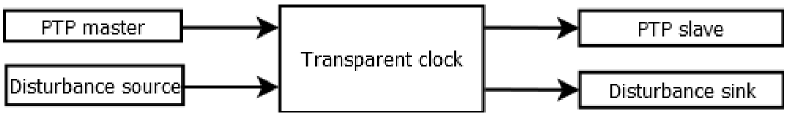


Figure 3: Measurement setup with a Transparent Clock

IV. Conclusion and future work

The paper introduced the basics of the message based computer network time synchronization. It showed which of the subtasks are in the scope of ongoing research and selected the delay measurement task for in-depth examination. The paper also showed some of the latest research findings of others which might be the base of my future work. The future work consists of the following tasks:

- the construction of the proposed measurement system,
- the collection of measurement data in various network environments,
- the integration of the findings with well-known clock discipline strategies as well as the in-research ones.

References

[1] D. L. Mills, *Computer Network Time Synchronization: The Network Time Protocol on Earth and in Space, Second Edition*, CRC Press, 2010.

[2] D. Jin, D. Nicol, and M. Caesar, “Efficient gigabit ethernet switch models for large-scale simulation,” in *Principles of Advanced and Distributed Simulation (PADS), 2010 IEEE Workshop on*, pp. 1–10, 2010.

[3] K. Correll, N. Barendt, and M. Branicky, “Design considerations for software only implementations of the IEEE 1588 precision time protocol,” in *Conference on IEEE*, vol. 1588, pp. 11–15, 2005.

[4] G. Giorgi and C. Narduzzi, “A resilient Kalman filter based servo clock,” in *Precision Clock Synchronization for Measurement Control and Communication (ISPCS), 2013 International IEEE Symposium on*, pp. 59–64. IEEE, 2013.

[5] T. C. on Sensor Technology (TC-9), *IEEE Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems*, IEEE Instrumentation and Measurement Society, 2008.

EXAMINATION OF ALGORITHMS USING DYNAMIC DATA FLOW GRAPHS

Gábor WACHA
Advisor: Béla FEHÉR

I. Introduction

In some of the algorithmically intensive embedded real time streaming applications, a single CPU is not sufficient to handle the task in real time, but the processing power of more processor cores would be theoretically enough. Algorithms are usually implemented in C/C++ languages which natively do not support parallel implementation and algorithm slicing [1] into multi-core processor systems.

Many of the embedded processing platforms – including FPGA based soft core processor systems, or Cell architecture – do not offer cache coherency among the processors. Standard profiling results – showing runtime information only – are sufficient to slice the algorithm into multiple CPU nodes, thus creating a balanced load distribution, but internal communication bandwidth often remains a bottleneck [2].

This paper relies on the visualization and generation of Aggregated Dynamic Data Flow Graphs (ADDFG) introduced in [3]. This ADDFG can help to visualize these critical data paths of a given software. Also, with the information of the critical data paths, an estimation of a possible slicing of the algorithm can be given. Finding the correct slices of an algorithm can achieve greater performance on multicore systems. Once the slices are found, a multithread program can be generated from a given task graph [4]. Silva, R.E. et al. use Dynamic Data Flow Graphs to measure the network contention of MPI-based applications in [5]. Their methods can be used to measure the performance of an automatically sliced algorithm.

This paper describes a basic communication model between specific functions of a given software. With this model, a clustering of the data flow graph can help to find the data-independent functions of an algorithm. The data-independent functions can be executed as different software threads, thus minimizing the necessary synchronization. An algorithm sliced in this way can perform better on multicore environments.

II. Introduction to program slicing based on run time analysis

As Weiser, M. introduces in [1], program slicing is a method for automatically decomposing programs to slices by analyzing their control and data flow. Each slice does a specific subset of the algorithm's behavior.

Weiser's approach is static in the form that it uses only compile-time analysis to slice algorithms. With this method, all possible data paths can be explored. Unfortunately there is no information about the run-time "importance" of the path. It is possible that the software has a data path which is rarely used for communication during the execution. The program will not be sliced through this data path – altough the path is almost never used.

Fig. 1. shows such a case: communication channels (for example memory or register reads and writes) between nodes `D` and `Q`, `O` and `P` are rarely used in most of the time. A compiler can not slice the algorithm into independent parts (for example `O`, `A`, `B`, `C`, `D` and `P`, `Q`, `R`, `S`) without run-time knowledge.

