# INTELLI 2018

The Seventh International Conference on Intelligent Systems and Applications

ISBN: 978-1-61208-646-0

June 24 - 28, 2018

Venice, Italy

**INTELLI 2018 Editors**

Gil Manuel Magalhães de Andrade Gonçalves, Universidade Porto - Faculdade Engenharia, Portugal

# INTELLI 2018

# Forward

The Seventh International Conference on Intelligent Systems and Applications (INTELLI 2018), held between June 24, 2018 and June 28, 2018 in Venice, Italy, continued the inaugural event on advances towards fundamental, as well as practical and experimental aspects of intelligent systems and applications.

The information surrounding us is not only overwhelming but also subject to limitations of systems and applications, including specialized devices. The diversity of systems and the spectrum of situations make it almost impossible for an end-user to handle the complexity of the challenges. Embedding intelligence in systems and applications seems to be a reasonable way to move some complex tasks form user duty. However, this approach requires fundamental changes in designing the systems and applications, in designing their interfaces and requires using specific cognitive and collaborative mechanisms. Intelligence become a key paradigm and its specific use takes various forms according to the technology or the domain a system or an application belongs to.

The conference had the following tracks:

- Intelligent Systems and Applications
- InManEnv - Intelligent Manufacturing Environments

We take here the opportunity to warmly thank all the members of the INTELLI 2018 technical program committee, as well as all the reviewers. The creation of such a high quality conference program would not have been possible without their involvement. We also kindly thank all the authors who dedicated their time and effort to contribute to INTELLI 2018. We truly believe that, thanks to all these efforts, the final conference program consisted of top quality contributions.

We also gratefully thank the members of the INTELLI 2018 organizing committee for their help in handling the logistics and for their work that made this professional meeting a success.

We hope that INTELLI 2018 was a successful international forum for the exchange of ideas and results between academia and industry and to promote further progress in the area of intelligent systems and applications. We also hope that Venice, Italy provided a pleasant environment during the conference and everyone saved some time to enjoy the unique charm of the city.

**INTELLI 2018 Chairs**

**INTELLI Steering Committee**

Lars Braubach, Hochschule Bremen, Germany
Leo van Moergestel, HU University of Applied Sciences Utrecht, Netherlands
Sungshin Kim, Pusan National University, Korea
Maiga Chang, Athabasca University, Canada

**INTELLI Industry/Research Advisory Committee**

# INTELLI 2018

## Committee

### INTELLI Steering Committee

Lars Braubach, Hochschule Bremen, Germany
Leo van Moergestel, HU University of Applied Sciences Utrecht, Netherlands
Sungshin Kim, Pusan National University, Korea
Maiga Chang, Athabasca University, Canada
Sérgio Gorender, Federal University of Bahia (UFBA) & Federal University of the South of Bahia (UFSB), Brazil
Chin-Chen Chang, Feng Chia University, Taiwan
Stefano Berretti, University of Florence, Italy
Antonio Martin, Universidad de Sevilla, Spain

### INTELLI Industry/Research Advisory Committee

David Greenhalgh, University of Strathclyde, Glasgow, UK
Carsten Behn, Technische Universität Ilmenau, Germany
Paolo Spagnolo, National Research Council, Italy
Luca Santinelli, ONERA Toulouse, France
Sourav Dutta, Bell Labs, Dublin, Ireland
Floriana Gargiulo, Gemass - CNRS | University of Paris Sorbonne, Paris, France

### INTELLI 2018 Technical Program Committee

Azizi Ab Aziz, Universiti Utara Malaysia, Malaysia
Witold Abramowicz, Poznan University of Economics and Business, Poland
Giovanni Acampora, University of Naples Federico II, Italy
Zaher Al Aghbari, University of Sharjah, UAE
Gábor Alberti, University of Pécs, Hungary
Raul Alcaraz Martinez, University of Castilla-La Mancha, Spain
Ana Almeida, GECAD-ISEP-PPorto, Portugal
Rachid Anane, Coventry University, UK
Davide Bacciu, Università di Pisa, Italy
Suzanne Barber, The University of Texas at Austin, USA
Mohammadamin Barekatain, Technical University of Munich, Germany
Kamel Barkaoui, Cedric-Cnam, France
Senén Barro Ameneiro, University of Santiago de Compostela, Spain
Ana Isabel Barros, TNO, Netherlands
Carmelo J. A. Bastos-Filho, University of Pernambuco, Brazil
Carsten Behn, Technische Universität Ilmenau, Germany
Nabil Belacel, National Research Council Canada | Université de Moncton, Canada
Giuseppe Berio, IRISA | Université de Bretagne Sud, France

Stefano Berretti, University of Florence, Italy
Ana Cristina Bicharra Garcia, Federal University in Rio de Janeiro, Brazil
Francisco Bonin Font, University of the Balearic Islands, Spain
Jonathan Bonnet, IRT Saint Exupery, Toulouse, France
Lars Braubach, Hochschule Bremen, Germany
Simeon C. Calvert, Delft University of Technology, Netherlands
Erik Cambria, Nanyang Technological University, Singapore
Valerie Camps, Paul Sabatier University | IRIT, France
Carlos Carrascosa, Universidad Politécnica de Valencia, Spain
Ana Casali, FCEIA - UNR, Argentina
Chin-Chen Chang, Feng Chia University, Taiwan
Maiga Chang, Athabasca University, Canada
Amitava Chatterjee, Jadavpur University, Kolkata, India
Sharon Cox, Birmingham City University, UK
Angelo Croatti, University of Bologna, Italy
Chuangyin Dang, City University of Hong Kong, Hong Kong
Andrea D'Ariano, Università degli Studi Roma Tre, Italy
Arianna D'Ulizia, National Research Council - IRPPS, Italy
Angel P. del Pobil, Jaume I University, Spain
Francisco José Domínguez Mayo, University of Seville, Spain
Elena Niculina Dragoi, "Gheorghe Asachi" Technical University, Iasi, Romania
Sourav Dutta, Bell Labs, Dublin, Ireland
Mudasser F. Wyne, National University, USA
Camille Fayollas, ICS-IRIT | University Toulouse 3 Paul Sabatier, France
Alena Fedotova, Bauman Moscow State Technical University, Russia
Manuel Filipe Santos, University of Minho, Portugal
Adina Magda Florea, University Politehnica of Bucharest, Romania
Rita Francese, Università degli Studi di Salerno, Italy
Marta Franova, CNRS | LRI & INRIA, Orsay, France
Simone Gabbriellini, University of Brescia, Italy
Claudio Gallicchio, Universita' di Pisa, Italy
Floriana Gargiulo, University of Namur, Belgium
Sanaa Ghouzali, King Saud University, Riyadh, Saudi Arabia
Fatemeh Golpayegani, Trinity College Dublin, Ireland
Sérgio Gorender, Federal University of Bahia (UFBA) & Federal University of the South of Bahia (UFSB), Brazil
David Greenhalgh, University of Strathclyde, Glasgow, UK
David Griol, Carlos III University of Madrid, Spain
Emmanuelle Grislin, LAMIH-CNRS UMR 8201 | Université de Valenciennes, France
Jerzy Grzymala-Busse, University of Kansas, USA
Sung Ho Ha, Kyungpook National University, South Korea
Maki K. Habib, The American University in Cairo, Egypt
Ibrahim A. Hameed, NTNU, Norway
Jessica Heesen, University of Tübingen, Germany
Wladyslaw Homenda, Warsaw University of Technology, Poland
Katsuhiro Honda, Osaka Prefecture University, Japan
Tzung-Pei Hong, National University of Kaohsiung, Taiwan
Wei-Chiang Hong, School of Education Intelligent Technology - Jiangsu Normal University, China

Christopher-Eyk Hrabia, Technische Universität Berlin | DAI-Labor, Germany
Michael Hübner, Ruhr-Universität Bochum, Germany
Chih-Cheng Hung, Kennesaw State University - Marietta Campus, USA
Sardar Jaf, University of Durham, UK
Richard Jiang, Northumbria University, UK
Maria João Ferreira, Universidade Portucalense, Portugal
Janusz Kacprzyk, Systems Research Institute - Polish Academy of Sciences, Poland
Epaminondas Kapetanios, University of Westminster, London, UK
Nikos Karacapilidis, University of Patras, Greece
Fakhri Karray, University of Waterloo, Canada
Alexey M. Kashevnik, SPIIRAS, Russia
Fouad Khelifi, Northumbria University at Newcastle, UK
Shubhalaxmi Kher, Arkansas State University, USA
Hyunju Kim, Wheaton College, USA
Sungshin Kim, Pusan National University, Korea
Ah-Lian Kor, Leeds Beckett University, UK
Sotiris Kotsiantis, University of Patras, Greece
Tobias Küster, DAI-Labor/Technische Universität Berlin, Germany
Ruggero Donida Labati, Universita' degli Studi di Milano, Italy
Ramoni Lasisi, Virginia Military Institute, USA
María Elena Lárraga Ramírez, Instituto de Ingeniería | Universidad Nacional Autónoma de México, Mexico
Antonio LaTorre, Universidad Politécnica de Madrid, Spain
Egons Lavendelis, Riga Technical University, Latvia
Frédéric Le Mouël, Univ. Lyon / INSA Lyon, France
George Lekeas, City Universty - London, UK
Carlos Leon de Mora, University of Seville, Spain
Chanjuan Liu, Dalian University of Technology, China
Daniela López De Luise, CI2S Labs, Argentina
Isabel Machado Alexandre, Instituto Universitário de Lisboa (ISCTE - IUL) & Instituto de Telecomunicações, Portugal
Prabhat Mahanti, University of New Brunswick, Canada
Mohammad Saeid Mahdavinejad, University of Isfahan, Iran
Giuseppe Mangioni, University of Catania, Italy
Francesco Marcelloni, University of Pisa, Italy
Antonio Martín-Montes, University of Sevilla, Spain
René Meier, Hochschule Luzern, Germany
António Meireles, GECAD - Research Group on Intelligent Engineering and Computing for Advanced Innovation and Development, Portugal
Michele Melchiori, University of Brescia, Italy
John-Jules Charles Meyer, Utrecht University, The Netherlands
Angelos Michalas, TEI of Western Macedonia, Kastoria, Greece
Dusmanta Kumar Mohanta, Birla Institute of Technology, India
Jose Manuel Molina Lopez, Universidad Carlos III de Madrid, Spain
Fernando Moreira, Universidade Portucalense - Porto, Portugal
Debajyoti Mukhopadhyay, Maharashtra Institute of Technology, India
Kenric Nelson, Boston University, USA
Filippo Neri, University of Napoli "Federico II", Italy

Sergi Valverde, Universitat Pompeu Fabra (UPF), Spain
Leo van Moergestel, HU University of Applied Sciences Utrecht, Netherlands
Jan Vascak, Technical University of Kosice, Slovakia
Jose Luis Vazquez-Poletti, Universidad Complutense de Madrid, Spain
Laura Verde, University of Naples “Parthenope”, Italy
Susana M. Vieira,  IDMEC - Instituto Superior Tecnico - Universidade de Lisboa, Portugal
Fangju Wang, University of Guelph, Canada
Longzhi Yang, Northumbria University, Newcastle upon Tyne, UK
Ali Yavari, Swinburne University of Technology, Australia
George Yee, Carleton University & Aptusinnova Inc., Ottawa, Canada
Katharina Anna Zweig, TU Kaiserslautern, Germany

## Copyright Information

## Table of Contents

# A 3D Convolutional Neural Network for Anomalous Propagation Identification

Hansoo Lee, Jonggeun Kim, and Sungshin Kim

Department of Electrical and
Computer Engineering
Pusan National University
Busan, South Korea
Email: {hansoo, wisekim, sskim}@pusan.ac.kr

*Abstract*—**Radar is one of the most popular and widely used weather observation devices because of its better performance compared to other remote sensing devices. However, the observation results of the radar unavoidably contain unwanted signals, called non-precipitation echoes, which include anomalous propagation. These represent a negative influence, especially in the quantitative precipitation estimation. Therefore, it is essential to remove the anomalous propagation in the radar data for accurate weather forecasting. In this paper, we implemented a three-dimensional convolutional neural network for classifying the anomalous propagation in the radar data. Without considering feature engineering, which is difficult and mostly handcrafted, we were able to obtain improved performance in the classification with actual occurrence cases of the echo.**

***Keywords*–Pattern recognition; Deep learning; 3D convolutional neural network; Anomalous propagation; Radar data analysis.**

## I. Introduction

Machine learning, which allows solving real-world problems by utilizing given data, applies to lots of practical fields including medicine [1], finance [2], genetics and genomics [3], etc. Additionally, deep learning [4], a sub-class of the machine learning, significantly influences many aspects of modern society by achieving outstanding improvements especially in large-scale image processing and speech recognition. One of the compelling advantages of deep learning is that it can derive remarkably successful results without considering feature selection [5] and extraction [6]. Therefore, there are a lot of ongoing active studies that aim to lower the expensive computational costs.

