



**Conference proceedings**  
**Sborník odborného semináře**

**Nové metody a postupy v oblasti přístrojové  
techniky, automatického řízení a informatiky 2019**  
**New Methods and Practices in the Instrumentation,  
Automatic Control and Informatics 2019**

**27. 5. – 29. 5. 2019, Zvíkovské Podhradí**

**ISBN 978-80-01-06617-1**

## OBSAH

<i>Mohammed Anas Hathif Sunkesula, Matouš Cejnek</i> <b>ON CORRELATION OF BITCOIN AND ALTCOINS</b> .....	3
<i>Haneen Farman Mohamed Mubarak</i> <b>ELECTRONICS &amp; CONTROL SYSTEM FOR LASER CUTTER</b> .....	11
<i>Mitja Avgunštinčič, Adam Peichl</i> <b>OPTICAL CHARACTER RECOGNITION FOR EXPIRATION DATES VALIDATION</b> .....	17
<i>Nina Kumer, Matous Cejnek, Adam Peichl</i> <b>INTERACTIVE JUPYTER NOTEBOOKS IN EDUCATION: CASE STUDY OF AN AIRCRAFT ENGINE COMPRESSOR</b> .....	24
<i>Marek Ciklamini</i> <b>SCALERS EFFECT ON PERFORMANCE OF STANDARD MACHINE LEARNING MODELS</b>	32
<i>Josef Kokeš</i> <b>TECHNOLOGIE BLOCKCHAIN A JEJÍ POUŽITÍ K ARBITRÁŽNÍM OBCHODŮM</b> .....	34
<i>David Janouch</i> <b>VÝHODY PARALELNÍCH ROBOTICKÝCH MANIPULÁTORŮ</b> .....	39
<i>Yan Shchankin</i> <b>MALTING PROCESS IN AN INDUSTRIAL PLANT</b> .....	44
<i>Zdeněk Novák</i> <b>ADAPTIVNÍ METODY PRO BEZSENZOROVÉ VEKTOROVÉ ŘÍZENÍ VYSOKORYCHLOSTNÍHO SYNCHRONNÍHO MOTORU S PERMANENTNÍMI MAGNETY</b>	51
<i>Vladimír Hlaváč</i> <b>POROVNÁNÍ ŘADICÍCH ALGORITMŮ</b> .....	62

Editor: Ing. Vladimír Hlaváč, Ph.D.  
Název díla: Nové metody a postupy v oblasti přístrojové techniky, automatického řízení a informatiky 2019  
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: 69 Pořadí vydání: 1  
Vydáno dne 31. 7. 2019 jako pdf soubor na CD.  
Dostupné na webových stránkách Ústavu přístrojové a řídicí techniky:  
<http://control.fs.cvut.cz/nmp>  
ISBN: 978-80-01-06617-1.

# ON CORRELATION OF BITCOIN AND ALTCOINS

*Mohammed Anas Hathif Sunkesula, Matouš Cejnek*

*Ústav přístrojové a řídicí techniky, ČVUT v Praze, Fakulta strojní, Praha*

*Abstrakt: V tomto projektu byla zkoumána korelace mezi kryptoměnami: Bitcoin, Monero, Ethereum a další. Korelace byla zkoumána pomocí Pearsonova a Spearmanova korelačního koeficientu. Bylo dohledáno, že v roce 2016 byla korelace mezi sledovanými kryptoměnami velmi nízká, skoro žádná. V roce 2017 korelace vzrostla. Následně v roce 2018 se korelace sledovaných kryptoměn výrazně zvýšila. V diskuzi tohoto článku jsou představeny a porovnány možné hypotézy proč tato korelace existuje.*  
*Klíčová slova: Bitcoin, kryptoměny, korelace*

*Abstract: In this Project, a research on topic of correlation analysis of Bitcoin, Monero, Ethereum and other top cryptocurrencies is reviewed and concluded. We investigate correlation coefficients of cryptocurrencies with two different approaches: Pearson and Spearman. It has been concluded that in 2016 the correlation between the selected cryptocurrencies was very weak - almost none, but in 2017 the correlation increased and became moderate positive. In 2018, almost all of the cryptocurrencies were highly correlated.*

*Keywords: Bitcoin, cryptocurrency, correlation*

## 1 Introduction

Cryptography derives from the Greek words *kryptos* (hidden, secret) and *graphein* (to write) [1]. Essentially is the method of sending a message in a concealed way to prevent other parties from reading it. The transmission of information becomes secure with the help of cryptography. Cryptography is useful for user authentication and in protecting the data from thieves.

Cryptocurrency (Herein referred to as cryptos) is a digital currency, based on principles of cryptography [1]. The main technology behind the cryptos is a blockchain [2]. A blockchain is essentially a distributed database of records, or public ledger of all transactions or digital events that have been executed and shared among participating parties. Each transaction in the public ledger is verified by consensus of a majority of the participants in the system. Once entered, information can never be erased. The blockchain contains a certain and verifiable record of every single transaction ever made. To use a basic analogy, it is easier to steal a cookie from a cookie jar, kept in a secluded place, than stealing the cookie from a cookie jar kept in a market place, being observed by thousands of people. Bitcoin (herein referred to as BTC) is the most popular example that is intrinsically tied to blockchain technology. It is also the most controversial one since it helps to enable a multibillion-dollar global market of anonymous transactions without any governmental control. Hence it has to deal with a number of regulatory issues involving national governments and financial institutions [3]. Due to the electronic nature of cryptos, they are extremely easy to use in different countries without having any technical restrictions. Though governments around the world have targeted crypto users (often under money-laundering regulations for individuals and financial regulations for would-be intermediaries), their success has been mixed. The protocols (crypto market) cannot be targeted or shut down; the best that can be done is to pinpoint prominent individual users, a drawn-out and expensive process. This is the reason that BTC has not succumbed to the same fate as the Liberty Dollar [4, 5].

Interesting feature of cryptos is lower transaction fees compared to the high fees that banks charge. [6]. Cryptos use cryptography for two major purposes. Firstly, to make sure that transactions are secured and secondly having control-govern of the monetary issuance.

BTC [1] is a digital currency and is probably one of the biggest discoveries in the financial system of the 21st century. It's production, storage, movement and all transactions with it, is made exclusively in electronic form between the participants of BTC network [7]. No country, government or bank produces or controls BTC.

As per One of the articles in CNBC [8], an analyst has suggested that the main reasons for different exchangers to have different rates on the cryptos are due to the fact that each exchange has different quantities of these cryptos. The smaller the demand in the market, the better the rates of the cryptos. According to the article, BTC's different prices were noted in one of the biggest crypto markets,

BTC Prices in different markets on 12th December 2017: Kraken - \$17,212; GDAX -\$17,150; Gemini - \$17,039; Bitfinex - \$16,957; Bitstamp - \$16,979.

Also one of the other reasons for this factor is that the prices for these currencies is not fixed so most of the other markets would be pricing the BTCs as per the flow of the demand. For example if one big market sets a price, then the other lower markets would try to adjust their prices accordingly but at the end of the day, it depends on how much supply each market has, that causes these change in the prices.

Another challenge is the inefficient way to manage a balance in different exchange markets. Traders who have a vast investment, practice the act of arbitrage, where the crypto is bought at a lower price and is sold in a different market for a higher price. Sometimes in order to make a profit, huge amount of transactions have to be made, which in the other hand consumes a lot of time, which can result in the change of the exchange rates too. So it can be quite risky to get into the act of arbitrage [8].

According to the data of BTC distribution, There are many addresses that have their own distinctive prices of the BTCs they are selling. This causes a lot of differences in the exchange rates of these cryptos in the market. Seeing this opportunity, consumers also try to practice the act of arbitrage, But this act involves a lot of time and risk. It is believed currently there are 23.7 million BTC addresses. But the active users are very less compared to this figure approx 3 to 6 million. This is found out through the survey by Cambridge center for alternative finance. Their justification to this figure is based on only the active users and not all those that hold an account for cryptos [9].

In this paper we study correlation in order to decide which cryptos one can invest in. For instance if a person chooses two cryptos of opposite correlation nature, the person may not loose much, as from either of the cryptos the person may gain some profit. Correlations help decide investors which cryptos to invest in and what is their behaviour going to be in near future in order to avoid huge loses.

## 2 Correlation of cryptos

In mathematics, specifically in statistics, we are interested in measuring and analyzing the degree of relationship between pairs of random variables. This statistical relationship can be examined through a statistical technique called correlation, that shows whether or not and how strong the variables are related to each other. The research of how variables are correlated is called correlation analysis. Correlation refers to the relationship between mean values and it helps us to understand two valuable things. First, we can measure and analyze the degree of relationship between a pair of variables and secondly it shows us a forecasting relationship that can be utilized in practice. However, the existence of the correlation is not adequate to deduce the existence of a causal relationship.

Another similarity measure is mutual information, [10] It is a distance between two probability distributions. Correlation is a linear distance between two random variables. The mutual information is between any two probabilities defined for a set of symbols, while there cannot be a correlation between symbols that cannot naturally be mapped into a R raised to N space. On the other hand, the mutual information does not make assumptions about some properties of the variables. If the data is smooth, correlation may tell more about them; for instance if their relationship is monotonic. If there are some prior information, then among both the methods we can be able to switch from one to another [11]. The equation for mutual information follows:

$$I(A; B) = \sum_{b \in B} \sum_{a \in A} p(a, b) * \log\left(\frac{p(a, b)}{p(a)p(b)}\right), \quad (1)$$

where  $A$  is the feature data and  $B$  is the output data [12]. The general rule is that the greater the mutual information, the more informative the feature. The price data of the other cryptos apart from BTC is collected, Once we collect this data, using Pearson correlation coefficient [13], we check how correlated the cryptos are. In the correlation method, If the correlation coefficient is +1 then both the cryptos are moving in the same direction, If the correlation coefficient is -1 then both the cryptos are moving in the opposite direction to each other. If it is 0, then there is no correlation between the two cryptos.

In Fig 1 we can see the daily percentage change price of BTC and Monero (hereafter referred as XMR)

It is widely understood that the use of larger samples in applications of factor analysis tends to provide results such that sample factor loadings are more precise estimates of population loadings and are also more stable, or less variable, across repeated sampling.

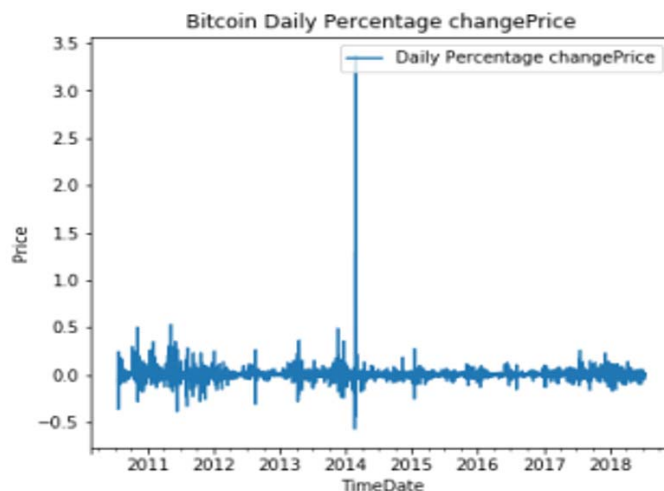


Fig. 1: Daily percentage change price of BTC and XMR [1]



Fig. 2: Prices of cryptos [1]

Summary of hypotheses provides a basis for the following hypotheses about effects of sample size in factor analysis:

1. As  $N$  increases, sampling error will be reduced, and sample factor analysis solutions will be more stable and will more accurately recover the true population structure.
2. Quality of factor analysis solutions will improve as communalities increase. In addition, as communalities increase, the influence of sample size on quality of solutions will decline. When communalities are all high, sample size will have relatively little impact on quality of solutions, meaning that accurate recovery of population solutions may be obtained using a fairly small sample. However, when communalities are low, the role of sample size becomes much more important and will have a greater impact on quality of solutions.
3. Quality of factor analysis solutions will improve as over determination of factors improves. This effect will be reduced as communalities increase and may also interact with sample size [14].

The relationship between BTC-XMR looks like linear for most data points, but for very positive returns seems to be confusing. Because the scatter plot between BTC-XMR doesn't give us clear results of their correlation, it is better to calculate the Pearson and Spearman correlation coefficient. A few top cryptos are selected from July 2016 - July 2018 to have a better opinion how more cryptos are correlated with each other. Before finding the correlation of those cryptos, it would be good to see their prices in a graph.

	BTC	XMR
<b>ADF Statistic</b>	-1.29466	-1.67071
<b>p-value</b>	0.631611	0.446265
<b>Critical Values</b>		
<b>1%</b>	-1.67071	-3.436
<b>5%</b>	-2.898	-2.864
<b>10%</b>	-2.586	-2.568

Fig. 3: Augmented Dickey-Fuller Statistics [1]

## 2.1 Why Is It Useful to Find the Correlation Between the Cryptos?

Portfolio management is about how you invest in the product's development resources – project prioritization and resource allocation across development projects. There are four goals in portfolio management – maximizing the value of the portfolio, seeking the right balance of projects, ensuring that portfolio is strategically aligned, and making sure there aren't too many projects for limited resources. And there are many tools – some quantitative, others graphical, some strategic – designed to help choose the right portfolio of projects. In short, Portfolio management is fundamental to new product success. But it's not as easy as it first seems. Not only we must seek to maximize the value of the portfolio, but the development projects in the portfolio must be appropriately balanced, there must be the right numbers of projects, and finally, the portfolio must be strategically aligned. No one portfolio model can deliver on all four goals, and so best-practice businesses tend to use multiple methods to select their projects. [15]

There are three important reasons why we need to find the Pearson correlation coefficient between the cryptos.

- Help us to invest in two different cryptos that don't cancel each other out. Assume two cryptos, BTC and Ethereum where their correlation coefficient  $r$  approximates  $-1$ . That means that the price of one cryptocurrency, for example BTC, will rise while the other will decrease, as we have a negative correlation. If the investor wants to invest on both of them, then any potential profit will be cancelled out.
- Hedging is an attempt of securing against negative events and it is very similar with the first reason. The difference is that someone can hedge his investment by choosing to invest in a crypto that moves opposite from his initial investment to make up any losses. The disadvantage of this choice is that we will have smaller profits when the initial crypto that we selected moves strong.
- Instead an investor putting all his money into one crypto, he can diversify the risk of his choice by purchasing other cryptos that have different correlation coefficient. For example, someone can look on different cryptos that have constantly correlation close to  $0$ . Diversification plays an important role in modern portfolio theory which says that we can diversify away risk, by reducing the correlation between the returns of the selected assets in the portfolio. Despite the fact that the Pearson correlation coefficient has some advantages, many analysts mention some of its drawbacks. First of all, Pearson's method is only valid when we have 100% linear dependencies and this is not so often observed. However, we can surpass that obstacle by using Spearman's correlation coefficient. Moreover correlation between cryptos changes very often because it is a new area and we are still in a period of precariousness. A common mistake that many do when they are trying to find the correlation is that they use "raw data". The correct correlation can be found from daily returns (i.e percentage changes) for each crypto. Returns are simpler to work with and give a better approximation to the investor for its return. They provide better results when the price series is non-stationary, but most important, returns will captivate changes in the strength of the network effects of different cryptos [1]. Usually lot of people use "raw data" (non-stationary) without making them returns concluding to false correlation and results.

In time series analysis the first thing that someone has to do before proceeding with any method, is to check whether the data that is being dealt with is stationary or not. If a series is non-stationary then we can easily conclude to inaccurate results. To check whether or not the time series are stationary or not, Augmented Dickey Fuller (Herein referred as ADF) test which is implemented in Python is used. If the ADF statistic is less than the critical value, then the null hypothesis is rejected implying that the series is stationary and we don't have to make further transformations. If the null hypothesis can't be rejected then the series is not stationary. In fig 3 we see the results of the ADF test. [1]

As we can see, the ADF test is above the 5% critical value and the  $p\text{-value} > 0.05$ . As a result, the null hypothesis of the ADF test cannot be rejected and the series is not stationary. To make the data stationary,



	BTC	XMR
<b>ADF Statistic</b>	-5.777333	-6.887006
<b>p-value</b>	0.000001	0.00002
<b>Critical Values</b>		
<b>1%</b>	-3.514	-3.436
<b>5%</b>	-2.898	-2.864
<b>10%</b>	-2.586	-2.568

Fig. 4: Augmented Dickey-Fuller Statistics [1]

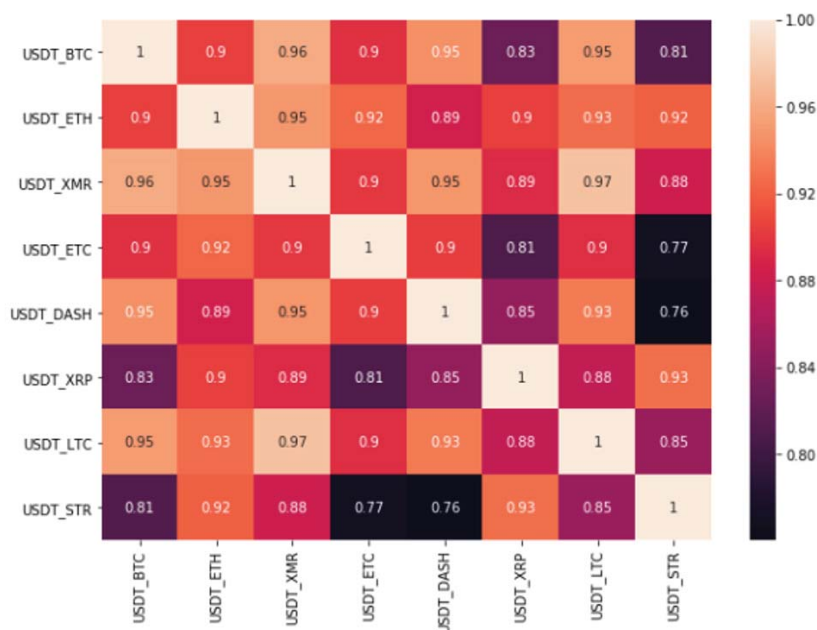


Fig. 5: Pearson correlation of "raw" data from July 2016-July 2018 [1]

we use first differences, i.e: [1]

$$y[t] = p[t] - p[t - 1], \tag{2}$$

As we can see in fig 4, after using first differences our data became stationary. The test statistic is less than the critical value at 5% and the p-value < 0.05, so it can be concluded with 99% confidence that the data is stationary. Now that we have our data stationary, we can look at the correlation of BTC and XMR.[1]

In a crypto time series analysis, the correlation requires stationary data which means constant variance over time and be mean reverting. Stationarity in correlation is achieved by converting the daily close prices into daily returns (percentage price changes). Then it will be mean-reverting, as the returns move above and below a constant mean and the variance will be constant. In fig 1 we can see the daily percentage change price of BTC and XMR. [1]

In Fig. 5 we can see the Pearson correlation of "raw data" and after we will show the Pearson Correlation coefficient of daily returns.

The difference between Spearman and Pearson is that Spearman correlation ranks the datapoints and after is running a Pearson correlation. In Spearman correlation the two variables are compared based on how much their rankings move together [1]. It's main purpose is to find co-motions in datasets that are non-linear. In a perfect positive correlation, Spearman has a value of +1, whereas Pearson is very close to +1. Now we will see, in figure 5 and 6, how the cryptos are correlated every year from July 2016 - July 2018.

## 2.2 Why are the cryptos correlated?

As per one of the research paper, The correlation of the cryptos (other cryptos to BTC) was higher in the year 2018, As high as 0.87 or higher [16]. In order to understand how correlated the cryptos are, we will require

	USDT_BTC	USDT_ETH	USDT_XMR	USDT_ETC	USDT_DASH	USDT_XRP	USDT_LTC	USDT_STR
USDT_BTC	1.000000	0.904083	0.959776	0.899917	0.945015	0.825950	0.951504	0.812522
USDT_ETH	0.904083	1.000000	0.948842	0.924132	0.885633	0.903664	0.926248	0.920089
USDT_XMR	0.959776	0.948842	1.000000	0.899866	0.948350	0.893156	0.974611	0.882038
USDT_ETC	0.896917	0.924132	0.899866	1.000000	0.903181	0.809784	0.895109	0.769262
USDT_DASH	0.945015	0.885633	0.948350	0.903181	1.000000	0.849962	0.934257	0.760369
USDT_XRP	0.825950	0.903664	0.893156	0.809784	0.849962	1.000000	0.875075	0.927506
USDT_LTC	0.951504	0.926248	0.974611	0.895109	0.934257	0.875075	1.000000	0.852919
USDT_STR	0.812522	0.920089	0.882038	0.769262	0.760369	0.927506	0.852919	1.000000

Fig. 6: Table pearson correlation of "raw data from July 2016- July 2018 [1]

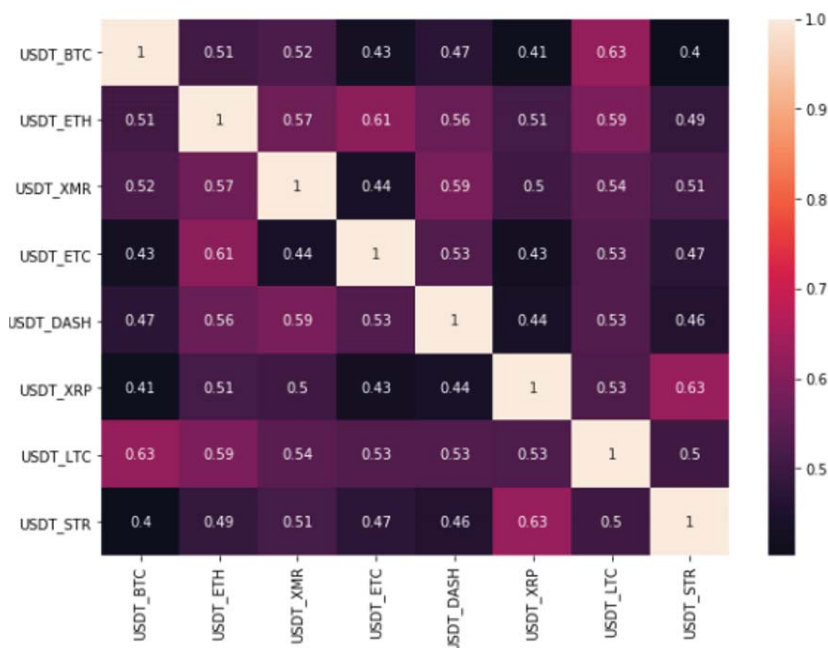


Fig. 7: Spearman correlation from July 2016-July 2018 [1]

information about: a) The price data of the crypto we would like to compare with other cryptos; b) The entire price data of the other cryptos.

The main reasons for the cryptos to be correlated to BTC are:

- Liquidity: BTC is tradable against almost all altcoins, volume represents the significant percentage of the industry, due to these factors it becomes feasible for trade to taking place in cryptos which means a rise in the top cryptos. Hence the correlation with the BTC and top cryptos rises [17]
- Dominance: Market capitalization accounts for nearly half of the crypto assets the cryptos are moderate correlated. The results showed that in 2016, there was a very weak correlation for almost all the cryptos and they were quite independent from each other. Only the pair of Litecoin (herein referred to as LTC) - BTC stood out with a strong positive correlation of 0.75, meaning that they were moving in the same direction. But that changed over the next years. In 2017 the correlation became moderate positive for most of the cryptos. The highest correlation for that year was between Ethereum (herein referred to as ETH)- Ethereum classic (herein referred to as ETC) and ETH - XMR with 0.6 and 0.55 respectively. ETC - BTC and Ripple (herein referred to as XRP) - BTC had the lowest value with 0.11 and 0.13 respectively. In 2018, almost all the cryptos that we selected were highly positive correlated with each other. This analysis give us valuable and interesting information about cryptos.

In study [18] relays this in a study and shows that, while not always, investor sentiment can predict stock prices. Considering these facts, investor sentiment could be a significant factor in the price movement of crypto prices,

This study is interesting because it shows several factors which may influence price and gives an overview of what are some of the more important variables. The research showed that 20% of BTC’s price is driven



	USDT_BTC	USDT_ETH	USDT_XMR	USDT_ETC	USDT_DASH	USDT_XRP	USDT_LTC	USDT_STR
USDT_BTC	1.000000	0.507222	0.520892	0.429775	0.472280	0.413657	0.626738	0.403122
USDT_ETH	0.507222	1.000000	0.570340	0.613165	0.562456	0.512416	0.592494	0.485525
USDT_XMR	0.520892	0.570340	1.000000	0.440158	0.585619	0.502613	0.543955	0.514640
USDT_ETC	0.429775	0.613165	0.440158	1.000000	0.529056	0.428267	0.530878	0.474469
USDT_DASH	0.472280	0.562456	0.585619	0.529056	1.000000	0.438555	0.532482	0.460311
USDT_XRP	0.413657	0.512416	0.502613	0.428267	0.438555	1.000000	0.526234	0.630520
USDT_LTC	0.626738	0.592494	0.543955	0.530878	0.532482	0.526234	1.000000	0.501635
USDT_STR	0.403122	0.485525	0.514640	0.474469	0.460311	0.630520	0.501635	1.000000

July 2018

Fig. 8: Table of Spearman correlation from July 2016 [1]

by investors' attractiveness to BTC as determined by the volume of Google search queries. This was the major determinant of price in the study. The results indicate that positive sentiment (conveyed through the variable: 'attractiveness to BTC') affects BTC's price, the authors showed that the remaining 70% of BTC's price movements is explained by 'its own innovative shocks' which is an ambiguous explanation, effectively relying on using the residual as the signal of systemic unexplained component of price formation.

If BTC acts as a safe haven or a hedge for currencies. Therefore, stock market indices have an expectation to be negatively correlated with the price of BTC. Considering the interpretations of fear in regards to asset prices and the contradicting arguments for and against BTC being a hedge against the stock market, it seems appropriate to isolate the interpretation of a correlation between cryptos and fear as being 'cryptos are a hedge against the stock market in times of fear' and not just 'cryptos are a hedge against the stock market' over all stock markets environments [18].

- Similar behaviour: One important reason why some cryptos are correlated with bitcoin is, they belong to the same family meaning bitcoin and the other cryptos have similar function and prospects for a long time. (eg: LTC and BTC share similar behavior) [19]
- Limited or no options for trading or investing other currencies than BTC/fiat, BTC/ETH. Investors must first buy BTC or ETH before they can exchange one of these for the desired crypto. This pushes the dominance of BTC over the market, resulting in a "follow the leader" behaviour. At Pool of Stake we think that any investor should take notice of the correlation between cryptos. This can be used as a useful tool to determine macro trends inside this crazy and volatile market. Some of the advantages of studying the correlations between pairs are: a) Avoiding investing in two coins that cancel each other out. You invest in two different cryptos but the correlation coefficient tends to -1. This means that if the price of one is going up, the other will go down, therefore cancelling out the potential profit of your acquisition. b) Diversify your portfolio. Instead of investing 100% of your portfolio in only one coin, you should find another closely correlated coin and split some of the investment to diversify the portfolio but still take advantage of the pair's inertia. If one asset is in danger of losing its value, you still have 50% of your account allocated to safer position. [19]

### 3 Conclusion

In this project, we focused on a new area, the area of cryptocurrencies specifically on Bitcoin, Ethereum, Litecoin and Monero. The goal of this project is to study the correlation of BTC with different cryptos. We focus on correlation of cryptos from July 2016 to July 2018. After we talked about time series and how to make the data stationary with the help of the Augmented Dickey-Fuller test (ADF) [20]. Additionally, we discussed the Pearson and Spearman correlation coefficient of the history data presented in other research papers. The main reasons for the cryptos to be correlated to BTC are: liquidity, dominance, similar behaviour, limited or no options for trading or investing other currencies than BTC/fiat, BTC/ETH.

### Acknowledgement

This work was supported by the Grant Agency of the Czech Technical University in Prague, grant No. SGS18/177/OHK2/3T/12.

## Literature

- [1] Angelos Kalaitzis. Bitcoin-monero analysis: Pearson and spearman correlation coefficients of cryptocurrencies, 2018.
- [2] Melanie Swan. *Blockchain: Blueprint for a new economy*. " O'Reilly Media, Inc.", 2015.
- [3] Michael Crosby, Pradan Pattanayak, Sanjeev Verma, Vignesh Kalyanaraman, et al. Blockchain technology: Beyond bitcoin. *Applied Innovation*, 2(6-10):71, 2016.
- [4] Cameron Harwick. Cryptocurrency and the problem of intermediation. *The Independent Review*, 20(4):569–588, 2016.
- [5] Lawrence H White. The troubling suppression of competition from alternative monies: The cases of the liberty dollar and e-gold. *Cato J.*, 34:281, 2014.
- [6] Colin LeMahieu. Raiblocks: A feeless distributed cryptocurrency network. URL [https://raiblocks.net/media/RaiBlocks\\_Whitepaper\\_\\_English\\_.pdf](https://raiblocks.net/media/RaiBlocks_Whitepaper__English_.pdf), 2017.
- [7] Satoshi Nakamoto et al. Bitcoin: A peer-to-peer electronic cash system. 2008.
- [8] Bob Pisani and Todd Haselton. Here's why bitcoins prices are different on each exchange-, 2017.
- [9] Alex Morris. U today- how many people own bitcoin and how they use it., 2018.
- [10] Joy A Thomas and TM Cover. Elements of information theory. *John Wiley & Sons, Inc., New York. Toni, T., Welch, D., Strelkowa, N., Ipsen, A., and Stumpf, MPH (2009), "Approximate Bayesian computation scheme for parameter inference and model selection in dynamical systems," Journal of the Royal Society Interface*, 6:187–202, 1991.
- [11] Pau Vilimelis Aceituno (<https://stats.stackexchange.com/users/135502/pau-vilimelis-aceituno>). Mutual information versus correlation. Cross Validated. URL:<https://stats.stackexchange.com/q/241312> (version: 2016-10-20).
- [12] Alex Greaves and Benjamin Au. Using the bitcoin transaction graph to predict the price of bitcoin. *No Data*, 2015.
- [13] David Freedman, Robert Pisani, and Roger Purves. Statistics (international student edition). *Pisani, R. Purves, 4th edn. WW Norton & Company, New York*, 2007.
- [14] Robert C MacCallum, Keith F Widaman, Shaobo Zhang, and Sehee Hong. Sample size in factor analysis. *Psychological methods*, 4(1):84, 1999.
- [15] Robert Gravlin Cooper, Scott J Edgett, and Elko J Kleinschmidt. Portfolio management for new products. 2001.
- [16] Binance Research. Are cryptoassets highly correlated?, 2019.
- [17] Adam Hayes. Cryptocurrency value formation: an empirical analysis leading to a cost of production model for valuing bitcoin. *Telematics and Informatics, Forthcoming*, 2016.
- [18] Constantin Gurdgiev, Daniel O'Loughlin, and Bartosz Chlebowski. Behavioral basis of cryptocurrencies markets: Examining effects of public sentiment, fear and uncertainty on price formation. *Forthcoming: Journal of Financial Transformation*, 49, 2019.
- [19] Pool of Stake. Cryptocurrency correlation, 2018.
- [20] Yin-Wong Cheung and Kon S Lai. Lag order and critical values of the augmented dickey–fuller test. *Journal of Business & Economic Statistics*, 13(3):277–280, 1995.

