

## **Lara Laban Publication List**

telephone: +46 72-374 01 50

[lara.laban@control.lth.se](mailto:lara.laban@control.lth.se)

1. 7th International Conference on Electrical, Electronics and Computing Engineering (IcETRAN)

### **CLASSIFICATION OF CHEST X-RAY IMAGES USING DEEP CONVOLUTIONAL NEURAL NETWORKS**

Lara Laban, Radiša Jovanović, Mitra Vesović, Vladimir Zarić | 2020

2. 5th International Conference "Mechanical Engineering in 21st Century"

### **CLASSIFICATION OF COVID-CT IMAGES UTILIZING FOUR TYPES OF DEEP CONVOLUTIONAL NEURAL NETWORKS**

Lara Laban, Mitra Vesović | 2020

3. The First International Student Scientific Conference "Multidisciplinary Approach to Contemporary Research"

### **SELECTION OF THE MOST EFFICIENT TEMPERATURE MEASUREMENT METHOD BASED ON DESIRED RESPONSE TIME AND EXPERIMENTAL RESULTS**

Lara Laban, Nedzad Rudonja, Milan Gojak | 2017

4. International Multidisciplinary Conference "Challenges of Contemporary Higher Education CCHE 2024"

### **POWER QUALITY IMPROVEMENT OF LED STREET LIGHTNING REFLECTORS**

Željko V. Despotović, Aleksandra Grujić, Ivana Vlajić Naumovska, Predrag Simić, Lara Laban | 2024

5. 7th International Conference on Electrical, Electronics and Computing Engineering (IcETRAN)

### **MODELLING AND CONTROL OF A SERIES DIRECT CURRENT (DC) MACHINES USING FEEDBACK LINEARIZATION APPROACH**

Mitra Vesović, Radiša Jovanović, Lara Laban, Vladimir Zarić | 2020

6. X International Conference "Heavy Machinery-HM 2021"

### **TRAJECTORY TRACKING OF A TWO – LINK GRIPPING MECHANISM**

Radiša Jovanović, Uglješa Bugarić, Lara Laban, Mitra Vesović | 2021

7. X International Conference "Heavy Machinery-HM 2021"

### **FEEDBACK LINEARIZATION CONTROL OF A TWO – LINK GRIPPING MECHANISM**

Mitra Vesović, Radiša Jovanović, Lara Laban, Uglješa Bugarić | 2021

8. 5th International Conference "Mechanical Engineering in 21st Century"

### **MODELING AND CONTROL OF A LIQUID LEVEL SYSTEM BASED ON THE TAKAGI-SUGENO FUZZY MODEL USING THE WHALE OPTIMIZATION ALGORITHM**

Radiša Jovanović, Vladimir Zarić, Mitra Vesović and Lara Laban | 2020

# Classification of Chest X-Ray Images Using Deep Convolutional Neural Networks

Lara Laban, Radiša Jovanović, Mitra Vesović and Vladimir Zarić

**Abstract**—In this paper a method is proposed for the classification of a chest X-ray (*normal, pneumonia-consisting of bacterial and viral*) image data set. A deep convolutional neural network with an architecture resembling the VGGNet is presented using dropout, decay and data scaling. Since the dataset had a class imbalance, this was solved using a simple method called data scaling. The training of the neural network was done using small batches with a binary cross entropy loss function. The same neural network was then implemented adding batch normalization layers, and comparisons were made. Furthermore, the chest X-ray dataset was also trained using transfer learning with a pre-trained neural network VGG16 on the ImageNet dataset. Later on juxtapositions were made on using both techniques. Additionally, in applying these method we were able to achieve a satisfying classification for the training and validation datasets, save computational time and have a higher classification accuracy.

**Index Terms**—convolutional neural networks; deep learning; transfer learning; batch normalization; chest X-ray dataset; image classification; dropout.

## I. INTRODUCTION

Convolutional neural networks (CNNs) are a subset of deep neural networks, which are used for classifying images. The main idea is to take a set of images correctly labeled as the input data and used them to train our neural network so as to achieve an output with an appropriate categorization. The inspiration for CNNs comes from the observation of the animal visual cortex. Conversely, the flourishing of these networks only came recently due to the increase of computational power and the development of many possible libraries that could be used to battle complex mathematically based problems, such as back propagation. The first paper that introduced the convolutional neural networks as we have come to know them today has [1] demonstrated that a model which consists of a multilayered network can be successfully used for recognition of stimulus patterns according to the

differences in their shapes. However, there is some debate that the true beginning was when a paper in 1990 [2] demonstrated that a CNN model which aggregates simpler features into progressively more complicated features can be successfully used for handwritten character recognition. In 2012 the ImageNet Large Scale Visual Recognition Challenge [3], at that moment consisting of 1000 categories and 1.2 million images received a submission that would propel the CNNs development once again. AlexNet [4] achieved a top-5 error of 15.3%, which at the moment surpassed by an astonishing 10% all of the other submissions, and had a much faster training time as it was implemented on a GPU. The following year, the same challenge, now with a larger dataset was won by ZFNet [5]. It had the top-5 error of 14.8%, however even more so important is that it was able to reduce the first layer filter size from  $11 \times 11$  to  $7 \times 7$  and had a stride of 2, rather than 4 in the pooling layer.

VGG16 is a convolutional neural network model proposed in the paper [6]. This model achieved 92.7% top-5 test accuracy. The main contribution of this model was that it used  $3 \times 3$  kernel sized filters, instead of the  $7 \times 7$ . It was trained for weeks using GPUs, and had a huge computational cost. However, it introduced a new idea using the same kernels throughout the entire architecture, this aided in generalization for classification problems outside of what they were originally trained on. If for a second we go back to LeNet [7] that was the foundation for all of these previously mentioned CNNs we can observe the main sequence of three layers convolution, pooling and non-linearity still play the key part, and sometimes it is beneficial not to import to many layers when training a smaller dataset [8]. Finally, in recent years transfer learning [9], which addresses crossdomain learning problems by extracting useful information from data in a related domain and transferring them for being used in target tasks, has been demonstrating a significant impact.

Pneumonia is one of the main causes of death amongst children, it was stated that 19% of all deaths of kids aged 5 years and less is connected with a viral or bacterial pneumonia [10]. Today, pneumonia is the single leading cause of mortality in young children according to the World Health Organization (WHO). An even scarier report made by WHO says that 95% of new-onset childhood clinical pneumonia occurs in developing countries, many of them located in Africa and South Asia [11].

The lungs of humans are made up of small sacs called alveoli, which fill with air when a healthy person breathes, in turn when a person with pneumonia breathes the alveoli which are filled in this case with pus and fluid, blocking the

Lara Laban is with the Faculty of Mechanical Engineering, University of Belgrade, Kraljice Marije 16, 11120 Belgrade, Serbia (e-mail: llaban@mas.bg.ac.rs)

Radiša Jovanović is with the Faculty of Mechanical Engineering, University of Belgrade, Kraljice Marije 16, 11120 Belgrade, Serbia (e-mail: rjovanovic@mas.bg.ac.rs)

Mitra Vesović is with the Faculty of Mechanical Engineering, University of Belgrade, Kraljice Marije 16, 11120 Belgrade, Serbia (e-mail: mvesovic@mas.bg.ac.rs)

Vladimir Zarić is with the Faculty of Mechanical Engineering, University of Belgrade, Kraljice Marije 16, 11120 Belgrade, Serbia (e-mail: vzaric@mas.bg.ac.rs)

oxygen from arriving to the lungs, limit the capacity to intake oxygen. One of the main ways to have a proper diagnosis is radiographic data. X-rays can help in distinguishing between different types of pneumonia. However, since rapid interpretation of images is sometimes very hard, especially in developing countries, new methods are always sought after and proposed.

For that reason, an idea to battle this kind of a problem was proposed in a brilliant paper [12] published in 2018 which suggested image-based deep learning to identify various medical diagnoses, including the chest X-ray images from children. Using convolutional neural networks, transfer learning to be precise, they achieved an accuracy of 92.8% in distinguishing between normal and pneumonia images, over the course of 100 epochs.

This paper is organised in the following manner: section 2 represents a description of a dataset which is used in the training and validation of the proposed neural network. In section 3 the main methods which are used are explained in detail, as well as the architecture of the CNN. As a result, in section 4 we discuss the results and compare the methods, based on accuracy and loss functions. In section 5, following a short summary a conclusion is made and future work and possible directions are stated.

## II. DATASET AND ITS IMPLEMENTATION

The dataset which is used in this paper consists of 5856 chest X-Ray images from children [13], including 4392 pneumonia ray (*bacterial and viral*) and 1464 normal. Being than the dataset consist of a couple of thousand pictures, there is no need to take an approach of data augmentation, where we increase the diversity of data by altering the original samples using translation, rotation, shearing, flips and adding them to the training set. However, we observe that the pneumonia part of the dataset is much larger than the part of the normal set, almost 4 times as big, resulting in a class imbalance. One way to correct this, so that our neural network may learn appropriately and not pick the pneumonia label naturally is to scale the data. This can be done by computing a weight for each class during the training, resulting in an array [1,3], and as an outcome amplifying the loss by a larger weight when we approach normal data. In this example treating an instance of normal as 3 instance of pneumonia, aids in this disportion.

During the preprocessing of images we resized all the images to a fixed size  $64 \times 64$ , and in doing so we also maintained the aspect ration. The reasoning behind this being that all the images in a dataset need to have a fixed feature vector size. This means all the images will have identical widths and heights, making it easier to quickly load and preprocess a dataset and briskly move through our convolutional neural network. The aspect ratio will enable us to resize the images along the shorter dimension, be it width or height, and in cropping it, will maintain the ratio. It is important to note that this step is not necessary if you are not working with a difficult dataset. Notwithstanding its benefits,

it was implemented in this particular dataset.

### A. ImageNet dataset

ImageNet is a dataset consisting of over 14 million images, which belong to one thousand classes. It was used as the dataset in the highly respected convolutional neural network model VGG16 which was proposed by Oxford scientists. In this paper the VGG16 network was used as a pre-trained convolutional neural network, in order to incorporate transfer learning and compare it to the original paper, mentioned beforehand, as well as the architecture that we propose.

## III. METHODS DESCRIPTION

In order to try and reduce overfitting and increase our classification accuracy on the chest X-ray dataset we endeavor in performing two types of neural network training techniques:

- dropout and decay (with and without batch normalization),
- transfer learning (neural networks as feature extractors)

The first technique that is used in order to improve the generalization error in the convolutional neural network is dropout [14]. Dropout is nothing more than a form of regularization, which succors us in controlling the model capacity. Furthermore, it increases validation accuracy, sometimes at the expense of the training accuracy in order to help avert overfitting. Simply put, some number of layers outputs are dropped, making the current layer have a different number of nodes, as opposed to the previous layer. This helps our model to observe multiple nodes and activate them when similar inputs are provided, therefore, directly aiding the generalization problem. The dropout layers are arranged in the network in such a manner that we have randomly disconnected nodes by a probability of 0.25 in the first few layers; and with a double increase in probability in the last layer. The reason for this is that if the first layers are dropped by a higher probability, then that will later affect the training. The dropout is implemented after the pooling layer, and before the next convolutional layer (or last flatten and dense layers). Decay that is used in this neural network is a standard decay that can be obtained using the Keras library in Python. Since the learning rate  $\alpha$  controls the step that is made along the gradient, larger steps are usually used in the beginning to make sure that we do not stagnate in the local optima, while smaller steps are used deeper in the network and near the end of the convolution in order to converge to a global minimum. We have initialized the learning rate to be 0.05, and applied the following formula to adjust it after each epoch,

$$\alpha_{i+1} = \frac{\alpha_i}{1 + k \cdot i} \quad (1)$$

where  $\alpha$  is the current learning rate,  $i$  is the epoch and  $k$  is the decay calculated as the division between the learning rate and the number of epochs. This type of adjustment of the learning rate each epoch, can increase accuracy, as well as reduce the loss function and the time necessary to train a network. Batch normalization [15] is used to normalize the

activations of a given layer's inputs by applying mean and standard deviation before passing it onto the next layer. In addition, the covariate shift refers to a change in the distribution of the input variables which are present in the training and validation data. Since it has been proven that the training of the neural network is the most coherent when the inputs to each layer are alike, the main intention is that even when the explicit values of inputs layers to hidden layers change, their mean and standard deviation will still remain relatively the same, thus reducing the covariate shift. Batch normalization has demonstrated an immensely effective approach to reducing the number of epochs necessary for training by allowing each layer to learn independently. Here the idea that differs from the original paper and is first proposed in [16] states that the batch normalization should be implemented after the activation layer. The main reasoning behind this is that we want to avoid setting the negative values coming out of the convolution layer to zero. Instead we pass them through the batch normalization layer, right after the activation (ReLU) layer, and assure that some of the features that otherwise would not have made it do. This yields a higher accuracy and lower loss, and is to this day a debate amongst the creators of Keras.

Finally, the second technique is [17] transfer learning, a machine learning technique where networks can behave as feature extractors. Transfer learning is nothing more than the ability to use a pre-trained model to learn patterns from data, on which the original network was not trained on. As previously stated deep neural networks trained on a large scale dataset ImageNet have demonstrated to be superb at this task.

When treating networks as feature extractors we choose a point, in this case before the fully connected layer and remove it. Subsequently, in this particular example while using the VGGNet pre-trained on the ImageNet we removed the fully connected layer and stopped at the last pooling layer where the output shape is  $7 \times 7 \times 512$ , 512 filters with the size  $7 \times 7$ . Now, our feature vector has  $7 \times 7 \times 512 = 25088$  values and it will be used to quantify the contents of the images, which were not included in the original training process. The format which allows us to extract these features is the hierarchical data format version 5 (hdf5), which is used to store and organize large amount of data.

Transfer learning is an optimization, which has been proven to yield a better performance and drastically save time. This is precisely why we used it in this paper, to see if we could obtain a higher classification, and perform faster. Transfer learning relaxes the hypothesis that the training data must be independent and identically distributed with the test data, which we clearly stated as a must in the beginning of this chapter. Moreover, transfer learning is able to solve the problem of insufficient training data. Furthermore, there is the option to remove the fully connected layers of the existing network in order to add a new fully connected layer to the CNN and fine tune the weights to recognize object classes. However, here it was not implemented since treating networks as arbitrary feature extractors was enough.

In the following sections we will demonstrate the architecture of a CNN that is based on VGGNet, its implementation with and without batch normalization, and additionally transfer learning will be presented instead of the CNN that was previously explained.

#### A. Convolutional Neural Network architecture

Into the bargain all that was explained, we picked the following CNN architecture shown in Fig. 1. It is consisted of multiple convolutional and pooling layers, as well as the fully connected layers. The first two convolutional layers learn 32 filter each with a size  $3 \times 3$ .

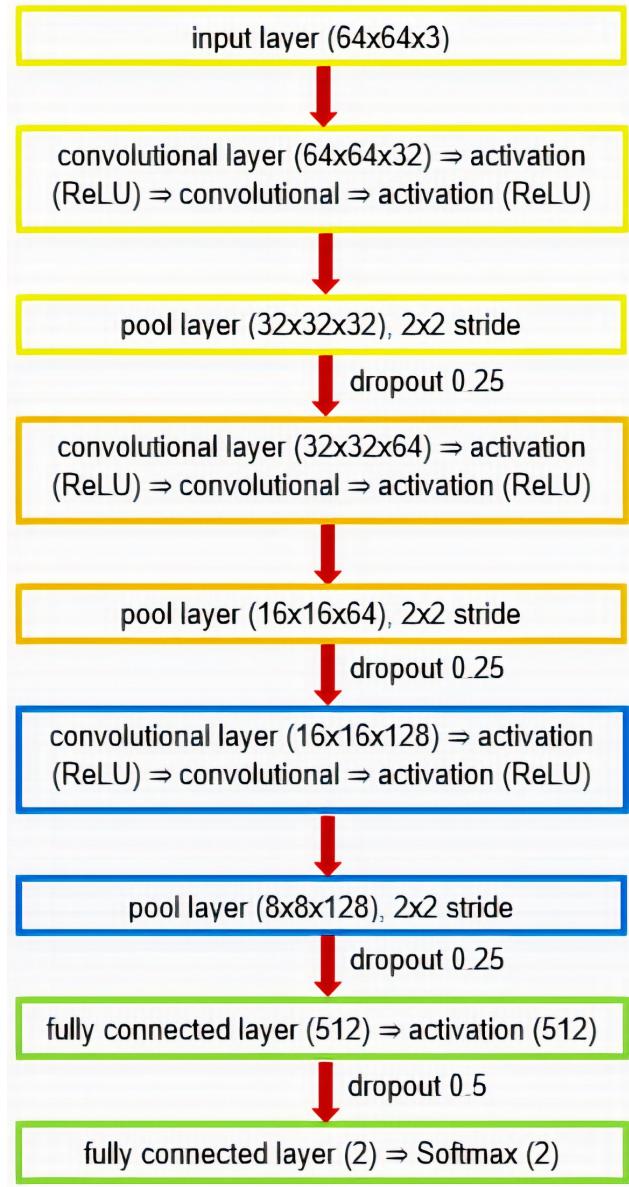


Fig. 1. A schematic of the convolutional neural network without batch normalization, that resembles the VGGNet. All of the convolutional layers that precede the fully connected layers have filters 32,64,128 that are the same size  $3 \times 3$ . The probability distribution is applied in the last layer using Softmax and the output yields two class labels normal and pneumonia.

Sequentially, the fourth and the fifth layers learn 64 filters with the size  $3 \times 3$  and the last two learn 128 filters with the size  $3 \times 3$ . The pool layer is used to reduce the computational load and the number of parameters, thus reducing the risk of overfitting. We used a max pooling layer with a pool size  $2 \times 2$  and a stride 2. Finally, we have the fully connected layer which consists of 8192 parameters, input values which learn 512 nodes. The activation layers which were used are Rectified Linear Unit (ReLU) defined as,

$$f(x) = \max(0, x) \quad (2)$$

where  $x$  is the input into the neuron.

Softmax or the normalized exponential function assigns normalized class probabilities for each prediction, and is represented by,

$$S(y_i) = \frac{e^{y_i}}{\sum_{j=1}^k e^{y_j}} \quad (3)$$

for  $i = 1, \dots, k$  and  $\mathbf{z} = (z_1, \dots, z_k) \in \mathbb{R}^k$ .

Softmax takes an input vector and normalizes it into a probability distribution between  $[0,1]$ . Therefore the sum of all output values is equal to 1, which in turn makes the training converge more quickly. In order to achieve this, before training we must include one hot encoding in order to convert the labels from integers to vectors.

In addition, later when we want to add the batch normalization layer, we can apply it after each activation layer, as discussed previously.

#### B. Implementation and training of a much simpler version of the VGGNet

Taking into the bargain all that was explained before, the implementation of this CNN was done by using the Python programming language.

We used Keras [18] which is mainly used for implementing of activation functions, optimizers, convolutional and pooling layers, and is actually able to do backpropagation automatically. Further, we used TensorFlow [19] an open-source library developed by the Google Brain team, with the main purpose to help combat convoluted numerical computations and large-scale machine and deep learning. The main advantage of TensorFlow is its compatibility with Scikit-Learn and Keras library.

Right after we load and preprocess our images dataset it is necessary to use one hot encoding. This is done by using a part of the Sklearn library LabelBinarizer. However beforehand we must split the training data and the validation data, here we opted to split it 75% and 25%, sequentially. The next step is the implementation of an optimizer, here we used the stochastic gradient descent (SGD) optimizer. The SGD optimizer was set to a learning rate of  $\alpha = 0.05$ , with a decay in order to slowly reduce the learning rate over time and converge to the global solution more efficiently. Decaying the learning rate is beneficial in reducing overfitting and obtaining a higher classification accuracy. The smaller the

learning rates are, the smaller the weight update will be enabling us to converge. The gradient descent method is an iterative optimization algorithm that operates over an optimization surface. It is a simple modification to the standard algorithm of gradient descent. The main purpose of SGD is to calculate the gradient and adjust the weights of the training data (but not on the whole dataset, but rather on a mini batch). The mini batch method is a blend of the SGD and batch methods were the neural network selects a part of the training data and updates the weights, but trains the network with the average weight update. Usually the smallest standard batch size which is used is 32, however we opted to use 24, as it complemented our data. The reasoning behind this is that present research confirms that using small batch sizes achieves the best training stability and generalization performance, for a given computational cost, across a wide range of experiments. The loss function which was used is the binary\_crossentropy function. This was done because we only had two classes, if there were more we would have had to use categorical\_crossentropy, but have in mind we could have used categorical as well, but studies show that binary is much more efficient in this case.

The training was done on 30 epochs since it was enough to achieve satisfying results. After the training we implemented a method that takes the weights and the state of the optimizer and serializes them to the disc in a hdf5 format, in order to load them and test the labeling.

#### C. Implementation using transfer learning

The first step in this process is to extract features from VGG16, in doing so we are forward propagating the images until a given layer, and then taking those activations and treating them as feature vectors. Here the main two differences are that we used the standard a batch size of 32 and the training and test split is done at the same time as training, we again split it into 75% training data and 25% test data. Once the extraction of the features was done, we trained the classifier on those features. We also implement the GridSearchCV class to assist us to turn the parameters to the LogisticRegression classifier.

The final results are presented in the following chapter, comparisons are made and a visual representation of the graphs is shown using Matplotlib in order to estimate if there is overfitting.

## IV. RESULTS AND COMPARISONS

The results of the CNN without batch normalization are presented in Table 1. We clearly see that our neural network has classification accuracy of 95%. In the following tables we use the term precision which represents true positive divided by a sum of true positive and false positive; recall which represents true positive divided by a sum of true positive and false negative. Therefore, precision is good to determine when the cost of false positives is high, on the other hand recall tells us the number of correctly labeled data. Ultimately, we have the f1-score used to find the weighted average of recall and precision.

TABLE I  
CNN WITHOUT BATCH NORMALIZATION: RESULTS

	precision	recall	f1-score
normal	0.93	0.91	0.92
pneumonia	0.97	0.97	0.97
accuracy			0.96
macro avg	0.95	0.94	0.95
weighted avg	0.96	0.96	0.96

In analyzing the curves shown in Fig. 2 we see that our network learned until the 30 epoch, beyond that was simply not necessary since we already had excellent results. We can also observe that our loss and accuracy curves both almost match for training and validation, with slight deviations.

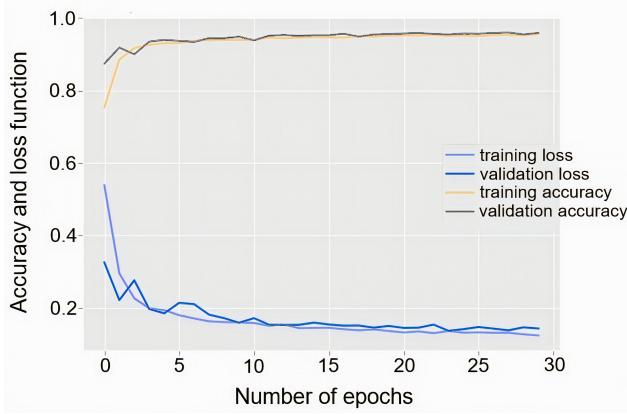


Fig. 2. A graph depicting a convolutional neural network without batch normalization, that resembles the VGGNet – training and validation loss and accuracy curves

In Fig. 3 we can see how the labeling looks, when we use the trained and saved model to label the data with this obtained accuracy.

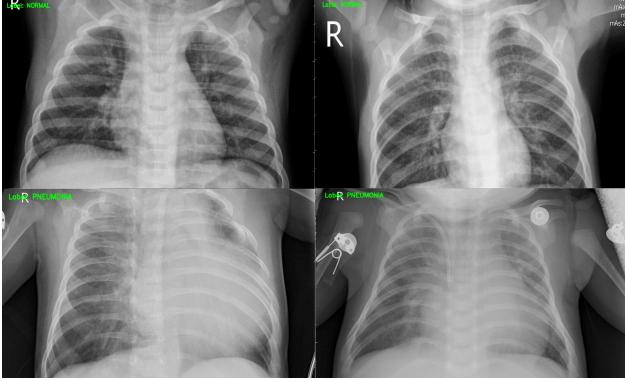


Fig. 3. The pre-trained CNN weights are loaded from the disk and make predictions for 30 randomly selected images. In the upper left and right corner we have an example of normal lungs, and in the lower left and right corner an example of pneumonia lungs.

TABLE II  
CNN WITH BATCH NORMALIZATION: RESULTS

	precision	recall	f1-score
normal	0.93	0.92	0.92
pneumonia	0.97	0.97	0.97
accuracy			0.96
macro avg	0.95	0.95	0.95
weighted avg	0.96	0.96	0.96

In Table 2 we see that the CNN with batch normalization obtained the same classification accuracy of 95% after 30 epochs. However, in analyzing the curves shown in Fig. 4 we see that our network learned until the 30 epoch, because further training past epoch 30 would result in overfitting and a wider generalization gap (loss function - the gap between the training loss and validation loss).

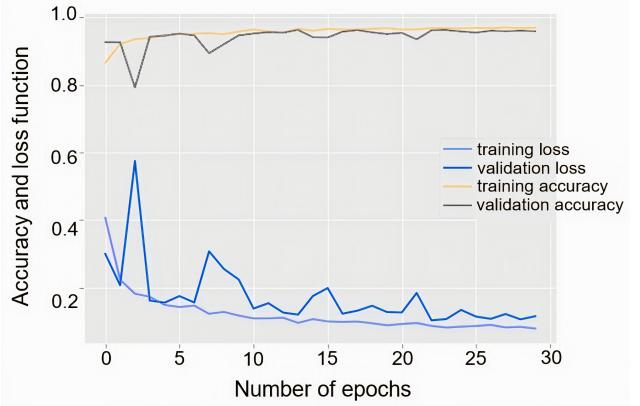


Fig. 4. A graph depicting a convolutional neural network with batch normalization, that resembles the VGGNet – training and validation loss and accuracy curves

TABLE III  
TRANSFER LEARNING USING VGG16

	precision	recall	f1-score
normal	0.96	0.93	0.95
pneumonia	0.97	0.98	0.98
accuracy			0.97
macro avg	0.97	0.96	0.96
weighted avg	0.97	0.97	0.97

In Table 3 we can see the results obtained by using transfer learning have a classification accuracy of 97%, which is by far the best. Furthermore, we observe that the CNN with batch normalization had a higher recall and a problem with overfitting past epoch 30, therefore the CNN without it seems like a better choice. Nevertheless, it is clear then when taking into account all three approaches we shall choose transfer

learning, because not only does it yield a higher classification accuracy, but it also wasted less computational time.

## V. CONCLUSION

In this paper we described two different approaches of using convolutional neural networks too classify a dataset consisting of normal and pneumonia infected lungs. We used a CNN that we constructed based on the VGGNet and implemented it with and without batch normalization. Furthermore, we used a transfer learning technique by extracting features of the neural network VGG16 trained on the ImageNet dataset. The main idea of this paper was to see if a different approach can have better results on this particular dataset, as well as see if a smaller neural network could have almost as good classification, as transfer learning. The final results, when compered had a higher classification accuracy by a couple of percentages, and also achieved so in just 30 epochs, as opposed to 100 epochs, so we can conclude the goal was obtained.

Further research will focus on implementing different types of optimizers, including metaheuristic algorithms as optimizers. Also, we will focus on battling larger datasets and obtaining high classification accuracy using various methods.

## ACKNOWLEDGMENT

This paper was conceived within the research on the project: “*Integrated research in the field of macro, micro and nano mechanical engineering - Deep machine learning of intelligent technological systems in production engineering*”, The Ministry of Education, Science and Technological Development of the Republic of Serbia (contract no. 451-03 - 68 / 2020-14 / 200105), 2020.

This work was financially supported by the Ministry of Education, Science and Technological Development of the Serbian Government, Grant TR-35029 (2018-2020).

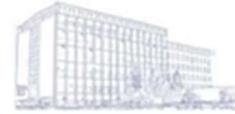
## REFERENCES

- [1] K. Fukushima, S. Miyake, “Neocognitron: A new algorithm for pattern recognition tolerant of deformations and shifts in position,” *Pattern Recognition*, vol. 15, no. 6, pp. 455-469, 1982.
- [2] Y. LeCun, B. E. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. E. Hubbard, L. D. Jackel, “Handwritten digit recognition with a back-propagation network,” *Advances in Neural Information Processing Systems* 2, pp. 396-404, June, 1990.
- [3] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, L. Fei-Fei, “ImageNet large scale visual recognition challenge,” *International Journal of Computer Vision*, vol. 115, no. 3, pp. 211-252, April, 2015.
- [4] A. Krizhevsky, I. Sutskever, G. E. Hinton, “ImageNet classification with deep convolutional neural networks,” *Advances in Neural Information Processing Systems*, vol. 25, no. 2, pp. 1097-1105, 2012.
- [5] M. D. Zeiler, R. Fergus, “Visualizing and understanding convolutional networks,” 13<sup>th</sup> European Conference, Zurich, Switzerland, pp. 818-833, September 6-12, 2014.
- [6] K. Simonyan, A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” arXiv preprint arXiv:1409.1556, 2014.
- [7] Y. LeCun, L. Bottou, Y. Bengio, P. Haffner, “Gradient-based learning applied to document recognition,” *Proceedings of the IEEE*, vol. 86, n. 11, pp. 2278-2324, November, 1998.
- [8] Z. Li, W. Yang, S. Peng, F. Liu, “A survey of convolutional neural networks: Analysis, applications and prospects,” arXiv preprint arXiv:2004.02806, 2020.
- [9] L. Shao, F. Zhu, X. Li, “Transfer learning for visual Categorization: A survey,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 26, no. 5, pp. 1019-1034, May, 2015.
- [10] I. Rudan, C. Boschi-Pinto, Z. Biloglav, K. Mulholland, H. Campbell, “Epidemiology and etiology of childhood pneumonia.”, *Bulletin of the World Health Organization*, vol. 86, no. 5, pp. 408-416, May, 2008.
- [11] World Health Organization, Pneumonia, <https://www.who.int/news-room/fact-sheets/detail/pneumonia>, (last accessed 06/07/2020).
- [12] D. S. Kermany, M. Goldbaum, W. Cai, C. C. S. Valentim, H. Liang, S. L. Baxter, A. McKeown, G. Yang, X. Wu, F. Yan, J. Dong, M. K. Prasadha, J. Pei, M. Y. L. Ting, J. Zhu, C. Li, S. Hewett, J. Dong, I. Ziyar, A. Shi, R. Zhang, L. Zheng, R. Hou, W. Shi, X. Fu, Y. Duan, V. A. N. Huu, C. Wen, E. D. Zhang, C. L. Zhang, O. Li, X. Wang, M. A. Singer, X. Sun, J. Xu, A. Tafreshi, M. A. Lewis, H. Xia, K. Zhang, “Identifying medical diagnoses and treatable diseases by image-based deep learning,” *Cell*, vol. 172, no. 5, pp. 1122-1131, February, 2018.
- [13] Public datasets; Chest X-Ray Images (Pneumonia), Version 2 <https://www.kaggle.com/paultimothymooney/chest-xray-pneumonia>, (last accessed 09/03/2020).
- [14] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, R. R. Salakhutdinov, “Improving neural networks by preventing co-adaptation of feature detectors”, arXiv preprint arXiv:1207.0580, Jul, 2012.
- [15] S. Ioffe, C.Szegedy, “Batch normalization: Accelerating deep network training by reducing internal covariate shift”, Proceedings of the 32<sup>nd</sup> International Conference on Machine Learning, vol. 37, pp. 448-456, July, 2015.
- [16] A. Rosebrock, *Deep Learning for computer vision with Python: Starter Bundle*, 1st ed. PyImageSearch, 2017.
- [17] C. Tan, F. Sun, T. Kong, W. Zhang, C. Yang, C. Liu, “A survey on deep transfer learning,” 27<sup>th</sup> International Conference on Artificial Neural Networks, Rhodes, Greece, October 4-7, 2018.
- [18] Keras; Python Deep Learning Library <https://keras.io>, (last accessed 09/03/2020).
- [19] TensorFlow; An end-to-end open source machine learning platform for everyone. <https://www.tensorflow.org>, (last accessed 09/03/2020).



THE 5<sup>th</sup> INTERNATIONAL CONFERENCE  
MECHANICAL ENGINEERING IN XXI CENTURY

December 9-10, 2020  
Faculty of Mechanical Engineering in Niš, Serbia



# Classification of COVID-CT Images Utilizing Four Types of Deep Convolutional Neural Networks

Lara LABAN and Mitra VESOVIĆ

First Author affiliation: Control Engineering, Faculty of Mechanical Engineering, Kraljice Marije 16, Belgrade, Serbia  
Second Author affiliation: Control Engineering, Faculty of Mechanical Engineering, Kraljice Marije 16, Belgrade, Serbia

[laralaban@mas.bg.ac.rs](mailto:laralaban@mas.bg.ac.rs), [mvesovic@mas.bg.ac.rs](mailto:mvesovic@mas.bg.ac.rs)

**Abstract**— In this paper a method is presented for the classification of COVID-CT (CT\_COVID, CT\_NonCOVID) image data set. Four different types of deep convolutional neural networks are proposed, two with the architecture resembling the VGGNet, one resembling the LeNet-5 and one using transfer learning. In addition, neural networks utilized the following techniques: decay, dropout and batch normalization. Since we needed to combat a significantly small dataset, we used data augmentation in order to transform and expand our dataset. Moreover, juxtapositions were made when observing the results given by these four neural networks, as well as the affect made by two different optimizers. The training of the neural networks was done using small batches with a binary cross entropy loss function, in order to achieve an up to scratch classification accuracy.