A run-time generated dynamic data flow graph can help to find these data paths which then can be used for more advanced slicing. Also, a higher level approach is preferred: instead of an instruction or

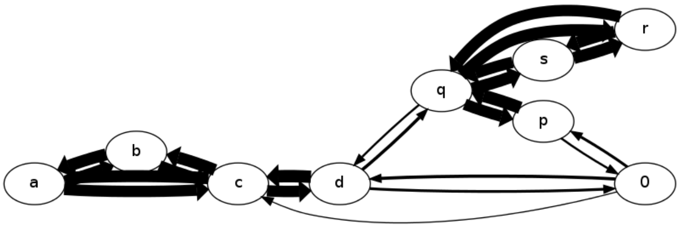


Figure 1: Dynamic data flow graph of an algorithm which can not be sliced using compile-time methods

statement level data flow graph a function level aggregated data flow graph (ADDFG) is used.

As introduced in [3], the ADDFG is a graph generated from run time data flow information. Each vertex of the graph is equivalent to a software function in the algorithm under test. An edge between two vertices means that communication happens between two specific functions during the execution of the program. This communication can be done through a memory or register access. The weight of an edge shows the amount of data transferred between the two functions during the execution.

III. Communication model

To enable the analysis of the algorithm under test a simplified communication model between functions is introduced.

In this basic model the following assumptions are made about the data transfer between functions:

- Control flow (function call, call graph) information separate from the ADDFG can be discarded. Since every function call encapsulates a data transfer, the ADDFG contains the call information. For example: the function arguments are written in the caller function and are read in the callee. A more frequent function call has an edge with greater weight in the ADDFG.
- The direction of the data transfer can be ignored, because the performance hit on each possible communication between multiple threads will usually be the same regardless the direction of the transfer.
- The communication rate is constant. Every function transfers data continuously at an average rate. This means that the actual timing of the data transfer is ignored.
- The slicing algorithm optimizes only for inter-core data transfer, not for a balanced processor load. For each function the execution time equals, the processor requirements of the algorithm are ignored.

The first two assumptions are correct to the effect that a real-world software has these characteristics. The consequence of the third simplification is that the slicing algorithm can not find slices which could be pipelined in the optimized program. An other simplifying consequence is that the weight of an edge in the ADDFG can be treated as the data communication rate between two functions.

The fourth simplification has problems. With this assumption the balanced load distribution on each processor core can not be guaranteed. The ADDFG does not hold the necessary information to enable a more balanced load.

IV. Program slicing based on spectral clustering

With the communication model introduced in Section III., a spectral clustering method on the adjacency matrix of the ADDFG can find a data-independent slicing of the algorithm.

The intent of the clustering is to group the vertices of the ADDFG so that the sum of the weight of the edges in the cut $W(A_i, \overline{A_i})$ between the partitions A_i and $\overline{A_i}$ are minimal. Because the edge weights represent the data communication rate between different functions, finding a minimal cut between these partitions of the ADDFG can be used to slice the program.

To avoid having separate vertices as a partition, each partition is weighted with the number of vertices inside the partition. Such a clustering, which satisfies the criteria in Equation 1., is called the minimum ratio cut [6]. Finding the minimum ratio cut is an NP-hard problem, the following algorithm is an approximation [7].

$$\min \frac{1}{2} \sum \frac{W(A_i, \overline{A_i})}{\|A_i\|} \tag{1}$$

The clustering algorithm is the following [8]. Let G be the weighted adjacency matrix of the ADDFG. Since the direction of the data transfer can be ignored, the adjacency matrix of the undirected ADDFG is $A = G + G^T$. A is a symmetric matrix, on which the spectral clustering method [8] can be used. Let L be the unnormalized Laplacian matrix of A [9]:

$$l_{ij} = \begin{cases} deg(i) & \text{if } i = j.\ deg(i) \text{ is the degree of the } i\text{-th vertex,} \\ w(i,j) & \text{if } i \neq j \text{ and there is an edge between } i \text{ and } j.\ w(i,j) \text{ is the weight of that edge,} \\ 0 & \text{otherwise.} \end{cases}$$

The unnormalized Laplacian matrix can be calculated as $D - A$, where D is the diagonal degree matrix of the weighted symmetric adjacency matrix A.

The eigenvector corresponding to the smallest eigenvalue of L has the coordinates of 1, the eigenvalue is 0. All the other eigenvalues are real [9].

