Conference proceedings
Sborník odborného semináře

# Nové metody a postupy v oblasti přístrojové techniky, automatického řízení a informatiky 2018
# New Methods and Practices in the Instrumentation, Automatic Control and Informatics 2018

28. 5. – 30. 5. 2018, Příbram - Podlesí

ISBN 978-80-01-06477-1

# OBSAH



Editor: Ing. Vladimír Hlaváč, Ph.D.
Název díla: Nové metody a postupy v oblasti přístrojové techniky, automatického řízení a informatiky 2018
Vydalo: České vysoké učení technické v Praze
Zpracoval: Ústav přístrojové a řídicí techniky Fakulty strojní ČVUT v Praze
Kontaktní adresa: Ústav přístrojové a řídicí techniky, FS ČVUT v Praze, Technická 4, Praha 6
Tel.: +420 2 2435 9750
Počet stran: 98 Pořadí vydání: 1
Vydáno dne 31. 7. 2018 jako pdf soubor na CD.
Dostupné na webových stránkách Ústavu přístrojové a řídicí techniky:
http://control.fs.cvut.cz/nmp
ISBN: 978-80-01-06477-1.

# PRODUCTION OF POLYMER MICROLENSES BY DROP DEPOSITION

*J.Pertuiset[1], C.Chanut[2] , J.Hosek[3]*

*[1] CTU Faculty of Mechanical Engineering, julien.pertuiset@insa-lyon.fr*
*[2] CTU Faculty of Mechanical Engineering, cecile.chanut@insa-lyon.fr*
*[2] CTU Faculty of Mechanical Engineering, jan.hosek@fs.cvut.cz*

*Abstract:*

A microlens is a small lens, generally with a diameter less than a millimetre and which may be as small as 10 micrometres. Microlenses are key components for optical devices. Thus they are widely applied in several application fields such as communications (optical fibers), 3D displays, optical data storage and photodetectors. Numerous classes of microlenses exist, depending on the technology and the specific applications. Today, polymer microlenses have been developed because of their low cost and good properties. Also, a large variety of fabrication processes have been developed for plastic/polymer-based microlenses, such as embossing, soft lithography, micromolding , photolithography, electron beam lithography, reactive ion etching, the laser assisted technique, and printing techniques. In this article, we will introduce drop deposition of polymer, a way we used to develop microlenses in laboratory.

*Keywords:* microlenses, polymer, drop deposition, curing, optical properties.

## 1 Introduction

Through our study, our goal was to produce very small polymer lenses by drop deposition on a substrate. To perform that, we have chosen a polymer with convenient optical properties. Then, we have used a precision dispenser for the deposition. The drops were then cured to make solid polymer lenses. The curing can be made either by heating or with UV, depending on the polymer type. The shape of the drop is affected by different parameters: the volume deposited (controlled with the dispenser), the surface tension, the contact angle between the polymer and the substrate and the curing if a shrinkage occurs. The deposition parameters, the polymer and the substrate used, or even the curing are then the parameters that we can play on to obtain microlenses in a reproducible way, with given dimensions and a given geometry (and thus a given focal distance) [1].

## 2 Experimental Setup

### 2.1 Setup

Our setup is divided in three different parts: the Ultimus II [2] is the dispensing system (from Nordson EFD), the syringe with the chosen tip and finally the air compressor.


Fig.1. Experiment setup for the drop deposition

On the picture just above we can distinguish:

- Ultimus II features 0-15 psi (0-1 bar) constant-bleed air pressure regulation and provides greater control when dispensing any type of fluid. We can play on three different parameters to change the diameter of the drop: the time of deposition (controlled by a foot pedal), the air pressure regulation and the vacuum.

- The syringe barrel, capacity of the syringe 3cc. It means it can contain 3 cubic centimeters (cc) of a liquid. The brand is also Nordson EFD. The syringe is molded from a polymer that gives chemical compatibility. A piston is used to provide consistent fluid deposit by prevending air entrapment. We put precision tips at the extremity of the syringe. There are 2 different diameters (0,20 mm and 0,15 mm) for the tips (same brand).

- The Air Compressor: The air compressor permits to deliver air at higher pressure from the compression of air at atmospheric pressure. Model MB0102S, brand: PUMA.

To measure the dimensions of the deposited drops, we use a microscope associated with a camera. The camera offers a view from the top, and we use a mirror prism to have also a view from the side. Then it is possible to make measurements on the drop.

## 2.2 Choice of the polymer

Before beginning our experiments, we had to find some polymers which respected different criteria for our study and which could be used with our devices. The criteria are:

- Transparency;
- Not expensive;
- A viscosity not too high in order to make the deposition easier;
- A low shrinkage after curing;
- A contact angle greater than 0°.

This why we have studied different articles found on the internet in order to find some interesting polymers and their properties. We have finally selected the NOA81 [3] of the brand Norland Product © with the following properties:

Tab. 1. Properties of the NOA81.

| Name | Curing Method | Refractive Index | Viscosity | Contact angle |
|---|---|---|---|---|
| NOA81 | UV (E=2J/cm$^2$) | 1,56 | 300 cps at 25°C | 40° on glass |

The NOA81 is easy to obtain and have a reasonable cost. For the curing, the polymer is sensitive to UV with wavelengths between 320 and 380 nm (peak at 365 nm) so it will be possible for us to cure it. The viscosity before the curing is not very high (300 centipoise corresponds to motor oil).

## 2.3 EDM

To perform a good reproducibility of our experiments, we have decided to fix the syringe on a device used for the Electrical Discharge Machining (EDM). The machine is a Sodick LP1 model. The motorized arm allow us to choose the motion of the syringe in every direction.

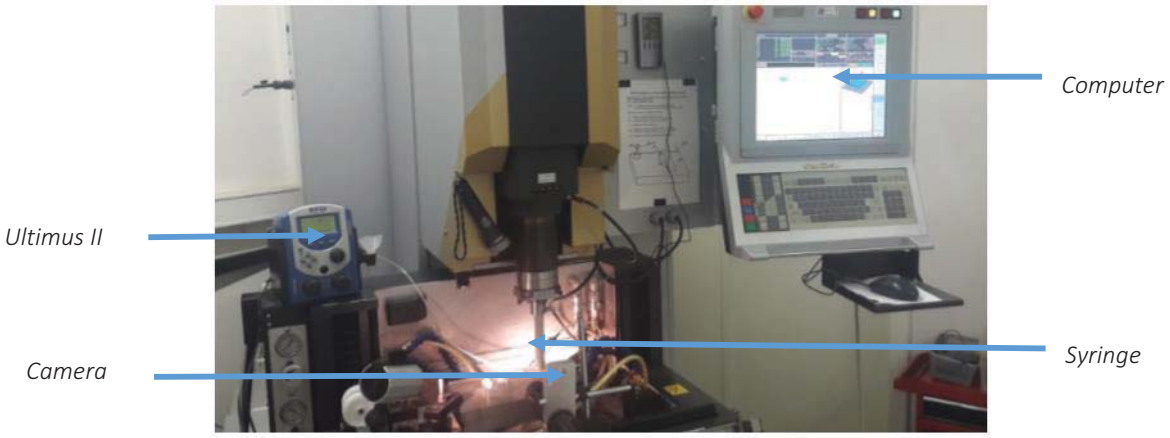



Fig.2. Experiment setup with the EDM

We can control the motion both thanks to a controller module (which does not appear on the picture above) or thanks to the computer. The computer has two different modes: manual or program (execute). In the manual mode we can change the values of the position (X,Y and Z axis) by modifying the absolute value or by incrementing the values on the three axis. With the device, it is then possible to control the position of the deposited drops, but also to set the inclination of the syringe. This parameter influences the amount of liquid that comes out of the syringe, and it is now possible to control it.

## 2.4 Image Analysis Software

We use different software to analyze our images during the laboratories sessions:

- IC Capture 2.4;
- IC Measure 2.4.

These two software allow us to take snapshots and videos thanks to the camera and make some measurements.

For the analysis of the snapshots and videos, we have used the software GIMP 2. This software allows to calculate distances in pixel. Thanks to that, it is possible to find the real distance by knowing the length of a pixel (calibration). The calibration is performed with a known distance such as the diameter of the syringe or the thickness of the substrate.

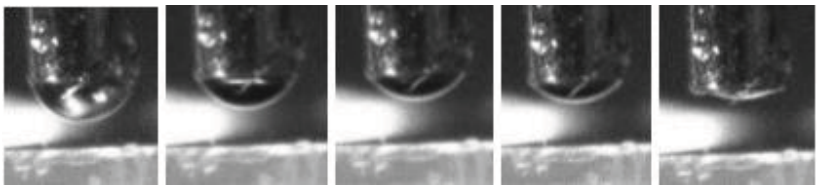
With the diameter (d, in µm) and the thickness (e, in µm), it is possible to find the contact angle of the drop. First we can compute the curvature radius (R, in µm) with the following formula $R = \frac{1}{2e} * (\frac{d^2}{4} + e^2)$ (1).

Then, the contact angle $\boldsymbol{\theta}$ is: $\theta = \sin^{-1}(\frac{d}{2R})$ (2). It is also possible to compute the apparent surface of the drop which can be a useful parameter to compare the drops: $S = R^2 * \cos^{-1}\left(\frac{R-e}{R}\right) - (R-e) * \sqrt{R^2 - (R-e)^2}$ (3).

# 3 Experiments

## 3.1 Drop deposition with the EDM

We choose an angle of 90 ° and create a program to have repeatable conditions. The initial position (z0) of the syringe is 0,4mm above the substrate. The pressure P2 is then set to create a meniscus:



P (inH2O)= 0,7 1 1,5 2 2,5

Fig.3. Meniscus created at the end of the tip when at the syringe is at 90°

A second position (z1) very close to the substrate is then set to bring the meniscus in contact with it. We then go back quickly to z0 and the deposition is made. The distance z1 will also influence the final shape of the drops. On the following picture, we can observe an array of lenses. The 3 first drops from the left are made with z1 = 0,05 mm, the 3 next are made with z1 = 0.04 mm and the 3 last are made with z1 = 0.03 mm :

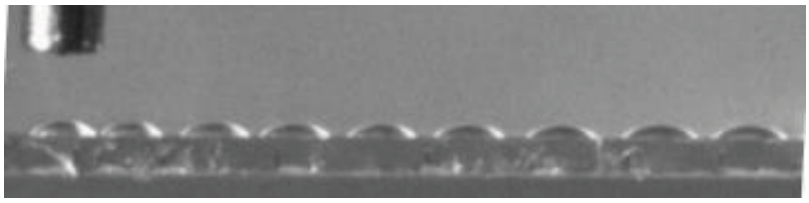


Fig.4. Array of lenses obtained with the previous deposition technique

We can see that the liquid is not spreading in the same way for each value of z1, so it is a parameter that has to be taken into account to determine the final shape of the lens. The problem is that glass substrate is not perfectly planar There can exist variations up to 0,02 mm along the surface. This is why it is hard to control perfectly z1. That is why we decide to fix z1, and only change the pressure of deposition with the Ultimus II. Finally, only one parameter will be modified to achieve the different sizes of drops that we want.

Fig.5. Top view of the array of lenses

By setting different pressures, we can create various height of meniscus. This height will modify the size of the deposited drop. From the experiments, it is possible to draw some graphs to represent the evolution of the dimensions of the deposited drops with the height of the meniscus:

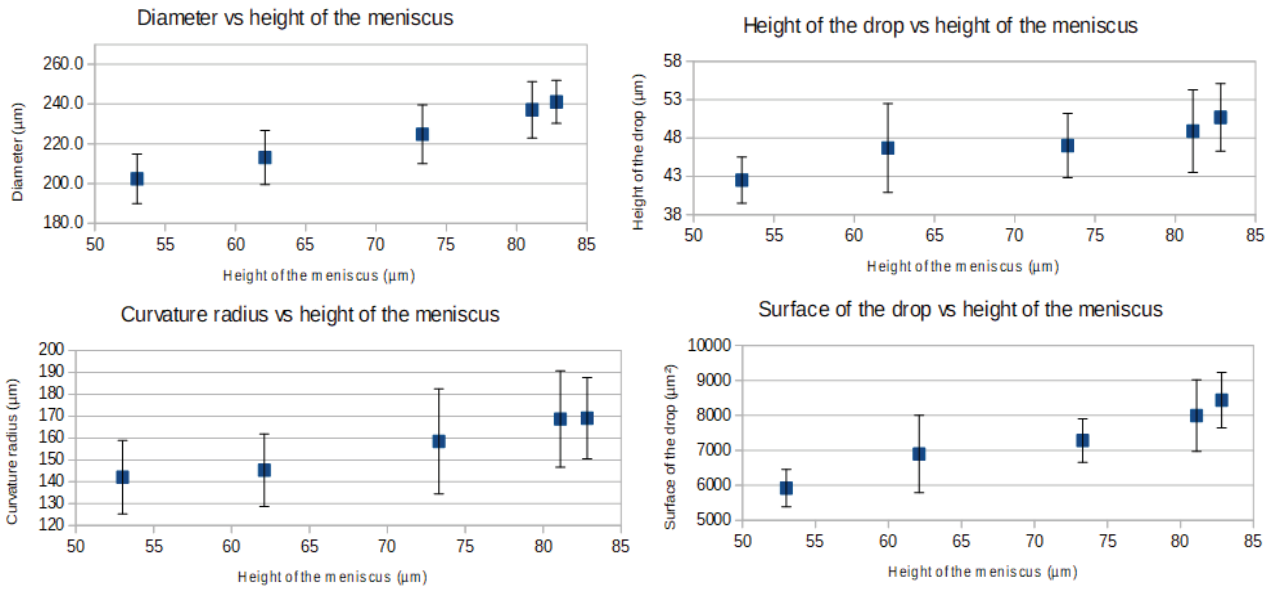



Fig.6. Dimensions of the drops vs height of the meniscus

We observe that the diameter, the height, the curvature radius and the surface of the drops are decreasing when the size of the meniscus is decreasing, which was predictable. We can also notice that for the curvature radius and the height of the drop, the standard deviation is approximately equal to 5%. Thus, these values are less reliable than the ones obtained for the diameter and the surface of the drops.

Tab. 2. Study of the contact angle of the drop and the height of the meniscus according to parameter P2

| P2 (inH2O) | 0,9 | 0,8 | 0,6 | 0,4 | 0,2 |
|---|---|---|---|---|---|
| Height of the meniscus (µm) | 53 | 62 | 73 | 81 | 83 |
| Contact angle (°) | 45,5 | 47,4 | 45,4 | 44,9 | 45,6 |
| Standard deviation | 2,0 | 2,6 | 2,8 | 2,6 | 2,2 |

It seems that the contact angle doesn't really depend on the meniscus, and is always worth between 45 and 47,5° approximately.

## 3.2 Curing

Before observing some images through our lenses, we need to cure our drops so as to form microlenses. For the UV curing of the drops of NOA81, we use a nail dryer. Its power is 6 mW/cm² (measured with the UV light meter "digital instrument YK – 35UV"). We know from the manufacturer of the NOA81 that 2 J/cm² are needed to fully cure the polymer. Then, we will have to put the drops 333s under the UV source to create our lenses.

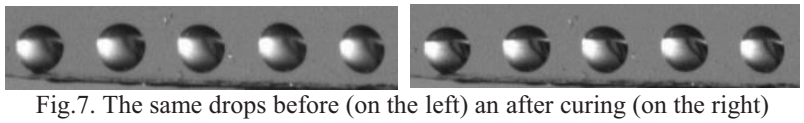


Fig.7. The same drops before (on the left) an after curing (on the right)

By eye, there is nearly no differences, but the drops are now cured and are then hardened. It is now a micro-lens. For 11 consecutive drops, the diameter has been measured (from the top view) before and after curing. The mean difference computed is 0,3%, so it is looking that the shrinkage after curing is very low, and don't have a big influence

on the final shape of the lenses.

### 3.3 Images obtained through the lenses

It is possible to observe objects through the microlenses. For example, we observed the cover of a book, placed at 50 mm behind an array of lenses that we created. The following pictures show the image obtained by focusing the camera on the surface of the glass and the image obtained on the image of the book through the lens.

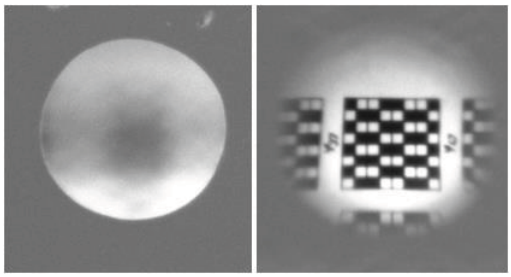


Fig.8. Focus on the surface of the glass (left) and on the image of the book through the lens (right)

From the distance between the surface of the glass and the image through the lens and the distance between the substrate and the book, we can compute the focal distance thanks to the following formula:

$$\frac{1}{f} = \frac{1}{D_{glass/object}} - \frac{1}{D_{glass/image}} \quad (4)$$

The focal distance (f) can also be computed with the curvature radius (R) and the refractive index (η):

$$\frac{1}{f} = \frac{n-1}{R} \quad (5)$$

We obtain results between 500µm and 800µm.

## 4 Conclusion

To conclude, we can say that we have reached our goal as we found a technique to deposit micro drops on glass. We have reached values ranging between 200 and 240 µm for the diameter, between 40 and 50 µm for the height leading to a contact angle of approximately 45° on the glass. We have also implemented one protocol with the EDM which allowed us to create arrays of lenses with a good reproducibility. Finally, we were able to cure our drops in order to study the optical properties and observe images through them.

If we have had more time, we could have tried to deposit other polymers than NOA81 and use more different substrates for the deposition so as to create lenses with different shapes and sizes. It could have been also interesting to create a template on our substrate to make perfectly reproducible arrays of lenses.

### Acknowledgement

Work has been supported by the Faculty of Mech. Eng. of CTU in Prague. We thank Jan Hosek for having suggested the idea of the project and for the help he gave us throughout the project.

## References

[1] H.Jiang, X.Zeng. Microlenses, *Properties, Fabrication and liquid lenses*. CRC Press. Taylor & Francis Group, 2013, 208.
[2] Nordson EFD Dispensing System [online] available on: www.nordson.com/en/divisions/efd/products/fluid-dispensing-systems/ultimus-i-ii-dispensers

[3] Norland Products [online] available on: www.norlandprod.com/adhesives/noa%2081.html

# DATA MINING FOR LABELLING DATA FROM SMART BUILDINGS

***Kristina Redenšek [1], Cyril Oswald[2]***

***[1] University of Ljubljana, krisitnaredensek@gmail.com***
***[2] Czech technical university in Prague, cyril.oswald@fs.cvut.cz***

*Abstract: In recent years, the development trend of buildings is in to maximize user's comfort and to minimize energy consumption. As a result of these trends, smart buildings are becoming more and more popular, equipped with many sensors that measure parameters that are connected with the residents' comfort and the energy efficiency of the building, today there is a flood of wireless sensors that can be easily installed even into an already existing building. As a result of a lot of buildings equipped with sensors it often happens that we get some data from sensors that is without labels. However, knowing the labels in further processing is crucial. Manually determining labels is time consuming work. Thus, our goal was to create a program which can categorized a set of data to one of multiple predefined category. A method that suited our task the most was a method with a decision trees. In order to use this method we did first preprocessing of data to choose the most appropriate features and after classification. Result of our work was a program that included preprocessing data and data classification using the decision trees method. Predictions were in more than 90% correct for all classes. To conlclude, this program greatly facilitates and accelerates data labelling, which was for before done manually.*

## 1 Introduction

In recent years, the development trend of buildings is in to maximize user's comfort and to minimize energy consumption. As a result of these trends, smart buildings are becoming more and more popular, equipped with many sensors that measure parameters that are connected with the residents' comfort and the energy efficiency of the building. If over the past years, the focus has been on installing sensors in buildings, today, there is a flood of wireless sensors that can be easily installed even into an already existing building. The result of a lot of sensors is a large amount of data. If we want to benefit from measuring all possible quantities, we need to use this data wisely. Therefore, in recent years, a great emphasis has been placed on the analytical treatment of these data. If data is properly captured and statistically processed, we can make excellent statistics models that dynamically regulate the operation of HVAC and other systems and thus minimize energy consumption and increase the comfort in buildings. The most popular approach in field of analysis of data is data mining. One of the most common methods that are used are: classification, clustering, regression, association rule learning. With these methods we can find some typical patterns and relations in the data.

As a result of a lot of buildings equipped with sensors it often happens that we get some data from sensors that is without appropriate labels. However, knowing the labels in further processing is crucial. Manually determining which data do we have is at the huge amount of data time demanding work. Thus, our goal was to create a program which can categorized a set of data to one of multiple predefined category. Doing that, we wanted to ensure a high percentage of accuracy in the implementation of this process. As the most suitable method we saw classification method of data mining. For getting better results we paid special attention in phase of preprocessing data, in which we also used visualization of data to have a better image of data properties.

The rest of paper is organized as follows. Section 2 some elements of data mining are introduced. Section 3 describes methodology of work. Section 4 reveals result of our work discuss it. In section 5 finally conclusion is made.

# 2 Data mining

Data mining is the analysis of observational data sets to find unsuspected relationships and to summarize the data in novel ways that are both understandable and useful to the data owner. Some of the methods involved in computing process of data mining are at the intersection of machine learning, statistics, and database systems. The relationships and summaries derived through a data mining exercise are often referred to as models or patterns. Examples include linear equations, rules, clusters, graphs, tree structures, and recurrent patterns in time series. Aside from the raw analysis step, it involves database and data management aspects, data pre-processing, model and inference considerations, interestingness metrics, complexity considerations, post-processing of discovered structures, visualization, and online updating. Typically deals with data that have already been collected for some purpose other than the data mining analysis. This means that the objectives of the data mining exercise play no role in the data collection strategy. This is one way in which data mining differs from much of statistics, in which data are often collected by using efficient strategies to answer specific questions. For this reason, data mining is often referred to as "secondary" data analysis [1, 2]. Most common classes of tasks supporting these steps are: anomaly detection (outlier/change/deviation detection), association rule learning (dependency modelling), classification, regression, summarization [3].

## 2.1 Decision trees

A “divide-and-conquer” approach to the problem of learning from a set of independent instances leads naturally to a style of representation called a decision tree. Nodes in a decision tree involve testing a particular attribute. Usually, the test at a node compares an attribute value with a constant. However, some trees compare two attributes with each other, or use some function of one or more attributes. Leaf nodes give a classification that applies to all instances that reach the leaf, or a set of classifications, or a probability distribution over all possible classifications. To classify an unknown instance, it is routed down the tree according to the values of the attributes tested in successive nodes, and when a leaf is reached the instance is classified according to the class assigned to the leaf [4].

### Metrics

Algorithms for constructing decision trees usually work top-down, by choosing a variable at each step that best splits the set of items. Different algorithms use different metrics for measuring "best". These generally measure the homogeneity of the target variable within the subsets. These metrics are applied to each candidate subset, and the resulting values are combined to provide a measure of the quality of the split [5].

### Gini impurity

Gini impurity is a measure of how often a randomly chosen element from the set would be incorrectly labeled if it was randomly labeled according to the distribution of labels in the subset. Gini impurity can be computed by summing the probability $p_i$ of an item with label $i$ being chosen times the probability $\sum_{k \neq i} p_k = 1 - p_i$ of a mistake in categorizing that item. It reaches its minimum (zero) when all cases in the node fall into a single target category [6].

To compute Gini impurity for a set of items with $J$ classes, suppose $i \in \{1,2,\ldots,J\}$, and let pi be the fraction of items labeled with class $i$ in the set.

$$I_G(p) = \sum_{i=1}^{J} p_i \sum_{k \neq i} p_k = \sum_{i=1}^{J} p_i(1 - p_i) = \sum_{i=1}^{J}(p_i - p_i^2) = \sum_{i=1}^{J} p_i - \sum_{i=1}^{J} p_i^2 = 1 - \sum_{i=1}^{J} p_i^2 \quad (1)$$

# 3 Methodology of work

Our goal was to create a program which can classify un unknown set of data to one of multiple classes. A method that suited our task the most was a method with a decision tree. For making a classification program, we received a piece of data from a data warehouse. The data was collected in an office building somewhere in Czech Republic in January 2017. The variables were: indoor air temperature, outdoor air temperature, supply water temperature, return water temperature, warm service water temperature, warm service circulation, pump operation, energy meter actual and energy meter cumulative. This data was already manually classified so we could use it as a training data.

## 3.1 Work flow

We can divide the process of our program into two steps. First step was preprocessing of data. This is a step that we need to do before starting an analysis of data and it is a very important step in doing analysis, because the quality of our later work (data mining) with this data depends on a quality of input data. That is the reason for paying a lot of attention at this step. Second step was data mining of data.

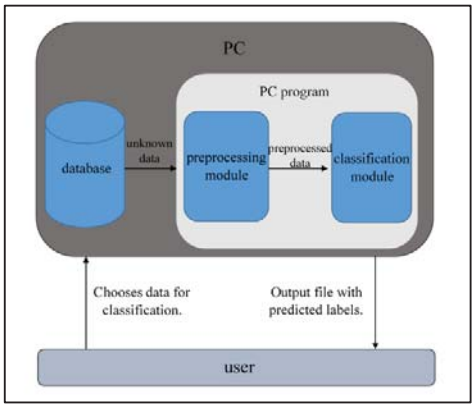



Fig. 1.: Block schematic presentation of a program.

### 3.2 Preprocessing

In preprocessing step, a target data set was assembled. As data mining can only uncover patterns actually present in the data, the target data set must to be large enough to contain these patterns while remaining concise enough to be mined within an acceptable time limit. To find out important feature of data sets we first visualized data in plots. After that we decided to consider next features: variable type (*boolean* or non *boolean*), measures of descriptive analysis (mean, maximum, minimum, standard deviation) and order of data (ascending, descending). We decided to drop missing data.

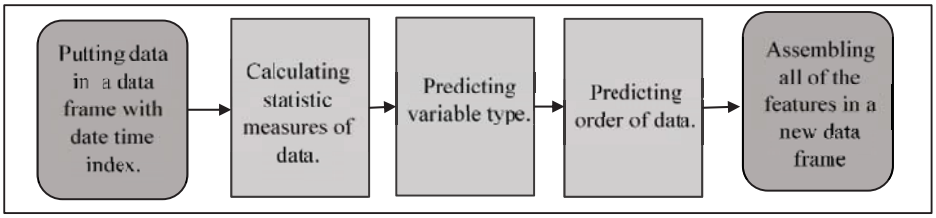



Fig. 2.: Flowchart of a preeprocessing stage.

### 3.3 Classification

Tab. 1.: Classes and matching labels.

| Class | Labels |
|---|---|
| indoor air temperature | indoor air temperature |
| outdoor air temperature | outdoor air temperature |
| water temperature | supply water temperature |
| | return water temperature |
| | warm service water temperature |
| energy meter actual | energy meter actual |
| energy meter cumulative | energy meter cumulative |
| boolean data | pump operation |
| | warm service water circulation |

After making a visualization of data we noticed that some of data had very similar shape and accordingly features, so it was hard to distinguish between them. Due to that, we decided that we join some label into one class. In a table 1 determined classes and labels that match a class can be seen.

We did tests and according to them we made first separation of classification manually in order to get better results. So we implemented an algorithm that separated *boolean* and non boolean data. Since the *boolean* data was

already a class for classification, we did not process this data anymore. For the other data further mining was needed.

We did a classification with a decision tree with this data. For executing this method, we used library scikit – learn. First we assembled training data to make a decision tree. The measure of the quality of the split set in a decision tree was Gini impurity. For keeping the decision tree understandable and not to complicated, we limited the depth of a decision tree to 5.

## 4 Results

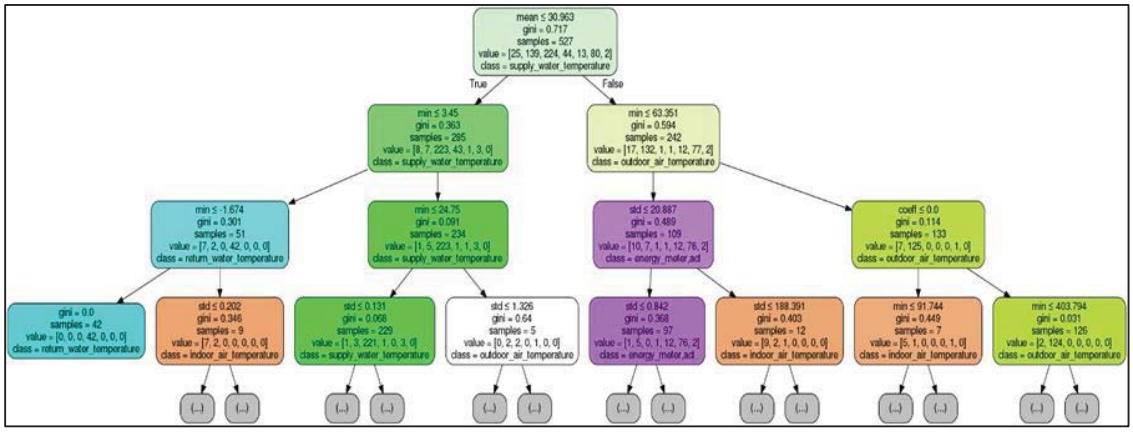



Fig. 3: A Classification tree for classifying non boolean data.

In Figure 3 is a decision tree that was made for classifying non *boolean* data into classes. As expected after seeing plots the first separation criteria were mean value of variables. In second step min values were obtained and in further steps the other features. Results of classification in percentage can be seen in Figure 4.

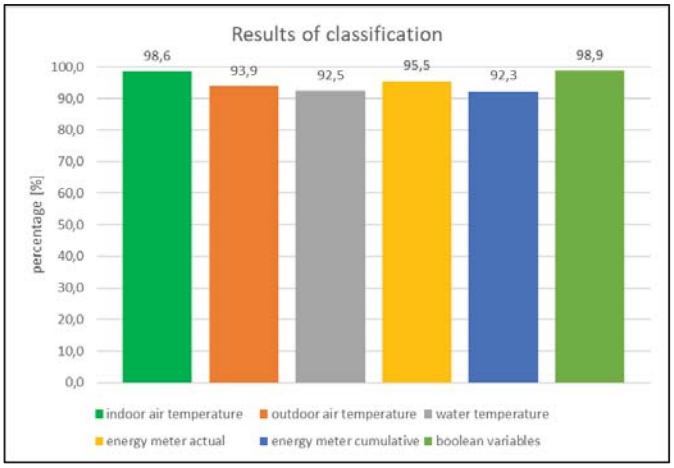



Fig. 4.: Results of classification in percentage.

Percentages of correct prediction of class are encouraging. All prediction results were in more than 90% correct. The best results were for *boolean* variables (98,9 %) and for indoor air temperature (98,6%). The worst results were for energy meter cumulative (92,3%), which is a surprise according to its unique shape of plot. A reason for that result are probably outliers such as always 0 value or lower values than usual. The result for water temperature is also one of the worst ones, even though we made a common class for 3 variables.

## 5 Conclusion

Our goal was to make a program that will be able to label a set of data. In order to fulfill this task, we made:

- A program that includes preprocessing data and data classification using the decision tree method;
- Preprocessing part of the program which takes care of data cleaning and computation of parameters of data for classification;

- Classification part of the program, which uses training data to build a decision tree and according to this decision tree than classifies tested data;
- A program that greatly facilitates and accelerates data labelling, which is for now done manually.

For classification, only those classes, which gave us satisfying results that are still reasonable and suitable for further use, were used. Thus, some classes included several variables. With this approach we achieved in more than 90% correct prediction for all the set classes.

In future program can be improved. One of the biggest improvement would be an algorithm that can classify all the variables with high enough accuracy. To accomplish that a deeper look at data is needed to find some specific features for each variable. Also other methods as regression and mutual information could be used.

## References

[1] Hand, D; Mannila, H; Smyth, P: *Principles of Data Mining.* A Bradford Book The MIT Press Cambridge, p. 5-20, 2001.
[2] KDD: *Data Mining Curriculum.* Available on: http://www.kdd.org/curriculum/index.html on 20. 12. 2017.
[3] Fayyad, U; Piatetsky-Shapiro, G;Smyth, P: *From Data Mining to Knowledge Discovery in Databases*. AI Magazine, 1996.
[5] Witten, I.H; Frank, E: *Data Mining (Practical Machine Learning Tools and Techniques). Second Edition.* Morgan Kaufmann, p. 62 – 82, 2005.
[6] Rokach, L; Maimon, O: *Top-down induction of decision trees classifiers-a survey.* IEEE Transactions on Systems, Man, and Cybernetics, Part C. 35 (4): 476–487, 2005.

# MATLAB TOOLBOX FOR ADAPTIVE IDENTIFICATION WITH GRADIENT DESCENT ALGORITHM

*Murat ÜNVER[1], Peter M. BENEŠ[2]*

*[1, 2] Department of Instrumentation and Control Engineering, Czech Technical University in Prague*

*{murat.unver, petermark.benes}@fs.cvut.cz*

*Abstract:*

*This paper presents a new MATLAB Simulink Toolbox for adaptive identification with use of high order neural units (HONUs) and gradient descent (GD) based learning. The toolbox allows users to investigate the potentials for adaptive identification via HONUs for dynamic modeling of linear and weakly non-linear industrial processes. The key contribution of this work is the development of a new toolbox for implementation of real-time identification and extension for adaptive control in the MATLAB Simulink framework.*

*Keywords:*

*adaptive identification, gradient descent (GD), higher order neural unit (HONU), linear neural unit (LNU), quadratic neural unit (QNU)*

## 1 Introduction

This paper presents a new toolbox implemented in MATLAB Simulink for adaptive identification via dynamic linear neural units (LNU) and dynamic quadratic neural unit (QNU) with use of the famous gradient descent algorithm (GD). Till now, HONUs have presented numerous successful results in both theoretical and well and real-time engineering applications [1]–[4]. As an initial, this paper recalls the key structures of HONU architectures for adaptive identification. Following this, the gradient decent (GD) algorithm with application to dynamic HONUs, as most recently presented in [5] is recalled. Then, this paper presents implementation of the HONU toolbox for real-time adaptive identification with connotations to the CTU Roller Rig application as a continuation of the works [4], [6].

## 2 Structure of Neural Unit (LNU and QNU)

The toolbox is created for two different neural unit structures; LNU (HONU, $r$=1) and QNU (HONU, $r$=2). The general form of an LNU may be given as follows

$$\tilde{y} = \sum_{i=0}^{n} x_i w_i = w_0 . x_0 + w_1 . x_1 + \ldots + w_n . x_n = \mathbf{w} \cdot col^{r=1}(\mathbf{x}) \tag{1}$$

where $r$ denotes the polynomial order, $\mathbf{x}$ is a vector of inputs and $\mathbf{w}$ is updatable vector of neural weights. Similarly a second order HONU architecture, more explicitly termed as a QNU (HONU, r=2) may be given as

$$\tilde{y} = \sum_{i=0}^{n} \sum_{j=0}^{n} x_i x_j w_{i,j} = w_{0,0} x_0 x_0 + w_{0,1} . x_0 x_1 + \ldots + w_{n,n} . x_n^{\ 2} = \mathbf{w} \cdot col^{r=2}(\mathbf{x}) \tag{2}$$

Where the long column vector form $col^{r=1}(\mathbf{x})$ comprises of previous step-delayed outputs $\tilde{y}(k-1)$ and further previous process inputs $u(k-1)$, more explicitly expressed as

$$col^{r=1}(\mathbf{x}) = \{x_i ; i = 0..n_x\}, \tag{3}$$

Further for a quadratic neural unit (QNU, i.e. HONU $r$=2), the long column vector maybe analogically expressed as

$$col^{r=2}(\mathbf{x}) = \{x_i x_j ; i = 0..n_x, j = i..n_x\}. \tag{4}$$

In case of dynamic linear unit, $col^{r=i}(\mathbf{x})$ step-delayed HONU model outputs, are incorporated however in the sense of static HONUs, the real process outputs may be chosen instead. This further simplifies the recurrent derivatives which may be seen in the proceeding section regarding the derived gradient descent update rule.

## 3 Gradient Descent (GD) Learning Algorithm for HONU Adaptive Identification

The toolbox created as purpose of this project, gradient descent algorithm is used. It is one of the fundamental algorithm behind neural unit identifier and controller applications. General form of the algorithm is adopted to LNU and QNU. It is expressed for LNU as below;

$$\Delta \mathbf{w} = -\frac{\mu}{2} \cdot \frac{\partial e^2(k)}{\partial \mathbf{w}} = -\mu \cdot e(k) \cdot \frac{\partial (y_r(k) - \tilde{y}(k))}{\partial \mathbf{w}} = \mu \cdot e(k) \cdot \frac{\partial \tilde{y}(k)}{\partial \mathbf{w}} \tag{5}$$

where $\mu$ represents learning rate of the weight adaptation, $e(k)$ represents current error between real and neural unit output, and $\partial y(k) \partial wi$ corresponds to the computed partial derivative of y with respect to each neural weight. Often for practical engineering applications, a modification of the classical rule yields via normalization of the learning rate. This modification is termed more explicity as the normalized gradient decent rule (NGD) which may be expressed as

$$\Delta \mathbf{w} = \frac{\mu}{\left\| col^{r=1}(\mathbf{x}) \right\|_2^2 + 1} \cdot e(k) \cdot \frac{\partial \tilde{y}(k)}{\partial \mathbf{w}}. \tag{6}$$

Once the weight update rule is established, each previous set of neural weights may be updated via the general update law. This form is also applicable in the sense of batch training algorithms.

$$\mathbf{w}(k) = \mathbf{w}(k-1) + \Delta \mathbf{w}. \tag{7}$$

## 4 Block Diagram of Adaptive Identification via HONUs

Above LNU, QNU, and GD concepts are explained in detail, here block diagram of adaptive identification is illustrated.

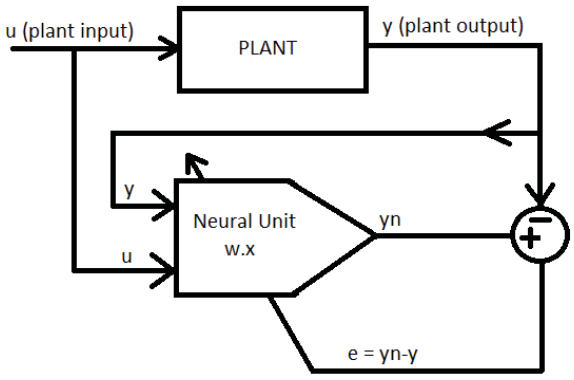



Fig. 1: Block Diagram of Adaptive Identification.

Process of identification can be described as, input signals inserted to system, as result output signal comes. These signals *u* and *y* are used to created vector of **x** as input vector and weights are updated by error calculation between system output and neural output. According to sampling rate these process repeat until neural unit identifies and learn the plant. QNU learns faster than LNU, therefore error rate of QNU is less than LNU. These comparison is shown in next section.

## 5 Matlab Simulink Toolbox

In this part, toolbox main structure, its function, and other details are presented with figures and explanations.Toolbox includes 5 main parts; system definition, **x** vector definition, LNU adaptive identifier, QNU adaptive identifier and visualization. In addition, there is also small script for parameters like time delay and initial conditions of neural weights. Parameters script given in appendix.

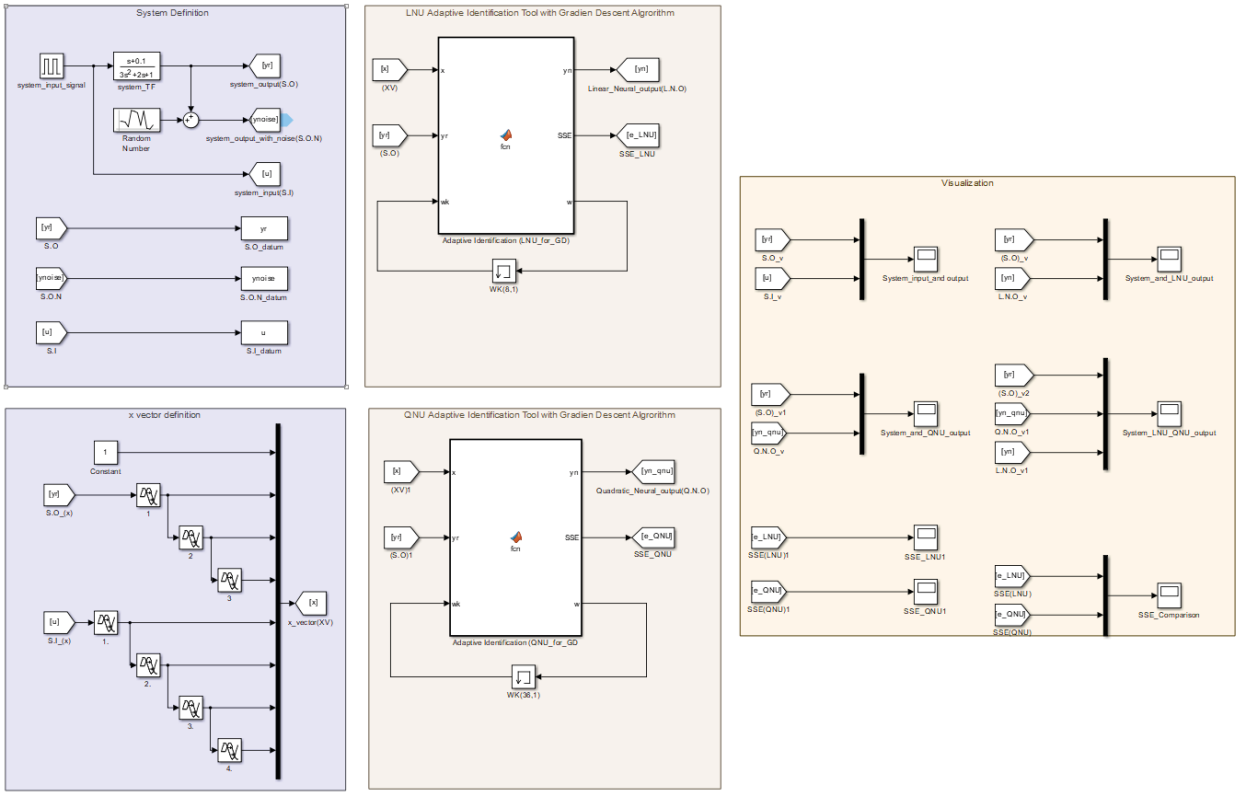



Fig. 2: MATLAB Simulink Toolbox.

In the following sections, toolbox sub tools are explained in detail.

### 5.1 System Definition

System definition part is created for users to insert their plant. For demonstration, the theorectical plant of second order is chosen as

$$\frac{s+0.1}{3s^2+2s+1} \tag{8}$$

The implementation is illustrated in Fig. 3;

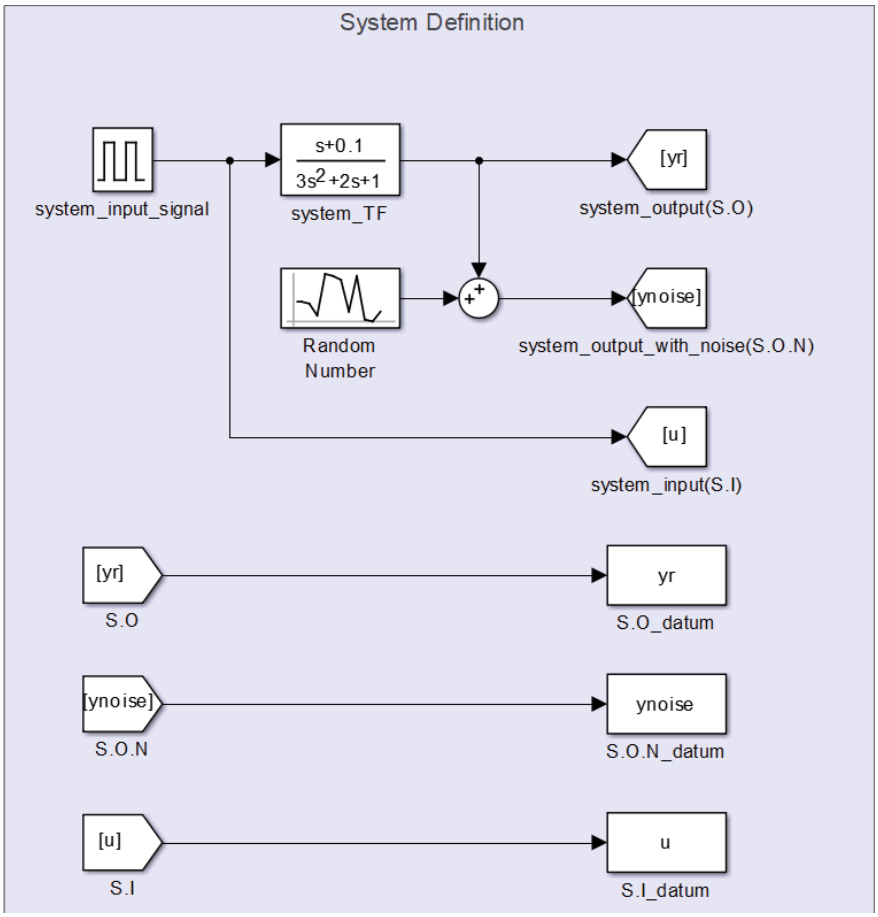



Fig. 3: System Definition.

Where u represents system input. Input signal is created by pulse generator for simulation. System output is assigned as "$y_r$" which means real system output. Also output signal with noise is add system for better understanding of adaptive identifier. All datum is connected to sink block to prevent chaos in the toolbox. And all datum are exported to workspace for observation in case of need.

## 5.2 Neural Input Vector Definition

As it is mentioned in previous sections, **x** is input vector for neural unit. It has components "*u*" and "*y*". After studying on sample and different systems, amount of "*u*" is chosen as 4, "*y*" is chosen as 3. Later these numbers are presented as $n_u = 4$, and $n_y = 3$.

Sampling of input datum is achieved by transport delay. Inside the configuration time delay is defined as dT to make users change this rate according to their applications.

Block of **x** vector definition is shown below.