# **ELECTRONICS & CONTROL SYSTEM FOR LASER CUTTER**

*Haneen Farman Mohamed Mubarak*

*České vysoké učení technické v Praze, Fakulta strojní,  
HaneenFarman.MohamedMubarak@fs.cvut.cz*

*Abstract: Laser cutting is a technology that uses a laser to cut materials, and is typically used for industrial manufacturing applications, but recently has been started to be used by schools for research purposes, small businesses, and hobbyists. The main principle in Laser cutting is to cut materials by directing the output of a high-power laser most commonly through optics. [1]*

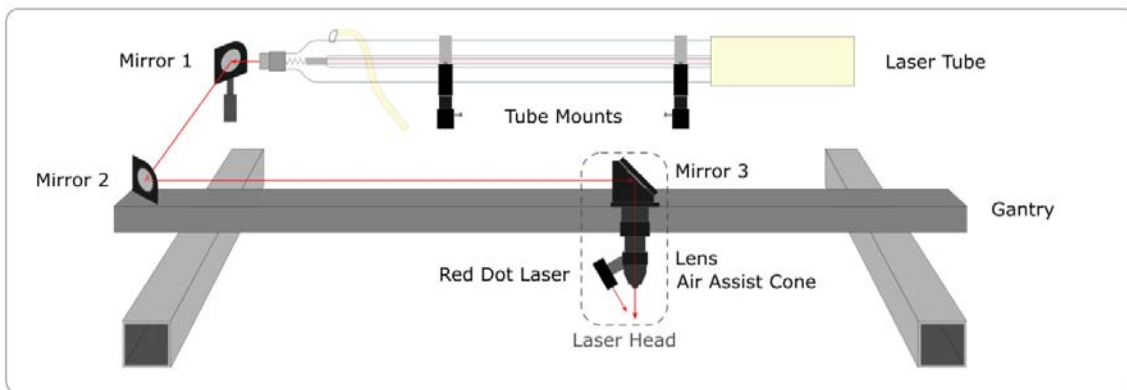
*The work is mainly to refurbish an existing PCB cutter and convert into a laser cutting machine. This will be done by studying the working of the existing components, installing new and modifying existing mechanical parts, installing the necessary optics and finally developing a control system that will drive the laser cutter.*

*Keywords: Laser Cutter, Stepper motor, Laser tube, optics, Smoothie board, CNC, G-Code.*

## 1 Introduction

The laser optics and CNC (Computer Numerical Control) are used to direct the laser beam generated onto the material that is to be cut. A typical commercial laser for cutting materials involves a motion control system to follow a CNC or G-code of the pattern to be cut onto the material. The focused laser beam is directed at the material, which then either melts, burns, vaporizes away, or is blown away by a jet of gas, leaving an edge with a high-quality surface finish. Industrial laser cutters are used to cut flat-sheet material as well as structural and piping materials. [1]

The movement of the laser beam is achieved by the use of Optical Mirrors which are mounted on an X-Y axis table. By movement of the X-axis and Y-axis, the laser beam can be directed to different parts of the work piece. The displacement of the X-Y axis is generally carried out by motors preferably Stepper Motors which are controlled by a Microcontroller.



Pic.1 Basic Components of Laser Cutter

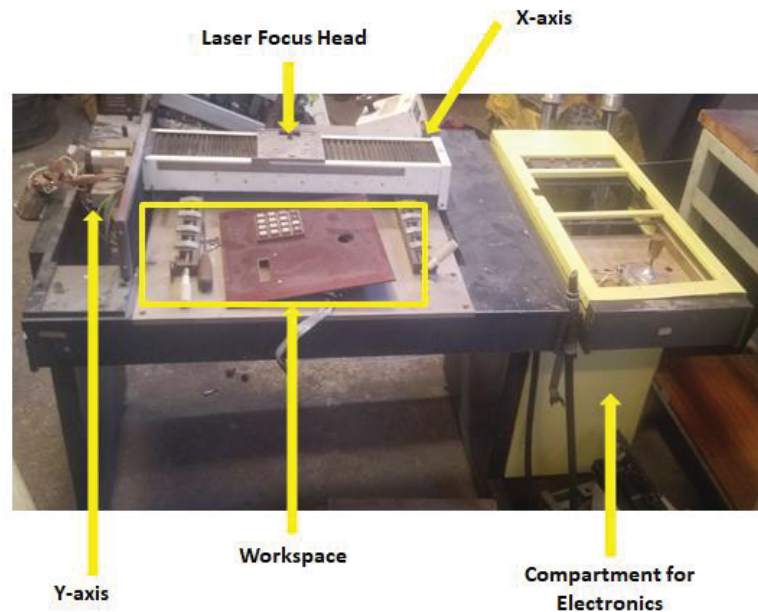
## 2 Scope of Work

The AIRSHIP Club under the Faculty of Mechanical engineering has in its laboratory an old PCB engraver that is approximately 50 years old. The main aim of my diploma thesis is to refurbish the PCB engraver and convert it into a Laser cutter/engraver with the purpose of the cutting Plexi glass into small components.

The project would be a joint work between the AIRSHIP Club and Department of Instrumentation and Control Engineering with the former taking care of the hardware construction and the latter developing the electronics and control system for the Laser Cutter.

Scope of my work for the part of Project 3:

1. Perform a case study of the existing PCB engraver and identify the functioning of its key components.
2. Study the working and technologies of Laser Cutters and the components required to build one.
3. Prepare and propose a Bill of Material to purchase components required.
4. Test and see the possibility of using the existing stepper motors.



Pic.2 Existing Setup

### 3 Component Requirement

After undertaking a detailed case study of laser cutter used in industries and those used by hobbyists, we concluded the components that will be needed to build the laser cutter. The components required were classified into two parts, the parts for building the laser and the parts for the control system.

Components to be needed for building the laser cutter:

1. CO2 Laser Tube
2. Laser Power Supply
3. Optics ( Focusing Lens, Mirror and Lens Stands)
4. Water Pump
5. X-Y axis table (Existing)

Components for the Control of the system:

1. Smoothie Board
2. Stepper Motors (Existing)
3. Driver for Stepper Motors
4. LCD Screen for the Smoothie Board

#### 4 Bill of Material

Taking into account of the limited budget we had, we had to request different quotations from many suppliers from China, Germany and The Czech Republic and also for different configurations of the Laser Cutter.

The final and approved Bill of Material is shown below:

Table.1

Item	Cost (CZK) (Approx.)
50 Watts Laser Tube	3500
50 Watts Laser Power Supply	1600
Optics	3000
Water Pump	900
Smoothie Board	2000
LCD Screen For Smoothie board	250
Total	11,250

#### 5 Testing

##### 5.1 Stepper Motors

The stepper motors shown in this picture are approximately 50 years old. Because of this we were unable to find any catalogues providing more information about them. Hence we had to test them step by step.

However, we know that these motors are Unipolar motor with 8 leads. The existing PCB printer has 5 available stepper motors.

##### Stepper Motors Test 1

In the first test, we tried to find which leads were paired. We did this by measuring the resistance across the leads and we were unable to find which leads were paired. We then gave a supply directly to the motors and found that torque was produced in the motors. All 5 motors passed this test.



**Results of Testing**

Table.2

Motor 1	
Ports	Ohms
1-2	41.4
3-4	42.3
5-6	41.5
7-8	42.0

Motor 2	
Ports	Ohms
1-2	40.6
3-4	40.9
5-6	40.8
7-8	41.2

Motor 3	
Ports	Ohms
1-2	42.3
3-4	40.9
5-6	41.0
7-8	40.3

Motor 4	
Ports	Ohms
1-2	40.8
3-4	41.2
5-6	40.8
7-8	41.8

Motor 5	
Ports	Ohms
1-2	42.1
3-4	42.0
5-6	40.4
7-8	40.8

**Conversion from Unipolar to Bipolar Stepper**

1. Since unipolar motors are less used these days, it was difficult to find a proper unipolar stepper motor driver with desired specifications like current control.
2. Hence, we decided to convert the unipolar stepper motor into a Bipolar Stepper motor. I did this by parallel connection of the ports.
3. To test the working of it, I connected the motors to a driver and with the help of an Arduino controller we tested the working of the motors.

**Parallel Connection of the Ports**

To convert the motors into Bipolar, we parallel connected the ports. In all the five motors, the ports which were connected in parallel are as follows:

Table.3

1	5
2	6
3	7
4	8

From the second test, we found that, only two of the five stepper motors were working and the other three were faulty. Since, 2 motors are required to drive each axis, another single and powerful motor has to be purchased to drive another axis. The new motor that is purchased is a bipolar stepper motor.

The parameters of the new motor are as follows:

Table.4

No of Leads	4
No of Phases	2
Step angle	1.8 <sup>0</sup>
Mass	0.24Kg
Rated Voltage	3.57 V
Rated Current	1.5 A
Torque	1.5 N.m

## 5.2 X-Y Axis Table

With the motors which were working, I tested the movement of the X axis and Y axis of the table respectively and observed that both the axes worked with any problem.

The speed of the movement of the X-axis and Y-axis were measured and was found about to be 1cm/second.

## 6 Future Development

The scope of the work for Project has been successfully carried out. For the remaining part of the project, I will carry it on as my Diploma thesis. The work that has to be for the successful completion of the project is as follows:

1. Configure the Electronics for the machine.
2. Configure a Control System for the Laser Cutter using a Smoothie Board controller.
3. Test the working of the system.
4. Prepare the necessary documentation of the Laser cutter.

## 7 Conclusion

The objectives which were laid for Project III were successfully carried out. The components that are required to construct the Laser Cutter have been ordered and are to be shipped. The motors were tested and arrangements to purchase and install the new motor have been made.

## References

[1] Laser Cutter 101, online < <https://github.com/Obrary/ebook-laser-cutter-101/raw/master/eBook/Laser-Cutter-101-eBook.pdf> >, accessed 26.10.2018,

# OPTICAL CHARACTER RECOGNITION FOR EXPIRATION DATES VALIDATION

Mitja Avgunštinič<sup>1</sup>, Adam Pechl<sup>2</sup>

<sup>1</sup>University of Ljubljana, mitja.avgustincic@gmail.com

<sup>2</sup>Czech Technical University in Prague, adam.pechl@fs.cvut.cz

*Abstract: This paper focuses on custom OCR algorithm created for expiration dates validation using mainly open source libraries. Two main approaches are proposed in this work: recognising every character separately and unified characters method. To boost accuracy of OCR algorithm, several preprocessing methods and algorithms are also introduced. The final module is written in python and reaches satisfying accuracy considering all the difficulties, which have come across during the implementation process.*

*Keywords: OCR, character recognition, image preprocessing, OpenCV*

## 1 Introduction

In recent years a lot of development and action was focused on the *optical character recognition* (OCR). Starting in the 1950, with the purpose of helping vision impaired persons to read, the OCR has developed and greatly improved throughout the decades, leaving its marks mainly in industry.

In nowadays industry, the OCR is generally used as an aid for automatization of the processes, with the processing speed being its stronghold. Primarily OCR is being used for document scanning, recording and tracking. One of the most used applications of the OCR system is in the banking, where the OCR typed fonts were developed for easier and more accurate character detection [1].

Among the other applications we can also find license plate recognition as surveillance technique and industrial packages validation, where product numbers (barcodes) and expiration dates are validated.

In our project we focused on the optical character recognition of the expiration dates on food packages with the purpose of validating the expiration date correctness in the production industry. Another possible application of our project would be to integrate our system into smart refrigerators, where every package is scanned for expiration date, before being put in the fridge, to ease the package and data management of the stored products.

The main focus was put on creating software for automatizing the process of reading an expiration date from image and its implementation in python. Our system takes an image as an input, process it using different algorithms and methods, and with the use of open source libraries recognises the characters in the expiration date. Two approaches were tested with the intention to compare them between themselves and find the method with the higher accuracy.

## 2 Optical character recognition (OCR)

Optical character recognition (OCR) is a process of classification of optical patterns contained in a digital image corresponding to alphanumerical or other characters. The character recognition is achieved through important steps of segmentation, feature extraction and classification [2].

OCR systems, with its capability of identifying graphic symbols that make up an alphabet of any language, simulate the human ability to read. These systems start from the acquisition of an image through optical equipment connected to a device to capture the graphic symbols (usually by a camera or a scanner). Digital image processing techniques, computer vision and pattern recognition are applied to the image in order to obtain a code, such as ASCII, which will represent each character [2].

OCR systems have become one of the most successful applications of technology in the field of pattern recognition and artificial intelligence and have also been widely applied in various areas, such as industrial applications

(Markov models, license plate recognition vertical, traffic sign recognition, etc). It is needed when the information should be readable both, to humans and to machine, and alternative inputs cannot be predefined. There are numerous commercial systems, but none so far have achieved recognition rates close to humans, especially when the system works in an uncontrolled environment, with the inevitable presence of noise. However, these systems have the advantage of total insensibility to eyestrain and in most cases outperform humans at speed of recognition [2], [3].

The algorithm described in this article can be divided into 3 stages, first we used various techniques to preprocess the image, second step was using algorithms for localization and segmentation of the image and the last step was character recognition using open source OCR software.

### 3 Experimental setup and used software

For our project, several random food packages with expiration dates were used. Images were obtained using the mobile phone *Huawei P9* with 12 megapixel *Leica* camera with lens aperture  $f/2.2$ . Data set of expiration date images was created for further processing and validation.

Software was created using *python 3.7 programming language*. There is a whole range of OCR software available today in the markets like: Desktop OCR, Server OCR, Web OCR, Mobile OCR etc. When considering possible OCR applications for our project, we tried different web OCR (websites), add-ons (Adobe Reader, Word) and even applications. Accuracy of extracting text of any of these OCR tool varies from 71 % to 95 %, with proper image preprocessing. Many OCR tools are available as paid and work really well but only few of them are open source and free.

Among many libraries and OCR software provided on the internet, the most popular and widely used library is *tesseract* [4]. The *tesseract* library was chosen based on some key properties such as being open source, high accuracy, constant development and improvement (currently under Google development) and different programming language integration (such as python, C++ and Java).

#### 3.1 OpenCV

*Open Source Computer Vision Library* (OpenCV) [5] is one of the most widely used libraries in image processing, which is developed and released by *Intel Corporation*. It is distributed under a BSD style license which allows for royalty free commercial or research use with no requirement that the user's code be free or open. OpenCV is written in C and C++ and runs under Windows, Linux or MAC OS, but the code is well behaved and has ported to many other operating systems [6].

OpenCV contains an optimized collection of C libraries spanning a wide range of computer vision algorithms, including motion segmentation and pose recognition, multi-projector display system, object and face recognition, and 3D reconstruction, etc. The broad functional areas supported by OpenCV include:

- basic structures and array manipulations
- image processing and analysis
- object structural analysis
- motion analysis and object tracking
- object and face recognition
- camera calibration and 3D reconstruction
- stereo, 3D tracking and statistically boosted classifiers
- user interface and video acquisition support

There is active development on interfaces for Python, Ruby, Matlab, and other languages [6].

#### 3.2 Tesseract

Tesseract is an optical character recognition engine for various operating systems. It is free software, released under the *Apache License*, Version 2.0. Since 2006, it's development has been sponsored by Google.

We are using python library PyTesseract [7], which is an optical character recognition module for Python. It takes an image as input and outputs a string. PyTesseract uses the Tesseract OCR engine, converting images to an accepted format and calling the Tesseract executable as an external script.

Tesseract's output will have very poor quality if the input images are not preprocessed properly. Images (especially screenshots) must be scaled up so that the text x-height is at least 20 pixels, any rotation or skew must be corrected or no text will be recognized, dark borders must be manually removed, or they will be misinterpreted as characters, etc.

## 4 Image Preprocessing

In order to preserve high accuracy, image preprocessing must be made. We used different methods and algorithms in order to prepare the image for letter recognition with the highest possible accuracy.

Usually, in a computer vision system, the preprocessing step aims to improve the quality of the digital image. Therefore, algorithms are applied to eliminate undesirable regions (noise), which arise due the conditions or the method of acquisition. *Low-pass filters* are among the most commonly used techniques to eliminate and reduce noise in an image. This type of filter smooths the image and reduces the number of gray levels and consequently minimizes the noise. High frequencies, which correspond to fast transitions are attenuated. *Median* and *Gaussian* are two types of low-pass filters widely used [3].

### 4.1 Gaussian Blur

A Gaussian filter softens the image based on the mathematical theory of the Gauss curve. This filter is applied on the convolution matrix defined as mask, and depending mainly on the standard deviation  $\sigma$ . Equation 1 represents the Gaussian filter in two dimensions, which means, that it is the product of two such Gaussian functions, one in each dimension:

$$G(x,y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \quad (1)$$

where  $x$  is the distance from the origin in the horizontal axis,  $y$  is the distance from the origin in the vertical axis, and  $\sigma$  is the standard deviation of the Gaussian distribution. When using two dimensional blurring, values from Gaussian distribution are used to build a convolution matrix which is applied to the original image. Each pixels new value is set to a weighted average of that pixels neighborhood. The original pixels value receives the heaviest weight (having the highest Gaussian value) and neighboring pixels receive smaller weights as their distance to the original pixel increases. This results in a blur that preserves boundaries and edges better than other, more uniform blurring filters.

### 4.2 Color to Grayscale Conversion

The next step in image preprocessing was conversion from color to grayscale.

Grayscale image is one in which the value of each pixel is a single sample representing only an amount of light. The image itself only carries intensity information.

Input images have *RGB* color channels. Classically, the grayscale image is obtained by a linearly weighted transformation:

$$\mathbf{J}(x,y) = \alpha \cdot \mathbf{R}(x,y) + \beta \cdot \mathbf{G}(x,y) + \gamma \cdot \mathbf{B}(x,y) \quad (2)$$

The most popular method selects the weighted values of  $\alpha$ ,  $\beta$  and  $\gamma$  by eliminating the hue and saturation information while retaining the luminance.

To this end, a color pixel is first transformed to the so-called NTSC color space from the RGB space by the standard NTSC conversion formula:

$$\begin{bmatrix} \mathbf{Y}(x,y) \\ \mathbf{I}(x,y) \\ \mathbf{Q}(x,y) \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ 0.596 & -0.274 & -0.322 \\ 0.211 & -0.523 & 0.312 \end{bmatrix} \begin{bmatrix} \mathbf{R}(x,y) \\ \mathbf{G}(x,y) \\ \mathbf{B}(x,y) \end{bmatrix} \quad (3)$$

where  $\mathbf{Y}, \mathbf{I}$  and  $\mathbf{Q}$  represent the NTSC luminance, hue, and saturation components, respectively. Then the luminance is used as the grayscale signal  $\mathbf{J}(x,y) = \mathbf{Y}(x,y)$ . Thus we have:

$$\alpha = 0.299, \quad \beta = 0.587, \quad \gamma = 0.114$$

With the grayscaling we reduced the noise of saturation and hue in the image, leaving only luminance on the image [8].

### 4.3 Thresholding

Another very common process in the preprocessing step is thresholding. This technique has set of values as the threshold, usually in gray levels, which are maintained in an interval of image tones [3].

If a pixel value is greater than a threshold value, it is assigned one value (may be white), else another value is assigned (may be black).

In our case the adaptive Gaussian threshold was used with the binary thresholding function. Binary threshold function returns only values that are either black (0) or white (255).

Normal thresholding uses a global value as threshold value. It turns out that this may result in poor results when dealing with images with different lightning conditions in different areas. Adaptive threshold on the other hand calculates the threshold for small regions of the image resulting in different thresholds for different regions of the same image giving us better results for images with varying illumination [9].

### 4.4 Image Closing

The last image preprocessing algorithm, we need to use, is image closing. Image closing is morphological operation. Morphological operations can be used to construct filters similar in concept to the *spatial filters*. The objective of the morphological operation is to eliminate the noise and its effects on the print while distorting it as little as possible.

The image closing is usually used in image processing that tends to smooth sections of contours. However it generally fuses narrow breaks and long thin gulfs, eliminating small holes and filling the gaps in the contour. For closing operations 3x3 kernel was used [10].

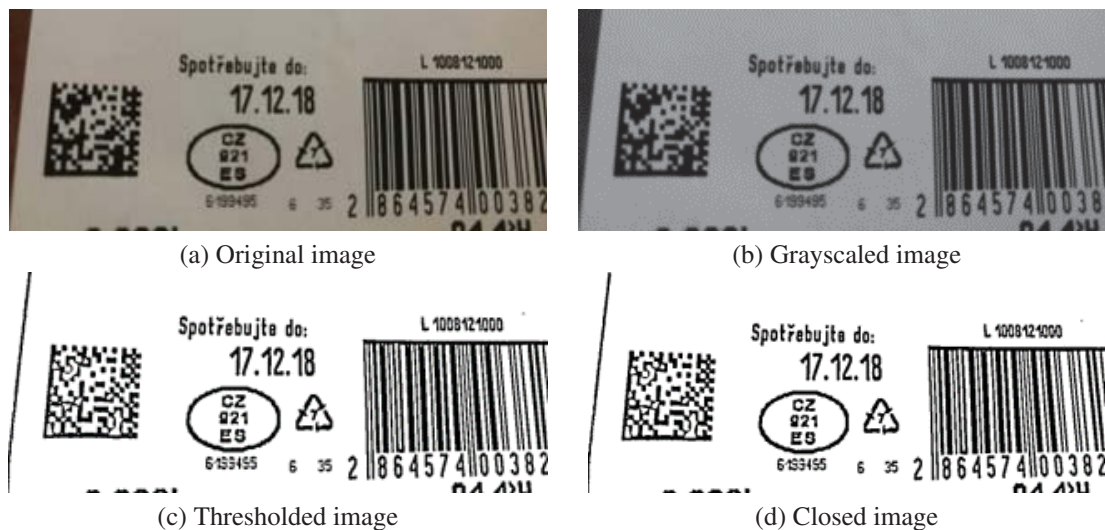


Table 1: Sequence of preprocessing of an image necessary before recognition

### 4.5 Segmentation

In the next step all of the contours in the image were found using OpenCV function *findContours* [11]. These contours can be divided into two groups:

1. contours of the characters we want to recognise
2. contours we recognise as a noise

There are many different approaches to split the contours into mentioned groups. The one used here takes the boundaries of width and height of contours as parameters, which are set in accordance to the width and height of the target characters on the preprocessed image, resulting in extracting of these character contours from every possible contour in the image.

OpenCV also includes function called *boundingRect* [12]. This function was used to determinate the position and the area of the contours in terms of coordinates. With information about the characters area and location, the letter extraction follows [12].

In order to extract letter, white image was created that would serve as a canvas, as an matrix with all values of 255 (white) and the same resolution of an input image. Later on, the position and area of each bounding box



around a character contour was taken and used to overwrite the values of the bounding box on the canvas white image in the same position as in original image. With this technique, we end up with image that only contains characters we want to recognise. The noise has been generally removed.

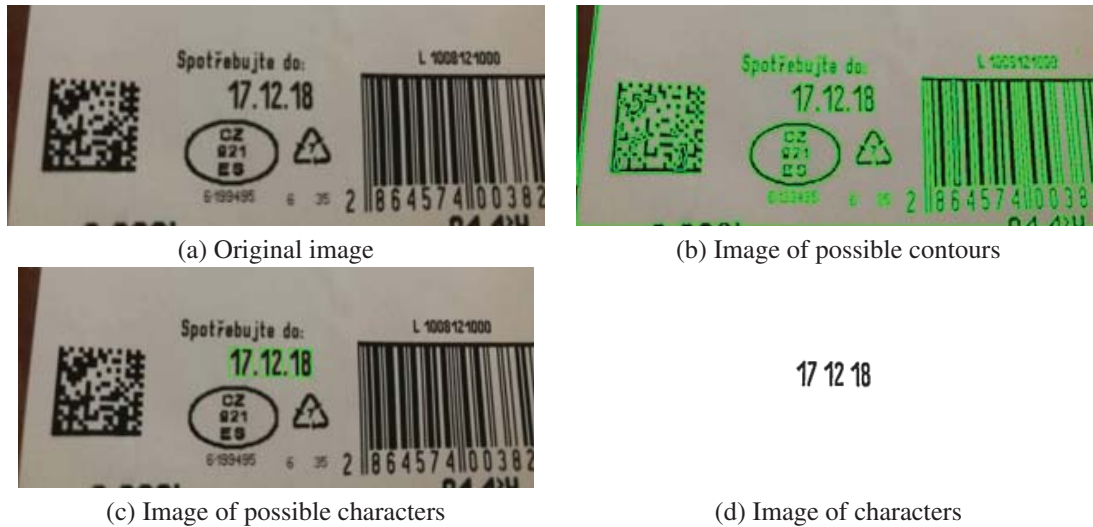


Table 2: Segmentation

## 5 Letter extraction

Final step in our software we had to take was the use of the library tesseract to recognise the characters from preprocessed image and return it as a string.

Although we reduced most of the noise from the image, it is still possible that there is a noise in the image. If we still have noise in the image or if the characters are not clearly shown, tesseract either recognises nothing or it recognises non ASCII characters. The noise removal in this case was solved using returned string and remove all the blank spaces and non ASCII characters as well as ASCII letters. The ending result is expiration date number in the following format DDMMYYYY [13].

## 6 Unified characters method

In the beginning of this research an assumption was made, that recognising every character separately can have its cons. These cons were confirmed with testing the software and are generally thought to be:

1. recognising more noise if the expiration dates characters are the same font size as noise text
2. lower accuracy because of punctuation<sup>1</sup>
3. lower accuracy because of different date formats

In order to improve the software and to avoid the cons used in *separate character recognition* method, an alternative approach was made.

Instead of using 3x3 kernel in image closing, horizontal size 30x5 kernel was chosen. With selecting kernel crucially bigger horizontally than vertically, we achieve image closing of the horizontal printed text. With this method, horizontally aligned text is merged. Contour functions are applied on the horizontally merged text and when we have the contours, we implement them on the original preprocessed image with 3x3 image closing kernel.

In this case, instead of filtering the contours using the height and width of the characters, we can filter the width of the whole expiration date. The input parameters are in this case easier to determine, since the expiration date is usually stand alone print on the package and it usually has different text width than noise in the image.

The rest of the algorithm, including image processing, segmentation and tesseract character recognition remain the same.

<sup>1</sup>For example character >< can be recognised as >0<

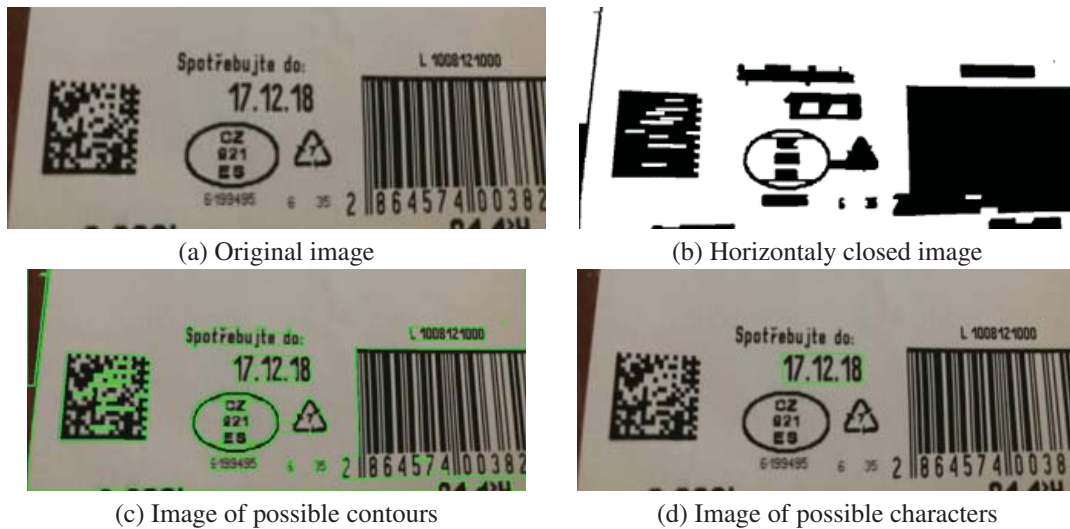


Table 3: States of Unified characters method

## 7 Conclusions

One of the first realisation made was that the accuracy of the both methods is the same, 66.7 %. In both cases low accuracy was achieved due to different lightning and other surrounding effects on the images. It should be also taken into consideration the cameras automatic mode for photograph optimisation, that changes parameters differently for every single image. One of the reason for not recognised expiration dates is also the bad quality of the printing expiration dates on the packages, because some of them were partially washed or brushed off.