The k-means clustering of the eigenvector corresponding to the second smallest eigenvalue of L can be used as an approximation for the clusterization of A [7]. Comparing the coordinates (which are real numbers) of the eigenvector to a given threshold value can give two clusters of the corresponding graph.

For more accurate results, a recursive approach on the Laplacian of the clusters, or a k-means clustering on all of the eigenvectors of L can be used [8].

V. Results

The clustering method was tested on the ADDFG shown on the left side of Fig. 3. The unclustered and the clustered adjacency matrix of the ADDFG is shown in Fig. 2. The adjacency matrix has two clusters, the algorithm will be sliced to run on two CPU cores. The test software was written so that the assumptions made in Section III. were valid. Each of the functions of the algorithm have about the same processor load.

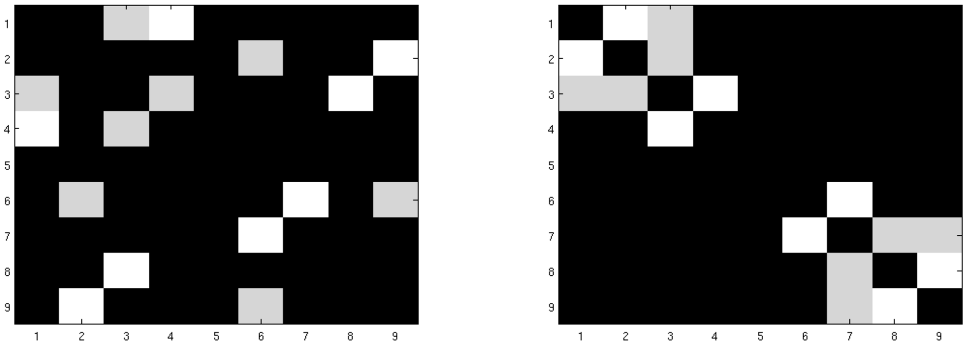


Figure 2: Adjacency matrix of the unclustered and the clustered data flow graph

To examine the performance of the sliced program, the execution time of the sliced and unsliced software was measured.

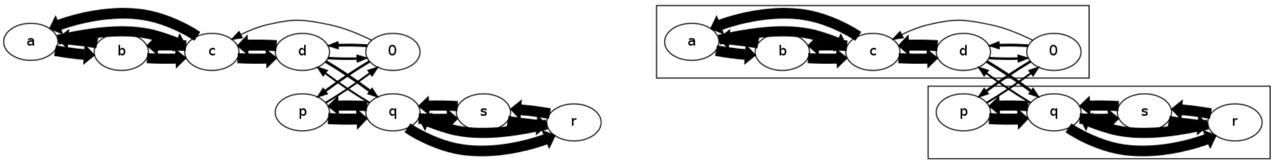


Figure 3: The slices of the program found by clustering the ADDFG

Compared to the single threaded algorithm, the execution time was 47% less when the program was sliced into two threads on a multicore CPU. To slice the algorithm, the slicing shown in Fig. 3 was used.

This measurement only shows that with the found slicing the performance of the software increases, it does not show that the slicing is optimal. A comparison with static slicing tools (for example CodeSurfer or Frama-C) should also be done to verify that a dynamic method can achieve better results.

VI. Further work

In Section III., four assumptions were made for a simpler inter-subroutine communication model. One of the assumptions was that the exact timing of the data transfer is ignored. With this simplification, pipeline structures can not be created. A future version of the ADDFG generation can include timing information in the edge weights, thus enabling the creation of pipelines.

As mentioned, the slicing algorithm does not take the computational requirements into regard, thus making load balancing not possible to attain. With the help of program profiling information this deficiency should be solved.

Usually the actual structure of the ADDFG depends from the input parameters of the algorithm. The research so far does not examine the input dependency, this should be addressed.

References

[1] M. Weiser, “Program slicing,” in *Proceedings of the 5th International Conference on Software Engineering*, ICSE ’81, pp. 439–449, Piscataway, NJ, USA, 1981. IEEE Press.

[2] M. A. Kim and S. A. Edwards, “Computation vs. memory systems: Pinning down accelerator bottlenecks.,” in *ISCA Workshops*, A. L. Varbanescu, A. M. Molnos, and R. van Nieuwpoort, Eds., vol. 6161 of *Lecture Notes in Computer Science*, pp. 86–98. Springer, 2010.