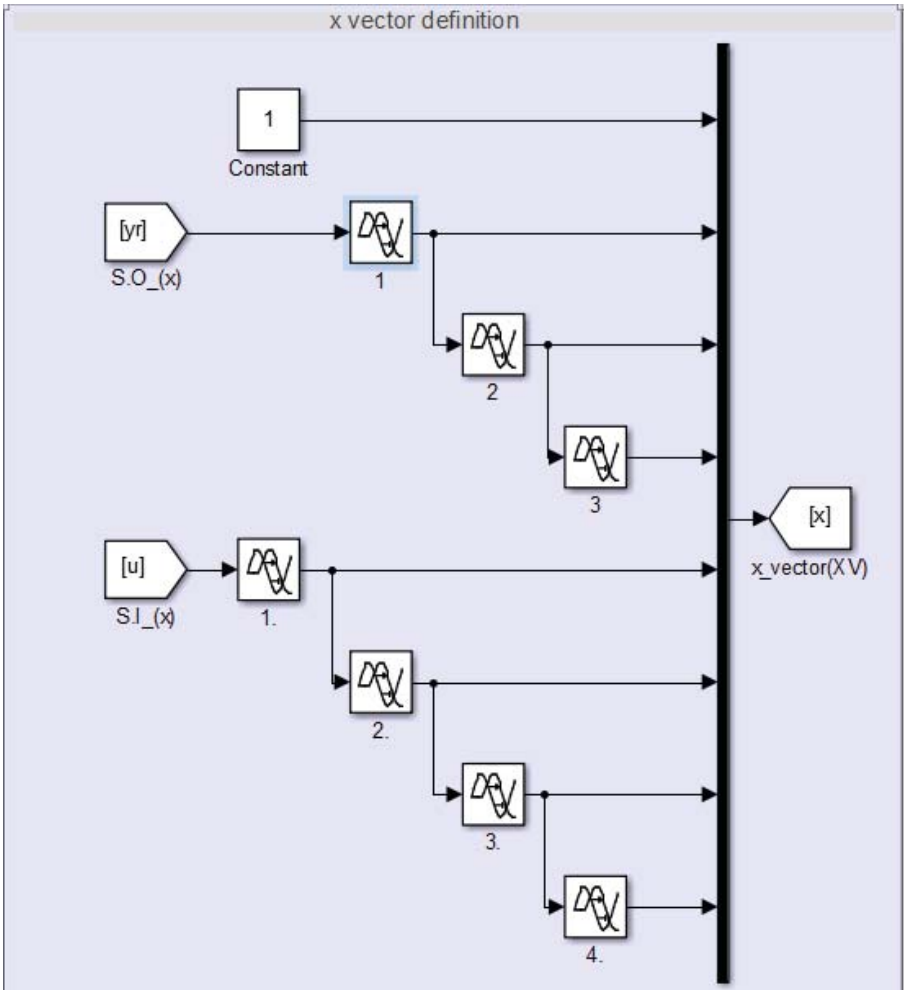



Fig. 4: Input Vector "**x**" Structure.

As it is visible in block, "**u**" and "**y**" signal come to the transport delay modules with sink blocks. **x** vector is inserted to another sink block to connect the MATLAB function module as input. Transport delay configuration creates vector as

$$\mathbf{x}(k) = [1 \quad \tilde{y}(k) \quad \tilde{y}(k-1) \quad \tilde{y}(k-2) \quad u(k) \quad u(k-1) \quad u(k-2) \quad u(k-3)] \tag{9}$$

Next parts mentions main tool for toolbox, LNU, and QNU adaptive identification tool.

### 5.3 LNU Adaptive Identification Tool

This part represents one of main tool for toolbox. Signals and input vector which are defined in previous tools are used as input for tool. These signals are processed inside the tool with embedded script. As result, tool gives 3 outputs there are; $y_n$ (neural output), $w$ (neural weights), and SSE (Sum of Square Error). The tool is illustrated as following;

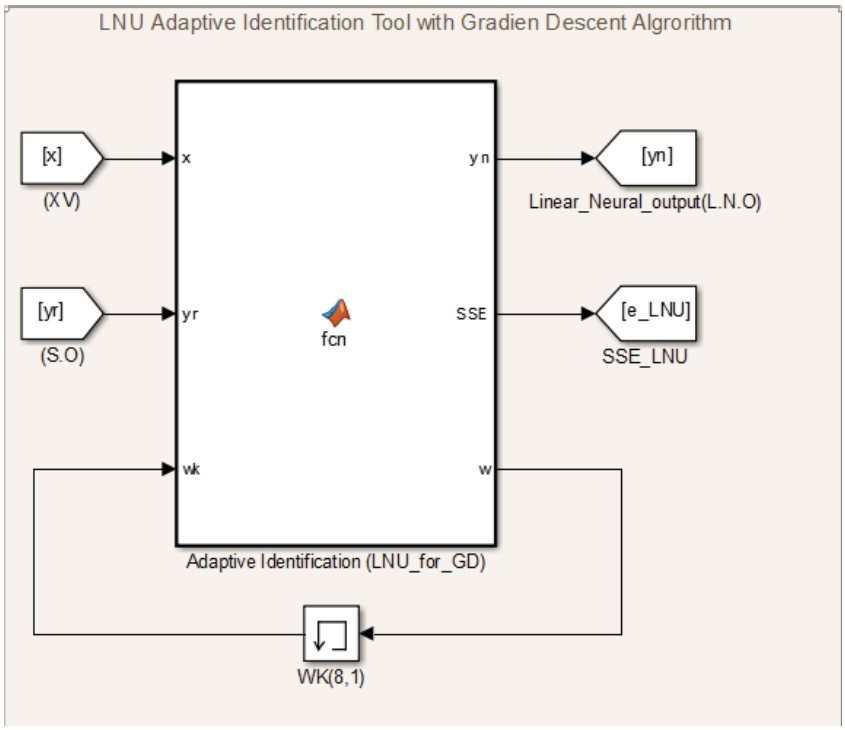



Fig. 5: LNU Adaptive Identification Tool.

Another variable in this tool "**w**($k$)". This variable updates itself in every cycle. For first cycle its initial value is zero. However, these initial conditions can vary so **w**($k$) is assigned as variable. As result, user can change this value in parameter section according to need.

### 5.4 QNU Adaptive Identification Tool

This section mentions another main tool for toolbox. Signals and input vector which are defined in previous tools are used as input for tool. These signals are processed inside the tool with embedded script. As result, tool gives 3 outputs there are; $y_n$ (neural output), **w** (neural weights), and SSE (Sum of Square Error). The tool is illustrated in Fig. 6. Another variable in this tool "**w**($k$)". This variable updates itself in every cycle. For first cycle its initial value is zero. However this initial conditions can vary so **w**($k$) is assigned as variable. As result, user can change this value in parameter section according to need.

*Remark: Dimension of weights vector is greater than LNU, due to* **x** *vector. For better understanding, readers may refer to the neural unit structure section.*

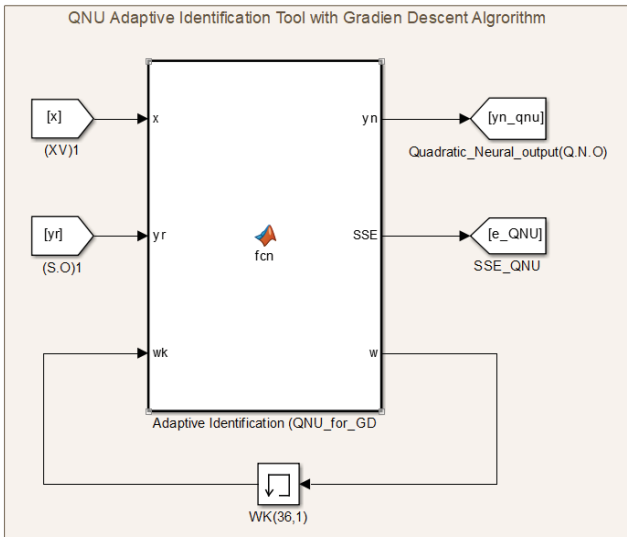



Fig. 6: QNU Adaptive Identification Tool.

## 5.5 Visualization

Visualization tool allows user to observer and compare results. This part includes; main system input and output signal graph, LNU output with system output graph, QNU output with system output graph, both LNU and QNU output with system output, SSE results, and SSE results for QNU and LNU for comparison. Tool is shown as below.

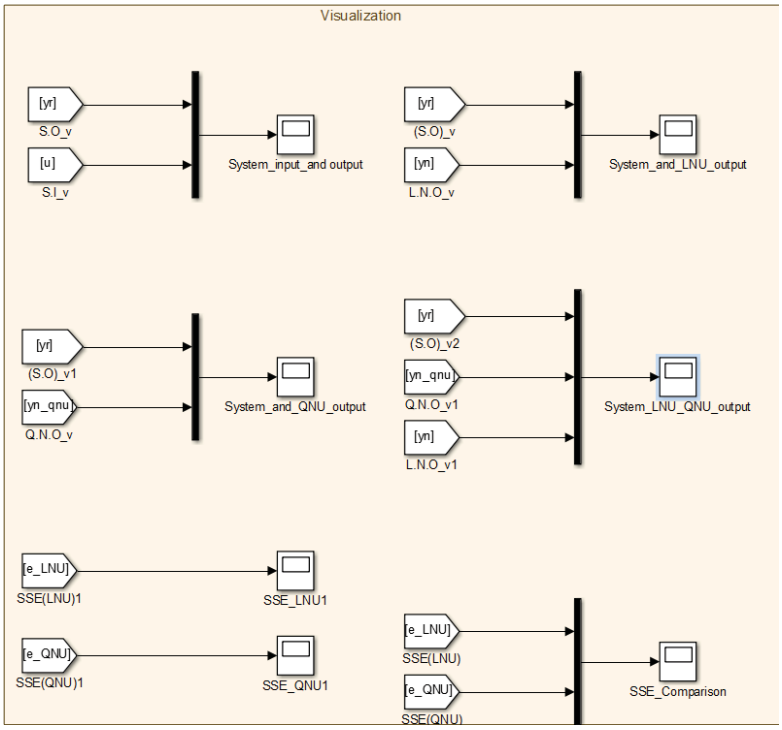



Fig. 7: Visualization.

For visualization scope tool of Simulink is used. In the following section results of example system are given.

## 6 Toolbox Results For Sample System

This section mainly illustrates result of the system chosen as example. Following figure shows system input ($u$) and output ($y$).

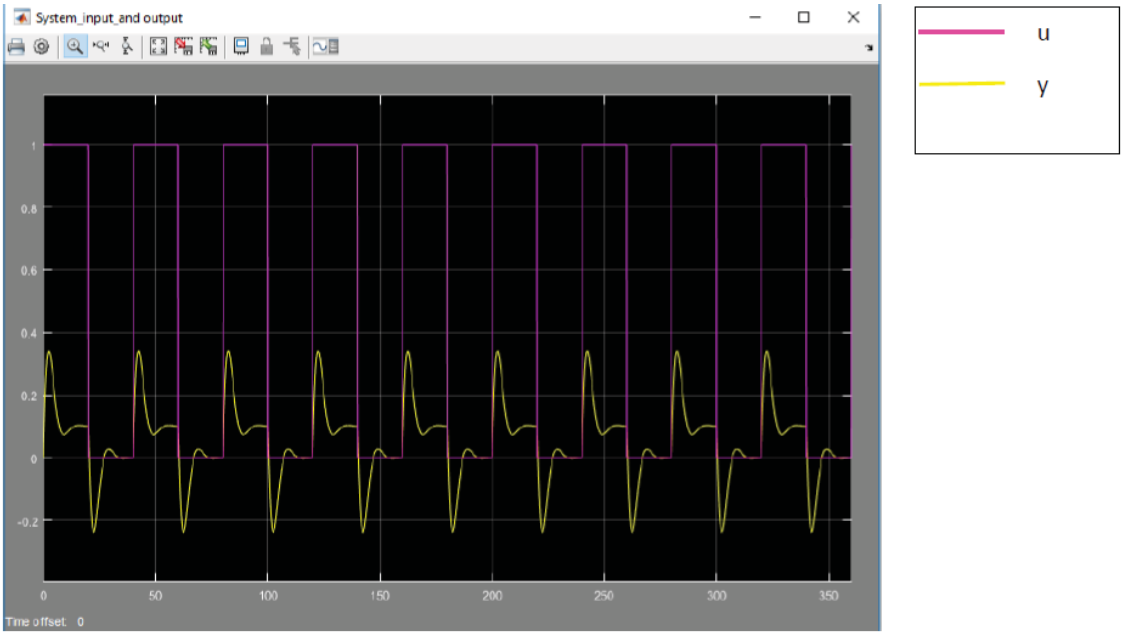



Fig. 8: System Input and Output.

Next figure illustrates system output with LNU output ($y_n$).

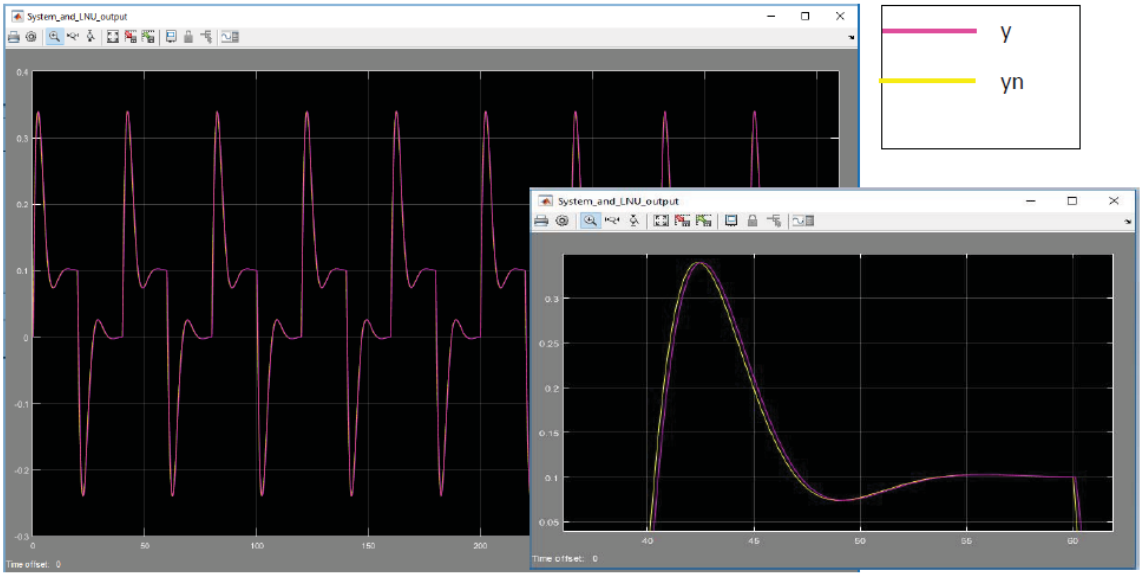



Fig. 9: System Output and LNU Output

Following figure shows system output with QNU output ($y_n$_QNU).

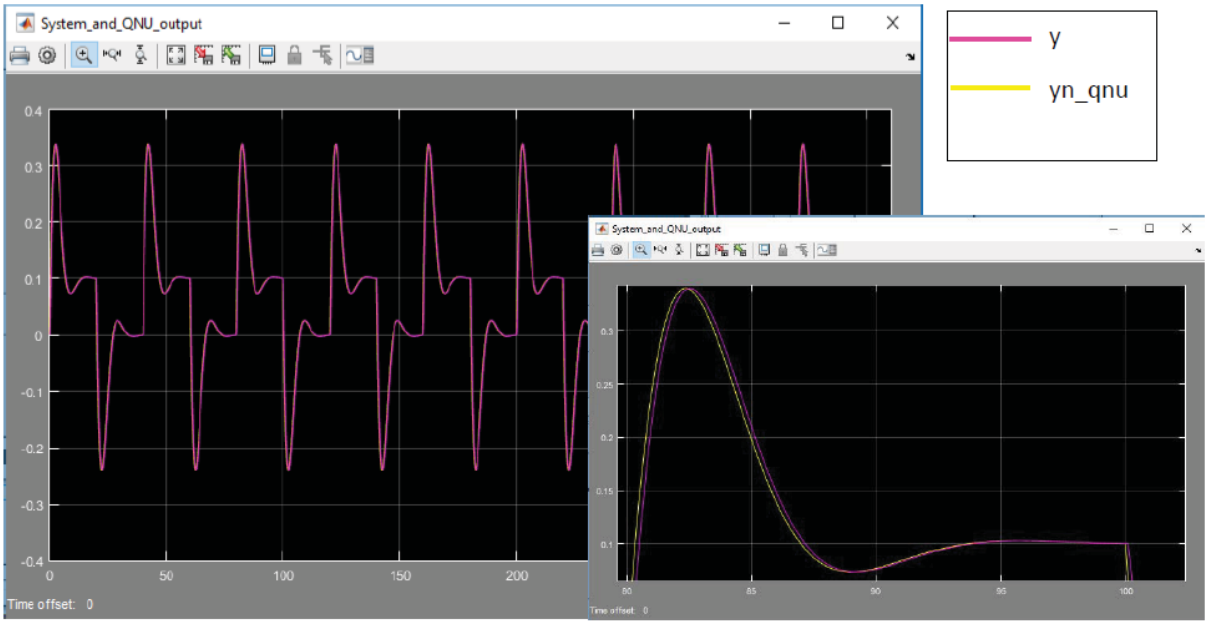



Fig. 10: System Output and QNU Output.

Next figure includes system output, LNU output, and QNU output together for easy comparison.

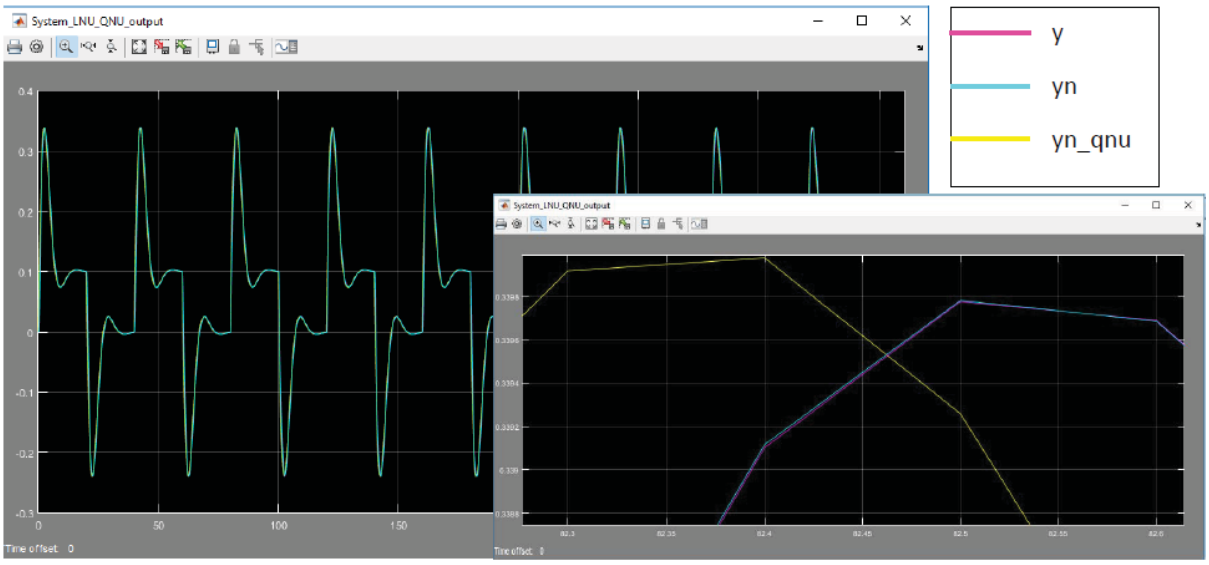



Fig. 11: System Output, LNU Output and QNU Output

Next graphs show SSE with respectively LNU, QNU, and LNU-QNU together.

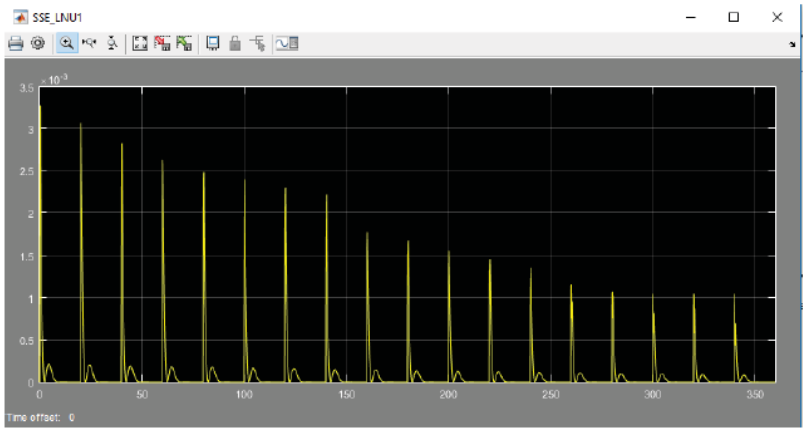



Fig. 12: SSE for LNU with respect to time.

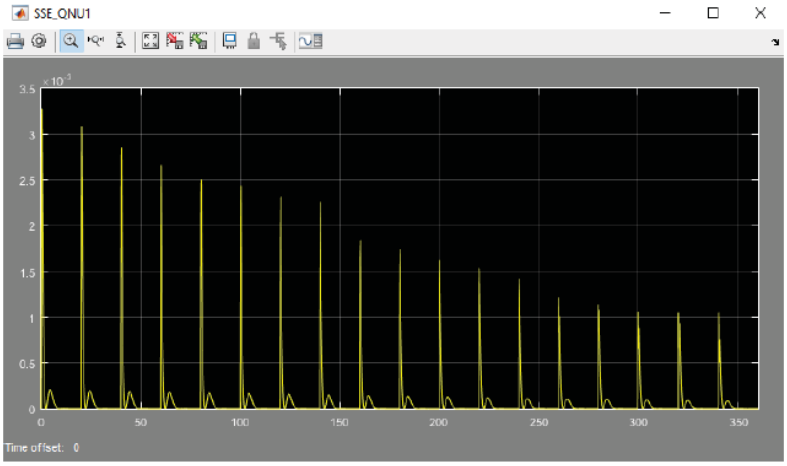



Fig. 13: SSE for QNU with respect to time.

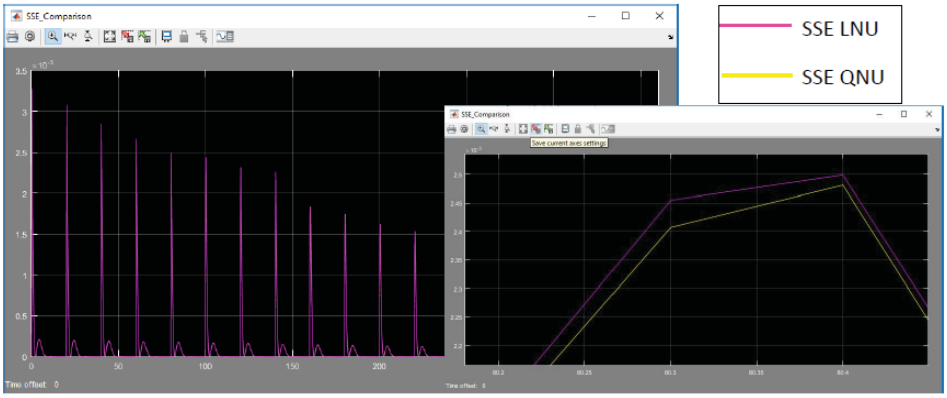



Fig. 14: SSE for QNU with respect to time

## 7 Conclusion

Following the results of this work, a MATLAB Simulink toolbox for adaptive identification was created and implemented on a theoretical linear system as a plant. To satisfy better understanding for readers, the fundamental gradient descent learning algorithm for LNU and further QNU architectures were recalled. From the produced experimental results, an important property was shown behind the comparison of a linear versus a quadratic neural unit architecture. From the presented results, it is evident that the QNU reduced its incremental SSE quicker than the same parameter settings and learning algorithm of a linear architecture. This highlights the capabilities of HONUs for efficient real time dynamic system identification in comparison to more complex conventional neural network (NN) architectures as such MLPs, which often require longer incremental training for similar order of approximation strength. Following the outcomes of this project, a future goal of this research is to extend the produced toolbox for application to the CTU roller rig, with focus to multiple-input-multiple-output (MIMO) configuration.

## References

[1] P. Benes and I. Bukovsky, "Neural network approach to hoist deceleration control," in *2014 International Joint Conference on Neural Networks (IJCNN)*, 2014, pp. 1864–1869.

[2] I. Bukovsky, P. Benes, and M. Slama, "Laboratory Systems Control with Adaptively Tuned Higher Order Neural Units," in *Intelligent Systems in Cybernetics and Automation Theory*, R. Silhavy, R. Senkerik, Z. K. Oplatkova, Z. Prokopova, and P. Silhavy, Eds. Springer International Publishing, 2015, pp. 275–284.

[3] L. Smetana, "Nonlinear Neuro-Controller for Automatic Control Laboratory System," Master's Thesis, Czech Technical University in Prague, Prague, Czech Republic, 2008.

[4] P. M. Benes, I. Bukovsky, M. Cejnek, and J. Kalivoda, "Neural Network Approach to Railway Stand Lateral Skew Control," in *Computer Science & Information Technology (CS& IT)*, Sydney, Australia, 2014, vol. 4, pp. 327–339.

[5] I. Bukovsky and N. Homma, "An Approach to Stable Gradient-Descent Adaptation of Higher Order Neural Units," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 28, no. 9, pp. 2022–2034, Sep. 2017.

[6] P. Benes, I. Bukovsky, and J. Kalivoda, "Achievements in Neural Network Approach to Railway Stand Lateral Skew Control," presented at the Nové metody a postupy v oblasti přístrojové techniky, automatického řízení a informatiky 2014, Herbertov, Czech Republic, 2014.

# MATHEMATICAL MODEL OF A HELICOPTER WITH SUSPENDED LOAD

*Matěj Čech (matej.cech@fs.cvut.cz)*

*Abstrakt: Práce se zabývá tvorbou modelu vrtulníku se zavěšeným břemenem, na kterém je možné zkoumat dynamické chování soustavy více těles. Cílem je úprava stávající konstrukce modelu vrtulníku pro lepší chování na delším lineárním vedení. Pro upravenou konstrukci pak vytvoření matematického modelu.*
*Klíčová slova: závěsné břemeno, vrtulník, matematický model*

*Abstract:This paper describes a creation of a model of helicopter with suspended load, on which is possible to study dynamic behaviour of multibody system. The goal is to adapt current model of a helicopter so it can be used with longer linear guide. The second goal is to create a mathematical model for this altered design.*
*Keywords: suspended load, helicopter, mathematical model*

## 1 Úvod

Manipulace se zavěšenými břemeny je náročná činnost, nejen z hlediska konstrukce manipulační techniky, ale hlavně z pohledu řízení. Při přesunu zavěšeného břemenem totiž dochází k jeho nežádoucímu rozkmitání, a proto je nutné, aby v cílové pozici byl v ustáleném stavu. Aby bylo možné simulovat chování takovéto soustavy, a následně navrhnou vhodné řízení, byl vytvořen laboratorní model vrtulníku se zavěšeným břemenem, na kterou navazuje tato práce.

## 2 Konstrukce

Model vrtulníku ve volném prostoru, který by umožňoval třídimenzionální pohyb, by byl velmi obtížně řiditelný a pokusy zaměřené na tlumení kmitů břemene tak jen těžko realizovatelné. Proto byla zvolena stávající podoba soustavy, kdy je model vrtulníku umístěn na lineárním vedení. Tím pádem se vrtulník pohybuje jednodimenzionálně, v konstantní vzdálenosti od země a na omezené vzdálenosti, určené délkou vedení. Nedochází tedy k nežádoucímu rozkývání břemene do stran, rotaci vrtulníku vůči břemenu a podobně. Znamená to sice, že není brán ohled na to, jestli by vrtulník v prostoru udržel výšku, a že do dynamiky chování promlouvá nezanedbatelná hmotnost vozíku a baterie, nicméně to není pro tuto úlohu problém. Záměrem bylo vytvořit soustavu, na které je dobře simulovatelné vzájemné působení těles a která má malé třecí ztráty ve směru pohybu.

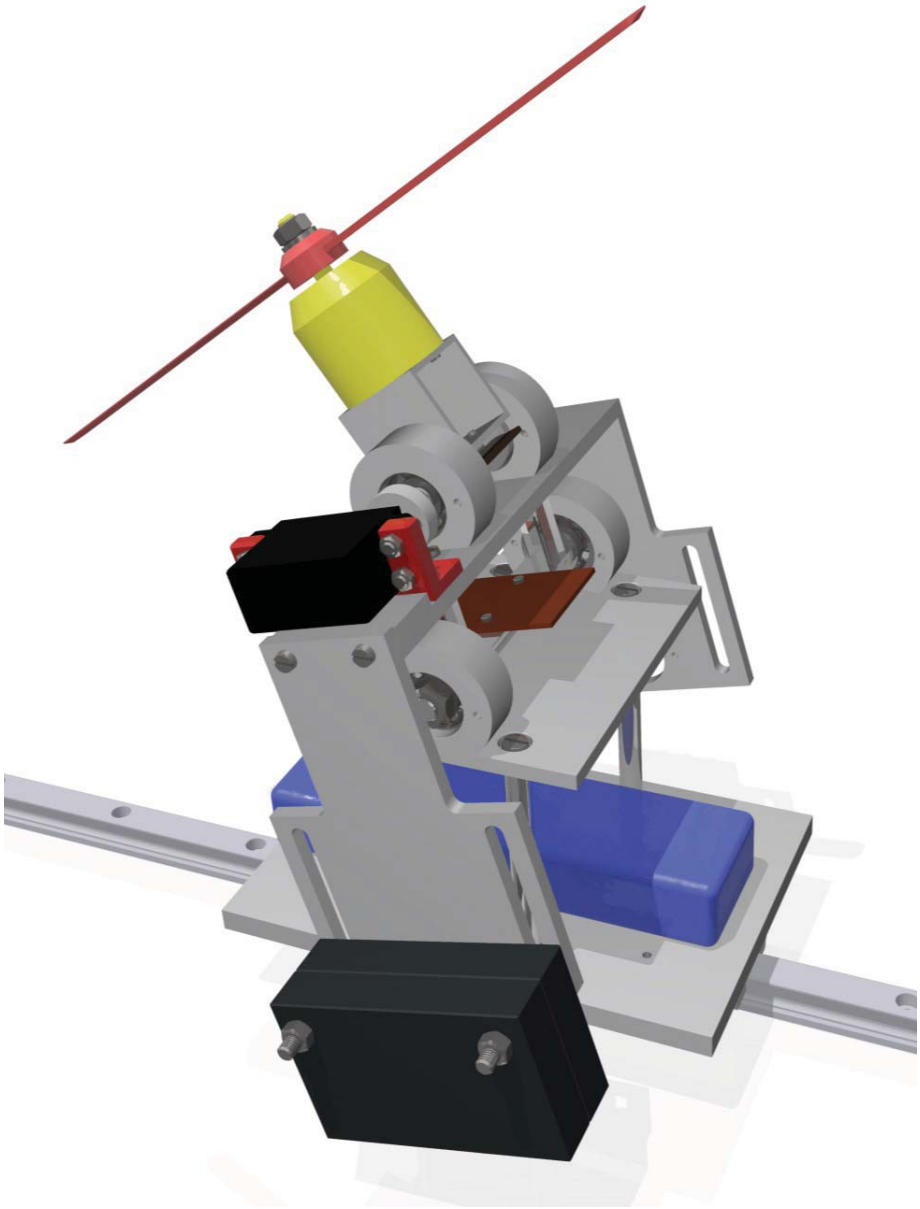
Hlavním nedostatkem původní podoby soustavy byla omezená délka lineárního vedení, na kterém se vozík s vrtulníkem pohyboval. Bylo proto rozhodnuto, že se zakoupí delší lineární vedení typu EGH 15 o délce 3 metry. K uchycení nových vozíků, typ EGH15CA, bylo nutné vytvořit jinou desku, na kterou bude uložena baterie a mikrokontroler STM32.

Vzhledem k šířce vedení bylo potřeba přesunout závěsné břemeno na protější stranu vozíku, než na které má být umístěno závaží střední části modelu, aby nevznikal příliš velký moment, který by mohl zvýšit tření vozíku a tak nežádoucím způsobem ovlivnit chování soustavy. Vnikl tedy nový díl pro uchycení servomotoru, který ovládá náklon motoru s vrtulí, a na který jsou uchyceny boční díly pro uchycení závaží střední části a zavěšeného břemene.

Při návrhu délky distančních sloupků bylo nutno brát v úvahu rozměry vrtule a úhly maximálního vychýlení v ložiscích tak, aby nemohlo dojít ke kontaktu mezi lineárním vedením a vrtulí. Rozsahy pohybu v ložiscích jsou vymezeny vyměnitelnými destičkami.

Kompletního přepracování se dočkalo uložení servomotoru, který ovládá natočení vrchní části s motorem a vrtulí. Pro vytvoření držáku byla využita metoda 3D tisku a držák servomotoru lze snadno vyměnit za jiný, který by měl sloužit pro uchycení jiného typu servomotoru. V horní desce jsou proto vytvořeny dvě díry se závitem M3 k uchycení držáku.

Zmíněné úpravy konstrukce vyřešily všechny požadavky na fungování, takže po osazení mikroprocesorem a snímači vychýlení břemene a střední části bude celek připraven k testování.



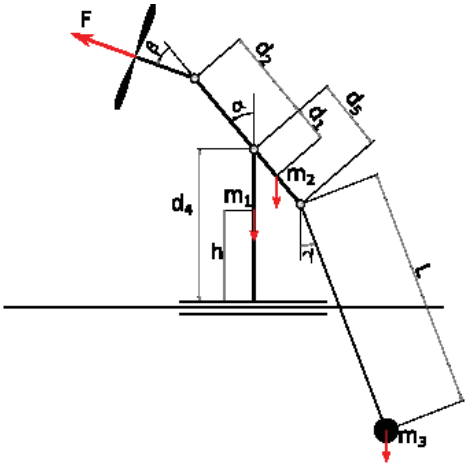
Obr. 1.: 3D model vrtulníku na lineárním vedení

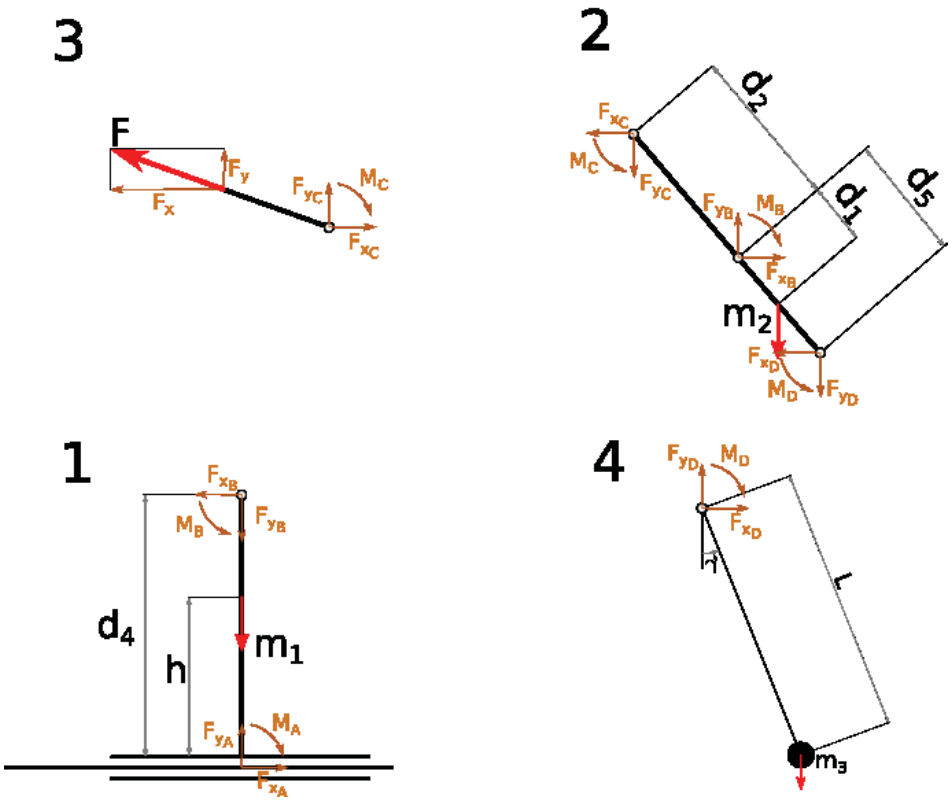
## 3 Matematický model

Dynamika vrtulníku se zavěšeným břemenem je popsána soustavou diferenciálních rovnic v následujícím tvaru

$$M(x(t))\ddot{x}(t) + C(x(t))\dot{x}(t) + K(x(t))x(t) + Q(x(t)) = L(x(t))u(t) \quad (1)$$

Vektor $\boldsymbol{x}(t) = [x, \alpha, \gamma]^T$ určuje pozici vrtulníku souřadnicí x, úhel $\alpha$ je sevřen mezi středním dílem a vertikální osou, úhel $\gamma$ pak určuje vychýlení zavěšeného břemene vůči vertikální ose. Vektor $\boldsymbol{u} = [F, \beta]^T$ obsahuje veličiny sloužící k ovládání soustavy. Síla $F$ reprezentuje tah vrtule, úhel $\beta$ pak její natočení vůči střednímu dílu.




Obr. 2.: Silové působení těles

Po uvolnění jednotlivých těles byly získány rovnice popisující silové působení mezi tělesy.

Těleso 1:

$$\ddot{x}_1(t).m_1 + F_{x_B} + f_{ta}.\dot{x}_1(t) = 0 \tag{2}$$

$$\ddot{y}_1(t).m_1 + F_{y_A} - F_{y_B} - m_1.g = 0 \tag{3}$$

$$M_A + F_{x_B}.h + M_B = 0 \tag{4}$$

Těleso 2:

$$\ddot{x}_2(t).m_2 + F_{x_C} - F_{x_D} - F_{x_B} = 0 \tag{5}$$

$$-\ddot{y}_2(t).m_2 + F_{y_D} + F_{y_B} - F_{y_C} - m_2.g = 0 \tag{6}$$

$$I_2.\ddot{\alpha}(t) + F_{x_B}.d_1.\cos\big(\alpha(t)\big) + F_{y_B}.d_1.\sin\big(\alpha(t)\big) - F.(d_1 + d_2).\sin\big(\beta(t)\big) - F_{x_D}.(d_5 - d_1).\cos\big(\alpha(t)\big) - F_{y_D}.(d_5 - d_1).\sin(\alpha(t)) + f_{tb}.\dot{\alpha}(t) + f_{tb}.(\dot{\alpha}(t) - \dot{\gamma}(t)) = 0 \tag{7}$$

Těleso 3:

$$\ddot{x}_3.m_3 + F_{x_D} = 0 \tag{8}$$

$$\ddot{y}_3.m_3 + F_{y_D} + m_3.g = 0 \tag{9}$$

$$I_3.\ddot{\gamma}(t) - F_{x_D}.l.\cos\big(\gamma(t)\big) - F_{y_D}.l.\sin\big(\gamma(t)\big) - M_D = 0 \tag{10}$$

Matice koeficientů diferenciální rovnice pak vychází z výše uvedených vztahů. Pro další práci bylo nutné provést linearizaci. Podmínky pro její provedení byly stanoveny následovně

$$\boldsymbol{x_0} = [0,0,0]^T \tag{11}$$

$$\dot{\boldsymbol{x}}_{\boldsymbol{0}} = [0,0,0]^T \tag{12}$$

$$\ddot{\boldsymbol{x}}_{\boldsymbol{0}} = [0,0,0]^T \tag{13}$$

$$\boldsymbol{u_0} = [1,0]^T \tag{14}$$

Výsledná podoba matic, popisujících chování soustavy po linearizaci, má tuto podobu

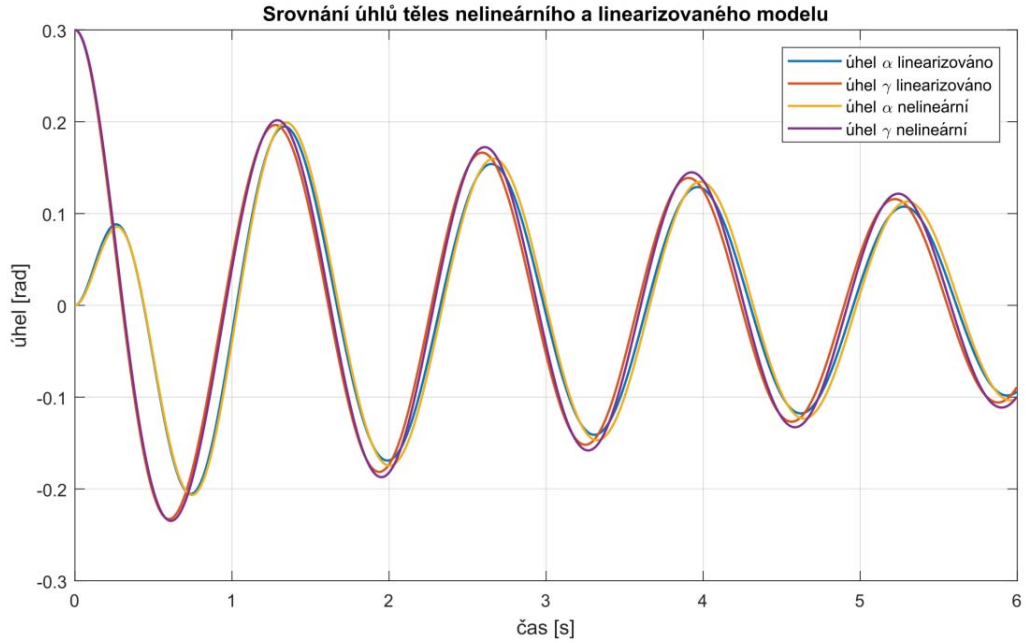
$$M_{lin} = \begin{bmatrix} m_1 + m_2 + m_3 & d_5.m_3 - d_1.m_2 & l.m_3 \\ -d_1.m_1 - (d_1 - d_5).m_3 & I_2 - d_5.m_3.(d_1 - d_5) & -l.m_3.(d_1 - d_5) \\ l.m_3 & d_5.m_3.l & m_3.l^2 + I_3 \end{bmatrix} \tag{15}$$

$$C_{lin} = \begin{bmatrix} f_{ta} & 0 & 0 \\ -d_1.f_{ta} & f_{tb} + f_{td} & -f_{td} \\ 0 & -f_{td} & f_{td} \end{bmatrix} \tag{16}$$

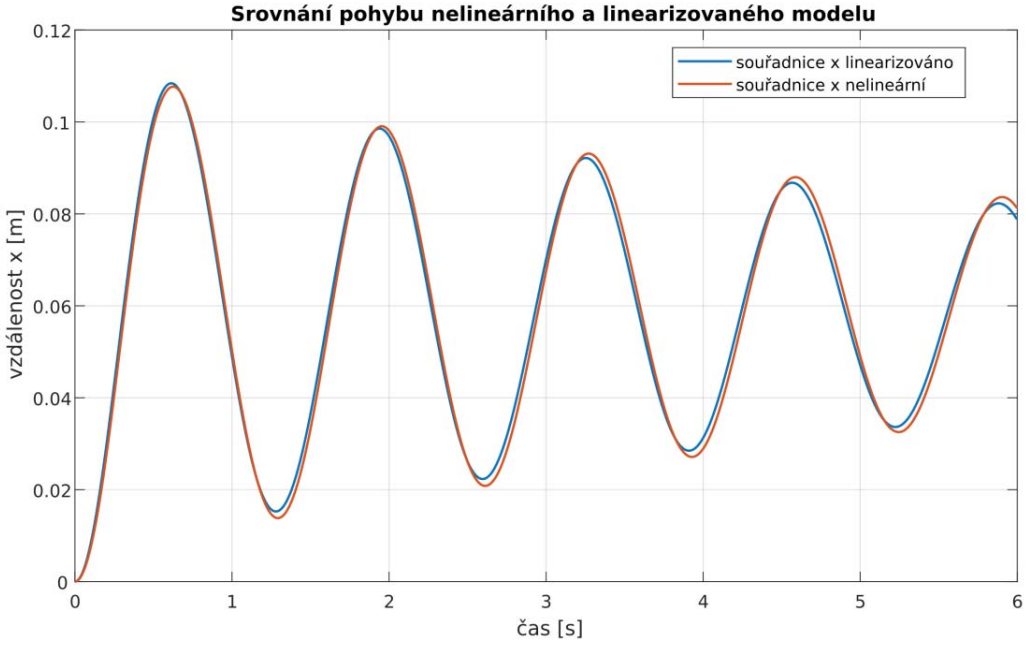
$$K_{lin} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & d_1.(g.m_2 + g.m_3 - 1) & 0 \\ 0 & 0 & g.l.m_3 \end{bmatrix} \tag{17}$$

$$L_{lin} = \begin{bmatrix} 0 & -1 \\ 0 & d_1 + d_2 \\ 0 & 0 \end{bmatrix} \tag{18}$$

Simulace chování pomocí programu Matlab umožňuje přímo porovnat chování nelinearizovaného a linearizovaného modelu. Při pohledu na graf, který ilustruje odezvu soustavy na vychýlení na začátku simulace, respektive nastavení nenulových počátečních podmínek na intergrátorech, lze konstatovat, že odchylky vzniklé linearizací nejsou příliš výrazné.




Obr. 3.: Srovnání úhlů těles soustavy nelineárního a linearizovaného matematického modelu




Obr. 4.: Srovnání pohybu nelineárního a linearizovaného matematického modelu

## 4 Závěr

Byly provedeny změny v konstrukci modelu vrtulníku se zavěšeným břemenem, které měly za cíl vyřešení požadavků na funkčnost soustavy za účelem testování tvarovačů signálu. Dále byla popsána tvorba matematického modelu této soustavy a ten byl následně porovnán s linearizovaným systémem. Ze srovnání vyplívá, že pro malé výchylky nedochází k větším rozdílům mezi linearizovanou a nelinearizovanou soustavou, viz Obr. 3. To, mimo jiné, nasvědčuje vhodně zvoleným parametrům pro provedení linearizace.