**Keywords**— deep learning; convolutional neural networks; image classification; data augmentation; batch normalization; COVID-CT dataset; dropout; transfer learning.

## I. INTRODUCTION

Convolutional neural networks (CNNs) are a subset of deep neural networks, which are used for classifying images. The main idea is to take a set of images correctly labeled as the input data and used them to train our neural network so as to achieve an output with an appropriate categorization [1]. The inspiration for CNNs comes from the observation of the animal visual cortex. Conversely, the flourishing of these networks only came recently due to the increase of computational power and the development of many possible libraries that could be used to battle complex mathematically based problems, such as back propagation. The first paper [2] that introduced the convolutional neural networks as we have come to know them today has demonstrated that a model which consists of a multilayered network can be successfully used for recognition of stimulus patterns according to the differences in their shapes. However, there is some debate that the true beginning was when a paper in 1990 [3] demonstrated that a CNN model which aggregates simpler features into progressively more complicated features can be successfully used for handwritten character recognition. In 2012 the ImageNet Large Scale Visual Recognition Challenge [4], at that moment consisting of over 1000 categories and 1.2 million images received a submission that would propel the CNNs development once again.

AlexNet [5] achieved a top-5 error of 15.3% , which at the moment surpassed by an astonishing 10% all of the other submissions, and had a much faster training time as it was implemented on a GPU. The following year, the same challenge, now with a larger dataset was won by ZFNet [6]. It had the top-5 error of 14.8%, however even more so important is that it was able to reduce the first layer filter size from to and had a stride of 2, rather than 4 in the pooling layer.

VGG16 is a convolutional neural network model proposed in the paper [7]. This model achieved 92.7% top-5 test accuracy. The main contribution of this model was that it used kernel sized filters, instead of the . It was trained for weeks using GPUs, and had a huge computational cost. However, it introduced a new idea using the same kernels throughout the entire architecture, this aided in generalization for classification problems outside of what they were originally trained on. If for a second we go back to LeNet [8] that was the foundation for all of these previously mentioned CNNs we can observe the main sequence of three layers convolution, pooling and non-linearity still play the key part, and sometimes it is beneficial not to import to many layers when training a smaller dataset [9]. Finally, in recent years transfer learning [10], which addresses cross domain learning problems by extracting useful information from data in a related domain and transferring them for being used in target tasks, has been demonstrating a significant impact.

Coronavirus disease 2019 (Covid19) is an infectious disease caused by severe acute respiratory syndrome coronavirus 2 (SARS-CoV-2) [11]. It was first identified in December 2019 in Wuhan, Hubei, China, and has resulted in widespread pandemic. At the moment Covid19 has caused over one million deaths all around the world and counting. There are several methods that include quick testing, however in order to grasp the full scope of the problem some of the most important ways to battle this disease is to examine the computed tomography (CT) scan images. Chest CT scanning in patients with Covid19 has shown ground-glass opacification, possibly with consolidation, as well as cases of pleural effusion, pleural thickening and lymphadenopathy. Data is collected daily and it is still scarce, howbeit one of the first datasets was

proposed and constructed for a marvellous paper [12], that is still pending publication at the moment, in order to try and aid the ongoing battle against the pandemic. In it multi-task learning and contrastive self-supervising learning were used and achieved an accuracy of 89% in distinguishing between CT\_COVID and CT\_NonCOVID images. In addition to all that was stated beforehand the pivotal goal of this paper is to try and implement various CNNs to combat this classification problem and if possible obtain a slightly better classification accuracy.

This paper is organised in the following manner: section 2 represents a description of the dataset. In section 3 the main methods which are used are explained in detail, as well as the architecture of the CNN. As a result, in section 4 we discuss the results and compare the methods, based on accuracy and loss functions. In section 5, following a short summary a conclusion is made and future work and possible directions are stated.

## II. DATASET AND ITS IMPLEMENTATION

The dataset which is used in this paper consists of 672 CT scan images from patients [13], including 349 CT\_COVID and 323 CT\_NonCOVID. We took the approach of data augmentation, where we increase the diversity of data by altering the original samples using translation, rotation, shearing, flips and adding them to the training set. Data augmentation covers a wide range of techniques used to generate new training samples using the original input images, by applying random jitters and perturbations in such a manner as to not change the class labels. The main idea here is to decrease the generalization error of the testing (sometimes at the expense of the training error) so as to achieve an increase of generalizability of the model. The neural network is then using slightly modified versions of the input data and it is able to learn more robust features.

However, we introduced scaling of the data, as well, by computing a weight for each class during the training and as an outcome amplifying the loss by a larger weight when we approach the smaller dataset. Even though the difference is small in this example, this benefited the training process. During the preprocessing of images we resized all the images to a fixed size  $32 \times 23$ , and in doing so we also maintained the aspect ratio. The reasoning behind this being that all the images in a dataset need to have a fixed feature vector size. This means all the images will have identical widths and heights, making it easier to quickly load and preprocess a dataset and briskly move through our convolutional neural network. The aspect ratio will enable us to resize the images along the shorter dimension, be it width or height, and in cropping it, will maintain the ratio. It is important to note that this step is not necessary if you are not working with a difficult dataset. Notwithstanding its benefits, it was implemented in this particular dataset.

### A. ImageNet dataset

ImageNet is a dataset consisting of over 14 million images, which belong to one thousand classes. It was used as the dataset in the highly respected convolutional neural network model VGG16 which was proposed by Oxford scientists. In this paper the VGG16 network was used as a pre-trained convolutional neural network, in order to incorporate transfer learning.

## III. METHODS DESCRIPTION

In order to try and reduce overfitting and increase our classification accuracy on the CT\_COVID dataset we endeavour in performing three types of neural network training techniques:

- dropout and decay
- batch normalization
- transfer learning (neural networks as feature extractors)

The first technique that is used in order to improve the generalization error in the convolutional neural network is dropout [14]. Dropout is nothing more than a form of regularization, which succours us in controlling the model capacity. The dropout layers are arranged in the network in such a manner that we have randomly disconnected nodes by a probability of 0.3 in the first few layers; and 0.6 probability in the last layer. The reason for this is that if the first layers are dropped by a higher probability, then that will later affect the training. The dropout is implemented after the pooling layer, and before the next convolutional layer (or last flatten and dense layers). This was used for the neural networks resembling the VGG with data augmentation (DA). The network resembling the VGG without DA used a dropout with a probability 0.25 in the first few layers and double the increase in the last layer, while the LeNet network did not utilize this method. Decay that is used in this neural network is a standard decay that can be obtained using the Keras library in Python. Since the learning rate controls the step that is made along the gradient, larger steps are usually used in the beginning to make sure that we do not stagnate in the local optima, while smaller steps are used deeper in the network and near the end of the convolution in order to converge to a global minimum. We have initialized the learning rate to be 0.01 (for the networks with DA) and 0.05 (for the network without DA), and applied the following formula to adjust it after each epoch,

$$\alpha_{i+1} = \frac{\alpha_i}{1+k \cdot i} \quad (1)$$

where  $\alpha$  is the current learning rate,  $i$  is the epoch and  $k$  is the decay calculated as the division between the learning rate and the number of epochs. This type of adjustment of the learning rate each epoch, can increase accuracy, as well as reduce the loss function and the time necessary to train a network. Batch normalization [15] is used to normalize the activations of a given layer's inputs by applying mean and standard deviation before passing it onto the next layer. In addition, the covariate shift refers to a change in the distribution of the input variables which are present in the training and validation data. Since it has been proven that the training of the neural network is the most coherent when the inputs to each layer are alike, the main intention is that even when the explicit values of inputs layers to hidden layers change, their mean and standard deviation will still remain relatively the same, thus reducing the covariate shift. Batch normalization has demonstrated an immensely effective approach to reducing the number of epochs necessary for training by allowing each layer to learn independently. Here the idea that differs from the original paper and is first proposed in [16] states that the batch normalization should be implemented after the activation layer. The main reasoning behind this is that we

want to avoid setting the negative values coming out of the convolution layer to zero. Instead we pass them through the batch normalization layer, right after the activation (ReLU) layer, and assure that some of the features that otherwise would not have made it do. This yields a higher accuracy and lower loss, and is to this day a debate amongst the creators of Keras.

Finally, the second technique is transfer learning [17], a machine learning technique where networks can behave as feature extractors. Transfer learning is nothing more than the ability to use a pre-trained model to learn patterns from data, on which the original network was not trained on. As previously stated deep neural networks trained on a large scale dataset ImageNet have demonstrated to be superb at this task.

When treating networks as feature extractors we choose a point, in this case before the fully connected layer and remove it. Subsequently, in this particular example while using the VGGNet pre-trained on the ImageNet we removed the fully connected layer and stopped at the last pooling layer where the output shape is  $7 \times 7 \times 512$ , 512 filters with the size  $7 \times 7$ . Now, our feature vector has  $7 \times 7 \times 512 = 25088$  values and it will be used to quantify the contents of the images, which were not included in the original training process. The format which allows us to extract these features is the hierarchical data format version 5 (hdf5), which is used to store and organize large amount of data.

Transfer learning is an optimization, which has been proven to yield a better performance and drastically save time. This is precisely why we used it in this paper, to see if we could obtain a higher classification, and perform faster. Transfer learning relaxes the hypothesis that the training data must be independent and identically distributed with the test data, which we clearly stated as a must in the beginning of this chapter. Moreover, transfer learning is able to solve the problem of insufficient training data. Furthermore, there is the option to remove the fully connected layers of the existing network in order to add a new fully connected layer to the CNN and fine tune the weights to recognize object classes. However, here it was not implemented since treating networks as arbitrary feature extractors was enough.

#### A. Convolutional Neural Network

Into the bargain all that was explained, we picked the following CNN architecture shown in Fig. 1. It is consisted of multiple convolutional and pooling layers, as well as the fully connected layers. The first two convolutional layers learn 32 filter each with a size  $3 \times 3$ .

Sequentially, the fourth and the fifth layers learn 64 filters with the size  $3 \times 3$  and the last two learn 128 filters with the size  $3 \times 3$ . The pool layer is used to reduce the computational load and the number of parameters, thus reducing the risk of overfitting. We used a max pooling layer with a pool size  $2 \times 2$  and a stride 2. Finally, we have the fully connected layer which consists of 2048 parameters, input values which learn 512 nodes. The activation layers which were used are Rectified Linear Unit (ReLU) defined as,

$$f(x) = \max(0, x) \quad (2)$$

where  $x$  is the input into the neuron. Softmax or the normalized exponential function assigns normalized class probabilities for each prediction, and is represented by,

$$S(y_i) = \frac{e^{y_i}}{\sum_{j=1}^k e^{y_j}} \quad (3)$$

for  $i = 1, \dots, k$  and  $\mathbf{z} = (z_1, \dots, z_k) \in \mathbb{R}^k$ .

Softmax takes an input vector and normalizes it into a probability distribution between  $[0,1]$ . Therefore the sum of all output values is equal to 1, which in turn makes the training converge more quickly. In order to achieve this, before training we must include one hot encoding in order to convert the labels from integers to vectors.

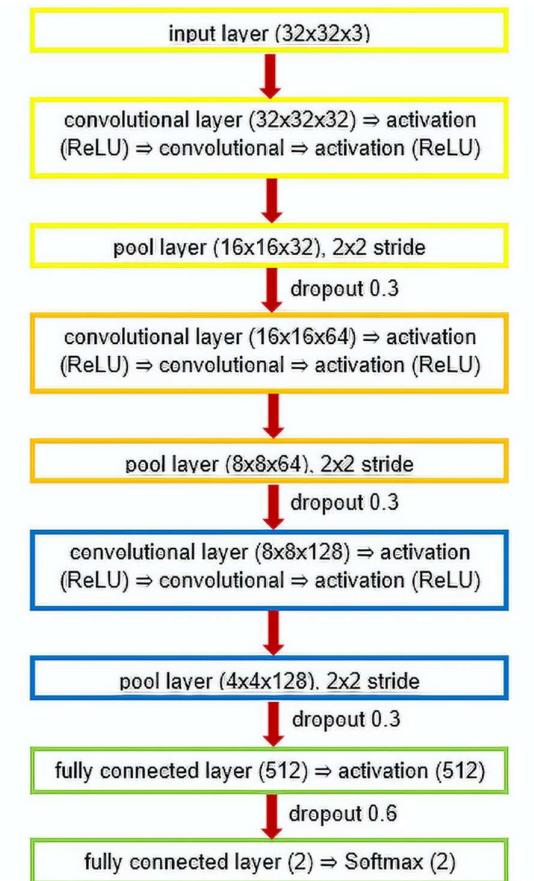


Fig. 1 A schematic of the convolutional neural network without batch normalization, that resembles the VGGNet. All of the convolutional layers that precede the fully connected layers have filters 32, 64, 128 that are the same size. The probability distribution is applied in the last layer using Softmax and the output yields two class labels CT\_Covid and CT\_NonCovid.

In addition, later when we want to add the batch normalization layer, we can apply it after each activation layer, as discussed previously.

#### B. Implementation and training of a simpler version of the LeNet

Taking into the bargain all that was explained before, the implementation of this CNN was done by using the Python programming language. We used Keras [18] which is mainly used for implementing of activation functions, optimizers, convolutional and pooling layers, and is actually able to do backpropagation automatically.

Right after we load and preprocess our images dataset it is necessary to use one hot encoding. This is done by using a part of the Sklearn library LabelBinarizer. However beforehand we must split the training data and the validation data, here we opted to split it 75% and 25%, sequentially. The next step is the implementation of an optimizer, here we used the Adam optimizer. The Adam optimizer is short for Adaptive Moment Estimation optimization algorithm [19]. Its main purpose is to attempt to rectify the negative effects of a globally accumulated cache by converting the cache into an exponentially weighted moving average, just like the Root Mean Square Propagation (RMSProp). The Adam optimizer is essentially a combination of momentum and RMSProp. Momentum is implemented into the neural network, by adding a temporal element to the update vector of the past time step to the current update vector,

$$\Delta w(k) = -\alpha \nabla E(k) + \gamma \Delta w(k-1) \quad (4)$$

where  $\gamma$  is usually set between 0.8 and 0.9 and function  $E$  is the index of performance.

This network resembles the architecture of the LeNet in such a way that we have 20 filters the size of  $5 \times 5$  in the first convolution layer, and 50 in the second convolutional layer. The mini batch method were the neural network selects a part of the training data and updates the weights, but trains the network with the average weight update. Usually the smallest standard batch size which is used is 32, however we opted to use 24, as it complemented our data. The reasoning behind this is that present research confirms that using small batch sizes achieves the best training stability and generalization performance, for a given computational cost, across a wide range of experiments. The loss function which was used is the binary\_crossentropy function. This was done because we only had two classes, if there were more we would have had to use categorical\_crossentropy, but have in mind we could have used categorical as well, but studies show that binary is much more efficient in this case. The training of the CNN was done in 30 epochs.

### C. Implementation and training of a simpler version of the VGGNet

We constructed two different neural networks resembling the VGG, the first one had a dropout of 0.25 in the first few layers and no data augmentation or batch normalization layers. The training data and the validation data were split 75% and 25%, sequentially. Here we utilized the SGD optimizer, which was set to a learning rate of 0.05, with a decay in order to slowly reduce the learning rate over time and converge to the global solution more efficiently. Decaying the learning rate is beneficial in reducing overfitting and obtaining a higher classification accuracy. The smaller the learning rates are, the smaller the weight update will be enabling us to converge. The gradient descent method is an iterative optimization algorithm that operates over an optimization surface. It is a simple modification to the standard algorithm of gradient descent. The main purpose of SGD is to calculate the gradient and adjust the weights of the training data (but not on the whole dataset, but rather on a mini batch). All the images were resized to  $32 \times 32$  aspect ration, the batch size we used was again 24 and the loss

function was the binary\_crossentropy function. The training of the CNN was done in 30 epochs.

The second neural network resembling the VGG had a similar architecture as depicted in Fig. 1, with the addition of batch normalization layers. Here our training data and the validation data were split 80% and 20%. Here we trained our network once with the SGD optimizer and once more with the Adam optimizer. The learning rate was set to 0.01, with decay and adjustment after each epoch. Both times data scaling, as well as data augmentation was used. All the images were resized to  $32 \times 32$  aspect ration, the batch size we used was 32 and the loss function was the binary\_crossentropy function. The training of the CNN was done in 30 epochs (utilizing the Adam optimizer) and in 100 epochs (using the SGD optimizer).

After the training we implemented a method that takes the weights and the state of the optimizer and serializes them to the disc in a hdf5 format, in order to load them and test the labeling.

### D. Implementation using transfer learning

The first step in this process is to extract features from VGG16, in doing so we are forward propagating the images until a given layer, and then taking those activations and treating them as feature vectors. Here the main two differences are that we used the standard a batch size of 32 and the training and test split is done at the same time as training, we again split it into 75% training data and 25% test data. Once the extraction of the features was done, we trained the classifier on those features. We also implement the GridSearchCV class to assists us to turn the parameters to the LogisticRegression classifier.

The final results are presented in the following chapter, comparisons are made and a visual representation of the graphs is shown using Matplotlib in order to estimate if there is overfitting.

## IV. RESULTS AND COMPARISONS

The results of the CNN resembling LeNet are presented in Table 1. We clearly see that our neural network has classification accuracy of 68%.

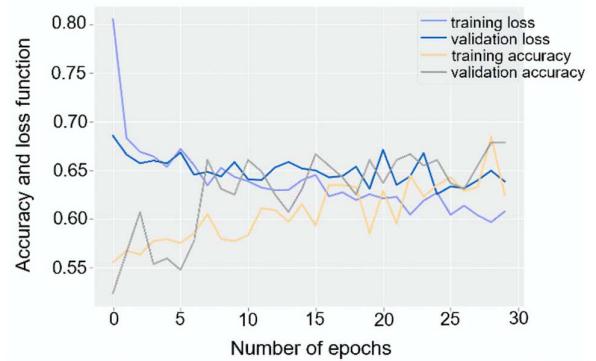


Fig. 2 A graph depicting a convolutional neural network that resembles the LeNet – training and validation loss and accuracy curves

In the following table we use the term precision which represents true positive divided by a sum of true positive and false positive, recall which represents true positive divided by a sum of true positive and false negative. Therefore, precision is good to determine when the cost of false positives is high, on the other hand recall tells us the

number of correctly labeled data. Ultimately, we have the f1-score used to find the weighted average of recall and precision. In analyzing the curves shown in Fig. 2 we see that our network learned until the 30 epoch, beyond that overfitting would occur, as we can clearly see a generalization gap forming in both loss and accuracy curves. Fig. 3 depicts the results when using the network resembling the VGG without data augmentation, here we can observe that the training and validation curves show a wide generalization gap at the 30 epoch resulting in overfitting. The classification accuracy is 76% (Table 1.), this is no good if we have overfitting, that is why the next approach uses data augmentation in order to combat this problem.

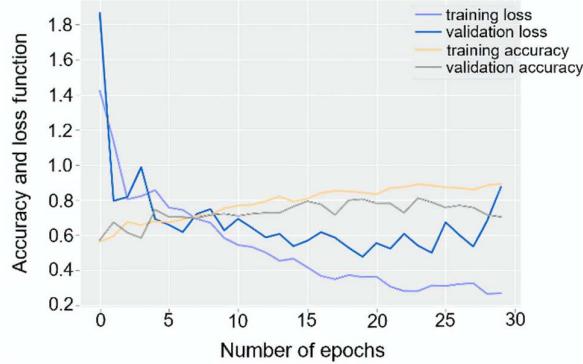


Fig. 3 A graph depicting a convolutional neural network without data augmentation and with batch normalization, that resembles the VGGNet – training and validation loss and accuracy curves

Fig. 4 represents the neural network resembling the VGG, with data augmentation and the SGD optimizer. The classification accuracy obtained after 30 epochs is 72%, and the training and loss curves show slight deviations.

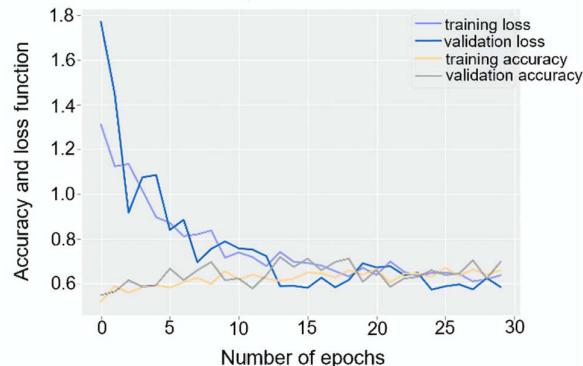


Fig. 4 A graph depicting a convolutional neural network with data augmentation and batch normalization (optimizer SGD), that resembles the VGGNet – training and validation loss and accuracy curves – 30 epochs

The same classification accuracy is acquired when utilizing the Adam optimizer, only then we need 100 epochs to achieve so. Fig. 6 depicts the same neural network explained beforehand when using the SGD optimizer over the course of 100 epochs resulting in a classification accuracy of 75%. We can conclude that data augmentation does indeed help in reducing the generalization gap, however this particular dataset was quite faulty to begin with.

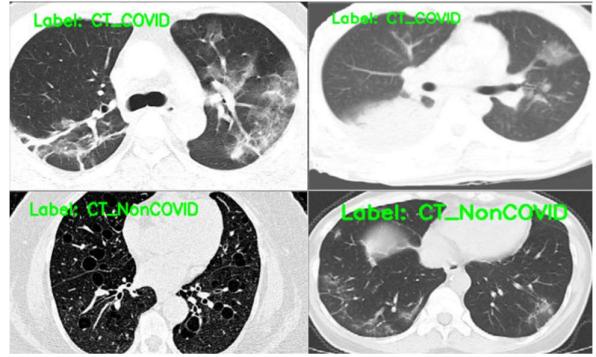


Fig. 5 The pre-trained CNN weights are loaded from the disk and make predictions for 30 randomly selected images. In the upper left and right corner we have an example of CT\_COVID scans, and in the lower left and right corner an example of CT\_NonCOVID scans.

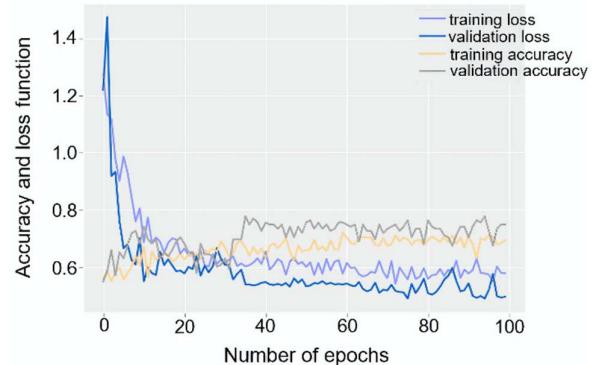


Fig. 6 A graph depicting a convolutional neural network with data augmentation and batch normalization (optimizer SGD), that resembles the VGGNet – training and validation loss and accuracy curves – 100 epochs

TABLE I  
EXPERIMENTAL RESULTS

	precision	recall	f1-score
<b>CNN resembling LeNet (Adam optimizer, with data augmentation)</b>			
macro avg	0.68	0.68	0.67
<b>CNN resembling VGG (SGD optimizer, without data augmentation, without batch normalization)</b>			
macro avg	0.76	0.71	0.69
<b>CNN resembling VGG (Adam optimizer, with data augmentation, 100 epochs, with batch normalization)</b>			
macro avg	0.72	0.72	0.72
<b>CNN resembling VGG (SGD optimizer, with data augmentation, 30 epochs, with batch normalization)</b>			
macro avg	0.72	0.70	0.69
<b>CNN resembling VGG (SGD optimizer, with data augmentation, 100 epochs, with batch normalization)</b>			
macro avg	0.75	0.75	0.75
<b>Transfer learning using VGG16</b>			
macro avg	0.90	0.91	0.90

In Table 1 we can see the results obtained by using transfer learning have a classification accuracy of 90%, which is by far the best. Furthermore, we observe that the CNN in Fig. 4 is the best one if we opted to use a method that does not include transfer learning. Nevertheless, it is clear then when taking into account all four approaches we shall

choose transfer learning, because not only does it yield a higher classification accuracy, but it also wasted less computational time. Compared with the original paper that combated this classification problem [12] we were able to achieve only slightly better classification accuracy, with an increase being 1%.

## V. CONCLUSIONS

In this paper we described four different approaches of using convolutional neural networks to classify a dataset consisting of CT\_COVID and CT\_NonCOVID images. We used CNNs that we constructed based on the VGGNet and LeNet5 and implemented them with and without data augmentation. Furthermore, we used a transfer learning technique by extracting features of the neural network VGG16 trained on the ImageNet dataset. The main idea of this paper was to see if a different approach can have better results on this particular dataset, as well as see if a smaller neural network could have almost as good classification as transfer learning. The final results, when compared showed a clear advantage when using transfer learning, however it also showed us the importance of data augmentation when approaching a rather small dataset.

Further research will focus on implementing different types of optimizers, including metaheuristic algorithms as optimizers. Also, we will focus on battling larger datasets consisting of Covid19 CT scans, once they become available, as well as obtaining a higher classification accuracy utilizing different methods.

## ACKNOWLEDGMENT

This research was supported by the Science Fund of the Republic of Serbia, grant No. 6523109, AI- MISSION4.0, 2020-2022.

This paper was conceived within the research on the project: “*Integrated research in the field of macro, micro and nano mechanical engineering - Deep machine learning of intelligent technological systems in production engineering*”, The Ministry of Education, Science and Technological Development of the Republic of Serbia (contract no. 451-03 -68 / 2020-14 / 200105), 2020.

This work was financially supported by the Ministry of Education, Science and Technological Development of the Serbian Government, Grant TR-35029 (2018-2020).

## REFERENCES

- [1] L. Laban, R. Jovanović, M. Vesović, V. Zarić, “Classification of Chest X-Ray Images Using Deep Convolutional Neural Networks”, International Conference on Electrical, Electronic, and Computing Engineering, Belgrade, Serbia, September 28-30, 2020, to be published.
- [2] K. Fukushima, S. Miyake, “Neocognitron: A new algorithm for pattern recognition tolerant of deformations and shifts in position,” Pattern Recognition, vol. 15, no. 6, pp. 455-469, 1982.
- [3] Y. LeCun, B. E. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. E. Hubbard, L. D. Jackel, “Handwritten digit recognition with a back-propagation network,” Advances in Neural Information Processing Systems 2, pp. 396-404, June, 1990.
- [4] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, L. Fei-Fei, “ImageNet large scale visual recognition challenge,” International Journal of Computer Vision, vol. 115, no. 3, pp. 211-252, April, 2015.
- [5] A. Krizhevsky, I. Sutskever, G. E. Hinton, “ImageNet classification with deep convolutional neural networks,” Advances in Neural Information Processing Systems, vol. 25, no. 2, pp. 1097-1105, 2012.
- [6] M. D. Zeiler, R. Fergus, “Visualizing and understanding convolutional networks,” 13th European Conference, Zurich, Switzerland, pp. 818-833, September 6-12, 2014.
- [7] K. Simonyan, A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” arXiv preprint arXiv:1409.1556, 2014.
- [8] Y. LeCun, L. Bottou, Y. Bengio, P. Haffner, “Gradient-based learning applied to document recognition,” Proceedings of the IEEE, vol. 86, n. 11, pp. 2278-2324, November, 1998.
- [9] Z. Li, W. Yang, S. Peng, F. Liu, “A survey of convolutional neural networks: Analysis, applications and prospects,” arXiv preprint arXiv:2004.02806, 2020.
- [10] L. Shao, F. Zhu, X. Li, “Transfer learning for visual Categorization: A survey,” IEEE Transactions on Neural Networks and Learning Systems, vol. 26, no. 5, pp. 1019-1034, May, 2015.
- [11] <https://www.mayoclinic.org/diseases-conditions/coronavirus/symptoms-causes/syc-20479963>, (last accessed 13/10/2020).
- [12] X. Yang, X. He, J. Zhao, Y. Zhang, S. Zhang, P. Xie, “COVID-CT-Dataset: A CT Scan Dataset about COVID-19”, <https://arxiv.org/abs/2003.13865>, 2020, to be published.
- [13] Dataset provided on GitHub, <https://github.com/UCSD-AI4H/COVID-CT>, (last accessed 23/07/2020).
- [14] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, R. R. Salakhutdinov, “Improving neural networks by preventing co-adaptation of feature detectors”, arXiv preprint arXiv:1207.0580, July, 2012.
- [15] S. Ioffe, C. Szegedy, “Batch normalization: Accelerating deep network training by reducing internal covariate shift”, Proceedings of the 32nd International Conference on Machine Learning, vol. 37, pp. 448-456, July, 2015.
- [16] A. Rosebrock, Deep Learning for computer vision with Python: Starter Bundle, 1st ed. PyImageSearch, 2017.
- [17] C. Tan, F. Sun, T. Kong, W. Zhang, C. Yang, C. Liu, “A survey on deep transfer learning,” 27th International Conference on Artificial Neural Networks, Rhodes, Greece, October 4-7, 2018.
- [18] Keras; Python Deep Learning Library <https://keras.io>, (last accessed 09/03/2020).
- [19] D. P. Kingma, J. Ba, “Adam: A Method for Stochastic Optimization”, 3<sup>rd</sup> International Conference for Learning Representations, San Diego, 2015.



# Selection of the Most Efficient Temperature Measurement Method Based on Desired Response Time and Experimental Results

**Lara G. Laban, Nedžad R. Rudonja, Milan D. Gojak**

Faculty of Mechanical Engineering, University of Belgrade

**Abstract:** This paper aims to present how temperature can be measured in the fastest and most accurate way using different types of sensors. It closely describes the use of simplistic temperature measurement via thermometer liquids and bimetallic thermometers, as well as the advanced measurements using thermocouples, NTC resistors, etc. Semiconductor materials which behave as ideal isolators at very low temperatures are depicted as the most slow rate response sensors, because of the time it takes for them to heat up. Whilst in most cases thermocouples type K have the quickest response time, and the most effective, due to the fact that their junction, when heated, generates thermoelectric voltage; this allows the measuring of the temperature. Also, the temperature measurement with thermistors resistance is very much explained using 2, 3, 4 -conductor connection. Furthermore, explained in detail is the main purpose which is to enable the documenting of characteristic curves via a 3 channel line recorder using measured values from the electrical temperature sensors and providing a response time through readings. In the finale stage of the paper experimental readings, calculations and measurement are shown and compared with one another according to the use of non protection tubes, tubes of high-grade steel and immersion tubes made of brass.

**Key words:** Temperature measurement, NTC thermistor, Pt100 sensor, Thermocouple, Response time.

## 1. INTRODUCTION

One can state that the temperature is a measure of a material's internal molecular activity. Whilst the level of molecular activity increases, the temperature of the material upsurges. Hot and cold are qualitative descriptions of a change in molecular activity. The necessity of more accurate ways to describe temperature led to the development of temperature-measuring devices otherwise known as sensors. They use standard, universally recognized temperature scales, because not only do these scales rely on fixed points in nature (for example freezing point of water), they also provide a way to describe temperature objectively and quantitatively [1].

The four temperature scales which are used today are Fahrenheit, Celsius, Kelvin, and Rankine. Fahrenheit based his research on the Romer scale, which was formerly introduced in using simple association; normal temperature was known as 24°C, and zero degrees was the coldest day. Fahrenheit described the mercury-in-glass thermometer, introducing three temperature fixed points: a mixture of ice, ammonium chloride and water was the zero point, mixture of water and ice was 32°C and the human body temperature was taken as 96°C. Furthermore, the development on the thermodynamic temperature scale was made by astronomer and physicist Celsius; he assigned 100°C to the temperature of boiling water. So in conclusion, the aim of any scale of temperature, especially the thermodynamic scale, is the representation of hotness and hotness relations between objects and events in the real physical world by real numbers [2].

Though there are many forms of temperature measurement, some are quite simple do not require units or special equipment. For example, when there is a use of a glass thermometer, the relative expansion of a liquid is measured in a vessel. Liquid expands at a regular,



measureable rate when it is heated. This is the main reason for the use of liquid in a narrow glass tube. Practically any liquid can be used as the thermometer filling. A distinction is made here between wetting and non-wetting liquids. Mercury, a non-wetting liquid, is one of the most familiar materials used in liquid thermometers; other liquids, such as kerosene or ethanol, may also be used in these types of thermometers.