[3] I. Szabó, G. Wacha, and J. Lazányi, “Aggregated dynamic dataflow graph generation and visualization,” *Carpathian Journal of Electronic and Computer Engineering*, 6(2):50–54, 2013.

[4] E. Jeannot, “Automatic multithreaded parallel program generation for message passing multiprocessors using parameterized task graphs,” 2001.

[5] R. Silva, G. Pezzi, N. Maillard, and T. Diverio, “Automatic data-flow graph generation of MPI programs,” in *Computer Architecture and High Performance Computing, 2005. SBAC-PAD 2005. 17th International Symposium on*, pp. 93–100, 2005.

[6] L. W. Hagen and A. B. Kahng, “New spectral methods for ratio cut partitioning and clustering.,” *IEEE Trans. on CAD of Integrated Circuits and Systems*, 11(9):1074–1085, 1992.

[7] U. Luxburg, “A tutorial on spectral clustering,” *Statistics and Computing*, 17(4):395–416, Dec. 2007.

[8] A. Y. Ng, M. I. Jordan, and Y. Weiss, “On spectral clustering: Analysis and an algorithm,” in *ADVANCES IN NEURAL INFORMATION PROCESSING SYSTEMS*, pp. 849–856. MIT Press, 2001.

[9] B. Mohar, “The Laplacian spectrum of graphs,” in *Graph Theory, Combinatorics, and Applications*, pp. 871–898. Wiley, 1991.

ACCURATE INDOOR ULTRASONIC POSITION TRACKING

Péter GYÖRKE
Advisor: Béla PATAKI

I. Introduction

In this paper a three dimensional ultrasonic position tracking system will be presented. The position calculation is based on distance measurements from a moving transmitter to multiple fixed receivers.

Indoor position estimation or tracking can be applied in several domains e.g.: mobile robot navigation, robotic arm identification, movement observation of a person [1], automated spotlight control on stages, etc. Position determination is usually based on distance measurements from special points to the target, e.g. in GPS the satellites are transmitters with known position and the receiver's position is to be determined. The distance is calculated with the Time of Flight (ToF) method, which means that the time is measured between transmission and reception of a message. In the case of GPS the message speed is the speed of light. If shorter distances are to be measured, this high speed needs extremely fast electronics if radio waves are used. In short distance applications, like indoor positioning, sound waves are also acceptable for time of flight measurement. The slower speed of the sound makes it possible to measure ToF with simple hardware (e.g. standard microcontrollers).

Besides ToF method there are other ways to measure distance. Alternatively, a phase-shift method [2,3] is used to estimate the distance between the transmitter and the receiver by measuring the phase difference between the transmitted and received signals. The phase-shift method is typically more accurate than the ToF method. However, with the phase-shift method, the maximum range that can be estimated is limited to one wavelength of the transmitted signal. Usually laser range finders use this method. If the modulation of the output signal can be varied, the range can be extended at the cost of the decreased precision.

The third and least accurate method to measure distance is to measure the received signal strength (RSSI)[4]. In general, radio waves are used for this method, but using sound waves are also possible. Since the radio signal strength is a nonlinear function of distance and disturbing objects can be between the transmitter and receiver, exact calculation is not possible. To deal with the problem interpolation between calibrated points is used. The main advantage of this method is that radio waves penetrates walls, thus more rooms can be covered with less radio transmitter (or receiver). For example rough positioning is feasible by measuring the signal strengths of WiFi access points in the building [5].

Considering the previously presented methods a ToF method was chosen which uses ultrasonic waves. The advantages of this approach are the following: **1)** accuracy is satisfying (a 1-3 centimeters error over a 0.1-10 meter range), **2)** a relatively simple hardware is capable for the measurement. Of course there are disadvantages as well: the visibility of the transmitter by the receiver must be provided.

In the following section the hardware considerations will be presented, after that in the third section the position calculation and filtering details will be shown.