## Literatura

[1] J. J. Potter, C. J. Adams and W. Singhose, "A Planar Experimental Remote-Controlled Helicopter With a Suspended Load," in IEEE/ASME Transactions on Mechatronics, vol. 20, no. 5, pp. 2496-2503, Oct. 2015.

[2] M. Hromčík and T. Vyhlídal, "Inverse Feedback Shapers for Coupled Multibody Systems," in IEEE Transactions on Automatic Control, vol. 62, no. 9, pp. 4804-4810, Sept. 2017.

# THE OPTIONS IN ROBOTIC CONTROL OF REHABILITATING PATIENT´S LOWER LIMBS

***Karel Vošahlík[1], Jan Hošek [2]***

*[1] Ústav přístrojové a řídicí techniky, Fakulta strojní, ČVUT v Praze, karel.vosahlik@fs.cvut.cz*
*[2] Ústav přístrojové a řídicí techniky, Fakulta strojní, ČVUT v Praze, jan.hosek@fs.cvut.cz*

*Abstrakt: V současné době se neustále rozšiřuje využití robotizace ve všech oblastech včetně zdravotnictví. Zdravotnictví má však velké množství oblastí, ve kterých se robotika využívá nebo ji lze využívat. Návrh nových rehabilitačních přístrojů a pomůcek musí vycházet ze znalostí anatomie a fyziologie pohybu jednotlivých částí lidského těla v kombinaci s vhodnou analýzou pohybu jednotlivých rehabilitačních metod. Tento článek analyzuje fyziologické pohyby při vybraných rehabilitačních metodách a navrhuje možnosti jejich realizace pomocí robotických zařízení.*
*Klíčová slova: léčebná rehabilitace, robotizace, konstrukce*

*Abstract:The use of robotics is currently expanding in all spheres including healthcare. There are many fields of healthcare, where robotics is being or can be used. A design of new rehabilitation devices and aids has to come out from the knowledge of anatomy and physiology of specific body-part movements combined with proper analysis of body movements in particular rehabilitation methods. This article analyses physiological movements in selected rehabilita-tion methods focused on lower limbs rehabilitation. It also proposes options for realization proprioceptive neuromuscular facilitation by using robotic devices.*
*Keywords:therapeutical rehabilitation, Robotization, Construction*

## 1 Úvod

Lidská populace je vystavována různému onemocnění či úrazům. Zaměříme-li se na skupiny neurologického nebo ortopedického onemocnění, lze uvést například: cévní mozková příhoda, dětská mozková obrna, roztroušená skleróza, paraplegie, centrální i periferní parézy, ataxie, skoliózy, svalové poruchy, kloubní poruchy, pooperační stavy.

Léčba pacientů je fyzicky náročná pro samotné pacienty a také pro fyzioterapeuty. Při některých léčebných rehabilitačních metodách cvičí pacient aktivně dle instrukcí získaných od fyzioterapeuta. Při jiných léčebných rehabilitačních metodách provádí fyzioterapeut danou techniku a pacient se podílí aktivním či pasivním způsobem. Jsou však onemocnění či úrazy, při kterých má pacient značně poškozen pohybový aparát a léčba takových pacientů vyžaduje asistenci dvou až tří fyzioterapeutů. Nejčastěji tomu tak je při trénovaní chůze, kdy dva fyzioterapeuti vedou dolní končetiny a třetí fyzioterapeut přidržuje pacienta.

Nemocniční a rehabilitační zařízení lze rozdělit do pěti základních skupin. Do první skupiny lze zahrnout lůžka pro neodkladnou lékařskou péči. Tato lůžka jsou používána v nemocničních zařízeních například na oddělení centrálního příjmu, nebo na odděleních operačních sálů. Druhou skupinu tvoří standardní nemocniční lůžka pro následnou péči, která jsou umísťována na lůžková oddělení nemocničních zařízení. Na lůžkových odděleních jsou pacienti léčeni po chirurgických zákrocích, ale také je zde prováděna léčebná rehabilitace nevyžadující speciální vybavení či velký prostor.

Třetí skupinu tvoří speciální rehabilitační lůžka. Tato lůžka jsou navržena za účelem postupného zatížení pohybového aparátu pacienta pomocí vertikalizace cvičení chůze. Zaměříme-li se na obor léčebné rehabilitace, je jedním ze zástupců lůžko ErigoPro od firmy Hocoma [1]. Toto lůžko má nastavitelnou výšku ložné plochy. U tohoto lůžka lze ložnou plochu vertikalizovat až do úhlu 90°. Lůžko ErigoPro je již vybaveno robotickým systémem, kterým je řízen pohyb dolních končetin. Noha pacienta je upevněna pomocí popruhů ke stupátku v oblasti kotníku. Dolní končetina je připevněna k pohonným jednotkám v oblasti stehna pomocí manžet. Pohyb dolní končetiny je tedy vyvolán působením pohonné jednotky ve stehenní oblasti dolní končetiny, čímž je aktivně vyvolána flexe a extenze v

kyčelním kloubu. Pohyb v kolenním kloubu, v hlezenním kloubu a klouben nohy je pasivní. Léčbu na tomto lůžku lze doplnit systémem řízené funkční elektrostimulace. Celá léčebná terapie je ovládána pomocí ovládacího panelu s intuitivním uživatelským rozhraním.

Druhým speciálním lůžkem je lůžko BTS Anymov od firmy BTS Bioengineering [2]. Oproti lůžku ErigoPro má toto lůžko plnohodnotnou ložnou plochu, složenou z několika segmentů, která je taktéž výškově nastavitelná. Hlavový a hrudní díl lze polohovat až do úhlu 75°. Segment pro dolní končetiny je polohovatelný v rozmezí úhlu 0°-35°. Ložnou plochu lze nastavit do Trendelenburgerovy polohy a dále ji lze polohovat laterálními náklony. Lůžko je vybaveno robotickými ortézami, které řídí pohyb dolních končetin. Simulaci chůze pomocí robotických ortéz lze provádět taktéž při vertikalizované ložné ploše. Stejně jako u lůžka ErigoPro může být i lůžko BTS Anymov vybaveno modulem zajišťující řízenou elektrostimulaci. Ovládání lůžka je zajištěno dotykovým ovládacím panelem.

Další skupinu tvoří rehabilitační roboty, kteří se využívají k postupnému zatěžování pohybovému aparátu a cvičení chůze pacienta. Jako příklad rehabilitačního robota lze uvést Lokomat od firmy Hocoma [3]. Pacient je zavěšen do nosného postroje, kterým je nastavována zátěž na pohybový aparát a pacient cvičí chůzi na pohyblivé ploše [4].

Poslední skupinu exoskelety. Jejich využití je u pacientů, kteří mají oslabený pohybový aparát. Exoskelet pacientovi dopomáhá při chůzi a pacient se může lépe pohybovat po prostoru [5],[6].

Z popsaných speciálních rehabilitačních lůžek, rehabilitačních robotů a exoskeletů je patrné, že se konstrukce robotického systému liší. Speciální rehabilitační lůžka jsou vybavena pouze robotickými ortézami. Rehabilitační roboty jsou tvořeny závěsným systémem pro pacienta, robotickými ortézami a pohyblivou plošinou. Exoskelety jsou robotické ortézy s vlastním zdrojem energie a jsou určeny pro pacienty, kteří se mohou pohybovat mimo lůžko. Já jsem se zaměřil na speciální rehabilitační lůžka jsou navržena za účelem simulace chůze, čehož využívá léčebná rehabilitační metoda proprioceptivní neuromuskulární facilitace. Lůžka však provádí nepřesný pohyb dolních končetin při chůzi, neboť tyto mechanismy nevykonávají rotaci v kyčelním kloubu. Cílem tohoto příspěvku je navržení mechanismu, splňujícího požadavky metody proprioceptivní neuromuskulární facilitace a tím jsou rotace v dolních končetinách.

## 2 Léčebná rehabilitace pacientů

S pacienty majícími funkční poruchy pohybového aparátu se setkávají lékaři mnoha oborů. Jedná se zejména o obory neurologie, ortopedie, neurochirurgie a fyzioterapie. Při léčebné rehabilitaci pacientů se provádějí různé léčebné rehabilitační metody. Z těchto metod lze jmenovat například Vojtova metoda, metodika senzomotorické stimulace a metoda proprioceptivní neuromuskulární facilitace.

Vojtova metoda je diagnostický a terapeutický systém [7]. V určitých výchozích polohách se v přesně vymezených oblastech těla provádí manuální aplikace tlaku na takzvané spoušťové zóny sloužící k vyvolání automatických lokomočních pohybů označených jako reflexní plazení a otáčení. Reflexní lokomoci lze aktivovat ze tří základních poloh: poloha vleže na břiše, na zádech a v kleče.

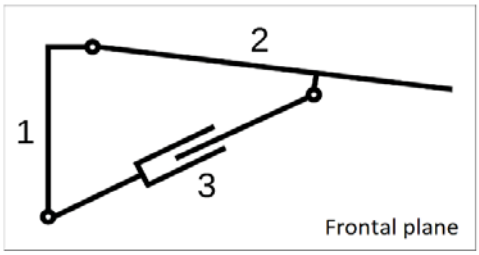
Metodika senzomotorické stimulace byla nejprve využívána pro terapii nestabilního kolene a kotníku. Později se začala používat při terapii funkčních poruch pohybového aparátu. Technika zahrnuje soustavu balančních cviků prováděných v různých posturálních polohách. Cviky prováděné ve vertikále jsou z celé metodiky nejdůležitější. V metodice se klade důraz na facilitaci pohybu chodidla. Metodika pracuje se dvoustupňovým modelem motorického učení [8].

Metoda proprioceptivní neuromuskulární facilitace (PNF) vychází ze základního neurofyziologického mechanismu. Základním pohybovým kamenem metody PNF jsou pohybové vzorce, které jsou vedeny diagonálním směrem vždy se současnou rotací. Pro každou část těla jsou určené dvě diagonály. Každá diagonála je tvořena dvěma vzorci, který má každý hlavní flekční nebo extenční komponentu. Pohyby ve směru těchto úhlopříček obsahují vždy tři pohybové složky v různých kombinacích: flexi nebo extenzi, abdukci nebo addukci, zevní nebo vnitřní rotaci [9],[10].

Využití robotického řízení pohybu u Vojtovy metody není vhodné, neboť tato metoda je prováděna na rehabilitačním lůžku nazývaném Vojtův stůl. Velikost ložné plochy tohoto lůžka je větší vzhledem k polohování pacienta. Fyzioterapeut zapolohuje pacienta a následně působí tlakem na spoušťové zóny až do chvíle, kdy uvidí reakci pacienta. Pomocí robotického zařízení by bylo náročné už jen samotné uvedení pacienta do jedné z poloh. Dále robotické působením tlakem na spoušťové zóny a následné pozorování reakce je zcela nevhodné. Robotické řízení je možné využít v metodice senzomotorické stimulaci dolní končetiny, kdy lze robotickým řízením nahradit pohyby, které jsou umožňovány na kruhovém nebo válcovém segmentu. PNF se vyznačuje složitým pohybem, neboť v rámci jedné diagonály dochází k pohybu ve vice kloubech současně. V současné době jsou již vyráběna lůžka a rehabilitační zařízení využívající metodu PNF, ale jejich roboticky řešený pohyb je nepřesný. Cílem článku je tedy uvést takové konstrukční řešení robotického systému, které odstraňuje nedostatky v léčebné metodě PNF.

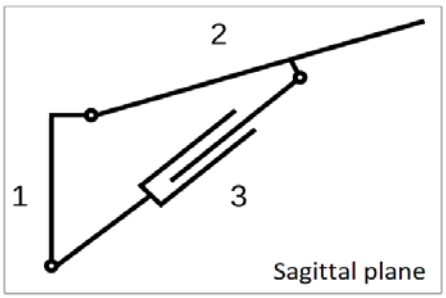
## 3 Řešení

Konstrukční řešení robotického systému, které zde bude uvedeno, je určeno pro speciální rehabilitační lůžka. První část robotického systému se nachází v oblasti kyčelního kloubu. Kyčelní kloub umožňuje abdukci a addukci, flexi a extenzi, vnější a vnitřní rotaci. Mechanismus provádějící abdukci a addukci (Obr. 1). Tento mechanismus je tvořen rámem lůžka (pozice 1), otočným ramenem (pozice 2) a lineárním aktuátorem (pozice 3). Změnou délky lineárního aktuátoru je prováděna abdukce a addukce dolní končetiny.



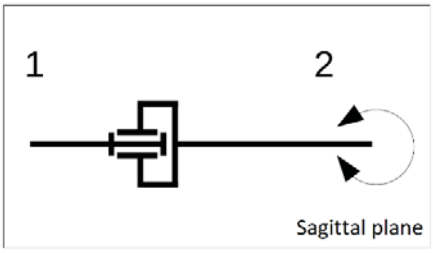

**Obr. 1.** Mechanismus abdukce a addukce kyčelního kloubu pravé dolní končetiny

Mechanismus vykonávající flexi a extenzi kyčelního kloubu je tvořen konzolou (pozice 1) upevněnou k otočnému ramenu mechanismu provádějícího abdukci a addukci dolní končetiny, držáku (pozice 2) a lineárním aktuátorem (pozice 3), který změnou své délky zajistí flexi či extenzi kyčelního kloubu (Obr. 2). Lineární aktuátory k vyvolání pohybu flexe-extenze či abdukce-addukce používají lůžka ErigoPro [1] a BTS Anymov [2].



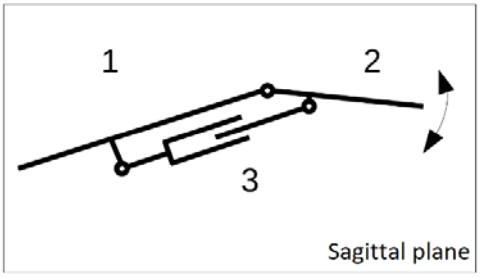

**Obr. 2.** Mechanismus flexe a extenze kyčelního kloubu dolní končetiny

Poslední možný pohyb v kyčelním kloubu je vnější a vnitřní rotace. Tento pohyb je velmi často opomíjen, i když rotace je velmi důležitým pohybem v metodě PNF. Konstrukce mechanismu pro rotaci dolní končetiny je však složitější s ohledem na uspořádání a rozdělení mechanismu v oblasti stehna dolní končetiny (Obr. 3). Další komplikací při řešení rotace dolní končetiny je skutečnost, že ve virtuální ose rotace se nachází dolní končetina pacienta. Mechanismus tvoří držák (pozice 1, Obr.2 – pozice 2), na kterém je upevněn krokový motor či servomotor. Konec hřídele je osazen pastorkem (ozubeným kolem), který je v záběru s ozubením na obloukovém vedení. Obloukové vedení je upevněno k sestavě obsahující podložku stehna dolní končetiny. Spojení držáku a sestavy s podložky stehna je dosaženo pomocí kladek, které zajišťují polohu obloukového vedení a jeho valivý pohyb.




**Obr. 3.** Mechanismus vnější a vnitřní rotace kyčelního kloubu dolní končetiny

Druhým kloubem, ve kterém je robotickým systémem prováděn pohyb, je koleno. Základním pohybem kolenního kloubu je flexe a extenze. Kolenní kloub však poskytuje i jistou možnost rotace, ale jen při současné flexi. Robotický systém (Obr. 4) pro kolenní kloub je tvořen sestavou pro oblast stehna (pozice 1), sestavou pro oblast bérce (pozice 2) a lineárním aktuátorem (pozice 3). Při změně délky lineárního aktuátoru je prováděna flexe či extenze kolenního kloubu.




**Obr. 4.** Mechanismus flexe a extenze kolene dolní končetiny

Pohyb v hlezenním kloubu a kloubech nohy je poměrně složitý. Vykonávané pohyby jsou flexe a extenze, pronace a supinace, everze a inverse. Mechanismus vykonávající flexi a extenzi je tvořen sestavou pro oblast bérce sestavou pro oblast nohy a lineárním aktuátorem. Pohyby everze, inverze, pronace a supinace jsou vytvořeny pomocí dvou krokových motorů nebo serv a dvou obloukových vedení s ozubením. Pohonné jednotky a přívodní kabely jsou umístěny na vnější a spodní straně robotických ortéz. Bezpečnost pacienta je zajištěna mechanicky i elektronicky.

## 4 Diskuze

Uvedené mechanismy lze dále modifikovat. Například polohu obloukového vedení a polohu krokových motorů či serv jde vzájemně zaměnit. Dále lze lineární aktuátory zajišťující flexi a extenzi kyčelního kloubu, flexi a extenzi kolenního kloubu, flexi a extenzi hlezenního kloubu zaměnit za krokové motory či servomotory doplněných převodovkou. Zatížení lineárních aktuátorů vychází z polohy uchycení lineárních aktuátorů, hmotnosti mechanismu robotického systému a hmotnosti pacienta.

Jednotlivé léčebné rehabilitační metody se od sebe vzájemně liší prováděným pohybem a využitím kloubního rozsahu. Z tohoto důvodu je nutná znalost pohybů v anatomických rovinách a rozsahů jednotlivých kloubů [11], [12]. Každý pacient má jiný rozsah kloubů, a proto je nutné před zahájením léčebné rehabilitační metody provést vyšetření kloubních rozsahů lékařem či fyzioterapeutem.

## 5 Závěr

V tomto článku jsou uvedeny různé typy lůžek a robotických zařízení používaných ve zdravotnictví a zejména v léčebné rehabilitaci. U současných konstrukcí bylo zjištěno, že uvedené speciální rehabilitační lůžka a rehabilitační roboty neprovádí rotaci dolních končetin, která je nezbytná při rehabilitační metodě PNF. V článku jsem navrhl konstrukční řešení, které pomocí robotického systému aktivně ovlivňuje pohyb v kloubech dolní končetiny včetně velmi důležité rotace v kyčelním kloubu. Navržený mechanismus dále poskytuje možnost prostorového pohybu dolních končetin, čímž lze realizovat diagonální pohyby dolních končetin dle metody PNF.

### Poděkování

Práce byla podpořena grantem SGS17/176/OHK2/3T/12

## Literatura

[1] Hocoma Erigo, Hocoma. [online]. [cit. 22.5.2018]. Dostupné z: https://www.hocoma.com/solutions/erigo.
[2] BTS Anymov, BTS Bioengineering. [online]. [cit. 22.5.2018]. Dostupné z: http://www.btsbioengineering.com/bts-anymov.
[3] Hocoma Lokomat, Hocoma. [online]. [cit. 22.5.2018]. Dostupné z: https://www.hocoma.com/solutions/lokomat.
[4] P.-Y. Cheng, P.-Y. Lai, Comparison of Exoskeleton Robots and End-Effector Robots on Training Methods and Gait Biomechanics, In: Intelligent Robotics and Applications, Springer Berlin Heidelberg, 2013, pp. 258-266.

[5] Ferris, D-P.: The exoskeletons are here. Journal of Neuroengineering and Rehabilitation 6(17), 1-3 (2009).
[6] Diaz, I.: Lower-Limb Robotic Rehabilitation: Literature Review and Challenges. Journal of Robotics 2011(2011), 1-11 (2011).
[7] Gajewska, E.: An attempt to explain the Vojta therapy mechanism of action using the surface polyelectromyography in healthy subjects: A pilot study. Journal of Bodywork and Movement Therapies, (2017).
[8] Senzomotorika, Medicina Ronnie. [online]. [cit. 22.5.2018]. Dostupné z: http://medicina.ronnie.cz/c-3839-senzomotorika-ii-uvod-zaklady.html.
[9] Surburg, P.R.: Proprioceptive Neuromuscular Facilitation Techniques in Sports Medicine: A Reassessment. Journal of Athletic Training 32(1), 34-39 (1997).
[10] Choi, Y-K.: The Effects of Taping Prior to PNF Treatment on Lower Extremity Proprioception of Hemiplegic Patients. Journal of Physical Therapy Science 25(9), 1119-1122 (2013).
[11] Čihák, R., Anatomie, Grada, 2011, ISBN 978-80-247-3817-8.
[12] Kolář, P et al.: Rehabilitace v klinické praxi, Galen, 2009, ISBN 978-80-7262-657-1.

# SYSTEM PLATFORM 2017 A INTOUCH OMI

***Lukáš Žídek***
***Fundel@seznam.cz***

*Abstrakt: Tato práce se zabývá softwarem pro tvorbu vizualizace Systém Platform. Především se jedná o předvedení nových možnosti v licencování tohoto softwaru. Následuje praktická část, která se zabývá tvorbou vizualizačních obrazovek pro Pražskou teplárenskou.*
*Klíčová slova: SCADA, HMI, Wonderware, System Platform. PTAS*

*Abstract: This thesis deals with System Platform visualization software. This is primarily about demonstrating new options in licensing this software. The following is a practical part, which deals with the creation of visualization screens for Prague Heat Plant.*
*Keywords: SCADA, HMI, Wonderware, System Platform, PTAS*

## 1 Úvod

Firma Pantek s.r.o. byla založena v roce 1993 a je dceřinou firmou nezávislé britské společnosti Pantek Ltd., která se specializuje na dodávku a technickou podporu především softwarových produktů pro průmyslovou automatizaci. Hlavní předností firmy Pantek je komplexní podpora softwaru od firmy Wonderware, která je součástí koncernu Schneider Electric.

## 2 Systém Platform 2017

Systém Platform 2017 je nová generace softwaru pro průmyslové automatizační a informační aplikace typu SCADA a HMI. Zkratka SCADA je složena ze slov Supervisory Control And Data Acquisition. Zkratka HMI představuje složeninu ze slov Human-Machine Interface.

Nová verze Systém Platform 2017 přináší zásadní modernizaci a mnoho vylepšení. Především se jedná o zefektivnění vývoje aplikací, úsporu času pro jejich vývoj a ulehčení používání v runtime režimu.

### 2.1 Novinky v Systém Platform 2017

Nová verze softwaru nyní podporuje práci s rozlišením Ultra 4k. Toto rozlišení lze použít pro všechna zařízení, která mohou s novým softwarem komunikovat. Jedná se především o tablety, chytré telefony ale samozřejmě i o standardní monitory. Jelikož systém podporuje 4k rozlišení, je možné zobrazovat v nejrůznějších kombinacích rozlišeních. S tímto spojená je také podpora vícemonitorového zobrazení s možností kombinace různých rozlišení v rámci jednoho projektu.

Funkčnost Objekt Wizard pro vytváření konfigurovatelných šablon vyspělých objektů Aplication Serveru, což znamená efektivní propojení funkčních objektů s grafickým zobrazením.

Visual Build je nový způsob vytváření projektu na základě grafiky, který využívá průvodce vytvářením objektů z předem vytvořených šablon. Layouts jsou předpřipravené typy rozvržení zobrazení a Screen Profiles, zobrazení profilů zobrazovacích zařízení.

Další novinky se týkají zobrazení vizualizace a práce sní. Je možné například zvětšení nebo zmenšení (zoom)

grafického zobrazení. Pro jednotlivé grafické objekty v symbolech lze nastavit podmínky pro viditelnost a úroveň detailů v závislosti na aktuálním zoomu. Lze tedy zobrazovat objekty od přehledných náhledů po detailnější informace. Toto zobrazení lze ovládat dotykovými gesty. Lze použít jak jeden prst tak více prstů pro různé funkce gest.

Další důležitou novinkou je možnost automatické tvorby navigace v klientské aplikaci dle modelu aplikačního serveru.

Linked Symbol umožňuje jednoduché provázání symbolů z aplikačního serveru na existující ArchestrA symboly.

Nový je také licenční systém, na který se blíže podíváme později v tomto článku.

Nový software je také mnohem méně náročný na výkon Hardwaru, na kterém je provozován.

V neposlední řadě je přidána možnost vestavěných aplikací jako jsou například mapy a nebo webový prohlížeč.

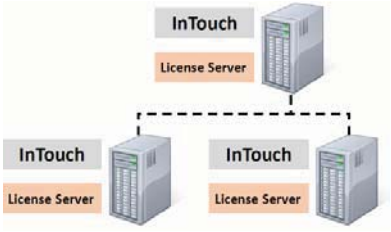
### 2.2 Nový licenční systém

Licencování nového systému probíhá za pomoci licenčních serverů. Není tedy potřeba jako v minulosti hardwarových klíčů.

Nově je tedy nutné mít v síti zapojen a nainstalován minimálně jeden licenční server (License Server). Tento server zajišťuje nepřetržitý přístup k licenci pro všechny zařízení v síti ve které je zapojen. Na tento server je možné aktivovat licence pro všechen software pro Systém Platform 2017. Server jako takový nepotřebuje žádnou licenci. Je tedy možné, aby každá síť měla zapojených najednou více licenčních serverů najednou. Toto řešení se zdá velmi vhodné, protože se vytváří redundantní řešení pro licence softwaru. Pokud tedy jeden ze serverů přestane fungovat, ať už z důvodu pádu systému nebo jiných, a nebo se odpojí od sítě, je možné i nadále používat licence ze serveru druhého (redundantního). Tímto může nastat několik možných scénářů zapojení, které si rozebereme v další části. [1]

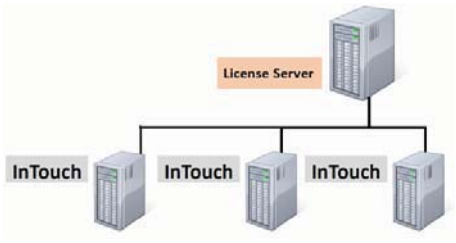
### 2.3 Možná zapojení licenčních serverů

Jedna z možností je nainstalovat licenční server na každý počítač v síti, který využívá licencí Wonderware viz Obrázek 1. Licenční server na všech zařízeních. Velikou výhodou tohoto zapojení je to, že nemůže dojít ke ztrátě licencí při výpadku serverů a nebo sítě jako takové. Každý počítač je sám sobě licenčním serverem a je to tedy velmi bezpečné řešení. Na druhou stranu je při tomto řešení nezbytná instalace a zpráva systémů na každém zařízením a tím se zvyšují nároky na obsluhu a složitost instalace zařízení. Tento způsob instalace je také málo přehledný. [1]



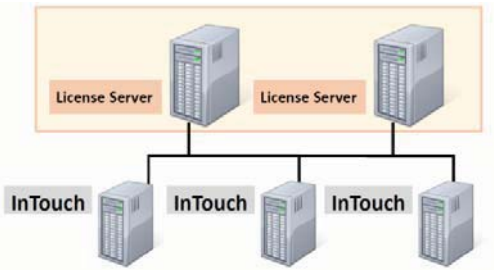

Obrázek 1. Licenční server na všech zařízeních

Druhá možnost je nainstalovat licenční server pouze na jedno zařízení, které je připojeno v síti k dalším zařízením viz Obrázek 2. Jeden licenční server. Tento způsob licencování je asi nejjednodušší možností, ale také nejnáchylnější k různým výpadkům. Pokud například přestane fungovat síť mezi serverem a jednotlivými klienty InTouch, licence přestanou fungovat a tím je znemožněna práce se softwarem. Podobný problém nastane při výpadku licenčního serveru, při jeho odstávce nebo aktualizaci a podobně. [1]



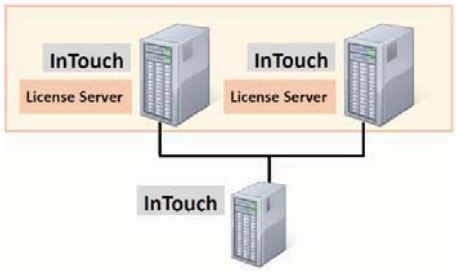

Obrázek 2. Jeden licenční server

Třetí možností licencování je mít dva servery, na kterých jsou nainstalované všechny licence viz Obrázek 3. Redundantní řešení licencování. Vznikne tedy redundantní řešení, které zajišťuje velmi spolehlivé řešení. Pokud máme dva licenční servery v redundanci a jeden ze serverů přestane z jakýchkoliv důvodů fungovat. Jeho práci převezme druhý z licenčních serverů a práce se softwarem je tedy nepřerušena. Po opětovném zprovoznění nefunkčního serveru se vrací Redundantní provoz. Tento způsob licencování je poměrně přehledný a je také snadná instalace i správa licencí. Nevýhoda tohoto licencování je ta, že není vyřešen problém výpadku sítě s licenčními servery. Pokud vypadne síť je znemožněno pracovat se softwarem InTouch. [1]




Obrázek 3. Redundantní řešení licencování

Další možnost licencování vychází z předchozí možnosti. Jde o možnost používat licenční server i jako klienta InTouch viz Obrázek 4. Redundance s klienty. Tato možnost využívá pro práci i Redundantní servery, což šetří náklady na pořízení dalších počítačů. Toto schéma má stejné výhody jako předchozí, ale přidává jednu nevýhodu. Nevýhoda spočívá v tom, že na tomto licenčním serveru může pracovat obsluha a tím pádem se zvyšuje riziko, že bude zařízení nějakým způsobem vypnuto, restartování, odpojeno ze sítě a podobně. Jelikož je licence serverů redundantní, dá se předpokládat že druhý server bude třeba při restartu prvního zařízení plně funkční a funkce jednotlivých licencí nebude narušena. Je ale potřeba poučit obsluhu o možnosti ztráty redundance. [1]



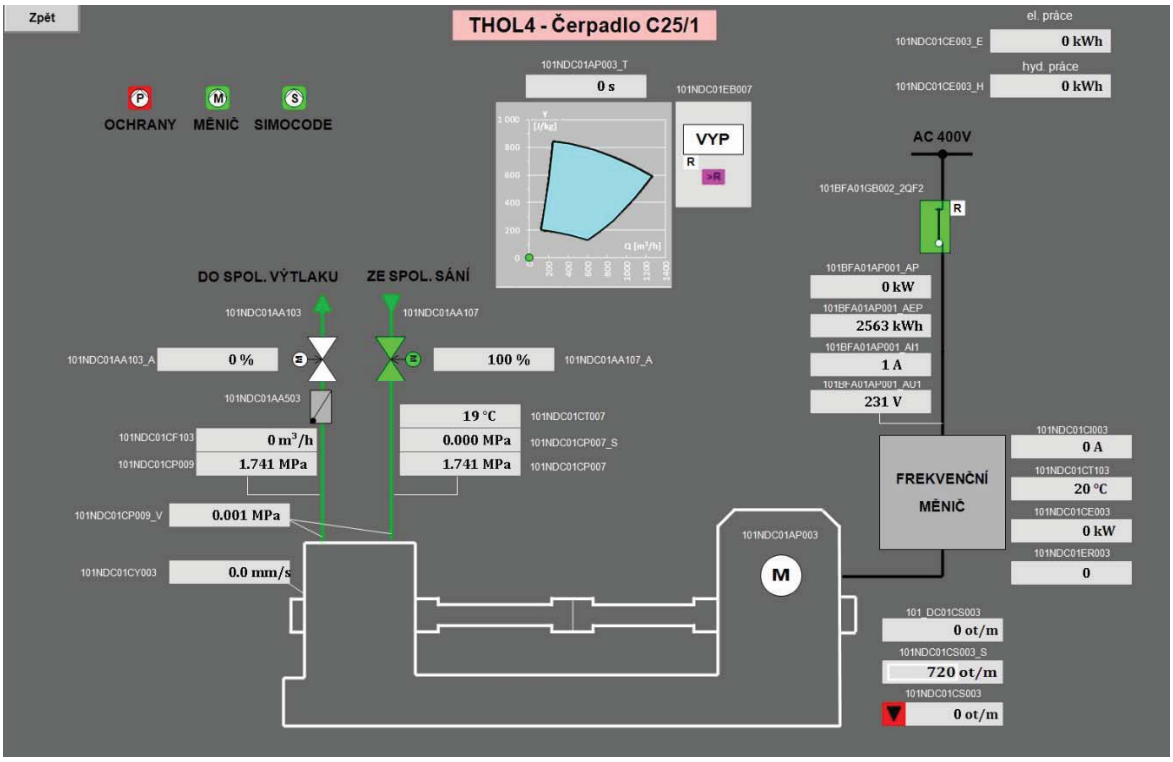

Obrázek 4. Redundance s klienty

## 3 Nově vytvořená obrazovka čerpadla PTS

Požadavkem bylo vytvořit novou obrazovku pro teplárnu v Holešovicích. Předmětem byla čtyři totožná čerpadla, která jsou řízena frekvenčním měničem.

Prvním krokem bylo zhotovení pozadí, které bude pro všechny čtyři čerpadla totožné, aby bylo rozdělení jednotlivých hodnot, symbolů, stejné. Po vyhotovení pozadí se ve vývojovém prostředí ArchestrA vytvořila obrazovka, na kterou bylo vloženo nové pozadí. Dalším krokem bylo nakreslit napájení, které je vyznačeno černou čárou a vede přes blok s měničem až k motoru čerpadla. Zelenou čarou a šipkami je vyznačen parovod, ve kterém se pára čerpá. Bíle je vyznačena silueta, která představuje čerpadlo. V levé části je poháněná turbína čerpadla, která je

přes hřídel připojena k motoru vpravo. Dále byly vytvářeny a vkládány všechny symboly, grafy a signalizace. Jelikož bylo šedé pozadí vytvořeno předem a takovým způsobem, aby bylo zřejmé kde má daný symbol být umístěný, nedocházelo ke špatnému umístění symbolů. Kvalitně vytvořeným pozadím bylo usnadněno zajištění toho, že všechny čtyři obrazovky byly naprosto stejné. Na Obrázku 5. vidíme obrazovku čerpadla C25/1 napojenou na síť.




Obrázek 5. Obrazovka čerpadla C25/1

U každého symbolu, který zobrazuje analogové hodnoty je vidět KKS kód. KKS kód představuje systém jednotného značení energetických výroben a zařízení. Jedná se tedy o značení zařízení a je složen ze čtyř stupňů.

## 4 Závěr

Cílem mé práce bylo seznámit se se softwarem pro tvorbu vizualizace Systém Platform v jeho verzích 2014 a 2017 včetně jeho licencování. Za dobu práce s tímto softwarem jsem se naučil spoustu věcí, bez kterých by nebylo možné vytvořit funkční vizualizaci. Výsledkem mé práce jsou čtyři plně funkční obrazovky čerpadel teplárny v Praze Holešovicích, které jsou využívány na velíně v Třeboradicích.

**Poděkování**

Rád bych poděkoval svému kolegovi Ing. Mirku Přibylovi za pomoc při práci, především za poskytnutí cenných rad ohledně celého systému Pražské teplárenské soustavy a tvorby vizualizace v System Platform.

## Literatura

[1] PICEK, Ivan, 2017. Wonderware software – Nové licencování s licenčním serverem (od verze 2017). Dostupné z: http://www.pantek.cz/produkty/wonderware-system-platform/vykonne-sluzby.html#login-box (cit. 31.5.2018)
[2] Energoservis, 2018. KKS-KOD. Dostupné z: http://kkskod.cz/kod-kks/

# GENETICKÉ ALGORITMY S GENOMEM TVOŘENÝM REÁLNÝMI ČÍSLY (REAL-CODED GENETIC ALGORITHMS)

*Vladimír Hlaváč*[1]

[1] *FS ČVUT v Praze, hlavac@fs.cvut.cz*

*Abstrakt: Řada problémů vede na hledání kombinace reálných čísel, například parametrů modelu nebo hodnot popisujících vhodné nastavení. Tyto jednotlivé proměnné tvoří n-rozměrný prostor, ve kterém hledáme řešení. Schopnost hledat řešení se obvykle demonstruje na nejvýše třech hledaných hodnotách a na dobře podmíněném problému, který neobsahuje příliš mnoho lokálních extrémů. Právě v případech, kdy tyto dvě vlastnosti nejsou splněny, mohou dobře posloužit evoluční nebo hejnové algoritmy, které jsou v zásadě výpočetně náročnější, ale neselhávají ani za velmi složitých podmínek. Článek popisuje především nejzákladnější z nich, genetické algoritmy s reálným genomem a metodou křížení BLX-alfa a jeho modifikací, inteligentním křížením, a jako příklad hejnového algoritmu pak SOMA.*
*Klíčová slova: Genetické algoritmy, Křížení, BLX-α, Diverzita, .*

*Abstract: There are many problems defined as a searching for a combination of real numbers, for example model parameters or searching for setting of parameters. Those real numbers create n dimensional search space. The ability for finding a solution is typically demonstrated on at most three searched values and well defined problem, without too many local optima. Just in the other cases, evolutionary algorithms or the swarm intelligence algorithms can be successfully used. Those methods are more time demanding, but can work in very complicated conditions. The article describes the most fundamental of them, real valued genetic algorithms, BLX-α, and its modification, known as the intelligent crossover, and the SOMA algorithms as a swarm particle algorithm example.*
*Keywords: Genetic algorithms, Crossover, BLX-α, Diversity.*

## 1 Úvod

Genetické algoritmy, nebo obecně evoluční výpočetní techniky [1], používají pro hledání řešení udržování populace tzv. kandidátních řešení. Již známá řešení se vhodným způsobem kombinují (vytvářejí se jejich potomci) a naopak, nejhorší řešení jsou z populace vyřazována. Terminologický rozdíl oproti hejnovým algoritmům je v tom, že u hejnových algoritmů se jednotlivá řešení stále upravují podle úspěšnoti „okolních“ řešení (v obou případech si řešení lze představit jako bod v mnoharozměrném prostoru). V obou případech se kvalita řešení posuzuje podle vhodně definované účelové funkce.

Velikost populace je tedy u hejnových algoritmů zpravidla stálá, zatímco u genetických se může měnit. U genetických algoritmů lze ovšem realizovat evoluci použitím tzv. turnajové selekce, která také trvale zachovává velikost populace. Typické řešení turnajové selekce (podle [2], str. 137) je výběr dvou dvojic jedinců z populace. Z každé dvojice se vybere lepší jedinec, horší se z populace vyřadí. Místo vyřazených jedinců se umístí dva potomci, vytvoření křížením nevyřazených jedinců. Ti v populaci také zůstávají (celkový počet jedinců je tedy zachován). Oproti běžnější ruletové selekci není třeba populaci řadit a ušetří se tedy často velmi náročné porovnání. Turnajová selekce ale neumožňuje řídit evoluční tlak a tím genetické algoritmy ztrácí hlavní výhodu oproti hejnovým algoritmům.

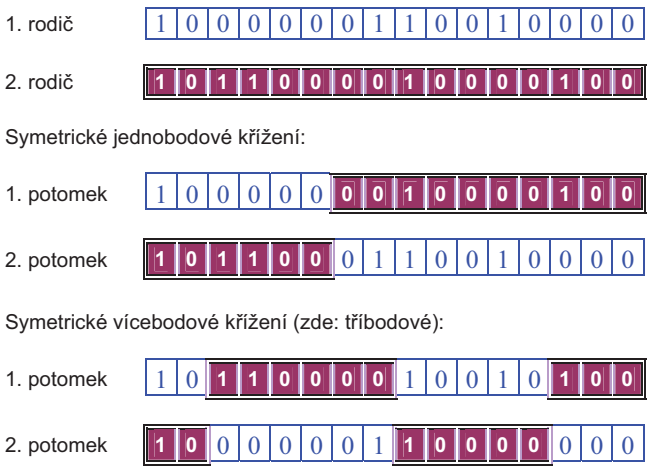
Rozdílný přístup vychází z použití ruletové selekce. Zde je v paměti pro hledané jedince vyhrazen podstatně větší počet míst, než kolik je jedinců v základní populaci. Pro vytváření nového jedince se používá křížení nebo některý typ tzv. mutace. Po vytvoření nových jedinců do volných míst je populace opět seřazena podle hodnoty

účelové funkce a v populaci je ponecháno opět jen tolik jedinců, kolik tvoří základní populaci. U ruletové selekce jsou navíc pro křížení vybíráni s vyšší pravděpodobností jedinci s lepší hodnotou účelové funkce. Pravděpodobnost výběru může obsahovat i samotnou hodnotu účelové funkce. Použití pouze pořadí v populaci ovšem dává lepší výsledky. Pokud je použito exponenciální rozložení pravděpodobnosti výběru jedince pro křížení, lze evoluční tlak definovat jako poměr pravděpodobnosti výběru dvou po sobě jdoucích jedinců v populaci. Při praktických testech vychází jako vhodný poměr čísla v rozsahu 1,01 až 1,001, v závislosti na velikosti základní populace. Podrobněji viz [3], kapitola 4.

Hejnové algoritmy svým názvem vedou k představě algoritmu jako udržování množiny řešení, pro které se v každém kroku vždy vyčíslí účelová funkce a pak se jednotlivé body posunou ve směru k jedincům, kteří aktuálně dosahují lepších hodnot účelové funkce. Stejnou představu je třeba uplatnit i na genetické algoritmy. V obou případech se vytvoří jakýsi mrak řešení, který prohledává blízký prostor a posouvá se ve směru, kterým se řešení zlepšuje. Z této představy vyplývá také to nejdůležitější, co musíme při udržování populace hlídat. Pokud se hejno bezhlavě slétne k nejlepšímu řešení, máme algoritmus, který rychle a spolehlivě uvízne v prvním lokálním minimu účelové funkce. Pokud máme problém, kde nejsou lokální optima, pak tato situace nenastává, ale z druhé strany máme lepší (a řádově rychlejší) metody, jak najít jediné řešení, například gradientní a iterační metody. Proto je třeba udržovat velikost prohledávaného prostoru. Ostatně český pojem „hejno“ evokuje hejno ptáků, kde také jedinci nemohou být příliš blízko sebe. Anglický pojem „swarm optimization“ (swarm je „částice“) ovšem nic podobného nezahrnuje. U genetických algoritmů tomuto problému odpovídá pojem diverzita. Pokud si všechna řešení začnou být podobná, veškerý vývoj se zastaví. Tato situace se označuje jako zamrznutí populace. Algoritmus to zpravidla řeší tak, že se při poklesu diverzity pokusí použít větší množství mutací, aby obnovil vývoj genomu ([4], str.149). Většina mutací (náhodných změn v genomu) ovšem vede na tvorbu defektních jedinců. Pokud mají nějak ovlivnit populaci, musí se i tito jedinci použít pro křížení (to vede ke změně pořadí operací, kdy se mutace provádí před křížením a výslední mutanti jsou náhodně zařazováni do populace, zatímco ostatní jsou posouváni na její konec). Další ochranou proti zamrznutí populace je vyřazování jedinců, kteří jsou starší než stanovený počet cyklů (např. 100) [5]. Proti vyřazením a mutacím bývá chráněno dosud nejlepší nalezené řešení. Tento přístup se označuje jako elitismus.

Základní představa zápisu genomu byla prostá kombinace nul a jedniček (binární genom), kdy křížení reprezentovalo střídavé kopírování genomu z jednoho nebo druhého rodiče [6]. Typické je symetrické třídění, kdy nevyužité části genomu vytvoří druhého jedince. Binární genom nebere ohled na zakódované informace ani na jejich hranice. Například rodiče mají hodnotu nějakého parametru jeden 192 a druhý 189, tedy binárně 11000000b a 10111101b. Kombinací například po změně zdroje na 3. bitu vznikne 11111101b a 10000000b, tj. 253 a 128. Chybí zde tedy jasný vztah mezi rodiči a potomky.

Výhodou binárního genomu je, že jej mohou tvořit slepená binární čísla různé délky. Zápis celých, např. 32bitových čísel, by zde zbytečně zpomaloval evoluci. Pokud například hledáme den a měsíc, což je v rozsahu do 31 a do 12, vyhradíme v genomu jen 5, resp. 4 bity. V základní verzi genetických algoritmů křížení hranice jednotlivých znaků ignoruje. To z druhé strany umožňuje změnu významu částí genomu.




Obr. 1.: Ukázka symetrického křížení s binárním genomem [3]. Význam jednotlivých bitů často závisí na kontextu.

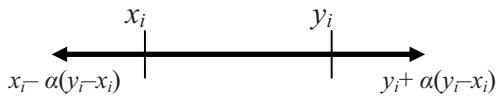
## 2 Hledání kombinace reálných čísel

### 2.1 BLX-α

Genetické algoritmy s genomem tvořeným reálnými čísly jsou definovány tak, že jednotlivé geny tvoří čísla (často v nějakém rozsahu). Křížení a mutace se od binárního genomu liší; použijí se číselné hodnoty a ty se kombinují. Nemůžeme vzít prostě průměrnou hodnotu obou čísel, to bychom ztratili diverzitu. O něco lepší přístup je vzít číslo z prvního rodiče s pravděpodobností $p_1$ a z druhého rodiče s pravděpodobností $p_2 = 1 - p_1$:

$$c_i = p_1 a_i + (1 - p_1) b_i \tag{1}$$

Index $i$ rozlišuje jednotlivé složky genomu. Popsáno již v [7]. I v tomto případě dochází k jistému omezení diverzity. Protože by po chvíli došlo ke zprůměrování všech hodnot a k zastavení vývoje, bylo navrženo následující opatření: Představme si, že jednotlivá čísla z genomů obou rodičů představují dva protilehlé vrcholy, definující v prostoru $n$-rozměrný kvádr. Za novou kombinaci čísel bude vzata taková $n$-rozměrná souřadnice, která leží kdekoli nejen v takto vymezeném kvádru, ale v kvádru, který má stejný střed a polohu, ale je (1+α)-krát ve všech dimenzích zvětšen. Protože bod je vybírán kdekoli v tomto prostoru, je metoda označována jako slepé křížení, blind crossover, s parametrem α, ve zkratce BLX-α [8]. Obr. 2 znázorňuje hodnotu souřadnice pro výchozí hodnoty $x_i$ a $y_i$.




Obr. 2.: Ukázka křížení jedinců X a Y pro jednotlivou souřadnici, kdy výsledná hodnota je kdekoli na dané úsečce.