However, wetting liquids can cause additional errors when the temperature drops. Organic liquids are dyed in order to be better visible and more easily readable in the capillary. The main portion of the liquid is contained in a spherical or cylindrical vessel which acts as the sensor component of the thermometer. The change in volume of thermometer liquids is governed by the following law [3]:

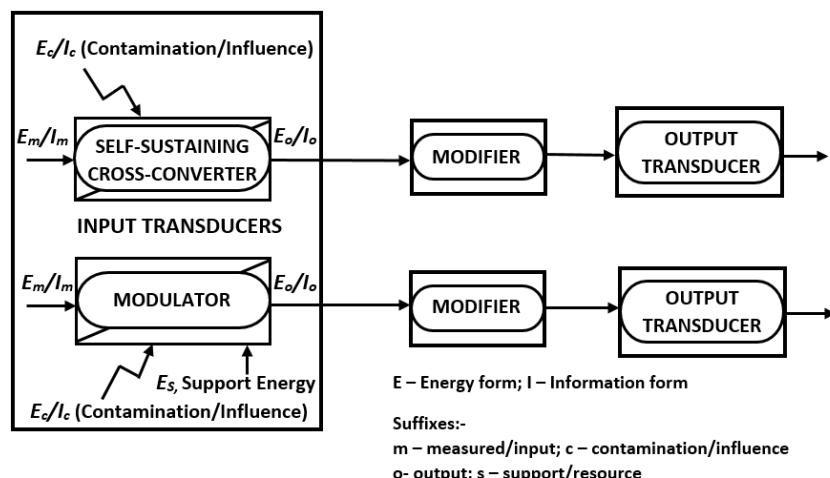
$$V_t = V_0 \cdot (1 + \gamma \cdot \Delta t) \quad (1)$$

On the other hand, bimetallic thermometers make use of the differential expansion of two different materials in order to indicate changes in temperature. Two or more layers of different materials are rolled on top of each other, making it possible to achieve different shapes depending on the intended application. The length of individual metals changes in accordance with the following formula [3]:

$$L_t = L_0 \cdot (1 + \alpha \cdot \Delta t) \quad (2)$$

## 2. TEMPERATURE MEASUREMENT

A temperature sensor is the initial part of a temperature measurement, therefore an instrumentation chain is depicted in the following photo.



**Figure 1:** Block diagram of temperature measuring chains [2]

These sensors may be either self-supporting cross-convertors or modulators. Self-sustaining cross-converter types of temperature sensors extract energy from the system under measurement during the conversion of an information bearing signal in the thermal energy domain into an information bearing signal in another, different, energy form. Modulating temperature sensors require the supply by an external power source to support the acquisition and flow of the temperature information. The sensor, which is also called and initial transducer, is the thermometer. In temperature sensors, which are the front end elements in



temperature instrumentation, the main output is an information output. This quantity, known as the measuring signal, is subjected to further transformation in a modifier, such as data converter, an amplifier, a filter or other kind of conditioner, into the desired output signal [2].

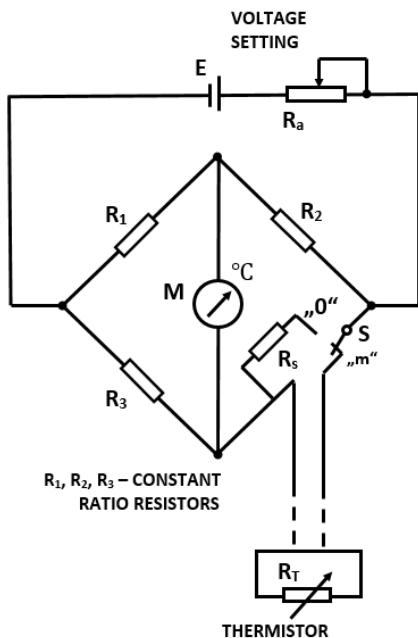
## 2.1. Temperature measurement with thermistors

These special resistance elements are made of semiconductor materials. There are two different types: NTC resistors (NTC thermistors) and PTC resistors (PTC thermistors). All semiconductor elements act as ideal isolators at very low temperatures. Their conductivity increase is described approximately by the following law [3]:

$$R_T = R_0 \cdot e^{b \cdot \left( \frac{1}{T} - \frac{1}{T_0} \right)} \quad (3)$$

Here,  $R_0$  and  $R_T$  are the sensor resistances at the absolute temperature  $T$  and a reference temperature  $T_0$  [K]. “ $b$ ” is a material constant, whose numerical value lies between 2000 and 6000 K [3].

The common forms of thermistor thermometer measuring circuit are deflection type bridge circuits [2].



**Figure 2:** Deflection type bridge circuit for a thermistor thermometer

The bridge energy source may be a battery cell or a rectified supply voltage. To ensure that the supplying voltage remains constant, a standardizing resistor,  $R_s$ , is provided. In the position ‘O’ of the switch, S, where  $R_s$  temporarily replaces the thermistor,  $R_t$ , the value of  $R_a$  is adjusted in such a way that the readings of the meter, M, are brought to a marked scale position. This is not necessary when a stabilised voltage source is used. Measuring temperatures ranges of 30 to 50°C may easily be achieved. The whole measuring range is divided into several selectable sub-ranges. Most producers now supply thermistor thermometers in deflection type bridge circuits with an IC output amplifier guaranteeing a precision of 0.5 to 1.0°C [2].

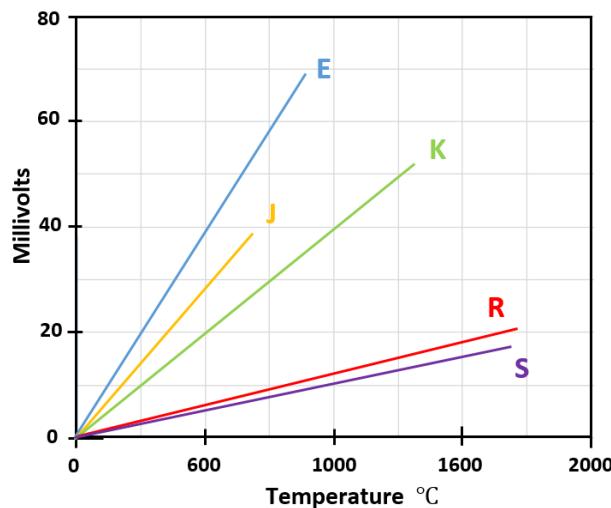


## 2.2. Temperature measurement with thermocouples

The thermoelectric effect was first technically utilized by Seebeck and Peltier for the purpose of temperature measurement [3].

The combination of two dissimilar conductors, which may be metals, alloys or non-metals, connected at one end is known as a thermocouple. The point of connection is called the measuring junction and their free ends are referred to as the reference junction [2].

The emf versus temperature values of the more commonly used thermocouples are shown in the following photo.



**Figure 3:** MilliVolts generated by various thermocouple types [1]

The sensitivity of a thermocouple is the algebraic sum of the thermoelectric voltages of both conductors. The greater the difference between the two thermoelectric voltages, the higher the sensitivity. If two different materials are welded to form a thermocouple and their junction is subsequently heated, an electromotive force - or thermoelectric voltage - is generated. The thermoelectric voltage permits temperature to be treated as a measurement variable.

Due to their high long-term stability, type K thermocouples are usually employed in practice. This type of thermocouple is 10 times more stable than the Fe-CuNi element and 100 times more stable than the Cu-CuNi element [3].

## 2.3. Temperature measurement with thermistors resistance thermometers

The dependence of the resistance of metallic conductors on temperature can be expressed very precisely using cubic equations. However, quadratic or even linear functions are considered sufficient for normal precision requirements [1]:

$$\Delta R = \alpha \cdot R \cdot \Delta t \quad \text{or} \quad R_t = R_0 \cdot (1 + \alpha \cdot t) \quad (4)$$

$R$ ,  $R_0$  and  $R_t$  [ $\Omega$ ] are resistances in the original state, at  $0^\circ\text{C}$  and at  $t[^\circ\text{C}]$ .  $[1/\text{K}]$  is the linear temperature coefficient of the resistor. As the linear equation does not apply precisely for most materials, i.e. is not constant, a mean relative change in resistance per degree between  $0$  and  $100^\circ\text{C}$  is defined [1]:



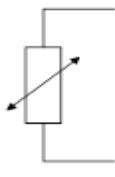
$$\alpha_{0,100} = \frac{1}{R_0} \cdot \frac{\Delta R}{\Delta t} = \frac{1}{R_0} \cdot \frac{R_{100} - R_0}{100} \quad [K^{-1}] \quad (5)$$

$R_{100}$  and  $R_0$  are measured at the boiling and freezing points of water. There are several possibilities of connecting resistance thermometers.

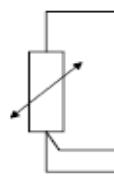
A two-conductor connection (displayed on Figure 4) does not take into account the influences of the line resistances generated by temperature fluctuations or high currents. Every change in line resistance misleadingly indicates a change in temperature on the sensor.

In the case of a three-conductor connection (displayed on Figure 5), one end of the shunt is connected via a measurement cable isolated from the supply. This makes it possible to reduce the influence of temperature on the lines. Here too, however, line balancing by means of adjustable resistors is necessary.

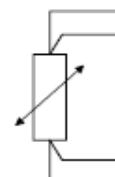
In the case of four-conductor connection (displayed on Figure 6) used for compensation measurements, two lines serve for current supply and two lines for voltage tapping via the shunt. In this case, the supply lines do not need to be balanced even if they are long [3].



**Figure 4** [3]



**Figure 5** [3]



**Figure 6** [3]

### 3. EXPERIMENTS AND RESPONSE TIME

The response of a heat sensor, the time taken by the sensor to indicate the actual temperature of a medium, depends essentially on its thermal resistance and heat storage capacity. The higher the thermal resistance, the more time the sensor takes to warm up.



**Figure 7:** WL920: advanced temperature measurement trainer consists of the ability to obtain a wide range of temperature measurements and record them simultaneously.

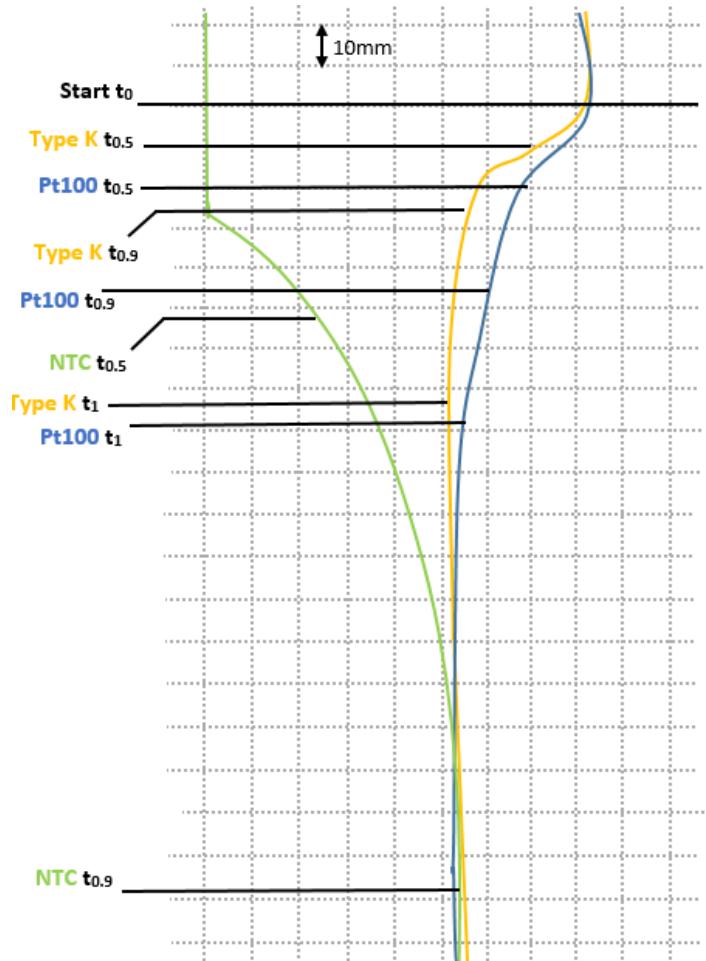


Short response times are achieved by using small sensors made of materials with a high heat conductivity. Response times are lengthened, in particular, by insulating air gaps between the measuring element and the protective tube, and between the sensor and any fitted immersion tube.

Protective tubes and immersion tubes of large diameters also result in long response times. Information on a response time can be obtained from the corresponding transient function, which indicates measurement values as a function of stepped changes in sensor temperature. It can be determined by placing a thermometer in a stream of water or air.

Two periods characterize the transient function: The half-life indicates the time taken for the measurement value to attain 50% of the final value; and the nine-tenths time indicates the time taken for the measurement value to attain 90% of the final value [3].

### 3.1. Response time without a protective tube



**Figure 8:** Transient response of an NTC, Pt 100 and type K thermocouple

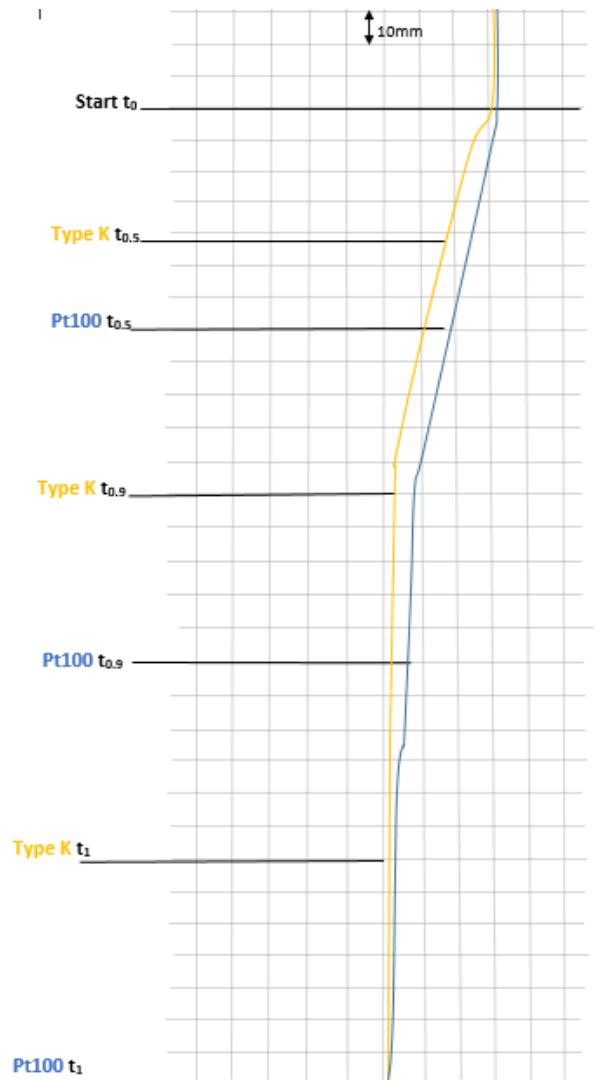
Pt 100; Initial temperature  $t_0=17.1\text{ }^{\circ}\text{C}$ . Half-life  $t_{0.5}=10.3\text{ s}$  (at  $30.2\text{ }^{\circ}\text{C}$ ). Nine-tenths life  $t_{0.9}=19.4\text{ s}$  (at  $42.5\text{ }^{\circ}\text{C}$ ) Final temperature  $t_1=46.1\text{ }^{\circ}\text{C}$  (after 39 s)

Type K thermocouple; Initial temperature  $t_0=16.0\text{ }^{\circ}\text{C}$  Half-life  $t_{0.5}=5.1\text{ s}$  (at  $33.2\text{ }^{\circ}\text{C}$ ) Nine-tenths life  $t_{0.9}=11.2\text{ s}$  (at  $44.1\text{ }^{\circ}\text{C}$ ) Final temperature  $t_1=42.6\text{ }^{\circ}\text{C}$  (after 33.4 s)

Thermistor NTC; Initial temperature  $t_0=\text{outside the measuring range}$ , Half-life  $t_{0.5}=25.2\text{ s}$  (at  $30.0\text{ }^{\circ}\text{C}$ ). Nine-tenths life  $t_{0.9}=87.1\text{ s}$  (at  $41.0\text{ }^{\circ}\text{C}$ ) Final temperature  $t_1=45.5\text{ }^{\circ}\text{C}$  (after 209 s)



### 3.2. Response time with a protective tube made of high-grade steel



**Figure 9:** Transient response of a Pt 100 and type K thermocouple with protective tubes made of high-grade steel

Pt 100; Initial temperature  $t_0=18.2\text{ }^{\circ}\text{C}$ . Half-life  $t_{0.5}=34.8\text{ s}$  (at  $36.0\text{ }^{\circ}\text{C}$ ). Nine-tenths life  $t_{0.9}=80.6\text{ s}$  (at  $42.3\text{ }^{\circ}\text{C}$ ). Final temperature  $t_1=47.8\text{ }^{\circ}\text{C}$  (after 145 s)

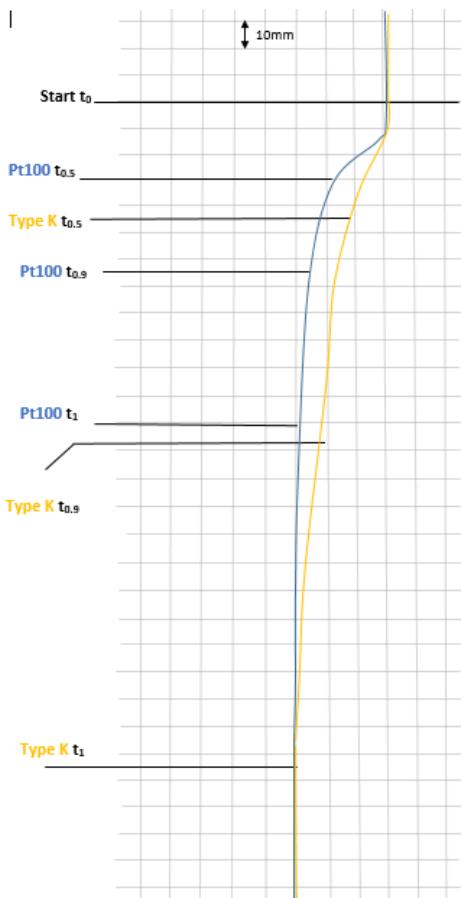
Type K thermocouple; Initial temperature  $t_0=18.0\text{ }^{\circ}\text{C}$ . Half-life  $t_{0.5}=26.5\text{ s}$  (at  $34.2\text{ }^{\circ}\text{C}$ ). Nine-tenths life  $t_{0.9}=59.5\text{ s}$  (at  $46.4\text{ }^{\circ}\text{C}$ ). Final temperature  $t_1=47.2\text{ }^{\circ}\text{C}$  (after 118 s)

### 3.3. Response time of the Pt 100 with an immersion tube made of brass

Pt 100; Initial temperature  $t_0=19.6\text{ }^{\circ}\text{C}$

Half-life  $t_{0.5}=15.4\text{ s}$  (at  $38.2\text{ }^{\circ}\text{C}$ ). Nine-tenths life  $t_{0.9}=30.2\text{ s}$  (at  $43.6\text{ }^{\circ}\text{C}$ ). Final temperature  $t_1=51.2\text{ }^{\circ}\text{C}$  (after 62.0 s).

Type K thermocouple; Initial temperature  $t_0=17.9\text{ }^{\circ}\text{C}$ . Half-life  $t_{0.5}=25.8\text{ s}$  (at  $32.9\text{ }^{\circ}\text{C}$ ). Nine-tenths life  $t_{0.9}=62.3\text{ s}$  (at  $46.2\text{ }^{\circ}\text{C}$ ). Final temperature  $t_1=56.2\text{ }^{\circ}\text{C}$  (after 124 s)



**Figure 10:** Transient response of a Pt 100 with a brass tube and a thermocouple with a high-grade steel immersion tube

#### 4. CONCLUSION

Response time without a protective tube can be seen as the measurement point of the thermocouple (the point at which the two conductive materials are welded together) has a low weight and high degree of contact, it responds more quickly than the Pt 100 sensor. The NTC thermistor responds much more slowly because its design (attachment to a relatively large tube) results in a comparatively extensive heat dissipation and high heat storage capacity. In contrast, the sensor itself warms up more slowly. Simultaneously, the use of immersion tubes considerably lengthens the response time. The response of the Pt 100 is much faster than that of the thermocouple. This is because the brass immersion tube has a much better thermal conductivity than the one made of high-grade steel.

#### REFERENCES

- [1] McMillan, K. G., Advanced Temperature Measurement and Control Second Edition, International Society of Automation, Research Triangle Park, NC, 2011.
- [2] Michalski, L., Eckersdorf, K., McGhee, J., Kucharski, J., Temperature Measurement Second Edition, John Wiley & Sons Ltd, UK, 2001.
- [3] <http://www.gunt.de/>

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/377841344>

# POWER QUALITY IMPROVEMENT OF LED STREET LIGHTING REFLECTORS

Conference Paper · January 2024

---

CITATIONS

0

READS

52

5 authors, including:



Dr Zeljko V Despotovic

Mihajlo Pupin Institute

661 PUBLICATIONS 888 CITATIONS

[SEE PROFILE](#)



Aleksandra Grujic

School of Electrical and Computing Engineering of Applied Studies

17 PUBLICATIONS 22 CITATIONS

[SEE PROFILE](#)



Ivana Vlajić Naumovska

Visoka Škola elektrotehnike i računarstva strukovnih studija u Beogradu

7 PUBLICATIONS 5 CITATIONS

[SEE PROFILE](#)

## **POWER QUALITY IMPROVEMENT OF LED STREET LIGHTING REFLECTORS**

Željko V. Despotović<sup>1</sup> Aleksandra Grujić<sup>2</sup> Ivana Vlajić Naumovska<sup>3</sup> Predrag Simić<sup>4</sup> Lara Laban<sup>5</sup>

**Abstract:** The main topic of this paper is the impact of LED lighting on the power quality, as well as the analysis of the most suitable methods for improving that quality. LED lighting with accompanying power electronics modules, regardless of individual LED lamps of relatively low power, is becoming an increasingly significant consumer of power, especially in systems where a large number of LED lamps or reflectors are used, such as street lighting or highway lighting systems. The simultaneous operation of a large number of such units in these systems can cause a significant impact on the power grid, which is reflected to the greatest extent by the generation of very unfavorable higher electrical current harmonics. The paper presents a concrete proposal for solving this problem using an power converter with implemented input power stage with power factor correction (PFC). Finally, the experimental results for one specific LED reflector for the cases without and with a power factor corrector are given. An adequate comparison of the obtained experimental results will also be performed.

**Key words:** power quality, LED light, power electronics, power converter, power factor, PFC

### **1. INTRODUCTION**

An important element of traffic, but also of the entire infrastructure of cities, is Light Emitting Diode (LED) lighting. Road lighting has a major impact on safety, whether for practical reasons related to visibility or for social security reasons. In addition to the optimal visibility it provides, LED lighting is also significant due to its low energy consumption and durability, while its maintenance is reduced to a minimum. The manufacturing quality of LED lamps and the reduced need for maintenance, in this case, become important factors of safety in traffic, but also allies in saving energy.

Public and street lighting primarily affect the quality of life by extending the duration of the day and allow people greater flexibility in organizing their own time and lifestyle. High-quality public lighting has a positive effect on the reduction of "light pollution", which for years has affected not only people, but also flora and fauna. By "light pollution" is meant the uncontrolled scattering of light, outside the intended area, which affects residential units, natural habitats of flora and fauna by the roads, and the effect of the so-called "dark sky" is lost, i.e. it becomes impossible to see the stars in a clear sky at night. Due to excessive use of artificial lighting and incorrect selection of lighting equipment, people's biorhythms may be disturbed, disorientation, depression and other diseases that affect the quality of life.

The speed of development and improvement of LED technology is such that all commercial lighting manufacturers are gradually "turning off" production lines of products with a classic light source and launching LED variants of products with superior technical characteristics, longer life and great savings in energy consumption[1].

Wider use of LED technology is supported by increasingly frequent guidelines and directives from the European Union (EU), which primarily relate to energy saving and energy efficiency, and the reasons for reducing "light pollution" are also not negligible. In response to these trends, efforts are being made

<sup>1</sup> PhD.E.E, Institute Mihajlo Pupin, University of Belgrade, Volgina 15, Belgrade, Serbia, email: [zeljko.despotovic@pupin.rs](mailto:zeljko.despotovic@pupin.rs)

<sup>2</sup> PhD.E.E, Academy of Technical and Art Applied Studies, School of Electrical and Computer Engineering, Vojvode Stepe 283, Belgrade, Serbia, [aleksandrag@viser.edu.rs](mailto:aleksandrag@viser.edu.rs)

<sup>3</sup> PhD.E.E, Academy of Technical and Art Applied Studies, School of Electrical and Computer Engineering, Vojvode Stepe 283, Belgrade, Serbia, [ivanav@viser.edu.rs](mailto:ivanav@viser.edu.rs)

<sup>4</sup> Master of applied study, JKP Javno osvetljenje, Ustanička 64, Belgrade, Serbia, [predrag.ssimic@bg-osvetljenje.rs](mailto:predrag.ssimic@bg-osvetljenje.rs)

<sup>5</sup> PhD student, Lund University, Dept. of automatic control, Box 117, 221 00 Lund, Sweden, [lara.laban@control.lth.se](mailto:lara.laban@control.lth.se)

to develop power electronics converters that offer complete LED driver solutions for LED street lights[1].

LED lighting is a good alternative to reduce energy consumption. A quality LED lamp or reflector must also have a quality DC power supply. The working life of LED lamps (reflectors) can be significantly increased by using a stable supply voltage by means of an adequate power converter. DC stable voltage for the LED reflector (usually 24Vdc, 36Vdc, 48Vdc) is most often obtained by combining an uncontrolled AC/DC converter (single-phase diode rectifier) to which a DC/DC converter is cascaded. DC/DC power converter stabilize the output DC voltage of LED reflector. This input rectifier significantly degrades the input power factor (< 0.7) and generates wide current harmonics spectrum, of which the third harmonic is dominant. Third harmonic creates additional problems in the power grid [2-3].

The impact becomes even more pronounced when there is a large number of LED reflectors in the system. This is exactly the case in practice in road lighting systems, public roads, street lighting, etc. With an adequate input rectifier, it is possible to significantly increase the input power factor (from 0.7 to 0.98 or even 1), and accordingly, the efficiency of the system power converter-LED reflector.

For the previously described reasons, this paper presents the realization and application of a grid power converter with power factor correction (PFC) in order to compensate reactive power and reduce higher harmonics, for each individual LED lamp (or LED reflector). In order to verify the implementation of PFC power rectifier, a laboratory set-up was realized where all the necessary measurements were performed. Also the behavior of the input power rectifier of specific LED reflector was compared for the case without and with PFC. After that, the comments of the obtained experimental results and the conclusions are given.

## 2. PROBLEMS DUE TO HIGHER HARMONICS IN LED LIGHTING SYSTEMS

Higher harmonics are one of the main parameters of the power quality, and they most often appear as a consequence of the operation of a large number of non-linear consumers, including LED lighting and the associated power converters. The non-linear nature of the voltage and current dependence of these consumers causes the distortion of the standard, sinusoidal voltage and current wave, and therefore more harmonics. For sensitive consumers, even the smallest changes in voltage and frequency can be the cause of reset, disconnection, incorrect operation or permanent damage.

Previous research has shown that LED lighting is a very sensitive consumer, and that even the slightest deviations in voltage and frequency from nominal values lead to noticeable disturbances and flickering that can be fatal for drivers and pedestrians on these roads. Special emphasis is placed on the switching character of the power electronic modules for powering LED reflectors, which causes the pulse character of the power, which is engaged from the power network[4].

The distorted current of uncompensated LED drivers (or groups of LED drivers) results in the degradation of the supply voltage waveform. Also, the harmonic spectrum of the current of uncompensated LED drivers is dominated by odd harmonics (third, fifth, seventh...).

Higher harmonics cause a variety of unwanted effects. These are computer malfunctions, protective equipment malfunctions, measurement errors, overvoltages, interference with communication and signaling devices, additional heating of electrical machines, etc. All these effects can have very serious technical and economic consequences, so today special attention is paid to the problem of eliminating higher harmonics that are injected into the power grid.

For the complete processing and analysis of higher harmonics, it is necessary to gain insight into various aspects of their appearance, influence, measurement (quantities, places and methods of measurement), methods of monitoring, standardization (standards, recommendations, instructions,

harmonization of regulations) and, finally, effective elimination of harmonics. All of these aspects of higher harmonics are covered under the overarching topic of electromagnetic compatibility (EMC).

The first standard for current harmonics, which appeared in 1982, was IEC 555. With the development of technology and new power systems, the standard underwent changes, and its current name is IEC 61000-3-2. It limits the harmonics of devices with an input current of up to 16 A, connected to 50 or 60 Hz, 220 or 240 V for single-phase circuits, as well as 380 V or 415 V for three-phase circuits. Besides the IEC 61000-3-2 standard, there are also other documents related to the management of current harmonics.

Standard IEC 61000-3-4 provides recommendations that apply to devices with a nominal current greater than 16 A, which are connected to a 50Hz or 60Hz network, with a nominal voltage of up to 240V<sub>eff</sub> (single-phase) or up to 440V<sub>eff</sub>(three-phase) (Table I).

*Table 1 – Permitted level of higher harmonics of the input current, standard IEC 61000-3-4*

Harmonic order (n)	$I_n[\%]$	Harmonic order (n)	$I_n[\%]$
3	21.6	21	$\leq 0.6$
5	10.7	23	0.9
7	7.2	25	0.8
9	3.8	27	$\leq 0.6$
11	3.1	29	0.7
13	2	31	0.7
15	0.7	33	$\leq 0.6$
17	1.2		
19	1.1	Even harm.	8/n or 0.6

### 3. REALIZED TOPOLOGY OF TWO STAGE POWER CONVERTER WITH PFC FOR SUPPLYING OF LED REFLECTOR

Two stage AC/DC power converter topology is the most popular topology for LED reflector drivers used above 100 W power. These topologies imply a high-voltage DC link circuit. In two-stage LED drivers, constant output current regulation is more easily implemented thanks to the DC link voltage. In addition, an isolated DC-DC stage with high frequency power transformer is required in the LED reflector driver to ensure constant current control and “flicker-free” LED lighting operation [5-6].

Two-stage power converter for LED lighting for higher powers consists of two power stages, i.e. the AC/DC power converter with implemented voltage booster and the high frequency DC/DC power converter with galvanic isolation by high frequency power transformer, as shown in Figure 1. The function of this two-stage power converter is to provide a constant current to the load (that is, the LED reflector). Figure 4(a) shows a simplified block diagram of a two-stage converter. The basic functional blocks are: (1) input mains EMI filter and rectifier, (2) PFC circuit, (3) power switching stage with MOSFETs, (4) high voltage high frequency (HVHF) step-down transformer T<sub>1</sub> for galvanic isolation, (5) low voltage rectifier with LC filter, (6) voltage and current measuring circuits and (7) control circuit (PWM + PFC) with over temperature, over currents and over voltage protections. The current and voltage measurements were made using galvanic isolation with suitable precise and linear opto-couplers having a wide bandwidth. Figure 4(b) shows more detailed views of the aforementioned basic functional blocks.

The input circuit EMI-filter is directly connected to the AC mains and must meet electromagnetic interference (EMI) requirements. The design of the EMI filter is based on measurements and analyses in scope of LED reflector driver conducted in the GHz frequency range.

This PFC rectifier circuit with corresponding PFC controller, provides active power factor correction, i.e. achieving a power factor close to one, at a very wide input voltage range of 85V-264V, 50(60)Hz. This part is actually an active rectifier that must meet the requirements of the IEC 61000-3-2 standard.

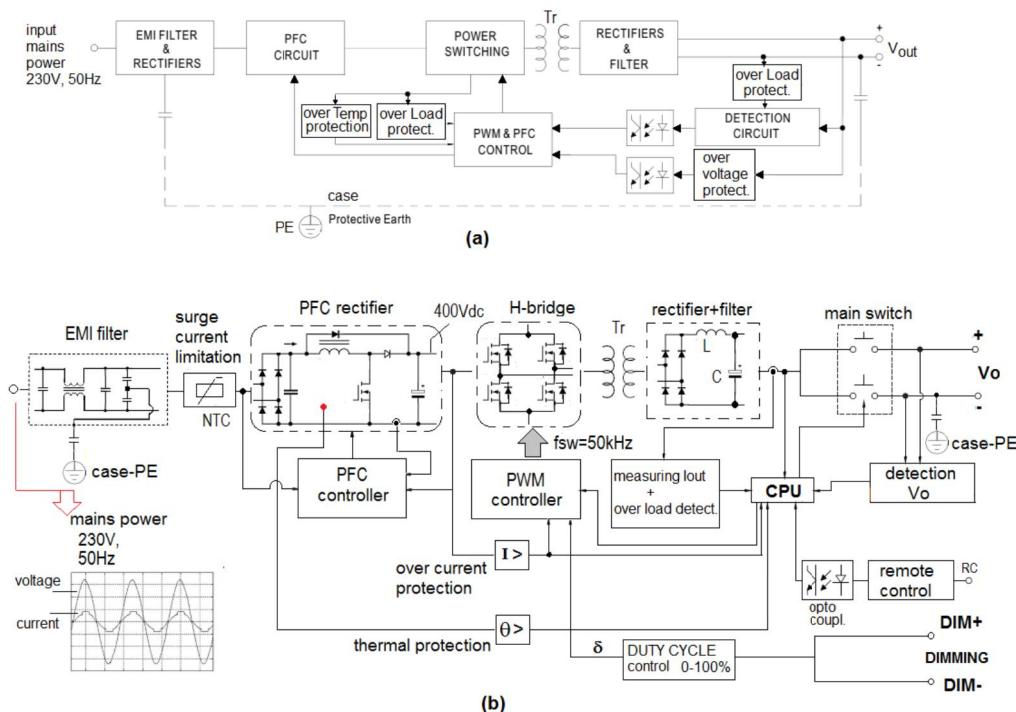


Figure 1 – Two stage power converter topology for supplying LED reflector; (a) basic functional block diagram, (b) detailed block scheme

The output DC voltage of this PFC stage is 400V. Between the EMI filter and the PFC rectifiers, there is an NTC resistor whose role is to limit the charging currents of the electrolytic capacitor at the output of the PFC circuit. Galvanic isolation between the input and output stages of the LED reflector driver is provided by the HVHF step-down power transformer  $T_r$  and the corresponding switching block, which is composed of an H-bridge consisting of four switching elements (MOSFET switches). The switching frequency for this part is 50 kHz. The control part of this H-bridge is realized with PWM controller. The duty-cycle control (0-100%) provides the function of dimming the LED lamp (inputs  $DIM(+)$  and  $DIM(-)$ ). The secondary voltage from the HVHF transformer leads to a full-wave diode rectifier with an LC filter, which provides a stabilized DC output voltage  $V_o$  to power the LED reflector. The two-stage power converter has possibility of galvanic isolated remote control  $RC$  via adequate opto-coupler. All current and voltage measurements are provided with adequate galvanic isolation. The following protections are implemented in the power converter: over current protection- $I >$ , thermal protection of PFC stage- $\theta >$ , over voltage protection of output voltage  $V_o$ - $U >$ , and over load protection at the output of the DC rectifier stage. At the output of LC filter of full-wave rectifier is located main switch for LED reflector.