When analysing the single OCR method conclusion was made, that it is easier and much more accurate to recognise the expiration dates on the packages, where expiration date font is sufficiently bigger than the other text. There are also eventual problems, with noise in the close region of the character, because simple long thin noise contour can be easily recognised as  $\gg I \ll$  or  $\gg 1 \ll$ . This kind of misinterpretation are not in the unified OCR method, because the tesseract library, when being fed with merged letters recognises and distinguish between characters and punctuation and sometimes even noise. However in some cases although the single OCR recognises the photo, no matter of the input parameters, the unified OCR recognises nothing.

Some general conclusions regarding the expiration dates were also made. The most clearly date expiration dates are the one who are printed with black color on white background. The OCR accuracy also depends on the material, the expiration date is printed on, the most success we get with carton, while plastics are harder to read due to reflective light our camera captures. Dot matrix character expiration dates are one of the most hard to read since for proper OCR it is needed to have closed contours. Stamped expiration dates also cause problems reading the expiration dates, since many of the character details are washed or brushed off, leaving incomplete letters, our program fails to recognize.

In order to have better OCR expiration date recognition in industry for validation purposes, the products provider should take an extra step in designing more OCR friendly packages. Some of the possible solutions are: different font sizes for expiration dates, OCR typed fonts, material where expiration date is printed (paper sticker instead of plastics) and most important of all, the controlled space where capturing and validation will be made.

Some of the future work considerations are using grayscale camera of lower resolution than 12 MP. With this action we reduce the computing time, making it more suitable for real-time industry and simplification of the software could be made with removing some of its features. Secondly, bigger amount of data set could be used in order to obtain better accuracy and one type of food packaging, captured in constant not-changing space should be used.

## Acknowledgements

This project was supported by grant *SGS18/177/OHK2/3T/12*. The project was programmed in *Python* and source codes are available on request.

## References

- [1] Amarjot Singh, Ketan Bacchuwar, and Akshay Bhasin. A survey of ocr applications. *International Journal of Machine Learning and Computing*, 2(3):314, 2012.
- [2] Pratixa Badelia Soumya K. Ghosh Arindam Chaudhuri, Krupa Mandaviya. *Optical Character Recognition Systems for Different Languages with Soft Computing*. Springer International Publishing AG 2017, 2017.
- [3] Gabriel B Holanda, João Wellington M Souza, Daniel A Lima, Leandro B Marinho, Anaxágoras M Girão, João Batista Bezerra Frota, and Pedro P Rebouças Filho. Development of ocr system on android platforms to aid reading with a refreshable braille display in real time. *Measurement*, 120:150–168, 2018.
- [4] pytesseract. <https://pypi.org/project/pytesseract/>. Accessed: 2019-02-19.
- [5] Opencv. <https://opencv.org/>. Accessed: 2019-02-20.
- [6] Opencv - about. <https://opencv.org/about/>. Accessed: 2019-02-20.
- [7] Pytesser. <https://pypi.org/project/PyTesseract/>. Accessed: 2019-02-22.
- [8] Color conversions. [https://docs.opencv.org/3.1.0/de/d25/imgproc\\_color\\_conversions.html](https://docs.opencv.org/3.1.0/de/d25/imgproc_color_conversions.html). Accessed: 2019-02-21.
- [9] Image thresholding. [https://docs.opencv.org/3.4.0/d7/d4d/tutorial\\_py\\_thresholding.html](https://docs.opencv.org/3.4.0/d7/d4d/tutorial_py_thresholding.html). Accessed: 2019-02-21.
- [10] Rafael C. Gonzalez and Richard E. Woods. *Digital image processing*. Prentice Hall, Upper Saddle River, N.J., 2008.
- [11] Opencv - findcontours. [https://docs.opencv.org/2.4/doc/tutorials/imgproc/shapedescriptors/find\\_contours/find\\_contours.html](https://docs.opencv.org/2.4/doc/tutorials/imgproc/shapedescriptors/find_contours/find_contours.html). Accessed: 2019-02-25.
- [12] Opencv - boundingrect. [https://docs.opencv.org/3.1.0/dd/d49/tutorial\\_py\\_contour\\_features.html](https://docs.opencv.org/3.1.0/dd/d49/tutorial_py_contour_features.html). Accessed: 2019-02-25.
- [13] python.org - string. <https://docs.python.org/2/library/string.html>. Accessed: 2019-02-25.

# INTERACTIVE JUPYTER NOTEBOOKS IN EDUCATION: CASE STUDY OF AN AIRCRAFT ENGINE COMPRESSOR

*Nina Kumer, Matous Cejnek, Adam Pechl*

*Ústav přístrojové a řídicí techniky, ČVUT v Praze, Fakulta strojní, Praha*

*Abstrakt: V tomto projektu je popsáno řešení teoretického problému a následná tvorba interaktivního souboru jupyter notebook pro vysvětlení problémů. Teoretický problém představený v tomto článku je studie kompresoru leteckého motoru. Problém je řešen pomocí dynamické analýzy. Všechny potřebné odvození a rovnice jsou prezentovány v tomto článku. Nakonec jsou tyto rovnice vyřešeny pomocí nástroje jupyter notebook.*

*Klíčová slova: jupyter notebook, dynamická analýza, modální souřadnice, periodická síla, impulzní síla*

*Abstract: With this project we want to show a way how to solve theoretical problem, creating an interactive Jupyter Notebook. For that purpose, we decided to study an aircraft engine compressor. Given task is theoretical, but it could be easily used in real-life situation. We start solving our problem by studying theoretical background of dynamic analysis. It contains deriving all the equations needed, calculation eigenvalues and eigenvectors, transformation into modal coordinates, and harmonic and impulse analysis. Derived equations are later on solved in Jupyter Notebook, using NumPy. All the graphs are created with Matplotlib.pyplot.*

*Keywords: jupyter notebook, dynamic analysis, modal coordinates, periodic force, impulse force*

## 1 Introduction

The purpose of this project is to show a possible way how to perform dynamic analysis on a practical case of an aircraft engine compressor. We are examining multidegree-of-freedom system, its modal analysis and response under a periodic and non-periodic (impulse) force.

First, theoretical research was made and important equations were derived. Actual computation was done with Python in Jupyter Notebook [1]. It is an open-source web application that allows you to create and share documents that contain live code, equations, visualisations and narrative text. Uses include data cleaning and transformation, numerical simulation, statistical modelling, data visualisation, machine learning, and much more. For the needs of this project, the most important packages are NumPy, which provides a high-performance multidimensional array object and tools for working with these arrays [2], and Matplotlib.pyplot, which is designed to be able to create simple and complex plots with a few command [3].

This project is important because it shows that once the theoretical equations are delivered, Python and Jupyter Notebook are a very useful tools for computing and visualising the actual results. That makes them more understandable and lifts our knowledge on a higher level.

## 2 Task

The explanation of the process of solving the problem is in this section. In subsection 2.1 the definition and goals are given. We start solving the problem in the next subsection, 2.2, where equations of motion are derived. We use them later in the subsection 2.3, where eigenvalues and eigenvectors are calculated. In subsection 2.4 physical coordinates are transformed into modal coordinates. We represent a continuous periodic function  $M_1(t)$  with Fourier series in subsection 2.5. Later on, we determine a steady-state response of the system. In the last subsection 2.6, we analyse a response of the system on impulse force, using convolution integral.

## 2.1 Definition and goals

Compressor is made of  $N$  disks with moment of inertia  $J_i$ . Free lengths of the shaft between the disks should be modeled as a torsion spring with coefficient  $k_i$ . Moment of inertia of each disk and spring coefficients are described as

$$J_i = J_0 e^{1-0.3i},$$

$$k_i = k_0 e^{1-0.3i},$$

where the parameters are given as follows:

Tab. 1: Given parameters

<i>parameter</i>	<i>unit</i>	<i>value</i>	<i>name/description</i>
$J_0$	Nm/rad	1400	moment of inertia
$M_0$	kgm <sup>2</sup>	0.4	moment
$N$	/	8	number of disks
$T$	s	0.17	time of 1 period
$t_0$	s	1.06	time, used when defining $M_2(t)$
$t_k$	s	4.66	time, used when calculation convolution integral

### Goals:

1. Determine equations of motion.
2. Determine eigenfrequencies and eigenvectors of the system. Graphically show eigenshapes. Normalise eigenvectors.
3. Make a transformation into modal coordinates.
4. Because of an injury on the first and the second part of the compressor, mentioned parts are periodically excited with function  $M_1(t)$ . Determine angular frequency of the system.

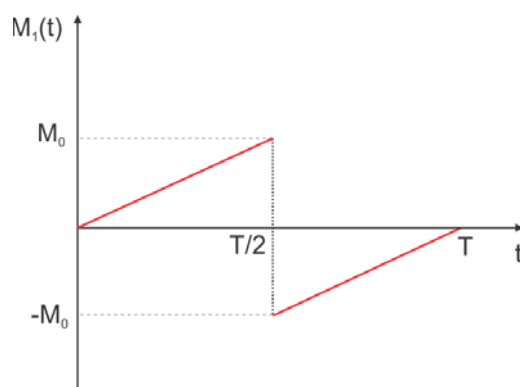


Fig. 1: Periodic function,  $M_1(t)$ .

5. Make approximation of the moment  $M_1(t)$  with Fourier series. Show it graphically.
6. Calculate the response of the system in steady state. Show it graphically.
7. On the system is now acting impulse  $M_2(t) = M_0 \tanh(-2\frac{t}{t_0})$ . Show the response of the system graphically.

## 2.2 Equations of motion

In order to be able to start solving any problem, we have to first make a model. It should be as simple as possible, but we should not neglect its important features.

If we make a discretization of a compressor with  $N$  elements, then this system has  $N$  degrees of freedom. In our case, that means 8 degrees of freedom.

The equations of motion of a vibrating system can be often derived in terms of generalized coordinates by the use

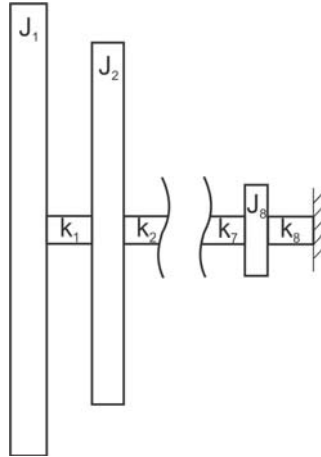


Fig. 2: Simplified model.

of Lagrange's equations. To be able to use them, we have to calculate a kinetic energy of each disk and potential energy of each torsion spring [4] as follows:

$$E_{kin} = \frac{1}{2} \sum_{n=1}^N J_n \dot{\varphi}_n^2, \quad (1)$$

$$E_{pot} = \frac{1}{2} \left[ \sum_{i=1}^{N-1} k_i (\varphi_i - \varphi_{i+1})^2 \right] + k_N \varphi_N^2. \quad (2)$$

We combine both expressions in Lagrangian - a function that summarizes the dynamics of an entire system. It is defined as  $L = E_{kin} - E_{pot}$ . Lagrange's equations can be now stated as

$$\frac{d}{dt} \left( \frac{\partial L}{\partial \dot{\varphi}_i} \right) - \left( \frac{\partial L}{\partial \varphi_i} \right) = 0, \quad (3)$$

We obtain a system of  $N$  equations of motion. We write them in a matrix form as

$$[M] \{\ddot{\varphi}\} + [K] \{\varphi\} = \{0\}, \quad (4)$$

where  $[M]$  and  $[K]$  represent a mass and stiffness matrix, respectively;  $\varphi$  and  $\ddot{\varphi}$  are a twist and acceleration vector, respectively.

$$M = \begin{bmatrix} J_1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & J_2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & J_3 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & J_4 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & J_5 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & J_6 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & J_7 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & J_8 \end{bmatrix} = \begin{bmatrix} 0.806 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.597 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.442 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.327 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.243 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.180 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0.133 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.099 \end{bmatrix}$$



$$[K] = \begin{bmatrix} k_1 & -k_1 & 0 & 0 & 0 & 0 & 0 & 0 \\ -k_1 & k_1 + k_2 & -k_2 & 0 & 0 & 0 & 0 & 0 \\ 0 & -k_2 & k_2 + k_3 & -k_3 & 0 & 0 & 0 & 0 \\ 0 & 0 & -k_3 & k_3 + k_4 & -k_4 & 0 & 0 & 0 \\ 0 & 0 & 0 & -k_4 & k_4 + k_5 & -k_5 & 0 & 0 \\ 0 & 0 & 0 & 0 & -k_5 & k_5 + k_6 & -k_6 & 0 \\ 0 & 0 & 0 & 0 & 0 & -k_6 & k_6 + k_7 & -k_7 \\ 0 & 0 & 0 & 0 & 0 & 0 & -k_7 & k_7 + k_8 \end{bmatrix} = \begin{bmatrix} 2819 & -2819 & 0 & 0 & 0 & 0 & 0 & 0 \\ -2819 & 4908 & -2089 & 0 & 0 & 0 & 0 & 0 \\ 0 & -2089 & 3636 & -1547 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1547 & 2693 & -1146 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1146 & 1995 & -849 & 0 & 0 \\ 0 & 0 & 0 & 0 & -849 & 1478 & -629 & 0 \\ 0 & 0 & 0 & 0 & 0 & -629 & 1095 & -466 \\ 0 & 0 & 0 & 0 & 0 & 0 & -466 & 811 \end{bmatrix}$$

### 2.3 Eigenvalue problem

We can find the solution of (4) by assuming a solution of the form  $\varphi_i(t) = \theta_i T(t)$ , where  $\theta_i = \text{const.}$  and  $T$  is a function of time. We substitute this new form in (4) and after some mathematical steps, we obtain eigenvalue problem equation:

$$[-\omega^2[M] + [K]]\theta = \{0\} \quad (5)$$

where  $\omega^2$  represents eigenvalue and its square root  $\omega$  is natural frequency of the system. Equation (5) represents a set of  $N$  linear homogeneous equations in the unknowns  $\theta_i (i = 1, 2, \dots, N)$ . For a nontrivial solution, the determinant of the coefficient matrix must be zero:

$$\det[-\omega^2[M] + [K]] = 0 \quad (6)$$

The expansion of (6) leads to an  $N$ -th order polynomial equation in  $\omega^2$ . The solution (roots) gives  $N$  values of  $\omega^2$ ; it can be shown that all roots are real and positive, when matrices  $[M]$  and  $[K]$  are symmetric and positive definite. If  $\omega_1^2, \omega_2^2, \dots, \omega_N^2$  denote  $N$  roots in ascending order of magnitude, their positive square roots give  $N$  natural frequencies of the system. The lowest value,  $\omega_1$ , is called the fundamental or first natural frequency. In order to calculate the values of natural frequencies, we first substitute  $\omega^2 = \lambda$ , then eq. (5) becomes:

$$[[M]^{-1}[K] - \lambda[I]]\theta = \{0\} \quad (7)$$

For solving this equation in Python we don't have to calculate when the determinant is going to be zero, but we'd rather compute dynamical matrix as

$$[D] = [M]^{-1}[K] \quad (8)$$

and then with the built-in function `numpy.linalg.eig` we obtain eigenvalues and eigenvectors of this matrix.

Tab. 2: Natural frequencies

$\omega_{01}$	$\omega_{02}$	$\omega_{03}$	$\omega_{04}$	$\omega_{05}$	$\omega_{06}$	$\omega_{07}$	$\omega_{08}$
6.5391	34.2906	56.6362	76.8568	94.3898	108.6722	119.2306	125.7111

### 2.4 Modal coordinates

We start the transformation into modal coordinates by referring to the eq. (4). Physical coordinates  $\varphi$  are now being replaced with modal coordinates as

$$\{\varphi\} = [\Phi] \cdot \{\eta\}, \quad (9)$$

where  $[\Phi]$  represents modal matrix, which contains all eigenvectors;  $[\eta]$  represents a vector of modal coordinates. If we insert (9) in (4), and multiply the expression with  $[\Phi]^T$  we obtain

$$[\overline{M}] \{\ddot{\eta}\} + [\overline{K}] \{\eta\} = \{0\}, \quad (10)$$

where  $[\overline{M}]$  and  $[\overline{K}]$  represent modal mass and modal stiffness matrix, respectively. They are both diagonally. The matrices are estimated as follows:

$$[\overline{M}] = [\Phi]^T \cdot [M] \cdot [\Phi] \quad (11)$$

$$[\overline{K}] = [\Phi]^T \cdot [K] \cdot [\Phi] \quad (12)$$

$$[\overline{M}] = \begin{bmatrix} 2.373 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 4.339 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 4.951 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 6.212 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 8.706 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 14.250 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 30.253 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 116.911 \end{bmatrix}$$

$$[\overline{K}] = \begin{bmatrix} 0.0102 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.5102 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1.5880 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 3.6692 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 7.7563 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 16.8284 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 43.0079 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 184.7850 \end{bmatrix} \cdot 10^4$$

## 2.5 Harmonic analysis; Fourier series

Any periodic function of time can be represented by Fourier series as an infinite sum of sine and cosine terms. If  $M_1(t)$  is a periodic function with period  $T$ , its Fourier series representation is given by:

$$M_1(t) = \frac{a_0}{2} + \sum_{n=1}^{\infty} [a_n \cos(n\omega t) + b_n \sin(n\omega t)] \quad (13)$$

$$a_n = \frac{2}{T} \int_0^T M_1(t) \cos(n\omega t) dt \quad (14)$$

$$b_n = \frac{2}{T} \int_0^T M_1(t) \sin(n\omega t) dt \quad (15)$$

Because function  $M_1(t)$  is a odd function (it holds:  $-M_1(t) = M_1(-t)$ ), coefficients  $a_0$  and  $a_n$  are equal to zero. Even though Fourier series is an infinite sum, we can approximate most periodic functions with the help of only a few terms. In Python, an interactive plot was made. User can change the number of terms and observe how the graph changes. We decided that 50 terms is the best compromise between the number of used terms and fitting the real function, Fig.3. Where function  $M_1(t)$  is not continuous, we get a bigger deviation while using Fourier series, and we can not fix that, even if we increase the number of terms.

Steady-state solution of eq. (13) is given by

$$\{x_{P3}\} = \sum_{n=1}^{\infty} \{x_{nP3}\} \quad (16)$$

$$\{x_{nP3}\} = \{X_{nP3} \sin(n\omega t)\} \quad (17)$$

$$X_{nP3}^{(j)} = \frac{b_n}{\omega_{0j}} \cdot \frac{1}{1 - \left(\frac{n\omega}{\omega_{0j}}\right)^2} \quad (18)$$

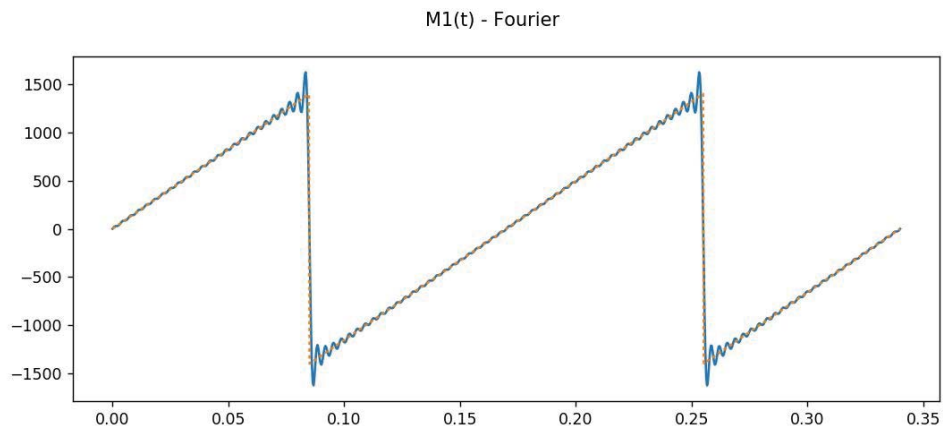
n: 

Fig. 3: Function  $M_1$  (orange) approximated with Fourier series (blue). We used 50 terms.

We have to keep in mind that we do not take into account any damping.

Because we are still dealing with multidegree-of-freedom system, we go back into the modal space. The equation is going to be similar to the eq. (10), only this time on the right side there is not a vector of zeros.

$$[\overline{M}] \{\ddot{\eta}\} + [\overline{K}] \{\eta\} = [\Phi^T] \cdot \{M_1\} \quad (19)$$

$$[I] \{\ddot{\eta}\} + [\overline{M}]^{-1} \cdot [\overline{K}] \{\eta\} = [\overline{M}]^{-1} \cdot [\Phi^T] \cdot \{M_1\} \quad (20)$$

The first and the second part of the compressor are periodically excited with function  $M_1(t)$ . We have to create a vector  $M_{1,act}$ , which has values of  $M_1$  only in the first two rows, others are zero. In that way we obtain vector  $b_n$

$$\{b_n\} = [\overline{M}]^{-1} \cdot [\Phi^T] \cdot \{M_{1,act}\} \quad (21)$$

The last step is a transformation of modal coordinates back into physical, using the relation (9), and plotting a graph.

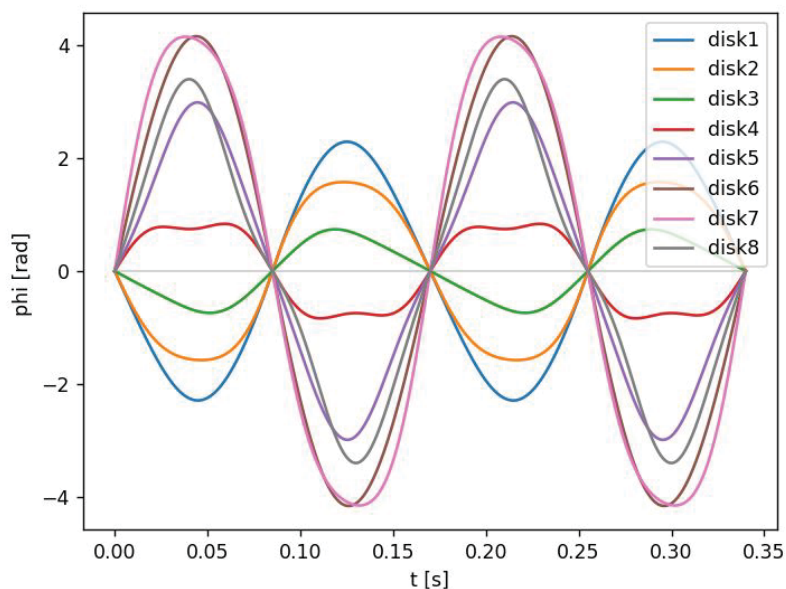


Fig. 4: Steady-state response.

## 2.6 Non-periodic force - impulse

We want to analyse the response of our system, when there is an acting impulse moment  $M_2(t) = M_0 \tanh(-2\frac{t}{t_0})$ .

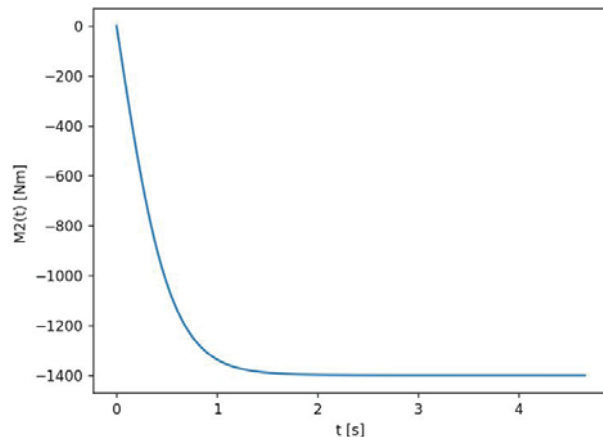


Fig. 5: Function  $M_2(t)$ .

We are solving this problem by the use of convolution integral [5]:

$$x(t) = \int_0^t m_2(\tau)g(t-\tau)d\tau \quad (22)$$

In the case of no damping, function  $g(t)$  looks like:

$$g(t) = \frac{1}{\omega_0} \sin(\omega_0 t) \quad (23)$$

We have to transform vector  $M_2$  into the modal coordinates in the same way as in the previous case of periodic function, eq. (20). Eq. (22) then looks like

$$x(t) = \int_0^t \{M_{2,mod}\}(\tau)g(t-\tau)d\tau \quad (24)$$

The last step is a transformation of modal coordinates back into physical, using the relation (9), and plotting a graph.

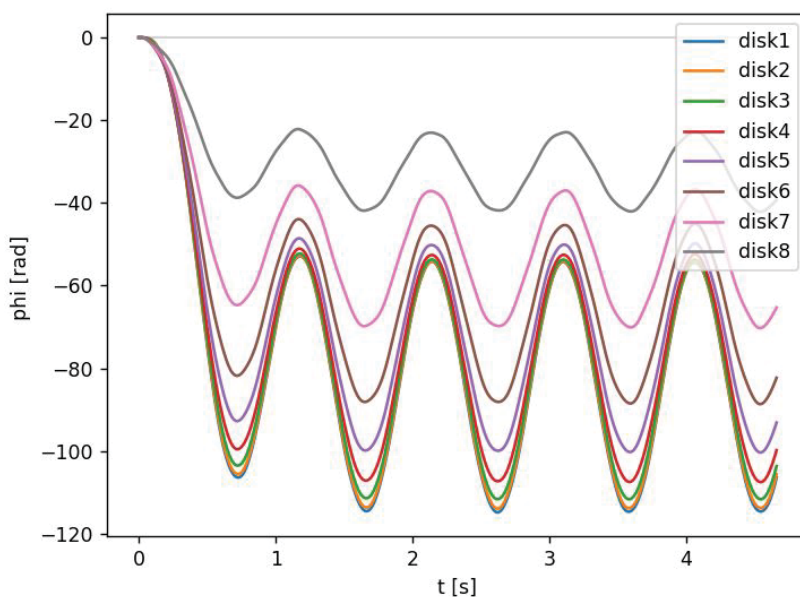


Fig. 6: Response of the system after impulse.

### 3 Conclusion

This project consists of two main parts. First, we study theoretical background of the problem and derive all the equations needed. Once that was done, we solved them with Python, creating a Jupyter Notebook. The most important packages we used were NumPy and Matplotlib.pyplot. With Numpy we created vectors and matrices and performed various operations: summation of arrays, dot products of two arrays, transposition of matrix, computation of eigenvalues and eigenvectors, sorting of values, and many more. We used Matplotlib.pyplot for plotting some graphs, for creating figures, containing several subplots and for creating an interactive plot. All the results we were able to predict from the theory were confirmed with calculations, for example: modal mass and stiffness matrices should be diagonal and they actually are. All plotted graphs also make sense, so we can conclude that our work was done correctly.

### Acknowledgement

Main author, N. Kumer, works on this project during her stay at CTU. Her main goal was to understand the theoretical background of the problem, because Mechanics is her major. Beside that she also gained a lot of knowledge about Python, and she will use it in the future for solving similar problems.

N. Kumer would like to thank the other two co-authors for all the support and help.

The related jupyter notebook can be obtained on request from authors (matous.cejnek@fs.cvut.cz).

This work was supported by the Grant Agency of the Czech Technical University in Prague, grant No. SGS18/177/OHK2/3T/12.

### Literature

- [1] Thomas Kluyver, Benjamin Ragan-Kelley, Fernando Pérez, Brian E Granger, Matthias Bussonnier, Jonathan Frederic, Kyle Kelley, Jessica B Hamrick, Jason Grout, Sylvain Corlay, et al. Jupyter notebooks-a publishing format for reproducible computational workflows. In *ELPUB*, pages 87–90, 2016.
- [2] Travis E Oliphant. Python for scientific computing. *Computing in Science & Engineering*, 9(3):10–20, 2007.
- [3] Niyazi Ari and Makhamadsulton Ustazhanov. Matplotlib in python. In *2014 11th International Conference on Electronics, Computer and Computation (ICECCO)*, pages 1–6. IEEE, 2014.
- [4] Singiresu S. Rao and Fook Fah Yap. *Mechanical vibrations*, volume 4. Prentice hall Upper Saddle River, 2011.
- [5] M. Boltežar. Nihanje linearnih mehanskih sistemov zaradi udarne motnje. 2004.

# Scalers effect on performance of standard Machine learning models

Marek Ciklamini <sup>1</sup>

<sup>1</sup> CTU FME, Praha 6, Technická 4, (Marek.Ciklamini@fs.cvut.cz)

*Abstrakt: Modely strojového učení (ML) mohou být často užitečné při zpracování různorodých dat z širokého spektra rozličných odvětví. Díky charakteru empiricky získaných dat jsou ML techniky využívány tam, kde exaktní přístupy matematického modelování nelze jednoduše použít. Tento text navrhuje základní kroky pro zvolení vhodného klasifikátoru určeného k vyhodnocování mnohorozměrného souboru dat.*

*Klíčová slova: škálování dat, dataset, modely strojového učení, přesnost modelu*

*Abstract: Machine learning models (ML) might be very useful for postprocessing various of dataset from different scientific fields. The empirical character of mined data is usually obstacle to create exact approach of mathematical model. Text suggests basic steps in order to achieve suitable classifier for evaluation of multidimensional dataset.*

*Keywords: scalers, dataset, machine learning models, model accuracy*

## 1 Úvod

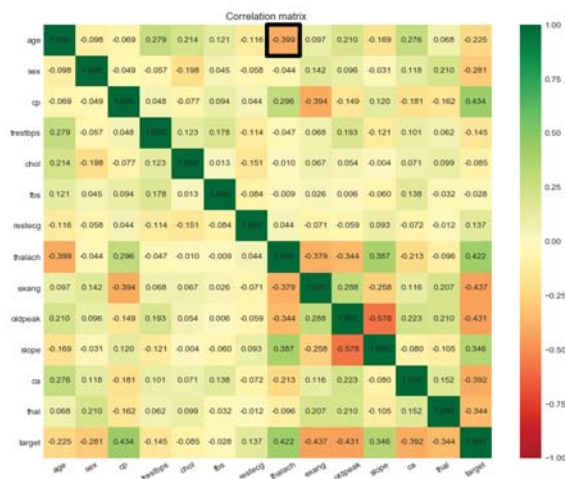
Pro ilustrativní účely a ulehčení popsání navrhovaného doporučení je vybrán soubor dat (DS) z lékařské oblasti. Heart disease DS [1] (soubor dat zabývající se srdečním onemocněním) je veřejně dostupný a často prezentovaný DS, který obsahuje v redukované podobě 14 atributů a to např. věk pacienta, pohlaví, zbytkový cholesterol, atd. a obsahuje především výstupní informaci o stavu srdečního onemocnění a to pro 303 pacientů.