Pro výpočet jednotlivých souřadnic v zásadě platí (1), ale parametr je třeba náhodně volit v rozsahu rozšířeném o hodnotu α:

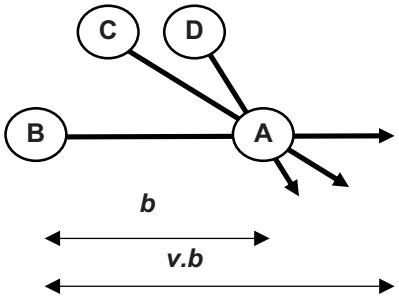
$$z_i = k_i x_i + (1 - k_i) y_i, k_i \in \langle -\alpha; 1 + \alpha \rangle. \tag{2}$$

### 2.2 Inteligentní křížení

U BLX-α se koeficient výběru pro každou souřadnici generuje náhodně znovu. Pokud použijeme pro všechny souřadnice stejnou hodnotu náhodného čísla, získáme tzv. inteligentní křížení (popsáno v [9]). Zde řešení leží na přímce, dané dvěma rodičovskými řešeními, ale jeho poloha může být i mimo jimi danou úsečku. Pro snadné srovnání můžeme míru o jakou může být řešení generováno mimo tuto úsečku označit opět α. Časté je ovšem nesymetrické rozšíření, například více do kladných hodnot než do záporných.

### 2.3 SOMA

Self-Organizing Migrating Algorithm (SOMA) [1] pracuje v podstatě podobně. SOMA je ovšem hejnový algoritmus, takže nemůže generovat nové členy populace. Místo toho mění jednotlivá řešení. Nejprve určí nejlepší řešení (vedoucí, leader) a pak všechna ostatní posune ve směru k tomuto řešení. Pokud postupujeme podle obdobného vzorce, jako je (2), můžeme řešení o jistý úsek naopak posunout ve směru od nejlepšího řešení. Tím je u této metody zajištěna diverzita. Lépe by ovšem bylo diverzitu zajišťovat v opačném směru, a interval by tedy měl být rozšířen nesymetricky (rozšíření nejprve zlepšuje úspěšnost metody při hledání řešení, vyšší hodnoty snižují stabilitu).




Obr. 3.: Schematické znázornění algoritmu SOMA [10]. Nejlepší řešení A se neposouvá, B, C a D se mění.

Existuje řada modifikací algoritmu SOMA. Program představený v [11] používá variantu, kdy jsou vybrány náhodně jen čtyři prvky populace, z nich se vybere nejlepší a ostatní tři jsou posunuty – náhodná vzdálenost je generována v zadaných mezích, osvědčily se nesymetrické až dvojnásobné ve směru nejlepšího řešení, s možností až 50% kroku směrem od nejlepšího řešení.

## 3 Data

Pro následující ukázky byla data generována stejnou funkcí, jako v [10]:

$$f(x, y) = k_1 \sin(k_2 x + k_3 y) \tag{3}$$

Konstanty byly opět nastaveny na hodnoty 3,71, 1,73 a poslední má hodnotu 1. Data byla vyčíslena v 555 bodech, pro hodnoty $x$ a $y$ náhodně generované v rozsahu $(-\pi; \pi)$. Generovaná data jsou opět zatížena malým šumem [10]. Účelová funkce byla vyčíslována jako součet čtverců odchylek v zadaných bodech.

## 4 Výsledky testování algoritmů

Následující grafy demonstrují základní problém, společný genetickým a hejnovým algoritmům, ztrátu diverzity. Z tohoto důvodu neznázorňují, jak se vyvíjí hledané řešení (které je reprezentováno nejlepším jedincem v populaci), ale nejmenší a největší hodnotu hledaného parametru v populaci. Úloha představuje hledání tří parametrů v datech, vygenerovaných podle funkce (3), z nichž první parametr zpravidla činí největší problémy. Proto je v následujících grafech zobrazována hodnota právě **jen** tohoto **prvního** parametru. Jak genetické, tak hejnové algoritmy představují jakýsi mrak řešení, který se pohybuje prostorem ve směru, kterým se nachází lepší hodnoty účelové funkce (zde menší součet odchylek ve všech zadávaných bodech). Pokud populace zcela ztratí diverzitu (hodnoty konstant u celé populace začnou být stejné), ztratí metoda schopnost zjistit tento směr, nebo dokonce vykonávat jakýkoli pohyb, protože míra tohoto kroku je dána jako násobek rozdílu hodnot parametrů u dvou různých řešení.

Grafy jsou s ohledem na velký počet křivek značně nepřehledné. Proto obrázek 4 ukazuje jen jeden konkrétní průběh na ukázku způsobu zobrazení. Vždy dvě křivky stejné barvy reprezentují maximální a minimální hodnotu hledaného parametru v populaci. Tenká červená čára, zvýrazněná označením bodů, pro které probíhá výpočet, znamená známé správné řešení problému. Program prezentovaný v [11] byl upraven tak, aby do log-souboru zaznamenával i nejmenší a největší prvek v populaci (jedná se o hodnotu parametru v genomu, číslo nemá nic společného s hodnotou účelové funkce).

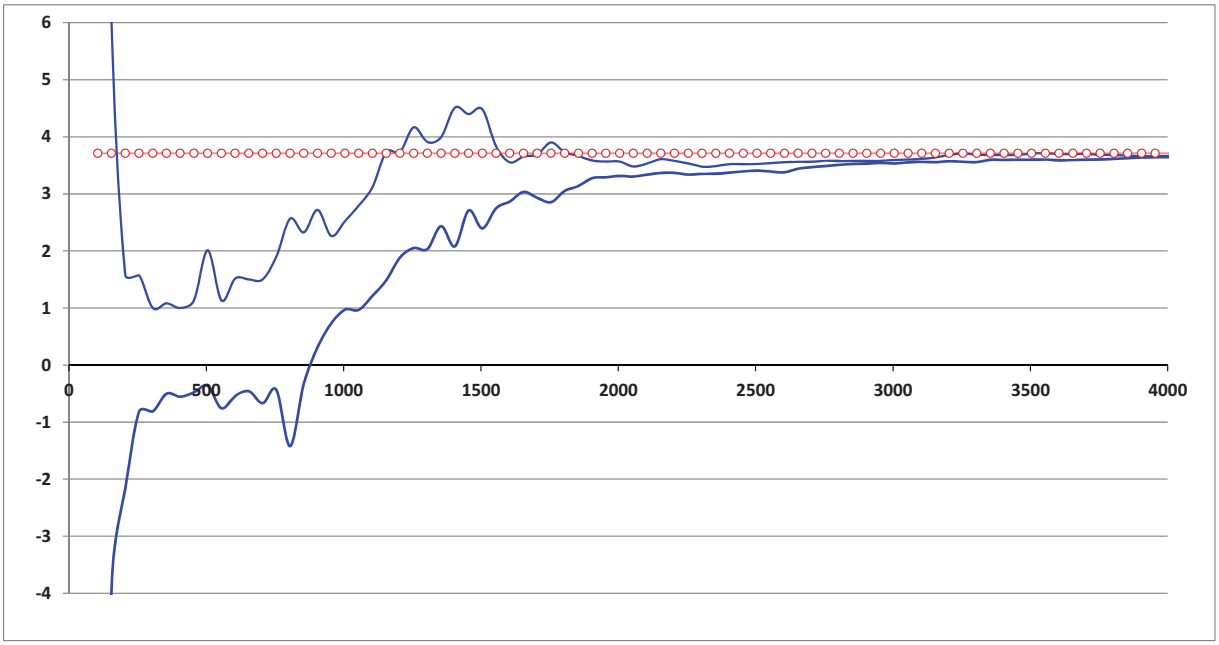
U grafů na obrázku 11 a 14 je doplněno i zobrazení nejlepšího řešení (podle hodnoty účelové funkce). Je to opět čára stejné barvy, ale na obrázku 11 je pro rozlišení tlustou čarou, na obrázku 14 bylo zvoleno různé provedení čáry (tečkovaně, čárkovaně), protože takové čáry jsou snáze viditelné, pokud se překrývají.

Právě graf na obrázku 4 nejlépe popisuje chování tohoto typu algoritmů. „Mrak řešení“ se nejprve zkoncentruje okolo zatím nejlepších řešení a pak se celý posouvá ve směru správného řešení. Když překročí nejlepší hodnotu účelové funkce, začne se okolo ní koncentrovat (rozdíly mezi jednotlivými hodnotami hledaných koeficientů se mezi různými řešeními začnou snižovat). Na konci evoluce mají všechna řešení přibližně stejnou hodnotu koeficientů a nejlepší řešení se vybere podle hodnoty účelové funkce.

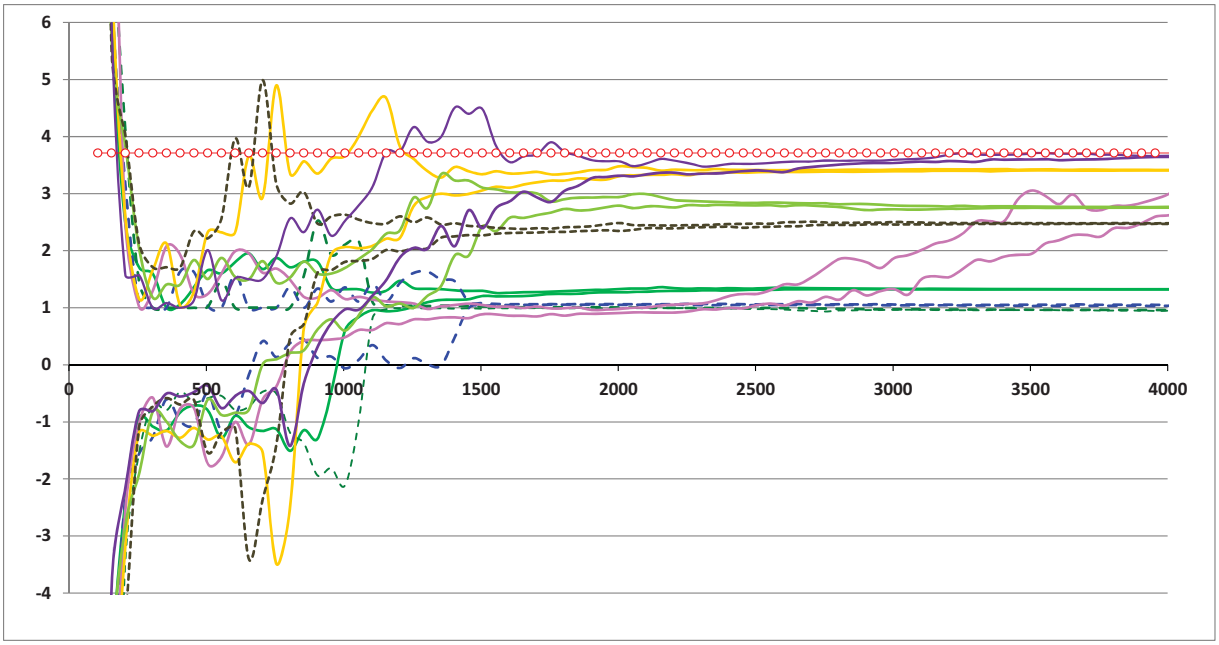
Grafy ve vodorovné ose nezačínají v nule. Je to tím, že jistý počet vyhodnocení účelové funkce proběhne hned při generování nové populace. Je to 100 vyhodnocení pro genetické algoritmy a 200 pro SOMA. Rozdíl je dán způsobem implementace těchto algoritmů a neměl by mít velký vliv na rychlost samotného výpočtu. V rámci testů bylo nastaveno 100 generací a komentáře u obrázků (nalezení správné hodnoty apod.) se vztahují k tomuto počtu generací. Zobrazený rozsah vodorovné osy (4000 vyhodnocení účelové funkce) ale odpovídá necelým 78 generacím u genetických algoritmů.

Aby byly výsledky srovnatelné, bylo vždy 50 jedinců v základní populaci. U genetických algoritmů bylo dalších 50 generováno v každé generaci (ať již křížením, nebo kombinací křížení a mutací). Pro každého nově vygenerovaného jedince je vždy ihned spočítána odpovídající hodnota účelové funkce. U genetických algoritmů jsou podle této účelové funkce seřazeni a pro další generaci je ponecháno jen 50 nejlepších. U SOMA místo toho proběhne 50 posunutí. SOMA nevyžaduje seřazení, ale pro účel zápisu do log-souboru jsou v programu nalezeni nejlepší jedinec a nejmenší a největší hodnoty prvního parametru, aby bylo možné vykreslit graf. Při praktickém použití metody lze v programu zápis do souboru a s tím spojené operace vypnout.

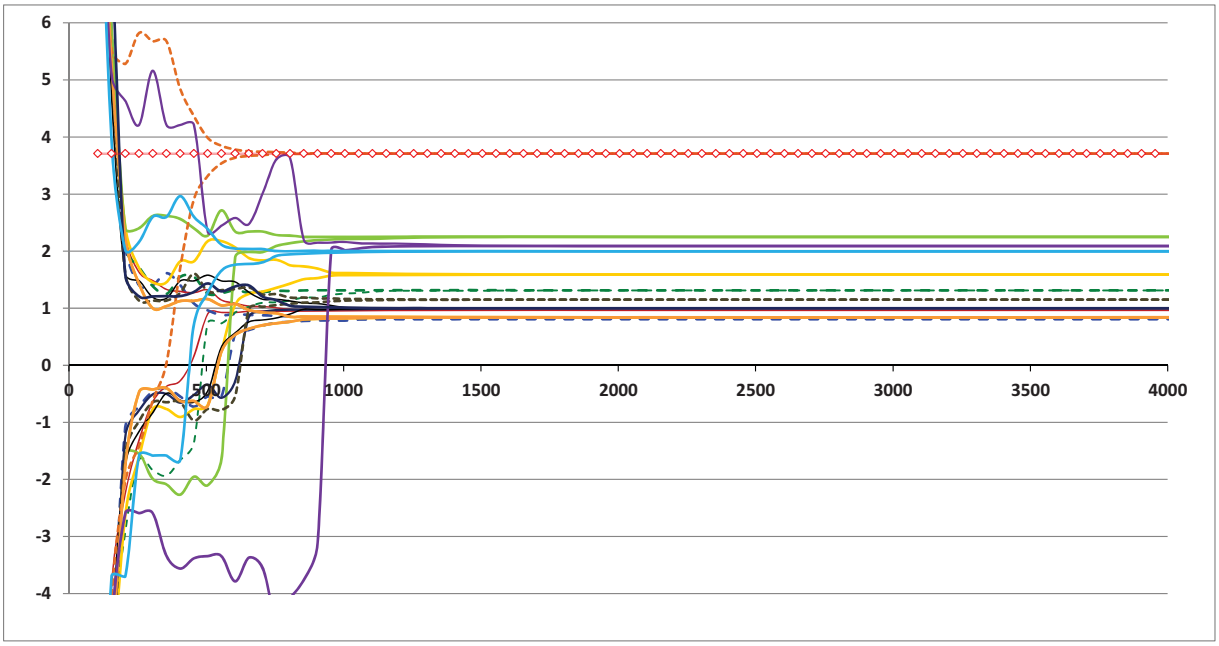
Každá metoda byla volána opakovaně a jednotlivé průběhy byly do grafů zpracovány programem MS Excel. Zde je nutné podotknout, že protože funkce sin je lichá, má úloha dvě možná řešení, kromě zadání {3,71, 1,73, 1,0} je to {–3,71, –1,73, –1,0}. V grafech by společné zobrazení vypadalo ještě více chaoticky. Proto byly průběhy, kde byly výsledné koeficienty nejlepšího jedince po sto generacích (konečné hodnoty) záporné ze zobrazení zcela vynechány. Za tento nesystematický přístup k datům se omlouvám a jsem přesvědčen, že není pro výsledné posouzení průběhů na závadu. Ve všech testovaných případech byly vždy všechny nalezené koeficienty u nejlepšího jedince stejného znaménka. Při snižování počtu grafů (z důvodu názornosti, viditelné na obr. 11, ale použité všude) byly mazány vždy poslední nahrané průběhy (a ponechané první), aby výsledky zachovaly alespoň částečnou reprezentativnost. Výjimkou je obr. 4, kde se jedná o třetí průběh.



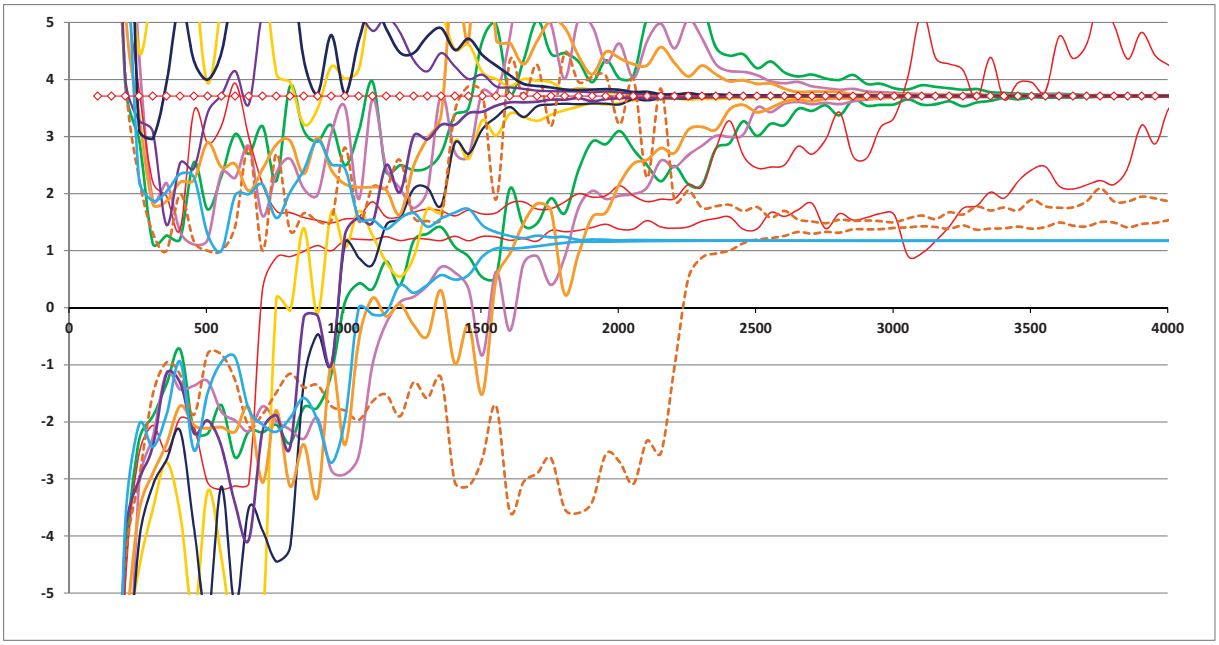

Obr. 4.: Ukázka konstrukce grafu pro jeden konkrétní průběh simulace s použitím inteligentního křížení (jedná se o jeden z grafů z obrázku 5, kde je fialovou barvou, nastavení viz obr. 5). Program hledá tři koeficienty, v grafech je zobrazena hodnota jen prvního z nich, se kterým mají popisované metody největší problémy. Vždy dvě křivky stejné barvy (resp. typu) reprezentují vždy největší a nejmenší hodnotu daného parametru v populaci v dané generaci. Červená konstanta zdůrazněná kroužky reprezentuje správnou hodnotu hledaného koeficientu (3,71), kroužky pak označují okamžiky, ke kterým jsou zobrazovány ostatní křivky. Vodorovná osa reprezentuje počet vyčíslení účelové funkce, což je nejlepší dostupné měřítko časové náročnosti metody. Pokud se ztratí diverzita, pak metoda podle obr. 2 již ztratí možnost pohybu v prostoru řešení.



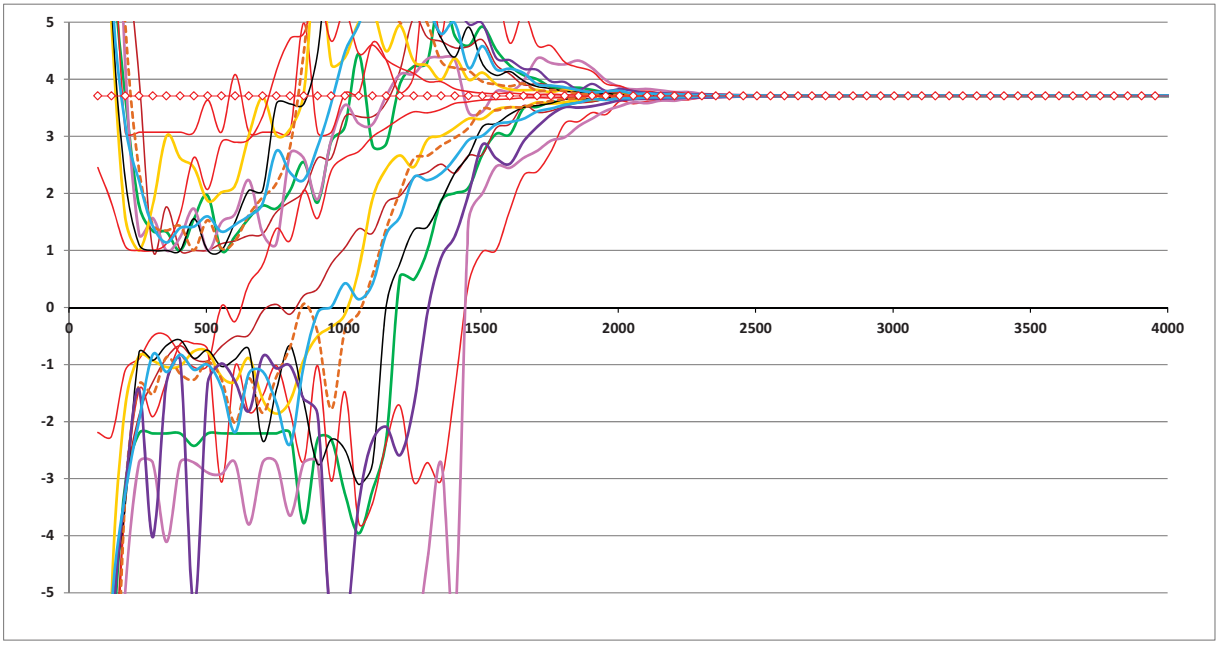

Obr. 5.: Základní nastavení pro porovnání: Velikost základní populace $n$=50, křížením vytvořeno $x$=50 potomků v generaci, evoluční tlak $q$=1,03, „intelligent crossover“, koeficient α=1 (velikost přesahu křížení, „jen mezi hodnotami rodičů“ = 0, může být i záporné (což by znamenalo nevybírání ani z celého rozsahu vymezeného rodiči), teoreticky se může blížit –0,5, ale pak se okamžitě ztratí diverzita a metoda zaručeně selže). Evoluční tlak vyjádřený parametrem $q$ popisuje, kolikrát je větší pravděpodobnost výběru rodiče v populaci oproti jinému, který je v seřazené populaci o jednu pozici níže. Pravděpodobnosti výběru rodiče podle pořadí v populaci pak tvoří geometrickou řadu [3].



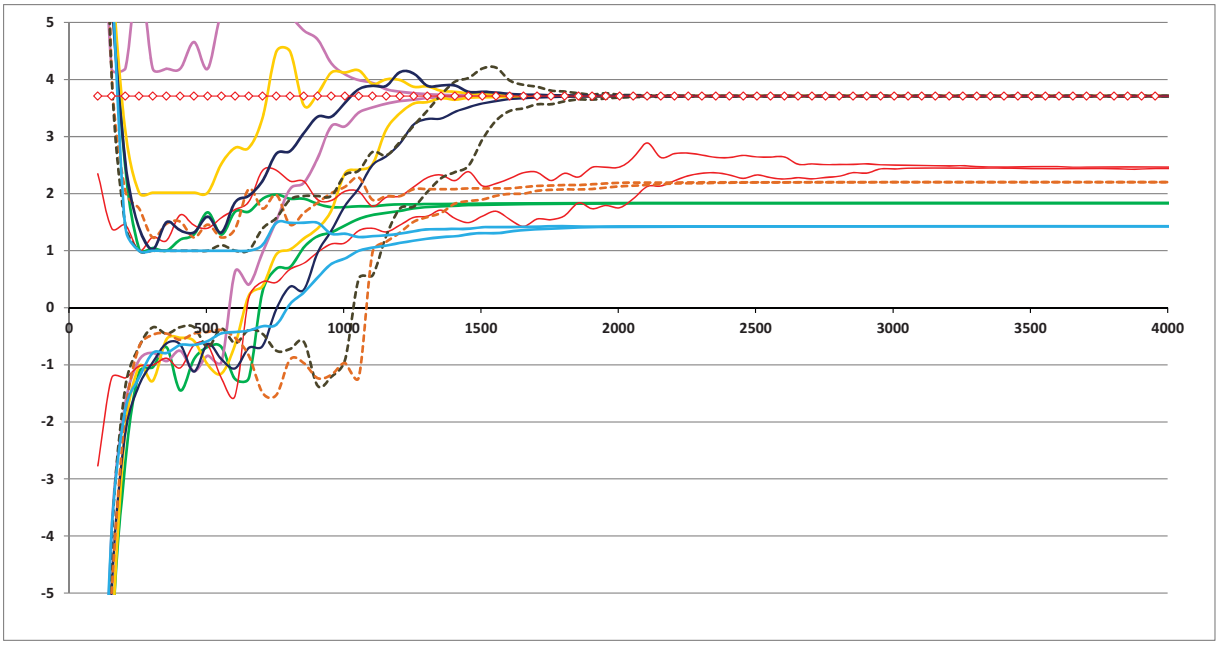

Obr. 6.: Změna hodnoty α na všeobecně doporučovanou hodnotu 0,3 [8]. Evoluce je rychlejší, ale vede na rychlé zamrznutí v bodě, který má k řešení daleko. V případě oranžové přerušované čáry bylo řešení nalezeno rychle, ale jedná se spíše o náhodu.



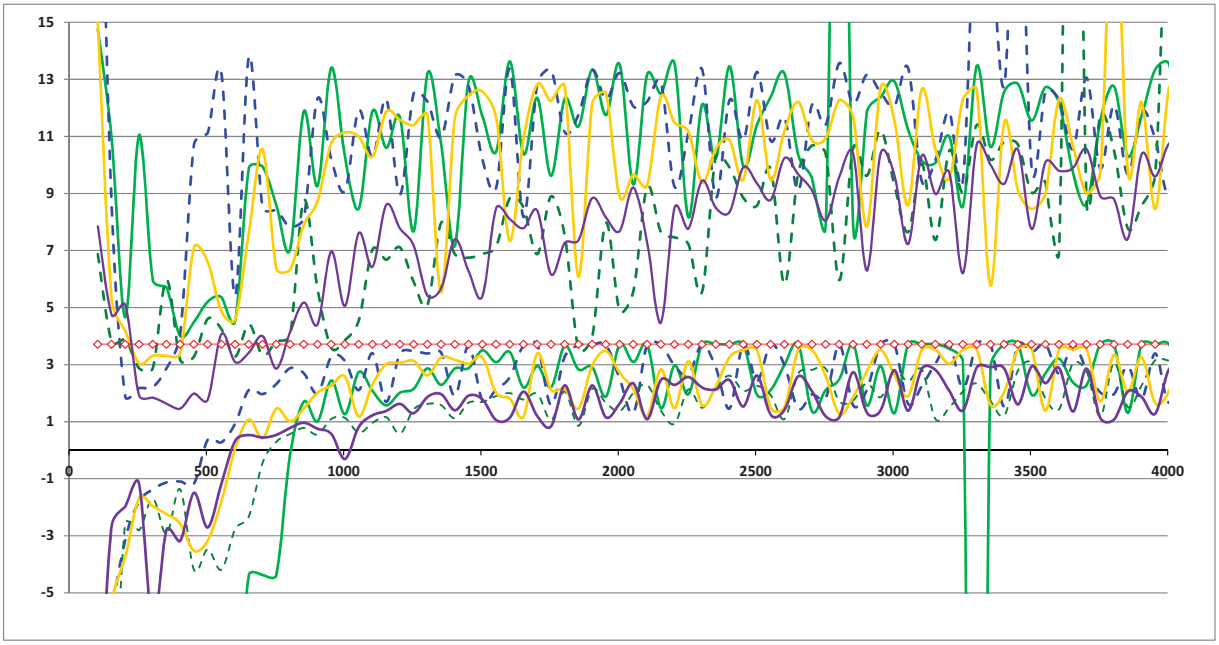

Obr. 7.: Změna hodnoty na α=2,0 . Evoluce se prodloužila, populace nemá tendenci konvergovat, ale ve většině případů nakonec nalezne správné řešení. Velká hodnota koeficientu α zde vlastně nahrazuje mutace, které jinak u genetických algoritmů zabraňují zamrznutí populace. Toto řešení není dokonalé, jak ukazuje světle modrá populace, která stačila „zamrznout“ po cca 30 generacích (graf zde zobrazuje 78 generací). Tenkou červenou čarou zobrazená populace se blíží k řešení, stejně se dá předpokládat, že je nalezne i oranžovou přerušovanou barvou vyznačená populace, která stále neztrácí diverzitu.




Obr. 8.: BLX-α. Od předchozí metody se liší tím, že pro každou ze souřadnic se náhodná hodnota v rozsahu (–α; 1+α) generuje vždy znovu. Při podrobnějším zkoumání zobrazených dat (číselně ve zdrojových datech grafu) lze konstatovat, že ve třetině případů končí 1 až 2 procenta od správné hodnoty, což na grafech není poznat. Při předchozích testech metoda občas také uvízla na chybné hodnotě (až z 10 procent).

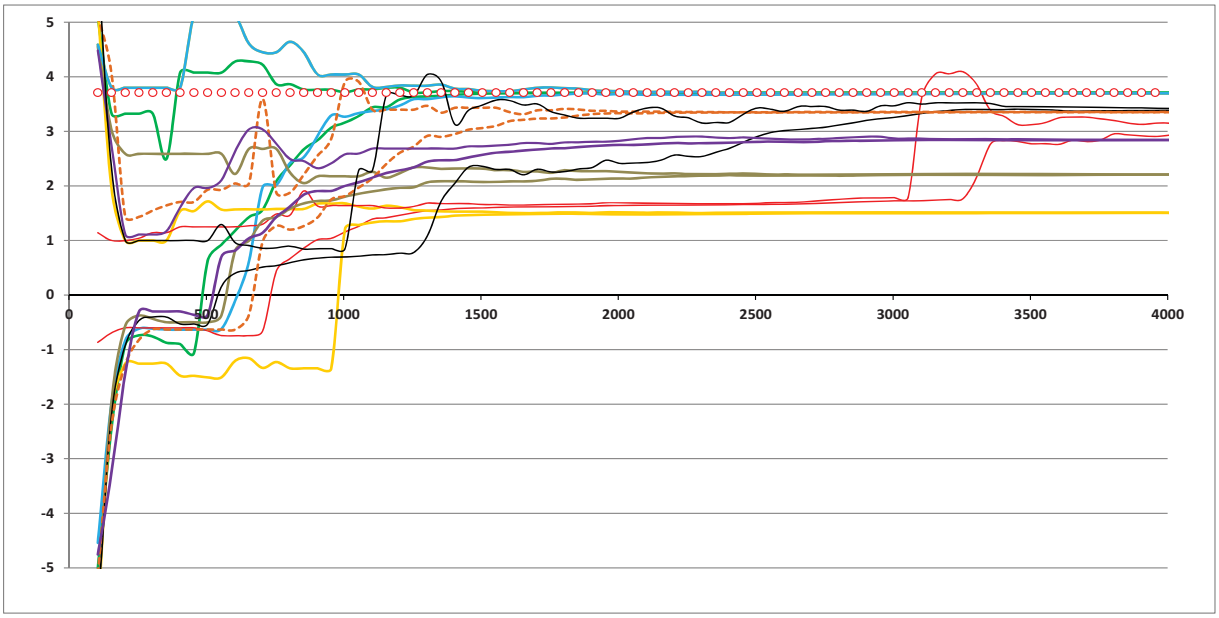



Obr. 9.: BLX-α, α=0,5. Tato hodnota je na mezi, kdy metoda ještě většinou (>50%) nalezne správné řešení.

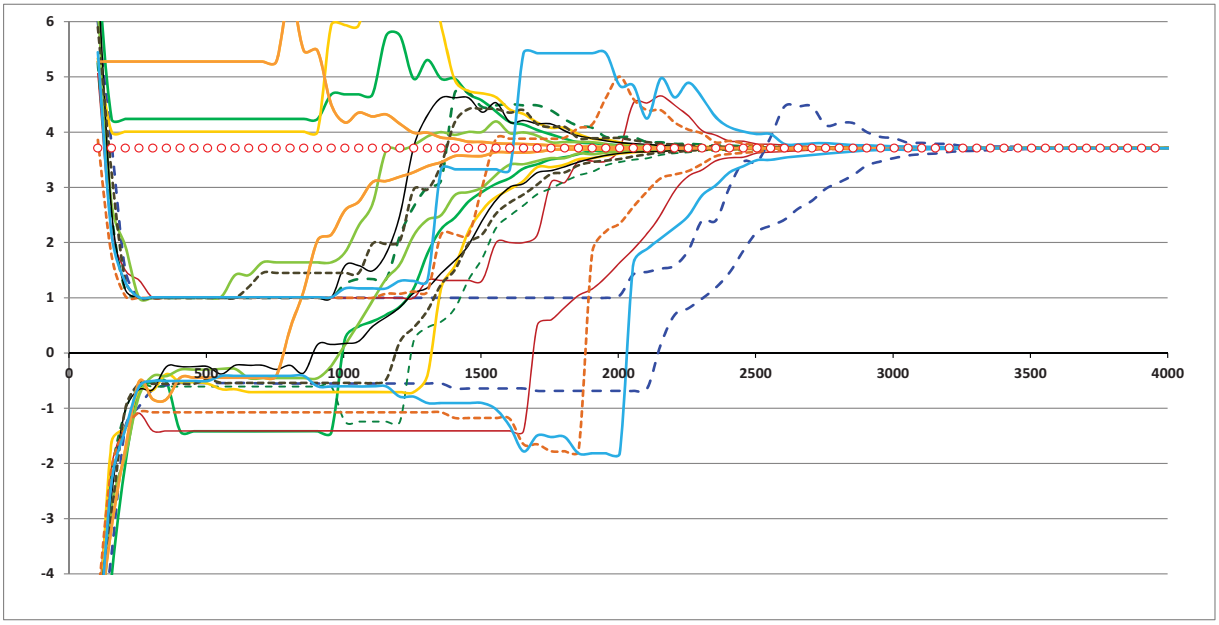



Obr. 10.: Původní nastavení (inteligentní křížení), v každé generaci 45 potomků inteligentním křížením a 5 mutací. V tomto případě není zobrazení mezí (největší a nejmenší hodnoty prvního koeficientu u jednotlivých členů populace) vhodné. Vysoká úroveň mutací (5 z 50 reprezentuje 10%) způsobuje chaotické chování a nestabilitu, nejlepší řešení ale může přesto konvergovat.

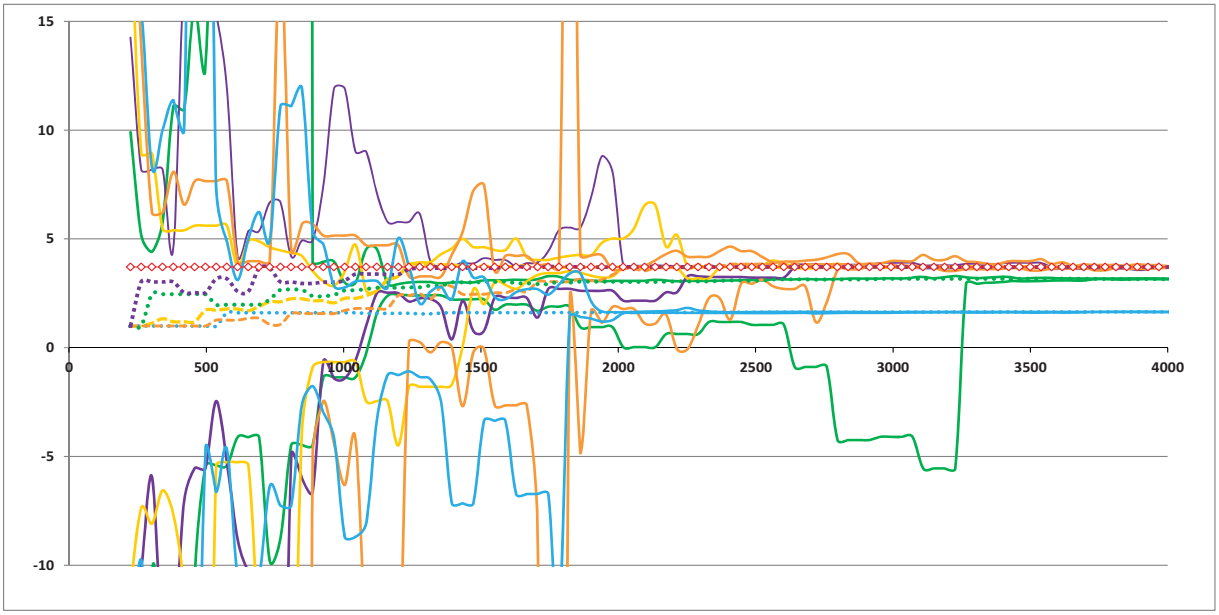

Obr. 11.: Jiné zobrazení dat z předchozího obrázku. Ponechány jen tři průběhy (oproti obr. 10 změněny barvy), ale je doplněno dosud **nejlepší řešení** (tlustá čára stejné barvy). Fialový průběh v rámci 78 generací vůbec nenalezne řešení. Je třeba zdůraznit, že zobrazena je jen první z hledaných konstant, proto se může řešení od nalezené hodnoty i vzdalovat (mohou se zlepšovat ostatní hledané konstanty). K průběhu mezí a nejlepšího řešení je třeba zdůraznit, že použití hladké čáry pro grafy je zde matoucí. Pouze v časech zobrazených červenými kroužky křivky prochází spočítanou hodnotou. Dosažené hodnoty: žlutá – nalezeno řešení (3,71), zelená – přiblížení na 3,7173, fialová – dosaženo 2,94.




Obr. 12.: 2 z 50 (4 %) potomků v generaci tvořeno mutací. V tomto případě je zobrazován stav po setřídění (mutovaní mají vliv na zobrazení, jen pokud přežijí do nové generace). Po těchto úpravách je chování algoritmu mnohem stabilnější, ale je patrné, že ve většině případů nenajde správnou kombinaci konstant. Tomu lze zabránit změnou pořadí operací, kdy mutace se použijí před křížením a výsledky se zařazují do populace místo vzorů (ty se přesouvají na konec, na své místo se mohou vrátit na konci cyklu, kdy jsou prvky znovu seřazeny). Zde nebyla tato modifikace testována.




Obr. 13.: Podobné nastavení jako v předchozím, ale metoda křížení je BLX-α. Testováno vícekrát, řešení nalezeno v tomto případě vždy.




Obr. 14.: Jen pro porovnání: jedna z možností implementace SOMA (angl. Self Organizing Migration Algorithm, česky samo-organizující se migrační algoritmus). Algoritmus udržuje vysokou diverzitu populace a nepotřebuje mutace (pozor na jiné měřítko na svislé ose). Technicky provádí turnajovou selekci. Jako hejnový algoritmus nemění počet jedinců v populaci; vylosuje čtyři jedince, z nich určí nejlepšího (toho ponechá beze změn) a ostatní posune ve směru k němu o vzájemnou vzdálenost násobenou koeficientem α, zde v rozsahu od –0,5 do 2. Protože diverzita populace je veliká, je opět doplněna hodnota nejlepšího zatím dosaženého řešení, ve stejné barvě, ale přerušovanou čarou. V případě světle modře označeného průběhu okolo 40. „generace“ algoritmus ztratí diverzitu populace (největší a nejmenší koeficient v populaci mají skoro stejnou hodnotu) a zamrzne na dosažené hodnotě. Opět je třeba připomenout, že program hledá kombinaci tří koeficientů pro nejlepší proložení bodů (regrese), ale zobrazen je jen první z nich. Řešená úloha je ale velmi nelineární. SOMA při tomto nastavení najde obvykle řešení v 80% případů.

## 5 Závěr

Z výsledků je patrný význam udržení diverzity populace a je názorně patrný smysl představy hejna řešení, které putuje prostorem ve směru nejlepších hodnot. Podrobněji je představena základní metoda BLX-α, která chyběla mezi ukázkami v [10]. Použitá velikost populace spadá do rozsahu, doporučovaném v literatuře (např. [4]). Při větším počtu hledaných parametrů by samozřejmě bylo nutné volit větší populace. Další, co by bylo třeba řešit, je vliv mezí, mezi kterými byla generována počáteční populace (zde v rozsahu ±10). Popisované algoritmy naleznou řešení, i když se nalézá mimo rozsah počáteční populace, ale vliv počáteční populace na rychlost konvergence by také bylo zajímavé popsat.

## 6 Literatura

[1] Zelinka, I. et al.: Evoluční výpočetní techniky: principy a aplikace. Praha: BEN - technická literatura, 2009.
[2] Mařík, V. et al.: Umělá inteligence (3). Vyd. 1. Praha: Academia, 2001.
[3] Hlaváč, V.: Nový algoritmus pro symbolickou regresi pomocí genetického programování s upřesňováním číselné hodnoty koeficientů. Doktorská práce. Fakulta dopravní ČVUT v Praze, 2017.
[4] Banzhaf, W. et al.: Genetic Programming, an Introduction. Morgan Kaufmann Publishers, Inc., San Francisco, California, 1998.
[5] Ghosh, A., Tsutsui, S., Tanaka, H.: Individual Aging in Genetic Algorithms. in: 1996 Australian New Zealand Conf. on Intelligent Information Systems, Adelaide, Australia.
[6] Mitchell, M.: An Introduction to Genetic Algorithms. Cambridge, MA: MIT Press, 1996.
[7] Radcliffe, N.: Forma Analysis of Random respectful Recombination. in: Proceedings of the Fourth International Conference on Genetic Algorithms, San Diego, 13-16 July 1991.
[8] Eshelman L.J., Schaffer J.D.: Real-Coded Genetic Algorithms and Interval-Schemata. in: Foundation of Genetic Algorithms 2, L.Darrell Whitley (Ed.). Morgan Kaufmann Publishers, San Mateo, 1993, pp. 187–202.
[9] Brandejsky T.: Genetic Programming Algorithm with constants pre-optimization of modified candidates of new population. In: Mendel 2004, Brno, 2004, pp. 34-38.
[10] Hlaváč, V.: Vyčíslování konstant progenetické programování. in: Nové metody a postupy v oblasti přístrojové techniky, automatického řízení a informatiky 2017. ČVUT v Praze, 2017.
[11] Hlaváč, V.: A program searching for a functional dependence using genetic programming with coefficient adjustment. in: SCSP Prague, 2016.

# VYVÁŽENÝ BAYESOVSKÝ KLASIFIKÁTOR (BALANCED BAYES CLASSIFIER)

***Pavel Trnka***

***ČVUT v Praze, FS, ústav přístrojové a řídicí techniky. Pavel.Trnka@fs.cvut.cz***

*Abstrakt: Článek představuje novou modifikaci zpracování statistik v Bayesovském klasifikátoru provozních poruchových stavů. Modifikace umožní pro generování statistik používat taková trénovací data, kdy rozsahy (délky) trénovacích množin pro jednotlivé provozní režimy se výrazně liší. Při použití běžných postupů by vedlo použití takových dat k nevyváženým statistikám, takže Bayesovský klasifikátor by generoval vychýlené odhady rozdělení pravděpodobností jednotlivých provozních režimů.*
*Klíčová slova:Bayesovský klasifikátor, diagnostika poruch, porucha*

*Abstract: The article presents a new modification of statistics processing in the Bayesian classifier of operational failure states. Modification allows to use training data sets with considerably different lengths for individual operating modes. Using common procedures to proces such data sets would lead to unbalanced statistics, so Bayesian classifier would generate biased estimates of the probability distribution of individual operating modes.*
*Keywords:Bayes classifier, fault diagnosis, fault*

## 1. Úvod

V úloze diagnostiky poruch je snahou odhalit co nejrychleji vzniklou odchylku od požadované funkce sledovaného systému (technologického procesu). Nežádoucí způsoby jeho chování (provozní režimy) jsou považovány za poruchy. Použití stochastických modelů v diagnostice poruch poskytuje možnost nejen odhadnout výskyt poruchy, resp. odhadnout provozní režim, ve kterém se technologický proces nachází, ale také kvalitativně posoudit věrohodnost takového odhadu. V oblasti pravděpodobnostních modelů znamenalo výrazný posun zavedení Baesovského přístupu k teorii pravděpodobnoti. Bayesovská statistika poskytuje možnost navrhnout výchozí pravděpodobnostní model založený na určitých apriorních znalostech o modelovaném systému a následně model korigovat podle aposteriorních informací získaných srovnáním chování modelu s modelovaným systémem [1], [2].