#### 4. EXPERIMENTAL RESULTS

In order to verify the realized power converter with PFC, a laboratory set-up was made on which all the necessary measurements were performed and on which the behaviour of the LED reflector 36Vdc/100W was compared for the case with and without the power factor corrector. Figure 2 shows the electrical diagram of the experimental set-up, where the quantities measured during testing are highlighted in red.

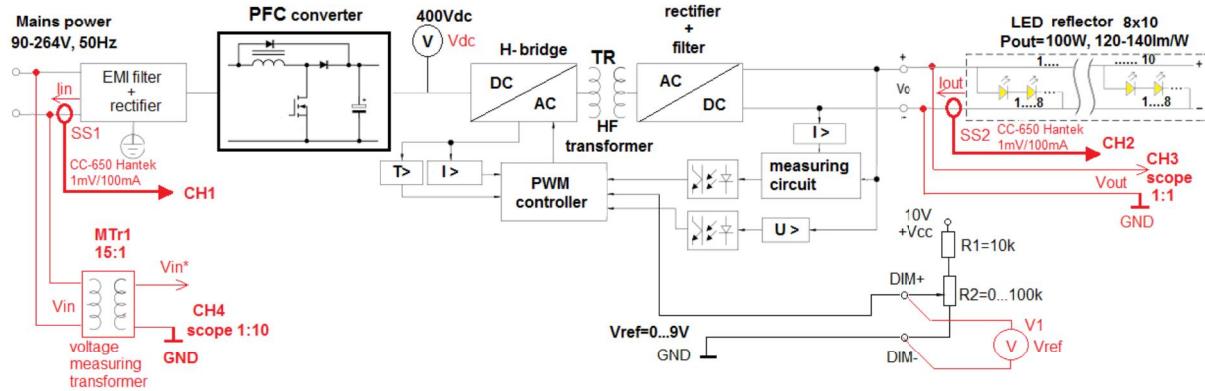


Figure 2 – Electrical diagram of the experimental set-up for the verification of the PFC converter for powering the LED reflector 36Vdc/100W

Changing the input mains voltage in the range 90-264V<sub>eff</sub>, 50Hz is provided with an auto-transformer built into the connecting laboratory panel. In this way, it is possible to test the PFC converter and the associated LED reflector at variable input voltage (this is a very common case in practice). In the concrete experimental verification, the input voltage of the PFC converter was changed in the range of 190V-240V, 50Hz. Testing was performed on an LED reflector (matrix 8x10 LED), with output power 100W, illumination 120-140 lm/W.

During the testing, measurements were made on a digital memory four-channel oscilloscope GWINSTEK MDO2204EX, bandwidth 200MHz, 1 GS/s. On the oscilloscope, the following values were measured for individual channels (CH1-CH4): CH1-mains current, CH2- LED reflector current, CH3-voltage of LED reflector, and CH4-mains voltage. As part of the experimental verification, measurements and tests were performed at the input mains voltage of the PFC converter of 190V, 50Hz and 240V, 50Hz.

Figure 3 shows characteristic oscilloscopic records for mains input voltage 190V<sub>rms</sub> and without PFC, at the reference value V<sub>ref</sub>= 7.2V. In this mode, the output voltage of the LED reflector was 30.6Vdc, while the output current was 2.68A, so the output power of the LED reflector was P<sub>out</sub>=82W.

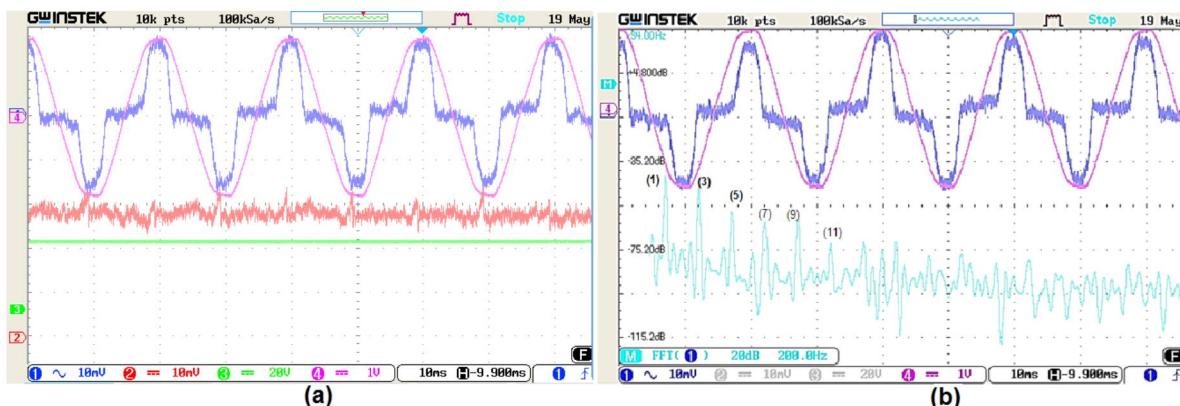


Figure 3-Oscilloscopic records for case mains voltage 190V and without PFC at V<sub>ref</sub>=7.2V;(a) CH1-mains current (1A/div), CH2-LED reflector current(1A/div), CH3- LED reflector voltage (20V/div), CH4-mains volatage (150V/div);time base 10ms/div, (b) Frequency spectar of input mains current

From this records on Figure 4(a), it can be seen that the input mains current is distorted and non-sinusoidal, which is expected considering that the input PFC converter was not connected. It is also observed that the mains current has an impulse value of 2A and a duration of 4ms. As a consequence of operation without PFC, the input current spectrum is obtained which is rich in higher odd harmonics

(3,5,7,9,11..., with frequencies 150Hz, 250Hz, 350Hz, 450Hz, 550Hz,...respectively), as shown the corresponding spectrum of the input current, which are given in Figure 3(b).

Figure 4 shows characteristic oscilloscopic records for mains input voltage 190V<sub>rms</sub> and with PFC, at the reference value V<sub>ref</sub>= 7V. From the oscilloscopic records on Figure 4(a), it can be seen that the mains voltage and the input current of the PFC converter are in phase and that the power factor is significantly better compared to the previously recorded cases (without PFC).

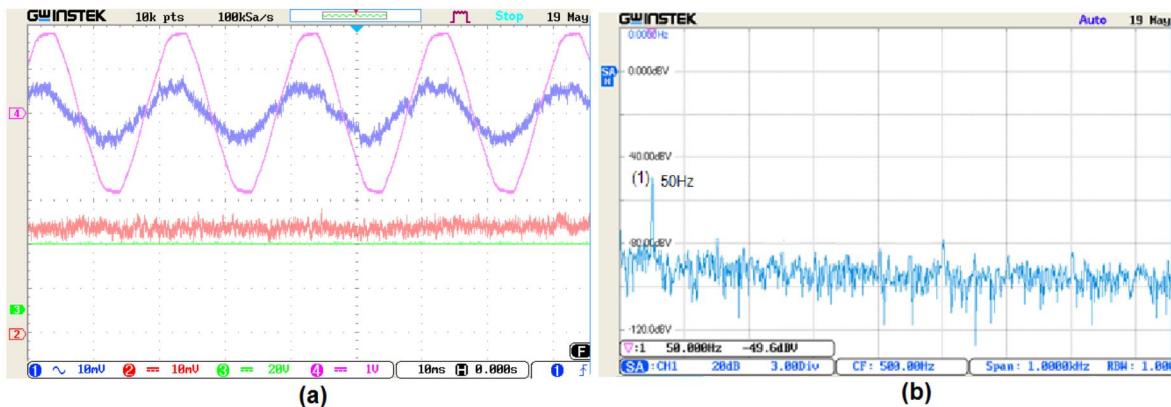


Figure 4- Oscilloscopic records for case mains voltage 190V and with PFC at Vref=7V;(a) CH1-mains current (1A/div), CH2-LED reflector current(1A/div), CH3- LED reflector voltage (20V/div), CH4- mains volatage (150V/div); time base 10ms/div, (b) Frequency spectar of input mains current

Under these conditions, the voltage of the LED reflector was 30V, while the output current was 2.4A, which gives the output power of the LED reflector of  $P_{out}=30V \cdot 2.4A=72W$ . Under these conditions, the effective value of the input mains current was 0.42A<sub>eff</sub>. At a power factor that is close to the unit value, the input power was  $P_{in}=79.8W$ . Taking into account the previous measurements, it is obtained that is efficiency of power converter  $\eta=72W/79.8W=0.90$ . The spectrum of the input current is given in Figure 4(b). In this case, it is observed that the first harmonic of frequency 50Hz is dominantly present in the spectrum of the input current, so in this case the harmonic spectrum is very favourable and the levels of higher harmonics are significantly reduced.

Figure 5 shows characteristic oscilloscopic records for mains input voltage 240V<sub>rms</sub> and with PFC, at the reference value V<sub>ref</sub>= 8.8V. In this case, full power is set on the LED reflector. Based on the displayed waveforms on Figure 5(b), it can be seen that the dominant harmonic of the input current is actually the fundamental harmonic of 50Hz, which is in phase with the input mains voltage.

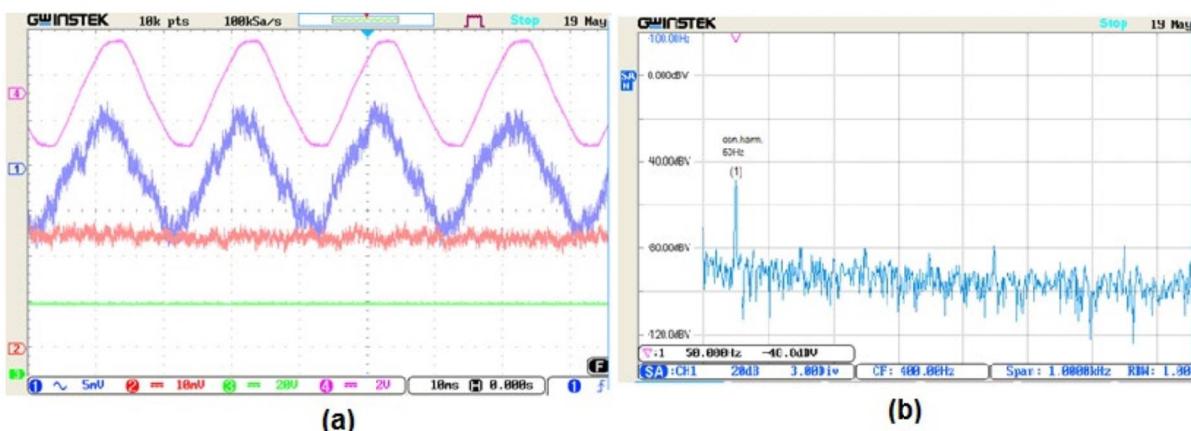


Figure 5- Oscilloscopic records for case mains voltage 240V and with PFC at Vref=8.8V;(a) CH1-mains current (0.4A/div), CH2-LED reflector current(1A/div), CH3- LED reflector voltage (20V/div), CH4- mains volatage (300V/div); time base 10ms/div, (b) Frequency spectar of input mains current

The current of the LED reflector for this case was 3A, while the voltage on the reflector was 33Vdc, so the output power was 99W. The effective value of the fundamental harmonic (dominant harmonic) of the current was  $0.45A_{\text{eff}}$ , which at an input voltage of  $240V_{\text{eff}}$  gives an input power value of 108W. Under these conditions, the efficiency was  $\eta = 99W/108W = 0.916$ .

The input current spectrum for this case is given in Figure 5(b). It is observed that the dominant harmonic of input current is the fundamental whose frequency is 50Hz. In this case, a very favourable spectrum of the input current is obtained, which is a consequence of the implementation of the designed PFC rectifier.

## 5. CONCLUSION

As part of this research, the implementation and experimental results of a two-stage power converter with PFC for supplying a typical 100W LED reflector used in the LED lighting system of one road are presented in detail. For the purpose of experimental verification, a laboratory setup was realized in which the implemented PFC converter was tested. As part of the experimental results, all quantities of interest (current and voltage) were recorded and a comparison of the power of the LED reflector with and without the PFC was performed. A spectral analysis of the obtained spectra of input mains current at different loads and at reduced input mains voltage ( $190V_{\text{eff}}$ ) and at increased input mains voltage ( $240V_{\text{eff}}$ ) was also performed. On the basis of oscilloscope recordings, it was shown that with LED power without PFC, the content of odd harmonics is significant, while in the case of LED power with PFC, the content of harmonics is significantly more favourable, i.e. the input current contains only the fundamental harmonic 50Hz, which is in phase with the input voltage.

This becomes more important when a large number of reflectors are used in the system (LED street lighting or road lighting) and when, collectively, the entire system actually behaves like a pure ohmic load to the mains, i.e. the total power factor is greater than 0.95, which is in accordance with all relevant standards and regulations in this area.

## 6. REFERENCES

- [1] J. Jin, H.; Jin, S.; Chen, L.; Cen, S.; Yuan, K., Research on the Lighting Performance of LED Street Lights with Different Color Temperatures. *IEEE Photonics J.* 2015, 7, 1–9.
- [2] A. Kusko; Power Quaity in Electrical System, 1st Edition, 2007 The McGraw-Hill Companies, Inc.
- [3] Wantuch, A.; Olesiak, M. Effect of LED Lighting on Selected Quality Parameters of Electricity. *Sensors* 2023, 23, 1582. <https://doi.org/10.3390/s23031582>
- [4] Lin, W.; Chen, H.; Ke, S. Research on a single-stage Flyback/boost LED driver with lower output ripple. In Proceedings of the 2016 IEEE 2nd Annual Southern Power Electronics Conference (SPEC), Auckland, New Zealand, 5–8 December 2016; pp. 1–5.
- [5] Wang, Y.; Guan, Y.; Xu, D.; Wang, W. A CLCL Resonant DC/DC Converter for Two-Stage LED Driver System. *IEEE Trans. Ind. Electron.* 2015, 63, 2883–2891.
- [6] Athalye, P.; Harris, M.; Negley, G. A two-stage LED driver for high-performance high-voltage LED fixtures. In Proceedings of the 2012 Twenty-Seventh Annual IEEE Applied Power Electronics Conference and Exposition (APEC), Orlando, FL, USA, 5–9 February 2012; pp. 2385–2391.

# Trajectory Tracking of a Two – Link Gripping Mechanism

Radiša Jovanović<sup>1\*</sup>, Uglješa Bugarić<sup>2</sup>, Lara Laban<sup>1</sup>, Mitra Vesović<sup>1</sup>

<sup>1</sup>Faculty of Mechanical Engineering/Department of Control Engineering, University of Belgrade, Serbia

<sup>2</sup>Faculty of Mechanical Engineering/Department of Industrial Engineering, University of Belgrade, Serbia

*The manufacturing industry frequently deals with the problem of gripping mechanism and their movement optimization. This paper presents an optimization methodology based on the whale optimization algorithm to design an optimal fuzzy PD controller of a two - link gripping mechanism (robot arm) as a part of mobile robot working cycle. The dynamical analysis of gripping mechanism investigates a coupling relation between the joint torques applied by the actuators and the position and acceleration of the robot arm. The proposed fuzzy controller optimizes the trajectory of the robot's end effector. Additionally, a simulation study was done for the specific initial case and the trapezoidal velocity profile was generated. Based on the predefined acceleration, movement of the robot arm is shown to be smooth and without an abrupt braking.*

**Keywords:** Gripping mechanism, Trajectory tracking, Fuzzy controller, Whale optimization algorithm

## 1. INTRODUCTION

Robots take part in an important role in the current manufacturing industry. Per se, an essential feature of the Industry 4.0 are the autonomous production methods powered by robots that can complete tasks intelligently, with a focus on safety, flexibility, versatility, and collaboration [1].

Intelligent mobile robots can be used for many different purposes, for example, in the production process, one of them is internal transport – material handling. In that case we can consider an intelligent mobile robot as sort of a transportation machine – device. The main characteristic of all transportation machines is their working cycle (single or complex). Moreover, we will be considering intelligent mobile robot as „a single position machine“ with a discontinue working regime. Single-position machine is a type of machine that will handle only one peace – product at a time and during handling it is on the machine the whole time [2].

Single working cycle of an intelligent mobile robot consists of: 1) Robot movement – from the starting point, to the position in front of the production machine, in reach of the gripping mechanism from where the transportation unit can be captured from the production machine; 2) Movement of the gripping mechanism - from starting (transport) position to the position needed for capturing the transportation unit; 3) Capturing of the transportation unit; 4) Reverse movement of the gripping mechanism with the transportation unit on it – from the position where the transportation unit is captured all the way to the starting (transport) position; 5) Reverse robot movement – from the position in front of the production machine to the starting point. 6) Activity 2), 3) and 4) are repeated, with the transportation unit releasing instead of capturing. However, if reverse robot movement is not finished at the same point i.e. starting point, then the working cycle is called complex.

Generally, the robot's motion in the environment is realized according to the predefined optimal path based on a defined criterion, and the current state of the robot is determined using preprocessed images obtained by a stereo vision system. In this paper, only a part of

intelligent robot working cycle, which refers to movement of the gripping mechanism, will be considered and optimized.

Many different techniques can be and are utilized to control the trajectory of the robot: traditional feedback controls (proportional integral derivative (PID) like controls), adaptive control, robust control, sliding mode control, optimal control, fuzzy control, and many others, as well as, a combinations of previous techniques.

Fuzzy logic controller (FLC) is only one of the intelligent controllers and represents a widespread control technique since it has a satisfactory performance for nonlinear and complex systems. The advantages of a fuzzy PID controller for trajectory tracking control of a mobile robot, and its gripping mechanism are paramount in its rapidity, stability, anti-interference and tracking precision [3-5]. The fuzzy PID controller can be designed with a trial-and-error approach and the optimization can be done by using the cross-entropy method [6]. The varying fuzzy PID and proportional-derivative (PD) controllers tend to use either the Mamdani or Takagi-Sugeno type of the fuzzy systems [7]. Implementation of metaheuristic algorithms can deal with nonconvex, nonlinear, and multimodal problems subjected to linear or nonlinear constraints with continuous or discrete decision variables as global optimization algorithms. Differential evolution and genetic algorithms have been utilized to conduct the optimum design of a fuzzy controller for mobile robot trajectory tracking [8-10]. A 2 DOF planar robot was controlled for a given trajectory where the parameters of Mamdani type FLC were tuned with the particle swarm optimization [11]. The genetic algorithm is applied to improve the performance of the PID controller in terms of control precision and speed of convergence in paper [13]. A fuzzy sliding mode tracking controller for robot manipulators with uncertainty in the kinematic and dynamic models is designed and analyzed in paper [14]. Further, a sliding mode controller, an adaptive fuzzy approximator, is designed in such way that it controls the position tracking of a robot manipulator with two degrees of freedom. Initially, by utilizing an inverse dynamic method, it reduces the uncertainties bound and finally,

\*Corresponding author: Kraljice Marije 16, Belgrade, Serbia and rjovanovic@mas.bg.ac.rs

sliding mode control eliminates the influence of the remaining uncertainties in closed-loop system stability [15]. In another paper a multiple-input multiple-output (MIMO) fuzzy logic unit was applied to the robot to track the desired trajectory with high accuracy. Moreover, in order to assess the performance of the proposed MIMO fuzzy sliding mode controller in the presence of parameter variations and external disturbances, a sudden load variation and noise were introduced to the robot system [16]. Feedback linearization controller is used to compute the required arm torque using the nonlinear feedback control law for a robotic manipulator with three degree of freedom. In addition, when all dynamic and physical parameters are known the FLC works remarkably, but given that a large amount of systems have uncertainties and the fuzzy FLC can reduce this kind of limitation [17]. However, various different approaches are included when combating the problem of the robotic arm, including the new methods using the neuro-fuzzy approach to estimate system uncertainties in paper [18].

The main goal of this paper is to design a fuzzy PD controller of a two-link gripping mechanism as a part of mobile robot working cycle. The whale optimization algorithm (WOA), as a novel optimization technique for solving optimization problems defined in [19], is used to determine the proper parameters of FLC in the trajectory tracking control of robot arm with two degrees of freedom (2-DOF).

## 2. DYNAMICS OF A TWO-LINK GRIPPER

Figure 1 shows the real object - mobile robot with gripping mechanism. As the desired task is to optimize the motion of this mechanism with two link and two degrees of freedom, it can be approximated with the scheme as



Figure 1: Real object: mobile robot with gripping mechanism

shown in Figure 2, where  $\theta_i$ , and  $m_i$  are respectively the link angle, the length and the mass of the  $i$ -th link,  $i = 1, 2$ . Without considering the friction and the disturbances, the dynamic model of a rigid two-link robot can be written as follows [20]:

$$M(\mathbf{q})\ddot{\mathbf{q}} + C(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + G(\mathbf{q}) = \boldsymbol{\tau} \quad (1)$$

where  $\mathbf{q}$ ,  $\dot{\mathbf{q}}$  and  $\ddot{\mathbf{q}} \in \mathbb{R}^{2 \times 1}$  are the robotic link position, the velocity and the acceleration vector, respectively;  $\boldsymbol{\tau} \in \mathbb{R}^{2 \times 1}$  is the torque input vector;  $M(\mathbf{q}) \in \mathbb{R}^{2 \times 2}$  is the positive definite inertia matrix;  $C(\mathbf{q}, \dot{\mathbf{q}}) \in \mathbb{R}^{2 \times 2}$  is the centripetal Coriolis force matrix; and  $G(\mathbf{q}) \in \mathbb{R}^{2 \times 1}$  is the gravitational vector.

Assuming that the centres of masses are in the middle of the levers, the elements  $M_{ij}(\mathbf{q})$  ( $i=1,2$ ) of the inertia matrix  $M(\mathbf{q})$  are as follows [20]:

$$\begin{aligned} M_{11} &= \frac{1}{3}m_1l_1^2 + \frac{1}{3}m_2l_2^2 + m_2l_1^2 + m_2l_1l_2 \cos q_2 \\ M_{12} = M_{21} &= \frac{1}{3}m_2l_2^2 + \frac{1}{2}m_2l_1l_2 \cos q_2, \quad M_{22} = \frac{1}{3}m_2l_2^2 \end{aligned} \quad (2)$$

In the case of robot from Figure 2,  $\mathbf{q}$  is the vector of angular displacements  $\theta_1$  and  $\theta_2$ ,  $\mathbf{q} = [\theta_1 \quad \theta_2]^T$ .

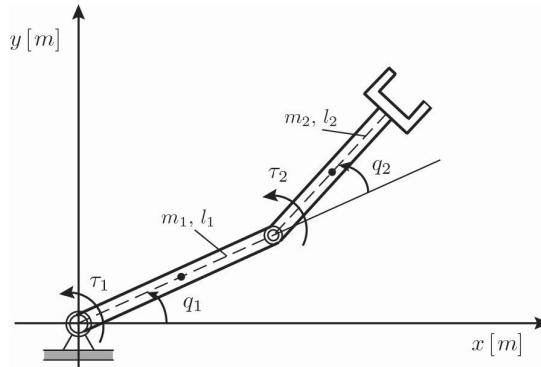


Figure 2: Scheme of the robotic gripper

The elements  $C_{ij}(\mathbf{q}, \dot{\mathbf{q}})$  ( $i, j = 1, 2$ ) of the matrix  $C(\mathbf{q}, \dot{\mathbf{q}})$  are presented as,

$$\begin{aligned} C_{11} &= -\frac{1}{2}m_2l_1l_2\dot{q}_2 \sin q_2 \\ C_{12} &= -\frac{1}{2}m_2l_1l_2 \sin q_2 (\dot{q}_1 + \dot{q}_2) \\ C_{21} &= -\frac{1}{2}m_2l_1l_2\dot{q}_1 \sin q_2, \quad C_{22} = 0 \end{aligned} \quad (3)$$

Finally, the elements of the gravitational torque vector  $G(\mathbf{q})$  are given by:

$$\begin{aligned} G_1 &= \left( \frac{1}{2}m_1l_1 + m_2l_1 \right)g \cos q_1 + \frac{1}{2}m_2l_2g \cos(q_1 + q_2) \\ G_2 &= \frac{1}{2}m_2l_2g \cos(q_1 + q_2) \end{aligned} \quad (4)$$

## 3. TRAJECTORY PLANING

In view of practical implementation, the trapezoidal velocity profile is one of the simplest motion profiles. It is composed of the ability to be accelerating to a constant velocity and decelerating to a rest state, and can therefore achieve fast motions. Its advantages are primarily that the time necessary to reach a constant speed is used and distributed so that the movement is smooth, without abrupt starting and stopping i.e. braking. The setting of this movement is actually done by setting the acceleration so that the speed decreases slightly until it reaches zero.

According to the time, the profile divides into three regions and outputs: the maximum acceleration, deceleration, or zero value as acceleration. As shown in Figure 3, in the constant acceleration region the acceleration is the maximum positive value  $\ddot{q}_{\max}$  until the velocity reaches the maximum value,  $\dot{q}_{\max}$ . After that the constant velocity region where the acceleration and

velocity are zero, and the maximum value as well, respectively, the velocity decreases to a zero with the maximum deceleration,  $-\ddot{q}_{\max}$ .

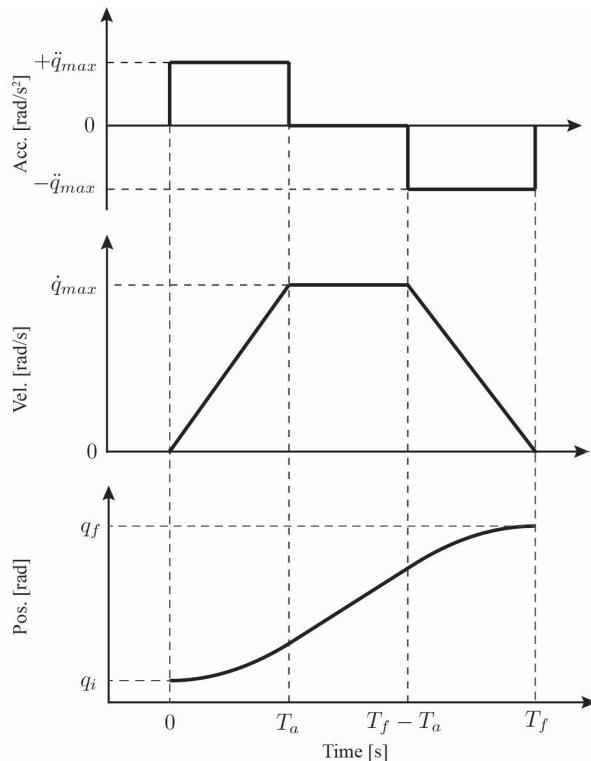


Figure 3: The trapezoidal velocity profile: the acceleration, velocity and position profiles

Using the following important parameters: the initial  $q_i$  and the final  $q_f$ , the total duration of the movement  $T_f$  and the time provided for acceleration  $T_a$ , the acceleration, velocity and position profiles can be described as:

$$\ddot{q}(t) = \begin{cases} +\ddot{q}_{\max} & 0 < t \leq T_a \\ 0 & T_a < t \leq T_f - T_a \\ -\ddot{q}_{\max} & T_f - T_a < t \leq T_f \end{cases} \quad (5)$$

$$\dot{q}(t) = \begin{cases} \ddot{q}_{\max} \cdot t & 0 < t \leq T_a \\ \ddot{q}_{\max} \cdot T_a & T_a < t \leq T_f - T_a \\ \ddot{q}_{\max} \cdot (T_f - t) & T_f - T_a < t \leq T_f \end{cases} \quad (6)$$

$$\begin{aligned} \dot{q}(t) = & \begin{cases} q_i + 0.5 \cdot \ddot{q}_{\max} \cdot t^2 & 0 < t \leq T_a \\ q_i + \ddot{q}_{\max} \cdot T_a \left( t - \frac{T_a}{2} \right) & T_a < t \leq T_f - T_a \\ q_f - 0.5 \ddot{q}_{\max} \cdot (T_f - t)^2 & T_f - T_a < t \leq T_f \end{cases} \quad (7) \end{aligned}$$

Clearly, from the acceleration output, the velocity and position profiles are generated by integration operations, while taking into account the initial conditions.

Maximum speed, maximum acceleration and the time provided for acceleration have the following relationship:

$$\dot{q}_{\max} = \ddot{q}_{\max} T_a \quad (8)$$

Using the following equation:

$$q_f - q_i = (\ddot{q}_{\max} \cdot T_a) \cdot (T_f - T_a) \quad (9)$$

the only unrevealed variable (maxima acceleration  $\ddot{q}_{\max}$ ) can be obtained:

$$\ddot{q}_{\max} = \frac{q_f - q_i}{T_a (T_f - T_a)} \quad (10)$$

In all previous expressions it is logically assumed that  $T_a \leq T_f / 2$ , that is, the acceleration period is shorter than half of the total time. In the case of equality, the shape of the function becomes a triangle [21].

#### 4. FUZZY LOGIC CONTROLLER

In the following section, we will be using the fuzzy control technique in order to design a fuzzy controller which is able to move a two link robot to track a desired trajectory. Consequentially, we will be designing two fuzzy controllers, one for each separate link. Some of the essential elements when designing a fuzzy controller include, first and foremost, defining the input and output variables, secondly the choice of fuzzification and defuzzification process, and most importantly determining the rule-base of the controller.

In this paper, a proportional derivative (PD) type of FLC is utilized. The inputs of this type of controller are the error and the change in error, whilst the output is the control signal. Nevertheless, in the considered robot trajectory control, the input variables of the FLC are the error and error derivation of link position. The output variable of the fuzzy controller is the link control input, i.e. torque. All membership functions for the controller inputs and outputs are defined on the common normalized interval  $[-1, 1]$ . For all of the membership functions we use symmetric triangular functions (except for the two membership functions at the ends, which are trapezoidal) with an equal base and 50% overlap with neighbouring membership functions as shown in Figure 4 and Figure 5.

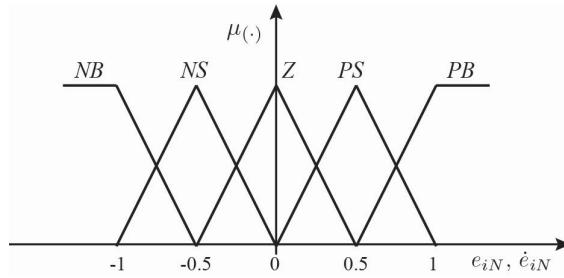


Figure 4: The input membership functions

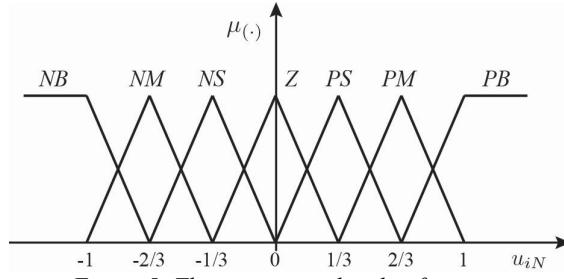


Figure 5: The output membership functions

Further, fuzzy controllers both of the links share a common membership function, where  $e_{iN}$  and  $\dot{e}_{iN}$ , for all  $i = 1, 2, \dots$  represent the normalized error and the normalized derivative of the error, respectively. The normalized control signals are represented by  $u_{iN}$ , for all  $i = 1, 2$  respectively, for the link 1 and link 2.