### 1.1 Jak lze hledat vhodný ML model?

1. Sestavení korelační matice
2. Definování metod škálování
3. Selektce standardních ML modelů
4. Vyhodnocování základních metrik

### 1.2 Korelační matice

či taktéž pod pojmem teplotní mapa je znám prvotní indikátor, který ukazuje, zda-li je vůbec smysluplné použití standardních ML technik, a to tehdy pokud jsou přítomny mezi atributy přinejmenším náznaky korelací (hodnoty  $> \text{abs}(0.2)$ ). Nejsou-li přítomné, často selhávají, i jiné metody umělé inteligence, jako je hlubokého učení a jiné.



Obr. 1: Korelační matice DS

### 1.3 Metody vícerozměrného škálování dat (Scalers)

jsou vybrány pro jejich dostupnost v knihovně Scikit-learn [2], která je dostupná v Python jazyku.

- Standard Scaler: standardizování odstraněním střední hodnoty a škálování k rozptylu jednotlivého atributu
- MinMax Scaler: transformování vektorů pomocí jejich extrémních hodnot do daného rozsahu, např.  $\langle 0, 1 \rangle$ .



- MaxAbs Scaler: vektory jsou škálovány k maximu jejich absolutních hodnot taktéž do daného intervalu
- Robust Scaler: použití statistické metody robustní k odlehlým bodům.
- Quantile transform: transformace odhadnutím kumulativní distribuce daného parametru tak, aby měl výsledný vektor požadované rozdělení (Gaussovo, či rovnoměrné).

## 1.4 Typické klasifikátory

Oblast strojového učení je rok od roku rozsáhlejší a je mimo rozsah článku vyjmenovat a využít všechny dostupné metody. Pro demonstrování přístupu jsou vybrány následující známé metody matematické statistiky [2]:

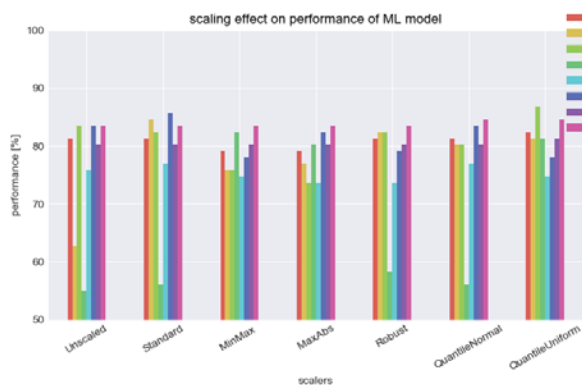
- LR ... Logistická regrese
- KNN ... Nejbližší sousedé
- SVL ... Podpůrné vektory s lineárním jádrem
- SVG ... podpůrné vekt. s radiální bázovou funkcí
- DT ... Rozhodovací strom
- RF ... Náhodný les
- AB ... Adaptive boosting
- GNB ... Najivní Bayessův klasifikátor

## 2 Vliv škálování na vlastnosti ML modelů

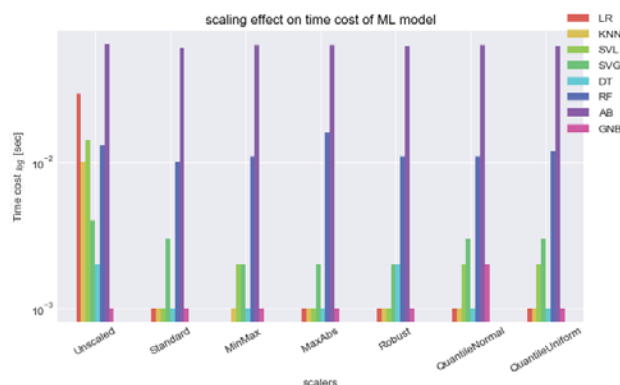
Připravené seznamy metod jsou zpracovány tak, že vznikne škálovaný DS, pro který je vždy vytvořena trénovací/testovací podmnožina v poměru 70/30% a dodatečně je vyžadováno náhodné promíchání vzorků. Jednotlivý typ ML modelu je trénován na škálovaném DS a ten je následně vyhodnocován pro jeho schopnost rozpoznat na testovací podmnožině výstup - pacientův stav.

**Přesnost** s jakou lze odhadnout v testovací množině výsledek, je nejvyšší pro SVL model a to s 86.8% , tento model je trénován na DS škálovaným Quantilovou transformací, jak lze vidět na Obr. 2

**Rychlost trénování** ML modelů pro jednotlivé metody škálování je velmi rozdílná tak, jak je patrné na Obr. 3. Hezkým příkladem je model LR trénovaný na původním DS, kdy náročnost je více jak 100 násobná oproti trénovaným modelům pro různě škálovaný DS. Tento aspekt roste na důležitosti s rozměrem DS, kde zde pro demonstrování DS je tento aspekt zanedbatelný, avšak pro DS spadající do Big Data (soubor dat mající velikost v GigaByte a více) už je škálování nevyhnutelně zapotřebí.



Obr. 2: Přesnost odhadu



Obr. 3: Rychlost trénování

## 3 Závěr

Každý model je špatný, avšak některý může být užitečný [3] a pro ML oblast toto tvrzení je opravdu výstižné. V článku byl okrajově demonstrován vliv škálování dat na důležité vlastnosti rozličných ML modelů. Je proto vhodné zdůraznit, že výsledná nejlepší kombinace škálování a modelu lze použít pouze pro daný lékařský DS, kdy na odlišném problému z podobné oblasti (přírodní, lékařské) může být s navrhovaným přístupem dosaženo podobné či stejné kombinace, avšak s nízkou pravděpodobností bude obdobných výsledků dosaženo například pro výrobní data.

## Literatura

- [1] Matthias Pfisterer Robert Detrano Andras Janosi, William Steinbrunn. Heart disease data set.
- [2] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.
- [3] George Box. *Journal of the American Statistical Association*, 1976.

# TECHNOLOGIE BLOCKCHAIN A JEJÍ POUŽITÍ K ARBITRÁŽNÍM OBCHODŮM (BLOCKCHAIN TECHNOLOGY AND ITS USE FOR ARBITRATION DEALS)

*Josef Kokeš<sup>1</sup>*

<sup>1</sup> ČVUT Praha, josef.kokes@fs.cvut.cz

*Abstrakt: Cílem této práce bylo zjistit, zda lze reálně napsat automatizovaný systém pro arbitrážní obchody na trhu s kryptoměny. V článku jsou vysvětleny pojmy nutné k pochopení problematiky a fungování technologie blockchain. V další části pak je naznačena funkcionální aplikace a shrnuty základní výsledky.*

*Klíčová slova: peníze, kryptoměna, blockchain, arbitrážní obchod, burzy*

## 1. Úvod

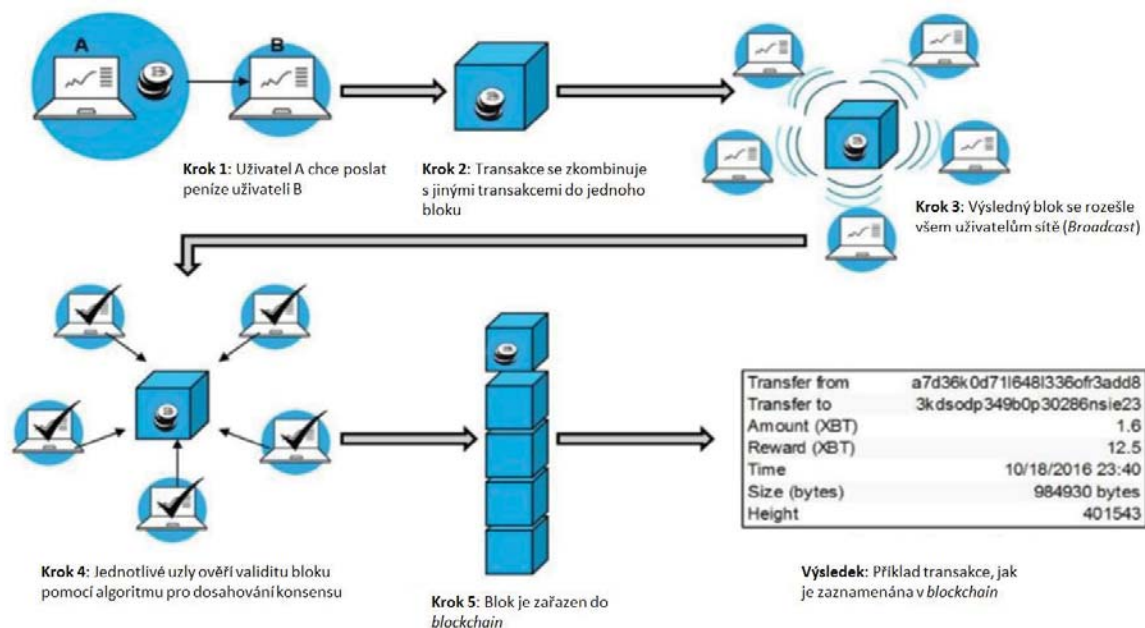
V poslední době je věnována pozornost technologii, zvané *blockchain*, zejména pak možnostem její aplikace na arbitrážní obchody s kryptoměny. Sdílená kontrola a praktická nezkorumpovatelnost této technologie umožňuje více stranám (případně i soupeřícím či nepřátelským) spolupracovat k oboustranné spokojenosti. Toho je využito k realizaci umělých platidel, kryptoměn (*cryptocurrency*).

## 2. Pojem *blockchain*

Označení *blockchain* má z technologického pohledu dva základní významy. Prvním z nich je označení množiny softwarových protokolů, které umožňují realizaci fungování *blockchainu* jako technologické platformy. Druhým významem je pak označení *blockchainové* databáze, nad kterou jednotlivé *blockchain* protokoly operují. Mezi klíčové vlastnosti *blockchainu* patří:

1. Systém funguje bez centralizované důvěry.
2. Všichni znají stav všech ostatních účtů.
3. Transakce jsou schvalovány konsensuálním způsobem, který je reprezentován procesem těžení.
4. Autenticita transakcí je chráněna asymetrickou kryptografií.
5. Za jednotlivé transakce jsou odváděny transakční poplatky.
6. Databáze je distribuována mezi jednotlivé účastníky.
7. Integrita databáze je silně chráněna procesem těžení a řetězením bloků.

Princip *blockchainu* schématicky znázorňuje obrázek 1 na následující stránce.

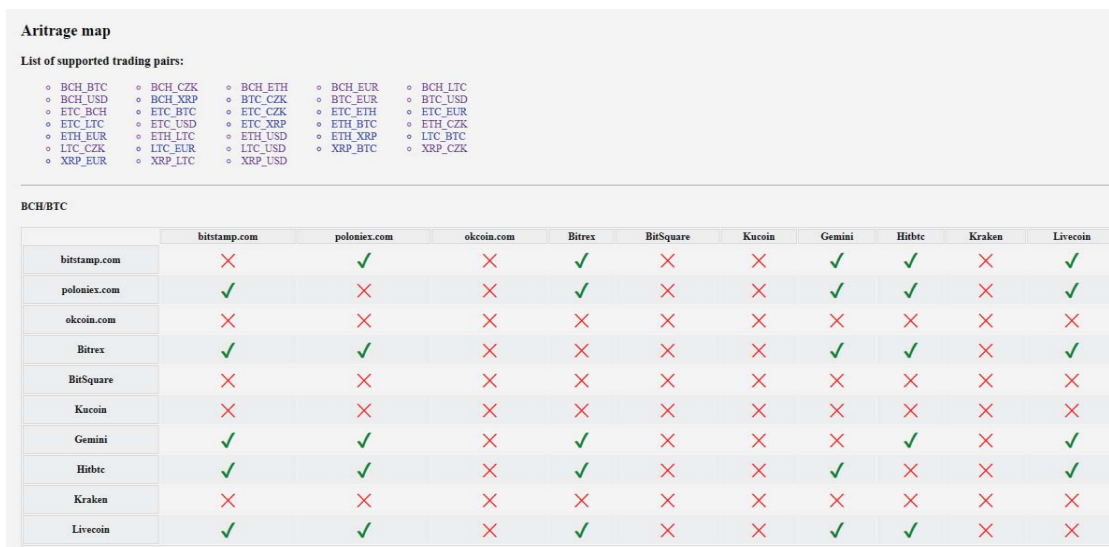
Obrázek 1 Princip technologie *blockchain*

### 3. Kryptoměny

Různých kryptoměn dnes existují desítky. K nejznámějším patří:

- **Bitcoin** se považuje za nejstarší kryptomenu. Bitcoin samotný byl definován v oficiálním dokumentu "white paper", volně dostupném na <https://bitcoin.org/bitcoin.pdf>.
- **Bitcoin Cash (BCH)** vznikla jako hard fork z Bitcoinu. Hard fork je tvrdé rozdělení existující měny na základě různé představy s fungováním měny. Změna spočívala v tom, že těžaři se dožadovali možnosti zpracovávat více transakcí v jednom bloku.
- **Ethereum** můžeme označit jako kryptomenu druhé generace. Jeho idea byla v tom, aby se výpočetní výkon počítačů potřebných pro dosažení konsenzu proof of work využíval efektivněji.
- **Litecoin** je měna podobná bitcoinu. Litecoin na rozdíl od Bitcoinu používá hashovací funkci Scrypt, která patří do kategorie paměťově náročných algoritmů. To znamená, že její algoritmus je paměťově náročný a současně je odolný vůči paralelizaci.
- **Ripple** usiluje o efektivnější převody mezi bankami, firmami a finančními institucemi.

Podobných kryptoměn existuje mnohem víc. Kromě toho, existují desítky decentralizovaných směnár, ve kterých se kryptoměny (všechny, anebo jen některé) průběžně obchodují a směňují navzájem. Na následujícím obrázku jsou příklady některých směnár a měnových párů, které v nich lze obchodovat:

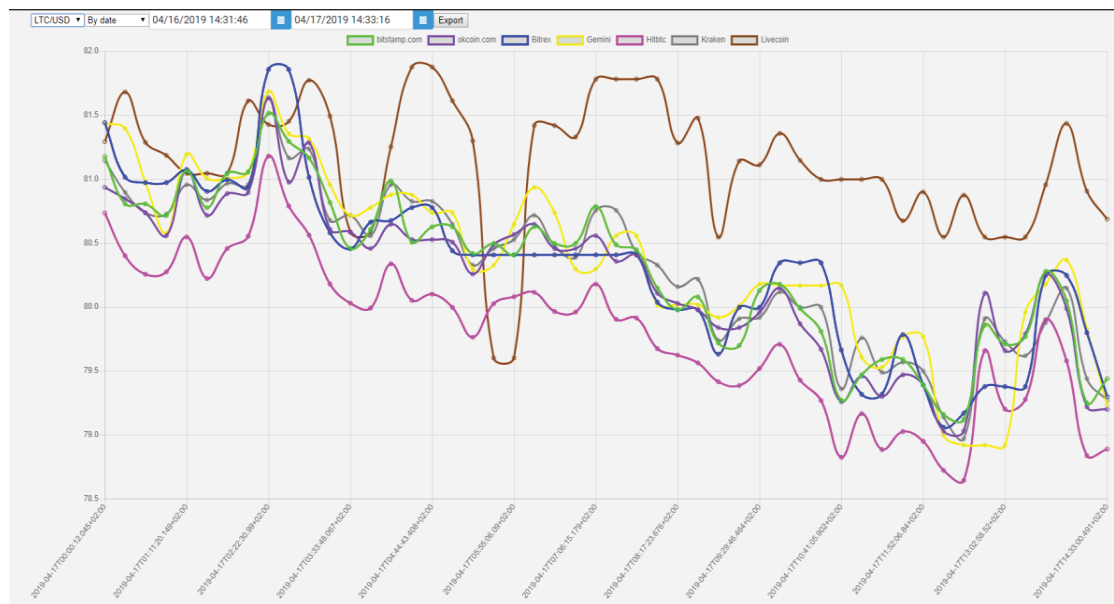


Obrázek 2 Některé směnárny a kryptoměnové páry

#### 4. Arbitrážní obchody

Odtud je už jen krůček k myšlence, porovnat ceny, za které lze různé kryptoměny v různých směnárnách v daném okamžiku zobchodovat, s cílem, dosáhnout zisku. Jedná se tedy o typickou úlohu počítačového řízení, zaměřeného na automatizaci a optimalizaci řídicího procesu. Toto je oblast, které jsme se ve svém výzkumu věnovali.

Tuto myšlenku podporuje ještě jedna okolnost – a sice, že všechny kryptoměny jsou vysoce volatilní, tzn. proměnlivé v čase. Demonstruje to 3.



Obrázek 3 Volatilita LTC proti USD v různých směnárnách

Omezujícím faktorem pro realizaci arbitrážních operací jsou jednak poplatky, jednak doby, za které lze dosáhnout potvrzení transakce. Poplatky se většinou skládají z poplatku pro burzu, který je převážně menší než 1% z transakce a z odměny pro těžaře. Tato odměna je u některých burz fixní, někde je tato

hodnota v určitém rozmezí a uživatel si může vybrat, nebo není vůbec uvedena. Všechny burzy považují transakci za uskutečněnou, až když se blok nachází dostatečně hluboko v *blockchainu*. Na základě množství potvrzení a průměrné doby řešení bloků jednotlivých měn se dopočítává trvání transakce.

Tím je způsobeno, že ne všechny transakce, které se zdají být v čase začátku transakce potenciálně v plusu, musí při zohlednění trvání transakce a měněního se měnového kurzu musí být v plusu i ve skutečnosti. Proto byla provedena počítačová simulace obchodů, při které se zohlednily jednak proměnlivé kurzu jednotlivých měn, jednak skutečná doba transakcí.

## 5. Simulace obchodů

Simulační výpočet byl proveden ex-post, to znamená, že nejprve byla z burz po určitou dobu stahována a ukládána data, načež zpětně byl proveden výpočet podle těchto dat. Výsledky okazuje obrázek 4:

Begin of transaction †	Base currency	Base currency type	Expected result of trans.	Expected profit at beginning of trans.	Duration	Transaction path	Result of trans.	Profit of trans.	Profit of trans converted to [USD]
2019-04-18T05:28:42.1213826+02:00	87.72635226516752	XRP	88.0666344367444	0.34028217157687823	02:40:00	BTC/XRP[Kucoin] ==> XRP/BTC[Livecoin]	87.30083326772923	-0.42551899743829047	-0.14656895487012556
2019-04-18T05:28:42.1213826+02:00	0.09378313894202472	BCH	0.11202076353005963	0.018237624588034906	02:40:00	BTC/BCH[bitstamp.com] ==> BCH/BTC[poloniex.com]	0.11202076353005963	0.018237624588034906	5.821162578020979
2019-04-18T05:28:42.1213826+02:00	0.09378313894202472	BCH	0.108244202312173	0.014461063370148271	03:10:00	BTC/BCH[bitstamp.com] ==> BCH/BTC[hitbtc]	0.10787376439859413	0.014090625456569405	4.497505758644943
2019-04-18T05:28:42.1213826+02:00	0.09378313894202472	BCH	0.11078239926983642	0.016999260327811697	04:50:00	BTC/BCH[Bittrex] ==> BCH/BTC[poloniex.com]	0.11078239926983642	0.016999260327811697	5.415503271931401
2019-04-18T05:28:42.1213826+02:00	0.09378313894202472	BCH	0.10703501673168779	0.01325187778966306	05:20:00	BTC/BCH[Bittrex] ==> BCH/BTC[hitbtc]	0.10869063563098472	0.014907496688959992	4.749124111199601
2019-04-18T05:28:42.1213826+02:00	0.09378313894202472	BCH	0.11117393105218946	0.017390792110164737	03:00:00	BTC/BCH[Cemini] ==> BCH/BTC[poloniex.com]	0.11117393105218946	0.017390792110164737	5.5448287541922445
2019-04-18T05:28:42.1213826+02:00	0.09378313894202472	BCH	0.10741732313435935	0.013634184192334625	03:30:00	BTC/BCH[Cemini] ==> BCH/BTC[hitbtc]	0.10711225450449437	0.013329115562469643	4.249815808880537
2019-04-18T05:28:42.1213826+02:00	0.17384759040037448	ETH	0.17528787041239774	0.001440280012023274	02:04:48	ETH/BTC[bitstamp.com] ==> BTC/ETH[BitSquare]	0.17528787041239774	0.001440280012023274	0.24922823839103986

Obrázek 4 Výsledek arbitrážních příležitostí při zohlednění trvání transakce

Na závěr jsou na následující stránce v obrázku 5 ukázány denní statistiky pro případ, že bychom provedli všechny transakce, které můžeme detekovat a které se zdají být ziskové v čase začátku transakce. Úplně první tabulka poskytuje údaje o celkové denní statistice. V řádku s názvem "*SUM of daily profit*" najdeme celkový součet všech kladných transakcí i se zohledněním délky trvání transakce. "*SUM of daily loss*" nám poskytuje informaci, jaké byly ztráty za tento den, i když v době začátku realizace transakce se zdálo, že tato transakce bude v plusu.

Řádky "*Number of daily profit*" a "*Number of daily loss*" obsahují počty, kolik bylo takových transakcí celkově za den, které byly ziskové případně ztrátové po započtení délky trvání transakce. V dalších tabulkách se nacházejí také denní statistiky, ale tyto nám ukazují, kolik bylo a jak výnosných arbitrážních obchodů pro jednotlivé arbitrážní cesty. Každá jedna z těchto transakcí je uskutečněna v množství ekvivalentním 40 amerických dolarů.

## 6. Závěr

Cílem této práce bylo, navrhnout automatizovaný systém, který by ověřil možnost arbitrážních obchodů na burzách. Byl realizován pokusný program, kterým byla tato možnost úspěšně obchodování na kryptoměnových burzách úspěšně vyzkoušena. Je napsán zatím jako nástroj, který obchoduje v teoretické rovině, tedy neuskutečňuje skutečné obchody se skutečnými penězi.

## Literatura

- [1] LÁNSKÝ, J. Kryptoměny. Praha: C.H. Beck, 2018. 312 s. ISBN 978-80-7400-722-4.
- [2] KALISKÝ, B. Bitcoin a ti druzí: nepostradatelný průvodce světem kryptoměn. Praha: IFP Publishing, 2018. 106 s. ISBN 978-80-87383-71-1.
- [3] SKALICKÝ, J. -- STROUKAL, D. Bitcoin a jiné kryptopeníze budoucnosti: historie, ekonomie a technologie kryptoměn, stručná příručka pro úplné začátečníky. Praha: Grada Publishing, 2018. 253 s. ISBN 978-80-271-0742-1.
- [4] HOSP, J. Kryptomeny. Bratislava: TATRAN, 2018. 178 s. ISBN 978-80-222-0945-8.

Total stats for day: 2019-04-28T00:00:00+02:00	
SUM of daily profit	390982.6248769814
SUM of daily loss	-636537.8688404412
Number of daily profit	73596
Number of daily loss	4498

Day stats for paths:	
<b>BTC/ETC[poloniex.com] ==&gt; ETC/BTC[BitSquare]</b>	
SUM of daily profit	14156.240014279047
SUM of daily loss	0
Number of daily profit	1340
Number of daily loss	0

<b>BTC/ETC[Bitrex] ==&gt; ETC/BTC[BitSquare]</b>	
SUM of daily profit	14244.736089759781
SUM of daily loss	0
Number of daily profit	1354
Number of daily loss	0

<b>ETC/BTC[BitSquare] ==&gt; BTC/ETC[Hitbtc]</b>	
SUM of daily profit	316.13610635352427
SUM of daily loss	-7.7661910505113925
Number of daily profit	1013
Number of daily loss	75

<b>ETC/BTC[BitSquare] ==&gt; BTC/ETC[Livecoin]</b>	
SUM of daily profit	7440.764616328014
SUM of daily loss	0
Number of daily profit	1348
Number of daily loss	0

Obrázek 5 Výsledky simulace ex-post



# VÝHODY PARALELNÍCH ROBOTICKÝCH MANIPULÁTORŮ

*David Janouch*

<sup>1</sup> ČVUT v Praze Fakulta strojní, d.janouch@seznam.cz

*Abstract: Článek srovnává sériové a paralelní manipulátory. Poukazuje na výhody jednotlivých druhů a jejich využití v praxi. Dále pojednává blíže o dvou paralelních mechanismech, které nejsou moc známé a v praxi se v podstatě nepoužívají. Jedná se o Canfield joint a Omni-wrist III.*

*Keywords: sériové manipulátory, paralelní manipulátory, Omni-wrist III, Canfield joint*

## 1 Úvod

Robotický manipulátor je robot, který dokáže provádět práci, kterou mu zadáme a pro kterou je uzpůsoben. V dnešní automatizaci je velmi často nahrazována lidská práce právě těmito manipulátory. Tyto manipulátory dokáží většinou pracovat efektivněji a bez chyb na rozdíl od člověka, ovšem potřebují vzdělanou obsluhu. Robotické manipulátory můžeme dělit podle různých prvků např. podle míry automatizace, podle odvětví, kde pracují, podle svých schopností nebo třeba podle svého vlastního kinematického modelu. Roboty rozdělíme na dvě skupiny, na které se většinou dělí, podle uspořádání jednotlivých částí, resp. podle kinematického modelu. Jedná se o dvě hlavní skupiny: robotické manipulátory sériové a paralelní.

Sériový manipulátor je v dnešní automatizaci asi nejpoužívanější. V podstatě v každém automatizovaném průmyslu takový manipulátor mají. Sériový manipulátor se vyznačuje tím, že jeho části jsou poskládány do série. Většinou se jedná o robustní stroje. Takovým nejznámějším zástupcem je šestiosý robot, viz obr. 1. Tito roboti se dají použít prakticky kdekoli v automatizaci např. svařování, překládání dílu z pásu na pás, vkládání dílů do frézy a následné vyndávání z frézy na pás apod.

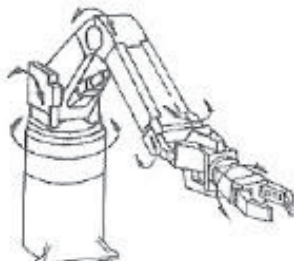


Obrázek 1: Šestiosý robotický manipulátor [1]

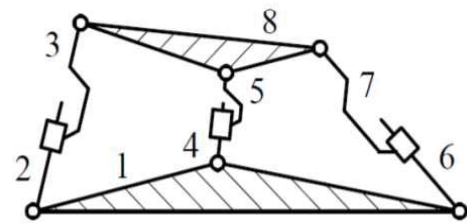
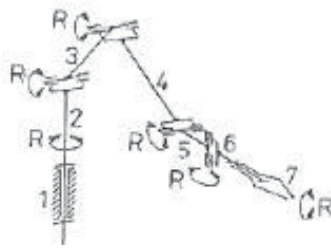
Paralelní manipulátory nejsou moc rozšířené v automatizaci. Vyznačují se tím, že mají většinou dvě základny, které jsou propojeny rameny, viz obr. 3. Většinou se jedná o tripody až hexapody, tedy třínohé až šestinohé manipulátory. Jejich kinematický model je složitý a říditelnost je obtížná. Zástupcem, který se téměř jako jediný rozšířeněji používá v automatizaci je Delta robot, viz obr. 4.

## 2 Sériové a paralelní mechanizmy – srovnání

Na první pohled je vidět, že sériové stroje nejsou nijak zanedbatelně malé. Čili první velkou nevýhodou je velikost, potažmo hmotnost dílů. Je to hlavně z důvodu, že díly jsou namáhané ve velké míře na ohyb, ovšem hlavní rozdíly jsou v přesnosti. Pokud každý díl v sérii, který zde bude zapojený, bude mít svou nepřesnost a tyto nepřesnosti se sčítají. U sériových robotů mluvíme v přesnosti řádově desetiny milimetrů. Tato úskalí se dají vyřešit právě použitím paralelní kinematiky. V přesnosti polohování jsou tyto stroje přesné na setiny milimetrů, což je řádově lepší a hlavně zanedbatelné. Dále jsou díly namáhány pouze na tah, tlak a vzpěr. Tedy dimenzování prvků je daleko jednodušší a prvky nemusí být tak masivní. Abychom se nebavili jen o kladech, tak je zapotřebí připomenout i velkou nevýhodu, a tou je velká složitost v řízení. Poté singulární body v pracovním prostoru a poměr zástavbového a pracovního prostoru. Ovšem pokud nám tyto dvě negativa nevadí a pokud vyřešíme i složitost řízení, tak nám tyto stroje mohou přinést velké výhody. Například nahradit klasický rotační pohyb, právě paralelním robotickým manipulátorem, který zajisté nebude umět jen tento pohyb.



Obrázek 2: Sériový manipulátor [3]



Obrázek 3: Paralelní manipulátor [3]

Často se používají kompromisy mezi těmito dvěma typy, a ještě častěji se používají sériové stroje. Od paralelních strojů byla obrovská očekávání, která se bohužel nenaplnila. Avšak myslím, že některé využitelné stroje stojí za zmínku. Znovu zde zmíním velmi známého Delta robota viz obr. 4. Nalezl jsem dva skoro neznámé typy mechanismů, kterými se budeme dále podrobněji zabývat.

Obrázek 4: Delta robot [4]



### 3 Omni-wrist III

První typ je takzvaný Omni-wrist III, viz obr. 5. Jedná se o mechanismus, který dokáže právě nahradit již zmiňovaný rotační pohyb. Tento manipulátor vymyslel a zkonstruoval Mark Rosheim, který si ho samozřejmě nechal i patentovat [6]. Tento manipulátor má dva stupně volnosti. Jeho mechanismus stojí pouze na rotačních vazbách. Vidíme dvě plošiny, které jsou spojeny přes dvě ramena a spojku, která je spojuje. Celkem obsahuje čtyři takto poskládaná ramena. Pohyb Omni-wristu III zajišťují dva lineární aktuátory, které pohybují s horní plošinou nebo se mohou lineární aktuátory nahradit klasickými rotačními krokovými motory.