## 2. Bayesovský klasifikátor stavů

Předpokládejme reálný dynamický systém (technologický proces) popsatelný řízeným markovským řetězcem *m*-tého řádu [4], [3] se vstupním vektorem $\mathbf{v}_k$ a výstupním vektorem $\mathbf{y}_k$, které jsou formalizovány podle vztahů

$$\mathbf{v}_k = \left[v_k[1], v_k[2], \ldots, v_k[\mu]\right]^T \in \varphi_v = \varphi_{v[1]} \times \varphi_{v[2]} \times \ldots \times \varphi_{v[\mu]}, \tag{1}$$

$$\mathbf{y}_k = \left[y_k[1], y_k[2], \ldots, y_k[\eta]\right]^T \in \varphi_y = \varphi_{y[1]} \times \varphi_{y[2]} \times \ldots \times \varphi_{y[\eta]}, \tag{2}$$

kde $\varphi_v$, $\varphi_y$ jsou množiny všech možných hodnot vektorů $\mathbf{v}_k$, $\mathbf{y}_k$ a $\mu$, $\eta$ jsou počty vstupních a výstupních veličin

$$v_k[j] \in \varphi_{v[j]} = \left\{1, 2, \ldots, N_{v[j]}\right\}, \quad N_{v[j]} < \infty, \quad j = 1, 2, \ldots, \mu, \quad k = k_0 + 1, k_0 + 2, \ldots, k_K \tag{3}$$

$$y_k[j] \in \varphi_{y[j]} = \{1,2,\ldots,N_{y[j]}\},\ N_{y[j]} < \infty,\ j = 1,2,\ldots,\eta,\ k = k_0+1, k_0+2,\ldots,k_K \tag{4}$$

Stejným způsobem zavedeme doplňující diskrétní veličinu reprezentující režim činnosti technologického procesu, kterou nazveme poruchový stav

$$f_k \in \varphi_f = \{0,1,2,\ldots,N_f - 1\},\ N_f < \infty,\ k = k_0+1, k_0+2,\ldots,k_K \tag{5}$$

kde $\varphi_f$ je konečná množina všech známých poruchových stavů;
$N_f$ je celkový počet všech známých poruchových stavů a každý poruchový stav je označen indexem z množiny přirozených čísel rozšířené o nulu. Pořadí indexů poruchových stavů není důležité s výjimkou bezporuchového stavu, který bude mít pro větší přehlednost vždycky přiřazen index $0$ (nula). Stochastický model založený na řízeném markovském řetězci podle výše uvedených předpokladů nazveme Bayesovským klasifikátorem poruchového stavu. Bayesovský klasifikátor vyjadřuje rozdělení podmíněné pravděpodobnosti jevu, že se proces v diskrétním čase $k$ nachází v poruchovém stavu $f_k$ za předpokladu, že je v tomto okamžiku pozorován regresní vektor $\mathbf{z}_k$

$$p\left(f_k \middle| D^k\right) = p\left(f_k \middle| \mathbf{z}_k\right) \text{ pro } k = k_0+1, k_0+2,\ldots,k_K, \tag{6}$$

kde $D^k$ je celá minulá historie dat naměřených na procesu až do diskrétního času $k$, viz (4). Vztah (6) ukazuje, že poruchový stav je podmíněně nezávislý na celé minulé historii procesu, jestliže známe regresní vektor definovaný zde jako

$$\mathbf{z}_k = \left\{D^k_{k-m}\right\}; m \geq 1, \tag{7}$$

kde $m \geq 1$ udává maximální hloubku ukládaných dat v základním regresním vektoru s obecnou strukturou

$$\mathbf{z}_k = \left[y_k[1]\ldots y_k[\eta], v_k[1]\ldots v_k[\mu], y_{k-1}[1]\ldots y_{k-1}[\eta], v_{k-1}[1]\ldots v_{k-1}[\mu],\ldots\right]^T. \tag{8}$$

Počet jeho prvků se rovná

$$\rho_z = (\eta + \mu)\cdot(m+1) \tag{9}$$

Základní regresní vektor v Bayesovském klasifikátoru stavů tedy zahrnuje vybraný úsek známé naměřené historie technologického procesu podle (7) včetně posledního známého vstupu a také poslední známé odezvy procesu na tento vstup. Regresní vektor naopak neobsahuje minulé hodnoty odhadů poruchového stavu $f^{k-1}$, neboť tato veličina nepřináší žádnou novou informaci o technologickém procesu k informaci obsažené již v pozorovaných datech $D^k$. Vytvoříme množinu hypotéz ${}_iH$, $i = 1,2,\ldots,r$ o struktuře regresního vektoru ${}_i\mathbf{z}_k$ a k nim příslušné odhady parametrů modelu (4.1.6), tedy rozdělení aposteriorních pravděpodobností v maticích ${}_i\mathbf{K}$ a získáme neúplně určený model

$$p\left(f_k \middle|\, {}_i\mathbf{z}_k, {}_i\mathbf{K}, {}_iH\right) \text{ pro } k = k_0+1, k_0+2,\ldots,k_K. \tag{10}$$

Regresní vektor se strukturou podle *i*-té hypotézy bude tvořen výběrem ze základního vektoru podle vztahu

$${}_i\mathbf{z}_k = {}_i\mathbf{J}\cdot\mathbf{z}_k, \tag{11}$$

kde ${}_i\mathbf{J}$ je výběrová matice podle i-té hypotézy. Matici ${}_i\mathbf{K}$ v Bayesovském klasifikátoru nazveme maticí klasifikace poruch. Vzhledem k (10) má význam

$${}_i\mathbf{K} = \left[{}_iK_{i\zeta,i\psi} = p\left(f_k = \psi \middle|\, {}_i\mathbf{z}_k = {}_i\zeta, {}_i\mathbf{K}, {}_iH\right)\right] \tag{12}$$

pro ${}_i\zeta \in \varphi_{iz}$, $\psi \in \varphi_f$, $k = k_0 + 1, k_0 + 2, \ldots, k_K$
kde:

${}_i\zeta$ je index jednoznačně přiřazený konkrétní realizaci regresního vektoru ${}_i\mathbf{z}_k$ a určuje, na jakém řádku matice ${}_i\mathbf{K}$ se prvek ${}_i K_{{}_i\zeta,\psi}$ nachází;

$\psi$ je index jednoznačně přiřazený konkrétní hodnotě poruchového stavu $f_k$ a určuje, v jakém sloupci matice ${}_i\mathbf{K}$ se prvek ${}_i K_{{}_i\zeta,\psi}$ nachází;

$\varphi_{iz} = \varphi_{iz[1]} \times \varphi_{iz[2]} \times \ldots \times \varphi_{iz[\rho_{iz}]}$ je konečná množina všech možných realizací (hodnot) regresního vektoru ${}_i\mathbf{z}_k$ o délce $\rho_{iz}$ sestaveného podle hypotézy ${}_iH$;

$\rho_{\varphi_{iz}}$ je mohutnost množiny $\varphi_{iz}$, tedy celkový počet možných hodnot RV ${}_i\mathbf{z}_k$;

$\varphi_f$ je konečná množina všech možných hodnot poruchového stavu $f_k$;

$\rho_{\varphi_f}$ je mohutnost množiny $\varphi_f$, tedy celkový počet všech možných poruch $f_k$;

Odvození statistik Bayesovského klasifikátoru, tedy aposteriorních hustot $p\left({}_i\mathbf{K} \middle| f^k, D^k, {}_iH\right)$, $p\left(f_k \middle| {}_i\mathbf{z}_k, {}_i\mathbf{K}, {}_iH\right)$ vede podle [4], [3] na určení matic absolutních četností ${}_i\mathbf{n}(k),\ i = 1, 2, \ldots, r$ pro jednotlivé hypotézy ${}_iH$ o struktuře RV

$$ {}_i\mathbf{n}(k) = {}_i\mathbf{n}(k_0) + {}_i\mathbf{n}^1(k),\ i = 1, 2, \ldots, r, \tag{13} $$

kde ${}_i\mathbf{n}(k_0),\ i = 1, 2, \ldots, r$ jsou matice, jejichž prvky ${}_i n_{{}_i\zeta,\psi}(k_0)$ nabývají apriorně zvolených nezáporných hodnot a vyjadřují naši subjektivní míru důvěry, že z naměřených dat $D^k_{k_0+1}$ získáme pomocí hypotézy ${}_iH$ sdruženou dvojici $\left\{{}_i\mathbf{z}_\kappa = {}_i\zeta, f_\kappa = \psi\right\}$ a můžeme je interpretovat jako počet takových událostí ještě před zahájením identifikace, tedy v diskrétních časech $\kappa \le k_0$.

${}_i\mathbf{n}^1(k),\ i = 1, 2, \ldots, r$ jsou matice, jejichž prvky ${}_i n^1_{{}_i\zeta,\psi}(k)$ představují objektivně zjištěný počet výskytů sdružených dvojic $\left\{{}_i\mathbf{z}_\kappa = {}_i\zeta, f_\kappa = \psi\right\}$ získaných z naměřených dat $D^k_{k_0+1}$ pomocí hypotézy ${}_iH$ pro diskrétní časy $k_0 < \kappa \le k$ a spočítáme je pomocí jednorázového vztahu

$$ {}_i n^1_{{}_i\zeta,\psi}(k) = \sum_{\kappa=k_0+1}^{k} \delta\left({}_i\zeta, {}_i\mathbf{z}_\kappa\right) \cdot \delta\left(\psi, f_\kappa\right) \quad \text{pro } \psi \in \varphi_f \text{ a } {}_i\zeta \in \varphi_{iz}, \tag{14} $$

nebo rekurzivního vztahu

$$ {}_i n^1_{{}_i\zeta,\psi}(k) = {}_i n^1_{{}_i\zeta,\psi}(k-1) + \delta\left({}_i\zeta, {}_i\mathbf{z}_k\right) \cdot \delta\left(\psi, f_k\right) \tag{15} $$

pro $i = 1, 2, \ldots, r$, $k = k_0 + 1, k_0 + 2, \ldots, k_K$, $\psi \in \varphi_f$ a ${}_i\zeta \in \varphi_{iz}$
kde $\delta(\psi, f_k)$ je Kroneckerův delta operátor definovaný obecným vztahem $\delta(\alpha, \beta) = 1$ pro $\alpha = \beta$, jinak $\delta(\alpha, \beta) = 0$.

Statistiky výše popsaného Bayesovského klasifikátoru jsou sice konečných rozměrů, ale přesto velice rozsáhlé. Pro zjednodušení výpočtů využijeme skutečnosti, že statistiky v matici ${}_i\mathbf{K}$ je možné počítat nezávisle po řádcích [4] a využitím úvahy, že aposteriorní pravděpodobnosti ${}_i K_{{}_i\zeta,\psi} = p\left(f_k = \psi \middle| {}_i\mathbf{z}_k = {}_i\zeta, {}_i\mathbf{K}, {}_iH\right)$ jsou nenulové pouze v případě, že se ve fázi učení objevila v trénovacích datech alespoň jednou příslušná dvojice konkrétních hodnot

$\{f_k = \psi, {}_i\mathbf{z}_k = {}_i\zeta\}$. Algoritmus pak reálně pracuje pouze s redukovanými maticemi ${}_i\mathbf{K}^*$, ${}_i\mathbf{n}^*(k)$ vzniklými z matic ${}_i\mathbf{K}$, ${}_i\mathbf{n}(k)$ vynecháním všech prázdných řádků a sloupců. Matice ${}_i\mathbf{n}^*(k)$ (a tím i ${}_i\mathbf{K}^*$) jsou stále ještě poměrně rozsáhlé, ale „řídké" – obsahují mnoho prázdných (nulových) prvků. K další redukci rozměrnosti a zefektivnění algoritmu můžeme použít například metody aproximace predikce založené na markovských řetězcích (AMCP) [3], která zároveň řeší situace, kdy v průběhu diagnostiky není aktuální RV nalezen v naučené statistice.
Diagnostika poruch Bayesovským klasifikátorem probíhá ve dvou fázích.
**Fáze identifikace** parametrů stochastického modelu probíhá podle algoritmu učení s učitelem na základě předběžně či průběžně získaných známých trénovacích dvojic $\left\{f_{k_0+1}^{k_k}, D_{k_0+1}^{k_k}\right\}$. Pro trénovací množinu dat tedy předpokládáme znalost poruchového stavu (režimu), ve kterém se systém nacházel v diskrétních okamžicích $k = k_0+1, k_0+2, \ldots, k_k$. Redukovaná matice četností ${}_i\mathbf{n}^{1*}(k)$,viz (13), začíná jako prázdná matice (s 0 řádky a 0 sloupci). K ní přiřadíme redukovaný vektor všech známých indexů regresních vektorů (zpočátku také prázdný)

$$ {}_i\mathbf{r}_k^* = \left[{}_i\zeta\right]^T \text{ pro } {}_i\zeta \in \varphi_{i}z^{k}, \tag{16}$$

kde $\varphi_{i}z^{k} \subseteq \varphi_{i}z$ je množina indexů všech realizací regresního vektoru ${}_i\mathbf{z}_k$, které byly pozorovány na trénovacích datech. Celočíselné indexy ${}_i\zeta$ nejsou regresním vektorům přiřazeny náhodně, ale fungují jako kód, ze kterého je možné hodnotu regresního vektoru jednoznačně rekonstruovat. Možné způsoby kódování stavů viz např. [5] nebo [3]. Každému řádku matice ${}_i\mathbf{n}^{1*}(k)$ přísluší jeden řádek vektoru indexů ${}_i\mathbf{r}_k^*$. Algoritmus identifikace probíhá v krocích, během kterých procházíme diskrétní časy $k = k_0+1, k_0+2, \ldots, k_k$ a postupně získáváme dvojice $\{f_k, {}_i\mathbf{z}_k\}$. V každém kroku inkrementujeme prvek matice četností ${}_i\mathbf{n}^{1*}(k)$

$$ {}_i n^1_{{}_i r_{{}_i\zeta_k}, f_k}(k) = {}_i n^1_{{}_i r_{{}_i\zeta_k}, f_k}(k-1) + 1 \tag{17}$$

na souřadnicích daných hodnotami $\left\{f_k, {}_i r_{{}_i\zeta_k}\right\}$, kde ${}_i r_{{}_i\zeta_k}$ je číslo řádku ve vektoru ${}_i\mathbf{r}_k^*$, na kterém se nachází index regresního vektoru ${}_i\zeta_k$ a kde $f_k$ je číslo poruchového stavu. Když je to nutné, přidáváme sloupce do matice četností ${}_i\mathbf{n}^{1*}(k)$ pro nové poruchy a řádky do matice ${}_i\mathbf{n}^{1*}(k)$ a vektoru indexů ${}_i\mathbf{r}_k^*$ pro nově objevené hodnoty regresních vektorů. Uvedený postup opakujeme, dokud nedojdou trénovací data. Máme-li k dispozici ještě apriorní rozložení četností reprezentované maticí ${}_i\mathbf{n}(k_0)$, zahrneme je předem do redukované matice absolutních četností obdobným způsobem, kdy opět použijeme pouze řádky a sloupce, které obsahují nenulové hodnoty. Výstupem fáze identifikace je redukovaná matice četností ${}_i\mathbf{n}^*(k)$ s $N_r$ řádky a $N_f$ sloupci a k ní příslušný redukovaný vektor indexů ${}_i\mathbf{r}_k^*$.

Ve **fázi diagnostiky** získáváme průběžně z nově naměřených dat hodnoty regresního vektoru ${}_i\mathbf{z}_k = {}_i\zeta$ v diskrétních časech $k \geq k_k + 1$, přičemž ${}_i\mathbf{r}_k^*$ ani ${}_i\mathbf{n}^*(k)$ se již nemění. Model (10) generuje v každém diskrétním okamžiku $k$ okamžitý odhad rozdělení aposteriorních pravděpodobností jednotlivých známých poruchových stavů ${}_i\hat{\mathbf{K}}^*_{{}_i\zeta}$, které přísluší aktuální hodnotě regresního vektoru ${}_i\mathbf{z}_k = {}_i\zeta$ podle vztahu

$$ {}_i\hat{K}_{{}_i\zeta,\psi} = \hat{p}\left(f_k = \psi \middle| {}_i\mathbf{z}_k = {}_i\zeta, {}_i\mathbf{K}, {}_iH\right) = \frac{{}_i n_{{}_i\zeta,\psi}(k)}{\sum_{\psi_p=0}^{N_f} {}_i n_{{}_i\zeta,\psi_p}(k)} \text{ pro } \psi = 0,1,\ldots,N_f - 1 \tag{18}$$

## 3. Dynamika přechodů mezi stavy soustavy

Bayesovský klasifikátor stavů (10), (12) předpokládá, že se chování sledovaného systému mění velmi zvolna, takže suficientní statistika obsažená v matici četnosti ${}_i\mathbf{n}(k)$ zahrnuje dostatečně obsáhlou informaci pro každý provozní režim a že případné přechodové děje jsou dostatečně významně zastoupeny v trénovací množině provozních dat. Posloupnost hodnot regresního vektoru použitá jako zdroj pro generování statistik, na jejichž základě následně probíhá rozpoznávání určitého provozního režimu (poruchy), v největší míře obsahuje ustálené údaje bezporuchového stavu. Stejná statistika však zahrnuje zároveň relativně krátké přechodové děje, ke kterým dochází při změnách z jednoho provozního režimu na jiný. Statistiky jsou tedy z hlediska dynamiky poměrně silně nevyvážené. Tvar a dynamika přechodových dějů jsou přitom pro správnou diagnostiku velmi důležité. Proto je nutné zajistit, aby se v rozhodovacím procesu dostatečně výrazně projevily.

### 3.1 Interpretace matice absolutních četností

Způsob, jakým generujeme statistiky v matici četností ${}_i\mathbf{n}(k)$ (resp. v redukované matici ${}_i\mathbf{n}^*(k)$) vede při bližším prozkoumání k důležitým poznatkům, které přímo vyplývají z vlastností statistik popsaných v předchozích kapitolách, přesto však nemusí být zcela zjevné.

Každý prvek (redukované) matice absolutních četností ${}_i\mathbf{n}^*(k)$ představuje počet výskytů dvojic $\{{}_i\mathbf{z}_k = {}_i\zeta, f'_k = \psi\}$ v trénovací množině a můžeme snadno odvodit odhad rozdělení aposteriorní pravděpodobnosti

$$\hat{p}\left({}_i\mathbf{z}_k = {}_i\zeta, f_k = \psi \,\middle|\, {}_i\mathbf{K}, {}_iH\right) = \frac{{}_i n_{{}_i\zeta,\psi}(k)}{\sum_{\psi_p \in \varphi_f^*} \sum_{{}_i\zeta_p \in {}_i\mathbf{r}_k^*} {}_i n_{{}_i\zeta_p,\psi_p}(k)} \quad \text{pro } {}_i\zeta \in {}_i\mathbf{r}_k^*,\ \psi \in \varphi_f^* \tag{20}$$

Tento odhad nemusí být pro diagnostiku poruch zcela vhodný, neboť nepotlačuje výběrovou chybu trénovacích dat. Z postupu učení je zřejmé, že při objevení nové poruchy přidáváme do matice četností nový sloupec, do kterého naplníme četnosti výskytu jednotlivých regresních vektorů z příslušné nově získané trénovací množiny. Učení tedy probíhá po sloupcích a každý sloupec reprezentuje rozdělení aposteriorní pravděpodobnosti

$$\hat{p}\left({}_i\mathbf{z}_k = {}_i\zeta \,\middle|\, f_k = \psi, {}_i\mathbf{K}, {}_iH\right) = I_\psi(k)^{-1} \cdot {}_i n_{{}_i\zeta,\psi}(k) \quad \text{pro } {}_i\zeta \in {}_i\mathbf{r}_k^*, \tag{21}$$

kde normovací konstanta $I_\psi(k)$ se vzhledem k vlastnostem pravděpodobnosti určí jako součet celého sloupce

$$I_\psi(k) = \sum_{{}_i\zeta_p \in {}_i\mathbf{r}_k^*} {}_i n_{{}_i\zeta_p,\psi}(k) \,. \tag{22}$$

Potom odhad rozdělení (21) nabude tvaru

$$\hat{p}\left({}_i\mathbf{z}_k = {}_i\zeta \,\middle|\, f_k = \psi, {}_i\mathbf{K}, {}_iH\right) = \frac{{}_i n_{{}_i\zeta,\psi}(k)}{I_\psi(k)} = \frac{{}_i n_{{}_i\zeta,\psi}(k)}{\sum_{{}_i\zeta_p \in {}_i\mathbf{r}_k^*} {}_i n_{{}_i\zeta_p,\psi}(k)} \quad \text{pro } {}_i\zeta \in {}_i\mathbf{r}_k^*. \tag{23}$$

Normovací konstanty $I_\psi$ příslušné jednotlivým sloupcům matice četností ${}_i\mathbf{n}^*(k)$ zřejmě odrážejí rozdělení pravděpodobnosti výskytu jednotlivých poruchových stavů

$$\hat{p}\left(f_k = \psi \,\middle|\, {}_i\mathbf{K}, {}_iH\right) \propto I_\psi(k) \quad \text{pro } \psi \in \varphi_f^*, \tag{24}$$

pro které je normovací konstantou součet všech prvků matice ${}_i\mathbf{n}^*(k)$

$$\hat{p}\left(f_k = \psi \,\middle|\, {}_i\mathbf{K}, {}_iH\right) = \frac{I_\psi(k)}{\sum_{\psi_p \in \varphi_f^*} I_\psi(k)} = \frac{\sum_{{}_i\zeta_p \in {}_i\mathbf{r}_k^*} {}_i n_{{}_i\zeta_p,\psi}(k)}{\sum_{\psi_p \in \varphi_f^*} \sum_{{}_i\zeta_p \in {}_i\mathbf{r}_k^*} {}_i n_{{}_i\zeta_p,\psi_p}(k)} \quad \text{pro } \psi \in \varphi_f^* \tag{25}$$

Spojením (21), (24) dostaneme

$$ {}_i n_{{}_i\zeta,\psi}(k) \propto \hat{p}\left({}_i\mathbf{z}_k = {}_i\zeta \middle| f_k = \psi, {}_i\mathbf{K}, {}_i H\right) \cdot \hat{p}\left(f_k = \psi \middle| {}_i\mathbf{K}, {}_i H\right) \text{ pro } {}_i\zeta \in {}_i\mathbf{r}_k^*, \tag{26} $$

odkud jasně vyplývá, že do statistik vnášíme apriorní rozdělení pravděpodobnosti $\hat{p}\left(f_k = \psi \middle| {}_i\mathbf{K}, {}_i H\right)$ jako důsledek velikostí trénovacích množin.
Vliv relativních délek trénovacích množin na statistiky modelu je nežádoucí, protože nepřináší žádnou užitečnou informaci. Je jasné, že v praxi bude vždy k dispozici nejvíce dat z bezporuchového stavu, zatímco data z poruch budou podstatně omezenější v závislosti na době, po kterou ponecháme technologický proces v daném poruchovém stavu. Přirozeně při běžném provozu je snahou poruchu odstranit ihned, jakmile je odhalena. I v případě, že připravujeme trénovací data a poruchu vyvoláme v modelových podmínkách uměle, nemusí být možné či přípustné setrvat v daném režimu příliš dlouho. Přitom je jasné, že si nepřejeme omezovat činnost diagnostického systému umělým potlačováním pravděpodobnosti vzniku poruchy kvůli nedostatku dat na straně jedné a zbytečným ořezáváním bohatých trénovacích dat pro bezporuchový stav na straně druhé. Proto je vhodné vliv rozdělení (24) v procesu detekce poruch potlačit.

Podle známé Bayesovy formule můžeme vztah mezi podmíněnými pravděpodobnostmi $p\left({}_i\mathbf{z}_k \middle| f_k\, {}_i\mathbf{K}, {}_i H\right)$ a $p\left(f_k \middle| {}_i\mathbf{z}_k, {}_i\mathbf{K}, {}_i H\right)$ vyjádřit jako

$$ p\left(f_k \middle| {}_i\mathbf{z}_k, {}_i\mathbf{K}, {}_i H\right) = \frac{p\left({}_i\mathbf{z}_k \middle| f_k\, {}_i\mathbf{K}, {}_i H\right) \cdot p\left(f_k \middle| {}_i\mathbf{K}, {}_i H\right)}{\sum_{\varphi_f^*} p\left({}_i\mathbf{z}_k \middle| f_k\, {}_i\mathbf{K}, {}_i H\right) \cdot p\left(f_k \middle| {}_i\mathbf{K}, {}_i H\right)} \propto p\left({}_i\mathbf{z}_k \middle| f_k\, {}_i\mathbf{K}, {}_i H\right) \cdot p\left(f_k \middle| {}_i\mathbf{K}, {}_i H\right) \tag{27} $$

To znamená, že nejen (20), ale také odhad (18) bude objektivní pouze v případě, že potlačíme vliv délek množin trénovacích dat jednotlivých provozních režimů (poruchových stavů) procesu. Odhadu nezávislého na délkách trénovacích množin dosáhneme nahrazením odhadů rozdělení aposteriorních pravděpodobností (18) vztahem

$$ {}_i\hat{K}_{{}_i\zeta,\psi} = \frac{\hat{p}\left({}_i\mathbf{z}_k = {}_i\zeta \middle| f_k = \psi, {}_i\mathbf{K}, {}_i H\right)}{\sum_{\psi_p \in \varphi_f^*} \hat{p}\left({}_i\mathbf{z}_k = {}_i\zeta \middle| f_k = \psi_p, {}_i\mathbf{K}, {}_i H\right)} = \frac{\dfrac{{}_i n_{{}_i\zeta,\psi}(k)}{\sum_{{}_i\zeta_p \in {}_i\mathbf{r}_k^*} {}_i n_{{}_i\zeta_p,\psi}(k)}}{\sum_{\psi_p \in \varphi_f^*} \dfrac{{}_i n_{{}_i\zeta,\psi_p}(k)}{\sum_{{}_i\zeta_p \in {}_i\mathbf{r}_k^*} {}_i n_{{}_i\zeta_p,\psi_p}(k)}} \text{ pro } \psi \in \varphi_f^*. \tag{28} $$

kde podle Bayesovského přístupu

$$ \frac{\hat{p}\left({}_i\mathbf{z}_k = {}_i\zeta \middle| f_k = \psi, {}_i\mathbf{K}, {}_i H\right)}{\sum_{\psi_p \in \varphi_f^*} \hat{p}\left({}_i\mathbf{z}_k = {}_i\zeta \middle| f_k = \psi_p, {}_i\mathbf{K}, {}_i H\right)} = \frac{\dfrac{\hat{p}\left(f_k = \psi_p, {}_i\mathbf{z}_k = {}_i\zeta \middle| {}_i\mathbf{K}, {}_i H\right)}{\hat{p}\left(f_k = \psi_p \middle| {}_i\mathbf{K}, {}_i H\right)}}{\sum_{\psi_p \in \varphi_f^*} \dfrac{\hat{p}\left(f_k = \psi_p, {}_i\mathbf{z}_k = {}_i\zeta \middle| {}_i\mathbf{K}, {}_i H\right)}{\hat{p}\left(f_k = \psi_p \middle| {}_i\mathbf{K}, {}_i H\right)}}. \tag{29} $$

Tento výraz můžeme upravit na tvar

$$ {}_i\hat{K}_{{}_i\zeta,\psi} = \frac{\dfrac{\hat{p}\left(f_k = \psi_p \middle| {}_i\mathbf{z}_k = {}_i\zeta, {}_i\mathbf{K}, {}_i H\right)}{\hat{p}\left(f_k = \psi_p \middle| {}_i\mathbf{K}, {}_i H\right)}}{\sum_{\psi_p \in \varphi_f^*} \dfrac{\hat{p}\left(f_k = \psi_p \middle| {}_i\mathbf{z}_k = {}_i\zeta, {}_i\mathbf{K}, {}_i H\right)}{\hat{p}\left(f_k = \psi_p \middle| {}_i\mathbf{K}, {}_i H\right)}} \propto \frac{\hat{p}\left(f_k = \psi_p \middle| {}_i\mathbf{z}_k = {}_i\zeta, {}_i\mathbf{K}, {}_i H\right)}{\hat{p}\left(f_k = \psi_p \middle| {}_i\mathbf{K}, {}_i H\right)} \text{ pro } \psi \in \varphi_f^*. \tag{30} $$

Když srovnáme vztah (30) se vztahem (27), je zřejmé, že úpravou podle (28) jsme dosáhli vyváženého odhadu rozdělení aposteriorních pravděpodobností známých poruchových stavů. Pokud bychom potřebovali předepsat jiné než rovnoměrné rozdělení poruchových stavů, můžeme upravit vyvážený vztah (28) nezávislý na délkách trénovacích množin do podoby

$$ {}_i\hat{K}_{{}_i\zeta,\psi} = p\left(f_k = \psi \,\middle|\, {}_i\mathbf{K}, {}_iH\right) \cdot \frac{\dfrac{{}_in_{{}_i\zeta,\psi}(k)}{\sum_{{}_i\zeta_p \in {}_i\mathbf{r}_k^*} {}_in_{{}_i\zeta_p,\psi}(k)}}{\sum_{\psi_p \in \varphi_f^*} \dfrac{{}_in_{{}_i\zeta,\psi_p}(k)}{\sum_{{}_i\zeta_p \in {}_i\mathbf{r}_k^*} {}_in_{{}_i\zeta_p,\psi_p}(k)}} \quad \text{pro } \psi \in \varphi_f^*, \tag{31} $$

kde $p\left(f_k \,\middle|\, {}_i\mathbf{K}, {}_iH\right)$ je námi zvolené apriorní rozdělení pravděpodobností výskytu poruchových stavů.
Klasifikační matice s prvky počítanými podle (31) představuje sufficientní statistiku Bayesovského klasifikátoru poruch, která je nezávislá na délce trénovacích množin, zachovává významy pravděpodobností podle (18) a přitom nám poskytuje volnost ve volbě apriorního rozdělení pravděpodobností jednotlivých provozních režimů (poruchových stavů).

## 4. Závěr

Modifikace výpočtu odhadu aposteriorních pravděpodobností podle (31) zajišťuje nezávislost statistik na vzájemných délkách trénovacích množin pro jednotlivé provozní režimy sledovaného systému. Poskytuje projektantovi stochastického modelu svobodu nejen při přípravě trénovacích dat, ale také při návrhu vzájemných vztahů mezi provozními režimy, například při potlačování či zdůrazňování určité poruchy. Význam uvedené modifikace není omezen pouze na diagnostiku poruch, ale je obecně aplikovatelný pro Bayesovský klasifikátor v libovolné oblasti použití.

### Poděkování

Práce byla podpořena grantem Studentské grantové soutěže ČVUT č. SGS16/210/OHK2/3T/12.

## Literatura

[1] KÁRNÝ, M. (Ed.): Optimized Bayesian Dynamic Advising. Theory and Algorithms. Springer-Verlag, London, 2006, 529 s., ISBN: 978-1-85233-928-9
[2] PETERKA, V.: Bayesian approach to system identification, Trends and Progress in System Identification, Eykhoff P. (Ed.). Pergamon Press, Oxford, 1981, pp. 239-304.
[3] GARAJAYEWA, G.: Bayesian approach to real-time fault detection and isolation with supervised training. Praha, 2005, ČVUT v Praze, vedoucí disertační práce Prof. Ing. Milan Hofreiter, CSc.
[4] HOFREITER, M.: Bayesovská identifikace technologických procesů. Habilitační práce. ČVUT v Praze, Praha, 1998.
[5] TRNKA, P. Diagnostika poruch neurčitých systémů. Diplomová práce. Praha, 2002, České vysoké učení technické v Praze, Fakulta strojní. Vedoucí práce M. Hofreiter.

# POSOUZENÍ KVALITY ČASTO POUŽÍVANÝCH SENZORŮ PRO IOT APLIKACE (QUALITY ASSESSMENT OF COMMON IOT SENSORS)

***Martin Doubek[1], Michal Haubner[1], Václav Vacek[1]***

***[1] Ústav fyziky, Fakulta strojní, ČVUT v Praze, martin.doubek@cern.ch***

*Abstrakt: Pro účely kalibrace nejčastěji užívaných čidel teploty a vlhkosti pro IoT byla sestavena měřicí trať s precizním DAQ systémem. V článku je stručně nastíněna důležitost ověřování přesnosti měření v rámci IoT aplikací. Je popsán princip nastavení a kontroly teploty a relativní vlhkosti (RH) v aparatuře. Kalibrace referenčních teplotních čidel byla provedena pomocí ultratermostatu a referenčních senzorů vlhkosti pomocí solných roztoků. Sada vybraných senzorů pro IoT aplikace pak byla ověřena vůči těmto zkalibrovaným referenčním senzorům.*
*Klíčová slova: internet věcí, senzory prostředí, verifikační měření, SCADA*

*Abstract: The precision of common and frequently used environmental sensors for IoT applications is verified on a newly commissioned measurement setup, which is based on a top-class SCADA system. The relevance of such measurements is briefly outlined with a focus on IoT framework. The working principle of temperature and relative humidity (RH) control within the setup is described. Calibration procedure for temperature and RH reference sensors is presented, using an ultra-thermostat and saturated salts, respectively. A set of IoT sensors was then tested against the calibrated reference sensors.*
*Keywords: internet of things, environmental sensors, verification measurement, SCADA*

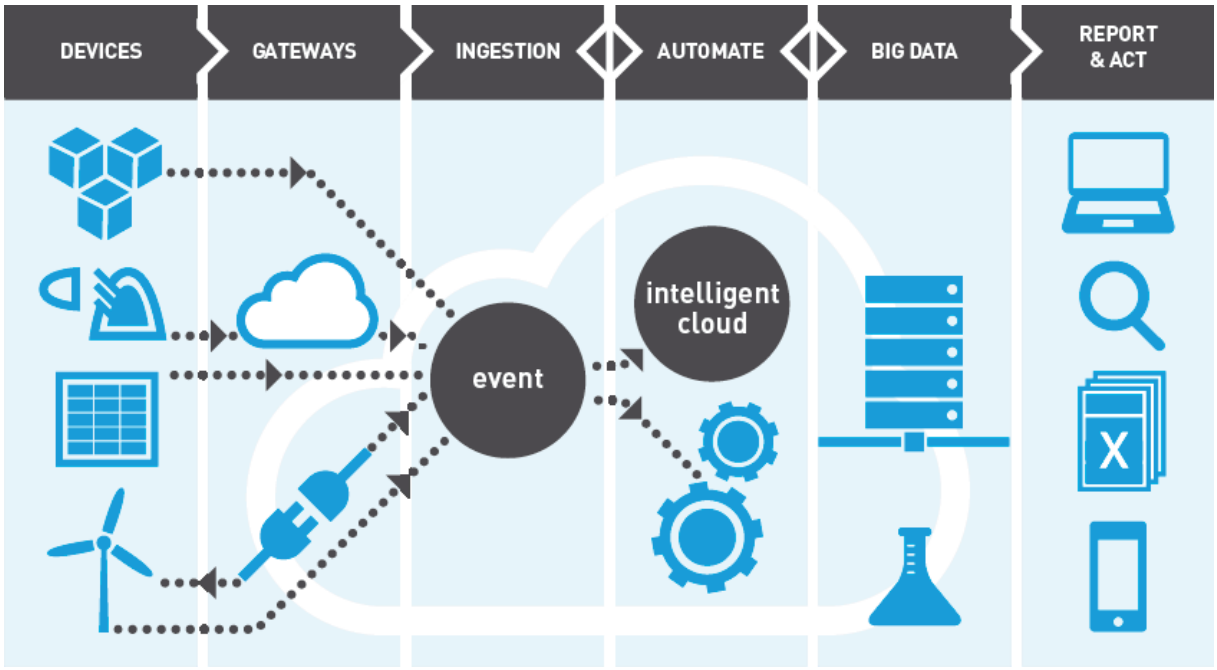
## 1. Úvod

Aplikace a řešení označované jako internet věcí (Internet of Things, IoT) se v posledních 5 letech vyskytují v praxi stále častěji, a to především díky cenové dostupnosti elektronických komponent a zlepšující se dostupnosti síťového připojení. S nárůstem počtu nainstalovaných IoT zařízení roste množství připojených senzorů a jimi vyprodukovaných dat (Big-Data). Kvalitní data naměřená v rámci budov, elektráren, průmyslových provozů, či zemědělství jsou v dnešní době velmi ceněna, jelikož poskytují nástroj k optimalizaci chodu provozu, výnosů či energetické úspornosti zmíněných aplikací. Koncept IoT se tak stal odezvou výrobců elektrotechniky na jasné tendence v mnoha oborech, které tímto způsobem přecházejí na data-driven přístup k řízení mnoha skutečných instalací v technické praxi.

IoT se svou univerzálností, dostupností a modularitou zaplňuje širokou vývojovou mezeru, která existuje mezi průmyslovou a domácí automatizací. Na jedné straně spektra stojí některé poměrně pokročilé funkce senzorů a DAQ systémů, které jsou typické např. pro průmyslové programovatelné kontroléry (PLC) řízené přes SCADA systémy. Na druhé straně spektra stojí modulární stavebnice postavené na integrovaných mikrokontrolerech typu Arduino a Raspberry PI. Obvykle jsou tato zařízení charakterizována nízkou pořizovací cenou elektroniky a přístupným open-source vývojovým prostředím, což jsou nezbytné podmínky pro jejich masové rozšíření. Funkcionalita u IoT zařízení tedy míří mezi tyto dva extrémy, přičemž spojuje výhody obou dvou. Běžně dostupné a populární jsou tak bezdrátově fungující zařízení, která používají relativně levný hardware a uživatelsky přístupné softwarové prostředí. Takovou kombinací vlastností se produkt spadající do kategorie IoT stává dostupným pro široké užití mimo průmysl.

## 1.1 IoT aplikace

Příkladem středně komplexní aplikace může být sběr dat senzorů prostředí v inteligentní budově, která jsou pak vyhodnocena a zpětně využita při řízení vytápění, větrání a klimatizaci (HVAC) budov. Bezdrátově připojené IoT zařízení osazené patřičnými moduly tak měří a odesílají na server časově proměnné hodnoty teploty, tlaku, vlhkosti, prašnosti, koncentrace $CO_2$, hluku, atd., které charakterizují lokální podmínky v místě měření. Obraz o celkovém stavu dané budovy, či jiného objektu vznikne vysokoúrovňovým vyhodnocením dat z jednotlivých senzorů.




Obr. 1: Možné schéma implementace IoT: Zařízení vybavené senzory různých veličin jsou připojeny na internet, kde dochází k archivaci, vizualizaci a vyhodnocení produkovaných dat. Schéma pochází ze zdroje [3].

Síťové připojení lze realizovat jak přes bezdrátové sítě s krátkým dosahem, jako je např. ZigBee, tak přes dedikované datové sítě v pásmu GSM, jako jsou LoRa nebo SigFox. Probíhající přechod k implementaci protokolu IPv6 rozšiřujícím rozsah IP adres na prakticky nevyčerpatelné množství vytváří podmínky k dalšímu růstu počtu IoT zařízení připojených k internetu.
Množství použitých senzorů by v klasickém zapojení vyprodukovalo značné množství dat, a tak samo IoT zařízení filtruje data ze senzoru a odesílá je pouze pokud změna měřené veličiny překročila určité předem definované pásmo hodnot. Tento přístup významně sníží datový tok z IoT zařízení a jejich energetickou náročnost. Tím se sníží zatížení datové sítě, která musí obvykle obsluhovat velké množství koncových zařízení produkujících jen malý datový tok. Dalším významným efektem je snížení odběru elektrického proudu zařízení na minimum, takže bateriově či solárně napájené zařízení mohou fungovat v řádu let bez potřeby údržby.

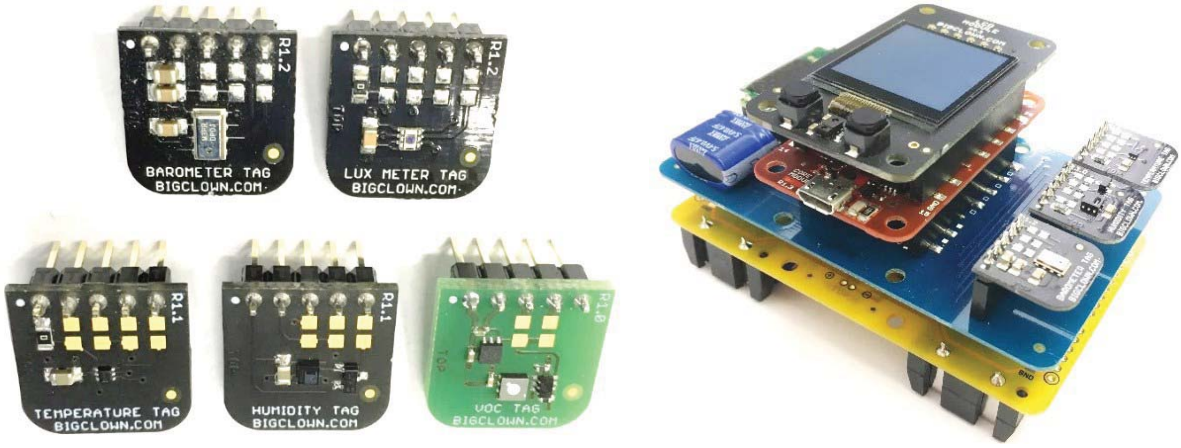
## 1.2 Senzory pro IoT

Citlivým místem rychle rostoucí IoT architektury zůstává, hned po nezbytné bezpečnosti nových a stále početnějších IoT instalací, přesnost senzorů, což je parametr přehlížený řadou koncových uživatelů. Nepřesně měřící senzor nelze plnohodnotně využívat a data jím produkovaná nelze spolehlivě použít pro potřeby měření a řízení. Nepřesnosti v měření mohou vznikat jednak nízkou kvalitou, či chybějící kalibrací použitého senzoru na straně výrobce a nebo špatnou implementací zařízení na straně uživatele. Výsledkem je, že široké spektrum měřených veličin a množství naměřených dat má diskutabilní vypovídající hodnotu.

## 1.3 Testované IoT řešení

Průzkum trhu a první pokusy s levnými senzory a elektronikou (Arduino a levné sady čidel dostupné z běžných eshopů) ukázaly, že dlouhodobě perspektivní pro další výzkum jsou kvalitní produkty vyšší střední třídy pocházející od renomovaných firem, spíše než nejlevnější čínské produkty. S ohledem na minimální zručnost převážné většiny

dnešních uživatelů bylo k testování vybráno komplexní modulární řešení pro IoT aplikace vyvinuté českým startupem Hardwario pod obchodním názvem BigClown.
Modulárně řešený systém sestává z CORE modulu, který funguje na 32-bitovém ARM mikrokontroléru se 192 kB flash paměti. Tento CORE modul je vybaven sub-GHz rádiovým modulem operujícím v pásmu 868/915 MHz, přes který probíhá plně duplexní bezdrátová komunikace s rádiovým USB zařízením připojeným do PC. Dále je CORE modul vybaven dvěma separátními digitálními $I^2C$ sběrnicemi, přes které se připojují další vstupní a výstupní periferie, jako jsou senzory a akuátory, ale i SigFox modul či LoRa modul pro komunikaci v pásmu GSM.
Senzory včetně podpůrné elektroniky a A/D převodníku jsou umístěny na tzv. tagu. Tyto tagy lze volitelně připojovat ke CORE modulu, který z nich odečítá data a dále je zpracovává a odesílá. Samotné senzory neelektrických veličin pro BigClown jsou výrobcem záměrně vybrány od výrobců preferující kvalitu a jsou poměrně přesné.




Obr. 2: Vlevo: základní senzory prostředí modulárního IoT systému BigClown firmy Hardwario. Vpravo: Příklad sestavy pro měření teploty, barometrického tlaku, relativní vlhkosti a koncentrací $CO_2$ vybavené bateriovým modulem a displayem.

## 2. Měřicí trať

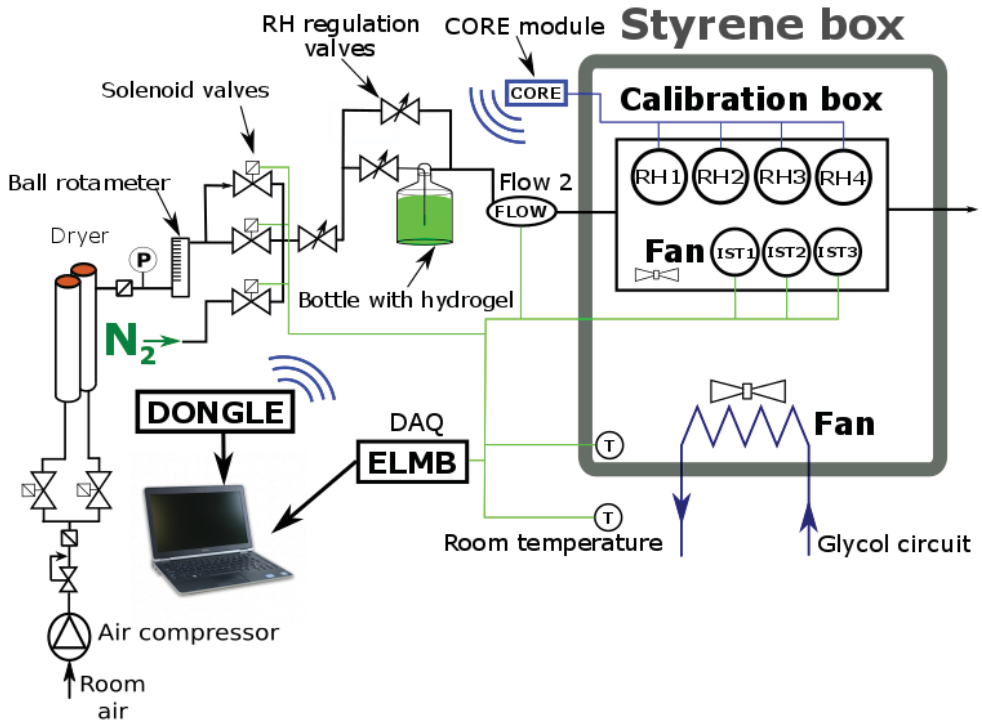
V první fázi projektu jsme se soustředili na ověření přesnosti měření teploty a vlhkosti IoT senzorů v modulech BigClown, což jsou nejčastěji měřené veličiny prostředí na nichž přímo závisí tepelná pohoda osob. Za tímto účelem ověření kvality byla postavena měřicí trať, Obr. 3, schopná vytvořit podmínky prostředí v rozsahu běžných hodnot teploty a vlhkosti. Toto rozmezí se pro teplotu pohybuje od 2 °C do 50 °C a pro relativní vlhkost od 0 % RH do 100 % RH. Pomocí měřící tratěError! Reference source not found. je možné dosáhnout velmi přesných hodnot teploty a relativní vlhkosti v kontrolované oblasti. Stabilita i opakovatelnost je pro teplotu 0.5 °C a pro relativní vlhkost 2 % RH. Tato přesnost je na hranici přesnosti RH senzorů.