Table 1: Fuzzy IF-THEN rules for the robot trajectory control

$e_N \backslash \dot{e}_N$	NB	NS	Z	PS	PB
NB	NB	NB	NM	NS	Z
NS	NB	NM	NS	Z	PS
Z	NM	NS	Z	PS	PM
PS	NS	Z	PS	PM	PB
PB	Z	PS	PM	PB	PB

In a standard fuzzy partition, each fuzzy set determines the value of the linguistic variable. The fuzzy linguistic variables NB, NM, NS, Z, PS, PM and PB

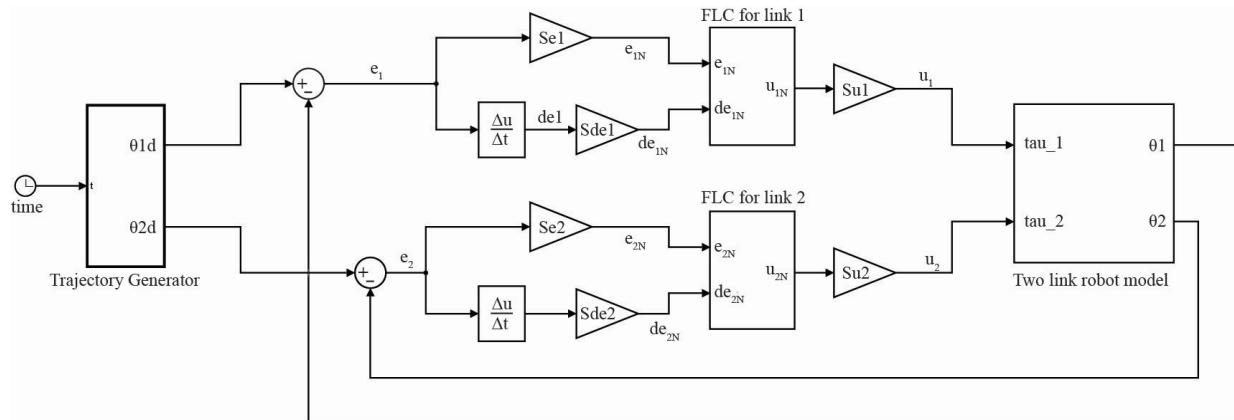


Figure 6: Simulink model of the 2-DOF gripping mechanism with fuzzy control

## 5. OPTIMIZATION OF FLC

### 5.1. Whale optimization algorithm

The highly utilised and implemented WOA was first suggested by Seyedali Mirjalili and Andrew Lewis in their paper [19]. The WOA has proven to be outstanding at resolving a variety of modes, multimodal and problems that are not linear. The foremost supremacies of this algorithm, and all metaheuristic algorithms in general, are that it has random distribution, which avoids getting stuck in the local minimum. The hunting method which they deeply rely on is the bubble-net feeding method. Using this method they dive deep, a couple of meters deep into the ocean and then start swimming upwards to the surface creating a bubbles in a spiral shape while encircling the prey.

#### 5.1.1. Encircling the prey

Since the whales can recognize the location of the prey, the WOA algorithm assumes that the current best solution is the target prey, or very close to it. Stressing this, after the best search agent is defined the other search

represent the negative big, negative medium, negative small, zero, positive small, positive medium and positive big values. Hence, the fuzzy IF-THEN rules for the robot trajectory control are given in Table 1.

In addition, the use of normalized domains requires a scale transformation, i.e. input normalization, which maps the physical values of the input variables into a normalized domain. Furthermore, output denormalization maps the normalized value of the control output variable into its respective physical domain. In stating the above, the relationships between scaling factors and the input and output variables are as follows:

$$e_{iN} = S_{e_i} \cdot e_i, \quad \dot{e}_{iN} = S_{de_i} \cdot \dot{e}_i, \quad u_i = S_{u_i} \cdot u_{iN}, \quad i=1,2 \quad (11)$$

where  $e_i$ ,  $\dot{e}_i$  and  $u_i$  are error, the derivative error and control input, respectively. Practically, the proposed FLCs are implemented in Matlab/Simulink, with the product inference engine and center average defuzzification method. Simulink model of the two link robot system with fuzzy control is shown in Figure 6.

agents will try to update their position towards it. The mathematical model of encircling the prey is proposed using the following equations (where  $\mathbf{D}$  is the distance vector and  $\mathbf{X}$  is the vector utilized to update the position):

$$\mathbf{D} = |\mathbf{C}\mathbf{X}^*(t) - \mathbf{X}(t)| \quad (12)$$

$$\mathbf{X}(t+1) = \mathbf{X}^*(t) - \mathbf{A}\mathbf{D} \quad (13)$$

where  $t$  indicates the current iteration,  $\mathbf{A}$  and  $\mathbf{C}$  are coefficient vectors,  $\mathbf{X}^*$  is the position vector of the best solution obtained so far,  $\mathbf{X}$  is the position vector [19].

#### 5.1.2. Exploitation phase: Hunting using the Bubble-net method

Moreover when stating the mathematical modeling of the exploitation phase it is found that there are two diverse approaches to it; the shrinking encircling mechanism and spiral updating position. An universal assumption suggests that there is a 50% chance that the whale will choose between one of these two approaches, when updating the position. The first approach is related to decreasing linearly the value of  $a$  from 2 to 0 over the

course of iterations. Hence, the random values for  $\mathbf{A}$  are between  $[-1,1]$ , where the new position of the agent is located between the current best agent and the original position. The second approach is based on the calculations of the prey and the whale.

The mathematical model for these approaches is henceforward, depicted as the following equation:

$$\mathbf{X}(t+1) = \begin{cases} \mathbf{X}^*(t) - \mathbf{AD} & \text{if } p < 0.5 \\ \mathbf{D} \cdot e^{bl} \cdot \cos(2\pi l) + \mathbf{X}^*(t) & \text{if } p > 0.5 \end{cases} \quad (14)$$

where  $p$  is a random number in  $[0,1]$ ,  $b$  is a constant for defining the shape of the logarithmic spiral,  $l$  is a random number in  $[-1,1]$  and  $\mathbf{D} = |\mathbf{X}^*(t) - \mathbf{X}(t)|$  indicates the distance of the  $i$ -th whale from the prey [19].

### 5.1.3. Exploitation phase: Search for the prey

The whales search for the prey randomly according to the position of each others locations. The parameter  $\mathbf{A}$  is used to force the search agent to move far away from a reference whale. The position update here is based on the position of a randomly chosen search agent, instead of the best agent so far. The mechanism and  $|\mathbf{A}| > 1$  emphasizes the exploration and allows the WOA algorithm to perform a global search, henceforward it favors exploration and local optima avoidance. In other words the agent is diverging and moving away from the prey, unlike the converging and the best solution selection when using  $|\mathbf{A}| < 1$  [19]. The mathematical model is as ensuing:

$$\mathbf{D} = |\mathbf{CX}_{rand} - \mathbf{X}| \quad (15)$$

$$\mathbf{X}(t+1) = \mathbf{X}_{rand} - \mathbf{AD} \quad (16)$$

### 5.2. Optimization of FLC using the whale optimization algorithm

In a general sense, fuzzy controllers have a large number of parameters that can be adjusted in an attempt to gain an optimal dynamical response. Those parameters are the shape of the membership functions, the number of the linguistic variables for input and output values of the set of rules, scaling factors, etc. Moreover, in using the predetermined membership functions for the input and output values defined in section 4, and on Figure 4 and Figure 5, as well as, the set of rules (Table 1), it becomes obvious that the performance of the fuzzy PD controller depends on the input and output scaling factors, in turn the design of the fuzzy controller can be simply be attributed to the choice of the input/output scaling factors.

In this paper we have been focused only on the tuning of the scaling factors, considering that is correspondent to the gains of the controller. Further, for the design of the optimal fuzzy PD controller the WOA optimization algorithm was used. Moreover, the mentioned parameters are all coded into one whale, i.e. one agent, that is presented with a vector which contains, in our case, six parameters. For the objective function we utilized the algebraic sum of ITAE (integral of time-weighted absolute error) performance criterion of both links, as defined in the ensuing equation:

$$J = \int_0^\infty t \cdot [ |e_1(t)| + |e_2(t)| ] dt \quad (17)$$

## 6. EXPERIMENTAL RESULTS

Finally, in order to demonstrate the methodology previously discussed, a 2-DOF robot, that is depicted in Figure 2, is used in order to perform the following simulation. The physical parameters for the gripping mechanism are  $m_1 = 0.00799 \text{ kg}$ ,  $m_2 = 0.00521 \text{ kg}$ ,  $l_1 = 0.05831 \text{ m}$  and  $l_2 = 0.0422 \text{ m}$ .

The desired end-effector trajectory of the 2-DOF manipulator is specified according to trapezoidal velocity profile defined in Section 3. The initial position of the robotic gripper is determined by the mechanism itself. In our case, the initial link configuration is defined as  $q_0 = [1.3963 \ -0.5236]^T \text{ rad}$  and lastly, the initial end-effector position is  $x_0 = 0.0373 \text{ m}$ ,  $y_0 = 0.0898 \text{ m}$ .

The control task is to move that point from its initials to the finals coordinates defined by angles  $q_f = [0.7854 \ -0.7854]^T \text{ rad}$ , and the end-effector position  $x_f = 0.0834 \text{ m}$ ,  $y_f = 0.0412 \text{ m}$ .

The time required to reach this position is set to be  $T_f = 6 \text{ s}$ , the maximum acceleration  $\ddot{q}_{\max}$  and the time provided for acceleration  $T_a$  are calculated based on the equation (10) and the following,

$$T_a = \frac{T_f}{3} = 2 \text{ s}, \quad |\ddot{q}_{\max,1}| = 0.0763 \frac{\text{m}}{\text{s}^2}, \quad |\ddot{q}_{\max,2}| = 0.0330 \frac{\text{m}}{\text{s}^2}$$

In the proposed WOA algorithm the population is set to 10, while the total number of iterations is set to 30. Furthermore, in this optimization method, one agent represents one potential optimal fuzzy controller. All of the parameter values that were used in the implementation of the WOA were taken from the original paper [19]. The convergence curve of the objective function value is depicted in Figure 7.

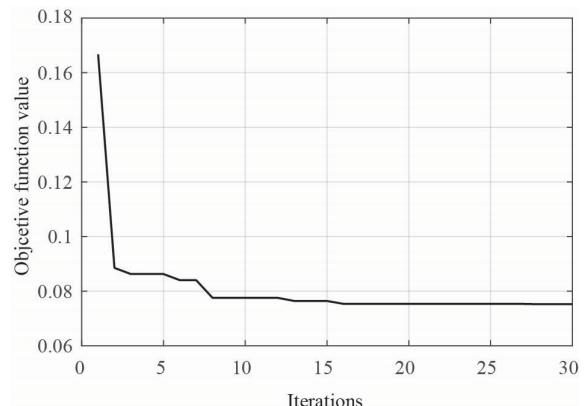


Figure 7: The convergence curve of the objective function value

In addition, after the optimization the obtained parameters for the scaling factors are:

$$S_{e_1} = 1.8505, \quad S_{de_1} = 0.0784, \quad S_{u_1} = 0.9218$$

$$S_{e_2} = 1.5, \quad S_{de_2} = 0.0025, \quad S_{u_2} = 1.4385$$

In the following two pictures, we have shown the comparison between the real trajectory and the desired trajectory of the link 1 (Figure 8) and link 2 (Figure 9).

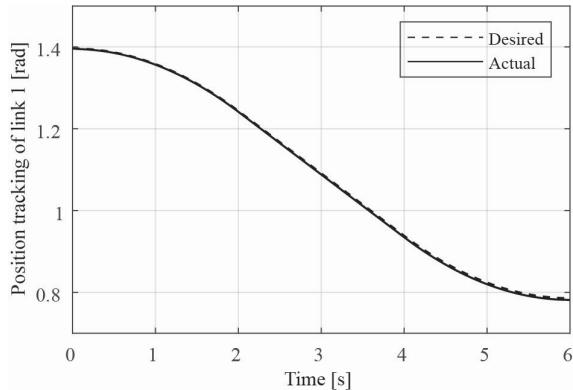


Figure 8: A comparison between the desired and real trajectory of link 1

There we can also observe that the real and desired trajectory curves both almost match, with very slight deviations, nearly neglectable.

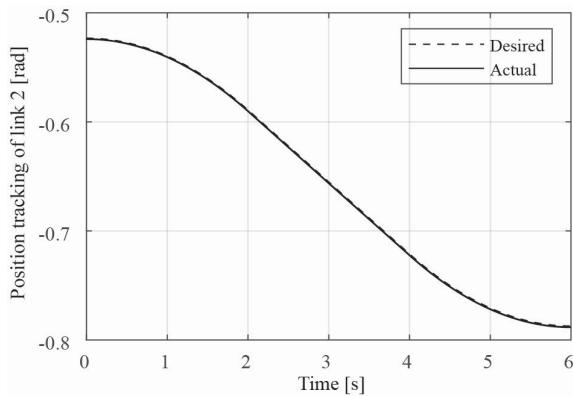


Figure 9: A comparison between the desired and real trajectory of link 2

Moreover, the errors of position tracking for link 1 and link 2, are given in Figure 10 and Figure 11, respectively.

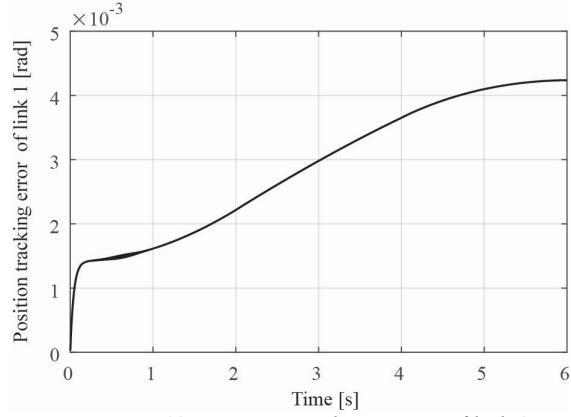


Figure 10: Position tracking error of link 1

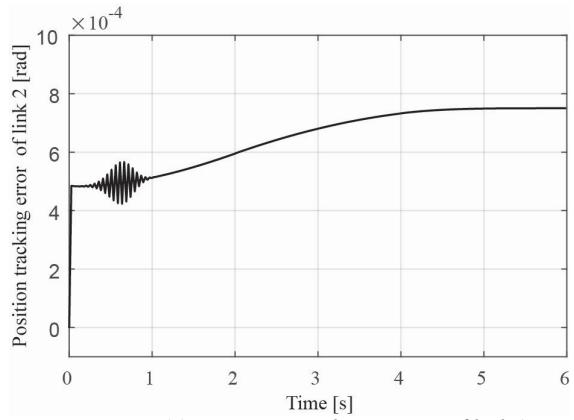


Figure 11: Position tracking error of link 2

The error for the position tracking of the first link is less than 0.005 rad, while for the second link it is less than 0.001 rad. Finally, in Figure 12 and Figure 13 we have depicted the control torque of both link 1 and link 2.

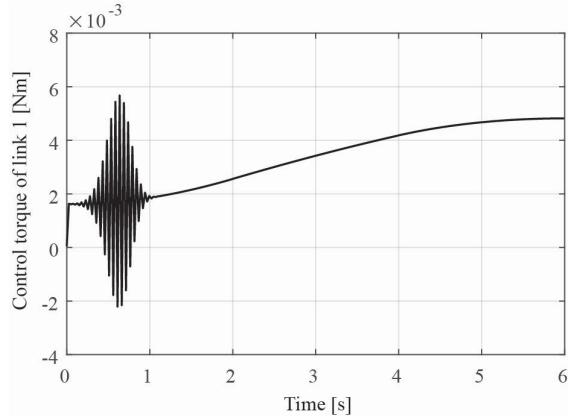


Figure 12: Control torque of link 1

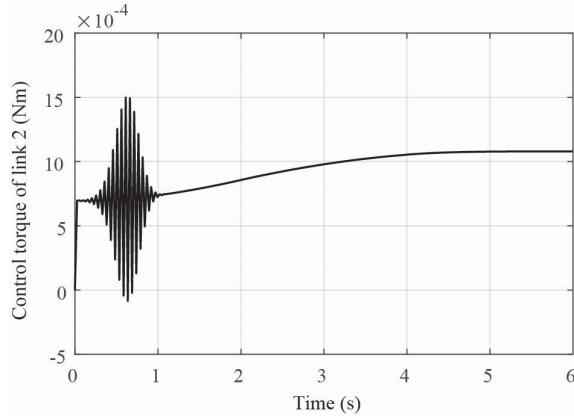


Figure 13: Control torque of link 2

Furthermore, the robustness of the designed fuzzy controllers is tested, as follows. Hence, in order to test the robustness, the mass of the robotic arm of each link is increased three times, as in [11].

The parameters of the fuzzy controller, which were optimized using the WOA algorithm, have remained unchanged, and comparisons of real and desired trajectories are given in Figure 14 and Figure 15.

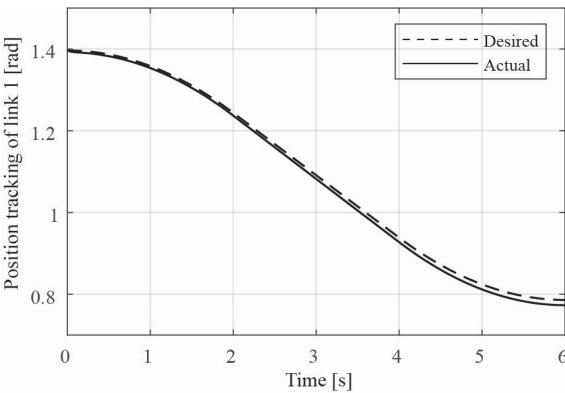


Figure 14: A comparison between the desired and real trajectory of link 1 (increased mass of links three times)

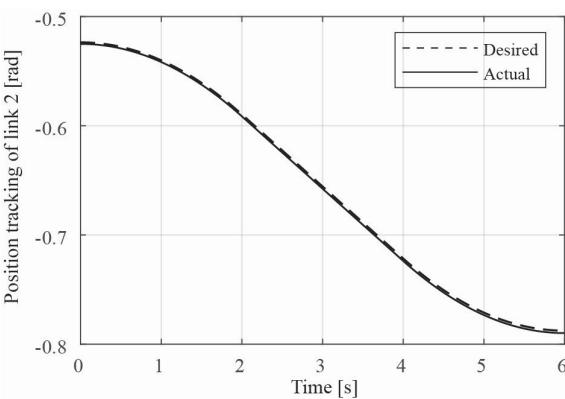


Figure 15: A comparison between the desired and real trajectory of link 2 (increased mass of links three times)

In the pictures above we can clearly notice that even though we enlarged the mass of link 1 and link 2, the optimized algorithm works exceptionally well.

In addition, the errors of position tracking for link 1 and link 2, where the given links have a mass that has been increased three times in order to test the robustness, are given in Figure 16 and Figure 17, respectively.

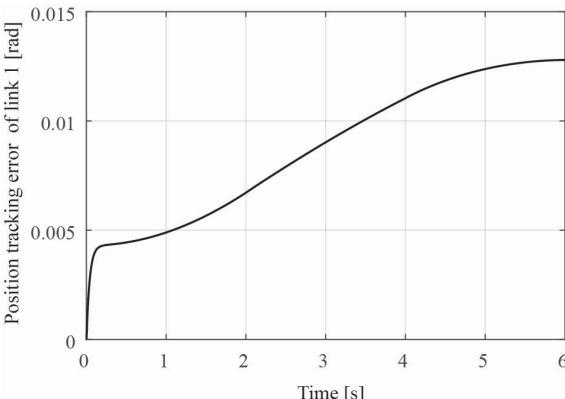


Figure 16: Position tracking error of link 1 (increased mass of links three times)

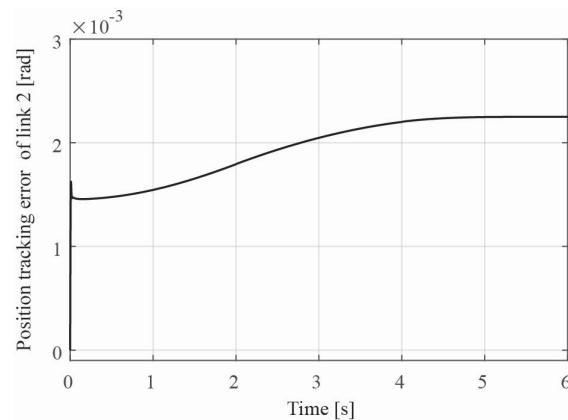


Figure 17: Position tracking error of link 2 (increased mass of links three times)

Here the error for the position tracking of the first link is about 0.015 rad, while for the second link it is about 0.001 rad.

## 7. CONCLUSION

In this paper, fuzzy controllers were proposed for the trajectory tracking control of a two-link gripping mechanism as a part of mobile robot working cycle. The whale optimization algorithm was used to optimize the scaling factors of the proposed fuzzy PD controller. Namely, optimal input/output gains for the fuzzy PD controller were generated according to ITAE performance criterion. Numerical simulations were done to analyze the trajectory tracking performance of the designed controller. Moreover, the robustness of the controllers was tested in the case of the mass changes. The simulation results have shown that the proposed controller was capable of dealing with the nonlinearities of the robot and the changing of its parameters. One possible area of the future work can be simultaneous optimization of the scaling factors as well as parameters of input and output membership functions.

## ACKNOWLEDGEMENTS

This research was supported by the Science Fund of the Republic of Serbia, grant No. 6523109, AIMISSION4.0, 2020-2022.

This work was financially supported by the Ministry of Education, Science and Technological Development of the Serbian Government, MPNTR RS under contract 451-03-9/2021-14/200105, Grant TR-35029 (2021).

## REFERENCES

- [1] M.A.K. Bahrin, M.F. Othman, N.H.N. Azli and M.F. Talib, "Industry 4.0: A review on industrial automation and robotic," Journal Teknologi, Vol. 8(6-13), pp. 137–143, (2016)
- [2] U. Bugarić, D. Popović and S. Tošić, "Methodology for analysis of working cycle," XVII International Conference on "Material Flow, Machines and Devices in Industry", Belgrade (Serbia), pp. 4–5, (2002)
- [3] Q. Xu, J. Kan, S. Chen and S. Yan, "Fuzzy PID Based Trajectory Tracking Control of Mobile Robot and its Simulation in Simulink," International Journal of Control and Automation, Vol. 7(8), pp. 233–244, (2014)

- [4] M. Tong, W. Lin, X. Huo, Z. Jin and C. Miao, "A model-free fuzzy adaptive trajectory tracking control algorithm based on dynamic surface control," *International Journal of Advanced Robotic Systems*, Vol. 17(1), pp. 1–11, (2020)
- [5] M. S. Abood, I. K. Thajeel, E. M. Alsaedi, M. M. Hamdi, A. S. Mustafa and S. A. Rashid, "Fuzzy Logic Controller to control the position of a mobile robot that follows a track on the floor," *4th International Symposium on Multidisciplinary Studies and Innovative Technologies*, Istanbul (Turkey), 22-24 October 2020, pp. 1–7, (2020)
- [6] T.A. Mai, T.S. Dang, D.T. Duong, V.C. Le and S. Banerjee, "A combined backstepping and adaptive fuzzy PID approach for trajectory tracking of autonomous mobile robots," *Journal of the Brazilian Society of Mechanical Sciences and Engineering*, Vol. 43, (2021)
- [7] B. Jiang, H. R. Karimi, S. Yang, C. Gao and Y. Kao, "Observer-Based Adaptive Sliding Mode Control for Nonlinear Stochastic Markov Jump Systems via T-S Fuzzy Modeling: Applications to Robot Arm Model," *IEEE Transactions on Industrial Electronics*, Vol. 68(1), pp. 466–477 (2021)
- [8] H. Nejat Pishkenari, S.H. Mahboobi and A. Alasty, "Optimum synthesis of fuzzy logic controller for trajectory tracking by differential evolution," *Scientia Iranica*, Vol. 18 (2), pp. 261–267, (2011)
- [9] R.M. da Silva, M.A de Souza Leite Cuadros and D.F.T. Gamarra, "Comparison of a Backstepping and a Fuzzy Controller for Tracking a Trajectory with a Mobile Robot," *Intelligent Systems Design and Applications*, Vol. 941, pp. 212–221, (2019)
- [10] J. Zhao, L. Han, L. Wang and Z. Yu, "The fuzzy PID control optimized by genetic algorithm for trajectory tracking of robot arm," *12th World Congress on Intelligent Control and Automation*, Guilin (China), 12 June-15 June 2016, pp. 556–559, (2016)
- [11] Z. Bingul and O. Karahan, "A Fuzzy Logic Controller tuned with PSO for 2 DOF robot trajectory control," *Expert Systems with Applications* Vol. 38, pp. 1017–1031, (2011)
- [12] H. Feng, Cb. Yin, Ww. Weng, W. Ma, Jj. Zhou, Wh. Jia and Zl. Zhang, "Robotic excavator trajectory control using an improved GA based PID controller," *Mechanical Systems and Signal Processing*, Vol. 105, pp. 153–168 (2018)
- [13] A. Alouache and Q. Wu, "Genetic Algorithms for Trajectory Tracking of Mobile Robot Based on PID Controller," *14th International Conference on Intelligent Computer Communication and Processing*, Cluj-Napoca (Romania), 6-8 September 2018, pp. 237–241, (2018)
- [14] F. Piltan, A.R. Nabaee, M.M. Ebrahimi and M. Bazregar, "Design Robust Fuzzy Sliding Mode Control Technique for Robot Manipulator Systems with Modeling Uncertainties," *International Journal Information Technology and Computer Science*, Vol. 8, pp. 123–135, (2013)
- [15] M.R. Soltanpour, P. Otadolajam and M.H. Khooban, "Robust control strategy for electrically driven robot manipulators: adaptive fuzzy sliding mode," *IET science measurement and technology*, Vol. 9 (3), pp. 322–334, (2015)
- [16] Y. Hacioglu, Y.Z. Arslan and N. Yagiz, "MIMO fuzzy sliding mode controlled dual arm robot in load transportation," *Journal of the Franklin Institute*, Vol. 348 (8), pp. 1886–1902, (2011)
- [17] F. Piltan, M.H. Yarmahmoudi, M. Mirzaie, S. Emamzadeh and Z. Hivand, "Design Novel Fuzzy Robust Feedback Linearization Control with Application to Robot Manipulator," *International Journal of Intelligent Systems and Applications*, Vol. 5(5), pp. 1–10, (2013)
- [18] D.C. Theodoridis, Y. Boutalis and M.A. Christodoulou, "A New Adaptive Neuro-Fuzzy Controller for Trajectory Tracking of Robot Manipulators," *International Journal of Robotics and Automation*, Vol. 26(1), pp. 64–75, (2011)
- [19] S. Mirjalili and A. Lewis, "The Whale Optimization Algorithm," *Advances in engineering software*, Vol. 95, pp. 51–67, (2016)
- [20] L. Sciavicco and B. Siciliano, "Modeling and control of robot manipulators," Springer-Verlag, London (England), (2000)
- [21] [https://www.ucg.ac.me/skladiste/blog\\_13269/objava\\_8796/fajlovi/35\\_industrijska781.pdf](https://www.ucg.ac.me/skladiste/blog_13269/objava_8796/fajlovi/35_industrijska781.pdf), University of Montenegro, pp. 87-102, (last accessed 30/05/21)

# Feedback Linearization Control of a Two – Link Gripping Mechanism

Mitra Vesović<sup>1\*</sup>, Radiša Jovanović<sup>1</sup>, Lara Laban<sup>1</sup>, Uglješa Bugarić<sup>2</sup>

<sup>1</sup>Faculty of Mechanical Engineering/Department of Control Engineering, University of Belgrade, Serbia

<sup>2</sup> Faculty of Mechanical Engineering/Department of Industrial Engineering, University of Belgrade, Serbia

*This paper presents a feedback linearization controller for trajectory tracking of two degrees of freedom (2DOF) gripping mechanism. To reach this goal, after deriving the dynamical equations of the gripping mechanism, the feedback linearization approach is utilized to change the nonlinear dynamics to a linear one. Classical proportional-derivative controller with feedback linearization is applied for positioning and tracking control. Furthermore, in order to achieve movement of the mechanism without the sudden stopping at the desired point, a trapezoidal velocity profile is used to obtain desired trajectory. Numerical simulations using Matlab/Simulink successfully demonstrate the effectiveness of the proposed method.*

**Keywords:** Gripping mechanism, Robotics, Feedback linearization, Control engineering

## 1. INTRODUCTION

A robot gripping mechanism is a device that enables the capturing, handling, releasing and tightening of an object that is manipulated. Modern robotics and their grippers use integrated mechanisms and controls to simulate movements of human hands. Robotic grippers are only one component of an automated system, that has been around for more than fifty years ever since the development of the first Stanford arm, an early robot that would come to be known as the first readily controllable gripper. Many of its design and control elements are used in grippers of today [1]. The focal point of this paper is on the implementation of feedback linearization for trajectory tracking of a robot gripping mechanism with two degrees of freedom. Furthermore a proportional integral-derivative (PID) controller is used along with the feedback linearization. The PID controller uses a control loop feedback mechanism, to control process variables and to keep the actual output from a process as close to the target as possible, hence it is to this day the cheapest and the most accurate and stable controller.

Since, it is necessary to implement feedback linearization control (FLC), nonlinear feedback terms are utilized and a mathematical model for the two-link robot is of the essence if this technique is applied. An exact mathematical models cannot be easily obtained, particularly for the system parameters of a two-link gripping mechanism. One way to battle this kind of problem was introduced in the paper which used real coded genetic algorithm with a multiple-crossover in order to estimate the unknown system parameters. In order to later on incorporate the resulted system model to the feedback linearization control such that the nonlinear robotic system can be transferred to a linear model with a nonlinear bounded time-varying uncertainty [2]. FLC is used to compute the required arm torque using nonlinear feedback control law. In addition, when all dynamic and physical parameters are known the FLC works remarkably. Given that a large amount of systems have uncertainties and the fuzzy FLC can reduce this kind of limitation. A robotic manipulator with three degree of freedom is controlled by a novel fuzzy sliding feedback

linearization controller. In this paper the work outline uses soft computing in order to increase the stability and robustness new mathematical switching sliding mode methodology is applied to the fuzzy FLC [3]. Additionally, a new optimal proportional-derivative (PD) feedback linearization controller is employed to achieve the finest trajectory tracking for nonholonomic wheeled mobile robots. In this paper a teaching-learning-based optimization is utilized in order for the proposed controller to handle the difficulty of the integrated kinematic and dynamic tracking [4]. In the ensuing paper a numerical algorithm, based on the Newton-Euler dynamic equations, is used in order to compute the inverse dynamics of robot manipulators with an arbitrary number of joints. Furthermore, a variant of the algorithm is used for implementing a FLC law for the accurate tracking of desired link and stiffness trajectories. Considering that the algorithm does not use numerical approximations, it grows linearly in complexity with the number of joints and is therefore suitable for online feed forward and real-time feedback control [5]. In the next paper utilizing a systematic method to build a kinematic model and dynamic model of a nonholonomic wheeled mobile robot with a longitudinal and lateral slip, a control law is used with the input-output feedback linearization method to drive the mobile robot to track a given trajectory while longitudinal, and lateral slip simultaneously exist. The asymptotical stability of the system is corroborated by solving second-order differential linear equations [6]. Moreover, a variety of control methods were used to control the space robots attitude to obtain time response in order to minimize the eulerint criterion. The PD controller was used, as well as, other control methods including LQR, pole placement and adaptive feedback linearization. All of this was conducted using the quaternion kinematics to determine which method yields the lowest value, control effort and simulation elapsed time [7]. Furthermore, the integration of a cable-driven parallel Robot with a wheeled mobile robot is proposed to overcome some of the issues related to each of these robots. In order to derive the dynamic equations the Gibbs-Appel formulation is used. Howbeit, based on some conditions, the equations are input-output linearizable via

\*Corresponding author: Kraljice Marije 16, Belgrade, Serbia and mvesovic@mas.bg.ac.rs

a static feedback. The platform trajectory is designed based on the given end-effector trajectory tests [8]. Ensuing all the following, in one of the papers the trajectory tracking problem is addressed of the end-effector of a single link flexible arm in which the gravity forces and the joint friction forces are taken into account. As an overall approach, a double loop cascade control is used to deal with the joint friction, while in its outer loop an input-state feedback linearization-based controller is implemented to suppress the vibrations and track an end-effector trajectory [9]. Similarly, a paper that develops a simple learning strategy for the FLC algorithm for uncertain nonlinear systems is proposed. Here, the strategy uses desired closed-loop error dynamics to update the controller coefficients and the disturbance term in the feedback control law, whilst the traditional feed forward control law is designed based on the nominal model by using FLC method [10]. Moreover, in a different paper an optimal super-twisting algorithm with time delay estimation is designed based on input/output feedback linearization for uncertain robot manipulators. Lyapunov theory is used to prove the finite-time convergence of the sliding surface and its derivative, into the bargain this structure is used to estimate unknown dynamics and to reduce the control effort and the chattering phenomenon [11]. A predictor-based controller for a high-DOF manipulator to compensate a time-invariant input delay during a pick-and-place task is proposed in this paper. The controller is formulated in the presence of input delay, in order to track desired trajectories later on being able to investigate the effects of input delays in the absence of a robust predictor. In due course this reveals a robustness of the formulated algorithm [12].

As the construction industry frequently deals with the problem of robot's movements and gripping optimization, the main goal of this paper is to apply some of the more conventional techniques like the PID with feedback linearization, so as to achieve the movement of the mechanism without the sudden stopping at the desired point. In those purposes, a trapezoidal velocity profile will be used to obtain a desired trajectory.

This study proposes only a fragment of intelligent machine working cycle, which refers to movement of the gripping mechanism.

## 2. MATHEMATICAL MODEL OF THE OBJECT

Real object is shown in the Figure 1. Robot has a gripper that can be represented as link mechanism with two degrees of freedom (2DOF), which is approximated with the scheme as shown in Figure 2.