These research works influence many academic and practical fields including weather prediction because the weather prediction is intimately connected with modern society [7]. For example, it is possible to protect lives and properties by forecasting storms and local torrential rainfalls. Also, these works help minimize economic damages from agflation caused by abnormal climate changes. Deep learning related studies have been gradually growing to respond to an increased demand for accurately analyzed results from observation devices such as radar and satellite.

Currently published researches related to weather prediction are mainly focused on precipitation nowcasting [8][9] and storm identification [10][11] based on accurately analyzed results of radar observations. The radar is the most popular weather observation device because it can generate spatiotemporal observation results with high resolution, and is able to provide three-dimensional precipitation information in a more direct way than other sensing devices. However, the radar observes all objects in the atmosphere without exception. In other words, the observation results inevitably contain unwanted signals, called non-precipitation echoes.

Non-precipitation echoes have many different causes. The typical non-precipitation echoes are as follows. First, interference [12] occurs by strong wireless impulse signals which have similar bandwidth to radars. Second, biological echo can appear [13] by a flock of birds or insects. Third, ground echo [14] and sea clutter [15] can be present in the radar image by artificial or natural objects on the surface of the earth and the sea. Fourth, chaff echo [16] occurs by scattered lightweight materials from an aircraft or battleship to avoid radar detection. Fifth, anomalous propagation [17] appears by refracted radar beam due to rapid changes in temperature or humidity. Among them, the anomalous propagation causes significant errors in radar rainfall estimation because it is less predictable and has changeable intensity of reflectivity or extension of areas.

In early days, a manual quality control process based on experts knowledge was used to eliminate anomalous propagation. After that, statistical-based [18] and machine learning-based [19] methods were complementary used. However, earlier studies applied conventional machine learning methods which included feature engineering. Considering that feature engineering negatively influences performance, many difficulties followed, unavoidably. In this paper, we implement a non-precipitation echo detection method based on a three-dimensional convolutional neural network. By using our deep learning architecture, it is unnecessary to go through additional feature selection and extraction.

This paper is organized as follows. In Section 2, we briefly present a background knowledge of radar operating principles and anomalous propagation. Section 3 explains convolutional neural network and introduces our 3D architecture. In Section 4, our experimental framework and results are described. Section 5 provides the conclusion and future works.

## II. Background

This section explains the operating principle of radar and occurrence properties of the anomalous propagation echo for providing background knowledge.

### A. Weather Radar

The primary operating principles of radar are radiating intense electromagnetic energy and gathering backscattered signals from floating objects in the observation hemisphere. In other words, by using electromagnetic energy as its measuring tools, radar computes valuable information for analyzing properties of the reflected signals including distance, power density, radial velocity, etc. [20].

The operating principle makes the radar one of the most popular measurement devices for weather forecasting because the electromagnetic waves travel their pre-set route from the transmitter of radar regardless of weather condition. In other words, radar can operate 24 hours a day, seven days a week in all weather conditions including severely low visibility conditions including fog, rain, snow, and hail.

There are two main types of scans: Range Height Indicator (RHI) and Plan Position Indicator (PPI). The RHI scan provides the image from the side. Lots of studies utilize the RHI scan when an improved vertical resolution is required. On the other hands, the PPI scan produces the image as seen from above [21]. The PPI scan is generally utilized in weather forecasting process because it facilitates to understand time series changes of radar echoes.

### B. Anomalous Propagation

The electromagnetic waves follow the quasi-optical laws because they behave similarly to light beams in a uniform and constant medium. But the precondition is rarely satisfied in the earth's atmosphere in practice. In other words, the refraction of the emitted electromagnetic waves is a common phenomenon because of several factors including pressure, temperature, and vapor pressure. Considering that the primary operating principle of the radar is based on the condition that the emitted electromagnetic waves travel in an ideal atmospheric environment, measurement results are inevitably wrong. Therefore, standard refraction based on these factors is commonly used in actual observation instead of no refraction condition.

From a different point of view, the radiated electromagnetic waves from the radar can travel in various directions due to refraction when the specific conditions are satisfied. For instance, the rapid changes of a temperature gradient, pressure or water vapor content can bend the waves or even trap them in a specific layer in the air. As a result, when the rapid changes of the atmospheric condition refract the radar beam, there is a chance that the radar cannot perceive the difference which can derive significant wrong results in weather forecasting.

There are two typical different cases of the refracted pathways: the radar shows nothing when raining, and the radar shows precipitation echoes without raining. The former situation occurs when the radar beams are refracted toward the opposite direction of the surface, which is called sub-refraction. And the latter situation occurs when the radar beams are refracted toward the surface, which is called super-refraction. In this case, the radar misrecognizes the objects on the earth or sea surface as precipitation echoes. The misrecognized echo is called anomalous propagation.

Notably, the anomalous propagation causes significant errors in radar rainfall estimation because it is less predictable and it has the changeable intensity of reflectivity or extension of areas. Therefore, the anomalous propagation should be removed from the radar observation result for accurate weather forecasting. Figure 1 and Figure 2 show individual cases of precipitation and anomalous propagation, respectively. As shown in the figures, it is difficult to distinguish which one is precipitation and which is anomalous propagation without a quality control process.

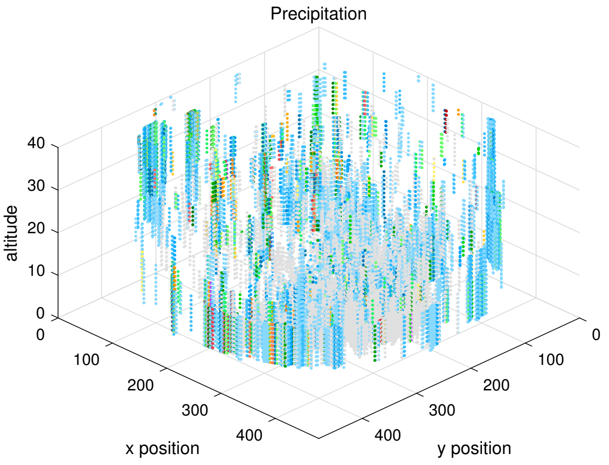


Figure 1. Precipitation case

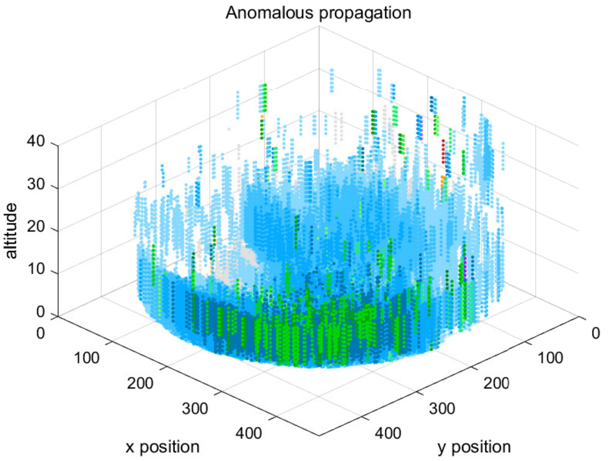


Figure 2. Anomalous propagation case

## III. Methods

This section provides brief elucidations about a conventional artificial neural network, a convolutional neural network that is one of the outstanding deep learning models, and detailed explanations about our implemented three-dimensional convolutional neural network.

### A. Artificial Neural Network

The artificial neural network is a mathematical algorithm for high-level data processing which is inspired by biological nerve systems. It is confirmed that the biological nerve system is a source of the artificial neural network because the operating principles of their basic components are considerably similar. Many practical applications frequently use the artificial neural network for solving their problems because the

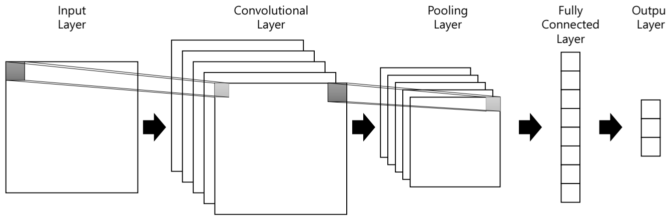


Figure 3. Simplified convolutional neural network

model has good performance in classification, regression, and clustering [22].

Layers are typical organizations of the artificial neural network. Nodes, which contain an activation function, are components of the layers. The artificial neural network can solve difficult problems by using highly interconnected and weighted nodes. There are three types of layers: input, hidden, and output. When the artificial neural network gets a multidimensional vector as an input, the input layer distributes the input to the hidden layer. And the hidden layer determines whether the outputs of the previous layer are helpful or harmful to the final result and distributes its output by using weighted sum and activation function. The output layer finalizes outputs of the previous layer. In summary, it is possible to describe the operating principle of the artificial neural network in (1).

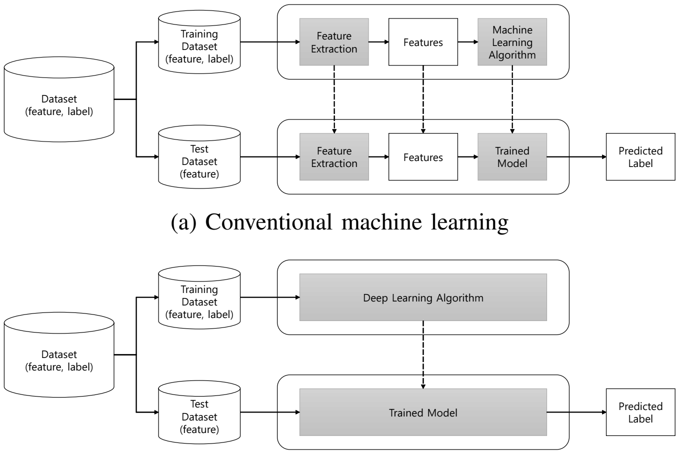
$$y = f_h \left( \sum_{i=1}^{n} \omega_i x_i - b \right) \tag{1}$$

where $f(\cdot)$ is an activation function, $n$ is the number of variables $x_i$ from the previous layer, $\omega$ is weights, and $b$ is biases.

However, despite the outstanding performances of the conventional artificial neural network in various practical fields, the requirement of significant computational complexity is the most substantial limitation of the model when it needs to deal with image processing. For example, the conventional model requires $12,228$ weights in the first hidden layer for analyzing a $64 \times 64$ color image. Additionally, considering that the network structure should be a lot larger than the input image, the conventional neural network seems not manageable for the given problem. In other words, there are two main reasons why the conventional neural network is not suitable for image processing. First, it is necessary to provide unlimited computational power and time for training the huge model. Second, it might cause over-fitting problem.

### *B. Convolutional Neural Network*

For solving the two problems of the conventional model, a convolutional neural network is suggested [23]. The convolutional neural network has similar components to the conventional artificial neural network. Namely, they have identical structures, components, and backpropagation based on the self-optimisation process. But a noticeable difference exists between the conventional neural network and the convolutional neural network in that the latter has three salient types of layers: convolutional, pooling and fully-connected layers. Figure 3 illustrates the simplified example of the convolutional neural network.