Obrázek 5: Ideový model Omni-wristu III [5]

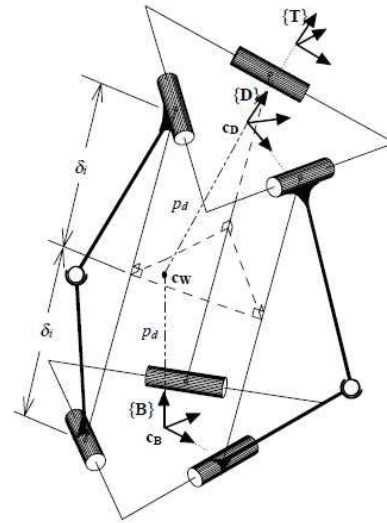
Rozebírat zde kinematiku nemá moc smysl, jelikož není úplně triviální. Za zmínku určitě stojí podobný počín, který vytvořil společně s kolegou Josef Sofka [7]. Jedná se o velmi podobný mechanismu jako je Omni-wrist III. Nazval ho Wide angle gimbal. Přeloženo do češtiny „širokoúhlý kardanový kloub“. Použití je právě v optice, kde hraje roli každá sebemenší odchylka. Dá se také použít jako stabilizační platforma nebo jako sledovací mechanismus. Dle mého názoru by se dal tento mechanismus použít v mnoha odvětvích, ovšem zatím není tak rozšířený a je pravda, že řízení není úplně jednoduché.

### 4 Canfield joint

Druhým typem, o kterém se budeme bavit a který se mi líbí ještě více, je Canfield Joint. Tento mechanismus rozpracoval ve své disertační práci Stephen Lee Canfield [8]. Tento mechanismus má z pohledu firem velkou výhodu, ačkoli byl chráněn patentem, který však v roce 2016 vypršel. Patent vypršel, protože nebyl zaplacen poplatek a vypadalo to, že se bude mnoho lidí předhánět v tom, jak tento mechanismus využijí. Ovšem očekávání, jako obvykle u těchto paralelních robotů, se nenaplnila. Jediné využití, které se plánovalo bylo pro NASA. Pro polohování motorů a tím pádem řízení ve stavu beztíže. Nenalezl jsem, jestli se toto zařízení skutečně používá anebo to byl jeden z mnoha návrhů, který se neuskutečnil.



Obrázek 6: Model Canfield jointu



Obrázek 7: Kinematický model Canfield jointu [6]

U tohoto mechanismu jsou tři stupně volnosti a skládá se ze tří ramen. Tato ramena mají dvě rotační vazby, které jsou na sebe kolmé a v podstatě nahrazují vazbu sférickou. Je nutné dodat, že pouze s vazbou sférickou by tento mechanismus nefungoval dobře. Opět zde máme dvě platformy, viz obr. 6. Zde se budeme zabývat trochu i kinematikou. Je vidět, že je tento robot symetrický. Symetrie je pro tento mechanismus nutná, aby dokázal napodobit rotační pohyb v 360 stupních. Na obr. 7. můžeme vidět kinematický model, kde je zobrazena rovina symetrie a vzdálenost průsečíku základny a desky jsou shodné a měli by být konstantní právě díky symetrii. Jelikož tento mechanismus má tři stupně volnosti (třetí stupeň volnosti je vytažení a stažení v ose z), tak správným řízením tří krokových motorů nebo třemi lineárními aktuátory funguje podobně jako Omni-wrist III a je schopen napodobovat rotační pohyb. Velkou výhodou je, že vnitřek manipulátoru je volný a při použití krokových motorů v základní rovině nemá velký zástavbový prostor.

## 5 Závěr

Shrnuto podtrženo, paralelní robotické manipulátory mají velké přednosti, ale také dvě velké nevýhody. První je špatná a složitá říditelnost a kinematika. Tou druhou je velký manipulační prostor.

Je nutné říci, že jsou paralelní mechanismy, které jsou v hojném počtu využívány. Důkazem je Delta robot. Ovšem je také nutné podotknout, že je velké množství mechanismů, které skončily v teoretické podobě nebo maximálně v podobě prototypu a jejich reálné využití je v nedohlednu. V dnešní automatizaci většinou stačí přesnost na desetiny milimetrů a setiny milimetrů jsou moc velký luxus za těžkou práci programátora s nejistým výsledkem. Ovšem v jemné mechanice nebo optice je taková přesnost potřebná. Já shledávám velký plus těchto manipulátorů, že dokáží napodobit téměř jakýkoli pohyb včetně rotačního.

Na závěr bych chtěl podotknout, že trend bude zřejmě stejný a bude velmi převládat použití velkých robustních sériových robotů, jelikož jsou pro naši výrobu dostačující, a také je tyto stroje daleko snazší naprogramovat a řídit.

## Literatura

- [1] FUNAC. LR Mate 200iD. URL <<https://www.fanuc.eu/cz/cs/roboty/str%C3%A1nka-filtru-robot%C5%AF/%C5%99ada-lmate/lmate-200-id>>.
- [2] ING. LUDĚK KOHOUT. Roboty a manipulátory - Učební text VOŠ a SPŠ Kutná Hora. URL <[http://www.edumat.cz/texty/Roboty\\_manipulatory.pdf](http://www.edumat.cz/texty/Roboty_manipulatory.pdf)>.

- [3] KOZÁK, T. Paralelní kinematické struktury. URL <[http://mech.fsid.cvut.cz/wp-content/uploads/2015/11/BP\\_Zavrel\\_Kozak.pdf](http://mech.fsid.cvut.cz/wp-content/uploads/2015/11/BP_Zavrel_Kozak.pdf)>.
- [4] OMRON. Delta robot OMRON. URL <<https://industrial.omron.us/en/products/x-delta>>.
- [5] SADEQI, SOHEIL. PARALLEL MANIPULATORS. URL <<http://www.soheil-sadeqi.com/parallelmanip/#orientation-manipulator>>.
- [6] Mark Rosheim. Omni-wrist III. URL <<https://www.anthrobot.com/anth/aboutrosheim.php>>
- [7] Josef Sofka. Wide angle gimbal. URL <<http://sofka.org/>>
- [8] STEPHEN LEE CANFIELD. Development of the Carpal Wrist; a Symmetric, Parallel-Architecture Robotic Wrist. Blacksburg, Virginia, 1997-05-21.

# MALTING PROCESS IN AN INDUSTRIAL PLANT

*Yan Shchankin*

*yanshankin@gmail.com*

*Abstrakt: Tento článek popisuje proces výroby sladu v průmyslové sladovně.*

*Klíčová slova: slad, sladování, sladovna, ječmen, máčení, klíčení, hvozdění, hvozd, pivo, pivovarnictví*

*Abstract: This article gives an overview of malt production in an industrial malting plant.*

*Keywords: malt, malting, malthouse, barley, steeping, germination, kilning, kiln, beer, brewing*

## 1 Úvod

Nejdůležitější surovinou pro pivovarský průmysl je slad. Slad je naklíčené a usušené obilné zrnó. Pro pivovarnictví se slad vyrábí převážně z ječmene. Pro různé účely a v násobně menší míře se také slad vyrábí z pšenice, čiroku, žita, ovsa, tritikále, kukuřice, prosa, rýže a jiných obilnin. Kromě piva se slad používá k výrobě whisky, octu, různých mléčných nebo ochucených nápojů a také pečených výrobků.

Dříve v každém pivovarů byly malé sladovny, v současné době se většinou staví oddělené od pivovarů vysokokapacitní sladovny, které celoročně vyrábějí slad, skladují ho a dodávají pivovarům. Pro výrobu sladu existuje celá řada různých typů zařízení.

Sladování je složitý a zdlouhavý technologický proces, a tak se v pivovarském průmyslu vyvinulo samostatné odvětví, které se výrobou sladu zabývá – sladařství (nebo také sladovnictví).

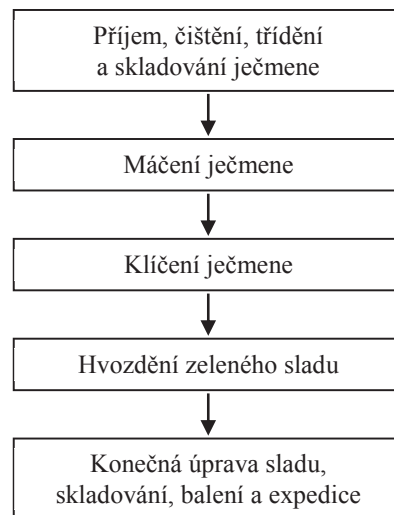
Cílem sladování je řízeným procesem za minimálních nákladů vyrobit z ječmene slad, který by obsahoval potřebné enzymy a také aromatické a barevné látky nutné pro výrobu piva. Pro tento účel je potřeba zajistit optimální podmínky pro klíčení ječmene, aby v zrnó došlo k aktivizaci a tvorbě technologicky důležitých enzymů. Dále následuje proces hvozdění, kdy vzduch se zvýšenou teplotou suší slad a vyvolává chemické reakce nutné pro chuť, vůni, barvu a kvalitu piva. Nakonec usušený slad projde konečnou úpravou a může být expedován odběrateli. Na obr. 1 je uvedeno schéma výroby sladu.

## 2 Příjem, čištění, třídění a skladování ječmene

Příjem ječmene se provádí na přijímací rampě sladovny z nákladních aut, vagonů, cisteren nebo lodí. Z každé dodávky se odebírají vzorky k analýze. Kontrolují se ukazateli jako obsah vody, bílkovin, zlomků a nečistot, klíčivost, podíl nad sítím 2,5 mm, napadení škůdci a další.

Přijímaný ječmen obsahuje polámané a cizí zrna, kamínky, hlínu, písek, pluchy, klasy, slámu, kovové příměsi a dřevo, proto je nutné ho vyčistit a roztřídit: odstranit prach, nečistoty a příměsi a roztřídit podle velikosti a kvalitativních znaků. K základnímu vybavení sladoven pro tyto účely patří: dopravní zařízení (šnekové, korečkové a pásové dopravníky, elevátory, sací či tlakové pneudopravy), automatická registrační váha, aspirátor, triér, třídiče, magnet, odklasňovač, přечиšťovací triér, jiné čističky a jímače prachu (cyklóny, proudové filtry).

Kovové příměsi (hřebíky, šrouby, matice atd.) se zachytí v **magnetickém odlučovači**. Používají se permanentní magnety, elektromagnety nebo bubnové magnety.

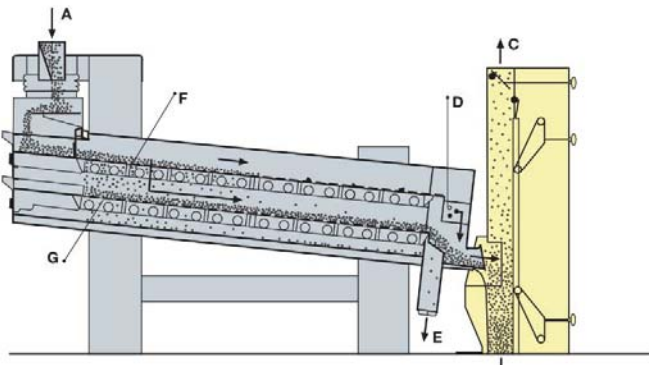


Obr. 1. Schéma výroby sladu



Prach a drobné příměsi se odstraní v **aspirátoru**, který využívá rozdílné měrné hmotnosti zrna, prachu a příměsí. Procházející zrnem vzduch odnáší prach a lehké příměsi do cyklonového odlučovače.

Polámaných a poškozených zrn a semen plevelů se zrno zbaví v **triéru** (překulovači), který představuje vodorovně se otáčející buben. Na vnitřní straně bubnu jsou důlky, kam zapadávají pouze semena plevelů a zlomená zrna. Zachycená semena jsou vznášena vzhůru, potom vypadávají do speciálního žlabu a odtud jsou odváděna pryč šnekovým dopravníkem.



Obr. 2. Čistička zrna Bühler, Separator Classifier MTRB [6]

Na obr. 2 je znázorněna **čistička** firmy Bühler, která odděluje zrno od hrubých a drobných příměsí proséváním. Žlutě je naznačen aspirátor, který přes připojený aspirační kanál odvádí prach a jiné lehké částice.

*Vysvětlení k obr. 2:*

- A přívod zrna
- B vývod vyčištěného zrna
- C připojení aspirace
- D vývod pro hrubé příměsi
- E vývod pro drobné příměsi
- F horní síto
- G spodní síto

Nakonec se zrno musí roztřídit na **sítech** na 3 druhy:

- I. Třída (prima) – zrna ječmene větší než 2,5 mm
- II. Třída (sekunda) – zrna ječmene od 2,2 mm do 2,5 mm
- Zadina – propad sítem, menší než 2,2 mm, není vhodné ke skladování a využívá se jako krmivo

Vyčištěný a vytříděný ječmen je registrován **automatickou váhou** a podle odrůdy a velikosti je skladován v silech, vybavených provzdušňovacími zařízeními. V menších skladovnách může být skladován na půdách. Uskladněný ječmen je živý organizmus, který je ovlivněn obsahem vody v zrně, teplotou, stupněm poškození, kvalitou čištění a třídění a přítomnosti plísní, škůdců atd.

### 3 Máčení ječmene

Máčení ječmene je velmi důležitou fází ve výrobě sladu a rozhoduje o konečné kvalitě sladu.

#### 3.1 Průběh máčení

Cílem je zvýšení obsahu vody z 10–15 % na 42–48 % pro zahájení činnosti enzymů a zajištění optimálních podmínek pro klíčení zrna. Zároveň se zrno omyje a zbaví se splavků a lehkých nečistot. Stupeň domočení se volí na základě druhu vyráběného sladu: pro světlý slad je to 42–45 %, pro tmavý slad je to 45–48 %.

Význam obsahu vody v ječném zrně:

- **8–10 %** konstituční voda pro zachování životaschopnosti
- **30 %** zvýšení intenzity životních pochodů
- **38 %** nejrychlejší naklíčení
- **40–48 %** rozpuštění endospermu, aktivizace a tvorba enzymů

Nejvýznamnější faktory, které ovlivňují příjem vody zrnem:

- **Teplota vody** – rozhodující faktor, čím je teplejší voda, tím je rychlejší příjem vody
- **Velikost zrna** – čím je zrno větší, tím je pomalejší příjem vody
- **Struktura zrna** – dáno odrůdou a klimatickými podmínkami ročníku (sklovitost a moučnatost)
- **Provětrávání ječmene** – vliv kyslíku a oxidu uhličitého (dýchání zrna)

**Současná technologie máčení se nazývá vzdušné máčení a má 3 fáze:**

- **1. namočení** – na 30 % obsahu vody (2-6 hodin pod vodou), následuje vzdušná přestávka 14-20 hodin
- **2. namočení** – na 38-40 % obsahu vody (6-10 hodin pod vodou), následuje vzdušná přestávka (obeschnutí)
- **3. namočení** – na 42-44 % obsahu vody (4-6 hodin pod vodou), následuje spuštění vody a 2-4 odkapávání (suché vymáčení) nebo se s třetí vodou vymáčí do pneumatických klíčidel (mokré vymáčení).

### 3.2 Zařízení máčření

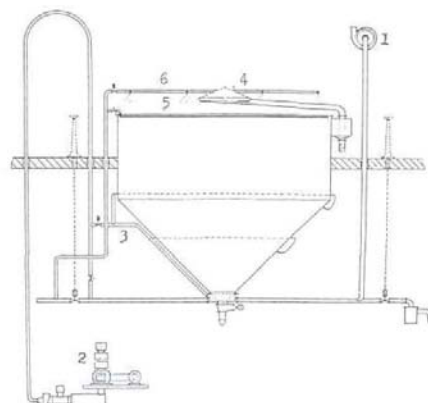
Staví se máčírny různých konstrukcí a mohou být jednodenní, dvoudenní, případně i třídenní. Máčení se provádí ve speciálních nádobách – náduvnících (také označovány jako máčecí štoky), které zajišťují dostatek vody a vzduchu, resp. kyslíku nutného pro dýchání zrna. Dneska se náduvníky většinou staví z legovaných ocelí, dříve též byly kamenné a betonové.

#### Pračka ječmene

Pračka ječmene (výkon je 6 tun/hod) je součástí moderních máčření a představuje šikmo umístěný šnek, ve kterém voda omývá ječmen a odstraňuje splavky. Velkou výhodou je úspora vody a možnost nahrazení první fáze namočení v náduvníku.

#### Kónický náduvník

Kónický náduvník představuje válcovou ocelovou nádobu se šikmým dnem pro jednodušší vyprazdňování. Kónické náduvníky se vyrábějí s maximální kapacitou do 50 tun namočeného ječmene a hloubkou do 6 m. Na obr. 3 je zobrazeno schéma kónického náduvníku: 1 – odsávání CO<sub>2</sub>, 2 – větrání tlakovým vzduchem, 3 – kruhy po rovnoměrném rozvod vzduchu, 4 – rozrážecí plech, 5 – vnitřní kruh trysek pro sběr splavků, 6 – trysky tlakové vody proti prachu.



Obr. 3. Kónický náduvník [5]

#### Wildův náduvník

Wildův náduvník je vybaven centrální trubkou pro přečerpávání ječmene tlakovým vzduchem nahoru, čímž je zajištěno promíchávání a provzdušení ječmene.

#### Plochý náduvník

Velkokapacitní (až 250 tun) náduvník s rovným děrovaným dnem. Je vybaven univerzálním strojem – žiraklérem (giracleur), který nastírá, vyklíží, dokrápí ječmen a čistí samotný náduvník. Výhodou je velká kapacita a homogenita, nevýhodou je vyšší spotřeba vody (musí být zaplaven prostor pod děrovaným dnem).

## 4 Klíčení ječmene

Cílem sladařského klíčení je aktivizace a syntéza enzymů v zrně a docílení požadovaného rozluštění zrna za minimálních nákladů a ztrát.

### 4.1 Průběh klíčení

#### Aktivizace a syntéza enzymů, látkové přeměny při klíčení

Nárůst aktivity a tvorba nových enzymů je nejdůležitějším procesem při sladování. Hlavními enzymy sladu jsou amylasy ( $\alpha$ -Amylasy,  $\beta$ -Amylasy), pomocí kterých později jsou odbourány škroby (rozluštěný škrob je zdrojem extraktu při přípravě piva). Aktivizace a syntéza enzymů vyžadují dostatečné množství metabolické energie, proto v průběhu máčení a na počátku klíčení je potřeba dostatek kyslíku pro dýchání zrna. Množství vzniklých při klíčení amylas je dáno odrůdou ječmene, klimatickými podmínkami ročníku, obsahem vody v zrně a technologickým procesem vedení klíčení (více amylas vzniká při studeném klíčení, i když vyšší teploty pro činnost enzymů působí příznivě).

Během klíčení za součinnosti enzymů probíhá proces rozluštění zrna (rozštěpení vysokomolekulárních látek na jejich štěpné produkty). Převážně se jedná o rozrušení buněčných stěn a rozštěpení škrobových zrn a bílkovinných řetězců. Tyto produkty štěpení se v omezeném množství spotřebovávají pro výživu zárodka a růst kořínků a klíčku (sladovací ztráty).

#### Růst klíčku a kořínků

Klíček (střelka) prorůstá pod pluchou zárovou stranou ke špičce obilky. Délka střelky u plzeňského sladu musí být od  $\frac{2}{3}$  do  $\frac{3}{4}$  délky zrna, u tmavého – od  $\frac{3}{4}$  do celé délky zrna. Přerostlá zrna jsou nežádoucí, protože jsou křehká a jsou následně zdrojem zlomených zrn a sladového prachu. Stadia růstu kořínků při klíčení:

- **Mokrá hromada** – vymočený ječmen
- **Suchá hromada** – do 24 h po vymočení se objevuje hlavní zárodečný kořínek (hromada špičkuje)
- **Pukavka** – růst dalších kořínků, intenzivní dýchání, hromada vyžaduje dostatek vzduchu (kypření), má okurkovou vůni

- **Mladík** – nejdůležitější stadium, intenzivní dýchání a enzymatické přeměny, kypření hromady
- **Vyrovnaná hromada** – délka kořínku a stříšky se vyrovnává, dýchání se zpomaluje
- **Stará hromada** – dýchání se nadále zpomaluje, postupně zavádění kořínků, kypření hromady, cílem je udržení v hromadě potřebné koncentrace oxidu uhličitého, který omezuje dýchání a růst vegetativních orgánů.

#### Základní parametry, které ovlivňují průběh klíčení:

1. **Vlastnosti** ječmene – dáno odrůdou, klimatickými podmínkami ročníku, podmínkami při skladování, čištění, třídění a máčení
2. Úprava **obsahu vody** – kropením hromady, větráním zvlhčeným a ochlazeným vzduchem (pro světlý plzeňský slad je obsah vody obvykle 43-45 %, pro tmavý bavorský slad 48-50 %)
3. Úprava **teploty** – obracením, nakypřením, větráním zvlhčeným a ochlazeným vzduchem
4. Úprava **obsahu kyslíku a oxidu uhličitého** – obracením, nakypřením, větráním zvlhčeným a ochlazeným vzduchem
5. **Délka** klíčení – nejčastěji 4-7 dní

Existují různé technologie klíčení: při konstantní, vzestupné, sestupné a vzestupně-sestupné teplotě. Rozdělení technologií podle hodnot teploty:

- Klíčení **studené** – do 12 °C
- Klíčení **při střední teplotě** – v rozmezí 14-18 °C
- Klíčení **teplé** – při vyšší teplotě, do 22 °C

#### 4.2 Typy sladovadel

Pro klíčení ječmene existuje mnoho různých zařízení, lze je rozdělit podle různých hledisek:

- Podle typu: klasická (humna), pneumatická (linka posuvné hromady, bubnová, skříňová, věžová klíčidla)
- Podle pohybu suroviny: sladovadla horizontální nebo vertikální
- Podle plynulosti procesu: hvozdy periodické, polokontinuální nebo kontinuální

##### Humna

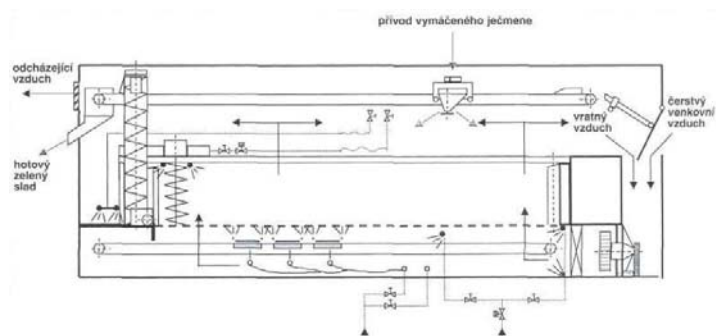
Klíčení na humnech je dneska již historický způsob klíčení na betonové nebo dlážděné podlaze. Vzduch je přiváděn studenými tahy nad podlahou a odváděn teplými tahy pod stropem. Na kropení ječmene je pod stropem rozvod vody s tryskami. Zrekonstruované humny mají kolejnice, po kterých jezdí zařízení, které může provádět namáčení, krápění, kypření, sbírání sladu nebo mytí humen. I přes použití částečné mechanizace procesu (vymáčecí vozíky, maltomobily, sedláčky) převažuje zde ruční práce a je obtížné splnit moderní požadavky trhu na objem a homogenitu vyráběného sladu.

##### Bubnové klíčidlo

Bubnové klíčidlo představuje horizontální nebo vertikální ocelový válec, ve kterém klíčení probíhá za pomalého a stálého otáčení, čímž je zajištěno promíchávání a obrácení ječmene. Do bubnu přes děravý plášť nebo větrací potrubí je vhnán zvlhčený a ochlazený vzduch, který prochází celou hromadou. Moderní bubnová klíčidla jsou již velkokapacitní a plně automatizovaná, ale energeticky náročná.

##### Skříňové klíčidlo

Skříňové klíčidlo je nejpoužívanější typ zařízení pro klíčení ječmene, představuje pevně zabudovanou sladovací skříň s dvojitým dnem. Plnění ječmenem se provádí shora. Zvlhčený a ochlazený vzduch je vhnán ventilátorem pod děrované dno, vzduch prochází celou vrstvou ječmene (sladu) nahoru. Řízením regulačních klapek vzduch může být částečně vrácen. Kypření je realizováno pomocí pojízdného šnekového obračeče, ke kterému je obvykle připojen vyklízeční šnek. Vyklízení může být řešeno i jiným způsobem, někdy tuto operaci plní obračeč. Kropení je pomocí trysek pod stropem, moderní klíčidla jsou navíc vybavena mycím strojem. Takové skříňe mohou mít různé konstrukce, např. kruhového půdorysu a být součástí věžové kruhové sladovny, jiným typickým příkladem je Saladinova skříň (obr.



Obr. 4 Saladinova skříň (fa Seeger) [5]

4). Skříňová klíčidla jsou obvykle automatizované a zajišťují homogenitu, vysokou jakost a velkou kapacitu vyráběného sladu.

#### Posuvná hromada

Linka posuvné hromady se na jedné straně plní ječmenem a na straně protilehlé se odebrává zelený (naklíčený) slad. Hromada je dvakrát denně posouvána obracečem vybaveným navíc přívodem vody. Provětrávání hromady je řešeno podobně jako u sladovacích skříní. Nevýhodou je obtížná údržba linky, vyšší poškození zrna a horší homogenita sladu.

#### Další typy sladovadel

Existují i jiné sladovací zařízení, např. **kontinuální pásová sladovadla**, na kterých posupně probíhají některé fáze sladování (klíčení a následně hvozdění). Dalším významným zařízením je **automatizovaná přehrnovací skříň**, která je vybavena zvedacími lískami pro přemísťování hromady do sousední lísky, která synchronizované klesá. Převrstvování hromady je realizováno pomocí lištového dopravníku.

## 5 Hvozdění zeleného sladu

Hvozdění je závěrečnou fází výroby sladu. Cílem hvozdění je za minimálních nákladů a ztrát převést zelený slad na trvanlivý a stabilní slad, který lze skladovat.

### 5.1 Průběh hvozdění

Základem tohoto procesu je snížení vysokého obsahu vody, zastavení vegetačních pochodů (klíčení a luštění) a vytvoření aromatických a barevných látek, charakterizujících daný druh sladu. Výsledný obsah vody u světlých sladů je 3-4 %, u tmavých sladů 1,5-2 %. Obvykle proces hvozdění trvá 17 až 24 hodin v závislosti na druhu vyráběného sladu a použitém zařízení.

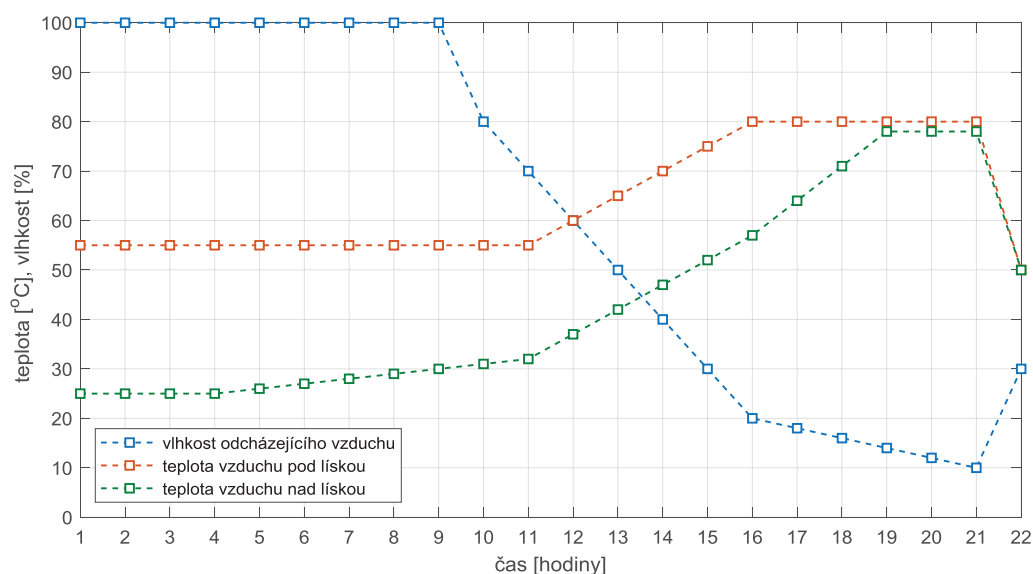
#### Proces hvozdění má dvě důležité fáze:

- Fáze **předsoušení** – snížení obsahu vody (u světlých sladů na úroveň 10-12 %). Zde je potřeba zajistit nízkou relativní vlhkost vstupujícího vzduchu a velký objemový průtok pro odstranění nevázané (volné) vlhkosti na povrchu zrna. U světlých sladů teplota sušícího vzduchu v této fázi není rozhodujícím faktorem (max. však 55 °C). Teplota vzduchu, stejně jako rychlost sušení, ovlivňuje činnost enzymů nezbytných pro daný druh sladu. Proto při výrobě sladu bavorského typu předsoušení musí být pomalejší než u sladu plzeňského typu.
- Fáze zvyšování teplot a dotahování – během této fáze se tvoří aromatické a barevné látky, charakterizující daný druh sladu. Rozhodující je zde teplota vzduchu a míra rozluštění zrna, protože dochází k odsoušení vázané vlhkosti obsažené uvnitř zrna. Rychlost sušení na rozdíl od fáze předsoušení již není konstantní a je klesající. Konečná dotahovací teplota u sladu plzeňského typu je 80-85 °C, u sladu bavorského typu až 105 °C. Po skončení dotahování je nutno slad zchladit na 50-55 °C.