II. Hardware structure

Time of flight distance measurement with ultrasonic waves can be performed by ultrasonic transmitters or receivers (or transducers if the device is applicable in both ways). If the transmitter and receiver is in one module the distance measurement can only performed in a reflective way. This method measures the time difference between the sent and received signal, so the distance of a reflective surface could be measured (see Fig. 1 section "A"). The next method (see Fig. 1 section

“B”) use separate transmitter and receiver, but a synchronization signal is needed to calculate the ToF. The synchronization signal can be a wired link between the two stations, although wireless links are also possible if the communication delay is known and jitter is not significant compared to the ToF. In the third method the transmitter is not synchronized with the receivers, so absolute distance cannot be measured, however from time difference between the signal incoming to the two (or more) receivers, a distance difference can be calculated (see Fig. 1 section “C”). This last method is promising for our purposes, with the appropriate number of fixed and known positioned receiver modules, the determination of the transmitter’s position is possible. If there are no synchronization signals between the transmitter and the receivers, only one transmitter can operate at a time, because the ultrasonic signal does not contain any information about the sender.

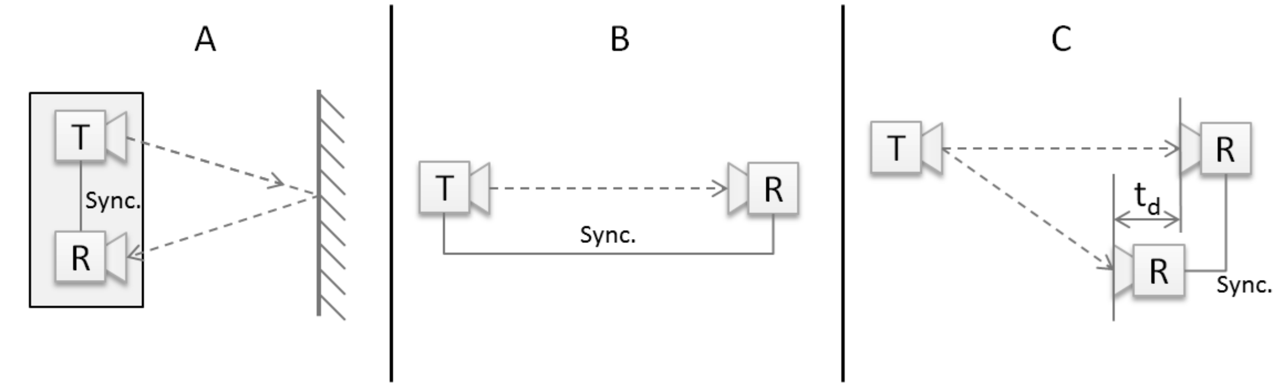


Figure 1: Possible ways to measure distance with ultrasonic waves. “T” means transmitter, “R” means receiver. Dashed lines are ultrasonic signals, continuous lines show (wired) synchronizations.

The ultrasonic signal in question needs further explanation. For generating and detecting ultrasonic signals, piezoelectric crystal based transducers are used. The piezoelectric crystal deforms when a voltage is applied, and the detection is also possible due to the reverse effect: voltage is generated when the crystal is mechanically deformed. The manufactured ultrasonic transducers contain a tuned crystal, shaped for a specific frequency, the typical value is 40 kHz. This is the resonant frequency of the piezoelectric crystal.

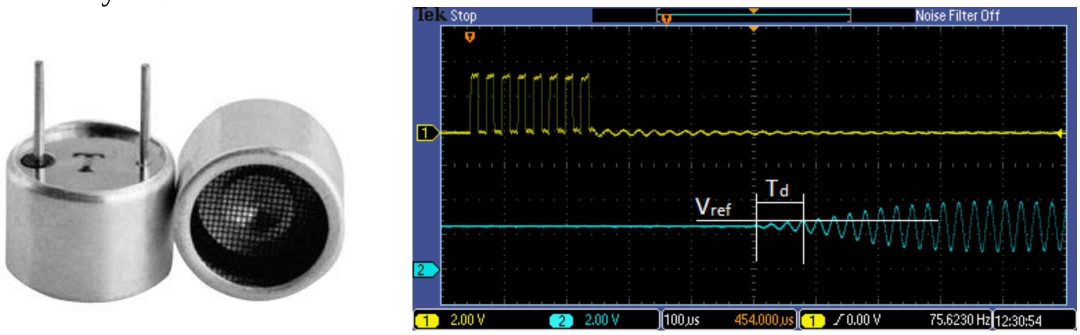


Figure 2: A typical open form piezoelectric ultrasonic transducer on the left; the waveform of an excitation signal and the corresponding received signal. The time delay (T_d) between true reception of the signal and the detection can cause errors.