(a) Conventional machine learning

(b) Deep learning

Figure 4. Comparison of model learning process

When the input layer distributes the pixels of the image as inputs, the convolution layer convolves each filter across the data to produce a two-dimensional activation map. By using a zero-padding process, it is possible to keep the size of each convolved data as given inputs. The pooling layer reduces the data from the convolution layer with activation function for curtailing the number of parameters and the computational complexity of the model. The fully-connected layer performs the same roles as the conventional neural network and attempts to derive scores from the activation functions. Finally, the convolutional neural network uses the derived scores for classification.

Furthermore, there is another advantage to notice in the convolutional neural network that the convolution layers in the model can extract features from given input data. In the majority of conventional machine learning algorithms, they should include feature engineering in a training phase. The principal point is that most of the features are hand-crafted, which is difficult, time-consuming and requiring domain expertise. Also, if the extracted features could not describe the given data well, it is possible to degenerate performances of the model. Figure 4a shows the learning and prediction phases of the conventional machine learning methods, which include the feature extraction in the process. On the other hand, it is unnecessary to put the time and effort into feature engineering when the convolutional neural network is applied. Figure 4b illustrates the learning and prediction phases of the deep learning including the convolutional neural network. It is easily noticeable in Figure 4 that the feature extraction process is not necessary for the deep learning implementation.

### *C. 3D Convolutional Neural Network*

In this paper, we implemented a three-dimensional convolutional neural network for practical utilization in radar data analysis. The architecture of the model is shown in Figure 5, which contains four hidden convolution layers and a fully-connected layer. The convolution layers contain convolution filter, ReLu activation function $(f(x) = max(x, 0))$, and max-pooling. Additionally, dropout is applied for preventing overfitting.

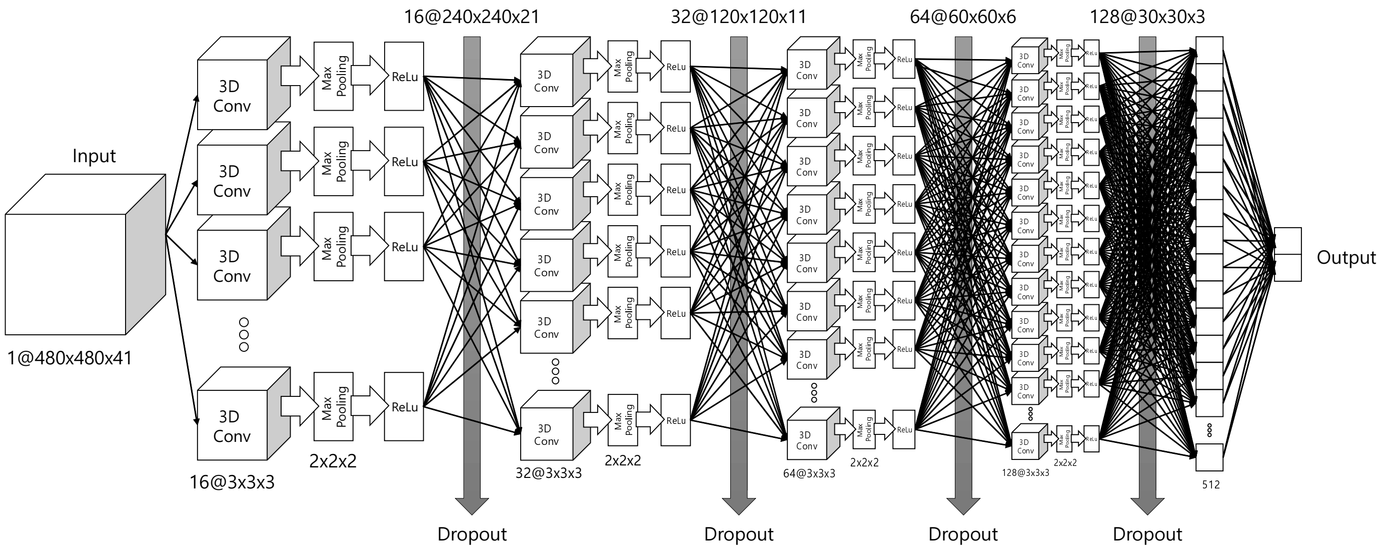


Figure 5. 3D Convolutional neural network structure

The implemented three-dimensional convolutional neural network is similar to the architecture of two-dimensional convolutional neural networks. But, unlike the bidimensional convolutional neural network structure, the implemented model utilizes tridimensional convolutional filters, activation functions, and max-poolings. We chose a $3 \times 3 \times 3$ size structure for convolutional filters by conducting empirical experimentations. Also, we designed the convolution layers so that the shape of input and output is identical by using a zero-padding process. In case of max-pooling, we chose a $2 \times 2 \times 2$ shape. This kind of max-pooling structure allows reducing the computational complexity of the convolutional network for both two- and three-dimensional structure. Similarity and dissimilarity of the convolutional neural network structures are easily found in (2) and (3).

$$v_{ij}^{xyz} = f\left(\sum_{m}\sum_{p=0}^{P_i-1}\sum_{q=0}^{Q_i-1}\omega_{ijm}^{pq} v_{(i-1)m}^{(x+p)(y+q)} + b_{ij}\right) \quad (2)$$

$$v_{ij}^{xyz} = f\left(\sum_{m}\sum_{p=0}^{P_i-1}\sum_{q=0}^{Q_i-1}\sum_{r=0}^{R_i-1}\omega_{ijm}^{pqr} v_{(i-1)m}^{(x+p)(y+q)(z+r)} + b_{ij}\right) \quad (3)$$

where $(x, y, z)$ is a coordinate of feature map and volume, $(p, q, r)$ is a spatial dimension index of kernel, $i$ indicates convolution layer, $b_{ij}$ means bias, and $f(\cdot)$ is an activation function.

Also, for applying the rule [24], we tried to add more layers in the architecture. As a result, we implemented another convolutional neural network, which additionally contains a fully-connected layers, as shown in Figure 6. As for the same structure described in Figure 5, the convolution layers contain a convolution filter, ReLu activation function, max-pooling and dropout. The difference between the two models is illustrated in Table I for readibility.

## IV. Experiments

Currently, the implemented network is designed as a binary classification as shown in Figure 5 because it is hard to obtain the sufficient number of individual recurrence case of each non-precipitation echo. Also, the simultaneous occurrence cases of the non-precipitation echoes are more frequent than the standalone occurrence cases. Therefore, we utilised learning of the implemented model by using two days of anomalous propagation and two days of precipitation. And we applied the other data for testing which contains both precipitation and anomalous propagation separately. In summary, we used 508 numbers of tridimensional radar images as training data and 144 number of radar images as test data. Also, we trained the implemented models with the Adam optimizer at a learning rate of 0.001.

The testbed environment configuration was as follows:

- CPU: Intel i7-7700K @ 4.20GHz × 8
- RAM: 16GB DDR4
- GPU: NVIDIA GeForce GTX1080/PCIe/SSE2
- Framework: TensorFlow 1.4.1, Python 3.5.2
- OS: Ubuntu 16.04 LTS

For evaluating the implemented three-dimensional convolutional neural network, we conducted evaluations using accuracy as shown in (4).

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \quad (4)$$

where $TP$ indicates true positive, $TN$ indicates true negative, $FP$ indicates false positive, and $FN$ indicates false negative. Also, we utilised the terms that true indicates the anomalous propagation echo, and that false indicates the non-anomalous propagation echo, respectively.

We derived the results from the implemented models in Figure 5 and Figure 6. By using the model in Figure 5, it showed classification accuracy as $68.75\%$ on average. On the

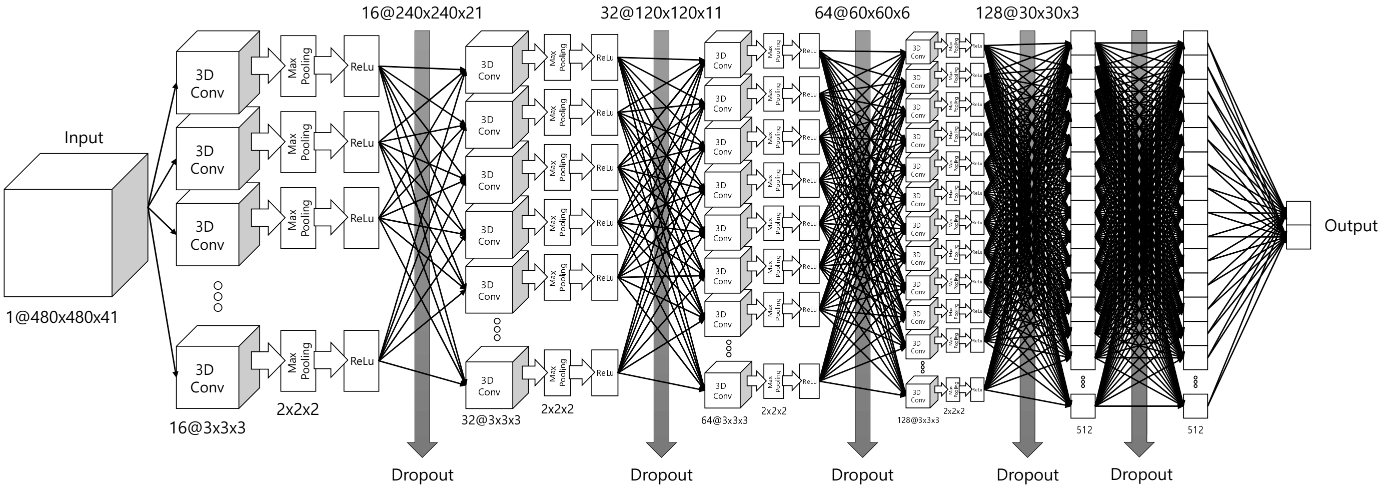


Figure 6. Extended 3D Convolutional neural network structure

TABLE I. THREE-DIMENSIONAL CONVOLUTIONAL NEURAL NETWORKS FOR EXPERIMENTATION

| | Configuration | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 3DCNN | input | 16@conv3d | maxpool | 32@conv3d | maxpool | 64@conv3d | maxpool | 128@conv3d | maxpool | FC-512 | softmax | |
| 3DCNN_Extended | input | 16@conv3d | maxpool | 32@conv3d | maxpool | 64@conv3d | maxpool | 128@conv3d | maxpool | FC-512 | FC-512 | softmax |

other hands, by using the model in Figure 6, it showed better average accuracy as $72.22\%$. From the experimental results, we can conclude that the three-dimensional convolutional neural network as shown in Figure 6 shows better results because the two sequentially connected fully-connected layers operate as the conventional neural network.

## V. Conclusion and Future Works

In this paper, we implemented a three-dimensional convolutional neural network for classifying the anomalous propagation in the radar data as a feasibility study. The implemented model was able to learn volumetric features in tridimensional radar data without information loss. As a result, the three-dimensional convolutional neural network was able to identify the anomalous propagation by using actual occurrence cases of the anomalous propagation.

In future works, we will try to implement multi-class classification method by using the proposed method as a prototype. Currently, the implemented network is designed as a binary classification to classify the whether the given tridimensional is an anomalous propagation or not. The convolutional neural network is easy to expand from binary to multi-class classification by expanding the number of layer elements of the output layer. Additionally, the multi-class classification based approach is a more beneficial way to utilize in practical fields because it is more prone to occur different types of non-precipitation echoes simultaneously.

## Acknowledgment

This work was supported by BK21PLUS, Creative Human Resource Development Program for IT Convergence.

## References

[1] R. C. Deo, “Machine learning in medicine,” Circulation, vol. 132, no. 20, 2015, pp. 1920–1930.

[2] W.-Y. Lin, Y.-H. Hu, and C.-F. Tsai, “Machine learning in financial crisis prediction: a survey,” IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews), vol. 42, no. 4, 2012, pp. 421–436.

[3] M. W. Libbrecht and W. S. Noble, “Machine learning applications in genetics and genomics,” Nature Reviews Genetics, vol. 16, no. 6, 2015, p. 321.

[4] Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning,” Nature, vol. 521, no. 7553, 2015, p. 436.

[5] G. Chandrashekar and F. Sahin, “A survey on feature selection methods,” Computers & Electrical Engineering, vol. 40, no. 1, 2014, pp. 16–28.