### 2.1 Princip funkce

Princip funkčnosti kalibrační tratě je následující. Kompresor nasává atmosférický vzduch, který přes filtr prašných částic vstupuje do sušičky vzduchu naplněné silikagelem. Sušička pracuje ve střídavém režimu, tzn. v případě nasycení silikagelu v jedné patroně se proud vzduchu přesměruje do patrony druhé. První patrona se zahřívá, přičemž dochází k regeneraci silikagelu. Suchý vzduch poté prochází přes odstředivý odlučovač prachových částic s filtrem. Dále proudí přes kuličkový rotametr k sérii solenoidových ventilů, které suchý vzduch distribuují do měricí trati. Následně je vzduch zvlhčován pomocí bubleru naplněného kuličkovým hydrogelem (síť vodou nasycených hydrofilních polymerních řetězců). Je možné použít i vodu bez hydrogelu, avšak za cenu vysokých oscilací tlaku, průtoku a vlhkosti, způsobené nerovnoměrnou tvorbou různě velkých bublin. Okolo bubleru s hydrogelem je obtokový ventil, který reguluje množství vzduchu proudícího přes hydrogel, a tak i vlhkost vzduchu na výstupu. Poté vzduch vstupuje přes elektronický průtokoměr Honeywell AWM5104VN do kalibračního boxu, v němž jsou umístěny referenční senzory relativní vlhkosti IST P-14 spolu s kalibrovanými senzory BigClown. Teplota uvnitř polystyrenového boxu je kontrolována utratermostatem s glykolovým oběhem připojeným na tepelný výměník. Výměník je opatřen ventilátorem pro dosažení homogenního teplotního pole v boxu se senzory. Vstupující vzduch je temperován na teplotu boxu průchodem sérií měděných spirál, které jsou umístěny před vstupem do kalibračního boxu. Přesnost udržení teploty v boxu je stěžejní, neboť přímo ovlivňuje relativní vlhkost, která je při dané absolutní vlhkosti funkcí teploty, viz. [1, 2].

### 2.2 SCADA systém

Referenční senzory teploty Pt1000, relativní vlhkosti IST P-14 a průtoku AWM5104VN jsou připojené do Embedded Local Monitor Board (ELMB) modulu sběru dat, z něhož jsou data přenášena přes OPC server do SCADA prostředí WinCC firmy Siemens. Modul ELMB je jádrem přesného referenčního měřicího systému. Jedná se o zařízení pro sběr dat, které je založeno na mikroprocesoru ATMEL Mega 128. Ten zpracovává data z 16-bitového sigma-delta A/D převodníku, který je vybaven stabilizovaným kalibrovaným napěťovým zdrojem s teplotně-kompenzačními obvody. Připojení měřicích kanálů do převodníku zprostředkovává digitální filtr napojený na 64-kanálový multiplexer. Vnitřní a vnější kalibrace jednotlivých analogových měřicích kanálů byla provedena v rámci návrhu a realizace systému. Dále modul ELMB nabízí po 16 vstupních a výstupních digitálních signálech, které slouží k ovládání relé spínajících výkonové prvky. Nastavení teploty uvnitř polystyrenového boxu, otevření/zavření solenoidových ventilů a řízení ventilátorů uvnitř kontrolované oblasti je rovněž realizováno v prostředí WinCC. V něm je možné naprogramovat časovou sekvenci teplot a spínání, tj. otevření/zavření ventilů pro automatizované měření.




Obr. 3: Schéma měřicí trati postavené pro kalibrace teplotních a vlhkostních senzorů. Vlevo je sestava upravující průtok a vlhkost vzduchu před vstupem do kalibračního boxu. Vpravo je tepelný výměník s ventilátory, který temperuje polystyrenový box. Zeleně je vyznačen SCADA systém, modře bezdrátové připojení IoT senzorů do DAQ počítače.

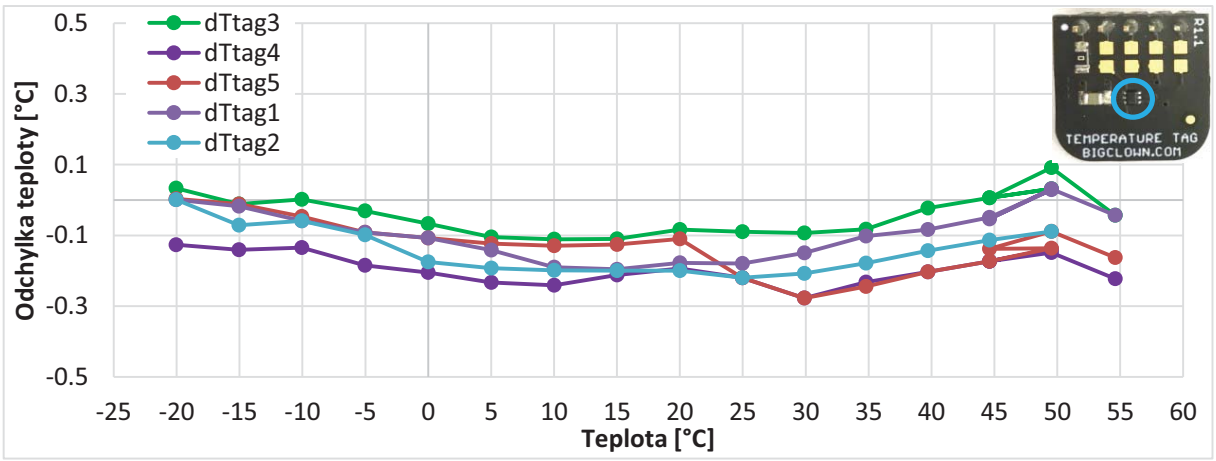
## 3. Vyhodnocení přesnosti IoT senzorů

Senzory relativní vlhkosti dodané společností Hardwario jsou připojené do jednotky CORE module, který pomocí radiového signálu odesílá naměřená data do PC přes USB dongle, viz. Obr. 3. Toto USB zařízení vytvoří v PC virtuální seriový port, ze kterého se data odečítají pomocí software vytvořeném v jazyce Python. Data jsou ve formátu textových řetězců ukládána do .csv souborů. Přenos mezi PC a USB donglem probíhá v sub-GHz rádiovém pásmu. Vzhledem k extrémně nízké spotřebě elektrické energie IoT zařízení, která je v řádu μA, lze zanedbat vlastní ohřev elektroniky joulovým teplem, a tak i jeho možný nežádoucí vliv na měření.

### 3.1 Teplota

IoT senzory teploty, tzv. teplotní tagy IoT systému BigClown používají digitální teplotní senzor TMP112 s rozlišením převodníku 0.01 °C (jedná se o rozlišení, tedy krok měření, nikoliv absolutní přenost). Pro potřeby ověřovacího, resp. kalibračního měření jsou tagy opatřeny kabelem pro prodloužení $I^2C$ sběrnice CORE modulu a možnosti zavedení tagů přímo do jímky ultratermostatu. Jímka je pro lepší vedení tepla vyplněna dielektrickou fluorinertní

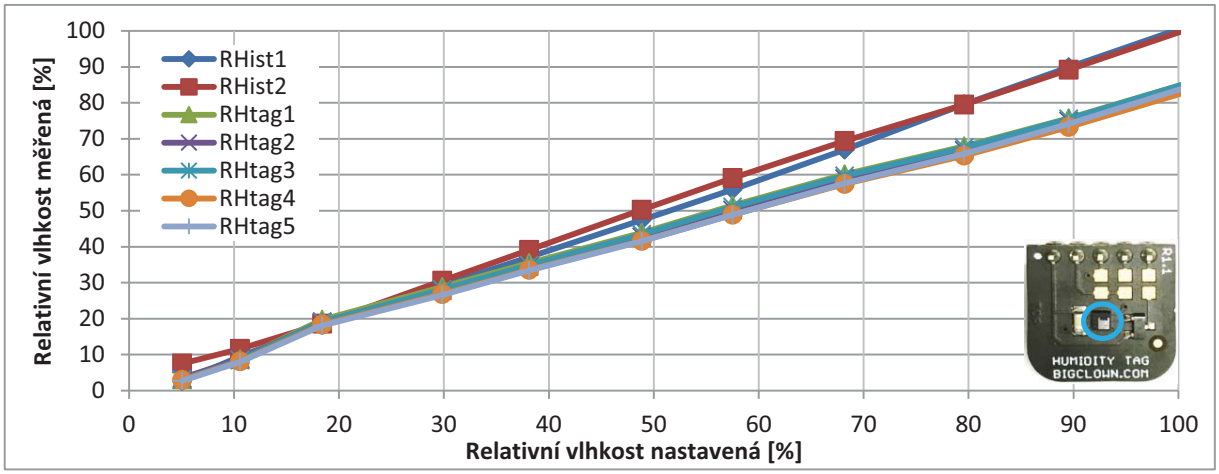
kapalinou, která v jímce ultratermostatu vytvářet stabilní a homogenní teplotní pole, avšak nevede elektrický proud. Kalibrační měření bylo provedeno pro 5 tagů současně v jedné dávce a v rozsahu teplot od -20 °C do 55 °C s krokem 5 °C. Vybrané rozpětí s rezervou pokrývá běžné teploty, kterým jsou tato zařízení v provozu vystavena. V jímce se kromě ověrovaných tagů, nacházel i referenční 4-vodičově zapojený teploměr Pt100, vůči kterému byly tagy vyhodnocovány. Zpracováním výsledků měření dostáváme teplotní závislost odchylky teplotních tagů od referenčního senzoru Pt100, která je vynesena na Obr. 4. Z odchylek senzorů je patrné, že rozdíl měřené a referenční teploty dosahuje v průměru 0.1 ÷ 0.2 °C a maximálně 0.3 °C. Tato přesnost je pro běžné IoT aplikace více než dostačující.




Obr. 4: Graf měření teplot mezi -20°C a +55 °C pro BigClown tagy vůči referenční 4-vodičové Pt100. Odchylky teploty dosáhly v průměru 0.11 °C a maximálně 0.30 °C. Na fotografii je vyznačen samotný digitální teplotní senzor umístěný na PCB desce.

### 3.2 Vlhkost

Jako další byly proměřeny tagy systému BigClown, které měří relativní vlhkost a fungují na kapacitním principu. Tag je vybaven digitálním senzorem s integrovaným AD převodníkem, který pracuje s rozlišením odpovídajícím teplotnímu rozdílu 0.1 %RH, ale přesnost a opakovatelnost senzoru jsou v praxi řádově 2 %RH. Senzor odesílá po $I^2C$ sběrnici hodnoty vlhkosti do Core modulu v intervalech definovaných v aktuálně nahraném firmware.
Z principu fungování kapacitního senoru vlhkosti plyne horší přesnost měření v extremálních hodnotách. V rozmezí 0÷10 %RH a 85÷100 %RH je měření vlhkosti problematické. Výrobce senzoru v doprovodné technické dokumentaci udává, že nejistota měření roste v těchto intervalech z jednotek na desítky procent.




Obr. 5: Naměřená závislost relativní vlhkosti naměřené tagy (viz. foto autora) systému BigClown vůči referenčnímu čidlu IST. Měření bylo provedeno v rozsahu 5÷90 % RH s krokem 10 %. Na fotografii je vyznačen samotný kapacitní RH digitální senzor připájený na PCB desce.

Výsledek provedeného měření je zobrazen v grafu na Obr. 5 a potvrzuje obtížnost měření vysokých relativních vlhkostí. Pro relativní vlhkost vyšší než 70 % RH již nepřesnost měření tagů přesahuje hodnotu 10 % RH. Tato nepřesnost je obdobná u všech tagů, je tedy systematická a opakovatelná s lineární tendencí, takže může být opravena (kompenzována) kalibrací senzoru.

# 4. Závěr

## 4.1 Měřicí trať

Postavená měřicí trať umožňuje ověřovací, resp. kalibrační měření nejen IoT senzorů v širokém a nastavitelném rozsahu teplot a vlhkostí, v rozsahu teplot do 50 °C a relativní vlhkosti prakticky od 0 % RH do 100 % RH. Podmínky měření je možné přesně reprodukovat a udržet stabilní s přesností lepší než 0.5 °C pro teplotu a 2 % RH pro relativní vlhkost. Hodnoty teploty a vlhkosti ve vlastním kalibračním boxu jsou odečítány pomocí předem přesně kalibrovaného řetězce sběru dat, a to v rozsahu běžném pro IoT senzory. Použité referenční senzory teploty byly přesně zkalibrovány v rozsahu teploty od -20 °C do 55 °C oproti certifikovanému referenčnímu senzoru Pt100. Senzory vlhkosti byly kalibrovány nad nasycenými roztoky solí, především se to týkalo hlavního referenčního kapacitního čidla IST. Prostředí WinCC firmy Siemens a použitý modul pro sběr dat ELMB umožňuje ovládání aparatury a automatizaci měření, včetně ovládání ventilů, čerpadel, ventilátorů a termostatu.

## 4.2 IoT senzory

Z hlediska přesnosti a stability byly na sestavené měřicí trati proměřeny senzory vlhkosti systému BigClown, tzv. tagy, v rozsahu 5 ÷ 90 % RH s krokem 10 %. Do relativní vlhkosti 60 % měří tagy s odchylkou měření do 10 %, což je dostatečná přesnost pro běžné IoT aplikace. Zvláště pak se zřetelem na obtížnost kapacitního měření vlhkosti, které je z principu nepřímé. Prakticky dosažitelná přesnost běžných RH senzorů se pohybuje v řádu jednotek procent. Odchylka měření tagů se jeví jako systematická, opakovatelná a s lineárním trendem. Bylo by tedy možné senzor zkalibrovat a dosáhnout tak o řád lepší přesnosti.

Teplotní tagy v počtu 5 kusů systému BigClown byly proměřeny v rozsahu teplot od -20 °C do 55 °C s krokem po 5 °C. Tagy vykazují průměrnou přesnost v řádu 0.1 °C, což je dáno vpodstatě přímým měřením této stavové veličiny. Bez výtky jsou tak testované tagy vhodné i pro relativně přesné aplikace nejen v doméně IoT.

Problémem by v praxi mohla být digitální podstata testovaných senzorů BigClown, ve kterých probíhá převod měřené fyzikální veličiny na digitální hodnotu ještě v samotném senzoru a poté je po $I^2C$ sběrnici hodnota poslána do CORE modulu. Kalibrační fit tedy nelze aplikovat na nízké úrovni ještě před digitalizací měřené veličiny.

# Poděkování

Práce byla podpořena grantem ČVUT v Praze č.: SGS18/058/OHK2/1T/12, Experimentální zařízení pro kalibrace širokého spektra senzorů pro IoT.

# Literatura

[1] NOŽICKA, J., *Základy termomechaniky. Vyd. 1. Praha: Vydavatelství ČVUT, 2001, 187 s*. ISBN 80-01-02409-1.
[2] ŠAFAŘÍK, P. , VESTFÁLOVÁ, M.: *Termodynamika vlhkého vzduchu*. Praha: České vysoké učení technické v Praze, 2016. ISBN 9788001060209.
[3] Web prezentace firmy CODIT. Dostupné online: https://www.codit.eu/how-can-we-help/internet-of-things
[4] Hallgren, B.I., Kvedalen, H., Burckhart, H.J. and Boterenbrood, H., 2001. The embedded local monitor board (ELMB) in the LHC front-end I/O control system. CERN Document Server.

# NEW METHODS OF CONTROL FOR HIGH-SPEED MACHINES

***Prathamesh M. Dusane*[1]**

[1] ***Czech Technical University in Prague, Faculty of Mechanical Engineering, Department of Instrumentation and Control Engineering***
***dusanpra@fs.cvut.cz***

*Abstract: This quarterly report describes the progress made on the topic titled above and also defines a direction for the research to progress. Currently the research is in the phase of literature review. Most of the sources are from IEEE along with a few from ResearchGate. The current focus of the review on the actual design specification of the high-speed machine (HSM) along with the topology and technique of controller.*

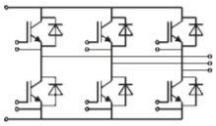
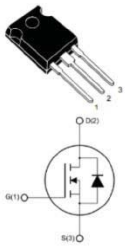
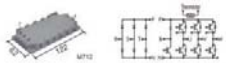
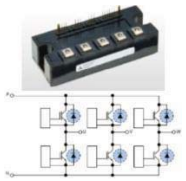
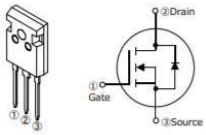
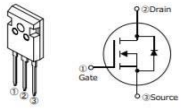
*Keywords: High-speed generator, automotive, multi-level converter*

## 1 Introduction:

With mankind's ever increase in use of electricity as source of energy, we are in need of compact and modular solutions to solve our future's energy demand. Recently, high-speed machines (HSMs) have gained attention of the scientific community in the field of more electric aircraft (MEA), distributed electric generation (DEG) and automotive applications. For this dissertation the field of application is automotive engineering. HSMs are today a common solution on power generation units based on micro-gas and air turbines, both for grid connected and stand-alone applications. For the above titled topic a high-speed generator (HSG) is chosen as the subject of research. A HSG is a high power to weight ratio machine, which operates in the range of 1 - 100 kW and 100,000 - 10,000 rpm. A HSG generates electrical power at high frequency usually 10 - 20 times line frequency due to the speed of its rotation. This calls for the power to be rectified first, then filtered and finally to be used by auxiliaries like DC-DC or DC-AC converters and inverters. Hence, it is of paramount importance to develop an efficient control scheme to maximize power gain from the HSG and minimize harmonic injection into the auxiliaries by the HSG. The controller topology and schematic, it's gate drive control along with all ancillaries is the prospective problem statement of this topic.

## 2 Literature review:

Initially the task was to define automotive ambient conditions like temperature, humidity, vibration and other supply chain parameters like lifetime, supply and failure rate. These were referenced from BOSCH's requirements for automotive semiconductors [1]. Also the task was to find out some examples of prospective semiconductor switches to be used here. Preference was given to SiC chemistry due to it's high frequency and high voltage operation. The selected voltage and current rating were 600 – 800 V at 55 – 100 A. This was chosen because there is high probability that automotive manufacturers will employ HV architecture for their power-trains [2]. The following table summarizes the findings.

| Manufacturer | Code | Type | Rating | Module |
|---|---|---|---|---|
| https://www.semikron.com/dl/service-support/downloads/download/semikron-datasheet-semix101gd066hds-27891200/ | SEMiX101GD066HDs | IGBT | 600 V, 100 A |  |
| http://www.st.com/content/ccc/resource/technical/document/datasheet/group3/0d/4c/81/d0/a5/9b/4a/27/DM00293874/files/DM00293874.pdf/jcr:content/translations/en.DM00293874.pdf | SCTW100N65G2AG | Power MOSFET | 650 V, 85 A, 200 °C |  |
| http://www.fujielectric.com/products/semiconductor/model/sic/hybrid.html | 7MSR100VAB060-50 | IGBT | 600 V, 100 A |  |
| http://www.mitsubishielectric.com/semiconductors/catalog/pdf/sicpowermodule_e_201505.pdf | PMH200CS1D060 | SiC + SBD | 600 V, 200 A |  |
| https://www.infineon.com/dgdl/Infineon-AIKW50N65DH5-DS-v02_01-EN.pdf?fileId=5546d4625cc9456a015d08132bd27f3c | AIKW50N65DH5 | IGBT + Diode | 650 V, 54 A, 175 ℃ |  |
| https://felib.fujielectric.co.jp/download/details.htm?dataid=19947488&site=global&lang=en | FMW60N025S2HF | MOSFET | 600 V, 60 A, 100°C |  |
| http://www.rohm.com/web/global/datasheet/SCT3017AL/sct3017al-e | SCT3022AL | N Channel SiC Power MOSFET | 650 V, 83 A, 100°C |  |

### 2.1 Research paper review:

Over the period of 4 months many papers were read and analyzed pertaining to design and development trends in HSMs along with controller topologies and strategies. It became evident that the control topology would be multi-level inverters due to their reduction in indiviudal switch rating and wave shaping function. Two of the papers are summarized below.

In a paper by D. Gerada, A. Mebarki, N. L. Brown, C. Gerada, Member, IEEE, A. Cavagnino, Senior Member, IEEE, and A. Boglietti, Fellow, IEEE, the authors have described trends in HSMs under the title '**High-Speed Electrical Machines: Technologies, Trends, and Developments [3]**'. The authors have hilighted application areas for high speed machines and their resuting system benefit. In one such application, the electrical machine is placed on the same shaft as the turbine and the compressor wheels in a turbocharger. Driveline efficiency can be further improved by installing an additional power turbine and a high-speed machine at the downstream of the turbocharger to extract waste heat from the exhaust gases, which is often called turbocompounding. The authors illustrates a flywheel developed by Williams Hybrid Power, used within the Porsche 911 GT3R. This flywheel rotates at 40,000 r/min and is used to generate/motorize up to 120 kW to the front-axle motors. The authors have briefly described turbomolecular pumps which are used to obtain and maintain a high vacuum. The relevant application for our topic shown by the authors is in using microturbines as range extenders within serial hybrid and electric vehicles, as a power unit that can charge the vehicle's batteries. The authors illustrate a 50-kW 80,000-r/min microturbine developed by Bladon and claims that such a technology can be just 5% of the size, weight, and parts of an equivalent piston engine. In the final section of the paper the authors have done benchmarking of the aforementioned machines.

A literature survey on **'Different Topologies and Control Techniques of Multi-level inverter [4]'** carried out by Zina Boussada, Omessaad Elbeji and Mouna Benhamed from the Photovoltaic Wind and Geothermal Systems Research Unit in the department of Electrical engineering at the National Engineering school of Gabes in Tunisia highlights some mainstream topologies and control schemes for (Multi-Level Inverters) MLIs. The authors have presented three topologies viz. 1] Diode Clamped Converter (DCC), 2] Flying Capacitor Converter (FCC) and 3] Cascaded Converter (CC) and three control techniques which are 1] Pulse Width Modulation (PWM), 2] Space Vector Modulation (SVM) and 3] Space Vector Control (SVC).

The DCC topology is the most commonly used MLI. Advantages of this type of topology are; 1] Minimization of capacitor requirements. 2] High efficiency for fundamental frequency switching, and 3] Low harmonic content with increase in number of levels is high enough. Concerning disadvantages the authors cite that the difficulty in controlling the intermediate DC levels will tend to unbalanced discharge.

To achieve different voltage levels in the output signals, FCC uses several floating capacitors in each phase instead the clamped diodes in DCC structure. Advantages of FCC topology are, 1] Clamping diodes are not needed, 2] The topology has switching redundancy within the phase, that can be used to balance flying capacitors then just one dc source is needed. 3] Capacitors enables the inverter to ride through sort duration outages and deep voltage sags. Concerning disadvantages the authors cite that, it is an expensive topology due to the large number of capacitors. Added to that, switching utilization and efficiency are poor for real power transmission and voltage tracking of all capacitor is complex.

The cascaded converter called also full-bridge converter is a combination of two single-phase inverters with independent voltage sources. The main advantages of the Cascaded Converter topology are: 1] The topology is based on full-bridge converters, which when connected each other which makes the scheme simple and modular. Further the authors cite that the main drawback is that this topology cannot be applied at lower power levels.

Next, the authors briefly describe the three control technique aforementioned.

Multi-level sinusoidal PWM uses several triangular-carrier signals while keeping only one modulating sinusoidal signal. (N-1) triangular-carrier is needed for N-level inverter. Each carrier is compared, at every instant with the modulating signal. If the modulating signal is greater than the carrier, the switch is switched "ON".
SVM can be extended easily to all multilevel inverters. It's advantages are 1] Low current ripple and easy implementation by a FPGA. These vector diagrams are universal regardless of the type of multilevel inverter. In other words, is valid for five-level diode-clamped, capacitor clamped, or cascaded inverter. The adjacent three vectors can synthesize a desired voltage vector by computing the duty cycle for each vector.
SVC is a control method based on the space-vector theory. This technique of control, called SVC, works with low switching frequencies and it is not as the SVM because it does not generate the mean value of the desired load voltage in every switching interval. The main idea in SVC is to deliver to the load a voltage vector that minimizes the space error or distance to the reference vector. The high density of vectors produced by high-level inverter will generate only small errors in relation to the reference vector; it is, therefore, unnecessary to use a more complex modulation scheme involving the three vectors adjacent to the reference. This method is simple and attractive for high number of levels. The authors stress on the fact that as the number of levels decreases, the error in terms of the generated vectors with respect to the reference will be higher; this will increase the load current ripple.

## 3 Future direction and conclusion:

The literature reviewed up till now points to a 20 – 50 kRpm Machine at a power rating of 50 – 150 kW. The desired control topology would be multi-level converter running on an appropriately investigated and studied control technique. Further research would also focus on MRAS (Model Reference Adaptive System) to estimate machine parameters online for better control. It would also be fruitful to theorize a hybrid NPC+FCC+CC topology.

**Acknowledgement:**

I acknowledge and hope for the continued support and guidance of my teacher doc. Dr. Ing. Martin Novak from the Czech Technical University in Prague, Faculty of Mechanical Engineering, Department of Instrumentation and Control Engineering.

## References

[1] Von Tilis, V. (2006). Design requirments for automotive reliability. [ebook] Montreux, Switzerland: Robert Bosch GmbH. Available at: http://www-g.eng.cam.ac.uk/robuspic/pub_present/ESSDERC06/6-ROBUSPIC-Workshop-ESSDERC06-VVonTils.pdf.

[2] Reber, V. (2016). e-Power: New Possibilities with 800-Volt Charging. [ebook] Porsche Engineering. Available at: https://www.porscheengineering.com/filestore/download/peg/en/pemagazin-01-2016-artikel-e-power/default/09d75d4f-3e8d-11e6-8697-0019999cd470/e-power-%E2%80%93-New-Possibilities-with-800-Volt-Charging-Porsche-Engineering-Magazine-01-2016.pdf.

[3] D. Gerada, A. Mebarki, N. L. Brown, C. Gerada, Member, IEEE, A. Cavagnino, Senior Member, IEEE, and A. Boglietti, Fellow, IEEE, "High-Speed Electrical Machines: Technologies, Trends, and Developments", IEEE Transactions on Industrial Electronics, June 2014.

[4] Zina Boussada, Omessaad Elbeji, Mouna Benhamed, "Different Topologies and Control Techniques of Multi-level inverter", Photovoltaic Wind and Geothermal Systems Research Unit, Electrical department, National Engineering school of Gabes, Zrig 6029 Gabes, Tunisia.

# ROTATION INDUCTION HEATING WITH EXTERNAL ROTOR

***Lubomír Musálek [1], Zdeněk Novák [2]***

***ČVUT v Praze, Fakulta strojní***

[1]lubomir.musalek @fs.cvut.cz, [2]zdenek.novak@fs.cvut.cz

*Abstrakt:*
*Článek se zabývá jinými možnostmi modelování ohřevu nemagnetických válcových ingotů. Zařízení pracuje s nehybným ingotem, kolem kterého rotují permanentní magnety. Vlastní model je řešen v SW Agros2D, který je založen na metodě konečných prvků s vyšším řádem přesnosti.*

*Klíčová slova:*
*Indukční ohřev, permanentní magnet, sdružené úlohy, numerická analýza*

*Abstract:*
*The article deals with other possibilities of modeling heating of non-magnetic cylindrical ingots. The device works with a stationary ingot, around which permanent magnets rotate. The model is solved in SW Agros2D, which is based on a finite element method with a higher order of precision.*

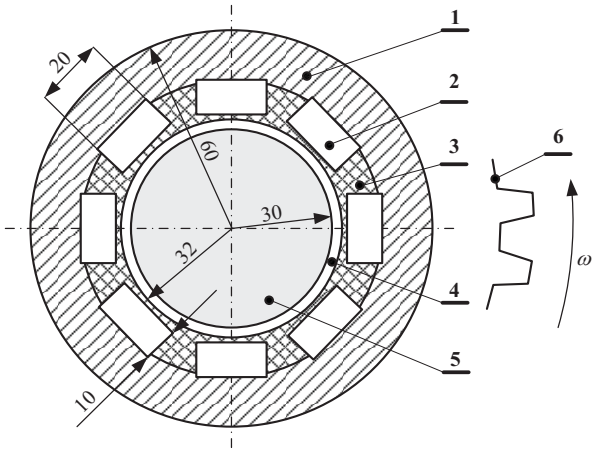
*Keywords:*
*Induction heating, permanent magnet, coupled problems, numerical analysis*

## 1. Úvod

Indukční ohřev válcových ingotů je běžný technologický postup v mnoha průmyslových aplikacích. Vzhledem k tomu, že metoda ohřevu klasickým indukčním ohřevem má poměrně malou účinnost, bylo nutno hledat nové možnosti. Před několika lety, byl navržen nový způsob ohřevu, kde se ingot otáčel ve statickém magnetickém poli vytvářeném supravodivými cívkami na stejnosměrný proud [1]. Ale tato technologie je nákladná a vyžaduje kompletní kryogenní infrastrukturu. Pro ingoty nižších průměrů jsme zahájili testování možnosti jejich vytápění rotací ve statickém magnetickém poli vytvářeném vhodně umístěných permanentních magnetů [2]. Ale i když je tento způsob účinnější a relativně levný pro realizaci, nemůžeme zabránit tepelným ztrátám v důsledku nadměrného chlazení rychle rotujícího ingotu od okolního vzduchu.

## 2. Nově navrhované uspořádání

Po projednání výše uvedených aspektů jsme se rozhodli otestovat inverzní možnost ohřevu - statický ingot se zahřívá v magnetickém poli vytvářeném permanentními magnety namontovaný na prsten z magneticky měkkého materiálu. Schematický pohled na nové uspořádání je znázorněno na obr. 1. rotační část se skládá z magnetického obvodu (1) reprezentované dutou trubkou oceli, na které jsou připevněny permanentní magnety (2). Magnety jsou umístěny v dobré tepelné izolace (3) skleněné vlny, kterou jsou nalepeny na vnitřní stěnu trubky, aby se zabránilo přehřátí permanentních magnetů v důsledku konvekce a radiace z povrchu vyhřívaného ingotu. Hnací moment je přenášen z asynchronního motoru externím ozubeným kolem (6) jak je vidět na obr 1. Ingot (5) je umístěn uvnitř rotační část a nehýbe se. Vzduchová mezera (4) je co nejtenčí, aby se dosáhlo maximální možnou účinnost procesu.




Obr. 1 Uspořádání indukčního ohřevu s vnějším rotorem (všechny rozměry v mm)

Cílem práce bylo modelovat proces indukčního ohřevu ve výše uvedených parametrech. Výpočty jsou prováděny pomocí SW Agros2D [3], který pracuje na principu metody konečných prvků.

## 3. Matematický model

Matematický model procesu je dán dvěma nelineárními parciálními diferenciálními rovnicemi, které popisují rotující magnetické pole a nestacionární teplotní pole v systému. Rozdělení magnetického pole je popsáno z hlediska magnetického vektorového potenciálu rovnicí:

$$\operatorname{curl}\left(\frac{1}{\mu}\operatorname{curl}\boldsymbol{A}-\boldsymbol{H}_{\mathrm{c}}\right)-\gamma\boldsymbol{v}\times\operatorname{curl}\boldsymbol{A}=\boldsymbol{0} \tag{1}$$

kde $\mu$ je permeabilita, $\gamma$ je elektrická vodivost, $\boldsymbol{v}$ je rychlost pohybu bodu, kde řeším pole, $\boldsymbol{H}_{\mathrm{c}}$ je koercitivní síla vyvolána permanentními magnety. Permeabilita $\mu$ a elektrická vodivost $\gamma$ jsou nelineárními funkcemi závislé na teplotě. Okrajová podmínka na obvodu modelu je typu dirichlet $(\boldsymbol{A}=\boldsymbol{0})$.

Teplotní pole je popsáno následující rovnicí:

$$\operatorname{div}\left(\lambda\operatorname{grad}T\right)=\rho c_{\mathrm{p}}\cdot\left(\frac{\partial T}{\partial t}+\boldsymbol{v}\cdot\operatorname{grad}T\right)-p_{\mathrm{j}} \tag{2}$$

kde $\lambda$ je tepelná vodivost, $\rho$ je hustota, a $c_{\mathrm{p}}$ je měrná tepelná kapacita při stálém tlaku. $p_{\mathrm{j}}$ jsou průměrné jouleovy ztráty vzniklé vířivými proudy (nezapočítáváme hysterezní ztráty). Ztráty jsou dány následující rovnicí:

$$p_{\mathrm{j}}=\frac{\left|\boldsymbol{J}_{\mathrm{eddy}}\right|^{2}}{\gamma},\quad\text{where}\quad\boldsymbol{J}_{\mathrm{eddy}}=\gamma\boldsymbol{v}\times\operatorname{curl}\boldsymbol{A} \tag{3}$$

## 4. Popis numerické metody

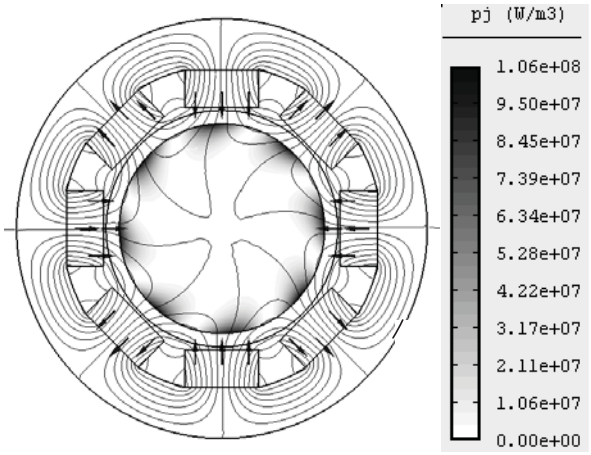
Výše uvedený model je řešen plně adaptivní metodou konečných prvků vyššího řádu přesnosti pomocí kódu Agros2D [3]. Tento SW je vytvořen v C++ kódu a je obecně určen pro numerické řešení soustav obecně druhého řádu parciálních diferenciálních rovnic, a proto se používá hlavně pro modelování složitých fyzikálních problémů. Kód je volně šiřitelný a ve 2D verzi a vykazuje řadu unikátních funkcí, jako je plně automatická hp-adaptivita, práce s uzly visí na jakékoli úrovni, multimesh technologií a možností kombinování trojúhelníkových, čtyřúhelníkových a zakřivených prvků pole.

## 5. Příklad výpočtu

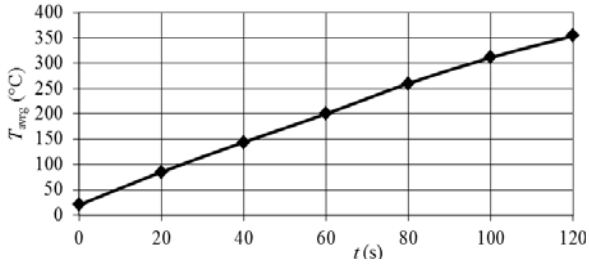
Rychlost rotace $n = 1500$ rpm. Magnetický obvod je z ocele typu 12 040. Neodymové magnety jsou typu VMM10 jejichž magnetická remanence je $B_r = 1.28$ T a relativní permeabilita je $\mu_r = 1.11$. Počáteční teplota je $T_0 = 20$ °C.

Obrázek 2 ukazuje rozložení magnetického pole v systému. Šipky ukazují směr magnetizace jednotlivých permanentních magnetů a v šedé škále je ukázána velikost jouleových ztrát v ingotu (maxima je dosaženo v oblasti pod permanentními magnety.

Obrázek 3 ukazuje časový vývoj průměrné teploty v ingotu. Tato závislost je téměř lineární, protože proces je téměř adiabatický. To je způsobeno tím, že jsme použili malé tloušťky vzduchové mezery, dobrou tepelnou izolaci a dostatečně dlouhý ingot




Obr. 2 Rozložení magnetického pole a výsledných jouleových ztrát




Obr. 3 Průběh průměrné teploty uvnitř ingotu za 120s ohřevu

## Literatura

[1] N. Magnusson, R. Bersas, M. Runde: Induction heating of aluminium billets using hts dc coils, Institute of Physics Conference Series (2004) 1104–1109.

[2] F. Mach, P. Karban, I. Doležel: Induction heating of cylindrical nonmagnetic ingots by rotation in static magnetic field generated by permanent magnets. Journal Comp. Appl. Math., 2012, in print.

[3] http://agros2d.org.

# ELECTRIC BUS DRIVE DEVELOPMENT

***Jaroslav Novák[1], Zdeněk Novák [2], Martin Novák[3]***

***ČVUT v Praze, Fakulta strojní***

***[1]jaroslav.novak@fs.cvut.cz, [2]zdenek.novak@fs.cvut.cz, [3]martin.novak@fs.cvut.cz***

*Abstrakt:*

*Příspěvek v úvodní části rekapituluje problematiku provozu silničních vozidel s akumulátorovým napájením. V další části jsou prezentovány aktivity Odboru elektrotechniky v rámci projektu MPO TRIO, který je zaměřen na vývoj nové koncepce pohonu elektrobusu pro MHD. Tyto aktivity jsou orientovány na oblast koncepčního návrhu pohonné soustavy akumulátor - měnič - motor - převodovka, počítačové simulace jízdy vozidla a testování parametrů akumulátorových článků. V závěrečné části příspěvku je uveden výhled dalších prací na projektu.*

*Klíčová slova:*

*Elektrobus, akumulátor, trakční charakteristika, elektrický pohon, synchronní motor s permanentními magnety*

*Abstract:*

*The paper recapitulates in first part the issue of the operation of road vehicles with accumulator supply. The activities of the Department of Electrical Engineering are presented in next part in the project MPO TRIO, which is focused on the development of a new concept of electric bus for public transport. These activities are focused on the area of concept of the drive systém: battery system - inverter - motor - gearbox, computer simulation of vehicle driving and testing of battery cell parameters. The final part of the paper presents the further work on the project.*

*Keywords:*

*Electric bus, accumulator, traction characteriscic, electric drive, permanent magnets synchronous motor*

## 1 Úvod

I když při pohledu na všední dopravní ruch v České republice nejsou ještě patrné nějaké mimořádné kvalitativní změny, světové dění v oblasti dopravních technologií a řada aplikací objevujících se i u nás nenechává na pochybách, že stojíme na prahu nové éry dopravní techniky, která směřuje k elektřině. Efekty jsou zřejmé, kromě markantních ekologických přínosů se jedná o kvalitativně nové trakční vlastnosti vozidel, energetické úspory, komfort jízdy a v neposlední řadě eliminaci závislosti na tak nejistém surovinovém zdroji, jakým je ropa.

Základním impulsem je velmi rychlý rozvoj techniky lithiových akumulátorů, i když v této oblasti nebyla primárním motivem elektromobilita, ale spotřební komunikační a informační elektronika, především mobilní telefony, notebooky, tablety. V současnosti však již ve světovém měřítku převyšuje produkce akumulátorů pro elektromobilitu, měřená v kapacitě vyrobených akumulátorových článků, produkci článků právě pro spotřební elektroniku. Rozvoj techniky akumulátorů elektrické energie zaznamenává v posledních desetiletích rostoucí tempo. Zatímco v devadesátých letech 20. století byl univerzálně pro trakci k dispozici olověný akumulátor s hmotnostní hustotou energie do 30 Wh/kg, v roce 2010 jsou již k dispozici Li – ion akumulátory s hmotnostní hustotou energie okolo 80 Wh/kg, v roce 2015 dosahuje hmotnostní hustota energie těchto akumulátorů 150 Wh/kg, v roce 2017 jsou běžně k dispozici články s hmotnostní hustotou energie 200 Wh/kg, avšak jsou v tomto roce nabízeny i špičkové články s hmotnostní hustotou energie 250 až 350 Wh/kg.

Sledujeme-li v ČR rozvoj elektrických dopravních prostředků s akumulátorovým napájením, které nahrazují vozidla se spalovacími motory, je v současnosti nejrychleji se rozvíjející oblastí MHD, a to jak na úrovni výrobců, tak na úrovni dopravců. V ČR se výrobou elektrobusů a parciálních elektrobusů zabývají firmy ŠKODA

ELECTRIC, SOR a EKOVA. Velmi výrazné je tempo nasazování těchto vozidel v MHD. Elektrobusy jsou provozovány nebo je připravován jejich provoz v MHD například v Ostravě, Plzni, Hranicích na Moravě nebo Třinci. Období let 2017 a 2018 však signalizuje výrazný nástup parciálních elektrobusů, tedy trolejbusů s pomocným akumulátorovým napájením. Tato vozidla se v uvedeném období rozjela nebo rozjedou v převážné většině měst v ČR s trolejbusovou dopravou (Zlín, Teplice, Plzeň, Brno, České Budějovice, Opava, Hradec Králové, Pardubice, Ostrava, Mariánské Lázně). V těchto městech budou parciální elektrobusy využívány převážně na kratších úsecích linek bez trolejového vedení. Velmi zajímavá je v tomto směru situace v Praze, kde byla v loňském roce, přesně 45 let od ukončení provozu trolejbusů, otevřena v Prosecké ulici trolejbusová trať určená pro dobíjení parciálních elektrobusů, které jsou však koncipovány pro provoz s akumulátorovým napájením v převážné části trasy. V současnosti zde probíhá testovací provoz parciálního elektrobusu SOR TNB 12 na lince č. 140 – obr. 1.

V oblasti osobních automobilů oznámila firma ŠKODA AUTO zahájení prodeje elektromobilů v roce 2020, v roce 2025 firma plánuje nabídku až pěti typů elektrických vozidel, [7].

Obr. 1.: Parciální elektrobus SOR TNB 12 v Prosecké ulici v Praze

## 2 Aktivity U12110 ve vývoji nové pohonné jednotky pro elektrobus

V rámci programu MPO TRIO byl podpořen projekt zaměřený na vývoj nové pohonné soustavy pro elektrobus SOR. Řešiteli projektu jsou firmy SOR a Rail electronics (dodavatel měničové výzbroje a řídicích systémů) a ČVUT v Praze, Fakulta strojní, především Odbor elektrotechniky U 12110.

Hlavní cíle projektu lze shrnout do následujících bodů:

1. Navrhnout nový koncept pohonu se zvýšenou účinností
2. Specifikovat hlavní parametry komponent pohonného řetězce
3. Simulačně ověřit a optimalizovat parametry komponent pohonného řetězce (simulace jízd vozidla v reálných trasách, vyhodnocení trakčních a energetických parametrů)
4. Specifikovat parametry akumulátorové baterie
5. Ověřit experimentálně vlastnosti uvažovaných akumulátorových článků a výsledky měření využít v simulačních výpočtech
6. Navrhnout konstrukci elektromechanického měniče
7. Navrhnout konstrukci a řízení měničové části
8. Navrhnout konstrukci akumulátorové výzbroje a systém batterymanagementu
9. Realizovat pohonnou jednotku a měničovou výzbroj
10. Provést laboratorní testy pohonné jednotky a měničové výzbroje
11. Instalovat pohonný systém, měničovou výzbroj a akumulátorovou baterii na vozidlo
12. Provést oživení pohonného systému
13. Provést rozsáhlé testování a vyhodnocení trakčních vlastností a energetiky provozu

Odbor elektrotechniky je zapojen zejména v aktivitách 1 až 5, 10 a 13 z výše uvedených bodů. V současnosti

jsou zpracovávány zejména aktivity dle bodů 1, 2, 3, připravuje se experimentální zázemí pro testování akumulátorových článků podle bodu 5.

## 3 Struktura pohonného řetězce elektrobusu

Struktura pohonného řetězce elektrobusu je relativně jednoduchá a standardní. Z akumulátorové baterie je přes energeticky dvojsměrný DC/DC měnič napájen vstupní stejnosměrný obvod trakčního střídače, který napájí třífázový elektromotor. Následuje soustava mechanického přenosu energie na nápravu.

DC/DC měnič slouží pro přizpůsobení napěťových úrovní akumulátorové baterie a vstupu trakčního střídače, zároveň stabilizuje vstupní napětí střídače a eliminuje kolísání napětí akumulátorové baterie dané jejím stupněm vybití a úbytkem napětí na vnitřním odporu.

Trakční střídač má standardní zapojení, jedná se o třífázové můstkové zapojení se šesti IGBT a šesti zpětnými diodami. Trakční střídač formuje výstupní třífázovou soustavu pomocí šířkově pulsní modulace.

Novinkou oproti současným elektrobusům z produkce SOR, které jsou vybaveny trakčními asynchronními motory, je použití synchronního motoru s permanentními magnety. Použití synchronního motoru s permanentními magnety lze charakterizovat v následujících bodech:

- Menší rozměry a hmotnost oproti asynchronnímu motoru
- Větší momentová přetížitelnost oproti asynchronnímu motoru (až 3x)
- Okamžitá pohotovost k přechodu do elektrodynamické brzdy díky stálému nabuzení permanentními magnety
- Nutnost řešení odpojitelnosti motorů pro případ poruch v trakčním obvodu (motor je stále nabuzen)

Dominantní je oproti asynchronnímu motoru snížení rozměrů a hmotnosti motoru. Některé odlišnosti jsou oproti asynchronnímu motoru i v řešení regulační struktury momentu a v nutnosti použití snímače úhlového natočení rotoru. Tato problematika byla již dříve zpracována na Odboru elektrotechniky, [9].

Zásadní koncepční otázkou, která ještě v současnosti nebyla dořešena, je koncepce přenosu mechanické energie z hřídele elektromotoru na nápravu. V každém případě je použita nápravová převodovka. Další struktura mechanického přenosu je zpracovávána ve dvou variantách:

1. Přenos z hřídele motoru přímo na vstup nápravové převodovky
2. Vložení řaditelné převodovky mezi motor a nápravovou převodovku

V případě použití řaditelné převodovky jsou uvažovány dva rychlostní stupně: první rychlostní stupeň s převodem vložené převodovky 1,8 a druhý rychlostní stupeň, kdy je moment přenášen z motoru přímo na vstup nápravové převodovky. Vložení řaditelné převodovky komplikuje konstrukci a zhoršuje jízdní vlastnosti – zejména se jedná o poklesy tažné a brzdné síly při řazení. Na druhou stranu použití řaditelné převodovky dává předpoklady k menším rozměrům a hmotnosti motoru, k dosažení vyšší stoupavosti vozidla a použití elektrické výzbroje na nižší napěťové hladině. Podporou pro rozhodnutí o finální variantě jsou průběžně získávané simulační výpočty jízdy vozidla v konkrétních trasách.