The transmitter is driven with a few periods (e.g. 8 periods) of a square signal (see Fig. 2), the frequency is 40 kHz. This small burst generates a short ultrasonic “beep”. The time between the bursts depends on the distance range to be measured, in closed spaces the interference of reverberated waves should be also taken into consideration. If the bursts are controlled such a way that they are far enough from each other in time, the reverberated waves can eliminated from the measurement (on usual surfaces in a room the ultrasonic wave is suppressed after 1-2 reflections).

The detection of an incoming burst needs more components than the generation. The voltage generated by the ultrasonic receiver is a few millivolts, therefore significant amplification of the signal is required. An instrumentation amplifier circuit was chosen to amplify the voltage between the two poles of the receiver crystal. The gain is approximately 1000. For the measurement of the ToF only the start time of the incoming burst should be measured, thus the amplified signal is compared to a reference voltage. The receiver modules are in a wired RS-485 network. The master of the network sends a synchronization message in every measurement cycle (see Fig. 3). The comparator's output triggers a timer, and the time elapsed from the synchronization message is stored. When all the receivers stored the time or set a flag when no incoming burst was detected, the master starts to request the measured times from the receivers.

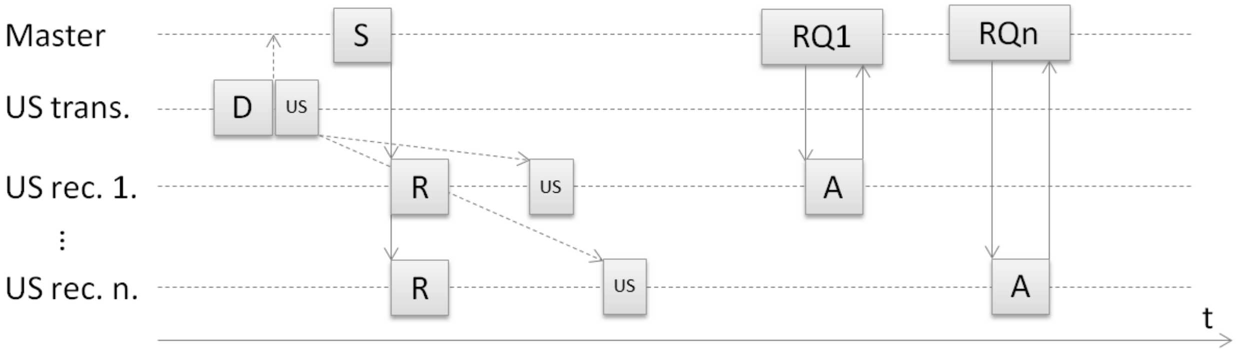


Figure 3: The timing diagram of a positioning cycle. Abbreviations: "D": data transmission, "US": transmission or reception of an ultrasonic burst, "S" synchronization message, "R": resetting timers, "RQn": requesting time data from device n, "A" answering request.

An additional synchronization between the US transmitter and the master unit (D in Fig. 3.) enables the system to measure absolute distances to the receivers. This synchronization signal is transmitted via radio frequency, the advantages of this: 1) the position can be determined from fewer sensors due to absolute distance measurement; 2) additional information can be added (e.g. transmitter ID), so the system will be capable of handling multiple transmitters with time multiplexing.

III. Position calculation

In order to determine the position (coordinates) of the unknown point several approaches are possible. If only distance differences are known it is called multilateration, in this case 4+1 receivers are necessary for three dimensional positioning. If absolute distances are known, 4 receivers are enough (trilateration) [4]. The following problem is given: there are a number of fixed three dimensional points (receivers):

$$R_1,\ R_2, \dots, R_n \in \mathbb{R}^3$$

and one point with unknown coordinates:

$$U \in \mathbb{R}^3$$

and the distances:

$$d_1,\ d_2, \dots, d_n \in\ \mathbb{R}$$

between the R_i and U are known as $\overrightarrow{UR_\iota}$ vectors. The following equation system should be considered:

$$\sqrt{(U_x - R_{ix})^2 + (U_y - R_{iy})^2 + (U_z - R_{iz})^2} = d_i \text{ where } i = 1, 2, \dots, n \qquad (1)$$

As it is expected this is a three dimensional second order equation system describing spheres. Consider the following: one receiver defines a sphere where U is a point on the surface of the sphere. Two receivers define two spheres where U is on the intersection circle of the spheres (if the spheres intersect each other). If a third sphere is defined by a third receiver and intersects the circle, U can be only in 2 points (see Fig. 4). The fourth receiver can define that which point is U. If additional information is given (e.g.: a range where U is possible) U can determined with 3 receivers. If n = 3 the equation (1) forms a second order equation system. there can be three kinds of result: 2 different solutions, 2 identical solutions or no solution (if at least two of the spheres do not intersect each other).