[6] J. Tang, S. Alelyani, and H. Liu, “Feature selection for classification: A review,” Data Classification: Algorithms and Applications, 2014, p. 37.

[7] X. Shi et al., “Deep learning for precipitation nowcasting: A benchmark and a new model,” in Advances in Neural Information Processing Systems, 2017, pp. 5622–5632.

[8] S. Xingjian et al., “Convolutional lstm network: A machine learning approach for precipitation nowcasting,” in Advances in neural information processing systems, 2015, pp. 802–810.

[9] S. Kim, S. Hong, M. Joh, and S.-K. Song, “Deeprain: Convlstm network for precipitation prediction using multichannel radar data,” arXiv preprint arXiv:1711.02316, 2017.

[10] W. Zhang, L. Han, J. Sun, H. Guo, and J. Dai, “Application of multi-channel 3d-cube successive convolution network for convective storm nowcasting,” arXiv preprint arXiv:1702.04517, 2017.

[11] Y. Liu et al., “Application of deep convolutional neural networks for detecting extreme weather in climate datasets,” arXiv preprint arXiv:1605.01156, 2016.

[12] E. Saltikoff et al., “The threat to weather radars by wireless technology,” Bulletin of the American Meteorological Society, vol. 97, no. 7, 2016, pp. 1159–1167.

[13] V. Lakshmanan, J. Zhang, and K. Howard, “A technique to censor biological echoes in radar reflectivity data,” Journal of Applied Meteorology and Climatology, vol. 49, no. 3, 2010, pp. 453–462.

[14] S. M. Bachmann and M. Tracy, “Data driven adaptive identification and suppression of ground clutter for weather radar,” in 25th Conference on IIPS for Meteorology, Oceanography, and Hydrology, Nashville, TN, USA, vol. 11, 2009, p. B3.

[15] P. Gerstoft, W. S. Hodgkiss, L. T. Rogers, and M. Jablecki, “Probability distribution of low-altitude propagation loss from radar sea clutter data,” Radio Science, vol. 39, no. 6, 2004, pp. 1–9.

[16] Y. H. Kim, S. Kim, H.-Y. Han, B.-H. Heo, and C.-H. You, “Real-time detection and filtering of chaff clutter from single-polarization doppler radar data,” Journal of Atmospheric and Oceanic Technology, vol. 30, no. 5, 2013, pp. 873–895.

[17] M. Grecu and W. F. Krajewski, “An efficient methodology for detection of anomalous propagation echoes in radar reflectivity data using neural networks,” Journal of Atmospheric and Oceanic Technology, vol. 17, no. 2, 2000, pp. 121–129.

[18] S. Moszkowicz, G. J. Ciach, and W. F. Krajewski, “Statistical detection of anomalous propagation in radar reflectivity patterns,” Journal of Atmospheric and Oceanic Technology, vol. 11, no. 4, 1994, pp. 1026–1034.

[19] M. A. Rico-Ramirez and I. D. Cluckie, “Classification of ground clutter and anomalous propagation using dual-polarization weather radar,” IEEE Transactions on Geoscience and Remote Sensing, vol. 46, no. 7, 2008, pp. 1892–1904.

[20] M. I. Skolnik, “Introduction to radar,” Radar Handbook, vol. 2, 1962.

[21] F. Fabry, Radar meteorology: principles and practice. Cambridge University Press, 2015.

[22] S. Haykin and N. Network, “A comprehensive foundation,” Neural Networks, vol. 2, no. 2004, 2004, p. 41.

[23] Y. LeCun and Y. Bengio, “Convolutional networks for images, speech, and time series,” The handbook of brain theory and neural networks, vol. 3361, no. 10, 1995, p. 1995.

[24] Y. Bengio, “Practical recommendations for gradient-based training of deep architectures,” in Neural networks: Tricks of the trade. Springer, 2012, pp. 437–478.

# Towards a Robust Imputation Evaluation Framework

Anthony Chapman, Wei Pang, George Coghill
Department of Computing Science
University of Aberdeen, Aberdeen, UK
Email: {r01ac14, pang.wei, g.coghill}@abdn.ac.uk

***Abstract*—Missing data research is hindered by a lack in imputation evaluation techniques. Imputation has the potential to increase the impact and validity of studies from different sectors (research, public and private). By creating robust evaluation software, more researchers may be willing to use and justify using imputation methods. This paper aims to encourage further research for robust imputation evaluation by defining a framework which could be used to optimise the way we impute datasets prior to data analysis. We propose a framework which uses a prototypical approach to create testing data and machine learning methods to create a new metric for evaluation. We introduce our implementation of such a framework and present some preliminary results. The results show how, for our dataset, records with less than 40% missingness could be used for analysis, which increases the amount of available data for future studies using that dataset.**

***Keywords*—*Missing Data; Evaluating Imputation; Imputation; Clustering; Prototypical Testing.***

## I. INTRODUCTION

The number of papers evaluating imputation methods (methods that predict missing values) is so large we cannot fit all of them in this paper, yet there is no evaluation software. Although individually evaluating an imputation method on a dataset has it's place, there are many problems (discussed in the sections to follow) currently associated with it. Even though imputation research has experienced a surge in recent years, evaluating imputation has not advanced at the same pace. This is problematic given recent findings, namely the potentially negative effects imputation can have on the validity and reliability of data analysis [2], thus, more of our attention must be directed to the evaluation of such methods.

Recent reports have noted an increase in the number of studies using imputation methods [3], [4]. Although imputation is being used more, the preferred method is still complete case analysis (aka likewise deletion or masking), where records with missing values are omitted from analysis [5], [6]. Consequently, newer statistical techniques which have eclipsed complete case analysis, in terms of appropriateness, for most circumstances [7], [8], are not being widely used. A robust imputation evaluation method could lead to a rise of popularity in imputation by allowing users to (relatively) easily see the effects an imputation method has on their datasets.

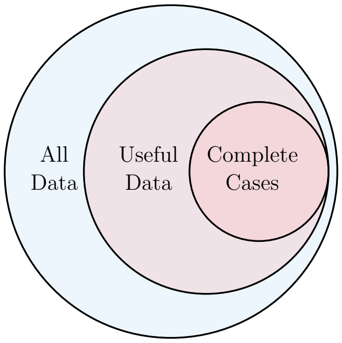


Figure. 1. Useful Data. There may be a larger amount of useful data than the subset of complete cases.

Evaluating imputation must be at the forefront of missing data research in order for imputation to be more widely accepted and, ultimately, used. By enabling others to evaluate imputation, they may be more inclined to consider imputing a dataset and, when appropriate, to use the imputed dataset for an appropriate study. This, in turn, could enable more of the available/meaningful data to be used [9] and this could help decrease uncertainty in studies, as illustrated in Figure I.

There are many challenges which need to be considered when evaluating imputation. One of which is that new imputation methods are constantly being developed and existing ones keep evolving [38]. Because of the evolving nature of imputation methods, any previous evaluations may become redundant/irrelevant and more up-to-date evaluations are constantly required. As there is currently no straightforward approach to evaluate imputation, we are constantly lagging behind new and improving imputation methods.

Another problem to consider is the complex structure of datasets: how will an imputation method behave with different datasets? Datasets could be very different in structure from one to another. They might consist of solely numerical values, solely non-numerical or could contain some mixture of the two types [10]. An evaluation method which could cope with such diversity could help overcome these issues.

Given the problems already stated, we will propose an imputation evaluation framework and the paper is structured as follows: Section II describes the motivations, implications and background related to this research. Section III describes the proposed framework and Section IV gives a breakdown of the benefits the framework could have on a system. Section V introduces, CLustering to Evaluate Multiple Imputation (CLEMI), our implementation of this framework and shows some preliminary results. The remaining sections consist of a discussion, which includes limitations, in Section VI, and finally, a conclusion and future work in Section VII.

## II. Motivation & Background

Missing data is a common occurrence [11] and can negatively affect inferences on the conclusions that may be drawn from data analysis [12], [13].

### A. Implications

Missing data prevention mechanisms may not ensure all data is recored or stored [14]. This may be due to a number of reasons ranging from study design to computer error. Ensuring all data is recorded is usually unfeasible in real life, and comping mechanisms for missing data, such as imputation, are paramount in maximising the use of available information [15].

Complete case analysis is still the most common mechanism used to cope with missing data, but records with missing values could also yield valuable information [16]. Although complete case analysis may be appropriate in some cases, ignoring records with missing values could lead to overestimation of results [8], depending on how the standard error is affected [17]. Furthermore, the analysis of information from a subset rather than the entire period of interest is also likely to alter the results of a study [18].

Incorrect imputation has the potential to produce drastically incorrect results [12] and some of the limitations of imputation (such as assuming regressions capture all necessary data for imputation) are discussed in Shih's paper [8]. Imputation could also lead to underestimation or overestimation of test statistics, depending on how standard errors are affected [19], so further analysis will be required by the users. These issues will require analysts to have in-depth knowledge of the data.

Evaluation methods could help with these problems by allowing the analysts to visualise the effects different imputation methods have on datasets of interest. Optimisation is another challenge met by those using imputation methods. Without evaluating imputation, how can we be sure that we have, not only the more appropriate imputation method, but also, optimised the chosen method to perform at its best capability.

### B. Missing Data

Although the effects different types of missing data have on imputation have been studied, we are still a long way from truly understanding the effects they have on imputation. An evaluation system could greatly advance current understanding regarding how imputation is affected by different types of missing data.

Missing data occurs for to a wide variety of reasons. The most common reason for missing data is participants dropping out of studies [20]. Other reasons include having too few participants, not reporting data, or the data not being applicable to the study [21]. Computer based reasons include computer error, from the mismatch of variables between datasets to improper merging [22]. These reasons could be minimised by improving study design, though it is unlikely that missing data can be prevented altogether.

Missing Completely at Random (MCAR), Missing at Random (MAR) and Missing not at Random (MNAR) are missing data mechanisms introduced by Rubin [23]. MCAR is when data are missing randomly throughout the dataset without any dependence between the variables. MAR is when missing data may be dependent on other observed data but the not dependent on other missing values. Finally, MNAR is when data are not MAR and the missing values are related to other missing values, with MNAR missing values could be linked to other missing values as well as the observed ones

These different mechanisms will make imputation behave differently. For example, datasets which have MCAR data are the least likely to produce bias [24] when complete case analysis is used. MNAR is seen as the most susceptible to bias but there are ways to minimise this. There are many assumptions and studies which look at the relationship between the different mechanisms and imputation, more research is required for us to understand the effects such mechanisms might have. A robust imputation evaluation methods will make testing the effects more accessible for anyone wishing to do so. Such tests could be done by controlling the type of mechanism used and analysing the effects imputation has, by doing this, we may be able to prove or disprove current notions of the behaviour of missing data mechanisms.

### C. Imputation

Imputation research has received a lot of interest in recent years as researchers and industry alike are trying to use as much of the available data as possible [25]. Although there are many methods at our disposal, which vary in appropriateness and complexity, imputation is still not being considered (as a method to deal with missing data) as much as it should be.

Imputation methods widely range in complexity and appropriateness. One of the simplest methods is default value imputation (where all missing values are replaced with a value, such as 0 or Female). Default value is generally regarded as taboo since they could potentially create bias in data. Mean imputation is slightly more intricate, it replaces any missing value with the mean of the corresponding column. Studies have found that mean imputation can overestimate results when there is more than 5%-10% missing values [42]. More complex methods also exist, such as multiple imputation, where regression models are created from the recorded data and missing values are imputed according to these regression models. Multiple imputation generates a number of datasets to account for uncertainty [26].

Imputation is still not being used adequately. One reason why it is not widely used could be due to the potential to underestimate values by imputing incorrectly [27]. Another reason could be a general lack of understanding on how imputation methods perform with different quantities and types of missing data [25]. Finally, imputation methods have, until recently, not been readily accessible to researchers [28].

One of the biggest factors against imputation is the lack of understanding in the effects they have on dataset with missing values. This understanding deficiency leads to a lack of trust in imputation methods, which leads to people not using as much of the available data, or the misuse of imputation methods, both of which change the data's underlying information and leads to a negative impact on studies.