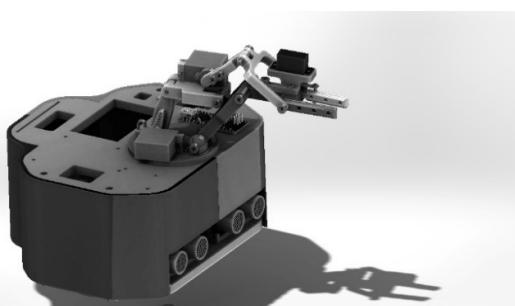


Figure 1: Mobile robot with gripping mechanism

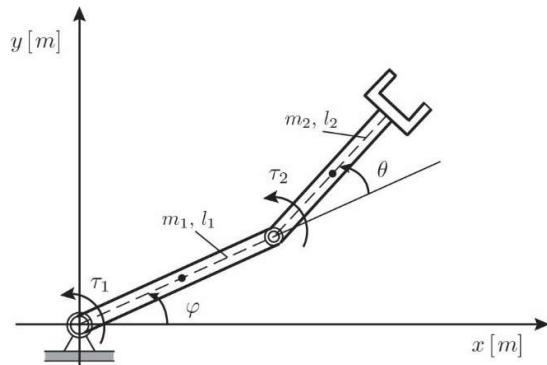


Figure 2: Scheme of the robotic gripperer

It is assumed that the centers of masses are in the middle of levers.

From the Figure 2 coordinates of centers of masses for the first:  $x_{1CM}, y_{1CM}$ , and the second link:  $x_{2CM}, y_{2CM}$  are:

$$\begin{aligned} x_{1CM} &= \frac{l_1}{2} \cos \varphi \\ y_{1CM} &= \frac{l_1}{2} \sin \varphi \\ x_{2CM} &= l_1 \cos \varphi + \frac{l_2}{2} \cos(\varphi + \theta) \\ y_{2CM} &= l_1 \sin \varphi + \frac{l_2}{2} \sin(\varphi + \theta), \end{aligned} \quad (1)$$

where  $l_1$  and  $l_2$  are lengths of links and  $\varphi$  and  $\theta$  are angles, taken to be generalized coordinates. Potential energy  $\Pi$  can be easily calculated:

$$\begin{aligned} \Pi = \frac{1}{2} m_1 g l_1 \sin \varphi + m_2 g l_1 \sin \varphi \\ + \frac{1}{2} m_2 g l_2 \sin(\varphi + \theta), \end{aligned} \quad (2)$$

where  $m_1, m_2$  are the masses of the links.

Kinetic energy of the system can be found as:

$$\begin{aligned} E_k = \dot{\varphi}^2 \left( \frac{1}{6} m_1 l_1^2 + \frac{1}{6} m_2 l_2^2 + \frac{1}{2} m_2 l_1^2 \right. \\ \left. + \frac{1}{2} m_2 l_1 l_2 \cos \theta \right) + \dot{\theta}^2 \left( \frac{1}{6} m_2 l_2^2 \right) \\ + \dot{\varphi} \dot{\theta} \left( \frac{1}{3} m_2 l_2^2 + \frac{1}{2} m_2 l_1 l_2 \cos \theta \right) \end{aligned} \quad (3)$$

Energy is conserved, so the Lagrangian of the system is described as:

$$L = E_k - \Pi \quad (4)$$

and the Euler - Lagrange equations for the two - link rotary system are:

$$\begin{aligned} \frac{\partial^2 L}{\partial t \partial \dot{\varphi}} - \frac{\partial L}{\partial \varphi} &= Q_1 \\ \frac{\partial^2 L}{\partial t \partial \dot{\theta}} - \frac{\partial L}{\partial \theta} &= Q_2 \end{aligned} \quad (5)$$

where  $Q_i, i = 1, 2$  are the generalized forces.

For:

$$Q_i = \tau_i \quad (6)$$

where  $\tau_i$  are torques. The Euler-Lagrange equations is a systematic method of finding the equations of motion, i.e., EOMs, of a system. Once the kinetic and potential energy are obtained and the Lagrangian is found, then the task is to compute various derivatives to get the EOMs. By introducing generalized coordinates and abbreviations:

$\varphi = q_1$ ,  $\theta = q_2$ ,  $S_1 = \sin q_1$ ,  $S_2 = \sin q_2$ ,  $S_{12} = \sin(q_1 + q_2)$ ,  $C_1 = \cos q_1$ ,  $C_2 = \cos q_2$ ,  $C_{12} = \cos(q_1 + q_2)$  and after going through this process, the nonlinear equations of motion i.e. dynamic model for the rotary two – linked gripping mechanism is:

$$\begin{aligned} M\ddot{\mathbf{q}} + C\dot{\mathbf{q}} + \mathbf{G} &= \boldsymbol{\tau} \\ \mathbf{Y} &= \mathbf{q} \end{aligned} \quad (7)$$

where:

$$\begin{aligned} \mathbf{M} &= \begin{bmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{bmatrix} \\ M_{11} &= \frac{1}{3}m_1l_1^2 + \frac{1}{3}m_2l_2^2 + m_2l_1^2 + m_2l_1l_2C_2 \\ M_{12} &= M_{21} = \frac{1}{3}m_2l_2^2 + \frac{1}{2}m_2l_1l_2C_2 \\ M_{22} &= \frac{1}{3}m_2l_2^2 \end{aligned} \quad (8)$$

is the inertia matrix,

$$\mathbf{q} = \begin{bmatrix} q_1 \\ q_2 \end{bmatrix} \quad (9)$$

is vector of generalized coordinates,

$$\begin{aligned} \mathbf{C} &= \begin{bmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{bmatrix} \\ C_{11} &= -\frac{1}{2}m_2l_1l_2\dot{q}_2S_2 \\ C_{12} &= -\frac{1}{2}m_2l_1l_2S_2(\dot{q}_1 + \dot{q}_2) \\ C_{21} &= -\frac{1}{2}m_2l_1l_2\dot{q}_1S_2 \\ C_{22} &= 0 \end{aligned} \quad (10)$$

is vector of Coriolis and centrifugal forces,

$$\begin{aligned} \mathbf{G} &= \begin{bmatrix} G_1 \\ G_2 \end{bmatrix} \\ G_1 &= \left( \frac{1}{2}m_1l_1 + m_2l_1 \right) gC_1 + \frac{1}{2}m_2l_2gC_{12} \\ G_2 &= \frac{1}{2}m_2l_2gC_{12} \end{aligned} \quad (11)$$

is a vector of gravity torques,

$$\boldsymbol{\tau} = \begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix} \quad (12)$$

is the vector of torque and it will be considered as the control (input signal);  $\mathbf{Y}$  is the output.

### 3. VELOCITY PROFILE OF THE GRIPPING MECHANISM

#### 3.1. Trapezoidal velocity profile

The trapezoidal velocity profile is a realistically feasible implementation of motion at a constant speed. Namely, since the robot starts from a state of rest it takes some time to reach that constant speed. Also, stopping does not happen abruptly. Instead it include braking in a controlled manner at the end of the movement, and the

speed decreases to zero. This movement can be described from the mathematical aspect. Important parameters are: the initial  $q_i$  and the final  $q_f$  generalized coordinates, as well as the total duration of the movement  $T_f$  and the time provided for acceleration, i.e. deceleration  $T_a$ . The position, velocity and acceleration profiles are shown in Figure 3.

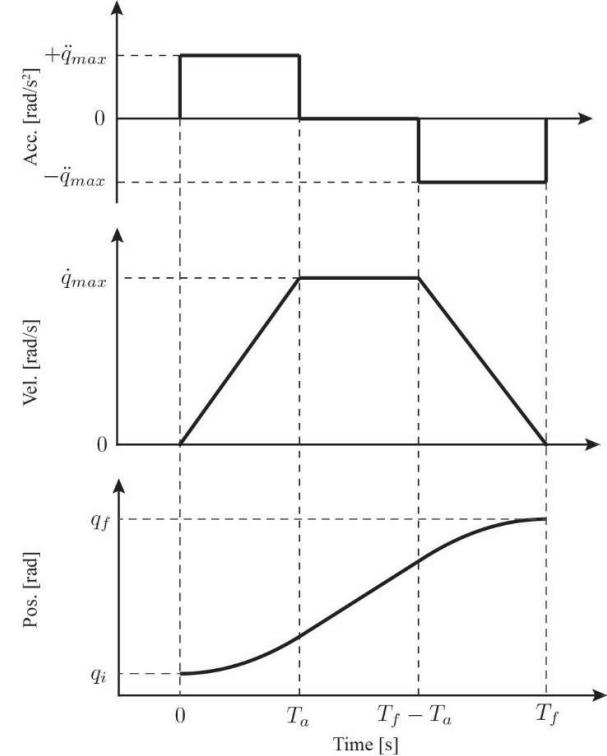


Figure 3: The trapezoidal velocity profile: the acceleration, velocity and position profiles

From Figure 3 it can be seen that, in the time interval from the initial moment to the moment  $T_a$ , acceleration is positive, constant and equal to  $\ddot{q}_{max}$ . It is clear that body moves with uniformly accelerated rectilinear motion, so the speed of movement increases linearly, from the value of zero, i.e. rest, to the value of  $q_{max}$ , while the position changes according to smooth curve. This interval represents robot acceleration period.

From the moment  $T_a$  to the moment  $T_f - T_a$ , the acceleration is equal to zero, so the body moves at a constant speed  $q_{max}$ . In this period, the change of position is a linear function of time.

Finally, from the moment  $T_f - T_a$  to the end of the motion, body has the negative constant acceleration  $-\ddot{q}_{max}$ , i.e. deceleration, so the movement is uniformly decelerated. Therefore the velocity is linear decreases from  $q_{max}$  cruising speed to zero when the robot stops. Change of position in this interval is again a squared function of time. This interval of movement is called the period robot braking. Due to the appearance of the velocity graph, this movement was called trapezoidal movement speed profile. Maximum velocity, maximum acceleration and the time provided for acceleration are not of independent magnitude:

$$\dot{q}_{max} = \ddot{q}_{max} T_a \quad (13)$$

From the Figure 3. acceleration is given with the equation:

$$\ddot{q}(t) = \begin{cases} +\ddot{q}_{max} & 0 < t \leq T_a \\ 0 & T_a < t \leq T_f - T_a \\ -\ddot{q}_{max} & T_f - T_a < t \leq T_f \end{cases} \quad (14)$$

The velocity can be determined by integrating equation (14), taking into account the initial conditions and connection given by expression (13), in the following form:

$$\dot{q}(t) = \begin{cases} \ddot{q}_{max} \cdot t & 0 < t \leq T_a \\ \ddot{q}_{max} \cdot T_a & T_a < t \leq T_f - T_a \\ \ddot{q}_{max} \cdot (T_f - t) & T_f - T_a < t \leq T_f \end{cases} \quad (15)$$

The expression for the position can be obtained by integrating the expression for velocity (15) and taking into account again initial conditions:

$$q(t) = \begin{cases} q_i + 0.5 \cdot \ddot{q}_{max} \cdot t^2 & 0 < t \leq T_a \\ q_i + \ddot{q}_{max} \cdot T_a \left( t - \frac{T_a}{2} \right) & T_a < t \leq T_f - T_a \\ q_f - 0.5 \ddot{q}_{max} (T_f - t)^2 & T_f - T_a < t \leq T_f \end{cases} \quad (16)$$

Since that assumption is made that the important parameters are the given range of position change from  $q_i$  to  $q_f$ , total movement execution time  $T_f$ , as well as acceleration or braking time  $T_a$ , the only unknown quantity in expressions (13) - (16) is the maximum value of acceleration  $\ddot{q}_{max}$ , which, obviously, must depend on the initial parameters. The required value of maximum acceleration is determined as follows. The change of the position from the beginning to the end, i.e. the distance, must be equal to the area below speed chart. Given the fact that the graph has the shape of a trapezoid, it is easy to establish connection:

$$q_f - q_i = (\ddot{q}_{max} \cdot T_a) \cdot (T_f - T_a), \quad (17)$$

where, according to the (15) expression  $\ddot{q}_{max} \cdot T_a$  represents trapezoid height, while the expression  $T_f - T_a$  stands for the length of the trapezoid midsegment. With all of these expression it is easy to obtain unknown parameter of the maximum acceleration as:

$$\ddot{q}_{max} = \frac{q_f - q_i}{T_a(T_f - T_a)} \quad (18)$$

In all previous expressions it is logically assumed that:

$$T_a \leq \frac{T_f}{2} \quad (19)$$

that is, the acceleration or braking periods cannot be longer than half of the total time. In the special case, when  $T_a = \frac{T_f}{2}$  the acceleration and deceleration periods last per half of the total time, as soon as the speed  $q_{max}$  is reached the braking starts. Trapezoidal velocity profile from Figure 3 is then deformed because the central part is lost for the

movement with the constant speed, this kind of movement is called a triangle speed profile. The other way for obtaining the trapezoidal velocity profile is taking into the account the robot's movements, from the point of view of velocity and acceleration, which are limited by the applied actuators. In other words, there is a maximum acceleration given by the actuator, so time parameters can be obtained according to it. Therefore, sometimes in practice, it is common, instead of setting the duration of the movement and acceleration or deceleration periods, setting the maximum acceleration, and to calculate the time parameters [13].

#### 4. FEEDBACK LINEARIZATION

##### 4.1. Introduction

One of the major engineering problems is finding a mathematical model that is good enough to describe the system, and at the same time easy enough to treat mathematically and analytically. Depending on the design goals, there are several formulations of the control problem. The task of stabilization, tracking, and disturbance rejection or attenuation (and various combination in them) lead to a number of control problems. There are many control tasks that require the use of feedback [14]. That is why feedback linearization is a very useful technique that strikes a great balance between a good model and simple nonlinear control algorithm. Precisely because of these features, this method has found wide usage in research and applied engineering. It has been successfully implemented in many applications of control, such as industrial robots, high performance aircraft, helicopters and biomedical engineering. The central idea of the approach is to algebraically transform a nonlinear system dynamics into a (fully or partly) linear one, so that linear control techniques can be applied. [15]. Feedback linearization requires extremely precise measurements of system parameters to eliminate the effect of nonlinearity from the system and thereby achieve the anticipated effects [16] – [19].

##### 4.2. Conditions

Feedback linearization approach differs from the classical linearization (about the desired equilibrium point) in that no approximation is used; it is exact [14]. This differs entirely from conventional linearization in that feedback linearization is achieved by exact state transformations and feedback, rather than by linear approximations of the dynamics [15].

In this section, the theoretical basis for the application of the proposed method will be presented. Also, the conditions for applying this algorithm on the 2DOF gripping mechanism will be checked.

Of specific concern will be creating the control approach in the sense of the feedback algorithm law which will bring the annulment of nonlinearity in the equations that describe the system.

For the single input – single output nonlinear SISO system [14]:

$$\begin{aligned} \dot{x} &= f(x) + g(x)u \\ y &= h(x) \end{aligned}$$

\*Corresponding author: Kraljice Marije 16, Belgrade, Serbia and mvesovic@mas.bg.ac.rs

where  $\mathbf{f}(\mathbf{x})$ ,  $\mathbf{g}(\mathbf{x})$  and  $h(\mathbf{x})$  are sufficiently smooth in a domain  $D \subset R^n$ .  $\dot{\mathbf{x}} = [x_1, x_2, \dots, x_n]^T$  is a state vector. It is necessary to find a state feedback control  $u$ , that transforms the nonlinear system into an equivalent linear system. Clearly, generalization of this idea is not possible in every nonlinear system: there must be a certain structural property that allows performing in such a manner of cancellation. There are four conditions that must be met to enable this type of control.

A. State equation of the system has to be in the following form:

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\gamma(\mathbf{x})[u - \alpha(\mathbf{x})]$$

Note: If the system is not in this form, in some cases it is possible to transform it by transforming the coordinates. All functions has to be smooth enough.

B. Pair  $(\mathbf{A}, \mathbf{B})$  must be controllable, i.e. there has to be a controllability matrix whose rank is equal to the order of the system:  $\text{rank}(\mathbf{U}) = n$ .

C.  $\gamma(\mathbf{x})$  must be nonsingular, or if it is a scalar value then it has be different from zero.

D. All functions i.e. functions has to be differentiable (smooth enough).

In order to determine form of the control law new term relative degree of the system will be explained.

The relative degree of a linear system is defined as the difference between the poles (degree of the transfer function's denominator polynomial number) and zeros (degree of its numerator polynomial) [20].

*Relative degree*, in notation  $r$ , of the system which can be described at a point  $x_0$  is defined if:

1.  $L_g L_f^k h(\mathbf{x}) = 0$  for all  $x$  in a neighbourhood of  $x_0$  and all  $k < r - 1$

2.  $L_g L_f^{r-1} h(\mathbf{x}) \neq 0$

$$\mathbf{U} = \begin{bmatrix} \frac{M_{22}}{\#2} & -\frac{M_{12}}{\#2} & \frac{-C_{11}M_{22}^2}{\#2^2} - \#1 & \frac{C_{21}M_{11}M_{12} + C_{11}M_{12}M_{22}}{\#2^2} \\ -M_{21} & M_{11} & \frac{C_{22}M_{11}M_{21} + C_{21}M_{21}M_{22}}{\#2^2} & \frac{-C_{22}M_{11}^2}{\#2^2} - \#1 \end{bmatrix}$$

where the following substitution was made:  $\#1 = \frac{C_{21}M_{11}M_{12}}{\#2^2}$ ,  $\#2 = M_{11}M_{22} - M_{12}M_{21}$

C.  $\gamma(\mathbf{x})$  is a scalar value, equal to 1

D. All functions are differentiable (since in them are only sine and cosine functions), so this condition is met.

It is obvious from (7) that the second derivative of the output  $\mathbf{Y}$  will be the first derivative that will be equal to the function, which depends on the input signal ( $\tau$ ). It is clear from the definition of relative degree that:  $r=2$  - relative degree of the system is two:

$$\ddot{\mathbf{Y}} = \ddot{\mathbf{q}} = \mathbf{M}^{-1}(-\mathbf{C}\dot{\mathbf{q}} - \mathbf{G} + \boldsymbol{\tau}) \quad (25)$$

The terms  $L_g$  and  $L_f^k$  represent the Lie derivative of  $h(\mathbf{x})$  taken along  $\mathbf{g}(\mathbf{x})$  and  $k$ -times along  $(\mathbf{x})$ , respectively.

Now finally, using these theoretical definitions, a control law can be formed by annulling nonlinearities in order to obtain linear equations:

$$u = \frac{1}{L_g L_f^{r-1} h(\mathbf{x})} [-L_f^r h(\mathbf{x}) + z] \quad (20)$$

This control signal will reduce the input – output map to:

$$y^r = z, \quad (21)$$

So this kind of linearization is called Input – Output Linearization, and  $z$  is called ‘new control signal’. The control signal given in (22) has a more general form:

$$u = \alpha(\mathbf{x}) + \frac{1}{\gamma(\mathbf{x})} z \quad (22)$$

The equation (20) has the same constraints as it is in C. and that is  $L_g L_f^{r-1} h(\mathbf{x})$  has to be different from zero.

#### 4.3. Control law synthesis

One of the first steps in the synthesis of a feedback linearizing controller is to check all of the constraints from the previous section.

A. State equation of the system could be easily transformed into a suitable form:

$$\begin{aligned} \dot{\mathbf{v}} &= -\mathbf{M}^{-1} \mathbf{C} \mathbf{v} + \mathbf{M}^{-1}(-\mathbf{G} + \boldsymbol{\tau}) \\ \mathbf{A} &= -\mathbf{M}^{-1} \mathbf{C} \\ \mathbf{B} &= \mathbf{M}^{-1} \\ \gamma(\mathbf{x}) &= 1, \alpha(\mathbf{x}) = \mathbf{G} \end{aligned} \quad (23)$$

B. Controllability matrix has the following form:

$$\mathbf{U} = \begin{bmatrix} \frac{M_{22}}{\#2} & -\frac{M_{12}}{\#2} & \frac{-C_{11}M_{22}^2}{\#2^2} - \#1 & \frac{C_{21}M_{11}M_{12} + C_{11}M_{12}M_{22}}{\#2^2} \\ -M_{21} & M_{11} & \frac{C_{22}M_{11}M_{21} + C_{21}M_{21}M_{22}}{\#2^2} & \frac{-C_{22}M_{11}^2}{\#2^2} - \#1 \end{bmatrix} \quad (24)$$

In order to fulfil (20) the control signal  $\boldsymbol{\tau}$  is chosen to be in the form:

$$\boldsymbol{\tau} = \mathbf{C} \dot{\mathbf{q}} + \mathbf{G} + \mathbf{M} z, \quad (26)$$

and new linear system is in the form:

$$\begin{aligned} \ddot{q}_1 &= z_1 \\ \ddot{q}_2 &= z_2 \end{aligned} \quad (27)$$

The new control signal  $z$  is chosen to be in the form of proportional derivative controller:

$$z = \begin{bmatrix} K_P \varepsilon + K_D \dot{\varepsilon} \\ K_P \ddot{\varepsilon} + K_D \dot{\varepsilon} \end{bmatrix} \quad (28)$$

Coefficient of the PD controller has been chosen as:

$$\begin{aligned} K_P &= \omega_0^2 \\ K_D &= 2\zeta\omega_0 \end{aligned} \quad (29)$$

where  $\omega_0$  is a natural frequency and  $\zeta$  is a damping factor and they were calculated with the respect of the overshoot and settling time:  $\zeta = 0.9815$  and  $\omega_0 = 9.3731$ , so the parameters of PD controllers are (they have same values):

$$K_P = 87.8555, K_D = 18.4000$$

## 5. EXPERIMENTAL RESULTS

### 5.1. The desired shape of the trajectory

The initial position of the robotic gripper is determined by the mechanism itself. The initial generalized coordinates are equal to:  $q_{10} = \varphi_0 = 80^\circ \approx 1.396260$  rad and  $q_{20} = \theta_0 = -30^\circ \approx -0.523599$  rad. This means that the gripping mechanism (end effector) from the Figure 2 has the coordinates:  $x_0 = 0.0373$  and  $y_0 = 0.0898$ .

Main goal of this study was to move that point from its initials to the finals coordinates. In order to do so, the feedback control system with the proportional derivative gains was developed. It is important to notice that its algorithm is convinient for any reference, i.e. it allows tracking of a given path to any endpoint with the desired velocity (and therefore with the desired position and acceleration) profile. For example, in this paper task will be to achieve angles (or in the Cartesian coordinates, end effector should be in):

$$\begin{aligned} q_{1f} &= \varphi_f = 45^\circ \approx 0.785398 \text{ rad} \\ q_{2f} &= \theta_f = -45^\circ \approx -0.785398 \text{ rad}, \\ x_f &= 0.0834 \\ y_f &= 0.0412 \end{aligned}$$

Additionally, the time required to reach this position is set to be  $T_f = 6$  s.

Maximum acceleration  $\ddot{q}_{max}$  and the time provided for acceleration, i.e. deceleration  $T_a$  are calculated according to (18) and Section 3 as:  $T_a = \frac{T_f}{3} = 2s$ ,  $|\ddot{q}_1| = 0.0763 \frac{m}{s^2}$  and  $|\ddot{q}_2| = 0.0330 \frac{m}{s^2}$ .

The parameters for the gripper are given in Table 1.

Table 1: Gripping mechanism parameters

	mass [kg]	length [m]
first link	0.00799	0.05831
second link	0.00521	0.0422

Comparisons between the desired trajectory signals and the output signals are shown on the following Figures. All Figures were obtained from Matlab and Simulink

graphical programming environment for modelling, simulating and analyzing dynamical systems.

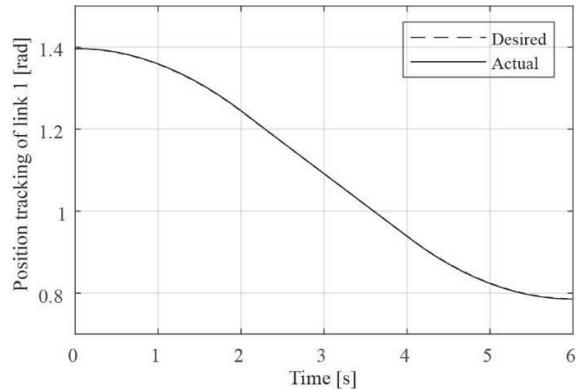


Figure 4: Comparison: Desired and simulated position for link 1

As it is very difficult to see, from the previous Figure (4), the difference between the desired and the output signal, the next one (Figure 5) shows an enlarged detail from it.

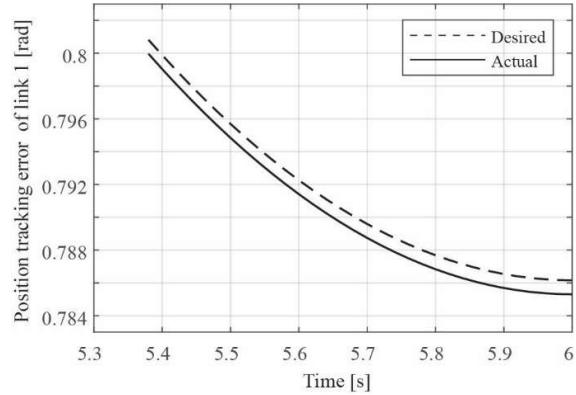


Figure 5: Increased detail from Figure 4

Velocity profile is given as trapezoidal function of maxima acceleration, time, and specific moments in time, as shown on Figure 6.

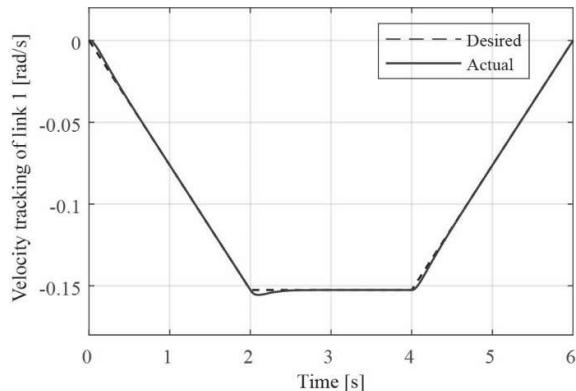


Figure 6: Comparison: Desired and simulated velocity for link 1

For constant velocity, the acceleration will be zero, Figure 7.

\*Corresponding author: Kraljice Marije 16, Belgrade, Serbia and mvesovic@mas.bg.ac.rs

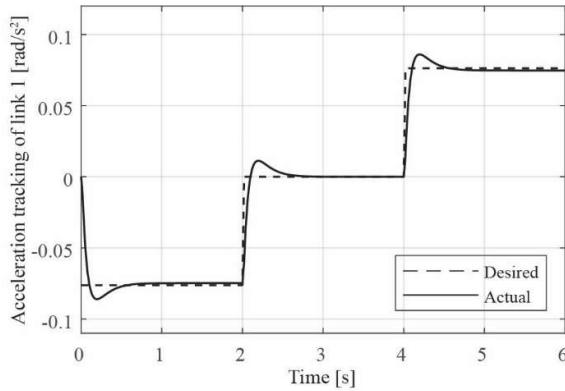


Figure 7: Comparison: Desired and simulated acceleration for link 1

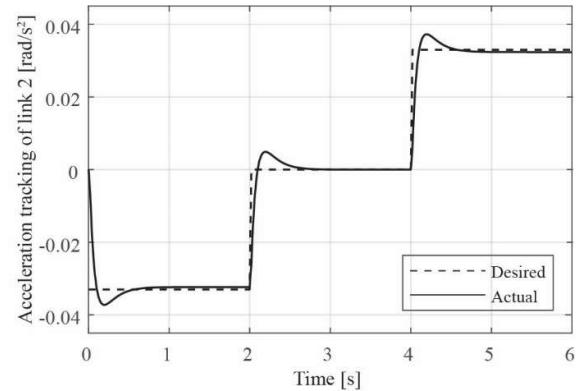


Figure 10: Comparison: Desired and simulated acceleration for link 2

Same conclusion can be drawn for Figures 8 – 11.

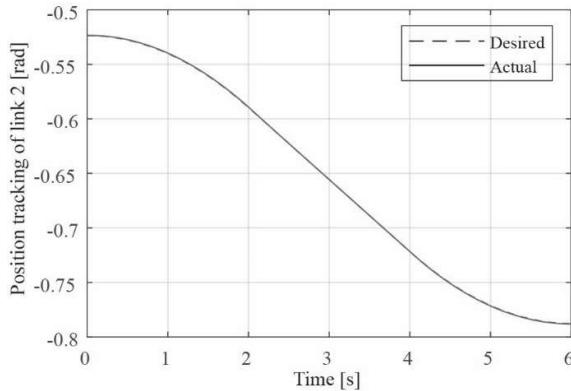


Figure 8: Comparison: Desired and simulated position for link 2

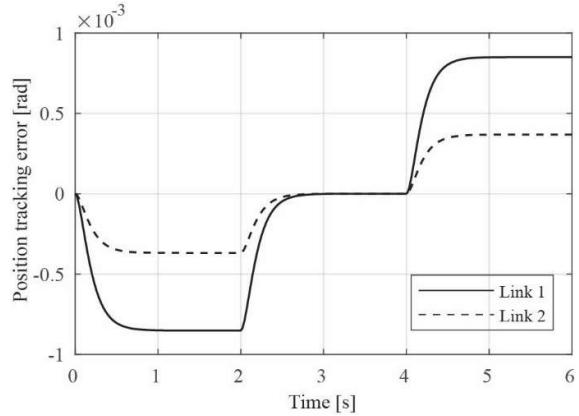


Figure 11: Position tracking errors

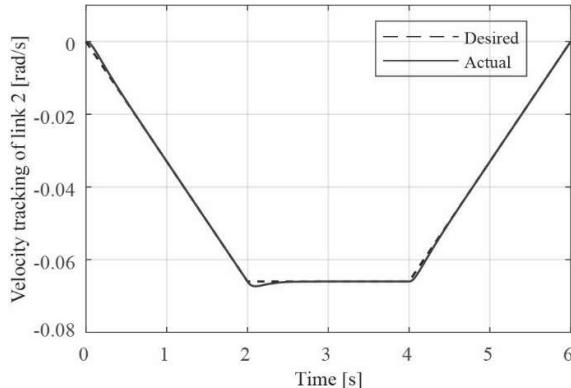


Figure 9: Comparison: Desired and simulated velocity for link 2

## 6. CONCLUSION

As a means to give an insight into the navigation and to obtain satisfying examination of the mobile robot's arm before the fabrication, this paper proposed detailed solution to the specific control problem for the two degrees of freedom (2DOF) gripping mechanism. Firstly, mathematical model has been obtained using Lagrange's approach and an evaluation of the theoretical bases was made. The nonlinear mathematical model has been derived using the positions and velocities of the points, along with the moments of inertia and energies of the system. Due to the existing nonlinearities that occur in the system, the feedback linearization with proportional derivative controller method was chosen. For the given trajectory controller parameters were chosen based on the new linear model, which was obtained using nonlinearities cancellation. Additionally, a simulation study was done for the specific initial case and the trapezoidal velocity profile was generated. This profile was used to obtain referent signal so the motion of the robot is shown to be flat and without an abrupt braking. Obtained experimental results are provided and they have verified the effectiveness of the proposed performance. The control algorithm has been validated using Matlab/Simulink software. Finally, by applying this approach satisfying results are achieved and the reasonableness of each assumption has been proven.

## ACKNOWLEDGEMENTS

This research was supported by the Science Fund of the Republic of Serbia, grant No. 6523109, AI- MISSION4.0, 2020-2022.

This work was financially supported by the Ministry of Education, Science and Technological Development of the Serbian Government, MPNTR RS under contract 451-03-9/2021-14/200105, Grant TR-35029 (2021).