### 5.2 Typy hvozdů

Hvozdy lze rozdělit podle různých hledisek:

- Podle uspořádání lísek: hvozdy horizontální nebo vertikální
- Podle počtu a tvaru lísek: hvozdy jednolískové, dvoulískové nebo třílískové, hvozdy pravoúhlé nebo kruhové
- Podle zatížení lísek: hvozdy běžné, normální nebo vysokovýkonné
- Podle způsobu a druhu ohřevu: hvozdy s přímým nebo nepřímým ohřevem, hvozdy plynové, parní, horkovodní
- Podle plynulosti procesu: hvozdy periodické, polokontinuální nebo kontinuální



Obr. 5. Diagram hvozdění sladu plzeňského typu, vytvořeno dle [5]

### Jednolískový hvozd

Jednolískový hvozd představuje vysokokapacitní zařízení pro hvozdění sladu pracující periodicky. Výška nastřeného sladu bývá od 0,8 do 1,2 m. Na obr. 6 je schéma jednolískového hvozdů: **A** je tlakový ventilátor, **B** je tepelný výměník, **C** je nastírací a vyklízecí šnek.

Ventilátor pod lískou nahání sušící vzduch, který prochází vrstvou sladu. Následně je vzduch odváděn pryč přes výměník tepla nebo vrácen přes recirkulační klapku a smíšen s čerstvým vzduchem. Pro chladič vzduchu je také k dispozici zvláštní klapka, umožňující proudění vzduchu bez průchodu tepelným výměníkem, který je pro ventilátor tlakovou ztrátou.

*Výhody:* nižší investiční náklady, jednoduché nastírání a vyklízení, automatizovaný provoz.

*Nevýhody:* vyšší spotřeba energie, nižší homogenita hvozdění způsobená vysokou vrstvou sladu.

### Dvoulískový hvozd

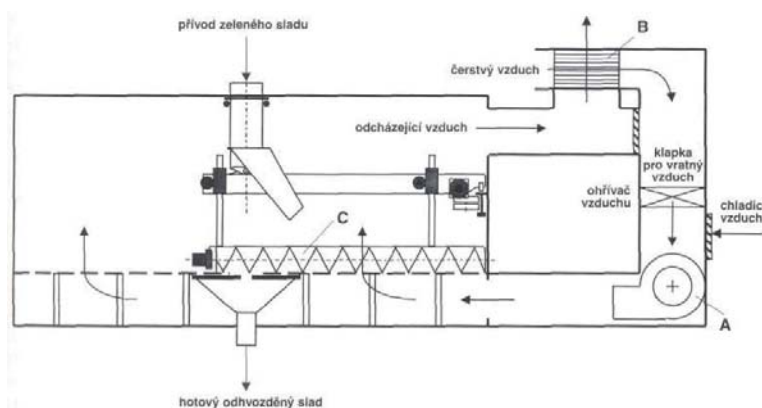
Dvoulískový hvozd z technologického hlediska je vhodnější pro výrobu světlého sladu. Na horní lísce proběhne fáze předsoušení na 10-12 % obsahu vody. Potom se slad vysype na spodní lísku, na které proběhne fáze zvyšování teplot a dotahování sladu. Přemístění sladu na spodní lísku je realizováno buď sklápěcími lískami nebo obdobným zařízením, jak na obr. 4 (nastírací a vyklízecí šnek).

*Výhody:* menší spotřeba tepla, vyšší homogenita hvozdění.

*Nevýhody:* vyšší investiční náklady, složitější konstrukce a náročnější obsluha, složitější regulace a automatizace.

### Další typy hvozdů

Ojediněle se také vyskytují hvozdů **třílískové**, u kterých spodní líska slouží k dotahování sladu a střední líska k předsoušení. Na horní lísce je slad předsoušen vzduchem o teplotě vzduchu vystupujícího ze střední lísky. Takové



Obr. 6. Jednolískový hvozd (fa Bühler) [5]



hvozdy jsou konstrukčně i provozně náročnější.

Dalšími typy jsou **skříňové** hvozdy s jednou lískou nebo **kruhové** hvozdy (jednolískové i dvoulískové) jako součást věžových sladoven. Existují také hvozdy **kontinuální**, které mohou být **horizontální** nebo **vertikální**. Základním principem kontinuálních hvozdu je postupné přemísťování sladu po dopravních pásech do různých teplotních pásem. Například: v prvním pásmu je teplota sušícího vzduchu 50 °C, ve druhém 67 °C, v třetím 81 °C a ve čtvrtém 85 °C, v každém pásmu se slad zdrží po dobu 6 hodin.

## 6 Konečná úprava sladu, skladování, balení a expedice

Konečná úprava sladu spočívá v odkličování sladu, sběru sladového květu, čištění a leštění sladu před balením a expedicí.

### Odkličování sladu

Na odkličovacích zařízeních se usušený slad zbaví kořínků, poškozených zrn a prachu. Odkličovačka je tvořena válcem s odkličovacími perutemi nebo upraveným šnekovým dopravníkem. Ulamovaný sladový květ, pluchy a poškozená zrna spadají perforovaným pláštěm do spodní části odkličovačky. Odkličovačka je připojena k aspiračnímu kanálu, který odsává lehké částice do cyklonového odlučovače. Těžší částice jsou potom vytříděny na sítích. Sladový květ se využívá pro zpracování v droždárnách, ve farmakologii a k přípravě krmiv (ze 100 kg sladu se získá 4-6 kg květu).

### Skladování sladu

Následně odkličený slad se váží na automatické váze a uskládňuje se v silech (dříve na půdách). Pro dobrou skladovatelnost slad musí být dobře odkličený, suchý a studený. V silech se slad nechává odležet (dozrát) po dobu 4-6 týdnů. Pro stálou kvalitu dodávaného sladu jednotlivé šarže sladu jsou navíc promíchávány pomocí systémů dopravníků.

### Balení a expedice

Před balením a expedicí je nutné odstranit zbytky nečistot sladu (sladový prach a pluchy). Dočistění se provádí na lištičkách pomocí kartáčů nebo na sítích, zařízení je navíc připojeno k aspiraci s cyklonovým odlučovačem. Vyčištěný slad pokračuje do expedičních sil, potom je balen a expedován odběrateli (příp. expedován volně ložený).

## 7 Závěr

Tento článek popisuje proces sladování a poskytuje přehled používaných zařízení pro výrobu sladu v moderní průmyslové sladovně. Proces výroby sladu lze rozdělit na 5 úseků, každému úseku je v článku věnována zvláštní kapitola, popisující průběh daného úseku sladování a typy používaných zařízení. V současné době je celá řada různých variací sladovacích zařízení a různých technologických postupů pro výrobu světlých, tmavých a speciálních druhů sladů. V článku jsou představeny jak starší, tak i moderní zařízení, které lze najít v českých a zahraničních sladovnách.

## Literatura

- [1] BASAŘOVÁ, Gabriela a Jaroslav ČEPIČKA. *Sladařství a pivovarství*. Praha: SNTL, 1985.
- [2] BASAŘOVÁ, Gabriela, J. ŠAVEL, P. BASAŘ a T. LEJSEK. *Pivovarství: teorie a praxe výroby piva*. Vyd. 1. Praha: VŠCHT, 2010. ISBN 978-80-7080-734-7.
- [3] BASAŘOVÁ, Gabriela. *Sladařství: Teorie a praxe výroby sladu*. Vyd. 1. Praha: Havlíček Brain Team, 2015. ISBN 978-80-87109-47-2.
- [4] HOFFMANN, Pavel a Iva FILKOVÁ. *Výrobní linky potravinářské*. Vyd. 2. Praha: Vydavatelství ČVUT, 1999. ISBN 80-01-02003-7.
- [5] KOSAŘ, Karel. *Technologie výroby sladu a piva*. Praha: Výzkumný ústav pivovarský a sladařský, 2000. ISBN 80-902658-6-3.
- [6] BÜHLER AG. *Separator Classifier MTRB* [online]. 2019, 6 s. [cit. 2019-05-15]. Dostupné z: <https://www.buhlergroup.com/>



# ADAPTIVNÍ METODY PRO BEZSENZOROVÉ VEKTOROVÉ ŘÍZENÍ VYSOKORYCHLOSTNÍHO SYNCHRONNÍHO MOTORU S PERMANENTNÍMI MAGNETY

Zdeněk Novák

Ústav přístrojové a řídicí techniky, Fakulta strojní, ČVUT v Praze, z.novak@fs.cvut.cz

*Abstrakt:* Adaptivní metody slouží ke zlepšení vlastností řízení v aplikacích, kde nelze dosáhnout požadované kvality řízení klasickými metodami. V tomto článku jsou představeny dvě nové techniky adaptace pro zlepšení zpěťovazebného bezsenzorového řízení otáček vysokorychlostního motoru s permanentními magnety (PMSM). První metoda se zabývá návrhem a aplikací Volterrova adaptivního filtru pro odstranění harmonických složek estimovaného napětí, které mají jinak negativní vliv na estimaci otáček motoru a tak i na kvalitu jejich řízení. Druhá technika se věnuje návrhu adaptivního sledovače frekvence, tzv. ortogonálního fázového závěsu (běžně označovaného jako QPLL nebo jednoduše PLL). Ten se běžně vyskytuje s pevnými parametry, a protože rozsah otáček vysokorychlostního PMSM je široký, pevné nastavení nemusí být ideální pro celé spektrum otáček. Průběžnou adaptací lze však tento nedostatek odstranit. Navržené metody je ověřena za pomoci simulace.

*Klíčová slova:* Vysokorychlostní PMSM, adaptivní Volterrovův filtr, FPGA, QPLL, vektorové řízení

## 1 Úvod

Vysokorychlostní synchronní motory s permanentními magnety (HS PMSM) se používají v aplikacích, kde je vyžadován dobrý poměr mezi fyzickým objemem motoru a dosažitelným momentem [1, 2]. Jsou tak nasazeny jako kompresory, turbo-molekulární pumpy, obráběcí stroje, kombinované jednotky tepla a elektřiny a tak dále. Kompaktnost provedení motoru lze dále vylepšit použitím bezsenzorových metod, které odstraní nutný prostor pro umístění senzoru polohy rotoru a přídatnou kabeláž [3–5].

Během posledních třech desetiletí bylo představeno velké množství metod pro bezsenzorové řízení PMSM. Jedna z často používaných technik je estimace indukovaných napětí (back-EMF), ze kterých lze dopočítat polohu rotoru [6–16]. Samotná estimace může být založena na použití pozorovatele, a to např. Luenbergerova pozorovatele nebo pozorovatele klouzavého typu (tzv. Sliding mode observer), popř. aplikací rozšířeného Kalmanova filtru. Protože jsou estimovaná napětí dvě a jsou vůči sobě posunuta o  $\pi/2$ , lze pak polohu rotoru možné dopočítat za pomoci funkce tangens (resp. jeho inverze) a poměru indukovaných napětí. Tento přístup však často bývá doprovázen velkou citlivostí na šum v datech, a tak se častěji používá tzv. fázový závěs (PLL), který dokáže sledovat vstupní frekvenci signálu, z čehož lze dopočítat hledanou polohu rotoru.

Problém nastává, jestliže jsou estimovaná napětí zatížena další chybou, a to přítomností harmonických složek. Ty jsou v signálu přítomny z důvodu nelinearity při generování napájení motoru za pomoci střídače, a dále pak z důvodů konstrukce motoru, kde jednotlivé přechody mezi pólovými nastavci způsobují krátkodobé zeslabení magnetického pole. Z harmonických složek se nejčastěji vyskytuje 5 a 7 harmonická složka, a protože je tato frekvence blízká nominální (fundamentální) frekvenci motoru, není možné ji jednoduše odstranit. Přítomné harmonické frekvence pak negativně ovlivňují sledování polohy rotoru a promítají se přímo do ní. Z těchto důvodů bylo vyvinuto několik metod, které aplikují adaptivní odstranění harmonických složek [17–21]. Aplikuje se tak přístup, kdy estimované napětí z výstupu pozorovatele je za pomoci zpětné vazby z PLL filtrováno, a tím tak PLL sleduje pouze fundamentální frekvenci. Metody [18, 19] k tomu používají jednovrstvou neuronovou síť, tzv. síť Adaline (adaptive linear neuron), zatímco metody [17, 21] využívají adaptivního úzkopásmového filtru a adaptivního vektorového filtru.

Vysokorychlostní motory se pohybují ve relativně širokém rozsahu fundamentální frekvence. PLL jsou však navrhovány s konstantními parametry, a tak nemusejí dosahovat dobrých sledovacích výsledků. Jakmile je PLL

konfigurován na vysokou rychlost sledování vstupního signálu, znamená to, že při nižších frekvencích bude i citlivý na případný šum v signálu, což ovlivní kvalitu řízení motoru při nižších otáčkách.

Pro zlepšení kvality řízení vysokorychlostního motoru se tento článek zabývá dvěma přístupy. Prvním je návrh adaptivního Volterrova filtru pro filtraci estimovaných napětí a druhým je adaptivní PLL, které má za úkol zlepšit jeho sledovací schopnosti, a to převážně při změně rychlosti motoru a ustálení v novém stavu.

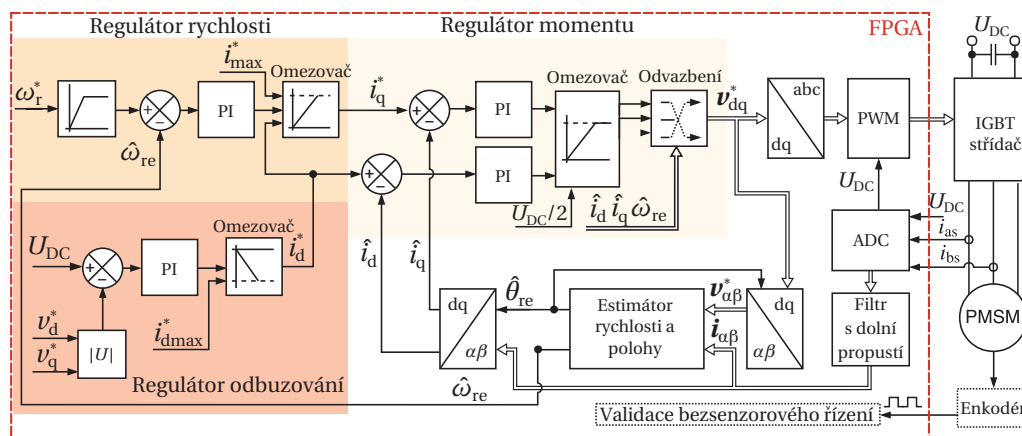
U standardních pohonů je dostačující nasazení digitálních procesorů, tzv. DSP (Digital Signal Processor). Ty však mohou dosahovat delších výpočetních časů v řádu desítek mikrosekund. HS PMSM však potřebují pro rychlou a kvalitní regulaci výpočetní časy mnohem nižší, a to v jednotkách mikrosekund. Pro tak náročné aplikace, které navíc využívají vektorového řízení (tzv. Field oriented control, FOC), se tak často používají Programovatelná logická pole (FPGA) [22–24]. Z tohoto důvodu je plánováno použití této aplikace za pomoci implementace na FPGA včetně časové optimalizace pro dosažení rychlé řídicí frekvence.

Struktura článku je následující. V Sekci 2 je představeno celkové schéma vektorového bezsenzorového řízení motoru s hlavním důrazem na estimátor polohy, popisem funkce pozorovatele a PLL. Poté je v Sekci 3 představen návrh adaptivního Volterrova filtru a v Sekci 4 pak návrh adaptivního PLL. Dosažené výsledky jsou k dispozici v Sekci 5, a to prozatím jen na základě simulací z důvodů časové náročnosti implementace na FPGA. Přínos tohoto článku je shrnut v závěru v Sekci 6.

## 2 Vektorové řízení motoru a estimátor polohy a rychlosti rotoru

Celkové schéma vektorového bezsenzorového řízení HS PMSM je zobrazeno na Obr. 1. Jedná se o kaskádovou regulaci, která je tvořena vnější smyčkou, jenž zajišťuje řízení rychlosti a jejíž výstupem je žádaný moment, a dále pak vnitřní smyčkou, která udává velikost modulovaného napětí a tím i velikost proudu ve fázi. Toto schéma je v praxi běžné a bylo již popsáno v předchozích publikacích [25, 26], nebude mu tedy věnován větší prostor.

Detail estimátoru rychlosti a polohy je zobrazen na Obr. 2. Jeho hlavními částmi jsou Luenbergerův pozorovatel proudů a indukovaných napětí, navržený adaptivní Volterrovův filtr a adaptivní PLL. Zároveň je použit blok pro normalizaci indukovaných napětí, jehož funkce bude popsána později.

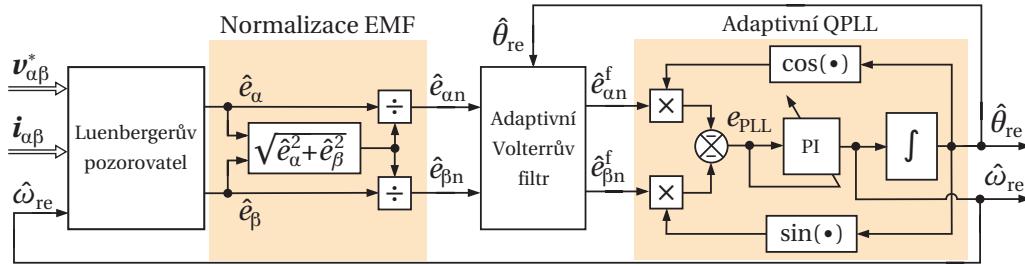


Obr. 1: Vektorové bezsenzorové řízení vysokorychlostního PMSM

Luenbergerův pozorovatel je navržen na základě modelu PMSM. Tento model motoru lze vyjádřit ve stacionární souřadné soustavě  $\alpha\beta$  jako [27]:

$$\begin{aligned}
 p i_{\alpha} &= \frac{1}{L_s} (v_{\alpha} - e_{\alpha} - R_s i_{\alpha}) \\
 p i_{\beta} &= \frac{1}{L_s} (v_{\beta} - e_{\beta} - R_s i_{\beta}) \\
 p e_{\alpha} &= -\omega_{re} e_{\beta} \\
 p e_{\beta} &= \omega_{re} e_{\alpha}
 \end{aligned} \tag{1}$$

kde  $i_{\alpha}$  a  $i_{\beta}$  jsou výsledné měřené proudy ve fázích motoru,  $v_{\alpha}$  a  $v_{\beta}$  jsou výsledná vstupní napětí motoru,  $e_{\alpha}$  a  $e_{\beta}$  jsou výsledná indukovaná napětí ve fázích motoru,  $R_s$  je odpor ve fázi motoru,  $L_s$  je synchronní indukčnost,  $\omega_{re}$  je elektrická úhlová rychlost rotoru a operátor  $p$  je náhrada za  $d/dt$ .



Obr. 2: Detail estimátoru rychlosti a polohy rotoru s adaptivními členy

Protože platí závislost mezi úhlovou rychlostí rotoru a indukovaným napětím, lze pro souměrné statorové vinutí vyjádřit indukované napětí jako:

$$\mathbf{e}_{\alpha\beta} = \omega_{re} \lambda_m \begin{bmatrix} -\sin \theta_{re} \\ \cos \theta_{re} \end{bmatrix} = E_{\alpha\beta} \begin{bmatrix} -\sin \theta_{re} \\ \cos \theta_{re} \end{bmatrix} \quad (2)$$

kde vektor indukovaných napětí je vyjádřen jako  $\mathbf{e}_{\alpha\beta} = [e_{\alpha}, e_{\beta}]^T$ ,  $\lambda_m$  je napěťová konstanta motoru,  $\theta_{re}$  je aktuální elektrická poloha rotoru.

Na základě výše uvedených vztahů a postupu uvedeného v [27], lze navrhnout Luenbergerova pozorovatele, který estimuje statorové proudy a indukovaná napětí, a který je zapsán ve formě:

$$\dot{\hat{\mathbf{x}}}_{\alpha\beta} = \mathbf{A} \hat{\mathbf{x}}_{\alpha\beta} + \mathbf{B} \mathbf{v}_{\alpha\beta}^* + \mathbf{K} \tilde{\mathbf{i}}_{\alpha\beta} \quad (3)$$

kde vektor estimovaných stavů je zapsán jako  $\hat{\mathbf{x}}_{\alpha\beta} = [\hat{i}_{\alpha}, \hat{i}_{\beta}, \hat{e}_{\alpha}, \hat{e}_{\beta}]^T$ , vektor žádaného vstupního napětí jako  $\mathbf{v}_{\alpha\beta}^* = [v_{\alpha}^*, v_{\beta}^*]^T$ , a vyjadřuje teoretická vstupní napětí, která slouží jako referenční hodnoty pro modulaci napětí střídačem (nejsou tedy aktuální vstupní napětí motoru); a vektor odchylky estimovaného a skutečného proudu jako  $\tilde{\mathbf{i}}_{\alpha\beta} = [i_{\alpha} - \hat{i}_{\alpha}, i_{\beta} - \hat{i}_{\beta}]^T$ . Systémové matice pak mají tvar:

$$\mathbf{A} = \begin{bmatrix} -\frac{R_s}{L_s} & 0 & -\frac{1}{L_s} & 0 \\ 0 & -\frac{R_s}{L_s} & 0 & -\frac{1}{L_s} \\ 0 & 0 & 0 & -\hat{\omega}_{re} \\ 0 & 0 & \hat{\omega}_{re} & 0 \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} \frac{1}{L_s} & 0 \\ 0 & \frac{1}{L_s} \\ 0 & 0 \\ 0 & 0 \end{bmatrix}, \quad \mathbf{K} = \begin{bmatrix} -k_1 & 0 \\ 0 & -k_1 \\ k_2 & 0 \\ 0 & k_2 \end{bmatrix} \quad (4)$$

kde matice zesílení  $\mathbf{K}$  zaručuje, že při volbě konstant  $k_1 > 0$  and  $k_2 = 1/L_s$ , je pozorovatel asymptoticky stabilní. Volba  $k_1$  však ovlivňuje rychlost konvergence nepřímo, a musí být vhodně zvolena s kompromisem vůči rychlosti konvergence a náchylností na šum v signálu.

Úloha normalizace estimovaných napětí se používá z nutnosti úpravy rozsahu vstupního signálu do PLL tak, aby funkce PLL nebyla ovlivněna amplitudou vstupního estimovaného napětí. PLL přenos lze pak vyjádřit jako:

$$G_{PLL}(s) = \frac{\hat{\theta}_{re}}{\tilde{\theta}_{re}} = \frac{K_p s + K_i}{s^2 + K_p s + K_i} \quad (5)$$

kde  $\tilde{\theta}_{re}$  je elektrický úhel vstupního estimovaného napětí a konstanty  $K_p$  a  $K_i$  jsou parametry PI regulátoru klasického PLL. Protože je v tomto článku navržen adaptivní PLL, bude jeho funkci věnována další pozornost v Sekci 4.

### 3 Filtrace estimovaných napětí

Fundamentální a harmonické složky indukovaného napětí lze vyjádřit jako součást modelu stroje:

$$\begin{bmatrix} u_{\alpha} \\ u_{\beta} \end{bmatrix} = \begin{bmatrix} pL_s + R_s & 0 \\ 0 & pL_s + R_s \end{bmatrix} \begin{bmatrix} i_{\alpha} \\ i_{\beta} \end{bmatrix} + \mathbf{e}_{f\alpha\beta} + \mathbf{e}_{h\alpha\beta} \quad (6)$$

kde je vektor indukovaných složek fundamentálního napětí vyjádřen jako  $\mathbf{e}_{f\alpha\beta} = [e_{f\alpha}, e_{f\beta}]^T$  a vektor harmonických složek napětí jako  $\mathbf{e}_{h\alpha\beta} = [e_{h\alpha}, e_{h\beta}]^T$ . Protože se harmonické složky vyskytují jako liché  $k$ -té násobky základní

elektrické frekvence rotoru, lze je dále rozepsat jako:

$$\begin{aligned}
 e_{\text{ha}} &= \sum_{k=1}^n E_{k\alpha} \sin((2k+1)\omega_{\text{re}}t + \theta_{(2k+1)\alpha}) \\
 &\quad + \sum_{k=1}^n E'_{k\alpha} \cos((2k+1)\omega_{\text{re}}t + \theta_{(2k+1)\alpha}) \\
 e_{\text{h}\beta} &= \sum_{k=1}^n E_{k\beta} \sin((2k+1)\omega_{\text{re}}t + \theta_{(2k+1)\beta}) \\
 &\quad + \sum_{k=1}^n E'_{k\beta} \cos((2k+1)\omega_{\text{re}}t + \theta_{(2k+1)\beta})
 \end{aligned} \tag{7}$$

kde  $E_{k\alpha}, E'_{k\alpha}, E_{k\beta}$  a  $E'_{k\beta}$  jsou amplitudy jednotlivých harmonických komponent a  $\theta$  je příslušný počáteční elektrický úhel indukovaného napětí; vše v příslušné soustavě  $\alpha\beta$ .

Tyto složky se promítanou až do určité míry i do estimované napětí. Častá tak bývá přítomnost 5. a 7. harmonické složky, ikdyž jejich velikost a počet závisí na mnoha faktorech, např. použité technice modulace napětí, nelinearitami střídače a konstrukcí motoru. Vzhledem k tomu, že se frekvence signálu mění s rychlostí motoru, není možné pro jejich odstranění použít filtry s konstantními koeficienty, a je nutné zvolit filtr adaptivní. Pro tento účel je navrhnout Volterrův filtr, který byl v nedávné době použit např. jako adaptivní filtr pro zlepšení parametrů napětí v distribuční síti [28].

### 3.1 Adaptivní Volterrův filtr

Volterrův filtr je polynomiální typ filtru, který lze zapsat jako rozvoj řady dané jako [29]:

$$y(t) = w_0 + \sum_{i=1}^S w_i X_i + \sum_{i=1, j=i}^S w_{ij} X_i X_j + \dots \tag{8}$$

kde  $w_0$  je ofset,  $w_i$  jsou koeficienty jednotlivých polynomů, které mají funkci tzv. Volterrových kernelů, a  $X_i$  jsou vstupní veličiny, které reprezentují charakteristiku vstupních dat. Výstupem filtru je  $y(t)$ , který je dán jako součet celého rozvoje řady. Protože lze výše uvedený rozvoj nekonečně rozepisovat, je pak jen důležité, do jaké míry je zapotřebí zachytit komplexnost modelovaného (aproximovaného) systému. Pro modelování jednotlivých harmonických složek bude postačovat modelování rozvoje sinu a cosinu a jejich amplitudy pak budou udávat koeficienty polynomů.

Rozvoj sinu a cosinu pro vícenásobný úhel lze získat více způsoby. V [28] byla použita Moivreova věta. V této aplikaci se jako praktičtější jeví použití vzorce pro výpočet tzv. Čebyševova polynomu prvního druhu, který slouží pro vyjádření identity kosinu vícenásobného úhlu, a jehož  $k$ -tý člen je dán jako [30]:

$$X_k(m) = \frac{1}{2} \left( \frac{1}{m + \sqrt{m^2 - 1}} \right)^k + \frac{1}{2} \left( \frac{1}{m - \sqrt{m^2 - 1}} \right)^k \tag{9}$$

kde  $X_k$  je hledaný polynom závislý na vstupní funkci  $m$ , kterou lze vyjádřit jako:

$$m = \hat{x}_{\text{cos}} = \cos \hat{\theta}_{\text{re}} \tag{10}$$

Obdobným způsobem bychom mohli definovat Čebyševův polynom druhého druhu pro identitu funkce sinus vícenásobného úhlu, který ve výsledku jen změní znaménka členů pro vybraná  $k=(3,7,\dots)$ . Pokud následně definujeme:

$$\hat{x}_{\text{sin}} = \sin \hat{\theta}_{\text{re}} \tag{11}$$

můžeme vstupní parametry Volterrova filtru pro odstranění harmonických složek definovat jako:

$$p_1 = 3\hat{x}_{\text{sin}} - 4\hat{x}_{\text{sin}}^3 = c_1\hat{x}_{\text{sin}} - c_2\hat{x}_{\text{sin}}^3 \tag{12}$$

$$p_2 = -3\hat{x}_{\text{cos}} + 4\hat{x}_{\text{cos}}^3 = -c_1\hat{x}_{\text{cos}} + c_2\hat{x}_{\text{cos}}^3 \tag{13}$$

$$p_3 = 5\hat{x}_{\text{sin}} - 20\hat{x}_{\text{sin}}^3 + 16\hat{x}_{\text{sin}}^5 = c_3\hat{x}_{\text{sin}} - c_4\hat{x}_{\text{sin}}^3 + c_5\hat{x}_{\text{sin}}^5 \tag{14}$$

$$p_4 = 5\hat{x}_{\text{cos}} - 20\hat{x}_{\text{cos}}^3 + 16\hat{x}_{\text{cos}}^5 = c_3\hat{x}_{\text{cos}} - c_4\hat{x}_{\text{cos}}^3 + c_5\hat{x}_{\text{cos}}^5 \tag{15}$$

a tak dále, až do potřebného počtu rozvoje harmonických složek.