Many imputation methods have a certain amount of flexibility which, potentially, allows users to impute datasets in a more efficient way. Without evaluation methods, it is sometimes difficult to choose the best parameters when imputing; this leads to sub-optimal imputation. Many studies either fail to look for optimal results or simply do not report how they have optimised imputation [29].

Having a robust and versatile imputation evaluation method could lead to better understanding how datasets are affected by different imputation methods.This could then enable and improve optimisation of imputation methods.

### D. *Evaluating Imputation*

Current imputation evaluation methods are mostly based on statistical regressions [11], [26], [30]–[35], some machine learning approaches have also been proposed [36]–[38]. Regression based evaluation performs well for individual cases but the results from such evaluations are not generalisable, the outcomes may not be applicable to others and are manually intensive to obtain.

Standards have been proposed to streamline imputation which, in theory, could lead to widespread evaluation to be carried out when imputing datasets [39]–[42]. These papers provide guidelines to handle missing data and suggest "good practices" for imputing datasets. They also provide useful information such as how some imputation methods might behave when applied to different types and/or quantities of missing data. Although these standards provide useful information, they do not advance on the problems posed for evaluating imputation. Some of the problems include:

- Results from evaluations may be unsuitable for other datasets or imputation methods
- It is not straightforward to evaluate the prediction of something which is truly missing
- Does the evaluation show whether the imputation method predict "true" values
- There are no standards to evaluate imputation methods
- Regression based evaluation is manually intensive.

Recent studies evaluating various imputation methods applied to a selection of datasets may be advantageous to resolve individual problems [31], [33], [38], [43]. However, due to inherent complexities of datasets, their results cannot be generalised for others to use. For example, [32] reported that for their particular dataset, multilevel imputation gave the best results in their study. This result may not be the same given a different dataset.

Similarly, three imputation methods were evaluated in [33] and four imputation methods were evaluated in [38]. Due to the different approaches used to evaluate the methods, these results are not comparable to each other; as one study may yield more appropriate results for their specified problem whereas an alternative study may find contradicting results. From this, we suggest that an evaluation method should be able to be compared to other methods in order for any results to be used by others.

By being able to compare different evaluation outcomes, may enable us to not only find the most appropriate method to impute a specific dataset, but also help us optimise an imputation method. By evaluating the same imputation method multiple times and changing any parameters every time and comparing the results, we may be able to optimise the method for a given dataset.

## III. Proposed Framework

Now that we have identified a lack of research on evaluating imputation, we propose a framework which could help with most of the current problems we face when evaluating imputation. The framework can be split into several stages. The first stage involves creating a benchmark dataset to evaluate imputation. The benchmark must be similar to the original dataset in order to preserve the relationship between them. In the second stage prototype datasets are created which can represent the original dataset for testing purposes. We will use these datasets to find the effect imputation has by comparing the imputed datasets to the benchmark.

In the third stage, the imputation methods are applied to the prototype datasets. It will be applied on all datasets in the same manner, specified by the user, in order to reduce uncertainty in the results. The forth stage will evaluate the imputed datasets by comparing them to the benchmark. In theory, a suitable imputation method will create values which are similar to the benchmark, conversely, an unsuitable imputation method will creates values which differ from the benchmark.

### A. *Benchmark*

In order to evaluate an imputation, a benchmark could be used. As different datasets are not guaranteed to behave the same, individual benchmarks must be used for every dataset.

We propose using the subset of the dataset consisting of the complete cases as the benchmark. Doing so, we reduce the variance between the dataset in question and the benchmark. This will decrease the evaluation uncertainty by maintaining a close link between the benchmark and the original dataset. This process is shown in Algorithm 1, line 3.

### B. *Prototypes*

To evaluate an imputation method, we could apply imputation to testing datasets and quantify the results. We will create prototypes from the benchmark, to act as our testing datasets. Then, impute the prototypes and compare the results to determine if the imputation method created realistic results, namely, whether the results have a small dissimilarity to the benchmark.

The prototypes are created by copying the benchmark and then analysing the missing data structure of the original dataset and imposing the same levels of missingness onto the copy, as illustrated in Figure 2. This is randomised, by creating different (but similar) prototypes, to increase the variability of the datasets. We randomise to reduce uncertainly when

**Algorithm 1:** Pseudo code for imputation evaluation framework. ©2018 Chapman, Pang & Coghill

```
   input : A dataset with missing values and parameters (if
           any) for the imputation methods.
   output: Evaluation Score: Difference between the
           imputed prototypes and the benchmark.
1  data ← original dataset with missing values ;
2  param ← Imputation parameters ;
3  bench ← complete cases from data ;
4  n ← amount of prototypes;
5  missingDist ← missingness distribution from data ;
6  for $i \leftarrow 1$ to $n$ do
7      p(i) ← bench.delete(missingDist) ;
8  end
9  for $i \leftarrow 1$ to $n$ do
10     pImp(i) ← impute(p(i), method=param) ;
11     pMEAN(i) ← impute(p(i), method=mean) ;
12 end
13 for $i \leftarrow 1$ to $n$ do
14     pClustImp(i) ← cluster(pImp(i)) ;
15     pClustMean(i) ← cluster(pMean(i)) ;
16 end
17 benchClust ← cluster(bench) ;
18 for $i \leftarrow 1$ to $n$ do
19     disImp(i) ← dissimilarity(pClustImpE(i),benchClust) ;
20     disMean(i) ←
         dissimilarity(pClustMean(i),benchClust) ;
21 end
```

analysing multiple imputed datasets. This process is shown in Algorithm 1, lines 4-8.

One simple to impose missingness onto the prototypes in a way that mimics the original, is to calculate the proportion of missing values per column in the original and then delete the same proportion from the prototype. Although this is a simple method, it does rely on some assumptions. One is that it assumes no relationship between the variables. Another is that any missing data mechanisms are not analysed. To have a strong relationship between the original dataset and the benchmark, these assumptions must be analysed further and maybe extended so any underlying relationships are accounted for.

### C. Imputation

By this stage, we should have an original dataset, a benchmark and multiple prototypes. The framework will now impute the prototypes, with any parameters specified by the user, independently. It is important to apply the imputation, with the same parameters, to each prototype in order to obtain reliable results. This process is shown in Algorithm 1, lines 9-12.

### D. Evaluation

The final stage will involve comparing the imputed prototypes to the benchmark. This will allow us to evaluate how well imputation has performed based on how similar the imputed datasets are to the benchmark. The underlying theory is that the better the imputation, the smaller the difference between the imputed prototypes and the benchmark will be, conversely, a bigger difference implies a worse imputation

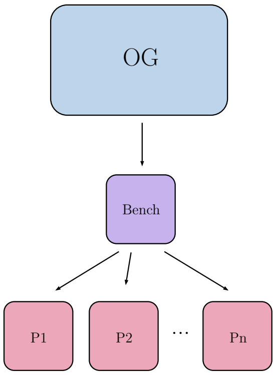


Figure. 2. Creation of Prototypes. The benchmark (Bench) is set as the subset of complete cases from the original (OG). Prototypes (P1 to Pn) are created by imposing missing values onto the bench.

One of the biggest challenges in comparing imputed datasets and the benchmark, is the problem posed by complex dataset dissimilarity measures, ie. how to define the distance between mixed data datasets. By clustering the datasets, we may be able to compare the clustering meta-data (such as cluster widths, density, size etc..) and compare the meta-data instead of the datasets. This may be possible due to the deterministic nature of clustering. If two datasets are similar, then their clustering meta-data will be similar. This process is shown in Algorithm 1, lines 13-21.

By comparing the meta-data, we mean that the (dis)similarity between clusterings can be represented by their structure. We would, for instance, create a similarity metric solely on the amount of data points per cluster (assuming the same amount of clusters) or on the density of their clusters. Thus, we could say two clusterings are similar, if their cluster sizes are similar. We could then move a step further and create a metric based on six clustering meta-data (cluster size, max dissimilarity, avg dissimilarity, cluster isolation score, individual silhouette widths and the avg silhouette width, as shown in Figure 3). We can then say two clustering are similar if their collective meta-data is similar.

## IV. Framework Benefits

Our framework expands the field in a number of ways. An underlying objective throughout our work has been to strive towards the provision of labour reducing imputation evaluation software. To do so, it is necessary to establish means to automatise processes, such as creating custom benchmarks, tailored ad-hoc prototypes and using a dissimilarity measure (created from clustering meta-data) which can be applicable to a variety of data types.

We have achieved this by using the subset of complete cases as a benchmark, creating a complete dataset with similar structure as the original. Using the missingness structure from the original to create prototypes, again, ensures these datasets follow a similar structure to the original. Finally, clustering techniques are used to define a dissimilarity measure between datasets with (possibly) mixed data.

The primary goal is to either use more of the available data (by showing it responds well to imputation) or justify not using imputation (by showing that it does not respond well). Whether the amount of extra available data justifies the means, ie whether it is worth it, is subjective. Even a dataset with relatively small amounts of missing data, may benefit from such methods, alternatively, some may think that this framework is not worth the effort required when there is only a small amount of missing data. Either way, we will not know whether the partially missing data is useful until it has been evaluated.

Creating an evaluation score enables results from different evaluations to be easily compared. By comparing scores, we may be able to reinforce post-imputation analysis and potentially discover more about the relationship between missing data and imputation.

Clustering techniques could be used to create a dissimilarity measure. This makes us able to not only quantify the difference between non-numerical data, we may also be able to create a metric which can be used to compare different evaluation scores to each other.

Using an evaluation score, we are able to run the evaluation system multiple times, and can change the imputation parameters every time. From this, we may be able to optimise the imputation parameters for a given dataset by comparing the scores produced by imputing with different parameters. Although this was possible before, through regression comparisons, our framework makes it more straightforward for someone who wants to optimise their imputation methods, since the tests will be carried out autonomously.

## V. Preliminary Results

We are currently implementing the framework proposed in this paper, called CLustering to Evaluate Multiple Imputation (CLEMI). CLEMI is being implemented in R, the statistical language, and we hope to make it a publicly available package/library once it is completed.

CLEMI is currently being validated using controlled tests by varing degrees of missing data and analysing the outcomes. We are also currently testing our metric, which uses clustering meta-data to find the dissimilarity between datasets. We hope to have enough results to publish shortly after the summer.

CLEMI uses MICE and Mean imputation (freely available on R), and compared the difference between 1. MICE imputed datasets and the benchmark and 2. Mean imputed datasets and the benchmark. We have used MICE as it is one of the more widely used imputation methods and we have used Mean imputation as a comparison as Mean imputation has shown to produced biased results in a lots of cases. The ultimate goal will be to find the smallest difference between MICE imputation and the benchmark (showing imputation yields similar results to the benchmark), and having results which are better than Mean imputation (if MICE produces similar results to Mean, then they are likely to produce biased results).

Figure 3 shows some preliminary user case results for CLEMI. Each of the six charts represents one part of a metric (one clustering meta-data value) used and all should be considered together for the final decision. When analysing the results, we look for the MICE box plot, blue box on the left, to be as low as possible whilst being lower than the Mean box plot, red box on the right.

This small user case uses a partially complete dataset with 10 variables which are mixed (both numerical and non-numerical data). We created 9 datasets which range from the amount of missing data we allow to remain, for example the first only have complete records and records with 9 recorded values, the second datasets contains all records with 8 recorded values or more, and so on until you have almost the original records (without records which are fully missing). From Figure 3, we can see that MICE is consistently lower than Mean and at its lowest point between 40–60% missing data within the records. So for this particular dataset, we should remove records with more than circa 60% missing values but we can impute the others; allowing us to decide how much of the available data should be used.

## VI. Discussion & Limitations

The proposed framework focuses on using prototypes and clustering meta-data to evaluate imputation. Research regarding imputation evaluation is crucial to informatics and having methods to cope with missing data, when missing data is present, will only strengthen data analysis. Evaluation methods could be used to optimise imputation and improve the credibility of studies using imputation. Such methods could also be used to improve our knowledge on the effect different types and/or quantities of missing data have on imputation.