## REFERENCES

- [1] Stanford Artificial Intelligence Laboratory: [http://infolab.stanford.edu/pub/voy/museum/pd/Calculator\\_s.htm](http://infolab.stanford.edu/pub/voy/museum/pd/Calculator_s.htm) (last accessed 25/05/2021).
- [2] J.L. Chen and W.D. Chang, "Feedback linearization control of a two-link robot using a multi-crossover genetic algorithm," *Expert Systems with Applications*, Vol. 36(2), pp. 4154–4159, (2009)
- [3] F. Piltan, M.H. Yarmahmoudi, M. Mirzaie, S. Emamzadeh and Z. Hivand, "Design Novel Fuzzy Robust Feedback Linearization Control with Application to Robot Manipulator," *International Journal of Intelligent Systems and Applications*, Vol. 5(5), pp. 1–10, (2013)
- [4] M.H. Khooban, "Design an intelligent proportional-derivative PD feedback linearization control for nonholonomic-wheeled mobile robot," *International Journal of Intelligent and Fuzzy Systems*, Vol. 26(4), pp. 1833–1843, (2014)
- [5] G. Buondonno and A. De Luca, "Efficient Computation of Inverse Dynamics and Feedback Linearization for VSA-Based Robots," *IEEE Robotics and Automation Letters*, Vol. 1(2), pp. 908–915, (2016)
- [6] N. V. Tinh, N. T. Linh, P. T. Cat, P. M. Tuan, M. N. Anh and N. P. T. Anh, "Modeling and feedback linearization control of a nonholonomic wheeled mobile robot with longitudinal, lateral slips," *IEEE International Conference on Automation Science and Engineering*, Fort Worth, TX (USA), 21 August-24 August 2016, pp. 996–1001, (2016)
- [7] M. Navabi and S. Hosseini, "Adaptive feedback linearization control of space robots," *IEEE 4th International Conference on Knowledge-Based Engineering and Innovation*, Tehran (Iran), 22 December 2017, pp. 0965–0970, (2017)
- [8] M.H. Korayem, M. Yousefzadeh and S. Manteghi, "Dynamics and input–output feedback linearization control of a wheeled mobile cable-driven parallel robot," *Multibody System Dynamics*, Vol. 40, pp. 55–73, (2017)
- [9] J.C. Cambera and V. Feliu-Batlle, "Input-state feedback linearization control of a single-link flexible robot arm moving under gravity and joint friction," *Robotics and Autonomous Systems*, Vol. 88, pp. 24–36, (2017)
- [10] M. Mehndiratta, E. Kayacan and E. Kayacan, "A Simple Learning Strategy for Feedback Linearization Control of Aerial Package Delivery Robot," *IEEE Conference on Control Technology and Applications*, Copenhagen (Denmark), 21 August-24 August 2018, pp. 361–367, (2018)
- [11] Y. Kali, M. Saad and K. Benjelloun, "Optimal super-twisting algorithm with time delay estimation for robot manipulators based on feedback linearization," *Robotics and Autonomous Systems*, Vol. 108, pp. 87–99, (2018)
- [12] M. Bagheri, P. Naseradimousavi and M. Krstić, "Feedback linearization based predictor for time delay control of a high-DOF robot manipulator," *Automatica*, Vol. 108, (2019)
- [13] [https://www.ucg.ac.me/skladiste/blog\\_13269/objava\\_8796/fajlovi/35\\_industrijska781.pdf](https://www.ucg.ac.me/skladiste/blog_13269/objava_8796/fajlovi/35_industrijska781.pdf), University of Montenegro, pp. 87-102, (last accessed 30/05/21)
- [14] H. K. Khalil, "Nonlinear Systems"; 3rd ed. New Jersey, United States: Prentice Hall, (2002)
- [15] W. Ghozlane, J. Knani, "Nonlinear Control via Input-Output Feedback Linearization of a Robot Manipulator," *Advances in Science, Technology and Engineering Systems Journal*, vol. 3, October, (2018)
- [16] Farzin, P.; Yarmahmoudi, M.H.; Mirzaie, M.; Emamzadeh, S.; Hivand, Z. "Design novel fuzzy robust feedback linearization control with application to robot manipulator" *J. Intell. Syst. Appl.*, Vol. 5, pp. 1–10, (2013)
- [17] Lin, C.-H. Precision Motion Control of a Linear Permanent Magnet Synchronous Machine Based on Linear Optical-Ruler Sensor and Hall Sensor. *Sensors*, 18, 3345. (2018)
- [18] Su, W.-T.; Liaw, C.-M. Adaptive positioning control for a LPMSM drive based on adapted inverse model and robust disturbance observer. *IEEE Trans. Power Electron.*, 21, pp. 505–517, (2006)
- [19] Yung-Te Chen & Chi-Shan Yu & Ping-Nan Chen, "Feedback Linearization Based Robust Control for Linear Permanent Magnet Synchronous Motors," *Energies*, MDPI, Open Access Journal, vol. 13(20), pp. 1-17, (2020)
- [20] M. Vesović, R. Jovanović, L. Laban, V. Zarić, "Modelling and Control of a Series Direct Current (DC) Machines Using Feedback Linearization Approach," 7th International Conference on Electrical, Electronic and Computing Engineering IcETRAN, 28 September – 29 September, pp. 191 – 197, (2020)

# Modeling and Control of a Liquid Level System Based on the Takagi-Sugeno Fuzzy Model Using the Whale Optimization Algorithm

Radiša Jovanović, Vladimir Zarić, Mitra Vesović and Lara Laban

**Abstract**—The liquid level control remains an important task for research and is used by process control engineers. Firstly, the linear models for the tank system are obtained for the three different operating points. From these identified linear models a Takagi-Sugeno (TS) model is obtained using triangular membership functions in the premises of the rules. Furthermore, the whale optimization algorithm is implemented to fine tune parameters of Takagi-Sugeno fuzzy model, according to the chosen objective function. By using the parallel distributed compensation (PDC), a fuzzy controller is created by the fuzzy blending of three PI controllers designed for each of the operating points. In order to evaluate performance of the PDC based fuzzy controllers, the comparison is made between several local linear PI controllers and the PDC. Moreover, the PDC controllers from the optimized and original TS plant model are compared. The experimentally results and the comparison results verify efficiency of the proposed method.

**Index Terms**—Takagi-Sugeno; liquid level control; parallel distributed compensation; whale optimization algorithm.

## I. INTRODUCTION

The liquid level control has a wide range of applications in the process industries such as petro-chemical, waste water treatment and purification, biochemical, spray coating, beverages and pharmaceutical industries.

In [1] authors have conveyed and stressed the issue of performance analysis of three control schemes for couple tank system, PI (based on pole placement, Ziegler Nichols and Ciancone correlation tuning methods), PI-plus-feedforward and model predictive control. Moreover, paper [2] addresses the nonlinear control design problem for a liquid level system. A model-based backstepping controller and an adaptive backstepping controller are developed for the liquid level system. Following, the article [3] dabbles with the fuzzy-PID controller applied to the nonlinear dynamic model of the liquid level of the coupled tank system, all the while taking into account the effects of noise. The fuzzy model proposed by Takagi and Sugeno [4] is described by fuzzy IF-THEN rules which depict local linear

Radiša Jovanović is with the University of Belgrade, Faculty of Mechanical Engineering, Kraljice Marije 16, 11020 Belgrade, Serbia (e-mail: rjovanovic@mas.bg.ac.rs).

Vladimir Zarić is with the University of Belgrade, Faculty of Mechanical Engineering, Kraljice Marije 16, 11020 Belgrade, Serbia (e-mail: vzaric@mas.bg.ac.rs).

Mitra Vesović is with the University of Belgrade, Faculty of Mechanical Engineering, Kraljice Marije 16, 11020 Belgrade, Serbia (e-mail: mvesovic@mas.bg.ac.rs).

Lara Laban is with the University of Belgrade, Faculty of Mechanical Engineering, Kraljice Marije 16, 11020 Belgrade, Serbia (e-mail: llaban@mas.bg.ac.rs).

input-output relations of a nonlinear system. Fuzzy logic has many varieties that can be implemented for control purposes. For instance, one of them is parallel distributed compensation (PDC). The PDC offers a chance to use a technique to design a fuzzy controller from a given TS fuzzy model. In paper [5], a fuzzy controller is constructed based on a PDC method and it is implemented in an experimental tank level control system. The paper [6] suggests a procedure used to make two-variable fuzzy logic controllers (FLCs) set for the levels in a laboratory coupled-tank system. The plant input and output experimental data are then used for derivation via genetic algorithms optimization of a Takagi-Sugeno-Kang (TSK) plant model needed for FLC improvements. The TSK model is validated on a different set of experimental data and used in designing of two variable linear proportional-plus-integral PI controller and PDC with local linear PI controllers. In [7] a novel modification to the original PDC method is submitted, so that, besides the stability issue, the closed-loop performance of the system can be considered at the design stage. The strong point is that, for example, a faster response can be obtained, for a given bound on the control input. The following paper [8] gave a unified approach to a nonlinear model following control that contains the regulation and servo control problems as distinctive cases. A parallel distributed compensation (PDC) for fuzzy reference models was proposed. As a result of the following paper [9] a captivating method that improves the quality of robust control by interpolating a robust and optimal controller is presented. That paper introduces a new method called advanced robust parallel distributed compensation (ARPDC) for automatic control of nonlinear systems.

The fuzzy design can be considered as an optimization problem, where the structure, antecedent, and consequent parameters are required to be identified. Metaheuristic methods as global optimization algorithms can deal with non-convex, nonlinear, and multimodal problems subject to linear or nonlinear constraints with continuous or discrete decision variables. The synergy of fuzzy models and nature-inspired optimization algorithms belongs to the actual trends in soft computing, where all individual contributing technologies are seamlessly structured together. Attractive points of view on this combination are treated in the literature [10]. Recently, several metaheuristic methods have been proposed. Some of them include the genetic algorithm (GA) [11], particle swarm optimization (PSO) [12]-[15], gray wolf optimization (GWO) [16], whale optimization algorithm (WOA) [17] and ant colony optimization algorithm (ACO) [18]. Notwithstanding, the paper [19] explores the potentiality of a bat algorithm for tuning the

PID controllers. A modified WOA (MWOA) is used to tune the AFPID (adaptive fuzzy logic PID) parameters and showed improved performance when compared with conventional PID [14].

In this study, the structure and consequent parameters are known (number of rules, shapes of input membership functions and linear models in the consequent part of the rules), and antecedent parameters are determined using the whale optimizing algorithm.

## II. SYSTEM MODEL

The plant consisting of a pump integrated with a water basin and the tank as shown on Fig. 1. A practical industrial applications of such plant can be found in the processing system of petro-chemical, paper making, and/or water treatment plants, to name a few.

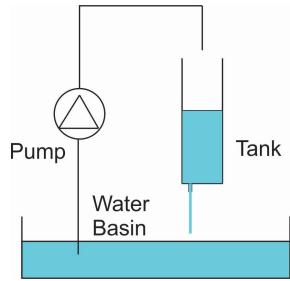


Fig. 1. The liquid level system.

### A. Mathematical modeling

The input into the process is the voltage to the pump  $V_p$  and its output is the water level in tank,  $H(t)$ . In acquiring the tank equation of motion the mass balance principle can be applied to the water level in tank, i.e.

$$A_t \dot{H} = Q_i - Q_o = KV_p - A_o V_o = KV_p - A_o \sqrt{2gH}, \quad (1)$$

where  $A_t$  is the area of tank,  $A_o$  is an area of the outlet orifice, while  $Q_i$  and  $Q_o$  are the inflow rate and outflow rate, respectively. The volumetric inflow rate to tank is supposed to be directly proportional to the applied pump voltage, such that Applying Bernoulli's equation for small orifices, the outflow velocity from tank,  $V_o$ , can be expressed by a succeeding relationship

$$V_o = \sqrt{2gH}, \quad Q_o = A_o V_o, \quad (2)$$

The nonlinear differential equation that describes the change in level in tank is:

$$\dot{H} = \frac{K}{A_t} V_p - \frac{A_o}{A_t} \sqrt{2gH}. \quad (3)$$

### B. Takagi Sugeno fuzzy model and identification

The main idea of the TS fuzzy modeling method is to partition the nonlinear system dynamics into several locally linearized subsystems, so that the overall nonlinear behavior of the system could be captured by fuzzy blending of such subsystems. Thus, a fuzzy model and identification of a liquid level system will be implemented in accordance with the TS model containing three rules. The fuzzy rule

associated with the  $i$ -th linear subsystem, can then be defined as  $i$ -th rule:

IF  $x(t)$  is  $M_i$  THEN

$$\dot{x}(t) = a_i x(t) + b_i u(t), \quad i = 1, 2, 3, \quad (4)$$

$$y(t) = c_i x(t), \quad i = 1, 2, 3. \quad (5)$$

Here  $M_i$  is the fuzzy set,  $x(t) \in \mathbb{R}$  is the state variable,  $u(t) \in \mathbb{R}$  is the input,  $y(t) \in \mathbb{R}$  is the output variable,  $a_i, b_i, c_i \in \mathbb{R}$ . In our case, the selected state space variable is equal to the output variable  $x(t) = y(t) = H(t)$ .

The overall output, using the fuzzy blend of the linear subsystems, will then be as follows:

$$\dot{x}(t) = \frac{\sum_{i=1}^3 w_i(x(t)) \{a_i x(t) + b_i u(t)\}}{\sum_{i=1}^3 w_i(x(t))}, \quad (6)$$

$$\dot{x}(t) = \sum_{i=1}^3 h_i(x(t)) \{a_i x(t) + b_i u(t)\}, \quad (7)$$

$$y(t) = \frac{\sum_{i=1}^3 w_i(x(t)) c_i x(t)}{\sum_{i=1}^3 w_i(x(t))} = \sum_{i=1}^3 h_i(x(t)) c_i x(t), \quad (8)$$

where  $w_i(x(t)) = M_i(x(t))$  is the grade of membership of  $x(t)$  in  $M_i$ . The linear models in the consequent rules (4) can be obtained by utilizing an analytical linearization of a nonlinear equation. Besides that, another approach is to apply the methods of identification in accordance with the measured input output data. The identification methods were used based on the step response. Since models obtained by identification experimentally turned out to be more of an adequate approximation, in comparison with the analytically obtained linearized models, they were used. Linear models can be represented by following transfer function,

$$G(s) = \frac{H(s)}{V_p(s)} = \frac{K_1}{\tau_1 s + 1}, \quad (9)$$

where  $K_1$  and  $\tau_1$  are tank's gain and time constant, respectively.

TABLE I  
NOMINAL VALUES AND LINEAR MODELS

$i$	$H_{Ni}$ [m]	$V_{pNi}$ [V]	$G_i(s)$
1	0.077	4.4	$\frac{0.002313}{s + 0.04758}$
2	0.1665	6	$\frac{0.002627}{s + 0.04235}$
3	0.2415	7.1	$\frac{0.002642}{s + 0.03469}$

Nominal levels in the tank  $H_{Ni}$ , nominal voltages  $V_{pNi}$  and corresponding identified transfer functions are given in Table 1. Voltage deviation represent control deviations so we can write  $u(t)=v_p(t)$ . Constants for the state space plant model  $a_i$  and  $b_i$  are given in Table II.

TABLE II  
CONSTANTS FOR THE STATE SPACE SYSTEM MODEL

$i$	1	2	3
$a_i$	-0.04758	-0.04235	-0.03469
$b_i$	0.002313	0.002627	0.002642

In this article a nonlinear model is obtained by combining three linear models around 0.08 m, 0.16 m and 0.24 m. The membership functions have a triangular shape and are depicted in Fig. 2. Moreover, the predefined parameters are arbitrary function parameters, and it is assumed that they are symmetric.

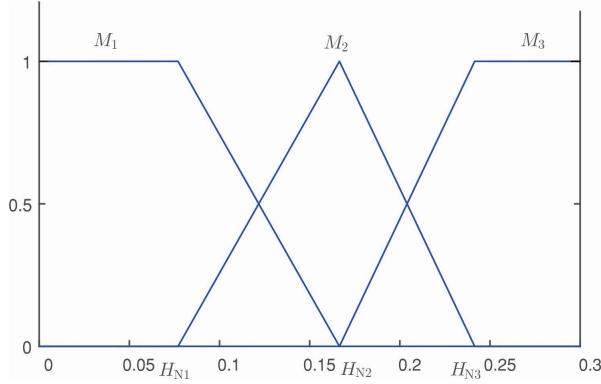


Fig. 2. Membership functions.

### III. THE WHALE OPTIMIZER

Whale Optimization Algorithm has proven to be outstanding at resolving a variety of modes, multimodal and problems that are not linear. The foremost supremacies of this algorithm, and all metaheuristic algorithms in general, are that it has random distribution, which avoids getting stuck in the local minimum. WOA was first suggested by Seyedali Mirjalili and Andrew Lewis in [17]. The paper was inspired by a dozen whales, working together in a sophisticated way to harvest the krill. A curtain of bubbles and the hunting horn hold a secret to an indigenous ways of fishing - the bubble net feeding. The leader whale (bubble blower) dives a couple of meters deep into the ocean. It's his job to find the fish. The rest follow information. Each takes exactly the same position in every lunch. Once the leader has located the fish, he blows a net of bubbles in a spiral shape, which completely encircles the prey. Another whale calls to synchronize the group. Panicked by the hearing sound of the blinding bubbles barrier the fish herds will be captured, allowing whales to swim up toward them. The hunt contains three phases. The first one is encircling the prey by defining the best search agent and updating the position of others. The mathematical model of this phase is proposed using the distance vector  $\mathbf{D}$  and vector  $\mathbf{X}$  which is used to update the position:

$$\mathbf{D} = |\mathbf{C}\mathbf{X}'(t) - \mathbf{X}(t)|, \quad (10)$$

$$\mathbf{X}(t+1) = \mathbf{X}'(t) - \mathbf{AD}, \quad (11)$$

$$A = 2ar - a, \quad (12)$$

where  $t$  indicates the current iteration,  $\mathbf{A}$  and  $\mathbf{C}$  indicate coefficient vectors; adjusting those values improves positions around the best agent  $\mathbf{a}$ ; where  $a$  is a random value between [0,2] which linearly decreases during the course of iterations.  $\mathbf{X}'$  is the position vector of the best solution obtained so far and  $\mathbf{X}$  is the position vector. The second phase – exploration is given either with shrinking encircling mechanism (defining the new position of the searching agent using  $\mathbf{A}$ ), or with spiral updating position (first calculation distance between whale and prey using helix – based movement. The new position of the agent is located between the current best agent and the original position. The function for this approach is:

$$\mathbf{X}(t+1) = \begin{cases} \mathbf{X}'(t) - \mathbf{AD} & \text{if } p < 0.5 \\ \mathbf{D}' - e^{bl} \cos(2\pi l) + \mathbf{X}'(t) & \text{if } p \geq 0.5 \end{cases}, \quad (13)$$

where  $p$  is a random number in [0,1],  $b$  is a constant for defining the shape of the logarithmic spiral,  $l$  is a random number in [-1,1] and  $\mathbf{D}'$  indicates the distance of the  $i$ -th whale from the prey [17]. The third one, exploration phase, is based on a random search, that provides a good balance between the last two phases. This is called adoptive variation that depends of the value search vector  $\mathbf{A}$ .

### IV. TAKAGI-SUGENO MODEL OPTIMIZATION

In the Fig. 2 we observe the beforehand mentioned TS model which was obtained based on the symmetric shape of the membership functions. The configuration of the functions is triangular and the centers of the membership functions are located in the selected nominal points in which the linear models are identified. However, in order to achieve a better approximation of the non-linear characteristics and overall behavior of the plant, a more adequate approximation of the non-linear model is presented by adjusting the parameters of the membership functions. We can view the parameters as the width of the membership functions. So in conclusion, in this case we only optimized the parameters that were located in the rule premise. Moreover, the mentioned TS parameters are all coded into one whale, per say one agent, that is presented with a vector which contains the premise parameters, in our case it has four parameters. In the proposed WOA algorithm the population is set to 20, while the total number of iterations is set to 30. Furthermore, in this optimization method, one agent represents one potential optimal fuzzy model. The mean square error (MSE) is taken as an objective function and it can be calculated as

$$\text{MSE} = \frac{1}{n} \sum_{i=1}^n (y(i) - y_m(i))^2, \quad (14)$$

where  $n$  is the number of data points,  $y(i)$  is the measured output of the plant,  $y_m(i)$  is the output of the model.

A dataset for the learning process of the WOA algorithm, in other words for the optimization of the TS model, is obtained from the plant operation in 1600 seconds. All of the parameter values that were used in the implementation of the WOA were taken from the original paper [17]. In the aim of identification we bring the input voltage which has a shape as depicted in Fig. 3.

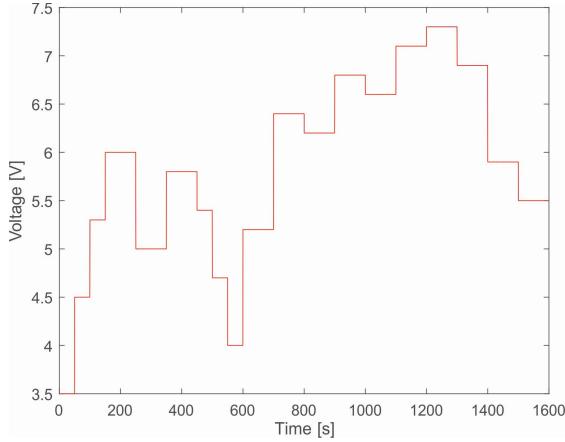


Fig. 3. Voltages used for model optimization.

There it should be observed that the values are between the nominal voltages, this is done in order to cover the range of interest. Optimized membership functions are shown on Fig. 4.

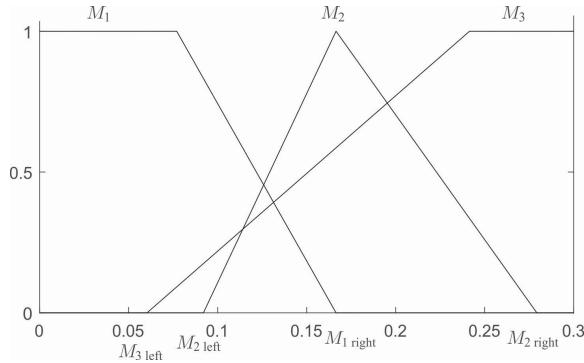


Fig. 4. Optimized membership functions.

where  $M_{2\text{left}}=0.092$ ,  $M_{1\text{right}}=0.1665$ ,  $M_{3\text{left}}=0.0603$ ,  $M_{2\text{right}}=0.2793$ . Comparison of the TS model based on initial membership functions and the TS model based on optimized membership functions is showed on Fig. 5.

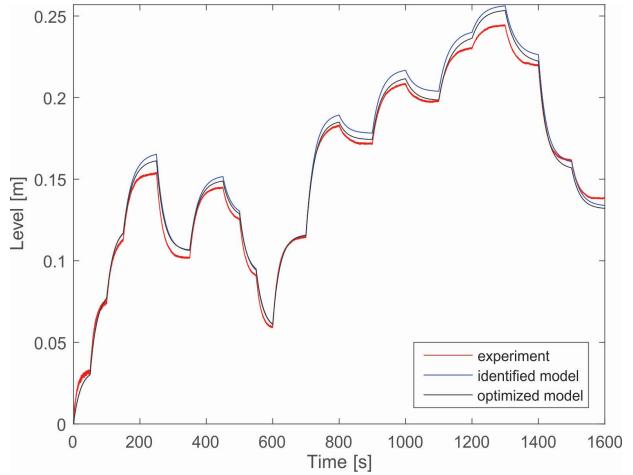


Fig. 5. Comparison of identified and optimized TS model.

## V. CONTROL SYSTEMS DESIGN

The main control objective is to maintain the liquid level in tank at a desired level by adjusting the pump flow rate. The requirement is that the control systems for all three operating points should satisfy the following specifications: the steady state error should be zero; the percentage overshoot in tank has to be less than 5%, the  $\text{PO} \leq 5\%$ ; the settling time for the tank should be less than 30 seconds,  $T_s \leq 30 \text{ sec}$ .

The history of the purported PDC was set in motion with a model-based design procedure proposed by Kang and Sugeno, [20]. The PDC proposes a procedure to design a fuzzy controller from a given TS fuzzy model. Furthermore, each control rule is designed from the corresponding rule of a TS fuzzy model during the construction of a PDC design. As a consequence, the designed fuzzy controller shares the same fuzzy sets as the fuzzy model in the premise parts. For our concrete model in this paper we have defined for each of the linearized models a linear PI controller. The control rule  $i$  of the fuzzy controller via PDC is:

IF  $x(t)$  is about  $H_{Ni}$ , THEN the controller is  $C_i$ .  
The overall fuzzy controller is represented by

$$C = \frac{\sum_{i=1}^3 w_i(x(t)) C_i}{\sum_{i=1}^3 w_i(x(t))} = \sum_{i=1}^3 h_i(x(t)) C_i, \quad (15)$$

where  $C_i$  are PI controllers defined in a complex domain as  $C_i = K_{P_i} + K_{I_i}/s$ ,  $i=1,2,3$ . Parameters for all three controllers  $C_i$ , were obtained based on the linear theory and according to the control objective. Meanwhile, proportional and integral gains for all of the above stated controllers are given in Table 3.

TABLE III  
PARAMETERS OF CONTROLLERS

$i$	1	2	3
$K_{P_i}$	94.72	85.389	87.803
$K_{I_i}$	16.139	14.21	14.129

## VI. FURTHER EXPERIMENTAL RESULTS

In order to display the effectiveness of the utilized methods we performed a couple of experiments and verified the efficiency of the optimization and identification, subsequently. Thus, the Fig. 6 depicts the difference between the plant response that is controlled by a local linear PI controller, which is designed to work around 0.08m, and the plant response that is controlled by the PDC. As can be seen in Fig. 6 the PDC achieves a better performance than the local PI because when we are operating in the range of 0.08 m to 0.12 m, both controllers that are designed to operate around 0.08 m and 0.16 m are active.

With all that being said, the PDC achieves a better performance than the local PI cause when we are operating in the range of 0.08 m to 0.12 m, both controllers that are designed to operate around 0.08 m and 0.16 m are active, see Fig. 2.

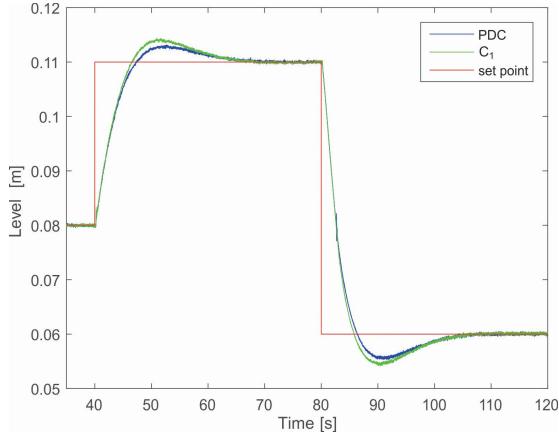


Fig. 6. Comparison of PDC and PI controller system around 0.08 m.

The same analysis applies to the operation of the plant around 0.16 m, which is shown in the Fig. 7. In this case with the PDC, all three local linear controllers that are designed to operate around 0.08 m, 0.16 m and 0.24 m are active, as can be seen from the Fig. 2.

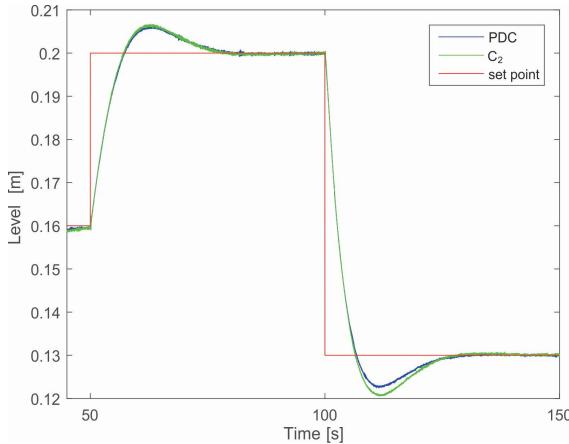


Fig. 7. Comparison of PDC and PI controller system around 0.16 m.

PDC controller was compared with the specifically designed controller for the nominal point 0.12 m. As to say,

that the most onerous challenge for the PDC is precisely this, because that point is the most further from the operation points of local linear controllers, which are designed to operate around 0.08 m and 0.16 m.

The requirements for this local linear PI controller are the same. In the same way we obtained parameters  $K_{p_i}=67.384$  and  $K_{i_i}=12.081$ . A juxtapose of the operation of this local PI controller with the PDC is shown in Fig. 8.

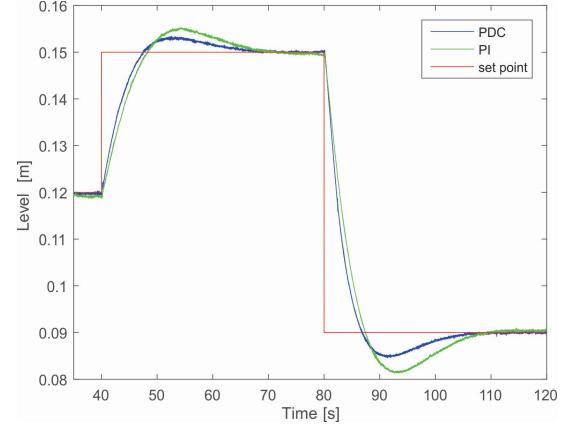


Fig. 8. Comparison of PDC and PI controller system around 0.12 m.

A smaller overshoot and settling time, were obtained when the plant was controlled using a PDC that contains information about the optimized model, than when the plant was controlled by a PDC with initial membership functions. In order for our results to be observed better, the filtered responses are shown in Fig. 9. The same moving average filter with a span of 30 data points has been used for both of the signals.

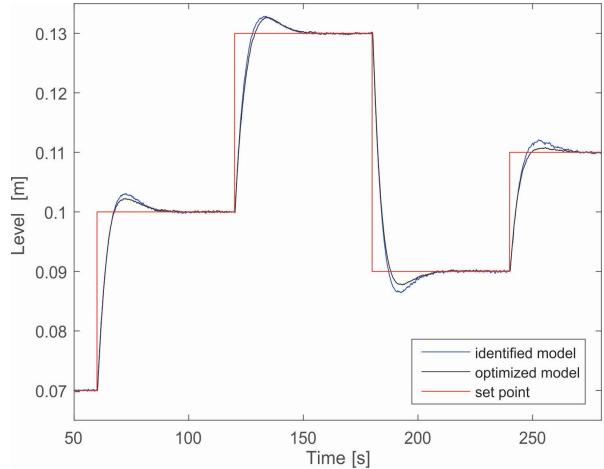


Fig. 9. Comparison of PDCs with an identified and optimized plant model (experiment with moving average filter).

Comparison of control signals of PDCs with an identified and optimized plant model is shown on the Fig.10.

Comparisons of system response percentage overshoots and settling times of system response, for the identified and optimized TS model, are shown in Table 4, Table 5, respectively.

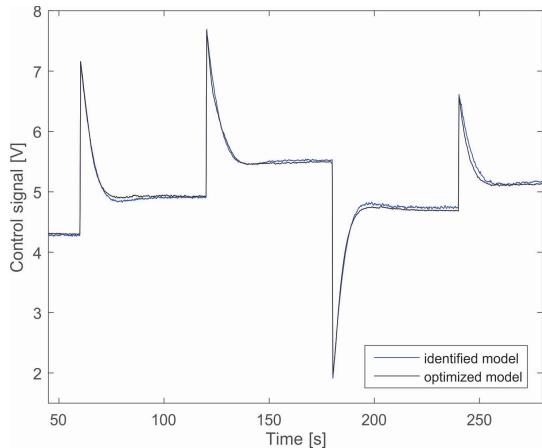


Fig. 10. Comparison of control signals of PDCs with an identified and optimized plant model.

TABLE IV  
PERCENTAGE OVERRUSH FOR DIFFERENT STEP RESPONSES

Step [m]	0.09-0.13	0.13-0.09	0.1-0.13	0.07-0.1
Id. [%]	10	8.75	9.3	10
Op. [%]	4	5.5	8.7	7.3

TABLE V  
SETTLING TIME FOR DIFFERENT STEP RESPONSES

Step [m]	0.09-0.13	0.13-0.09	0.1-0.13	0.07-0.1
Id. [s]	24	25.3	22.7	26
Op. [s]	23.6	25	21	21

## VII. CONCLUSION

Initially, in this paper, the mathematical model of the liquid level system was obtained experimentally. Further, TS fuzzy model was obtained based on three identified local linear models. Regardless of the superiority in “catching” the nonlinear behavior of the plant, TS model was optimized using WOA metaheuristic and verified by comparing with the original. Consequently, based on the given requirements three local linear PI controllers were designed. Then, by using the PDC method, two fuzzy controllers were designed based on identified and optimized TS model. The designed controllers, which were based on the PDC, were implemented in an experimental setup in order to prove their performance. Given the very satisfying results, the developed TS models are tremendously simple and consists only of three fuzzy rules. Using the whale optimization method the accuracy of TS model is improved, which enlarges the efficiency of TS based PDC controller. Future research will focus on exploiting these possibilities in terms of using more fuzzy rules.

## ACKNOWLEDGMENT

This paper was conceived within the research on the project of the Ministry of Education, Science and Technological Development of the Republic of Serbia (contract no. 451-03 -68 / 2020-14 / 200105), 2020. This work was financially supported by the Ministry of Education, Science and Technological Development of the Serbian Government, Grant TR-35029 (2018-2020).