## 4 Specifikace hlavních parametrů pohonu

Při návrhu hlavních parametrů pohonu a motoru je nutno vyjít z hlavních požadavků na vozidlo a z dalších limitujících požadavků daných konstrukčními záležitostmi mechanické i elektrické části. Pro rychlý výpočet základních parametrů, zejména elektromotoru, byl na ČVUT FS sestaven výpočet parametrů v prostředí MS EXCEL. Výpočet předpokládá použití synchronního motoru s permanentními magnety (PMSM). Jako vstupní veličiny výpočtu jsou zadávány tyto hodnoty:

- Mezní napětí $U_{max}$, které je limitováno v první řadě napěťovou hladinou výkonových polovodičových součástek trakčního měniče. Maximální napětí je limitované na 1000V z důvodů bezpečnosti a hlavně z důvodů legislativy, aby na zařízení mohli pracovat osoby proškolené pro nízké napětí.
- Celkový mechanický převod $i_{pcelk}$ zadaný jako součin převodu nápravové převodovky a převodovky motoru.
- Celková účinnost převodu $\eta_{pcelk}$ jako součin účinností obou dílčích převodovek.
- Poloměr kola vozidla $r_k$.

- Maximální rychlost vozidla $V_{max}$.
- Poměrná rychlost přechodu motoru do odbuzování vztažená k maximální rychlosti vozidla $n_{pomodb}$.
- Poměrný úbytek napětí na impedanci R a L motoru $\Delta u_{RL}$pom při jmenovitých otáčkách a jmenovitém momentu.
- Hmotnost vozidla $m_v$.
- Stoupavost vozidla s v %.
- Zrychlení vozidla na maximálním stoupání a.
- Součet jízdních odporů vozidla $F_j$ kromě odporu ze zrychlení a odporu ze stoupání (odpor vzduchu, odpor valení, odpory v komponentách vozidla…) – zadává se jako konstantní přibližná hodnota a platí přibližně pro rozjezd a nízké rychlosti.
- Součinitel rotačních hmot $\xi$.
- Přetížitelnost motoru $p_M$ jako poměr maximálního a jmenovitého momentu.
- Jmenovitá účinnost motoru $\eta_M$ – odhad.
- Jmenovitý účiník motoru $\cos\varphi_n$ – odhad.
- Maximální a minimální napětí článku aku baterie $U_{clmax}$ a $U_{clmin}$.
- Maximální napětí baterie $U_{batmax}$.
- Účinnost měniče $\eta_T$ – odhad.

Výpočet je založen na definování jmenovitých hodnot veličin (napětí, proud, moment) motoru, se kterými je motor možno provozovat po neomezeně dlouhou dobu, a na mezní hodnoty veličin v přetížení, se kterými motor může pracovat po omezenou dobu. Pracuje-li motor v přetížení, je doba tohoto provozu závislá na velikosti okamžitého přetížení. Vzhledem ke struktuře vektorové regulace momentu motoru se ve výpočtu předpokládá rovnost hodnot momentové a proudové přetížitelnosti. Tato rovnost platí zcela exaktně v režimu bez odbuzování, v režimu odbuzování je tato rovnost přibližná.

Postup výpočtu je rozebrán v literatuře [1]. Na základě výpočtů bylo stanoveno 10 variant parametrů elektromotoru a převodovky a z těchto deseti variant byly vybrány dvě varianty prioritní, jedna s řaditelnouu převodovkou, druhá s pevným přenosem točivého momentu z motoru na vstup nápravové převodovky. Varianty lze charakterizovat následujícími hlavními parametry:

Varianta s pevným převodem:

| | |
|---|---|
| Jmenovitý výkon motoru | 154 kW |
| Stoupavost vozidla | 20 % |
| Hmotnost vozidla | 18,4 t |
| Mezní napětí | 1000 V |
| Maximální rychlost vozidla | 80 km/h |
| Celkový převod (na nápravě) | 6,19 |
| Zrychlení na maximálním stoupání | 0,1 $m/s^{-2}$ |
| Jmenovitý moment motoru | 1040 Nm |
| Jmenovité otáčky motoru | 1416 ot/min |
| Maximální otáčky motoru | 2832 ot/min |
| Jmenovité napětí motoru | 392 V |
| Jmenovitý proud motoru | 281 A |
| Momentová a proudová přetížitelnost | 3,23 |

Varianta s řaditelnou převodovkou:

| | |
|---|---|
| Jmenovitý výkon motoru | 129 kW |
| Stoupavost vozidla | 22 % |
| Hmotnost vozidla | 18,4 t |
| Mezní napětí | 750 V |

| | |
|---|---|
| Maximální rychlost vozidla | 90 km/h |
| Celkový převod (na nápravě) | 11,14 |
| Zrychlení na maximálním stoupání | 0,1 $m/s^{-2}$ |
| Jmenovitý moment motoru | 704 Nm |
| Jmenovité otáčky motoru | 1752 ot/min |
| Maximální otáčky motoru | 3185 ot/min |
| Jmenovité napětí motoru | 324 V |
| Jmenovitý proud motoru | 285 A |
| Momentová a proudová přetížitelnost | 3 |
| Preferovaná rychlost řazení | 50 km/h |

Ze srovnání obou variant je u varianty s řaditelnou převodovkou příznivý nižší jmenovitý výkon motoru a zejména nižší napěťová hladina, což zlevňuje a zjednodušuje konstrukci elektrické výzbroje a vede k její vyšší účinnosti. Naproti tomu se jedná o variantu s komplikovanější mechanickou částí a s problematickými propady tažné a brzdné síly při řazení.

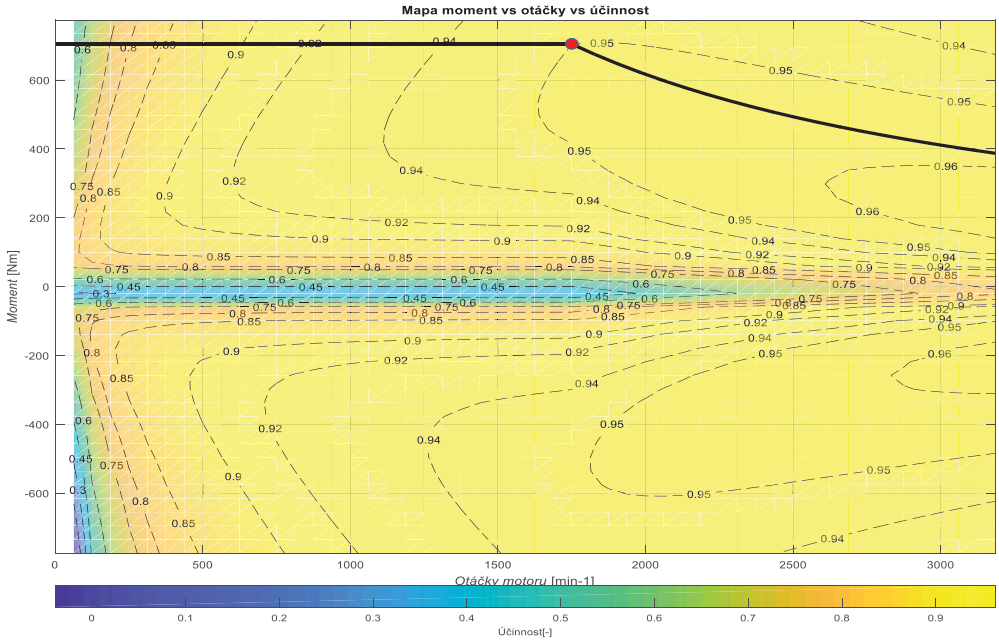
## 5 Simulační výpočty

Pro komplexní vyhodnocení trakčních a energetických vlastností pohonné soustavy byl na Odboru elektrotechniky sestaven simulační model výpočtu jízdy vozidla v definované trase. Vstupními parametry modelu jsou zejména:

- Parametry trasy (zastávky, rychlosti, sklony)
- Parametry vozidla a pohonné jednotky (hmotnost, parametry motoru, převodové poměry, jízdní odpory)

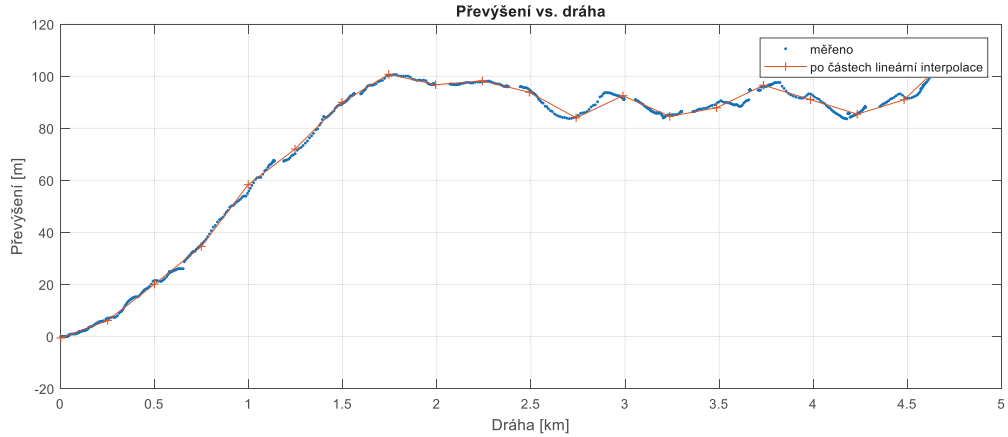
Celý výpočet je založen především na numerickém řešení pohybové rovnice vozidla.

Pro vyčíslení ztrát motoru je základním podkladem účinnostní mapa vypočtená ze štítkových hodnot motoru. Pro ilustraci je na obr. 2 znázorněna účinnostní mapa pro motor z varianty s řaditelnou převodovkou.



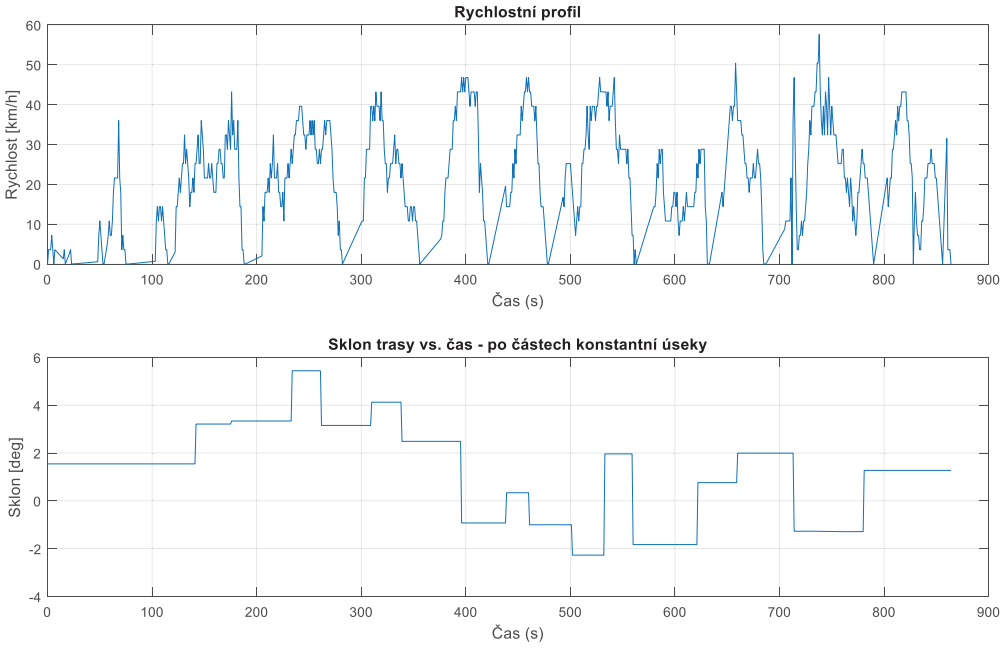

Obr. 2.: Účinnostní mapa motoru pro variantu s řaditelnou převodovkou

Poměrně komplikovaným problémem bylo určení detailních parametrů trasy, zejména sklonu. Výchozí data byla zjištěna měřením pomocí GPS. Z výškového profilu bylo spočítáno převýšení trasy a z něj sklon trasy. Sklon trasy byl určený z výškového profilu po částech lineární interpolací, rozdělením na 20 lineárních úseků. Pro další výpočty je sklon trasy v daném lineárním úseku považován za konstantní. Toto opatření je zavedeno pro eliminaci šumu v měřených datech nadmořské výšky. Na obr. 3 je uveden postup získání vstupních dat sklonu pro výpočet z dat naměřených.




Obr.3.: Příklad převýšení trasy a po částech lineární interpolace, která je dále použitá pro výpočet sklonu trati, jízdní trasa “Na Knížecí” -> “Jinonice”, bus 137

Na obr. 4 Jsou uvedeny průběhy rychlostního profilu a po částech určené hodnoty sklonu trasy.




Obr.4.: Rychlostní profil v závislosti na čase a po částech lineární úseky sklonu trasy, jízdní trasa “Na Knížecí” -> “Jinonice”, bus 137

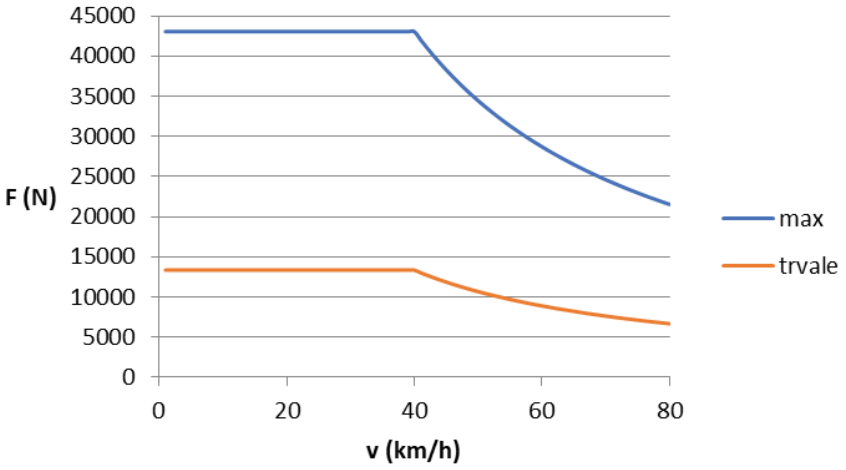
V tabulce Tab. 1 jsou uvedeny příklady výsledků výpočtů spotřeby energie na dvou trasách.

| Trasa | Délka (km) | Spotřeba (kWh) bez uvažování sklonu trasy | Spotřeba (kWh/100km) Bez uvažování sklonu trasy | Spotřeba (kWh) s uvažováním sklonu trasy | Spotřeba (kWh /100km) s uvažováním sklonu trasy |
|---|---|---|---|---|---|
| Cyklus SORT 2 | 0,937 | 0,78 | 83,3 | - | - |
| Na Knížecí -> Jinonice, linka 137 | 4,73 | 2,7 | 57 | 5,8 | 122,6 |
| Na Knížecí -> Jinonice, linka 137, **jízda 2** | 4,73 | 2,2 | 46,5 | 5,9 | 124,7 |
| Na Knížecí -> Jinonice, linka 137, **jízda 3** | 4,73 | 2,7 | 57 | 5,8 | 122,6 |
| Na Knížecí -> Jinonice, linka 137, **jízda 4** | 4,73 | 2,2 | 46,5 | 5,9 | 124,7 |
| Smíchovské nádraží -> Sídliště Zbraslav | 13,2 | 9,2 | 69,7 | Sklon není k dispozici | Sklon není k dispozici |

Tab. 1 Příklady výsledků simulačních výpočtů

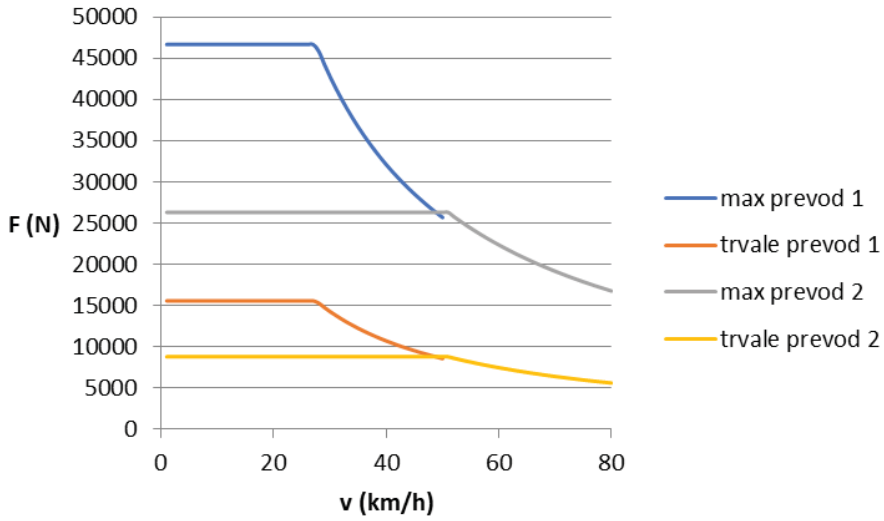
## 6 Závěr

V současnosti byly zahájeny simulační výpočty, které porovnávají trakční a energetické vlastnosti vozidel se dvěma uvedenými preferovanými variantami pohonné jednotky na různých trasách. Na obr. 5 a 6 jsou uvedeny průběhy trakčních charakteristik pro obě preferované varianty.




Obr.5.: Trakční charakteristiky vozidla s pohonnou jednotkou s pevným převodem

Jsou sledovány i varianty jízdy s pohonnou jednotkou s řaditelnou převodovkou, ale s minimalizací řazení či bez řazení a jízdou s trvale zařazeným stupněm 1 (městský provoz) nebo 2 (meziměstský provoz v rovinatém úseku trati). Zároveň se na základě energetických simulačních výpočtů připravuje specifikace akumulátorové baterie. V dalším kroku budou rovněž preferované varianty konzultovány s výrobcem motoru a ohledem na konstrukční záležitosti motorů.




Obr.6.: Trakční charakteristiky vozidla s pohonnou jednotkou s řaditelnou převodovkou

## Poděkování

Příspěvek vznikl v rámci projektu MPO TRIO FV 30213 Výzkum a vývoj trakčního systému elektrobusu s vyšší účinností.

## Literatura

[1] Novák, J., Novák M.: Vybrané problémy návrhu elektrického pohonu elektrobusu, Technická zpráva, ČVUT v Praze, Fakulta strojní, 2018
[2] Novák, M.: Dílčí zpráva k projektu MPO TRIO „č. FV30213 „Výzkum a vývoj trakčního systému elektrobusu s vyšší účinností“
[3] Novák, J., Sadílek, O., Sýkora, P.: Lithiové trakční akumulátory pro elektromobilitu, část 2. Časopis ELEKTRO 2016, roč. 26, č. 11 a 12, vydavatel FCC Public, Praha 2016, ISSN 0322-9025
[4] Firemní materiály SOR
[5] Firemní materiály Railelectronics
[6] Firemní materiály VUES
[7] www.skoda-auto.cz
[8] Hinčica, L: Nenápadný návrat trolejbusů do Prahy, časopis Československý dopravák, roč. XIV., č. 5, vydavatel MH Development s.r.o., Ostrava 2017, ISSN 1804-2309
[9] Novák, J.: Regulační struktura momentu pro trakční synchronní motor s permanentními magnety, Technická zpráva pro firmu Railelectronic, ČVUT v Praze, Fakulta strojní, Praha 2013

# RYCHLÉ ALGORITMY PRO ADAPTIVNÍ DETEKCI NOVOSTI

## Fast algorithms for adaptive novelty detection

***Matouš Cejnek***

***Ústav přístrojové a řídici techniky, Fakulta strojní, ČVUT v Praze, matousc@gmail.com***

*Abstrakt: Adaptivní algoritmy pro detekci novosti v datech jsou populární nástroj díky své schopnosti kompenzovat některé aspekty nestacionarity data generujících procesů. V tomto článku jsou porovnány dva adaptivní algoritmy pro detekci novosti (Learning Entropy, Error and Learning Based Novelty Detection). Oba nástroje pro detekci novosti jsou testovány na následujících adaptivních algoritmech: NLMS, NLMF, RLS a GNGD. Výsledky experimentální analýzy přináší nové informace o vlivu učících algoritmů na přesnost detekce za použití jejich parametrů. Během experimentální analýzy byly použita syntetická data zatížená různou úrovní šumu a obsahujícť concept drift.*
*Klíčová slova: detekce novosti, adaptivní filtry, algoritmy*

*Abstract: Adaptive algorithms are a popular tool for novelty detection because their ability to compensate some aspects of nonstationarity in data generating processes. Two adaptive algoritms (Learning Entropy, Error and Learning Based Novelty Detection) are compared in this study. Both adaptive novelty detection tools are tested with following learning algorithms: NLMS, NLMF, RLS and GNGD. The results of the experimental analysis reveals new insights in learning algorithm influence on final detection performance. A synthetic data with various levels of noise and concept drift were used for the experimental analysis.*
*Keywords: Detekce novosti, adaptivní algoritmy, concept drift*

# 1 Úvod

Detekce novosti (*novelty detection*) je klíčové téma v oblasti strojového učení. Označení detekce novosti se používá pro automatizovaný proces vyhledávání neočekávaných hodnot v měřených datech. Tyto neočekávané hodnoty mohou představovat chyby měření, změny v procesu, který generuje data, nebo jakoukoliv jinou větší či menší anomálii. Už z podstaty věci, je často nezbytné aby nástroje pro detekci novosti byly dostatečné rychlé pro nasazení v reálném čase [1]. V současnosti existuje velká množina algoritmů pro detekci novosti, postavených na různých principech. Více informací o různých přístupech k detekci novosti je možné získat například ze srovnávací studie [2].

Tento článek pojednává o dvou specifických algoritmech *Error and learning based novelty detection* (ELBND) [3] a *Learning Entropy* (LE) [4]. Tyto algoritmy spadají do kategorie algoritmů pro adaptivní detekce novosti. Adaptivní znamená, že algoritmus se přizpůsobuje změnám v datech a za novost označuje pouze to, co do dat nezapadá z pohledu krátkodobého kontextu. Délka tohoto kontextu závisí na rychlosti adaptace. Oba studované algoritmy (ELBND a EL) jsou použitelné v kombinaci s libovolných adaptivním modelem, který používá adaptivní parametry. Výhoda detekce novosti na základě krátkého kontextu je ta, že je možné ignorovat dlouhodobější změny. Tyto dlouhodobější změny se mohou vyskytovat v datech přirozeně, aniž by znamenali novost, kterou je potřeba detekovat.

Jako adaptivní nosný model pro detekci novosti, je zde v tomto článku použit lineární adaptivní filtr. Algoritmy, které byly použity pro adaptaci těchto adaptivních filtrů v této studii jsou *Normalized least-mean squares* (NLMS), *Normalized least-mean fourth* (NLMF), *Recursive least squares* (RLS) a *Generalized normalized gradiend descent* (GNGD). Tyto algoritmy mají rozdílné vlastnosti a budou více představeny v sekci 2.

Jako data pro provedení experimentální analýzy je použit synteticky vytvořený rozsáhlý data-set. Tento data-set byl vytvořen tak, aby představoval různé výzvy, s kterými je možné se setkat při detekci novosti. Jedna z těchto výzev je *concept drift* [5]. Concept drift (nadále jen drift) je možné popsat jako silnou ne-stacionaritu

dat, která může mít různý původ a různý průběh. Nejčastější případ je graduální drift. V použitém data-setu byl simulován graduální opakující se drift modelovaný jako harmonická vlna přidaná k signálu nesoucímu informace. Další výzva, kterou obsahuje vytvořený data-set, je aditivní šum o různých úrovních. Šum je nejčastější forma znehodnocení dat, proto je použit i v této studii.

Hlavní přínos této studie je otestování a porovnávaní více adaptivních algoritmů. Speciálně vhodnost GNGD a NLMF pro detekci novosti pomocí LE a ELBND zatím nikdy nebyla zkoumána.

# 2 Metody

V této sekci jsou vysvětleny metody použité v této studii, včetně postupu křížové validace, která byla použita pro vyhodnocení výsledků této studie.

## 2.1 Adaptivní filtry

Výstup adaptivního filtru $\tilde{y}(k)$ je definován

$$\tilde{y}(k) = w_1 \cdot x_1(k) + ... + w_n \cdot x_n(k) = \mathbf{x}^T(k)\mathbf{w}(k), \tag{1}$$

kde $(.)^T$ je označení pro maticovou transpozici, $\mathbf{w}(k) = [w_1(k), ..., w_n(k)]$ je vektor adaptivních vah a$\mathbf{x}$ je vstupní vektor. Na začátku jsou adaptivní váhy obvykle nastaveny na náhodné hodnoty s normální distribucí, nulovou střední hodnotou a jednotkovou standardní deviací. Vstupní vektor $\mathbf{x}$ (pro adaptivní filtr velikosti $n$) vypadá následovně

$$\mathbf{x}(k) = [x_1(k), ..., x_n(k)]. \tag{2}$$

Chyba adaptivního filtru je následovně vyčíslitelná jako

$$e(k) = y(k) - \tilde{y}(k), \tag{3}$$

Jednotlivé algoritmy adaptace užité v této studii jsou vysvětleny níže.

### NLMS

Adaptivní filtr NLMS [6] je pravděpodobně nejpopulárnější adaptivní filtr. Vektor adaptivních vah NLMS filtru $\mathbf{w}$ je adaptován na základě pravidla

$$\mathbf{w}(k+1) = \mathbf{w}(k) + \Delta\mathbf{w}(k) = \mathbf{w}(k) + \eta(k)\mathbf{w}(k)e(k), \tag{4}$$

kde $\eta(k)$ je rychlost učení normalizovaná podle výkonu vstupu

$$\eta(k) = \frac{\mu}{\varepsilon + ||\mathbf{x}(k)||^2}, \tag{5}$$

kde $\varepsilon \in \mathbb{R}$ je malá pozitivní konstanta (regularizační výraz) zavedený proto, aby se zachovala stabilita i při vstupech blízkých nule. NLMS algoritmus s tímto regularizačním výrazem je často nazýván $\varepsilon$-NLMS.

### NLMF

Adaptivní filtr NLMF je používaný, protože má obecně vyšší schopnost potlačit šum než NLMS filtr [7]. Na druhou stranu zajištění jeho stability je mnohem komplikovanější než u NLMS algoritmu [8, 9]. NLMF adaptace [6] je velice podobná NLMS adaptaci. Vektor adaptivních vah NLMF filtru $\mathbf{w}$ je adaptován na základě pravidla

$$\mathbf{w}(k+1) = \mathbf{w}(k) + \Delta\mathbf{w}(k) = \mathbf{w}(k) + \eta(k)\mathbf{w}(k)e(k)^3. \tag{6}$$

kde $\eta(k)$ má stejný význam jako v případě NLMS.

### GNGD

Adaptivní filtr GNGD [10] je vytvořen rozšířením NLMS adaptivního filtru. Vektor adaptivních vah GNGD filtru $\mathbf{w}$ je adaptován na základě pravidla

$$\mathbf{w}(k+1) = \mathbf{w}(k) + \Delta\mathbf{w}(k) = \mathbf{w}(k) + \eta(k)\mathbf{w}(k)e(k)^3, \tag{7}$$

kde $\eta(k)$ má odlišný význam než v NLMS a NLMF. Adaptivní rychlost učení $\eta(k)$ počítaná podle (5) využívá proměnlivý regularizační výraz $\varepsilon$, získaný následovně

$$\varepsilon(k) = \varepsilon(k-1) - \rho\mu \frac{e(k) - e(k-1)\mathbf{x}^T(k)\mathbf{x}(k-1)}{(||\mathbf{x}(k-1)||^2 + \varepsilon(k-1))^2}. \tag{8}$$

**RLS**

Pravidlo RLS pro výpočet přírustku adaptivní vah $\Delta\mathbf{w}(k)$ je definováno následovně

$$\Delta\mathbf{w}(k) = \mathbf{R}(k)\mathbf{x}(k)e(k), \tag{9}$$

kde $\mathbf{R}(k)$ je inverzí matice k autokorelační matici a je spočítána následovně

$$\mathbf{R}(k) = \frac{1}{\mu}(\mathbf{R}(k-1) - \frac{\mathbf{R}(k-1)\mathbf{x}(k)\mathbf{x}(k)^T\mathbf{R}(k-1)}{\mu + \mathbf{x}(k)^T\mathbf{R}(k-1)\mathbf{x}(k)}). \tag{10}$$

Počáteční hodnota inverzní matice je zvolena

$$\mathbf{R}(0) = \frac{1}{\delta}\mathbf{I}, \tag{11}$$

kde $\mathbf{I}$ je matice identity a $\delta$ je malá pozitivní konstanta.

## 2.2 Klasifikace a křížová validace

V této studii je použit podobný klasifikační framework jako v předchozích studiích [11] a [3]. Tento klasifikační framework má za úkol otestovat míru úspěšnosti testovaných metod na velkém množství dat na základně anotovaných segmentů dat. Veškerá data jsou rozdělena na segmenty a každý segment je anotovaný informací, zda-li obsahuje změnu (novou událost) nebo ne. Klasifikační framework následně testuje, zda skutečná anotace dat odpovídá detekovanému stavu pro různé prahy citlivosti detekce. Práh se mění v celém rozsahu spektra - od hodnoty kde jsou všechny segmenty klasifikovány jako nové, až po hodnotu kde jsou všechny segmenty dat klasifikovány jako normální. Z takto získaných dat (množství úspěšných detekcí a množství neúspěšných detekcí) je možné spočítat *specificitu* a *senzitivitu*. Z těchto metrik je následně možné určit *reciever operating characteristic* (ROC) křivku, která je užita pro vizuální porovnání výsledků pro různé nastavení prahu. Výsledky jsou dále také redukovány na jedno číselné metriky popisující přesnost algoritmu - plocha pod ROC křívkou (AUROC) a maximální přesnost. Tyto metriky jsou zvoleny z důvodu, že se dobře doplňují. Zatímco maximální přesnost popisuje ideální podmínky klasifikace (optimální nastavení prahu), tak AUROC popisuje schopnost klasifikace bez ohledu na to jak byl klasifikátor nastaven. Pro splnění základní podmínky pro korektní stanovení ROC křivky bylo zapotřebí mít vyvážený data-set ke klasifikaci. Vyvážení tohoto data-setu bylo dosaženo tím, že všechny segmenty dat byly stejně dlouhé a byl stejný počet pozitivních i negativních segmentů.

## 2.3 Metody detekce novosti

Dvě adaptivní metody detekce novosti jsou studovány v tomto článku - ELBND a LE. Obě metody využívají přírůstky adaptivních vah učícího systému. Metoda LE využivá čistě tyto přírůstky, zatímco ELBND metoda navíc používá i samotnou chybu predikce adaptivního modelu. Přírůstky adaptivních vah jsou s chybou korelovány, takže chyby ovlivňuje výstup jak ELBND, tak LE. Primární rozdíl v těchto metodách leží v tom, že ELBND klade na chybu větší důraz. Detailní popis těchto dvou metod následuje v navazujících podsekcích.

### 2.3.1 Error and learning based novelty detection (ELBND)

Metoda ELBND anotuje každý nový vzorek množstvím obsažené novosti na základě přírůstku adaptivních vah a chyby podle následujícího pravidla

$$\text{ELBND}(k) = \Delta\mathbf{w}(k)e(k). \tag{12}$$

Výstup 12 je vektor, který popisuje míru novosti pro každou adaptivní váhu zvlášť v daném diskrétním čase $k$. Tento vektor lze dále zobecnit na skalár, pro jednodušší vyhodnocení novosti v daném vzorku. Běžný způsob jak získat z vektoru $\text{ELBND}(k)$ skalár $\text{elbnd}(k)$ je funkce maxima z absolutních hodnot

$$\text{elbnd}(k) = \max|\text{ELBND}(k)|. \tag{13}$$

Všimněte si, že tento způsob výpočtu novosti nepotřebuje žádné další parametry, takže problém s optimalizací metody je minimalizován. Na druhou stranu je nutné podotknout, že výstup metody je silně závislý na adaptivním algoritmu (v tomto případě adaptivním filtru). Různé nastavení adaptivního filtru můžou způsob různé výsledky této metody detekce novosti. Jak bylo ale ukázáno v [3], tak vysoká přesnost predikce není nutně podmínkou pro vysokou přesnost detekce novosti.

### 2.3.2 Learning entropy (LE)

Množství novosti v každém novém vzorku může být spočítáno pomocí Approximate Individual Sample Learning Entropy (AISLE v [4]). Tento způsob ja zkráceně nazýván jako Learning Entropy (LE). Výpočet LE je podle následujícího pravidla:

$$\mathrm{LE}(k) = \frac{1}{n \cdot n_\alpha} \sum f(\Delta w_i(k), \alpha) \; ; \; \forall \alpha \in \alpha \; , \tag{14}$$

kde $n$ je počet adaptivních vah a $n_\alpha$ je počet citlivostí detekce, které si volí uživatel

$$\alpha = [\alpha_1, \alpha_2, \ldots, \alpha_{n_\alpha}] \; ; \; \alpha_1 < \alpha_2 < \ldots < \alpha_{n_\alpha}. \tag{15}$$

Funkce $f(\Delta w_i(k), \alpha)$ je definována jako

$$f(\Delta w_i(k), \alpha) = \begin{cases} 1, & \text{if } |\Delta w_i(k)| > \alpha \cdot \overline{|\Delta w_{Mi}(k)|} \\ 0, & \text{v ostatních případech} \end{cases} \tag{16}$$

kde $\overline{|\Delta w_{Mi}(k)|}$ je střední hodnota okna použitého pro výpočet LE. Tato velikost okna by měla být zvolena s ohledem na možnou periodicitu v datech [4]. Počet citlivostí pro detekci $\alpha$ je volitelný. Doporučení pro volbu hodnot $\alpha$ podle [4] je volit takové hodnoty, aby výsledná $\mathrm{LE}(k)$ byla nižší než 1 alespoň pro jeden vzorek v datech a zároveň aby maximálně jeden vzorek měl hodnotu $\mathrm{LE}(k) = 0$ na daných datech.

# 3 Experimentální analýza

## 3.1 Použitá data

Pro experimentální analýzu byla použita jedna sada umělých dat. Umělá data byla použita z několika důvodů:

- Řízená generace dat umožňuje přesně anotovat kde jsou změny systému.
- U reálných dat není možné zajistit správnou velikost driftu.
- U reálných dat není možné zajistit přesné množství šumu.
- Reálná data je problematické sehnat ve velkém rozsahu tak, aby splňovala výše uvedené důvody v dostatečné míře.

Tyto umělé data byly vytvořeny skládáním tří různých časových řad:

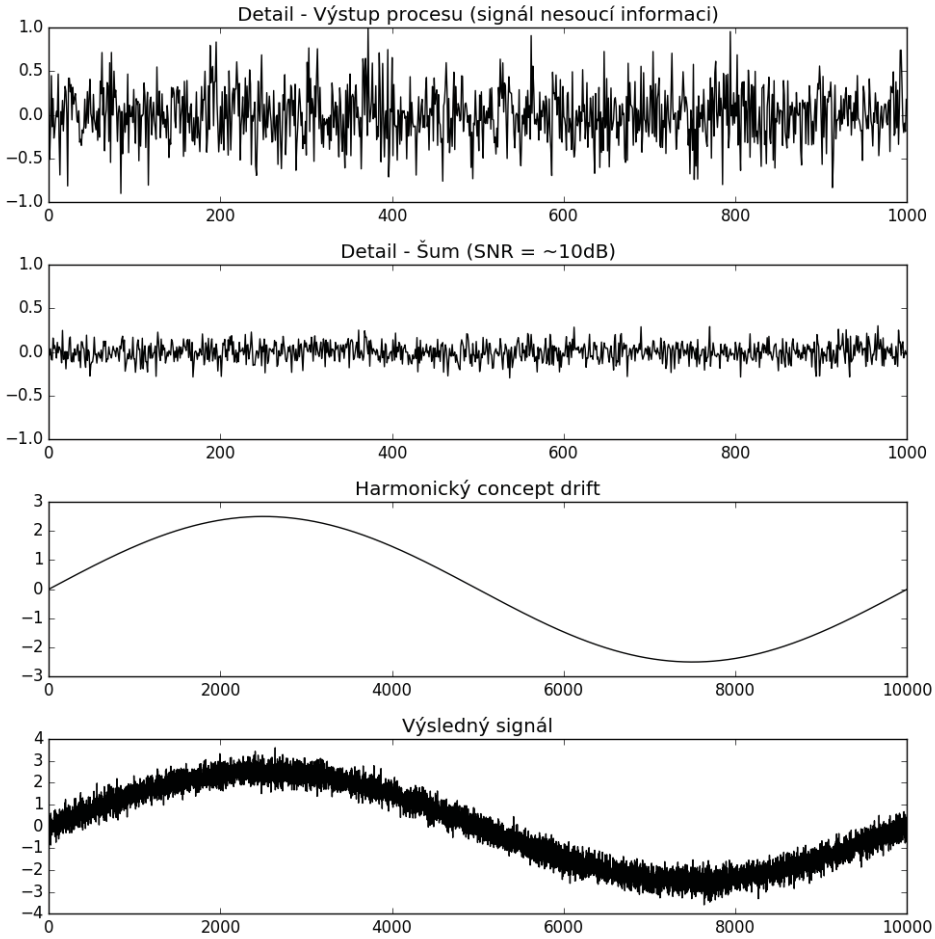
1. Výstup systému obsahující změny, které mají být detekovány.
2. Šum vytvořený tak, aby splňoval požadované parametry
3. *Concept drift* (proměnlivý offset), který byl vytvořený jako harmonická vlna s relativně pomalou periodou.

Signál $y(k)$, který nese informaci o změnách, byl vygenerován podle rovnice

$$y(k) = h_1(k)x_1(k) + \ldots + h_n(k)x_n(k), \tag{17}$$

kde $h_i(k)$ jsou parametry procesu a $x_i(k)$ jsou vstupní proměnné. Jako vstupní proměnné byli použito deset nezávislých časových řad bílého Gaussovského šumu s jednotkovou standardní deviací a nulovou střední hodnotou. Parametry procesu $h_i(k)$ byly náhodně změněny za jiné parametry každých 500 vzorků. Tyto změny byly ostré v jednom bodě. Nové parametry byly brány z normálního rozložení s standardní deviací 0.5 a nulovou střední hodnotou. celá data obsahují 500 takových změn ($250 \times 10^3$ vzorků). K vygenerovaným datům byl přičten bílý Gaussovský šum (*Additive white Gaussian noise* (AWGN)). Nakonec byl k datům přičten ještě *concept drift*. Ten byl získán vygenerováním harmonické vlny s periodou ($10 \times 10^3$ vzorků) vzorků a různou amplitudou. První dvě vlny ($20 \times 10^3$ vzorků) byly použity k natrénování adaptivních filtrů to ustáleného stavu, zbytek dat byl pak následně použit ke srovnání metod.

Pro lepší představu o skládání dat je přiložen obrázek 1, kde je ukázáno složení ukázkového signálu výše popsaným způsobem (perioda driftu=10000, amplituda driftu=5, SNR přibližně 10dB).



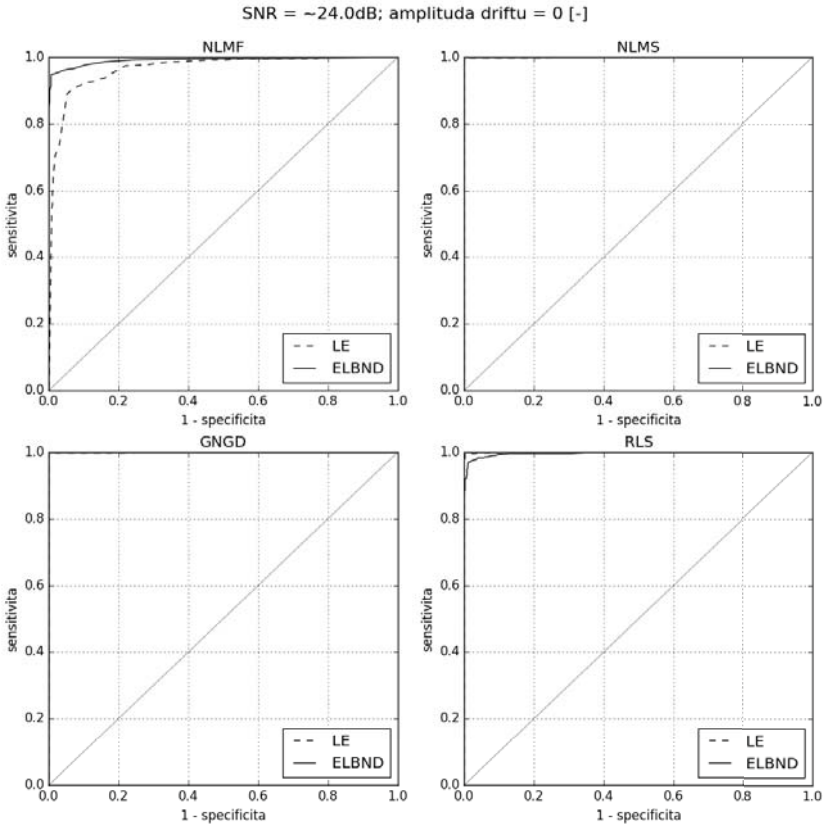

Obr. 1: Časové řady z kterých je složen referenční signál pro adaptivní algoritmy (měřený výstup simulovaného systému).

## 3.2 Výsledná přesnost detekce

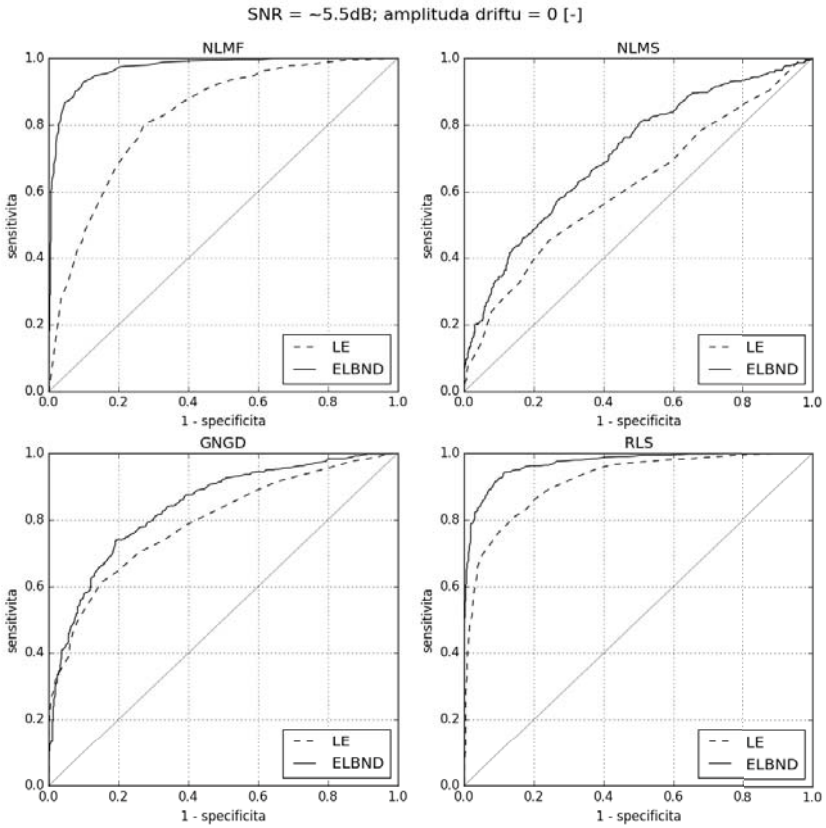
Jak již bylo zmíněno v sekci 2, jako metriky pro číselné srovnání získaných výsledků byli použity AUROC a maximální přesnost (MAX ACC). Samotné výsledné hodnoty pro všechny testované nastavení je možné najít v Tab.1. Kompletní ROC křivky pro vizuální srovnání, z kterých bylo stanoveno AUROC jsou zobrazeny na obrázcích: Obr. 2, Obr. 3, Obr. 4, Obr. 5, Obr. 6 a Obr. 7. Výsledky patrné z obrázků a tabulky je možné shrnout do následujících bodů:

- Poznatek 1. Při vyšší úrovní šumu obecně dosahuje lepších výsledků ELBND než LE.
- Poznatek 2. Při vyšší úrovni driftu dosahuje lepších výsledků LE než ELBND.
- Poznatek 3. Adaptivní algoritmy NLMS a GNGD se zdají být vhodnější pro použití s detekcí pomocí LE, zatímco algoritmy RLS a NLMF se zdají být vhodnější pro ELBND.

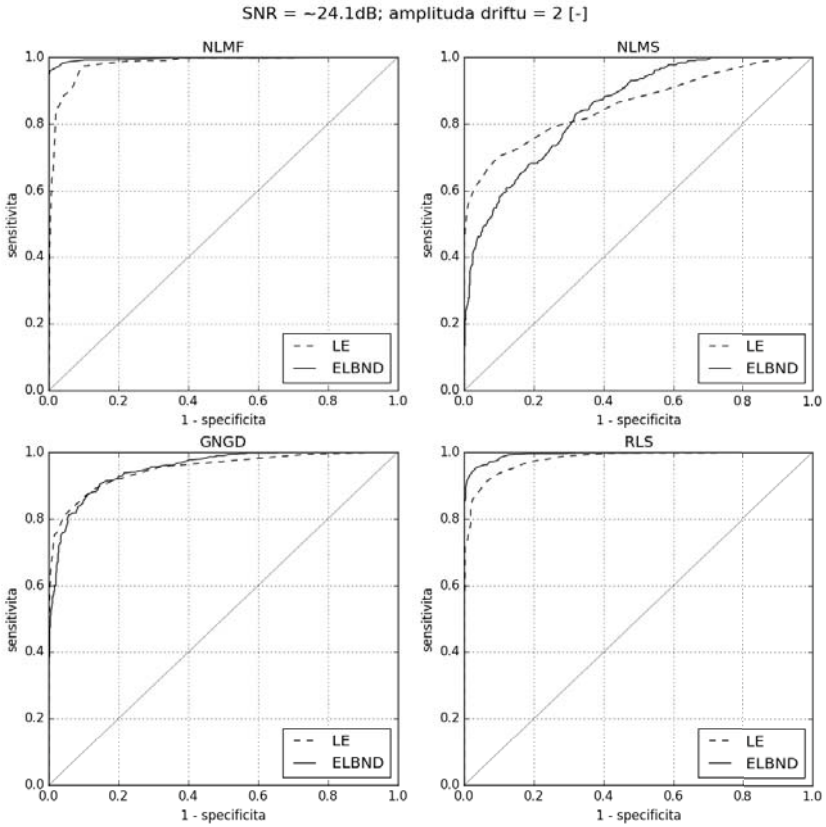
Získané výsledky (Poznatek 1., Poznatek 2.) potvrzují výsledky naší předchozí práce [11] a [3]. Poznatek 3. je nový, jelikož algoritmy GNDG a NLMF předtím nebyly nikdy testovány v tomto kontextu.