A number of limitations with the framework were identified. One limitation is that the framework assumes that the prototypes represent the original dataset and the evaluation of the prototypes will reflect the imputation of the original. Some work will be needed to show whether this assumption is acceptable or not.

A more technical limitation lies at the heart of modeling theory. Although regressions have great modeling power, they also include a degree of uncertainty. Most multiple imputation methods rely on regressions to predict the missing values, this is done multiple times to reduce uncertainty but we cannot guarantee that the regressions created for the prototypes will be exactly those which were created for the original dataset. A good imputation method relies on a good regression model, but a good regression model is not guaranteed in every run.

Using clustering, we are able to create a dissimilarity measure between the prototypes and the benchmark. However, this is not easy in practice and there are many things to consider, for example, clustering creates meta-data, which we can use to create the metric for evaluation but it may not be easily interpreted, as shown in the preliminary user case.

## VII. Conclusions & Future Work

This paper has identified weaknesses in existing imputation evaluation research, which, if not addressed, could lead to

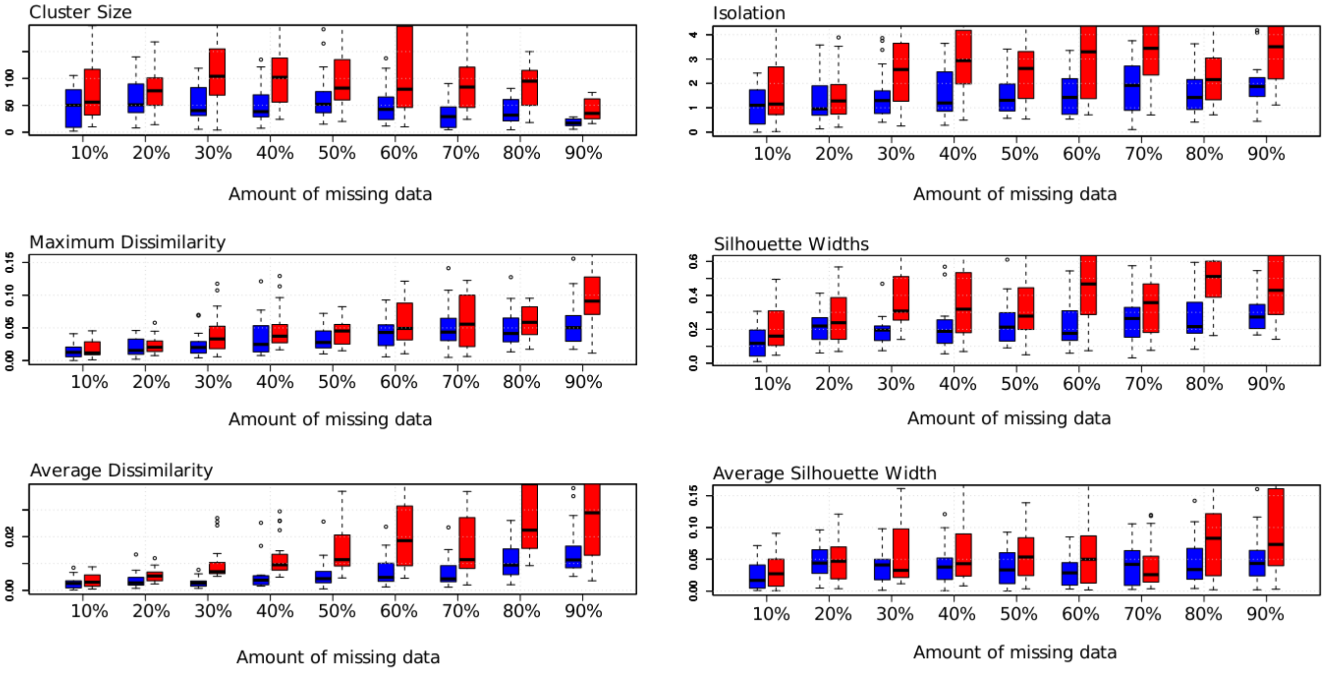


Figure. 3. Removing data (10% to 90%) to see how MICE and Mean are affected by the different amounts of missing data.

studies having a wost impact than they otherwise would have. The first problem identified is that complete case analysis seems to be the norm when faced with an incomplete dataset, this method does not use as much of the available data as possible, as illustrated in Figure 1.

Another problem identified, in current research, is the lack of imputation evaluation software, although we notice that there are many imputation methods. Finally, we identified a gap in literature for efficiently optimising imputation methods without having to create a new system for every dataset which needs to be imputed.

The proposed imputation evaluation framework may be used on a large variety of datasets, without having to manually create different methods for every evaluation. The framework includes a method for comparing the dissimilarity of datasets, efficiently, by using clustering to define a dissimilarity measure, this measure may work on both numerical and non-numerical data. Using such an evaluation method, we may be able to use more of the available data, and consequently, increase the impact from a given study.

We introduced CLEMI which will output dataset specific evaluation scores. Users will then have to decide whether the scores imply a satisfactory imputation method, for use in their studies, or not. Using an imputed dataset with a low evaluation score may lead to unreliable or biased results. The proposed framework will enable users to not only use more of the available data but even possibly strengthen the validity of their conclusions. This is especially important as we live in a world where the quantity of data being gathered may increase at a faster pace than data mining techniques and mechanisms.

Finally, future investigation could be carried out to address the already discussed limitations. To justify the prototypical nature of our framework, we might test the appropriateness of using the prototypes as a representative of the original dataset. We could do this by externally validating the similarities between the prototype datasets and the original dataset.

Using machine learning techniques, such as metric learning or feature ranking we may be able to create a standardised evaluation score, using the clustering meta-data, which is both reliable and user friendly (easy to interpret). Our initial idea is that some cluster information is more relevant than others for an evaluation score and, using machine learning techniques, we may be able to combine the information to create a score.

## References

[1] D. B. Rubin, “Inference and missing data,” *Biometrika*, vol. 63, no. 3, pp. 581–592, 1976.

[2] P. E. McKnight, K. M. McKnight, S. Sidani, and A. J. Figueredo, *Missing data: A gentle introduction*. Guilford Press, 2007.

[3] E. A. Akl, K. Shawwa, L. A. Kahale, T. Agoritsas, R. Brignardello-Petersen, J. W. Busse, A. Carrasco-Labra, S. Ebrahim, B. C. Johnston, I. Neumann *et al.*, “Reporting missing participant data in randomised trials: systematic survey of the methodological literature and a proposed guide,” *BMJ open*, vol. 5, no. 12, p. e008431, 2015.

[4] J. A. Hussain, M. Bland, D. Langan, M. J. Johnson, D. C. Currow, and I. R. White, “Quality of missing data reporting and handling in palliative care trials demonstrates that further development of the consort missing data reporting guidance is required: a systematic review,” *Journal of Clinical Epidemiology*, 2017.

[5] C. A. Manly and R. S. Wells, “Reporting the use of multiple imputation for missing data in higher education research,” *Research in Higher Education*, vol. 56, no. 4, pp. 397–409, 2015.

[6] I. Eekhout et al., “Missing data in a multi-item instrument were best handled by multiple imputation at the item score level,” *Journal of Clinical Epidemiology*, vol. 67, no. 3, pp. 335–342, 2014.

[7] J. L. Schafer and J. W. Graham, “Missing data: our view of the state of the art.” *Psychological methods*, vol. 7, no. 2, p. 147, 2002.

[8] W. J. Shih, “Current Controlled Trials in Problems in dealing with missing data and informative censoring in clinical trials,” vol. 7, pp. 1–7, 2002.

[9] R. Somasundaram and R. Nedunchezhian, “Evaluation of Three Simple Imputation Methods for Enhancing Preprocessing of Data with Missing Values,” *International Journal of Computer Applications*, vol. 21, no. 10, pp. 14–19, 2011.

[10] C. Kuchler and M. Spiess, “The data quality concept of accuracy in the context of publicly shared data sets,” *AStA Wirtschafts-und Sozialstatistisches Archiv*, vol. 3, no. 1, pp. 67–80, 2009.

[11] S. P. Mandel J, “A Comparison of Six Methods for Missing Data Imputation,” *Journal of Biometrics & Biostatistics*, vol. 06, no. 01, pp. 1–6, 2015.

[12] J. W. Graham, “Missing data analysis: Making it work in the real world,” *Annual review of psychology*, vol. 60, pp. 549–576, 2009.

[13] R. J. Little et al., *et al.*, “The prevention and treatment of missing data in clinical trials,” *New England Journal of Medicine*, vol. 367, pp. 1355–1360, 2012.

[14] R. O’neill and R. Temple, “The prevention and treatment of missing data in clinical trials: an fda perspective on the importance of dealing with it,” *Clinical Pharmacology & Therapeutics*, vol. 91, pp. 550–554, 2012.

[15] V. Tresp, R. Neuneier, and S. Ahmad, “Efficient methods for dealing with missing data in supervised learning,” in *Advances in neural information processing systems*, 1995, pp. 689–696.

[16] J. Osborne, *Best Practices in Data Cleaning*. Sage, 2013.

[17] M. Soley-Bori, “Dealing with missing data: Key assumptions and methods for applied analysis,” *Boston University*, 2013.

[18] S. F. Messner, “Exploring the consequences of erratic data reporting for cross-national research on homicide,” *Journal of Quantitative Criminology*, vol. 8, no. 2, pp. 155–173, 1992.

[19] I. R. White, P. Royston, and A. M. Wood, “Multiple imputation using chained equations: Issues and guidance for practice,” *Statistics in Medicine*, vol. 30, no. 4, pp. 377–399, 2011.

[20] W. J. Shih, “Problems in dealing with missing data and informative censoring in clinical trials,” *Current Controlled Trials in Cardiovascular Medicine*, vol. 3, no. 1, p. 4, 2002.

[21] A. I. for Research, “Three reasons for missing data: Engaging consumers in quality information,” *Robert Wood Johnson Foundation*, 2012.

[22] SPSS, “Missing Data : The Hidden Problem,” *Ibm Spss*, pp. 1–8, 2009.

[23] R. Little and D. Rubin, *Statistical Analysis with Missing Data*, ser. Wiley Series in Probability and Statistics. Wiley, 2014.

[24] B. K. Vaughn, “Data analysis using regression and multi-level/hierarchical models, by gelman, a., & hill, j.” *Journal of Educational Measurement*, vol. 45, no. 1, pp. 94–97, 2008.

[25] J. Scheffer, “Dealing with missing data,” 2002.

[26] R. W. Wiggins, M. Ely, and K. Lynch, “A comparative evaluation of currently available software remedies to handle missing data in the context of longitudinal design and analysis,” *NCDS User Support Group Working Paper 51*, pp. 1–25, 2000.

[27] N. K. Malhotra, “Analyzing marketing research data with incomplete information on the dependent variable,” *Journal of Marketing Research*, pp. 74–84, 1987.

[28] J. A. C. Sterne, I. R. White, and J. B. Carlin, “Multiple imputation for missing data in epidemiological and clinical research : potential and pitfalls,” pp. 1–11, 2017.

[29] K. Swarts et al., *et al.*, “Novel methods to optimize genotypic imputation for low-coverage, next-generation sequence data in crop plants,” *The Plant Genome*, vol. 7, no. 3, 2014.

[30] C. Westermeier and M. M. Grabka, “Longitudinal Wealth Data and Multiple Imputation: An Evaluation Study,” *Survey Research Methods*, vol. 10, no. 3, pp. 237–252, 2016.

[31] O. Akande, F. Li, and J. Reiter, “An Empirical Comparison of Multiple Imputation Methods for Categorical Data,” vol. 27708, pp. 1–30, 2015.

[32] J. R. van Ginkel and P. M. Kroonenberg, “Evaluation of multiple-imputation procedures for three-mode component models,” *Journal of Statistical Computation and Simulation*, vol. 87, no. 16, pp. 3059–3081, 2017.

[33] J. Baker, N. White, and K. Mengersen, “Missing in space: An evaluation of imputation methods for missing data in spatial analysis of risk factors for type II diabetes,” *International Journal of Health Geographics*, vol. 13, no. 1, pp. 1–13, 2014.