## REFERENCES

- [1] A. S. Tijjani, M. A. Shehu, A. Alsabari, Y. A. Sambo, N. L. Tanko, “Performance analysis for coupled-tank system liquid level control using MPC, PI and PI-plus-feedforward control scheme,” *Journal of Robotics and Automation*, vol. 1, no. 1, pp. 42-53, June, 2017.
- [2] H. Wong, V. Kapila, M. de Queiroz, “Experimental validation of a nonlinear backstepping liquid level controller for a state coupled two tank system,” *Control Engineering Practice*, vol. 13, no. 1, pp. 27-40, January, 2005.
- [3] T. L. Mien, “Liquid level control of coupled-tank system using fuzzy-PID controller,” *International Journal of Engineering Research & Technology*, vol. 6, no. 11, pp. 459-464, November, 2017.
- [4] T. Takagi, M. Sugeno, “Fuzzy identification of systems and its applications to modeling and control,” *IEEE Transactions on Systems, Man, Cybernetics*, New York, New York, vol. 15, pp. 116-132, 1985.
- [5] M. S. Sadeghi, B. Safarinejad, A. Farughian, “Parallel distributed compensator design of tank level control based on fuzzy Takagi-Sugeno model,” *Applied Soft Computing*, vol. 21, pp. 280-285, August, 2014.
- [6] S. Yordanova, “Fuzzy logic approach to coupled level control,” *Systems Science & Control Engineering*, vol. 4, no. 1, pp. 2015-222, September, 2016.
- [7] A. D. Marka, M. Seidi, “Performance-oriented parallel distributed compensation,” *Journal of the Franklin Institute*, vol. 348, pp. 1231-1244, 2011.
- [8] T. Taniguchi, K. Tanaka, K. Yamafuji, H. O. Wang, “Nonlinear model following control via Takagi-Sugeno fuzzy model,” *Proceedings of the American Control Conference*, San Diego, California, vol. 3, pp. 1837-1841, 1999.
- [9] M. Polansky, “Advanced robust PDC fuzzy control of nonlinear systems,” *International Journal of Computer and Information Engineering*, vol. 1, no. 11, pp. 3715-3720, 2007.
- [10] F. S. Gharehchopogh, H. Gholizadeh, “A comprehensive survey: Whale optimization algorithm and its applications,” *Swarm and Evolutionary Computation*, vol. 48, pp. 1-24, August, 2019.
- [11] O. Cordon, F. Herrera, “A two-stage evolutionary process for designing TSK fuzzy rule-based systems,” *IEEE Transactions on Systems, Man, and Cybernetics - Part B: Cybernetics*, vol. 29, no. 6, pp. 703-715, December, 1999.
- [12] S. H. Tsai, Y. W. Chen, “A novel identification method for Takagi-Sugeno fuzzy model,” *Fuzzy Sets and Systems*, vol. 338, pp. 117-135, May, 2018.
- [13] R. E. Precupa, M. C. Sabaua, E. M. Petriub, “Nature-inspired optimal tuning of input membership functions of Takagi-Sugeno-Kang fuzzy models for Anti-lock Braking Systems,” *Applied Soft Computing*, vol. 27, pp. 575-589, 2015.
- [14] L. Lin, F. Guo, X. Xie, B. Luo, “Novel adaptive hybrid rule network based on TS fuzzy rules using an improved quantum-behaved particle swarm optimization”, *Neurocomputing*, vol. 149, pp. 1003-1013, February, 2015.
- [15] S. Rastegar, R. Araújo, J. Mendes, “Online identification of Takagi-Sugeno fuzzy models based on self-adaptive hierarchical particle swarm optimization algorithm,” *Applied Mathematical Modelling*, vol. 45, pp. 606-620, May, 2017.
- [16] M. Ghanamijaber, “A hybrid fuzzy-PID controller based on gray wolf optimization algorithm in power system,” *Evolving Systems*, vol. 10, pp. 273-284, 2018.
- [17] S. Mirjalili, A. Lewis, “The whale optimization algorithm,” *Advances in Engineering Software*, vol. 95, pp. 51-67, May, 2016.
- [18] M. Z. Kamali, K. Nallasamy, K. Ratnavelu, “Takagi-Sugeno fuzzy modelling of some nonlinear problems using ant colony programming,” *Applied Mathematical Modelling*, vol. 48, pp. 635-654, August, 2017.
- [19] N. Katal, P. Kumar, S. Narayan, “Optimal PID controller for coupled-tank liquid-level control system using bat algorithm,” *International Conference on Power, Control and Embedded Systems*, Allahabad, pp. 1-4, 2014.
- [20] M. Sugeno, G.T. Kang, “Structure identification of fuzzy model,” *Fuzzy sets and systems*, vol. 28, pp. 15-33, 1988.

# Modelling and Control of a Series Direct Current (DC) Machines Using Feedback Linearization Approach

Mitra Vesović, Radiša Jovanović, Lara Laban, Vladimir Zarić

**Abstract**—The nonlinear feedback control system applied to the direct current - DC motor is proposed in this research. Nonlinear mathematical model has been obtained using dead zone, Coulomb and viscous friction. The system stability has been analyzed using Lyapunov stability theory. The effectiveness and the comparison of the performance between linear and nonlinear control algorithm have been validated using Matlab/Simulink software. From the conclusions, based on the simulation and experimental results that have been provided, it is easy to see that nonlinear control systems are more suitable and have a better reach for controlling position. The validity of using feedback linearization in DC motors has been proven.

**Index Terms**—feedback linearization; nonlinear systems; nonlinear control; identification

## I. INTRODUCTION

Position control in a direct current motor (DC) has been one of the most fundamental and challenging tasks, that has been largely studied for decades. Many studies have been done to model electrical machines. For example, serial DC motor has often been modeled as linear object. On the other hand, models in which motor current or flux are found as essential parameters are considered to be nonlinear [1]. This paper presents the design and implementation concerning both, linear and nonlinear models for the system. They are obtained for identification and control purposes. The major nonlinearities in the system, such as Coulomb friction and dead zone, are investigated and integrated in the nonlinear model [2]. In the different types of application accurate control of position in DC machines is a great challenge for engineers. Disparate controllers have been proposed to lead the position of DC machines into the desired value. For example Proportional-Integral-Derivative (PID) controller is a popular controller in industries due to simple structure, low cost and easy to implementation. It provides reliable performance for the system if PID parameter is identified properly. But it suffers due to lack of robustness [1]. The linear approximation, of the nonlinear state space representation of the series DC motor,

Mitra Vesović is with the Faculty of Mechanical Engineering, University of Belgrade, Kraljice Marije 16, 11120 Belgrade, Serbia (e-mail: [rjovanovic@mas.bg.ac.rs](mailto:rjovanovic@mas.bg.ac.rs))

Radiša Jovanović is with the Faculty of Mechanical Engineering, University of Belgrade, Kraljice Marije 16, 11120 Belgrade, Serbia (e-mail: [mvesovic@mas.bg.ac.rs](mailto:mvesovic@mas.bg.ac.rs))

Lara Laban is with the Faculty of Mechanical Engineering, University of Belgrade, Kraljice Marije 16, 11120 Belgrade, Serbia (e-mail: [llaban@mas.bg.ac.rs](mailto:llaban@mas.bg.ac.rs))

Vladimir Zarić is with the Faculty of Mechanical Engineering, University of Belgrade, Kraljice Marije 16, 11120 Belgrade, Serbia (e-mail: [vzaric@mas.bg.ac.rs](mailto:vzaric@mas.bg.ac.rs))

around the equilibrium point and PI controller design the tracking performance is deteriorated in the periods in which the speed is reduced. This is due to the fact that the input signal ( $t$ ) is limited to a minimum of 0 [V]. That is, in this condition the motor is actually operating in open loop [3].

Besides linear, there are plenty of nonlinear controllers: the fuzzy logic and genetic – based new fuzzy models [4], artificial neural networks [5], adaptive control technique [6], and others.

It is important to make this comparison to find out under what conditions a technique presents a superior performance over the other one and thus have the certainty when it is useful to implement nonlinear controllers, which have greater complexity [7].

Modelling a nonlinearity is often a very complicated challenge. One of the first steps in the synthesis of a control system is to create a mathematical model, because it saves time and it brings the cost-effectiveness.

The main objective of this research is the development and later implementation of a nonlinear control system, by the feedback linearization method, for a laboratory installed DC motor, SRV02 Rotary Servo Base Unit, which has been considered as a single-input-single-output (SISO) system.

Feedback linearization is an approach to nonlinear control design which has attracted a great deal of research interest in recent years. By a combination of a nonlinear transformation and state feedback (feedback linearization), the nonlinear control design is reduced to designing a linear control law [8]. The central idea of the approach is to algebraically transform a nonlinear system dynamics into a (fully or partly) linear one, so that linear control techniques can be applied. This differs entirely from conventional linearization in that feedback linearization is achieved by exact state transformations and feedback, rather than by linear approximations of the dynamics [9]. This technique has been successfully implemented in many applications of control, such as industrial robots, high performance aircraft, helicopters and biomedical dispositifs, more tasks used the methodology are being now well advanced in industry [10].

## II. LINEAR MODEL OF SYSTEM DYNAMICS

Constructing an accurate model is a pivotal stage in practical control problems. An appropriately developed system model is essential for reliability of the designed control. A DC series motor is an example of a simple, controlled process that can serve as a vehicle for the evaluation of the performance of the various controllers [4].

A schematic diagram of the DC motor is given in Fig. 1.

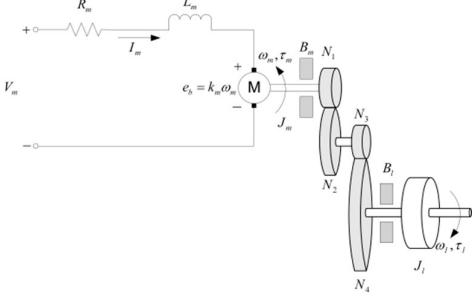


Fig. 1. SRV02 DC motor armature circuit and gain train [11]

The equations that describe the motor electrical components are as follows:

$$V_m(t) = R_m I_m(t) + L_m \frac{dI_m(t)}{dt} + e_b(t) \quad (1)$$

$$e_b(t) = k_m \omega_m(t) \quad (2)$$

where  $V_m$ ,  $e_b$ ,  $k_m$  and  $\omega_m$  are motor voltage, back electromotive voltage, back electromotive voltage constant and speed of the motor shaft, respectively. Since the motor inductance  $L_m$  is much less than its resistance  $R_m$ , it can be ignored [11]. Solving the system of equations for motor current  $I_m$ , we get an electrical equation of DC motor:

$$I_m(t) = \frac{V_m(t) - k_m \omega_m(t)}{R_m}. \quad (3)$$

The linear model can be obtained using the Second Newton's Law of Motion and connection between moment of inertia, viscous friction constants, and torque of load and motor:

$$J_l \frac{d\omega_l(t)}{dt} + B_l \omega_l(t) = \tau_l(t) \quad (4)$$

$$J_m \frac{d\omega_m(t)}{dt} + B_m \omega_m(t) + \tau_{ml}(t) = \tau_m(t) \quad (5)$$

so the mechanical equation is:

$$J_{eq} \frac{d\omega_l(t)}{dt} + B_{eq} \omega_l(t) = \eta_g K_g \tau_m(t) \quad (6)$$

where  $J_{eq}$  and  $B_{eq}$  are total moment of inertia and damping term.  $\eta_g$  and  $K_g$  are, respectively, the gearbox efficiency and the total gear ratio.

Combining electrical and mechanical equations, assuming that motor torque is proportional to the voltage, the final equation becomes:

$$\left( \frac{d}{dt} \omega_l(t) \right) J_{eq} + B_{eq,v} \omega_l(t) = A_m V_m(t) \quad (7)$$

where the equivalent damping term is given by:

$$B_{eq,v} = \frac{\eta_g K_g^2 \eta_m k_t k_m + B_{eq} R_m}{R_m} \quad (8)$$

and the actuator gain equals:

$$A_m = \frac{\eta_g K_g \eta_m k_t}{R_m} \quad (9)$$

Linear mathematical model that defines the relationship between voltage and angular position is:

$$J_{eq} \ddot{\theta}_l(t) + B_{eq,v} \dot{\theta}_l(t) = A_m V_m(t). \quad (10)$$

Choosing  $y = \theta_l$  as output variable and  $u = V_m$  as input signal, state equation of the system is obtained as follows:

$$J_{eq} \ddot{y}(t) + B_{eq,v} \dot{y}(t) = A_m u(t). \quad (11)$$

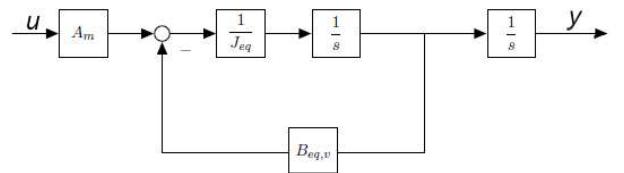


Fig. 2. Block diagram of a linear system

### III. EXPERIMENTAL VERIFICATION OF THE OBTAINED LINEAR MATHEMATICAL MODEL

Responses of the system represented with the block diagram in the Fig. 2 are shown in the Fig. 3 and Fig. 4. After recording the responses of the object, comparisons were made with the responses obtained by simulations of the linear model, for step and sinusoidal inputs.

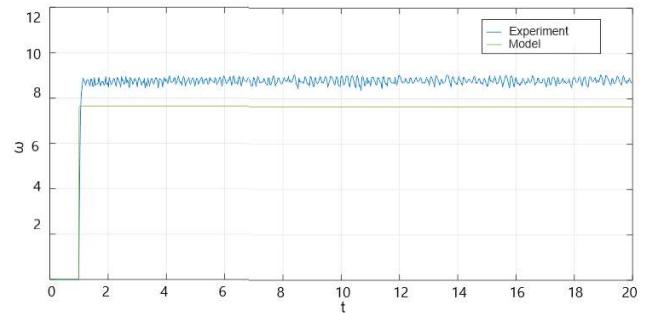


Fig. 3. Experimental results: comparison between real and model data for step input

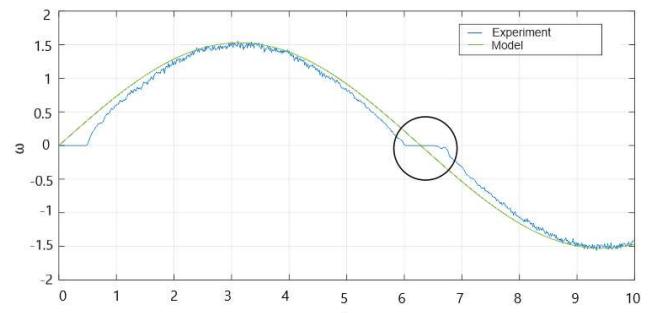


Fig. 4. Experimental results: comparison between real and model data for sinusoidal input

From this simulated example, an important conclusion can be drawn. Simulated linear model of the plant does not match well response of the real system. It is obvious that mathematical model of the series DC motor is nonlinear.

#### IV. FEEDBACK LINEARIZATION

In this section, the conditions for the linearizing transformation and nonlinear feedback allowing the DC motor to be controlled are outlined. Of particular interest will be the coordinate transformation also known as diffeomorphism, and the feedback law which will allow it to be accomplished.

Consider the single – input – single – output nonlinear SISO system [12]:

$$\begin{aligned}\dot{x} &= f(x) + g(x)u \\ y &= h(x)\end{aligned}\quad (12)$$

where  $f(x)$ ,  $g(x)$  and  $h(x)$  are sufficiently smooth in a domain  $D \subset R^n$  (the mapping  $f : D \rightarrow R^n$ ,  $g : D \rightarrow R^n$  are vector fields on  $D$ ) and  $\dot{x} = [x_1 \ x_2 \ \dots \ x_n]^T$  is a state vector. It is necessary to find a state feedback control  $u$ , that transforms the nonlinear system into an equivalent linear system. Clearly, generalization of this idea is not possible in every nonlinear system: there must be a certain structural property that allows performing in such a manner of cancellation.

Using feedback to cancel nonlinearities requires the nonlinear state equation to have a structure:

Definition [12]:

$$\dot{x} = Ax + B\gamma(x)[u - \alpha(x)] \quad (13)$$

where  $A$  is  $n \times n$  and  $B$  is  $n \times p$  matrix, the functions  $\alpha : R^n \rightarrow R^p$ ,  $\gamma : R^n \rightarrow R^{p \times p}$  are defined on domain  $D \subset R^n$  that contains the origin. Furthermore, two conditions must be satisfied. The first one is that the pair  $(A, B)$  must be controllable. The second one is that  $\gamma(x)$  must be nonsingular for all  $x \in D$ . This is consequence of the control law form:  $u = \alpha(x) + \frac{1}{\gamma(x)}v$  that provides a new control signal  $v$ .

Even if the state equation does not have the structure (13), sometimes it is possible to execute feedback linearization for another choice of variables. Therefore, a more comprehensive definition is given [12]:

A nonlinear system:

$$\dot{x} = f(x) + G(x)u \quad (14)$$

where  $f : D \rightarrow R^n$  and  $G : D \rightarrow R^{n \times p}$  are sufficiently smooth on a domain  $D \subset R^n$ , is said to be feedback linearizable (or input – state linearizable) if there exist a diffeomorphism  $T : D \rightarrow R^n$  such that  $D_z = T(D)$  contains the origin and the change of variables  $z = T(x)$  transforms the system (14) into the form:

$$\dot{z} = Az + B\gamma(x)[u - \alpha(x)] \quad (15)$$

with  $(A, B)$  controllable and  $\gamma(x)$  nonsingular for all  $x \in D$ .

#### V. DETERMINATION OF RELATIVE DEGREE

The relative degree of a linear system is defined as the difference between the poles (degree of the transfer function's denominator polynomial number) and zeros (degree of its numerator polynomial). To extend this concept to nonlinear systems more mathematical treatment will be needed. The following definition is given and repeated here for completeness:

Definition [13]: Given the Single Input – Single Output System, SISO, outlined in (12), it is said to have relative degree  $r$  at a point  $x_0$  if:

- i)  $L_g L_f^k h(x) = 0$  for all  $x$  in a neighborhood of  $x_0$  and all  $k < r - 1$
- ii)  $L_g L_f^{r-1} h(x) \neq 0$

The terms  $L_g$  and  $L_f^k$  represent the Lie derivative of  $h(x)$  taken along  $g(x)$  and  $k - times$  along  $(x)$ , respectively.

#### VI. NONLINEAR MATHEMATICAL MODEL

The nonlinear mathematical model of the DC motor was obtained considering the speed dependent friction nonlinearity. Reference [14] shows that in this case the nonlinear mathematical model of DC motor can be adopted as follows:

$$J_{eq}\ddot{\theta}_l + T_{st}(\dot{\theta}_l) + B_{eq,n}\dot{\theta}_l = A_m V_m \quad (16)$$

TABLE I  
THE NUMERICAL VALUES OF THE PLANT PARAMETERS

Parameters	Values and units
$J_{eq}$	$0.0021 \text{ kgm}^2$
$R_m$	$2.6 \Omega$
$k_t$	$0.0077 \text{ Nm/A}$
$\eta_m$	0.69
$\eta_g$	0.9
$K_g$	70

The part of the obtained friction curve  $T_{st}(\dot{\theta}_l)$ , for low angular velocity values, where the Stribeck effect is dominant, is shown in Fig. 5. It is assumed that friction characteristics are symmetrical, for negative and positive values of angular velocity. Applying standard optimization techniques with Matlab, the friction parameters were obtained, as follows:

$$\begin{aligned}T_{st} &= 0.0174 \operatorname{sgn}(\dot{\theta}_l) + \\ &0.0087 e^{-\frac{\dot{\theta}_l}{0.064}} \operatorname{sgn}(\dot{\theta}_l), \quad B_{eq,v} = 0.0721\end{aligned}\quad (17)$$

In order to overcome the jump discontinuity of the proposed friction model, at  $\dot{\theta}_l = 0$ , that jump is replaced by a line of finite slope, up to a very small threshold  $\epsilon$ , as is shown in Fig. 3 [14].

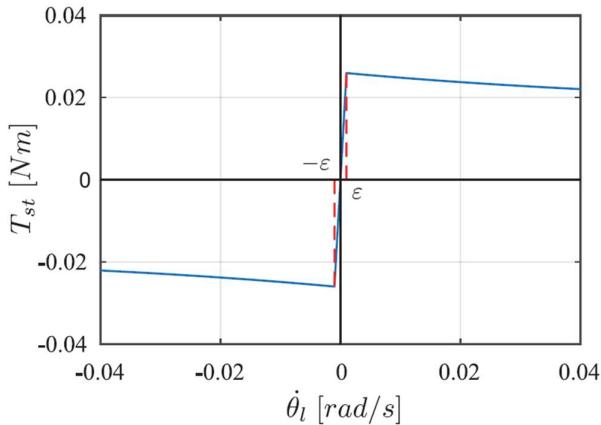


Fig. 5. Friction characteristics of DC motor [14]

This the line of finite slope will be used only for comparison with the hyperbolic tangent function (Fig. 6), because method of feedback linearization requires differentiable functions (as can be seen from the given definitions in the previous section). In this way only Coulomb and viscous friction is modeled and static friction is neglected. Choosing  $x_1 = \theta_l$ ,  $x_2 = \dot{\theta}_l$  as state variables,  $y = \theta_l$  as measured variable and  $u = V_m$  as control variable and denoting nonlinearity by  $f(\mathbf{x})$ , state equation of the system was obtained as follows:

$$\begin{aligned}\dot{\mathbf{x}} = \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} &= \begin{bmatrix} 0 & 1 \\ 0 & -\frac{B_{eq,n}}{J_{eq}} \end{bmatrix} \mathbf{x} + \begin{bmatrix} 0 \\ -1 \end{bmatrix} f(\mathbf{x}) \\ &+ \begin{bmatrix} 0 \\ \frac{A_m}{J_{eq}} \end{bmatrix} u\end{aligned}\quad (18)$$

$$y = [1 \ 0] \mathbf{x} \quad (19)$$

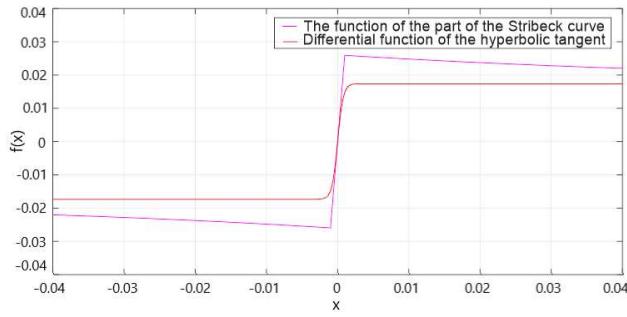


Fig. 6. Differential function of the hyperbolic tangent

To ensure that this model is an equivalent representation of the original system, an experiment was performed, with the results shown below on Fig. 5 for step and Fig. 6 for sinusoidal response.

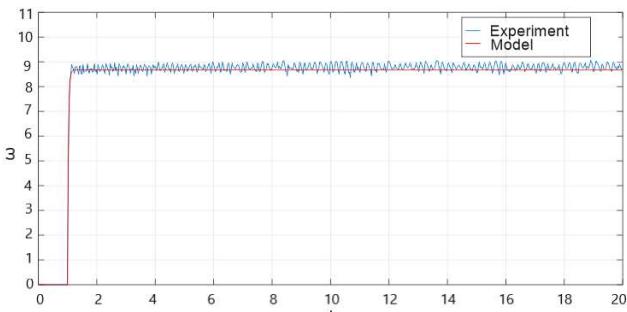


Fig. 5. Experimental results: comparison between real and model data for step input

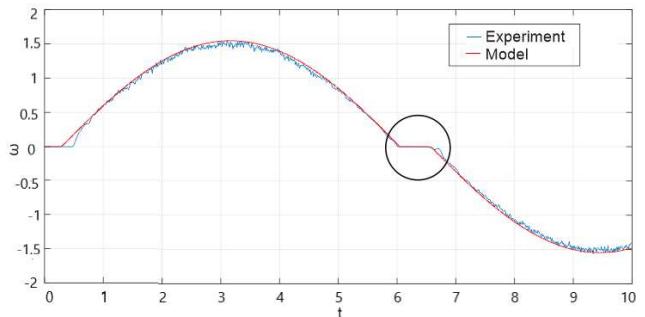


Fig. 6. Experimental results: comparison between real and model data for sinusoidal input

## VII. EXPERIMENTAL RESULTS

Applying Definition [12] to the system (18) – (19) yields:

$$A = \begin{bmatrix} 0 & 1 \\ 0 & -\frac{B_{eq,n}}{J_{eq}} \end{bmatrix} \quad (20)$$

$$B = \begin{bmatrix} A_m \\ \frac{A_m}{J_{eq}} \end{bmatrix} \quad (21)$$

$$\alpha(\mathbf{x}) = \frac{J_{eq}}{A_m} f(\mathbf{x}) \quad (22)$$

$$\gamma(\mathbf{x}) = 1. \quad (23)$$

First condition is met:

$$U = [B \ AB \ A^2B \ \dots \ A^{n-1}B]. \quad (24)$$

Order of system is  $n = 2$  and, because  $\text{rank } U = n$ , the pair  $(A, B)$  is controllable:

$$U = [B \ AB] = \begin{bmatrix} 0 & \frac{A_m}{J_{eq}} \\ \frac{A_m}{J_{eq}} & -\frac{B_{eq,n}}{J_{eq}^2} \end{bmatrix}. \quad (25)$$

System transformation is not required and all functions are smooth and differentiable.  $\gamma(\mathbf{x})$  is not equal to zero, so the second condition is also met. With both conditions fulfilled feedback linearization is allowed.

The first derivative of the system (18) – (19) output does not depend on the control signal, which means that the relative degree of the system is not 1:

$$\dot{y} = L_f h(\mathbf{x}) + L_g h(\mathbf{x}) u. \quad (26)$$

$$L_g h(\mathbf{x}) = 0 \text{ and } L_f h(\mathbf{x}) = x_2 \quad (27)$$

$$\ddot{y} = \dot{x}_2 = -\frac{B_{eq,n}}{J_{eq}} x_2 - f(\mathbf{x}) + \frac{A_m}{J_{eq}} u \quad (28)$$

$$L_f^2 h(\mathbf{x}) = -\frac{B_{eq,n}}{J_{eq}} x_2 - f(\mathbf{x}) \quad (29)$$

$$L_g L_f h(\mathbf{x}) = \frac{A_m}{J_{eq}} \quad (30)$$

Conclusion is that relative degree of this system is equal to the system order  $r = 2$ . The desired time – domain specifications for controlling the position of the load shaft are:  $PO = 0\%$ ,  $t_s \leq 2.3$  s. Choosing the control signal in the following form:

$$\begin{aligned} u &= \frac{1}{L_g L_f h(\mathbf{x})} [-L_f^2 h(\mathbf{x}) + v] \\ &= \frac{J_{eq}}{A_m} \left[ \frac{B_{eq,n}}{J_{eq}} x_2 + f(\mathbf{x}) + v \right] \quad (31) \end{aligned}$$

with  $= -K_0 x_1 - K_1 x_2 + K_0 x_{ref}$ , where  $K_0 = 400$ ,  $K_1 = 40$ , and  $x_{ref}$  is desired output or reference, system is linearized. Linear control is obtained in the same way, with the same coefficients, but without canceling the nonlinearity:

$$u_l = -K_0 x_1 - K_1 x_2 + K_0 x_{ref} \quad (32)$$

The experiments were performed with Quanser rotary servo motor, SRV02. This model is equipped with the optical encoder and tachometer, for motor position and speed measuring, respectively [14].

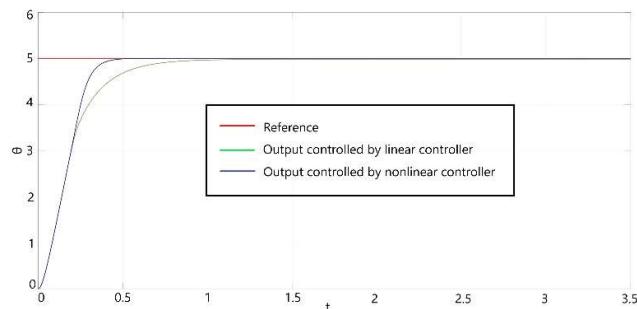


Fig. 7. Experimental results: position tracking of step signal for the linear and nonlinear controller

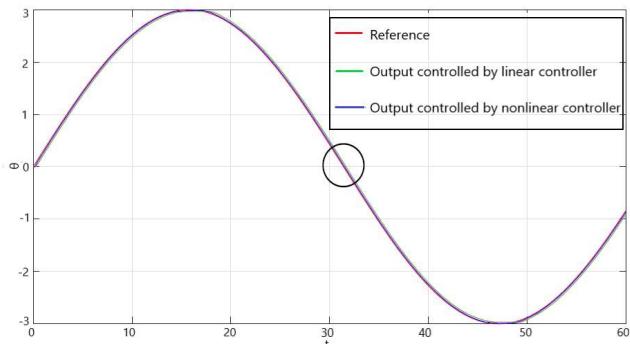


Fig. 8. Experimental results: position tracking of sine signal for the linear and nonlinear controller

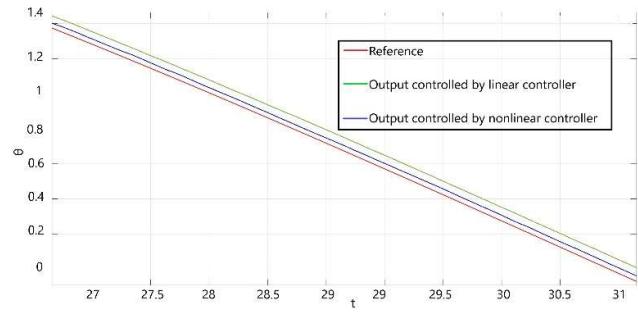


Fig. 9. Detail from Fig. 8.

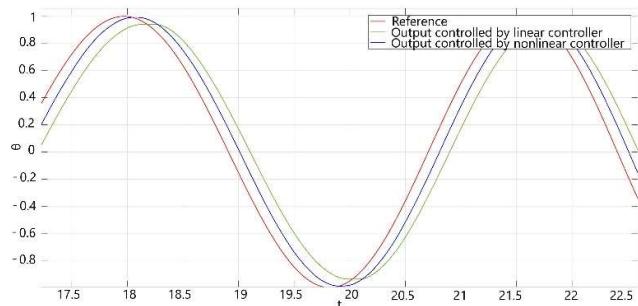


Fig. 10. Experimental results: position tracking of chirp signal for the linear and nonlinear controller

It can be observed, from the Fig. 8, Fig. 9 and Fig. 10, that the specific requirements are met. The overshoot and the settling time are in the domain of desired values. Furthermore, it is observed that the nonlinear controller is more convenient and has better achievements for position management.

## VIII. CONCLUSION

The feedback linearization technique was used for controlling the nonlinear system. The primary aim was to corroborate this method for controlling position of DC motor. First, the modelling of an object has been obtained.

After it has been experimentally confirmed that linear equations did not describe this object well enough, the nonlinear model was presented by including Stribeck model of the friction. Using the concise presentation of the feedback theory the conditions for accomplishing this technique were considered. In order to satisfy those conditions an approximation of the function, which represent nonlinearity, was found as hyperbolic tangent. Then the fulfillment of the conditions for the synthesis of the control law was proven.

At the end it could be observed, through the experiment and analysis results, that the desired response (output signal of the model reference) was tracked by the plant response. The comparison of the linear and nonlinear controller is given. The results show that the controllers, synthesized in this way, are able to satisfy desired position, but that nonlinear controller gives better outcome.

## ACKNOWLEDGMENT

This paper was conceived within the research on the project: “Integrated research in the field of macro, micro and nano mechanical engineering - Deep machine learning of intelligent technological systems in production engineering”, The Ministry of Education, Science and Technological Development of the Republic of Serbia (contract no. 451-03 - 68 / 2020-14 / 200105), 2020.

This work was financially supported by the Ministry of Education, Science and Technological Development of the Serbian Government, Grant TR-35029 (2018-2020).

## REFERENCES

- [1] S. K. Sarkar, S. K. Das, “High Performance Nonlinear Controller Design for AC and DC Machines: Partial Feedback Linearization Approach”, International Journal of Dynamics and Control, vol. 6, pp. 679 – 693, May, 2017.
- [2] T. Kara, I. Eker, “Nonlinear modeling and identification of a DC motor for bidirectional operation with real time experiments.” Energy Conversion and Management, vol. 45, pp. 1087–1106, 2004.
- [3] J. U. Liceaga-Castro, I. I. Siller-Alcalá, J. Jaimes-Ponce, R. A. Alcántara-Ramírez, E. A. Zamudio, “Identification and Real Time Speed Control of a Series DC Motor,” Mathematical Problems in Engineering, vol. 2017, March 2017.
- [4] D. Park, A. Kandel, G. Langholz Author, “Genetic-Based New Fuzzy Reasoning Models with Application to Fuzzy Control,” IEEE Transactions on Systems, Man, and Cybernetics, vol. 24, no. 1. January 1994.
- [5] S. Valluru, N. Singh, M. Kumar, “Implementation of NARMA-L2 neuro controller for speed regulation of series connected DC motor,” IEEE 5th India International Conference on Power Electronics (IICPE), Delhi, India, pp. 1 – 7, 2012.
- [6] T. Furuhashi, S. Sangwongwanich, S. Okuma, “A Position and Velocity Sensorless Control for Brushless DC Motors Using an Adaptive Sliding Mode Observer,” IEEE Transactions on Industrial electronics, vol. 39, no. 2, pp. 89 – 95, 1992.
- [7] C. Rengifo, N. Casas, D. Bravo, “A performance comparison of nonlinear and linear control for a DC series motor,” Revista Ciencia en Desarrollo, vol. 8, May 2017.
- [8] S. Metha, J. Chiasson, “Nonlinear Control of a Series DC Motor: Theory and Experiment,” Proceedings of the American Control Conference, Albuquerque, New Mexico, June, 1997.
- [9] J. E. Slotine, W. Li, “Feedback linearization,” in Applied Nonlinear Control, Englewood Cliffs, New Jersey, United States: P.H, 1991, ch. 6, sec. x, p. 207.
- [10] W. Ghozlane, J. Knani, “Nonlinear Control via Input-Output Feedback Linearization of a Robot Manipulator,” Advances in Science, Technology and Engineering Systems Journal, vol. 3, October, 2018.
- [11] J. Apkarian, M. Levis, H. Gurocak, Student Workbook, SRV02 Base Unit Experiment For Matlab/Simulink Users
- [12] H. K. Khalil, Nonlinear Systems; 3rd ed. New Jersey, United States: Prentice Hall, 2002
- [13] A. Isidori, Nonlinear Control Systems, 3rd ed. London, United Kingdom: Springer, 2002.
- [14] L. T. Gruijitch, Z. M. Bućevac, R. Ž. Jovanović & Z. B. Ribar (2019): Structurally variable control of Lurie systems, International Journal of Control