Pro další popis funkce filtru, je vhodné použít jeho schéma, které lze vidět na Obr. 3. Vstupem filtru jsou základní hodnoty dané rovnicí (10) a (11). Ty jsou nejdříve rozvinuty v potřebné mocniny, a poté jsou vynásobeny koeficienty jednotlivých složek ( $C_{i,j} = [c_1 c_2 \dots]$ ), které jsou uloženy v paměti ve formě tabulky. Výsledkem jsou jednotlivé polynomy ( $p_1, p_2, \dots$ ), které jsou vynásobeny příslušnými váhami ( $w_1, w_2, \dots$ ) a sečteny dohromady pro získání  $y_\alpha$ . Je nutné podotknout, že každý výsledný výstup se skládá z kombinací sinů i kosinů, a tak pro odstranění např. 5. a 7. harmonické složky je zapotřebí 4 polynomů pro filtraci jednoho estimovaného napětí. Nakonec, při uvažování v diskretním čase, je pak výsledné filtrované estimované napětí v ose  $\alpha$  a v kroku  $h$ , dáno jako:

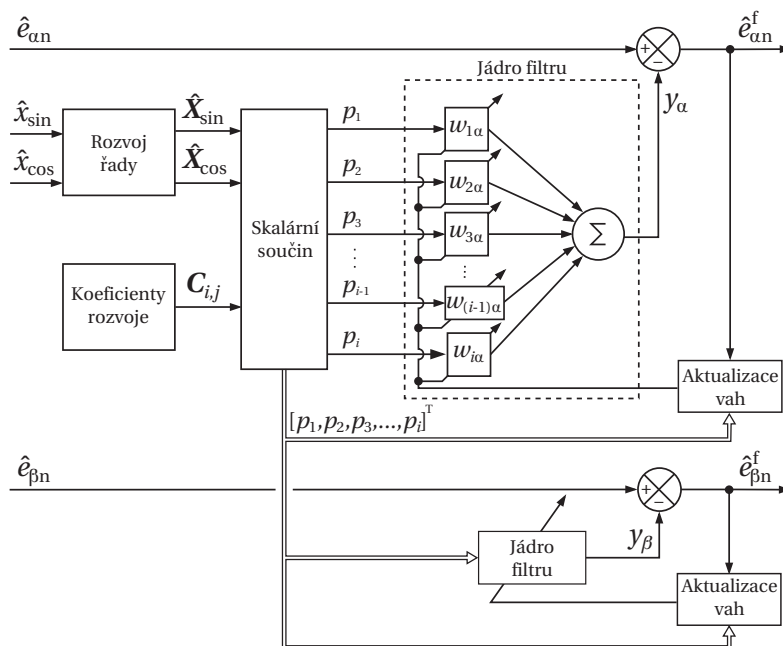
$$\hat{e}_{an}^f[h] = \hat{e}_{an}[h] - y_\alpha[h] \quad (16)$$

V procesu aktualizace vah pro  $\hat{e}_{an}^f$  lze následně použít standardní metodu LMS (Least mean squares), která je dána předpisem:

$$w_i[h+1] = w_i[h] + \mu \hat{e}_{an}^f[h] p_i[h] \quad (17)$$

kde parametr  $\mu$  určuje korekci chyby váhy  $w_i$  polynomu  $p_i$

Pro rychlejší adaptaci vah je také možné zvolit metodu RLS, jejíž princip lze najít např. v [31]. Při správné funkci filtru je výstup ze sumačního členu filtru roven rozvoji harmonických složek rovnice (7), který odečte všechny harmonické složky od estimovaného napětí, a tak zůstane pouze jeho fundamentální složka. Simulaci takového procesu lze najít v Sekci 5.



Obr. 3: Detail architektury Volterrova filtru pro kompenzaci harmonických složek

## 4 Adaptivní PLL

Sledovače frekvence jsou častým prvkem používaným v elektronice a elektrotechnice. Vyskytují se v různém provedení a to jak v "softwarové" verzi jako algoritmy, tak i realizované hardwarově. Jeden z prvních článků, který se zabývá jejich použitím v takovém tvaru, v jakém se používá pro aplikace s elektrickými motory, je pak článek [32]. Existují i jeho variace, které jsou založeny na minimalizaci odchylky chyby jako v [33], autor článku se však s nimi u aplikací s elektromotory nesetkal. Je to z toho důvodu, že běžné elektromotory fungují v relativně malém frekvenčním pásmu (ve srovnání s vysokorychlostními nebo vysokofrekvenčními), a z tohoto důvodu je pro ně postačující, pokud má PI regulátor v rámci PLL konstantní parametry.

PI regulátor a jeho konstanty se pro běžné PLL se navrhuje tak, aby PLL fungovalo v oblasti nominálních otáček motoru. Ty jsou specifikovány jednak maximální rychlostí a jednak dynamikou stroje. PLL, jehož přenosová funkce je uvedena v rovnici (5), má pak za úkol včas na změnu otáček zareagovat úpravou svého

výstupu a dorovnat odchylku vstupního signálu  $e_{PLL}$  (viz. Obr. 2). Pokud by však bylo nastaveno příliš citlivé, to jest PLL by mělo široké pásmo přenosu, reagovalo by i citlivě na jakýkoliv šum, který by byl zesílen, a tak zhoršil by parametry estimace. V tomto článku je proto navržen adaptivní regulátor tak, aby zmenšil chybu odchylky pozorovatele v rámci přechodových dějů (např. zrychlení a zpomalení motoru).

Protože je regulátor tvořen dvěma hlavními koeficienty  $K_p$  a  $K_i$ , optimalizace dvou parametrů na jednu může být nevýhodná. Proto je vhodné napsat charakteristickou rovnici pro přenosovou funkci rovnice (5) jako:

$$s^2 + 2\sigma\rho s + \rho^2 = 0 \quad (18)$$

kde  $\sigma$  vyjadřuje tlumení a  $\rho$  je ladící parametr. Tak jsou konstanty regulátoru vyjádřeny jako:

$$\begin{aligned} K_p &= 2\sigma\rho \\ K_i &= \rho^2 \end{aligned} \quad (19)$$

Pokud budeme tlumení uvažovat jako konstantní a předem zvolené, je pak možné měnit pouze ladící parametr. Adaptaci lze založit na minimalizaci druhé mocniny vstupní chyby regulátoru, která je dána jako:

$$e_{PLL}^2 = \left( -\hat{e}_{\alpha n}^f \cos \hat{\theta}_{re} - \hat{e}_{\beta n}^f \sin \hat{\theta}_{re} \right)^2 \quad (20)$$

a aktualizace parametru  $\rho$  v kroku  $h$  je tak dána použitím LMS algoritmu:

$$\rho[h+1] = \rho[h] - \eta \frac{1}{2} \left( \frac{\partial e_{PLL}^2[h]}{\partial \rho[h]} \right) \quad (21)$$

kde  $\eta$  je parametr aktualizace vah, který musí být zvolen dostatečně nízký vzhledem k vzorkovací periodě PLL. Derivací (21) podle  $\rho$ , kde je estimovaný úhel vyjádřen jako výstup PI regulátoru PLL, lze postupnými úpravami získat:

$$\frac{1}{2} \left( \frac{\partial e_{PLL}^2[h]}{\partial \rho[h]} \right) = 2T_s e_{PLL}[h-1] (\rho[h] + \sigma) \left( \hat{e}_{\alpha n}^f[h] \sin(\hat{\theta}_{re}[h-1]) - \hat{e}_{\beta n}^f[h] \cos(\hat{\theta}_{re}[h-1]) \right) \quad (22)$$

kde  $T_s$  je perioda vzorkování a kterou lze společně s koeficientem 2 dále zahrnout do  $\eta$  pro další zjednodušení. Nové koeficienty regulátoru se získají následným dosazením parametru  $\rho$  do rovnice (19).

Uvedený výsledek je platný, pokud se rychlost v rámci dvou následujících iterací příliš neliší, což je zajištěno dostatečnou vzorkovací frekvencí.

## 5 Simulace a dosažené výsledky

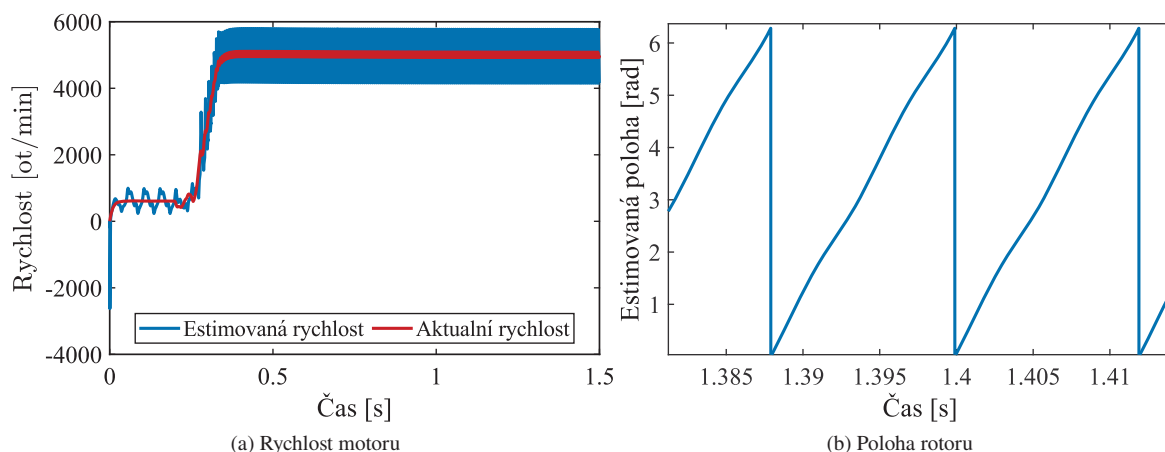
Model motoru, řídicí algoritmus a adaptivní PI regulátor pro PLL byl převeden pro simulaci do diskrétního tvaru za pomoci Tustinovo aproximace, a to při uvažování vzorkovací frekvence 250 kHz, která je stejná jako u současné hardwarové implementace bezsenzorového vektorového řízení. Adaptivní Volterrův filtr byl zpracován dle výše uvedených diskrétních rovnic a pozorovatel byl nahrazen aproximací zoh (zero-order hold).

Tab. 1: Parametry vysokorychlostního PMSM

Parametr motoru	Symbol	Hodnota
Napěťová konstanta	$\lambda_m$ [V s rad <sup>-1</sup> ]	0.0088
Odpor ve fázi	$R_s$ [ $\Omega$ ]	0.2
Indukčnost v ose $q$	$L_q$ [mH]	0.13
Indukčnost v ose $d$	$L_d$ [mH]	0.13
Jmenovitý proud	$I_n$ [A]	10
Indukované napětí na 1000 ot/min	$U_{emf}$ [V]	0.9215
Jmenovitá rychlost	$n_r$ [ot/min]	100 000
Maximální rychlost	$n_{max}$ [ot/min]	130 000
Jmenovitý moment	$T_e$ [Nm]	0.162
Moment setrvačnosti rotoru	$J$ [kg m <sup>2</sup> ]	$12 \cdot 10^{-6}$
Koeficient tření rotoru	$B_{ms}$ [Nm s rad <sup>-1</sup> ]	$29 \cdot 10^{-7}$

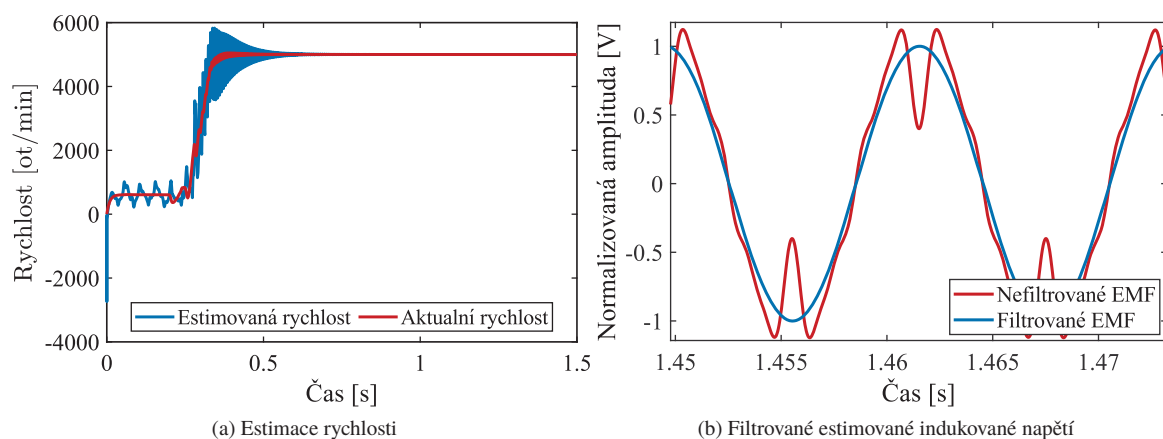


Nejdříve budou představeny výsledky simulace se samotným adaptivním Volterrovo filtrem s PLL s konstantními parametry, poté bude následovat simulace s doplněním adaptivního PLL. Motor je vždy rozeběhnut pomocí zpětné vazby od senzoru a v čase 0.2 se přepne na řízení bezsenzorové. To je nutné z toho důvodu, že estimátor funguje pouze pokud je přítomné minimální indukované napětí. Jiná možnost je motor rozeběhnout metodou U/f, pro kterou už senzor polohy není nutný, a poté se opět přepnout do bezsenzorového řízení. Protože se harmonické složky projevují převážně v nízkých otáčkách, kde je není možné odstranit dolnoproustním filtrem společně se spínacími složkami střídače, budou i testy provedeny v relativně nízkých otáčkách. Na Obr. 4 je ukázka vektorového řízení rychlosti HS PMSM, u kterého byly motorové proudy doplněny o harmonické složky (3,5,7,9,11) v sestupném poměru. Je znatelný vliv harmonických složek, které zhoršují přesnost estimace polohy a rychlosti motoru, což má negativní důsledek na regulaci jeho otáček.



Obr. 4: Estimace rychlosti a polohy rotoru při přítomnosti harmonických složek v estimovaném napětí.

Na Obr. 5 je pak stejná situace při použití adaptivního Volterrova filtru, který je po dosažení nových otáček odfiltruje harmonické složky estimovaného napětí a tak zlepšit regulaci motoru. Rychlost filtrace závisí na nastavení rychlosti adaptace vah filtru. Pokud by byla použita metoda RLS, byl by tento průběh mnohem rychlejší. Obr. 6 zobrazuje přítomnost harmonických složek před a po filtraci.

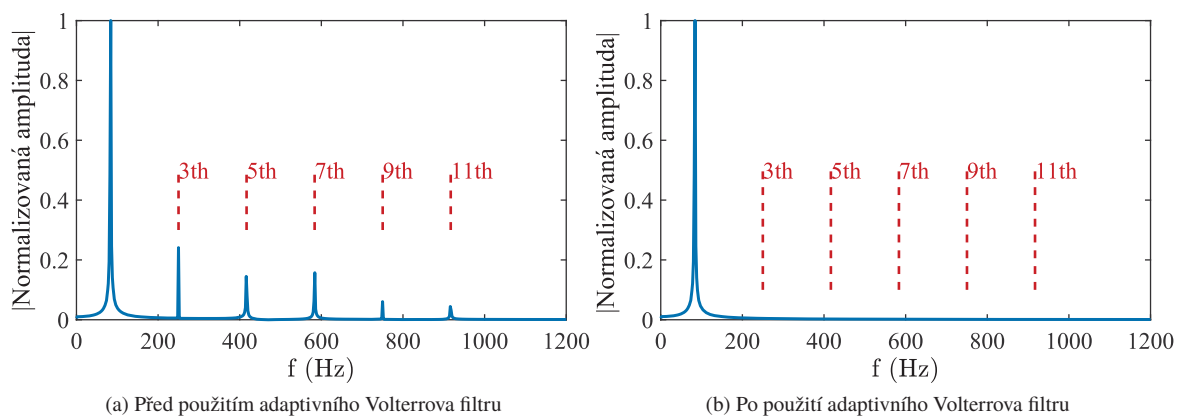


Obr. 5: Estimace rychlosti rotoru po filtraci harmonických složek.

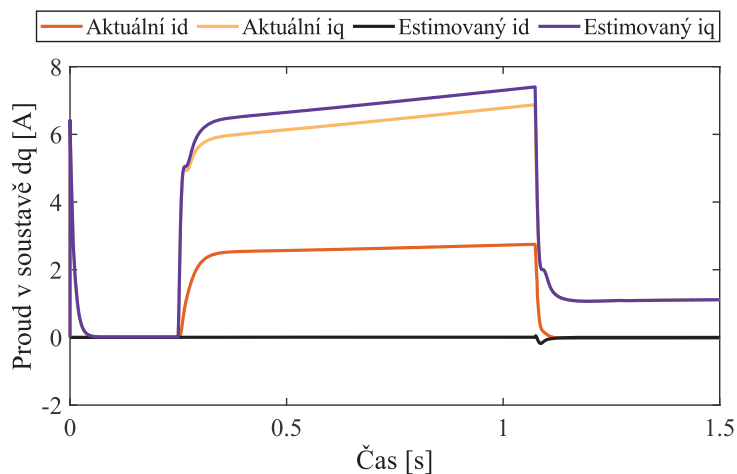
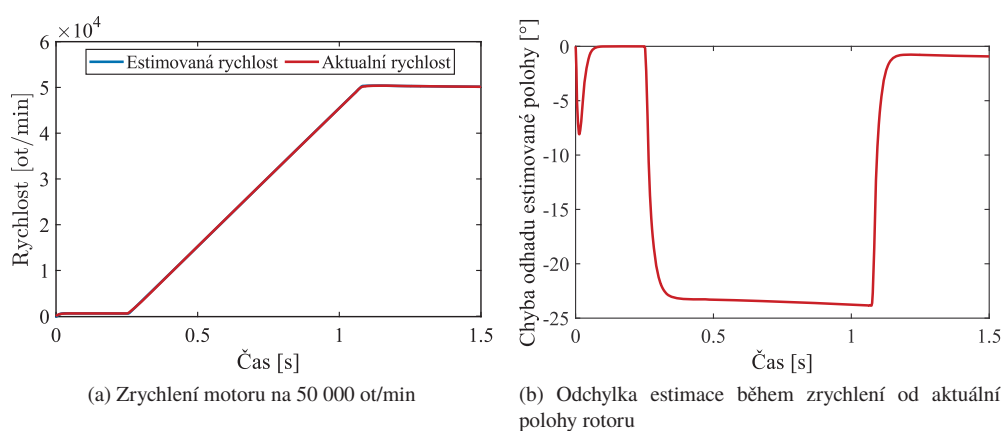
Pro lepší názornost bude simulace adaptivního PLL představena bez zahrnutí harmonických složek do estimovaných napětí. Průběh zrychlení motoru na 50 000 ot/min lze sledovat na Obr. 7. Lze sledovat opožděnou reakci sledovače frekvence, což způsobí zpoždění estimované polohy, a tak nedokonalé odstavění momentotvorné a tokotvorné složky proudu  $i_q$  a  $i_d$ .

Při změně regulátoru PLL na adaptivní lze pak sledovat podstatné zlepšení na Obr. 8.





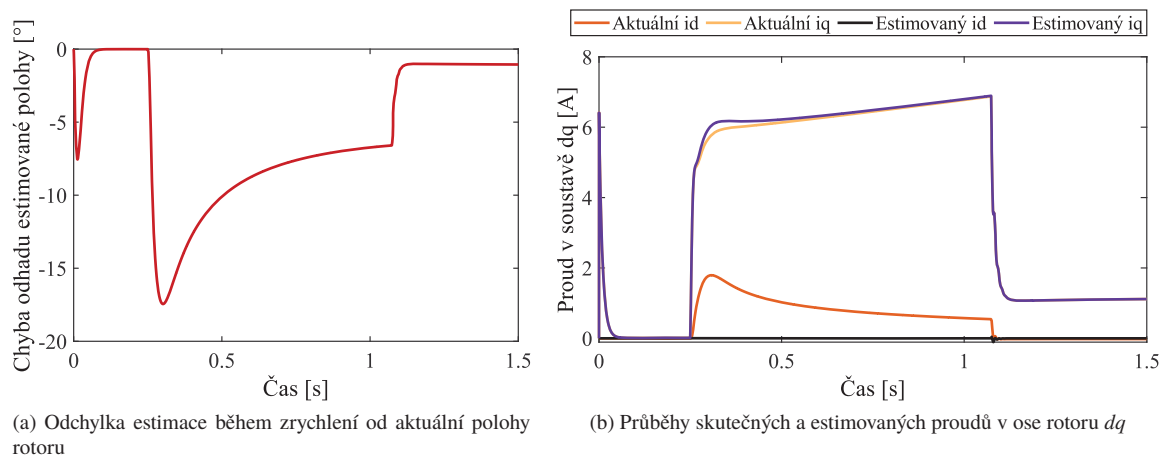
Obr. 6: FFT pro estimované indukované napětí.



Obr. 7: Zrychlení motoru s použitím PLL s konstantními parametry.

## 6 Závěr

V tomto článku byl představen adaptivní Volterraův filtr pro filtraci estimovaných napětí a adaptivní PLL. Výsledky simulací dokazují jejich schopnosti zlepšit vlastnosti řízení vysokorychlostního PMSM. Jejich schopnosti lze dále rozšířit za pomoci jiných nástrojů adaptace vah tak, aby vykazovali ještě rychlejší odezvu na změny v estimovaném signálu. Pro budoucí aplikaci bude použita implementace na FPGA a vlastní experimentální vysokorychlostní PMSM, který byl navržen speciálně pro testování bezsenzorových metod.



Obr. 8: Zrychlení motoru s použitím PLL s adaptací parametrů.

## Acknowledgement

Tato práce byla podpořena grantem Studentské grantové soutěže ČVUT č. SGS18/176/OHK2/3T/12 "Vysokorychlostní elektrické stroje pro automobilové aplikace - řízení, výkonová elektronika". Za tuto podporu autor vřele děkuje.

## Literatura

- [1] D. Krahenbuhl, C. Zwyssig, H. Weser, and J. W. Kolar. A miniature 500 000-r/min electrically driven turbocompressor. *IEEE Trans. Ind. Appl.*, 46(6):2459–2466, Nov 2010.
- [2] L. Zhao, C. Ham, L. Zheng, T. Wu, K. Sundaram, J. Kapat, and L. Chow. A highly efficient 200 000 RPM permanent magnet motor system. *IEEE Trans. Magn.*, 43(6):2528–2530, June 2007.
- [3] N. Bianchi, S. Bolognani, and F. Luise. Potentials and limits of high-speed PM motors. *IEEE Trans. Ind. Appl.*, 40(6):1570–1578, Nov 2004.
- [4] D. Gerada, A. Mebarki, N. L. Brown, C. Gerada, A. Cavagnino, and A. Boglietti. High-speed electrical machines: Technologies, trends, and developments. *IEEE Trans. Ind. Electron.*, 61(6):2946–2959, June 2014.
- [5] A. Tenconi, S. Vaschetto, and A. Vigliani. Electrical machines for high-speed applications: Design considerations and tradeoffs. *IEEE Trans. Ind. Electron.*, 61(6):3022–3029, June 2014.
- [6] S. Morimoto, K. Kawamoto, M. Sanada, and Y. Takeda. Sensorless control strategy for salient-pole PMSM based on extended EMF in rotating reference frame. *IEEE Trans. Ind. Appl.*, 38(4):1054–1061, Jul 2002.
- [7] F. Genduso, R. Miceli, C. Rando, and G. R. Galluzzo. Back EMF sensorless-control algorithm for high-dynamic performance PMSM. *IEEE Trans. Ind. Electron.*, 57(6):2092–2100, June 2010.
- [8] J. M. Liu and Z. Q. Zhu. Improved sensorless control of permanent-magnet synchronous machine based on third-harmonic back EMF. *IEEE Trans. Ind. Appl.*, 50(3):1861–1870, May 2014.
- [9] S. Po-ngam and S. Sangwongwanich. Stability and dynamic performance improvement of adaptive full-order observers for sensorless PMSM drive. *IEEE Trans. Power Electron.*, 27(2):588–600, Feb 2012.
- [10] Z. Qiao, T. Shi, Y. Wang, Y. Yan, C. Xia, and X. He. New sliding-mode observer for position sensorless control of permanent-magnet synchronous motor. *IEEE Trans. Ind. Electron.*, 60(2):710–719, Feb 2013.
- [11] S. C. Yang and G. R. Chen. High-speed position-sensorless drive of permanent-magnet machine using discrete-time EMF estimation. *IEEE Trans. Ind. Electron.*, 64(6):4444–4453, June 2017.
- [12] T. Bernardes, V. F. Montagner, H. A. Gründling, and H. Pinheiro. Discrete-time sliding mode observer for sensorless vector control of permanent magnet synchronous machine. *IEEE Trans. Ind. Electron.*, 61(4):1679–1691, April 2014.

- [13] T. Tuovinen, M. Hinkkanen, L. Harnefors, and J. Luomi. Comparison of a reduced-order observer and a full-order observer for sensorless synchronous motor drives. *IEEE Transactions on Industry Applications*, 48(6):1959–1967, Nov 2012.
- [14] M. Comanescu and T. D. Batzel. Full order EMF observer for PMSM — design, analysis and performance under improper speed signal. In *2010 IEEE International Systems Conference*, pages 86–90, April 2010.
- [15] C. Olivieri, F. Parasiliti, and M. Tursini. A full-sensorless permanent magnet synchronous motor drive with an enhanced phase-locked loop scheme. In *2012 XXth International Conference on Electrical Machines*, pages 2202–2208, Sept 2012.
- [16] S. Bolognani, S. Calligaro, and R. Petrella. Design issues and estimation errors analysis of back-EMF-based position and speed observer for spm synchronous motors. *IEEE J. Emerg. Sel. Top. Power Electron.*, 2(2):159–170, June 2014.
- [17] S. Y. Jung and K. Nam. PMSM control based on edge-field hall sensor signals through ANF-PLL processing. *IEEE Trans. Ind. Electron.*, 58(11):5121–5129, Nov 2011.
- [18] G. Wang, H. Zhan, G. Zhang, X. Gui, and D. Xu. Adaptive compensation method of position estimation harmonic error for EMF-based observer in sensorless IPMSM drives. *IEEE Trans. Power Electron.*, 29(6):3055–3064, June 2014.
- [19] G. Wang, T. Li, G. Zhang, X. Gui, and D. Xu. Position estimation error reduction using recursive-least-square adaptive filter for model-based sensorless interior permanent-magnet synchronous motor drives. *IEEE Trans. Ind. Electron.*, 61(9):5115–5125, Sept 2014.
- [20] G. Zhang, G. Wang, D. Xu, and N. Zhao. Adaline-network-based PLL for position sensorless interior permanent magnet synchronous motor drives. *IEEE Trans. Power Electron.*, 31(2):1450–1460, Feb 2016.
- [21] G. Zhang, G. Wang, D. Xu, R. Ni, and C. Jia. Multiple-AVf cross-feedback-network-based position error harmonic fluctuation elimination for sensorless IPMSM drives. *IEEE Trans. Ind. Electron.*, 63(2):821–831, Feb 2016.
- [22] E. Monmasson, L. Idkhajine, M. N. Cirstea, I. Bahri, A. Tisan, and M. W. Naouar. FPGAs in industrial control applications. *IEEE Trans. Ind. Informat.*, 7(2):224–243, May 2011.
- [23] N. K. Quang, N. T. Hieu, and Q. P. Ha. FPGA-based sensorless PMSM speed control using reduced-order extended kalman filters. *IEEE Trans. Ind. Electron.*, 61(12):6574–6582, Dec 2014.
- [24] L. Idkhajine, E. Monmasson, M. W. Naouar, A. Prata, and K. Bouallaga. Fully integrated FPGA-based controller for synchronous motor drive. *IEEE Trans. Ind. Electron.*, 56(10):4006–4017, Oct 2009.
- [25] Z. Novak and M. Novak. Design of high-speed permanent magnet synchronous motor for advanced and sensorless control techniques validation. In *2018 18th International Conference on Mechatronics - Mechatronika (ME)*, pages 1–8, Dec 2018.
- [26] Z. Novak and M. Novak. Implementation of a novel angular hall encoder for field oriented control of high speed and experimental motors. In *2016 17th International Conference on Mechatronics - Mechatronika (ME)*, pages 1–7, Dec 2016.
- [27] M. Comanescu and T. D. Batzel. Full order EMF observer for PMSM — design, analysis and performance under improper speed signal. In *2010 IEEE International Systems Conference*, pages 86–90, April 2010.
- [28] S. K. Patel, S. R. Arya, and R. Maurya. Nonlinear adaptive volterra filter for control of distribution static compensator. *IEEE J. Emerg. Sel. Top. Power Electron.*, 5(1):559–567, March 2017.
- [29] Nikolay Y. Nikolaev and Hitoshi Iba. *Adaptive learning of polynomial networks: genetic programming, backpropagation and Bayesian methods*. Springer, 2011.
- [30] Benjamin Sinwell. Chebyshev polynomials: Patterns and derivation. *Mathematics Teacher*, 98(1):20–25, 2004.
- [31] Anthony Zaknich. *Principles of adaptive filters and self-learning systems*. Springer, 2005.

- [32] L. Harnefors and H. P. Nee. A general algorithm for speed and position estimation of AC motors. *IEEE Trans. Ind. Electron.*, 47(1):77–83, Feb 2000.
- [33] H. Karimi, M. Karimi-Ghartemani, and M. R. Iravani. Estimation of frequency and its rate of change for applications in power systems. *IEEE Transactions on Power Delivery*, 19(2):472–480, April 2004.

# POROVNÁNÍ ŘADICÍCH ALGORITMŮ

*Vladimír Hlaváč*

ČVUT FS, hlavac@fs.cvut.cz

*Abstrakt: Práce obsahuje výsledky praktického testování několika řadicích algoritmů a možnosti porovnání jejich skutečné časové náročnosti s jejich asymptotickou složitostí. V první části je prezentováno porovnání náročnosti operace porovnání pro celá čísla a pro textové řetězce, které nejsou kódovány přímo v ASCII, ale v některé z rozšířených znakových sad.*

*Klíčová slova: Bubble sort, Selection sort, Insertion sort, Heap sort, Merge sort*

## 1. Metodika porovnávání

Pro porovnání bylo vybráno několik nejčastěji používaných řadicích algoritmů, jejichž chování je všeobecně známé.