[34] A. K. Waljee et al., “Comparison of imputation methods for missing laboratory data in medicine,” *BMJ open*, vol. 3, no. 8, p. e002847, 2013.

[35] S. Seaman, J. Bartlett, and I. White, “Multiple imputation of missing covariates with non-linear effects and interactions: an evaluation of statistical methods.” *BMC medical research methodology*, vol. 12, no. Mi, p. 46, 2012.

[36] N. A. Samat, M. Najib, and M. Salleh, “A Study of Data Imputation Using Fuzzy C-Means with Particle Swarm Optimization,” vol. 549, no. 2, 2017.

[37] R. Veroneze, F. O. De França, and F. J. Von Zuben, “Assessing the performance of a swarm-based biclustering technique for data imputation,” *2011 IEEE Congress of Evolutionary Computation, CEC 2011*, pp. 386–393, 2011.

[38] Y. Liu and V. Gopalakrishnan, “An Overview and Evaluation of Recent Machine Learning Imputation Methods Using Cardiac Imaging Data,” *Data*, vol. 2, no. 1, p. 8, 2017.

[39] D. Salfr, P. Jordan, and M. Spiess, “Missing data : On criteria to evaluate imputation methods,” no. 4, 2016.

[40] G. Vink, “Towards a standardized evaluation of multiple imputation routines,” pp. 1–16.

[41] Y. He, A. M. Zaslavsky, M. Landrum, D. Harrington, and P. Catalano, “Multiple imputation in a large-scale complex survey: a practical guide,” *Statistical methods in medical research*, vol. 19, pp. 653–670, 2010.

[42] T. Li et al., “Standards should be applied in the prevention and handling of missing data for patient-centered outcomes research: A systematic review and expert consensus,” *Journal of Clinical Epidemiology*, vol. 67, no. 1, pp. 15–32, 2014.

[43] A. D. Shah et al., “Comparison of random forest and parametric imputation models for imputing missing data using mice: A caliber study,” *American Journal of Epidemiology*, vol. 179, no. 7, pp. 764–774, 2014.

[44] N. Mittag, “Imputations: Benefits, Risks and a Method for Missing Data,” 2013.

# Comparison of Artificial Intelligence Based Oscillometric Blood Pressure Estimation Techniques: A Review Paper

Ekambir Sidhu, Voicu Groza
School of Electrical Engineering and Computer Science
University of Ottawa
Ottawa, Canada
Emails: {esidh097, vgroza}@uottawa.ca

*Abstract* - **Accurate Blood Pressure (BP) measurement is an important physiological health parameter in the field of health monitoring, which is significant in determining the cardiovascular health of the patient under observation. Nowadays, automated blood pressure measurement systems are generally used by patients at home, and this requires less expertise to operate. The major requirement in the design of Automated Blood Pressure (ABP) measurement systems is the degree of accuracy and repeatability. There are various Artificial Intelligence (AI) based blood pressure estimation techniques and algorithms developed by various researchers in recent years and some of them are commonly employed by the BP monitoring market in the design of their automated blood pressure systems for accurate estimation of patient's systolic and diastolic blood pressures. In this review paper, various AI based Systolic Blood Pressure (SBP) and Diastolic Blood Pressure (DBP) estimation techniques and algorithms are analyzed and compared in terms of their ability for accurate estimation of real time patient blood pressure. The performance of various AI based blood estimation techniques are analyzed in terms of their complexity, Mean Absolute Error (MAE) and Standard Deviation Error (SDE).**

***Keywords - Artificial Neural Network (ANN); Adaptive NeuroFuzzy Inference System (ANFIS); Arterial blood pressure measurement; Principal Component Analysis (PCA).***

## I. INTRODUCTION

Accurate measurement of blood pressure plays an important role in the assessment and analysis of cardiovascular risk factors in clinical patient health monitoring because high blood pressure is a major risk for stroke or heart disease [1]. The accurate measurement of blood pressure is important for precise cardiovascular risk assessment, and for real time monitoring of the treatment effect by the doctors and health practitioners [1]. It has been repeatedly demonstrated in the studies carried out by various researchers that the blood pressure assessment in clinical practice is not precise, especially when the blood pressure is measured manually using the manual sphygmomanometer [2][3].

The deviations in blood pressure measurement techniques can lead to inaccuracy and misclassification of cardiovascular risk by the doctors and health practitioners [4]. For example, measuring the pressure manually with the arm positioned below the level of heart atria can lead to a blood pressure overestimation by 7-10/8-11 mm Hg [5]. In addition, leg crossing during manual blood pressure estimation also leads to the deviation in blood pressure by 8-10/4-5 mm Hg [6]. With the passage of time, the inaccurate manual blood pressure measurement systems have been replaced by Automated Blood Pressure (ABP) measurement systems which employs AI based BP measurement systems and digital display for the blood pressure readings. The digital blood pressure measurement systems suffer from the limitation of terminal digit preference i.e., rounding errors [7]. Thus, there is a need for accurate blood pressure measurement techniques and algorithms for the design of precise and accurate blood pressure measurement systems.

This review paper focuses on the study and comparison of various AI based BP estimation techniques and algorithms which have been developed by various researchers in the recent years for automated measurement of blood pressure precisely and accurately. Section II provides an introduction to blood pressure and various methods commonly employed for blood pressure measurement. Section III describes and compares the various non-invasive methods commonly used for the estimation and measurement of blood pressure. Section IV describes and classifies the various types of algorithms commonly used for the blood pressure measurement based on the stage at which the blood pressure estimation is carried out. Section V describes and goes into finer details of the various commonly employed AI based oscillometric blood pressure estimation algorithms for automated blood pressure measurement. Section VI compares the various AI based automated blood pressure estimation techniques and Section VII concludes this review paper.

## II. BLOOD PRESSURE

In clinical terms, the *arterial blood pressure* is defined as the measure of pressure exerted by the blood against the walls of the brachial artery (the main artery in the upper arm of humans). The blood pressure is necessary to pump and circulate the oxygenated blood across the body in order to supply oxygen to the living cells, which is vital for the human survival. The blood is oxygenated through the lungs and circulated across the body via human heart at each cardiac cycle. The blood circulation system of the human body is shown in Figure 1 below.

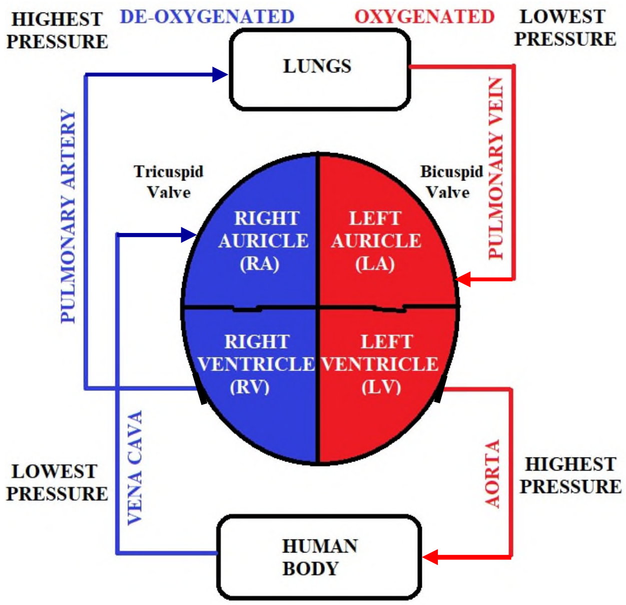


Figure 1. Human blood Circulation Model.

The heart operates like a pair of synchronized pumps and two pairs of valves (one between each auricle and ventricle and one between each ventricle and the blood vessel connected to it). The oxygenated blood is carried from lungs to the body by the left part of the heart, while the de-oxygenated blood is collected from the body and sent back to the lungs through the right part of the human heart. During the left ventricle contraction, the oxygenated blood is pumped into the aorta, which carries it to the various parts of body. During the left ventricle contraction, the blood pressure in arteries is highest and is known as arterial Systolic Blood Pressure (SBP), while the lowest pressure is established during ventricle relaxation period which is called Diastolic Blood Pressure (DBP) [8]. The arterial blood pressure as a function of time is shown in Figure 2.

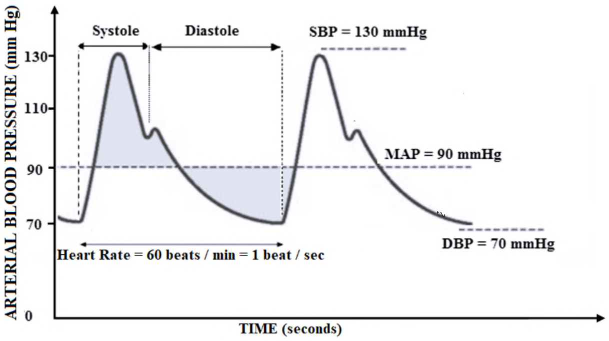


Figure 2. Arterial blood pressure as function of time.

The low blood pressure signifies that the force with which the blood is pushed from the aorta into the distributing arteries in human body will be quite low. In other words, it signifies that the blood could not be supplied in sufficient quantity throughout the human body if blood pressure is too low. Conversely, if the human blood pressure is too high, the blood vessels may be injured [8]. The blood pressure is generally measured in millimeters of Mercury (mm Hg) and the standard blood pressure of a healthy human is generally specified as 120/80 mm Hg, where the larger number (i.e., 120 mm Hg) signifies the Systolic Blood Pressure (SBP) and the smaller number (i.e., 80 mm Hg) indicates the Diastolic Blood Pressure (DBP).

There are two basic types of methods which have been generally employed for the measurement of blood pressure: [9]

(a) Invasive blood pressure measurement,
(b) Non-invasive blood pressure measurement

The invasive methods of blood pressure measurement are *in-vivo* methods, i.e. the blood pressure measurements are performed by inserting a pressure sensor inside the human body. The term '*in-vivo'* is a Latin word which means 'within the living'. Table I compares the two basic types of blood pressure measurement techniques.

TABLE I. COMPARISON OF INVASIVE AND NON INVASIVE BLOOD PRESSURE MEASUREMENT METHODS

| S. No. | Invasive BP measurement | Non-Invasive BP measurement |
|---|---|---|
| 1 | The artery is punctured and pressure sensor is inserted inside the artery for pressure measurement i.e., the sensor is placed inside the human body for pressure measurement. | The blood pressure is measured by the sensor placed in close proximity to the human body and sensor is not inserted inside the human body by puncturing the artery carrying the human blood. |
| 2 | For example, placing the cannula needle inside the radial artery of patient for BP measurement [10] | For example, BP measurement using auscultatory or palpatory method [11] |
| 3 | Requires extremely high expertise as the sensor needs to be inserted into the punctured artery | Requires less expertise as the readings have to be noted by examiner or nurse from mercury manometer scale or from digital display. |
| 4 | The Blood Pressure is measured beat by beat | The Blood Pressure is not measured beat by beat |
| 5 | High accuracy | Comparatively less accurate |
| 6 | High complexity | Less complexity |
| 7 | Requires extensive overhead and time for measurement to be performed | Requires comparatively less overhead and time for measurement to be performed |
| 8 | Risk to the patient is involved | Safe for the patient and non-risky |

## III. NON INVASIVE BLOOD PRESSURE ESTIMATION METHODS

There are three common methods commonly employed for non-invasive blood pressure estimation:

(a) Palpatory method, (manual BP measurement)
(b) Auscultatory method, (manual BP measurement)
(c) Oscillometric method (automated BP measurement)

The above mentioned methods rely on sensing side effects generated on occluding an artery by inflating/ deflating a cuff around a subject's limb. It is to be noted that the palpatory and ausculatory methods of BP measurement are manual methods and involve the doctor or nurse to note the readings manually from the mercury scale, whereas the oscillometric method of BP measurement is an automated method which senses, measures and displays the BP magnitude on a digital display readout. It is also to be noted that the palpatory method was earlier used for measurement of SBP and was not considered suitable for measurement of DBP and Mean Arterial Pressure (MAP). However, a new palpatory technique for DBP pressure measurement has been proposed and is discussed in the following sections.