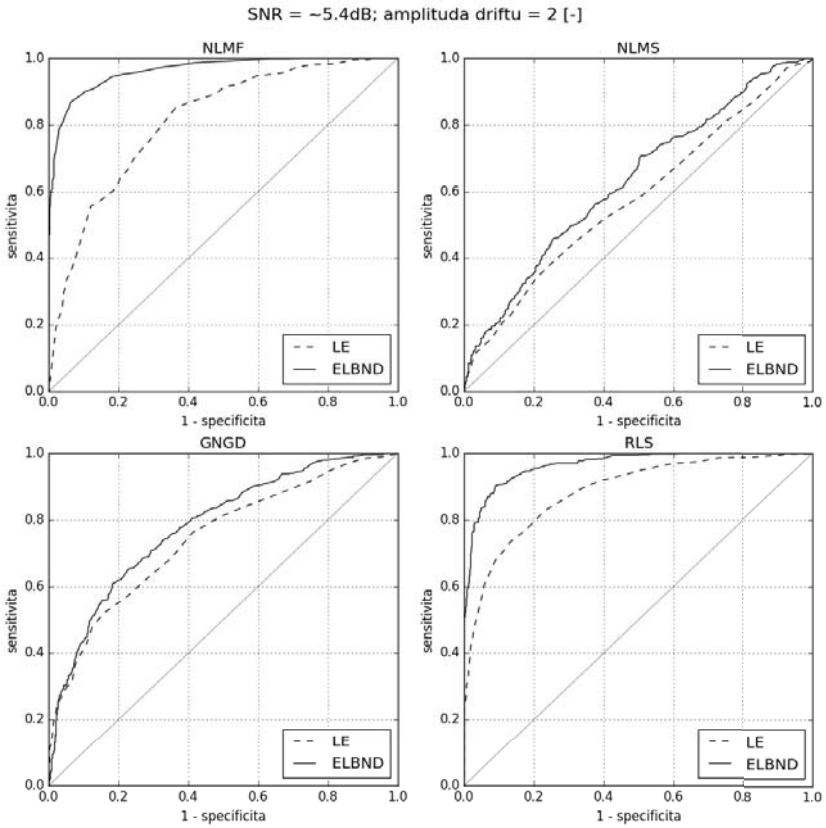

Obr. 2: ROC křivka pro výsledky na datech bez driftu a s nízkou úrovní šumu. Prázdné grafy představují výsledky s nulovou nebo téměř nulovou chybou.



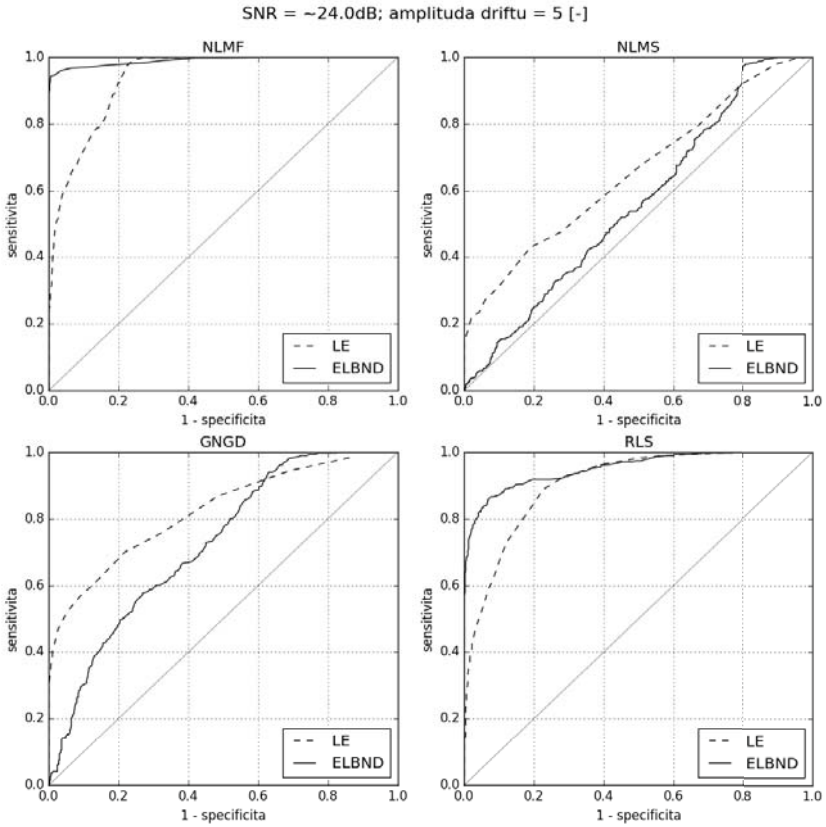

Obr. 3: ROC křivka pro výsledky na datech bez driftu a s vysokou úrovní šumu. Při této konfiguraci dosahuje ve všech případech lepších výsledků ELBND než LE.



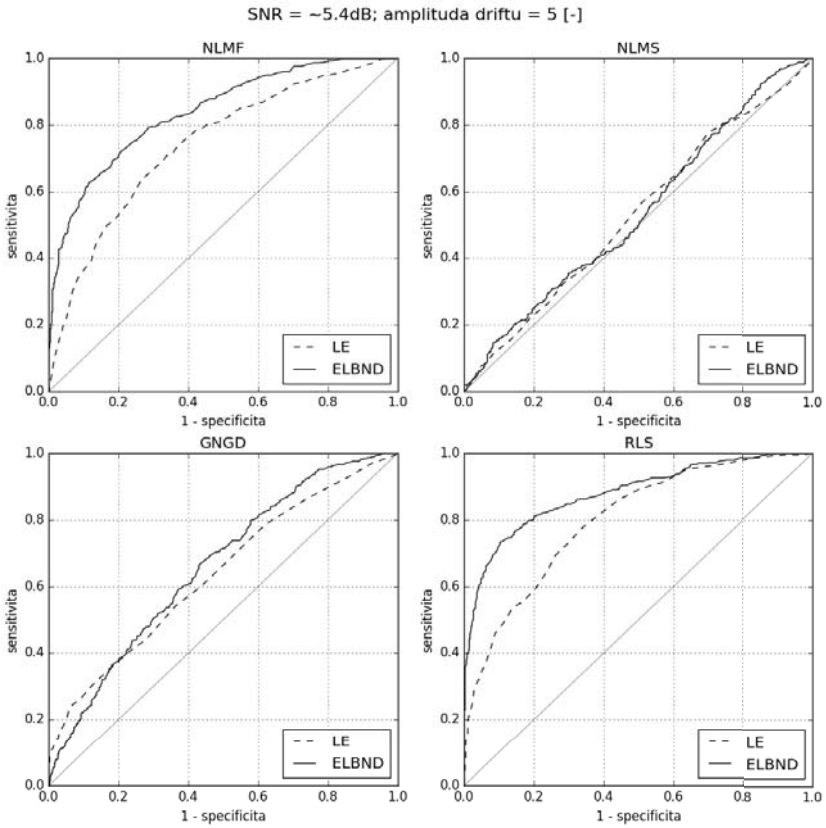

Obr. 4: ROC křivka pro výsledky na datech se středně velkým driftem a s nízkou úrovní šumu. Při této konfiguraci nelze jednoznačně říci zda je lepší LE nebo ELBND. Výsledky jsou závislé na použitém adaptivním algoritmu.




Obr. 5: ROC křivka pro výsledky na datech se středně velkým driftem a s vysokou úrovní šumu. Při této konfiguraci dosahuje lepších výsledků ELBND.




Obr. 6: ROC křivka pro výsledky na datech s velkým driftem a s nízkou úrovní šumu. Při této konfiguraci nelze jednoznačně říci zda je lepší LE nebo ELBND. Výsledky jsou závislé na použitém adaptivním algoritmu.




Obr. 7: ROC křivka pro výsledky na datech s velkým driftem a s vysokou úrovní šumu. Při této konfiguraci jsou výsledky podobné, s tím že pro některé adaptivní algoritmy funguje lépe ELBND.

Tab. 1: Tabulka výsledných všech výsledných hodnot přenosti detekce s EL a ELBND.

| Adaptace | Detekce | AUROC [%] | MAX ACC [%] |
|---|---|---|---|
| SNR = 5.4dB; amplituda driftu = 5 [-] | | | |
| NLMF | LE | 73.9 | 68.6 |
| | ELBND | 83.7 | 75.7 |
| NLMS | LE | 53.1 | 53.9 |
| | ELBND | 53.6 | 53.5 |
| GNGD | LE | 63.6 | 58.9 |
| | ELBND | 65.8 | 61.7 |
| RLS | LE | 79.6 | 71.9 |
| | ELBND | 87.6 | 81.5 |
| SNR = 24.0dB; amplituda driftu = 5 [-] | | | |
| NLMF | LE | 93.4 | 87.6 |
| | ELBND | 99.0 | 96.9 |
| NLMS | LE | 65.5 | 62.0 |
| | ELBND | 56.2 | 58.6 |
| GNGD | LE | 81.9 | 74.1 |
| | ELBND | 72.0 | 65.5 |
| RLS | LE | 90.3 | 83.1 |
| | ELBND | 95.0 | 89.5 |
| SNR = 5.4dB; amplituda driftu = 2 [-] | | | |
| NLMF | LE | 81.0 | 74.4 |
| | ELBND | 96.1 | 90.3 |
| NLMS | LE | 57.9 | 56.7 |
| | ELBND | 63.1 | 60.2 |
| GNGD | LE | 74.2 | 68.1 |
| | ELBND | 77.9 | 71.3 |
| RLS | LE | 88.3 | 80.1 |
| | ELBND | 96.4 | 90.6 |
| SNR = 24.1dB; amplituda driftu = 2 [-] | | | |
| NLMF | LE | 97.4 | 94.0 |
| | ELBND | 99.5 | 97.9 |
| NLMS | LE | 85.8 | 80.2 |
| | ELBND | 84.7 | 75.5 |
| GNGD | LE | 94.8 | 88.3 |
| | ELBND | 94.9 | 88.0 |
| RLS | LE | 97.7 | 92.4 |
| | ELBND | 99.3 | 96.0 |
| SNR = 5.5dB; amplituda driftu = 0 [-] | | | |
| NLMF | LE | 82.3 | 76.2 |
| | ELBND | 96.7 | 91.5 |
| NLMS | LE | 61.5 | 60.5 |
| | ELBND | 71.3 | 65.3 |
| GNGD | LE | 79.1 | 73.1 |
| | ELBND | 83.8 | 77.3 |
| RLS | LE | 91.5 | 83.3 |
| | ELBND | 96.8 | 91.4 |
| SNR = 24.0dB; amplituda driftu = 0 [-] | | | |
| NLMF | LE | 96.1 | 91.9 |
| | ELBND | 99.0 | 97.1 |
| NLMS | LE | 99.9 | 99.9 |
| | ELBND | 100.0 | 100.0 |
| GNGD | LE | 100.0 | 99.9 |
| | ELBND | 100.0 | 100.0 |
| RLS | LE | 100.0 | 99.7 |
| | ELBND | 99.6 | 97.9 |

### 3.3 Časová náročnost

Oba algoritmy EL i ELBND mohou využívat parametry učících algoritmů ve stejné podobě. Tak to bylo realizované i v této práci. Z tohoto důvodu nemá cenu zahrnovat výpočetní náročnost užitých adaptivních algoritmů do této studie. Samotná výpočetní náročnost ELBND a LE je závislá na implementaci nižších funkcí, které využívají (průměrná hodnota pole, dělení desetinných čísel, maximální hodnota pole). Z tohoto důvodu byla dána přednost měření rychlosti namísto určování počtu operací na iteraci. Každopádně už z pravidel které používají oba algoritmy pro svojí detekci je evidentně nižší výpočetní náročnost u algoritmu ELBND než u LE.

Měřené časy byly stanoveny na osobním počítači. Čas byl měřen kumulativně pro všechny simulace, s tím že pro každou simulaci byla změřena ELBND a LE před tím než se přešlo na další simulaci. Toto nastavení bylo provedeno proto, aby se rovnoměrně distribuovalo kolísání výkonu počítače mezi oba algoritmy (ve srovnáním s nastavením kdy by se provedly v jeden čas všechny simulace pro ELBND a následně v další čas všechny simulace pro LE). Naměřený kumulativní čas pro všechny simulace v této studii je 2.963s a pro ELBND a 184.631s pro LE. Pro dané podmínky ELBND dosahovala přibližně 62x vyšší rychlost. Je potřeba si ale uvědomit že tato rychlost závisí silně na zvoleném nastavení. Obecně lze předpokládat že užití kratšího okna pro porovnání vah u LE by výrazně snížilo rychlostní rozdíl. Na druhou stranu při užití delšího vstupního vektoru by se rozdíl v rychlosti navýšil.

## 4 Závěr a diskuze

V této práci byly porovnání algoritmy LE a ELBND pro detekci novosti na datech zatížených šumem a opakujícím se harmonickým driftem. Práce navázala na naší předchozí práci a potvrdila výsledky již dříve získané. V neposlední řadě tato práce poukázala na vliv volby adaptivního algoritmu na samotné výsledky detekce novosti. Podle získaných výsledků se zdá, že algoritmy NLMF a RLS fungují lépe s ELBND zatímco GNGD a NLMS pracují obecně lépe s LE. Tento rozdíl by se dal vysvětlit tím že NLMS a GNGD jsou gradientní metody, které jsou navzájem velice podobné. Algoritmus NLMF je také gradientní metoda, ale je od výše zmíněných výrazně odlišný tím, že ve svém učení dává větší důraz na chybu modelu ($e(k)^4$ namísto $e(k)^2$). Algoritmus RLS není gradientní a je zásadně jiný od všech ostatních. Tento poznatek je zajímavým námětem na další výzkum.

## Poděkování

Tento projekt byl podpořen grantem *SGS18/177/OHK2/3T/12*. Všechny simulace byly provedeny v jazyce *Python*. Zdrojové kódy je možné dostat na požádání od autora.

## Literatura

[1] Subutai Ahmad, Alexander Lavin, Scott Purdy, and Zuha Agha. Unsupervised real-time anomaly detection for streaming data. *Neurocomputing*, 262:134 – 147, 2017.

[2] Marco AF Pimentel, David A Clifton, Lei Clifton, and Lionel Tarassenko. A review of novelty detection. *Signal Processing*, 99:215–249, 2014.

[3] Matous Cejnek and Ivo Bukovsky. Concept drift robust adaptive novelty detection for data streams. *Neurocomputing*, 2018.

[4] Ivo Bukovsky. Learning entropy: Multiscale measure for incremental learning. *Entropy*, 15(10):4159–4187, 2013.

[5] Jeffrey C Schlimmer and Richard H Granger. Beyond incremental processing: Tracking concept drift. In *AAAI*, pages 502–507, 1986.

[6] Ali H Sayed. *Fundamentals of adaptive filtering*. John Wiley & Sons, 2003.

[7] V Sreenivasa Arun Kumar et al. Comparison of stable nlmf and nlms algorithms for adaptive noise cancellation in ecg signal with gaussian, binary and uniform signals as inputs. *International Journal of Engineering Research and Applications*, 4(8):28–33, 2014.

[8] Victor H Nascimento and José Carlos M Bermudez. Probability of divergence for the least-mean fourth algorithm. *IEEE transactions on signal processing*, 54(4):1376–1385, 2006.

[9] Pedro Inacio Hubscher and José Carlos M Bermudez. An improved statistical analysis of the least mean fourth (lmf) adaptive algorithm. *IEEE transactions on Signal Processing*, 51(3):664–671, 2003.

[10] Danilo P Mandic. A generalized normalized gradient descent algorithm. *IEEE signal processing letters*, 11(2):115–118, 2004.

[11] Matous Cejnek and Ivo Bukovsky. Influence of type and level of noise on the performance of an adaptive novelty detector. In *2017 IEEE 16th International Conference on Cognitive Informatics & Cognitive Computing (ICCI* CC)*, pages 373–377. IEEE, 2017.

# RYCHLOST ADAPTIVNÍCH ALGORITMŮ PRO DETEKCE NOVOSTI

***Matouš Cejnek, Adam Peichl***

***Ústav přístrojové a řídicí techniky, Fakulta strojní, ČVUT v Praze, adam.peichl@fs.cvut.cz***

*Abstrakt: Tento článek se zabývá analýzou a porovnáním rychlosti několika adaptivní algoritmů (ELBND, LE, MD, FD). Rychlost je klíčová vlastnost algoritmů pro detekci novosti, které jsou používány pro zpracování dat v reálném čase. Adaptivní detekce novosti je pro procesy měřené v reálném čase zajímavá speciálně díky své robustnosti proti vysoké ne-stacionaritě měřených dat. Zkoumané algoritmy jsou v tomto článku analyzovány teoreticky pomocí prostředků asymptotické složitosti a získané závěry jsou validovány experimentálně na testovacích datech. Získané výsledky ukazují, že rozdíly v časové náročnosti jednotlivých algoritmů nejsou zanedbatelné. Zatímco ELBDN a LE mají lineární časovou složitost s malými multiplikativními a aditivními konstantami, MD vykazuje vlastnosti časové složitosti kvadratické a FD má linární časovou složitost s podstatně vyššími multiplikativními a aditivními konstantami než ELBDN a LE.*
*Klíčová slova: detekce novosti, asymptotická složitost, adaptivní algoritmy*

*Abstract: This paper deals with the analysis and comparison of speeds of several adaptive algorithms (ELBND, LE, MD, FD). Novelty detection algorithms are used for real-time data processing, thus speed is a key atribute. Adaptive novelty detection is especially interesting for real-time measured processes due to its robustness against high non-stationarity of measured data. The studied algorithms are analyzed theoretically through time complexity and results are validated experimentaly on testing data. Obtained results shows, that distinctions in time complexity between algorithms are not negligible. While ELBDN and LE have linear time complexity with small multiplicative and additive constants, MD has quadratic time complexity and FD has linear time complexity with substantially higher multiplicative and additive constants than ELBDN and LE.*
*Keywords: Novelty detection, time complexity, adaptive algorithms*

## 1 Úvod

Detekce novosti (*novelty detection*) je téma z oblasti strojového učení (*machine learning*), které se zabývá detekci neočekávaných stavů. Tyto stavy můžou představovat různé anomálie, které představují nestandardní chování sledovaného procesu. Takové chování je potřeba často detekovat, aby mohl být spuštěn krizový scénář reagující vhodným způsobem na daný neočekávaný stav. Toto téma je dnes obzvláště zajímavé, díky vysokému nárůstu procesů, jenž je potřeba monitorovat v reálném čase. Zdrojem těchto procesů jsou často moderní paradigmata jako Internet of Things (IoT) a Industry 4.0. Tyto nové zdroje monitorovaných procesů jsou příležitosti pro uplatnění strojového učení a přinášejí řadu nových výzev, jenž ještě nebyly spolehlivě vyřešeny. Jedna z těchto výzev, jenž online procesy přináší, je nutnost vysoké rychlosti při udržení dostatečné robustnosti proti různým variantám šumu v datech. Nejčasnější oblasti kde jsou tyto algoritmy nasazeny jsou: sledování průmyslových procesů, monitoring lékařských signálů, detekce událostí pro automatické obchodování a detekce webových útočníků. S ohledem na charakteristiku těchto oblastí, je evidentní že ve spoustě případů je potřeba, aby algoritmy pracovaly dostatečně rychle - stovky až tisíce vzorků za sekundu.

Velké množství doposud publikovaných algoritmů určených pro detekci novosti nesplňuje alespoň jednu ze základních podmínek zpracování dat v reálném čase - nejsou dostatečně rychlé, nebo nejsou dostatečně robustní. Algoritmy jenž jsou studovány v této práci mají zajištěnou jistou míru robustnosti díky své adaptivní podstatě. Adaptace je totiž intuitivní způsob jak kompenzovat offset nebo jinou postupně vznikající nerovnováhu v datech [1, 2, 3]. Jejich rychlost ale zatím nebyla studována detailněji. Z těchto důvodů jsou výsledky této práce přínosné pro budoucí uživatele těchto algoritmů. Výpočetní rychlost algoritmu strojového učení (detekce novosti v tomto

případě) záleží na množství faktorů. Některé faktory je možné ovlivnit a jiné ne. Obvykle ovlivnitelné faktory rychlosti procesu jsou hardware a software. I když i zde jsou často limitace dány možnostmi daného pracoviště a nebo místa nasazení daného procesu. Vcelku neovlivnitelný faktor daného procesu je složitost použitého algoritmu ve své podstatě, respektive jeho optimalizovaná podoba.

V tomto článku se pokoušíme testované algoritmy porovnat jak analyticky - za použití teoretických konceptů z oblasti asymptotické složitosti [4], tak experimentálně pomocí testovacího frameworku, který jsme zhotovili pro daný účel. Získané výsledky o implementační složitosti a chování adaptivní algoritmů jsou cenné poznatky užitečné pro každého, kdo by chtěl zkoumané algoritmy použít v libovolné reálné aplikaci.

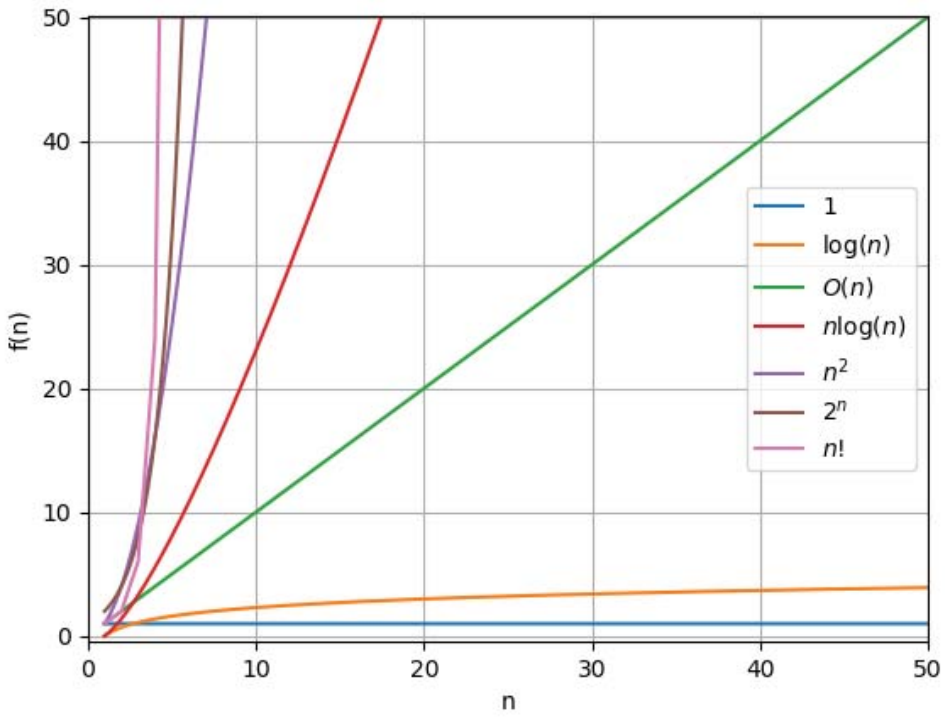
# 2 Metody

V této sekci jsou představeny postupy a metody, jenž jsou použity v tomto článku. V podsekci 2.1 je představen koncept, který je použit k analytickému porovnání testovaných algoritmů. V podsekci 2.3 jsou představy zkoumané metody adaptivní detekce novosti, jmenovitě jejich algoritmy včetně jejich složitostí. Následně v podsekci 2.2 jazyk Python, v kterém jsou implementovány všechny experimenty provedené v tomto článku.

## 2.1 Asymptotická složitost

Asymptotická složitost [4] je způsob dělení algoritmů podle operační náročnost algoritmu. Asymptotická složitost algoritmu vypovídá o tom, jakým způsobem se bude chovat algoritmus v závislosti na změně rozsahu vstupních dat. Zapisuje se pomocí Landauovy notace (Omikron notace) jako $O(f(n))$, kde $n$ je počet vstupních dat. Asymptotická složitost je funkce $f : \mathbb{N} \longrightarrow \mathbb{N}$, která vyjadřuje vztah mezi velikostní vstupních dat a množstvím spotřebovaného výpočetního času. V našem případě předpokládáme, že každá elementární aritmetická operace (násobení, sčítání, . . . ) spotřebuje předem dané (konstantní) množství času.

V našem případě budeme používat horní odhad složitosti $O(f(n))$, což je nejhorší možná složitost[1]. Bude tedy platit, že náš algoritmus proběhne asymptoticky rychleji nebo při nejhorším stejně rychle, jako náš odhad.




Obr. 1: Porovnání vybraných tříd asymptotických složitostí. Vysvětlení k legendě je možné najít v Tab. 1

[1]Složitost pro nejhorší možný vstup.

Tab. 1: Vybrané třídy složitosti a jejich typické příklady

| O-notace složitosti | název | příklad |
|---|---|---|
| $O(1)$ | konstantní | výběr prvku z pole |
| $O(\log n)$ | logaritmická | hledání v seřazeném poli |
| $O(n)$ | lineární | hledání minima, maxima |
| $O(n \log n)$ | lineárně logaritmická | quicksort |
| $O(n^2)$ | kvadratická | bubblesort |
| $O(2^n)$ | exponenciální | řešení problému obchodního cestujícího |
| $O(n!)$ | faktoriální | brute force řešení problému obchodního cestujícího |

V této sekci jsou vysvětleny metody použité v této studii, včetně postupu křížové validace, která byla použita pro vyhodnocení výsledků této studie.

## 2.2 Jazyk Python

Python je vysokoúrovňový skriptovací programovací jazyk, který v roce 1991 navrhl Guido van Rossum [5]. Nejrozšířenější implementace jazyka Python je v jazyce C (často označováno jako CPython). Z tohoto důvodu je výkon Pythonu silně spjatý s výkonem jazyka C. Pro vyšší optimalizaci výpočetně náročných úkolů (operace s maticemi atp.) jsou pro Python dostupné knihovny, které umožňují tyto operace provádět přímo v jazyce C tak, že Python figuruje pouze jako klient. Pro implementaci algoritmů v tomto článku byla použita knihovna Numpy [6], která je považována za standard pro výpočetní operace v Pythonu. Několik příkladů operací s polem v Pythonu (Python list) včetně jejich asymptotické složitosti je v Tab. 2.

Tab. 2: Asymptotické složitosti pro datový typ list v jazyce Python

| Operace | Příklad | Třída asymptotické složitosti |
|---|---|---|
| index | pole[i] | $O(1)$ |
| přiřazení | pole[i] = None | $O(1)$ |
| délka | len(pole) | $O(1)$ |
| append | pole.append(False) | $O(1)$ |
| pop | pole.pop() | $O(1)$ |
| clear | pole.clear() | $O(1)$ |
| řez | pole[a:b] | $O(b-a)$ |
| extend | pole.extend(l) | $O(len(l))$ |
| construction | list(l) | $O(len(l))$ |

## 2.3 Metody detekce novosti

### 2.3.1 Error and learning based novelty detection (ELBND)

Metoda ELBND detekuje novost základě okamžitého přírůstku adaptivních vah a chyby podle následujícího pravidla

$$\mathrm{ELBND}(k) = \Delta \mathbf{w}(k) e(k). \tag{1}$$

Všimněte si, že v rovnici 1 není k výpočtu použity žádné předchozí hodnoty, ale pouze okamžité hodnoty. Výstup 1 je vektor, který popisuje míru novosti pro každou adaptivní váhu zvlášť v daném diskrétním čase $k$. Tento vektor lze dále redukovat na jednu hodnotu, pro jednodušší vyhodnocení novosti v daném vzorku. Běžný způsob jak získat z vektoru $\mathrm{ELBND}(k)$ skalár $\mathrm{elbnd}(k)$ je funkce maxima z absolutních hodnot

$$\mathrm{elbnd}(k) = \max |\mathrm{ELBND}(k)|. \tag{2}$$

Všimněte si, že tento způsob výpočtu novosti nepotřebuje žádné další parametry, takže problém s optimalizací metody je minimalizován. Na druhou stranu je nutné podotknout, že výstup metody je silně závislý na adaptivním algoritmu. Touto problematikou se více zabývala například práce [7]. Tento algoritmus je zde posuzován na základě jeho implementace v knihovně Padasip [8]

Asymptotická složitost tohoto algoritmu je představena v tabulce 3. Jak je z tabulky patrné, žádný krok nemá vyšší asymptotickou složitost než $O(n)$, můžeme tedy prohlásit, že algoritmus má *lineární* asymptotickou složitost.

Tab. 3: Časová náročnost a počet operací pro jeden krok algoritmu ELBND ($n$ je počet adaptivních parametrů).

| pořadí | operace | složitost | sčítání | násobení | poznámka |
|---|---|---|---|---|---|
| 1. | $o_1 = \mathbf{\Delta w(k)}e$ | $O(n)$ | 0 | $n$ | - |
| 2. | $o_2 = \lvert o_1 \rvert$ | $O(n)$ | 0 | 0 | abs() |
| 3. | $\max(o_2)$ | $O(n)$ | 0 | 0 | max() |

### 2.3.2 Approximate Individual Sample Learning Entropy (AISLE)

Approximate Individual Sample Learning Entropy [9] je často označována jen jako Learning Entropy. Výpočet AISLE pro každý vzorek je dán následujícím pravidlem:

$$\text{AISLE}(k) = \frac{1}{n \cdot n_\alpha} \sum f(\Delta w_i(k), \alpha) \; ; \; \forall \alpha \in \boldsymbol{\alpha} \, , \tag{3}$$

kde $n$ je počet adaptivních vah a $n_\alpha$ je počet citlivostí detekce, které si volí uživatel

$$\boldsymbol{\alpha} = [\alpha_1, \alpha_2, \ldots, \alpha_{n_\alpha}] \; ; \; \alpha_1 < \alpha_2 < \ldots < \alpha_{n_\alpha} . \tag{4}$$

Funkce $f(\Delta w_i(k), \alpha)$ je definována jako

$$f(\Delta w_i(k), \alpha) = \begin{cases} 1, & \text{if } |\Delta w_i(k)| > \alpha \cdot \overline{|\Delta w_{Mi}(k)|} \\ 0, & \text{v ostatních případech} \end{cases} \tag{5}$$

kde $\overline{|\Delta w_{Mi}(k)|}$ je střední hodnota okna použitého pro výpočet AISLE. Tato velikost okna $m$ by měla být zvolena s ohledem na možnou periodicitu v datech [9]. Další parametr volený uživatelem je počet citlivostí pro detekci $\alpha$ a jejich hodnoty. Tento algoritmus je zde posuzován na základě jeho implementace v knihovně Padasip [8].

Asymptotická složitost tohoto algoritmu je představena v tabulce 4. Jak je z tabulky patrné, žádný krok nemá vyšší asymptotickou složitost než $O(n, n_\alpha)$, můžeme tedy prohlásit, že algoritmus má *lineární* asymptotickou složitost.

Tab. 4: Časová náročnost a počet operací pro jeden krok algoritmu ELBND ($n$ je počet adaptivních parametrů a $n_\alpha$ je počet citlivostí detekce $\alpha$).

| pořadí | operace | složitost | sčítání | násobení | poznámka |
|---|---|---|---|---|---|
| 1. | $o_1 = \lvert \mathbf{\Delta W_M(k)} \rvert$ | $O(n)$ | 0 | 0 | abs() |
| 2. | $o_2 = \bar{o_1}$ | $O(n, m)$ | $m \cdot n$ | $n$ | - |
| 3. | $o_3 = \lvert \mathbf{\Delta w_i(k)} \rvert$ | $O(n)$ | 0 | 0 | abs() |
| 4. | $o_4 = \alpha \cdot o_2$ | $O(n, n_\alpha)$ | 0 | $n \cdot n_\alpha$ | - |
| 5. | $o_5 = (o_3 > o_4)$ | $O(n, n_\alpha)$ | 0 | 0 | - |
| 6. | $o_6 = \sum o_5$ | $O(n, n_\alpha)$ | $n_\alpha$ | $(n_\alpha \cdot n) - 1$ | - |
| 7. | $o_6 / (n \cdot n_\alpha)$ | $O(2)$ | 0 | 2 | - |

### 2.3.3 Mahanobilis Distance (MD)

Tato metoda využívá k dektekci novosti Mahanobilisovu vzdálenost[10], což je vzdálenost, která bere v úvahu závislosti mezi jednotlivými veličinami[2].
Nechť je naše pozorování adaptivních parametrů ve vzorku k:

$$\mathbf{\Delta w(k)} = [\Delta w_0(k), \Delta w_1(k), \ldots, \Delta w_n(k)] \tag{6}$$

A množina předchozích pozorování $\mathbf{\Omega(k)}$:

$$\mathbf{\Omega(k)} = \begin{bmatrix} \mathbf{\Delta w(k-1)} \\ \mathbf{\Delta w(k-2)} \\ \mathbf{\Delta w(k-3)} \\ \vdots \\ \mathbf{\Delta w(k-M)} \end{bmatrix} \tag{7}$$

[2]Kovarianční matici.

Dále nechť $\mathbf{S}$ je kovarianční matice $\mathbf{\Omega(k)}$:

$$\mathbf{S} = \mathbf{\Omega^T(k)\Omega(k)} \tag{8}$$

A $\overline{\mathbf{\Delta w_M(k)}}$ je těžiště $\mathbf{\Omega(k)}$, tedy vektor průměrných hodnot každého sloupce $\mathbf{\Omega(k)}$:

$$\overline{\mathbf{\Delta w_M(k)}} = \frac{\sum_{i=0}^{M-1} \mathbf{\Omega_i(k)}}{M} = \frac{\sum_{j=1}^{M} \mathbf{\Delta w(k-j)}}{M} \tag{9}$$

Potom $D_M$ je mahanobilisova vzdálenost $\mathbf{w(k)}$ a $\mathbf{\Omega(k)}$:

$$D_M\left(\mathbf{\Delta w(k)}, \mathbf{\Omega(k)}\right) = \sqrt{\left(\mathbf{\Delta w(k)} - \overline{\mathbf{\Delta w(k)}}_\mathbf{M}\right) \cdot \mathbf{S^{-1}} \cdot \left(\mathbf{\Delta w(k)^T} - \overline{\mathbf{\Delta w(k)}}_\mathbf{M}^\mathbf{T}\right)} \tag{10}$$

Zjednodušený kód v jazyce python[3]:

```
import numpy as np

def mahanobilis_distance(omega,dw):
    S = np.cov(omega,rowvar=false) # ekvivalentni np.dot(omega.T,omega)
    S_1 = np.linalg.inv(S) # inversni matice, v realnem kodu se obcas pouziva Moore-Penrose
                                                    inverse
    mu = np.mean(omega,axis=0) # teziste omega
    mu[:np.newaxis] # doplneni rozmeru vektoru teziste z (n,) na (n,1)
    mahanobilis = np.sqrt(np.abs(np.matmul(np.matmul(dw.T-mu.T,S_1),(dw-mu))))
    return mahanobilis
```

Asymptotická složitost tohoto algoritmu je představena v tabulce 5. Z pohledu asymptotické složitosti jsou zajímavé řádky 2 a 3, algoritmus má v každém případě *polynominální* složitost a vzhledem k tomu, že s nejvyšší pravděpodobností bude platit $M > n$, bude výsledná asymptotická složitost $O(Mn^2)$.

Tab. 5: Časová náročnost a počet operací pro jeden krok algoritmu MD(n je počet adaptivních parametrů, M je počet řádků matice Ω)

| pořadí | operace | složitost | sčítání | násobení | poznámka |
|---|---|---|---|---|---|
| 1. | $o_1 = \overline{\mathbf{\Delta w_M(k)}}$ | $O(Mn)$ | $(M-1)n$ | $n$ | těžiště |
| 2. | $o_2 = \mathbf{\Omega^T\Omega}$ | $O(Mn^2)$ | $(M-1)n$ | $Mn^2$ | kovarianční matice |
| 3. | $o_3 = (o_2)^{-1}$ | $O(n^3)$ | - | - | inverzní matice |
| 4. | $o_4 = \mathbf{\Delta w(k)} - o_1$ | $O(n)$ | $n$ | - | - |
| 5. | $o_5 = o_4 o_3$ | $O(n^2)$ | $(n-1)n$ | $n^2$ | - |
| 6. | $o_6 = (o_4)^T$ | $O(1)$ | - | - | transpozice vektoru |
| 7. | $o_7 = o_5 o_6$ | $O(n^2)$ | $(n-1)n$ | $n^2$ | - |
| 8. | $D_M = \sqrt{o_7}$ | $O(\sqrt{n})$ | - | - | odmocnina |

### 2.3.4 Fuzzy Density (FD)

Tato metoda detekuje novost na základě odhadu funkce hustoty pravděpodobnosti pomocí fuzzy množin[11] podle následujícího předpisu:

Nechť je naše pozorování adaptivních parametrů ve vzorku k:

$$\mathbf{\Delta w(k)} = [\Delta w_0(k), \Delta w_1(k), \ldots, \Delta w_n(k)] \tag{11}$$

A množina předchozích pozorování $\mathbf{\Omega(k)}$:

$$\mathbf{\Omega(k)} = \begin{bmatrix} \mathbf{\Delta w(k-1)} \\ \mathbf{\Delta w(k-2)} \\ \mathbf{\Delta w(k-3)} \\ \vdots \\ \mathbf{\Delta w(k-M)} \end{bmatrix} \tag{12}$$

Potom fuzzy hustota $\Phi_\mathbf{\Omega}(\mathbf{X})$:

$$\Phi_\mathbf{\Omega}(\mathbf{X}) = \frac{1}{M} \sum_{\forall \Omega_i} \prod_{\forall x_j \in \mathbf{X}} \mu_G(x_j, \Omega_{i,j}, \sigma_j) \tag{13}$$

[3]Byly odstraněny některé operace, které nejsou uvedeny v popisu algoritmu výšše: např. transformace 1D vektoru o délce n ja matici o rozměrech (n,1)

Kde funkce $\mu_G(x_j, \Omega_{i,j}, \sigma_j)$ je gaussovská funkce příslušnosti:

$$\mu_G(x_j, \Omega_{i,j}, \sigma_j) = e^{-\left(\frac{x_j - \Omega_{i,j}}{\sigma_j}\right)^2} \tag{14}$$

Přičemž platí, že $\sigma_j$ je *average distance* j-tého sloupce $\Omega$:

$$\sigma_j = f_{AD} = \frac{1}{M} \sum_{k=0}^{M-1} \sum_{l=l+1}^{M-1} |\Omega_{k,j} - \Omega_{l,j}| \tag{15}$$

Zjednodušený kód v jazyce python:

```
import numpy as np

def average_norm(omega):
    n = np.shape(omega)[1]
    sigma = []
    for i in range(n):
        X, Y = np.meshgrid(omega[:,i],omega[:,i])
        norm = np.sum(np.abs(X-Y))/(2*n*(n-1))
        sigma.append(norm)
    return sigma


def gaussian(x,center,sigma):
    return np.exp(-((x-center)/sigma)**2)

def fuzzy_density(omega,dw,M):
    sigma = average_norm(omega)
    value = 0
    for row in omega:
        p = 1
        for k in range(len(row)):
            p = p*gaussian(dw[k],row[k],sigma[k])
        value += p
    return value/M
```

Tab. 6: Časová náročnost a počet operací pro jeden krok algoritmu FD(n je počet adaptivních parametrů, M je počet řádků matice $\Omega$)

| pořadí | operace | složitost | sčítání | násobení | poznámka |
|---|---|---|---|---|---|
| 1. | $\sigma = f_{AD}(\mathbf{\Omega})$ | $O(nM^2)$ | $\frac{1}{2}M^2n$ | - | average distance |
| 2. | $\Phi_{\mathbf{\Omega}}(\mathbf{X})$ | $O(Mn)$ | $M(n+1)$ | $5Mn+1$ | fuzzy density |

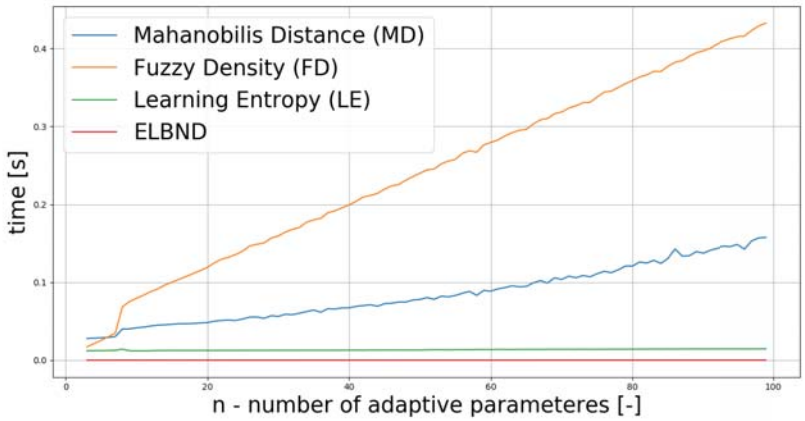
# 3 Experimentální analýza

Během experimentální analýzy, byl měřen čas na detekci novosti při použiti všech testovaných algoritmů. Adaptivní algoritmy pro detekci byly testovány v rozsahu 3-99 vstupů (parametr $n$), jelikož tento interval popisuje nejčastější použití daných algoritmů. Testovací data byla rozdělena do segmentů o délce 1000 vzorků. Každý segment byl změřen 1000x. Výsledky čas na segment se získal zprůměrováním těchto 1000 nezávislých testů. Každý jednotlivý test se provedl pro každý algoritmus předtím, než se postoupilo na další test, aby se rovnoměrně rozložilo případné kolísání výkonu testovacího hardware. Algoritmy, které používají paměť předchozích vektorů adaptivních vah, měli všichni shodně nastaveny délku paměti na 500 posledních vektorů.

Získané výsledky jsou zobrazeny v grafu na Obr. 2. Výsledky pro vybrané hodnoty parametru $n$ jsou číselně zaznamenány v Tab. 7.

# 4 Závěr

V toto článku byla porovnána časová náročnost čtyř různých algoritmů pro detekci novosti (ELBND, LE, FD, MD). Časová náročnost byla posuzována pomocí asymptotické složitosti algoritmů, počtu operací (násobení, adice) v jejich implementaci a také pomocí změření rychlosti v jazyce Python. Metody ELBND, LE a FD mají lineární asymptotickou složitost a metoda MD má kvadratickou asymptotickou složitost(Vzhledem k počtu adaptivních parametrů), což potvrdily výpočetní testy. Do budoucna by bylo zajímavé validovat výsledky i pomocí jiných implementačních jazyků a nástrojů.




Obr. 2: Výsledky - v tomto grafu je znázorněna závislost časové náročnosti jednotlivých algoritmů pro detekci novost a počtu adaptivních parametrů.

Tab. 7: Časová náročnost v ms v závislosti na parametru $n$ pro jednotlivé algoritmy.

| n | ELBND | LE | FD | MD |
|---|---|---|---|---|
| 3 | 0.040565 | 12.106009 | 17.114869 | 27.900899 |
| 13 | 0.057102 | 12.484739 | 91.514165 | 45.028556 |
| 23 | 0.067401 | 12.566594 | 132.167673 | 51.513525 |
| 33 | 0.078859 | 12.785707 | 170.275357 | 60.336528 |
| 43 | 0.091092 | 13.055424 | 211.303988 | 71.177681 |
| 53 | 0.102716 | 13.517995 | 251.945065 | 82.149545 |
| 63 | 0.112905 | 13.724406 | 291.961026 | 95.538772 |
| 73 | 0.128387 | 14.128523 | 330.522018 | 108.705011 |
| 83 | 0.140568 | 14.402364 | 370.774985 | 128.212707 |
| 93 | 0.152008 | 14.572935 | 409.528680 | 146.476234 |

## Poděkování

Tento projekt byl podpořen grantem *SGS18/177/OHK2/3T/12*. Všechny simulace byly provedeny v jazyce *Python*. Zdrojové kódy je možné dostat na požádání od autora.

## Literatura

[1] Terran Lane and Carla E Brodley. Approaches to online learning and concept drift for user identification in computer security. In *KDD*, pages 259–263, 1998.

[2] Jeffrey C Schlimmer and Richard H Granger. Beyond incremental processing: Tracking concept drift. In *AAAI*, pages 502–507, 1986.

[3] Leandro L Minku, Allan P White, and Xin Yao. The impact of diversity on online ensemble learning in the presence of concept drift. *IEEE Transactions on Knowledge and Data Engineering*, 22(5):730–742, 2010.

[4] Sanjeev Arora and Boaz Barak. *Computational complexity: a modern approach*. Cambridge University Press, 2009.

[5] Guido Van Rossum and Fred L Drake Jr. *Python tutorial*. Centrum voor Wiskunde en Informatica Amsterdam, The Netherlands, 1995.

[6] Stéfan van der Walt, S Chris Colbert, and Gael Varoquaux. The numpy array: a structure for efficient numerical computation. *Computing in Science & Engineering*, 13(2):22–30, 2011.

[7] Matous Cejnek and Ivo Bukovsky. Concept drift robust adaptive novelty detection for data streams. *Neurocomputing*, 2018.

[8] Matous Cejnek et al. Padasip: Python adaptive signal processing, 2016–. [Online; accessed 2018-06-25].

[9] Ivo Bukovsky. Learning entropy: Multiscale measure for incremental learning. *Entropy*, 15(10):4159–4187, 2013.

[10] Prasanta Chandra Mahalanobis. On the generalized distance in statistics. National Institute of Science of India, 1936.

[11] Mohsen Arefi, Reinhard Viertl, and S Mahmoud Taheri. Fuzzy density estimation. *Metrika*, 75(1):5–22, 2012.