### 1.1 Měření skutečné časové náročnosti

Protože se jedná o modelové řešení, jsou pochopitelně řazena čísla. Při praktickém použití algoritmů jsou nejčastěji řazeny záznamy (nebo objekty) podle zvoleného klíče, který obvykle spočívá v porovnání dvou řetězců. Při jakékoli zvolené metodě je tedy záměna pořadí dvou vyhledaných záznamů provedena záměnou patřičných ukazatelů, nebo při sestavování indexového souboru (například u databází) záměnou dvou čísel v poli indexů. Protože u rodiny x86 instrukce XCHG neumožňuje adresovat dvě adresy v paměti, jsou na záměnu vyžadovány minimálně tři instrukce procesoru (typicky  $2 \times \text{MOV}$  a  $1 \times \text{MOVSD}$ ). Přesto je tato časová náročnost zanedbatelná ve srovnání s porovnáním dvou klíčů. Pokud má řazení být obecné, je třeba volat funkci operačního systému pro porovnání ve zvolené znakové sadě (např. UTF-8). Časová náročnost takové operace je řádově větší. Z tohoto důvodu bývá jako měřítko náročnosti algoritmu měřen počet nutných porovnání, nikoli operací se záznamy v paměti.

Časová náročnost (skutečný čas běhu programu v sekundách) pro kódování UTF-8 nebyla v rámci testů zjištěna, v následující tabulce je porovnání časů, které jsou třeba k porovnání dvou celých čísel, dvou čísel v řádové čáře a dvou řetězců v sadě Windows CP1250, ovšem nikoli prostředky operačního systému, ale rychlou knihovnou pro textové funkce. Za tímto účelem byl připraven prázdný cyklus (s ohledem na optimalizaci se musí stále uvnitř cyklu provádět funkce, které kompilátor nemůže zjednodušit), který byl následně doplněn o požadovanou funkci. Všechna data jsou získána jako průměr z deseti pokusů:

Tab. 1. Výpočet skutečné časové náročnosti některých operací

	čas [s]	opakování	čistý čas na provedení funkce [ns]	poměrná hodnota
prázdný cyklus (hodnota je v dalších řádcích odečítána)	1.131	100000000	(11.31)	---
celočíselné porovnání	1.291	100000000	1.596	1
reálná čísla	1.312	100000000	1.814	1.137
text v kódování CP1250	12.97	100000	129710	81272

V tab. 1 je v prvním sloupečku skutečný čas pokusu, v druhém počet opakování v rámci pokusu, ve třetím pak podíl těchto hodnot. Jednojádrový Celeron na 1.8 GHz. Čas měřen prostředky operačního systému. Porovnání dvou řetězců je v tomto případě  $81272 \times$  pomalejší, než porovnání dvou celých čísel.

## 1.2 Testovací data

Pro testovací účely byla přes výše uvedené používána náhodná čísla. Funkce porovnání dvou čísel byla oddělena a doplněna o počítání počtu volání funkce. Pro porovnání s asymptotickou složitostí byla generována pole o počtu 16, 256, 4096 a 65536 prvků.

Protože některé metody pracují hůře s daty, která nejsou náhodná, byla jako kontrolní vzorek použita data, vzniklá načtením souboru „explorer.exe“ z adresáře operačního systému tak, že byl popsán jako pole neznaménkovaných celých 16bitových čísel. Pro generování více testovacích sad byl soubor čten vždy z jiného místa. Pro jedno konkrétní načtení byly pro vytvoření pole pomocí MS Excel zjištěny četnosti získaných hodnot. Zatímco hodnota „0“ se v rámci pole 65536 hodnot vyskytla  $3255 \times$ , hodnota „65535“  $1205 \times$ , pak 54699 čísel se nevyskytlo ani jednou a 6011 čísel právě jednou. Na grafech je zjevné, že na výsledky testů nemá tato skutečnost žádný vliv.

## 2. Testované metody

Jednotlivé metody řazení jsou nejjednodušeji popsány na webových stránkách Algoritmy.net [1] (česky). Rozbor pokročilejších metod lze najít v publikacích [2] a [3]. Kniha [2] je stále dohledatelná na internetu, původně byla volně šiřitelná, citace odpovídá tištěné verzi.

### 2.1 Bubble sort

Bublíkové řazení. Byla vybrána základní varianta, kdy rozsah vnitřního cyklu se opakovaně zmenšuje o jedničku.

### 2.2 Selection sort

Řazení výběrem. Z neseřazeného zbytku pole je opakovaně vybírán nejmenší prvek, který je pak zaměněn s prvním prvkem neseřazeného seznamu – seřazená část se tím zvětší o nově zařazený prvek.

### 2.3 Insert sort

Základní verze algoritmu řazení vkládáním vezme první z neseřazené části, nalezne pro něj místo mezi již seřazenými tím, že jej porovná postupně od začátku, až najde první větší. Ten a celý zbytek seřazené oblasti posune a na uvolněné místo vloží řazený prvek. Metodě by prospělo přepsání do strojového kódu, protože většina procesorů (resp. jak ARM, tak Intel) mají pro takový přesun instrukci, která optimálním způsobem spolupracuje s pamětí.

### 2.4 Insert sort s vkládáním půlením intervalu

Jedna z mnoha variant řazení vkládáním využívá faktu, že část dat je již seřazena a správné místo na vložení lze najít rychleji. Hledání správného místa tak má místo náročnosti  $O(n)$  jen  $O(\log n)$ , takže výsledná metoda patří do kategorie  $O(n \log n)$ . Nevýhodou je, že pokud porovnávaný prvek je stejný, jako vkládaný, vložíme jej na toto místo. Ve výsledku se tím ale může poškodit předchozí seřazení podle jiného klíče (říkáme, že metoda není stabilní).



Metodu lze upravit tak, že nalezne konec úseku, kde jsou stejné prvky, a nově řazený vloží tam. Pak již ale nelze zaručit uvedenou časovou náročnost (metoda se vrací do kategorie  $O(n^2)$ ).

## 2.5 Quick sort

Před deseti až dvaceti roky nejoblíbenější metoda řazení. Z pole vezmeme jeden prvek (označuje se „pivot“) a pak začneme pole prohledávat z obou konců. Zleva hledáme první, který je větší. Zprava první, který je menší. Tyto dva pak vyměníme. Použitý prvek pak umístíme na místo, kde se oba výběry sejdou. Tím máme dvě části pole, první, kde jsou všechny prvky menší než pivot a druhé, kde jsou větší. Speciální případ je pak již zařazený prvek (pivot). Pro to, aby se jednalo o metodu „rozděl a panuj“, „divide-and-conquer“, která přenáší rychlost složky algoritmu z  $O(n)$  na  $O(\log n)$  stačí, aby rozdělené části byly v nějakém jiném poměru, než 1:(n-1), postačí i 1:10. Z toho také vyplývá neschopnost této metody seřadit již seřazený seznam, kdy má náročnost  $O(n^2)$ .

Aby tato časová náročnost byla alespoň nepravděpodobná, používá se většinou randomizovaná (znáhodněná) verze algoritmu, kdy počáteční prvek se z pole vybírá náhodně. Algoritmus pak řadí velmi rychle téměř vždy, ale protože problematický případ s opakovanou volbou nejmenšího či největšího prvku nelze zcela vyloučit, formálně stále spadá do kategorie  $O(n^2)$ .

Protože všechny algoritmy byly realizovány v Pascalu, byla pro toto testování zvolena varianta metody, se kterou přišla firma Borland inc. (v grafech je tato metoda také popsána jménem této firmy). Inovace spočívá v tom, že jako rozhodující prvek pro řazení (pivot) se vybere ten, který je uprostřed pole. Pokud se pak při hledání z jedné strany (první větší než pivot zleva, první menší než pivot zprava) nenajde požadovaný prvek, ale z druhé strany ano, pak se jediný nalezený zamění s pivotem. V tom případě se ale musí vyhledávání z této strany opakovat, takže výsledná náročnost algoritmu je formálně stále  $O(n^2)$ . Druhá nevýhoda této varianty je, že není stabilní (přesunem pivotu může být porušeno pořadí prvků se stejným klíčem). Mimochodem, popsany jeden cyklus této metody představuje nejrychlejší možnost nalezení mediánu.

## 2.6 Heap sort

Klíčovým prvkem je zde hromada (na [1] označovaná jako halda), sloužící jako prioritní fronta. Hromada je zvláštní druh stromu, kde (protože řadíme od nejmenšího) má rodič vždy menší hodnotu než jeho dva potomci. Hromada se zapisuje do pole tím způsobem, že pokud je rodič na adrese  $i$ , pak jeho potomci jsou na adresách  $2i$  a  $2i+1$ . Prvky na hromadu přidáváme tak, že nový přidáme na konec pole, a pak jej porovnáme s rodičem (na adrese např.  $j \div 2$ ); pokud je menší, tyto dva prohodíme a operaci opakujeme.

Seřazené pole pak získáme postupným odebíráním z hromady. Základní varianta algoritmu říká, že po odebrání vezmeme poslední prvek, umístíme ho místo chybějícího (tím se hromada zmenšila o jednu) a hromadu opravíme [1]. Rychlejší metoda spočívá v posunu hromady tak, že místo chybějícího vezmeme menšího z potomků, a akci opakujeme, dokud se nedostaneme na konec hromady. Teprve zde poslední prvek hromady dáme na poslední uvolněné místo. Formálně je třeba i pak hromadu opravit, ale většinou dojde jen k porovnání, případně k minimálnímu počtu opravných prohození.

Výhodou Heap sortu je, že po vybudování hromady máme vždy nejmenší prvek k dispozici okamžitě. Tam, kde jde o nějakou formu paralelních algoritmů, nemusí druhý algoritmus čekat na uspořádání hromady, to může proběhnout na pořadí. Prosté vybrání prvního prvku je pak v kategorii  $O(1)$ .

## 2.7 Merge sort

Řazení slučováním. Protože u Quick sortu byl problém, že pole se nám nemuselo dělit rovnoměrně, je zde postup otočen. Pole se rozdělí na jednotlivé prvky a ty se opakovaně slučují. V průběhu algoritmu slučování probíhá tak, že máme dva ukazatele (do obou slučovaných polí), tyto prvky porovnáme a menší zapisujeme do výsledného pole. Důsledkem tedy je, že musíme mít druhé stejně velké pole, kam zapisujeme mezivýsledky. Obě pole můžeme samozřejmě střídat, takže prostorová náročnost je  $2n$ . To je důvod, proč v dobách, kdy paměť byla drahá, nebyl tento algoritmus populární. Z druhé strany je stabilní a vykoná maximálně  $O(n \cdot \log_2 n)$  porovnání, což je i nejmenší možný počet porovnání u dat, o kterých nemáme předběžné informace (pokud bychom například věděli, že data jsou v nějakém rozsahu a nějak rozložena, můžeme pro zrychlení použít Bucket sort, česky např. příhrádkové řazení).

### 3. Výsledky

Program je napsán tak, že zvolenou metodu spustí  $16\times$ ,  $256\times$ ,  $4096\times$  a  $65536\times$ , a změřené počty porovnání zapíše do okna typu Memo, které může mít více řádek. Bylo doplněno tlačítko, které spustí každou metodu  $5\times$ , takže program může pracovat samostatně a data lze pak snadno zpracovat do grafu. Data pro jednotlivou metodu vypadají například následovně:

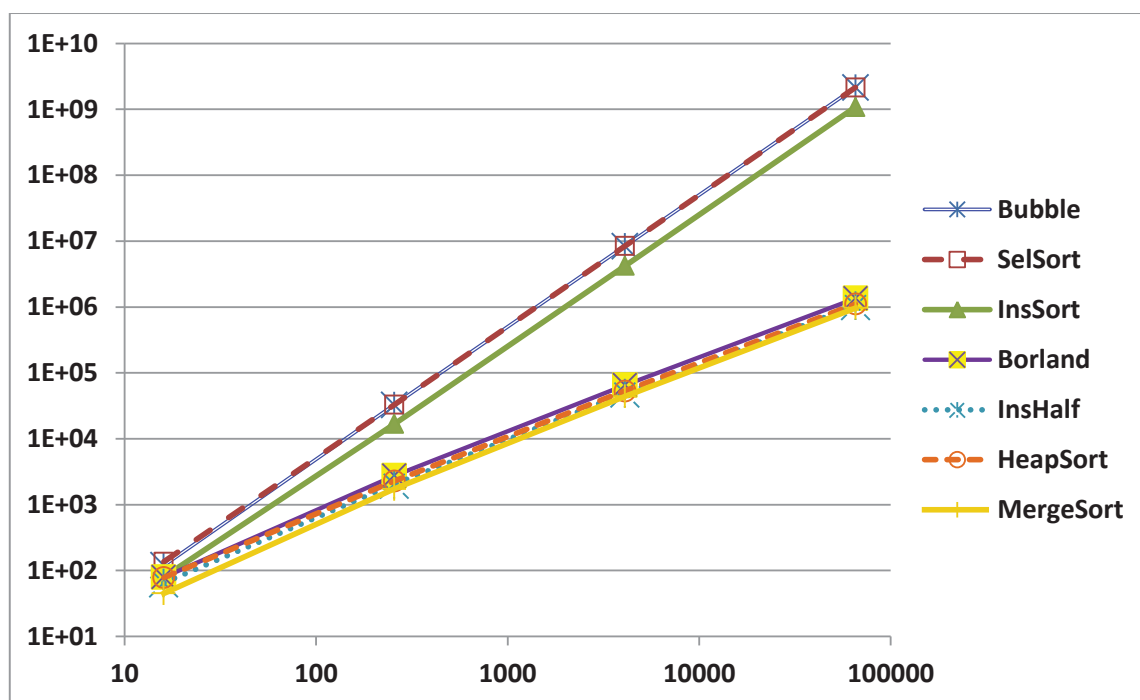
Tab. 2. Ukázka vygenerovaných dat pro jednu metodu. Umístění v řádcích je zavádějící, data v jednom řádku nemají společného nic jiného, než že vznikla při stejném spuštění.

Borland	69	2687	65320	1393786
Borland	78	2686	67526	1372598
Borland	76	2509	62157	1362484
Borland	90	3136	64287	1430948
Borland	82	2742	63409	1437918

Pro porovnání byla dávka spuštěna dvakrát pro náhodná data a dvakrát pro data získaná načtením souboru „explorer.exe“. Výsledné hodnoty byly zprůměrovány (deset od jedné metody) a vyneseny do grafu. Zde je třeba zdůraznit, že časová náročnost některých metod (tak, jak byly implementovány) nezávisí na datech, a místo průměru tak mohl být vzat první řádek. Jedná se o Bubble sort, Selection sort a variantu Insert sortu s vyhledáváním půlením intervalu.

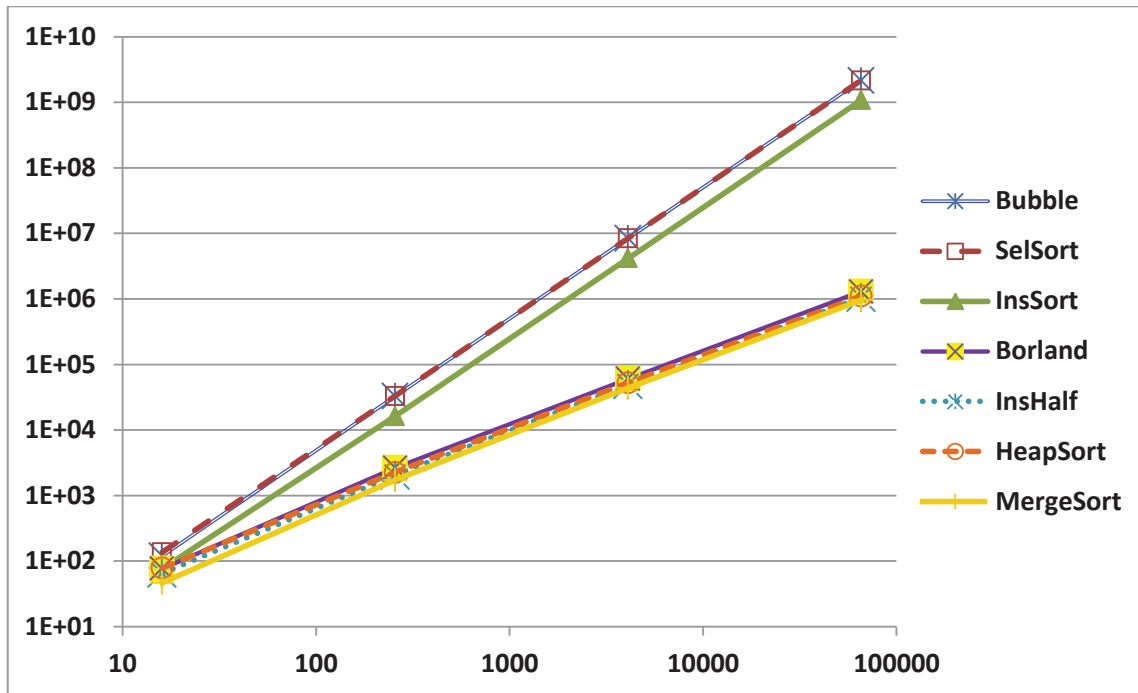
#### 3.1 Souhrnný graf

Pro zobrazení v grafu byl použit graf se dvěma logaritmickými souřadnicemi, na vodorovné ose je počet prvků seřazovaného pole, na svislé je počet porovnání. Jsou použity dekadické logaritmy, logaritmus o základu 2 by umístil počty prvků i počty porovnání přehledněji.



Obr. 1. Odshora: Bublínkové řazení, řazení výběrem, řazení vkládáním, varianta Quick-sortu dodávaná s Turbo Pascalem firmy Borland, řazení vkládáním s půlením intervalu, řazení hromadou a slučování.

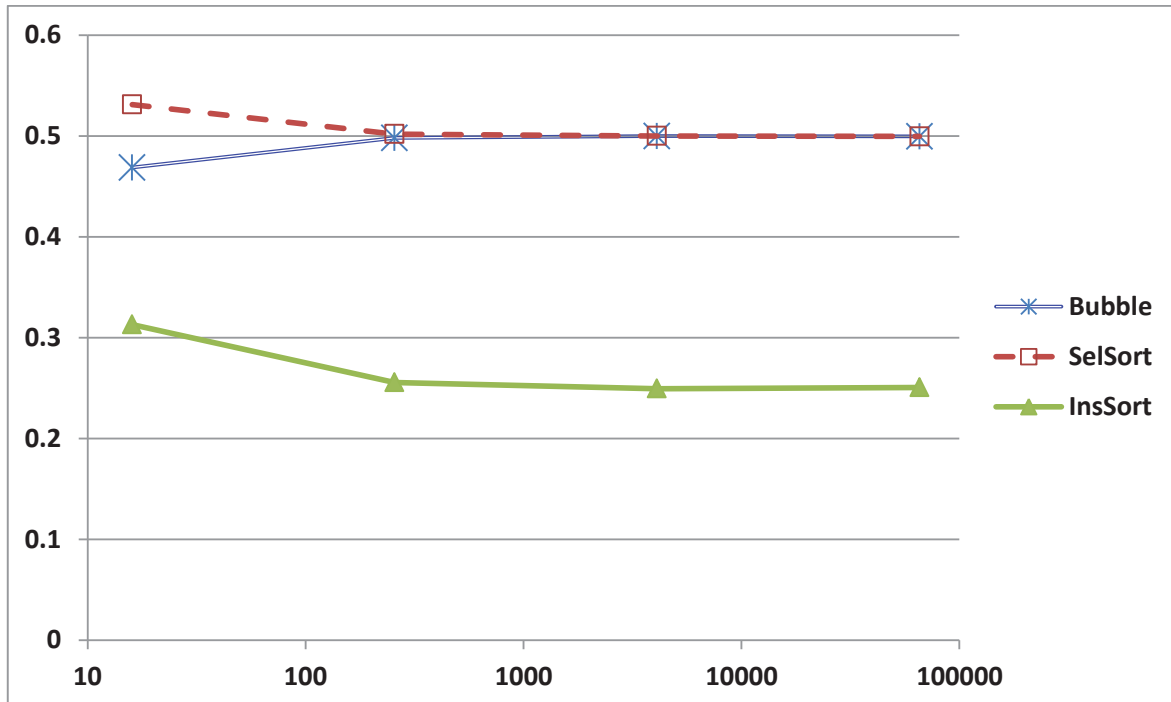
Tentýž graf pro druhý soubor, kde data nebyla náhodně rozdělena (četnosti), nicméně byla na začátku do značné míry náhodně seřazena.



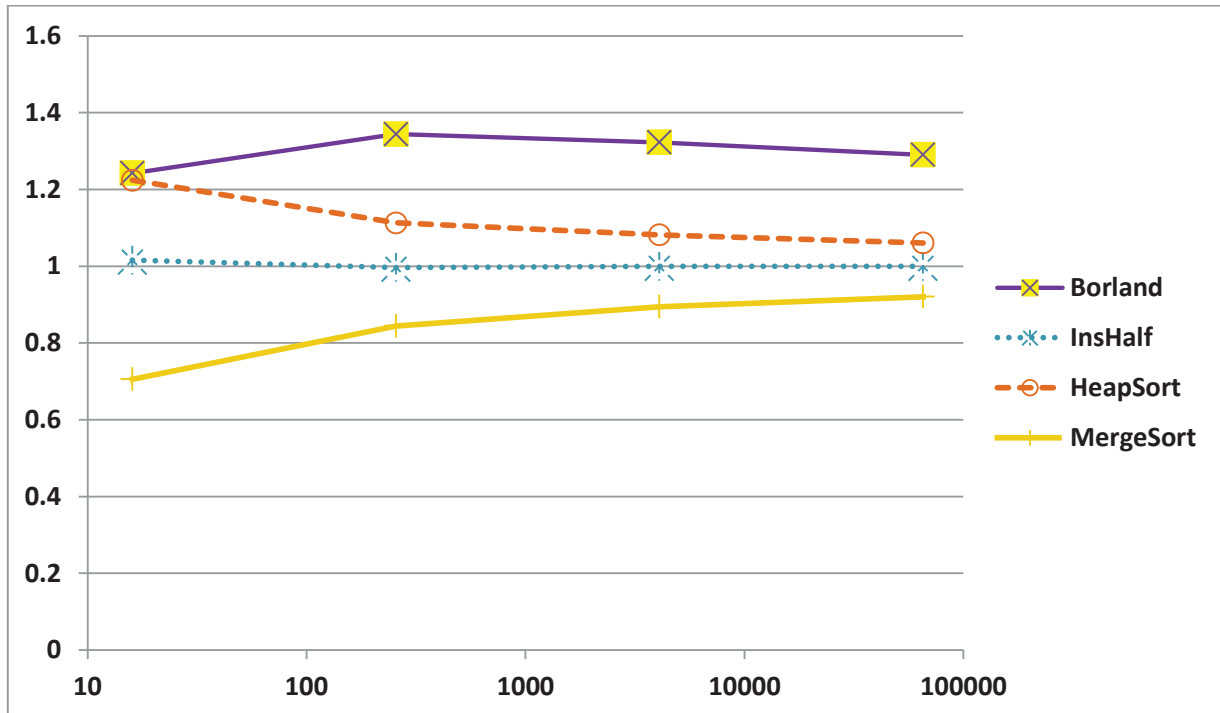
Obr. 2 Grafy pro data s jiným rozložením hodnot vypadají naprosto stejně. Jiné hodnoty by bylo možné získat, kdyby data byla předem v různé míře seřazena.

### 3.2 Graf pro porovnání s teoretickou náročností

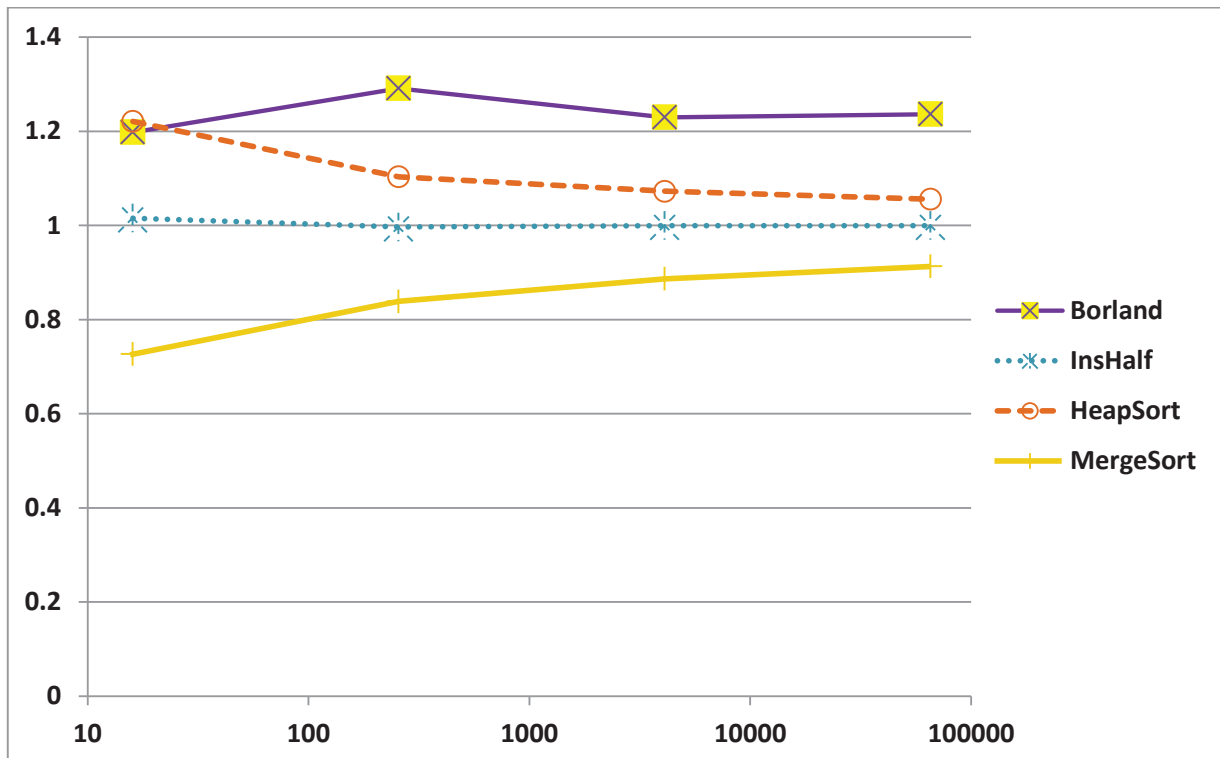
Přehlednější zobrazení výsledků bylo dosaženo tím, že jsou vyneseny jen zvolené grafy, a jejich skutečná časová náročnost je podělena jejich asymptotickou náročností (číslem  $n^2$ , resp.  $n \cdot \log_2 n$ ). Protože jde o očekávanou (nejpravděpodobnější) hodnotu, je Quick sort zařazen mezi algoritmy kategorie  $O(n \cdot \log n)$ .



Obr. 3 Hodnoty náročnosti po podělení hodnotou  $n^2$ . U prvních dvou je výsledná hodnota přibližně jedna polovina, protože oba algoritmy procházejí oblast o délce postupně od  $n$  do 1. U řazení vkládáním je nejen průměrná délka procházené části již seřazených dat poloviční, ale i samotná seřazená část má na začátku nulovou délku, takže ve výsledku se časová náročnost blíží jedné čtvrtině.



Obr. 4 Quick sort (první graf) do uvedené kategorie vlastně ani nepatří, takže i když byl očekáván lepší výsledek, jedná se o úspěch. Řazení hromadou je obecně pomalejší, protože nejprve je třeba složit hromadu, a všechno se tak dělá vlastně dvakrát. Modifikace řazení vkládáním má pevnou časovou délku, modifikací algoritmu by ji šlo dokonce zkrátit, ale je nutno připomenout, že metoda není stabilní. Řazení slučováním (Merge sort) šetří čas při dořazování zbytku druhého pole – pokud se při slučování první jedno z polí vyčerpá, pak se druhé kopíruje bez porovnání. Speciální případ je tedy první krok, kdy se porovnávají jednotlivé prvky, a na každé dva případně jen jedno porovnání. Odpovídá to tomu, že se jedná o dodnes nejrychlejší známou metodu řazení (bez požadavku na předběžnou znalost vlastností dat).



Obr. 5 Tytéž průběhy pro data získaná načtením souboru „explorer.exe“. Nerovnoměrnost dat má vliv jen na Quick sort. Kdyby bylo pořadí dat předem částečně seřazené, projevil by se vliv i na Merge sort, ale zde se nejedná o tento případ.

#### 4. Závěr

Při porovnání algoritmů kategorie  $n^2$  byla získána očekávaná data. Z těchto algoritmů má jistý smysl Bubble sort, protože jej lze napsat rychleji, než se dohledá knihovna řadicích algoritmů na internetu, a je také velmi spolehlivý. Při posuzování časové náročnosti programu je třeba sečíst celkovou dobu, kdy jej budou uživatelé používat (krát hodinová cena práce operátorů) a čas, který s psaním aplikace stráví programátor (krát jeho hodinová sazba). Pokud program nebude prakticky používán, je zbytečné jej optimalizovat.

Při porovnání výkonnějších algoritmů překvapila nízká výkonnost Quicksortu v porovnání s Heapsortem. Opět tu ale platí jednoduchost implementace, která má přímou vazbu na spolehlivost programu.

Pro vytvoření testovacích programů bylo použito prostředí Lazarus. Nebylo možné otestovat přímo UTF-8, který je svou proměnnou délkou znaků velmi problematický, protože tato znaková sada nebyla dostupná v kombinaci se starší verzí Windows (není přímá cesta k načtení dat). Nativní podporu mají textové soubory v UTF-8 v Lazarusu jen v případě, že je provozován v prostředí Linux.

#### Literatura

- [1] Neckář, Jan: Algoritmy.net. Příklady algoritmů v jazyce Java, Perl, Python, řešení složitých matematických úloh Webová stránka, online: <http://algoritmy.net>
- [2] Dasgupta, Sanjoy, Papadimitriou, Christos H., Vazirani, Umesh V.: Algorithms. McGraw-Hill Education – Europe, 2006.
- [3] Erickson, Jeff: Algorithms. 2015. Dostupné on-line: <http://www.cs.illinois.edu/~jeffe/teaching/algorithms/>