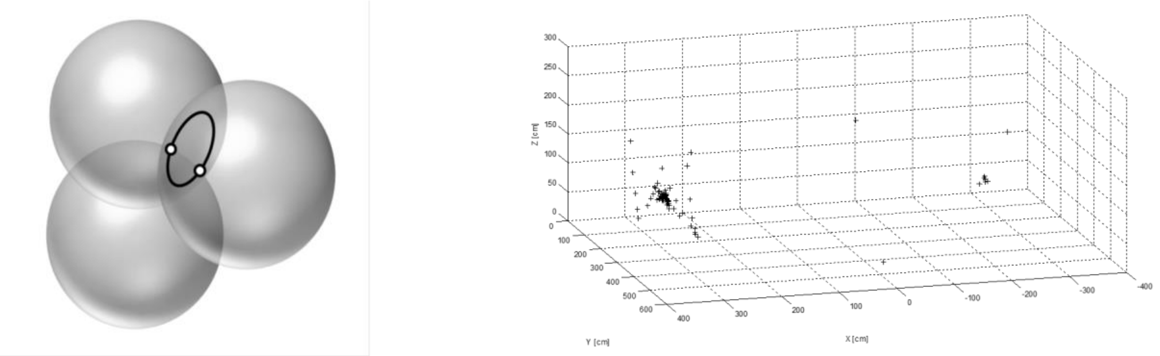


Figure 4: The illustration of the three spheres' intersection on the left, 3D plot of the possible position candidates on the right.

To conclude the reasoning above, the position can be determined from 3 receivers: if the receivers are placed on one plane (e.g. each one in each corner of the room and in the same height) one of the solutions (if exist) will not be in the room, so it can be discarded. Thus 3 receivers are enough for positioning, nevertheless the accuracy can be improved by using more receivers. The following filtering method is used: all possible three receiver subsets are selected from the set of the receivers; then $\widehat{U_k}$ is calculated for every subset. These estimations have to be aggregated; a final position candidate has to be selected. Simple averaging over the points will distort the result because of the outlier points. The task is similar to the center point selection in clustering problems. In Fig. 4 the estimated points are illustrated, there are clearly some outlier points. The rest of the points are grouped in a sphere which center point is to be determined. The approach is the following: a threshold is specified (e.g. 10 cm), and the number of neighbor points within the threshold distance is calculated for every point. The point with the maximal neighbor count is selected, than those points are averaged which are within the threshold range of this point.

IV. Conclusion

In this paper a method of ultrasonic ranging was presented. Accurate positioning has been achieved by using a filtering method specialized for this type of data. The first tests of the system have shown that ±5 cm accuracy in all the 3 dimensions could be achieved over a 0.1-5 meter range with 6 receivers in the room.

References

[1] Nonaka, H., & Da-te, T. "Ultrasonic position measurement and its applications to human interface", *IEEE Trans. on Instrumentation and Measurement, 44*(3), 771-774. 1995.

[2] Saad, M. M., Bleakley, C. J., & Dobson, S. "Robust high-accuracy ultrasonic range measurement system", *IEEE Trans. on Instrumentation and Measurement, , 60*(10), 3334-3341. 2011.

[3] Huang, Y. P., Wang, J. S., Huang, K. N., Ho, C. T., Huang, J. D., & Young, M. S. "Envelope pulsed ultrasonic distance measurement system based upon amplitude modulation and phase modulation", *Review of scientific instruments*, *78*(6), 065103-065103. 2007.

[4] Awad, A., Frunzke, T., & Dressler, F. *Adaptive distance estimation and localization in WSN using RSSI measures*, 10th Euromicro Conference on Digital System Design Architectures (pp. 471-478). IEEE. Aug. 2007.

[5] Ching, W., Teh, R. J., Li, B., & Rizos, C. *Uniwide WiFi based positioning system*, IEEE International Symposium on Technology and Society (ISTAS), (pp. 180-189). IEEE. June, 2010.