On the other hand, the auscultatory method is suitable for measurement of both SBP and DBP. It is because of this reason that the auscultatory method is more commonly used for BP measurement over the palpatory method. The palpatory method is limited for BP measurement by doctors in emergency situations [10]. However, the cuff based non-invasive manual BP measurement method requires the patient to position his arm above the level of heart atria and the mercury scale reading to be carefully observed by a nurse or observer for the accurate measurement of blood pressure.

The basic principles of various non-invasive BP measurement methods - auscultatory method, palpatory method and oscillometric method will be described below in detail.

### A. Auscultatory method of BP Measurement

The most common blood pressure measurement method used by doctors and nurses in hospitals and clinics is the auscultatory method, which employs the usage of sphygmomanometer and stethoscope for blood pressure measurement [13]. A sphygmomanometer comprises of an inflatable cuff and mercury manometer. The inflatable cuff is placed around the subject's upper arm (around the brachial artery) and the subject's arm is placed above the level of heart atria for accurate pressure measurement, as shown in Figure 3 [14].

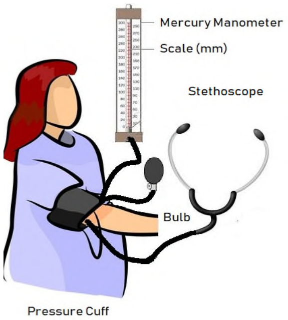


Figure 3. Auscultatory method of non-invasive BP measurement set up.

The cuff is inflated to suprasystolic pressure so that the artery is completely occluded. The cuff pressure is then slowly released and a trained nurse or doctor listens to the Korotkoff sounds with the help of a stethoscope placed between the arm and the cuff in order to identify the SBP and the DBP magnitudes, as shown in Figure 4.

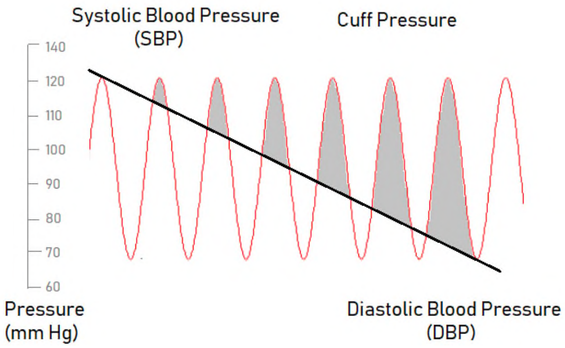


Figure 4. SBP and DBP measurements.

The cuff pressure at which the first Korotkoff sound is heard is the SBP and the pressure at which the sounds becomes muffled is the DBP. It should be noted that the auscultatory BP measurement method requires a trained health practitioner to note the accurate SBP and DBP magnitudes manually [14].

### B. Palpatory method of BP measurement

In the palpatory method of BP measurement, the inflatable cuff is placed around the subject's upper arm (around the brachial artery) at the same height as the human heart for accurate pressure measurement [14]. The cuff is inflated to suprasystolic pressure. The cuff pressure is slowly released and a trained nurse or doctor senses the blood flow by placing a finger on the radial artery at patient's wrist, as shown in Figure 5 [12].

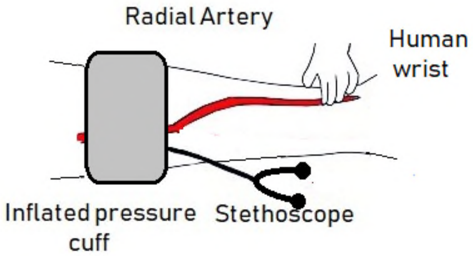


Figure 5. Palpatory method of non-invasive BP measurement set up.

The pressure at which the pulse disappears during inflation, and then subsequently, reappears during deflation is known as SBP. In a recent technique proposed for the measurement of DBP by palpatory method, the doctor places his first three fingers over the elbow bend and tracks the pulsating thrill over the elbow bend as he/she inflates and deflates the pressure cuff. The pressure measured on the manometer scale at which this pulsating thrill disappears is known as DBP [15]. Although it is a new technique for DBP measurement by using the palpation method, this method for BP measurement can lead to errors up to 25 percent and is commonly used by doctors and nurses during emergency medical situations [10].

### *C. Oscillometric method of BP measurement*

The oscillometric method of BP measurement is one of the most common techniques for automated BP measurement and is suitable for the measurement of SBP, DBP and MAP [16]-[18]. The oscillometric principle of BP measurement is based on sensing the pressure pulses within a cuff wrapped over the brachial artery around the patient's arm or over the radial artery at the patient's wrist. The cuff wound around the patient's arm or wrist is inflated to a suprasystolic pressure. The cuff is then slowly deflated and the pressure oscillations are sensed by means of the pressure sensor in the cuff. Figure 6 clearly illustrates the principle used to sense the pressure pulsations by using the pressure sensor inside the cuff.

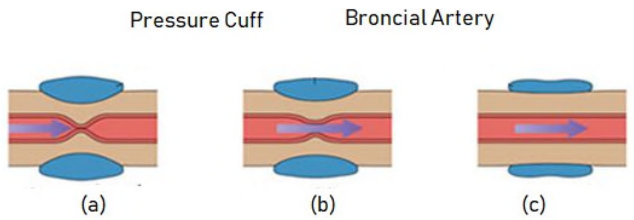


Figure 6. (a) Occluded brachial Artery with cuff pressure greater than 120mm Hg, (b) Blood flowing through relaxed brachial artery when cuff pressure is between 80mm Hg and 120 mm Hg, (c) Silent blood flowing when cuff pressure is below 80 mm Hg .

As the cuff is inflated to a suprasystolic pressure (i.e., greater than SBP), the cuff pressure leads to the occlusion of the artery and the blood flow within the artery stops, as shown in Figure 6(a). The cuff is then deflated slowly which leads to the flow of blood exerting pressure on the walls within the artery. The cuff is then deflated gradually to a subdiastolic pressure where the artery is no longer compressed and the blood starts flowing silently through the artery. During the deflation period (i.e., when the pressure is reduced in the cuff), the cuff pressure is recorded by means of the pressure sensor and the cuff pressure waveform is extracted at output of cuff pressure sensor. This extracted waveform is known as the cuff deflation curve, as shown in Figure 7.

The cuff deflation curve is comprised of two main components: (as shown in Figure 7)

(a) the slow-varying component due to the applied cuff pressure,

(b) pressure pulsations caused by arterial pressure (Oscillometric Waveform - OMW)

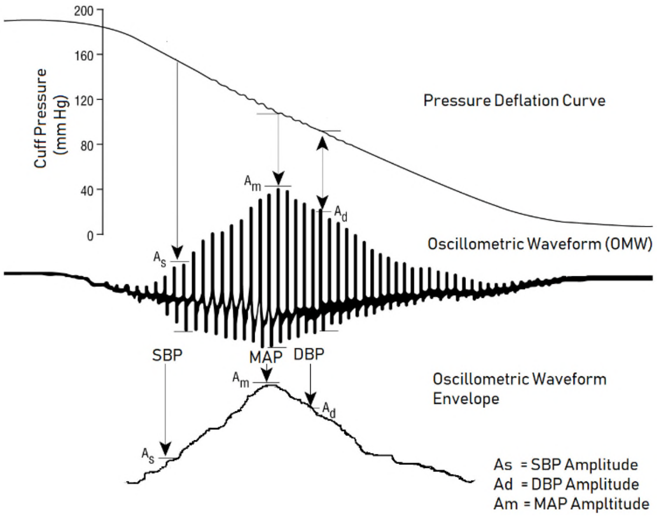


Figure 7. Oscillometric waveform for BP measurement.

The pressure pulsations, also known as Oscillometric Waveform (OMW) pulses, are extracted from the cuff deflation curve by extracting the slowly varying component through filtering techniques [19]-[21] or detrending techniques [22][23]. The filtering method is based on removing the cuff deflation frequency components using a bandpass filter [48]-[50]. The lower cut-off frequency of the filter is usually set to 0.1–0.5 Hz and the upper cut-off frequency of the filter is set around 20 Hz. In the detrending method, the trending curve is subtracted from the cuff deflation curve in order to obtain oscillometric waveforms OMW, as shown in Figure 7. The trending curve is basically a line of best fit which represents the decreasing cuff pressure [24]. It is to be noted that the BP information lies in the amplitudes of OMW and thus, the BP can be estimated from it. Most of the oscillometric algorithms proposed by various researchers for detection of BP are based on analyzing the Oscillometric Waveform Envelope (OMWE) [25]-[29].

The OMWE can be extracted and analyzed in terms of peak-to-peak [23] [31], baseline-to-peak [25], or the area of the oscillometric pulses during the cuff deflation period [33]. The peak-to-peak oscillometric waveform envelope ($OMWE_{p-p}$) is obtained by subtracting the consecutive peaks and troughs of oscillometric waveform (OMW). The baseline-to-peak oscillometric waveform envelope ($OMWE_{b-p}$) is obtained by subtracting the baseline from the peaks of the OMW where the baseline is the cuff deflation curve without the pressure oscillations. The computation of the area of the oscillometric pulses is based on the integration of the oscillometric pulses [29].

TABLE II. COMPARISON OF VARIOUS NON-INVASIVE BLOOD PRESSURE MEASUREMENT METHODS

| S. No. | Parameter | Ausculatory method | Palpatory method | Oscillatory method |
|---|---|---|---|---|
| 1 | Working Principle | Detection of Korotkoff sounds by placing stethoscope over brachial artery with pressure cuff inflated and deflated slowly | Pulse detection by placing finger over radial artery with pressure cuff inflated and deflated slowly | Estimation of pressure from the oscillometric waveforms generated from cuff deflation or inflation waveform |
| 2 | Body target source employed for measurement | Brachial artery at upper arm | Radial artery at wrist | Brachial artery at upper arm or Radial artery at wrist |
| 3 | Output readout | Mercury Manometer | Mercury Manometer | Digital Display |
| 4 | Nature | Manual | Manual | Automated |
| 5 | Complexity | Less | Less | High |

The oscillometric method of blood pressure measurement is more accurate than the auscultatory and palpatory methods of blood pressure measurement techniques. Table II compares the three non-invasive blood pressure measurement techniques. The various algorithms have been proposed by various researchers in the recent years for the estimation of SBP, DBP and MAP magnitudes from the OMWE. The various algorithms proposed by various researchers in the recent years for the detection of BP have been reviewed and compared in the following Section IV.

## IV. CLASSIFICATION OF BP ESTIMATION ALGORITHMS

In the past few years, various BP estimation algorithms have been developed by various researchers. The BP estimation algorithms are usually applied on the signals recorded at the various stages which can be OMW, OMWE or ECG in BP measurement process. However, the BP estimation algorithms are generally applied on OMWE. Figure 8 below shows the basic flow process of automated BP estimation.

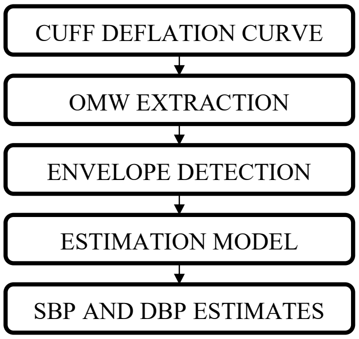


Figure 8. Basic flow process showing the process of BP measurement.

Table III shows the various blood pressure estimation algorithms and their principle of estimation.

TABLE III. VARIOUS BP ESTIMATION TECHNIQUES

| S. No. | BP Estimation Algorithm | Principle of Estimation |
|---|---|---|
| 1 | Maximum Amplitude Algorithm (MAA) | Empirical coefficient detection from the peak amplitude of oscillometric waveform envelope |
| 2 | Derivative Oscillometry | Slope Estimation Detection |
| 3 | Neural Network Estimation | Machine learning approach |
| 4 | Model Based Algorithms | Mathematical modelling of envelope to estimate BP |
| 5 | Pulse Transit Time Estimation | Employ both cuff deflection curve and ECG (heart) signal for BP measurement |

Figure 9 classifies the various BP estimation techniques based on the stage at which the BP estimation algorithm is applied.

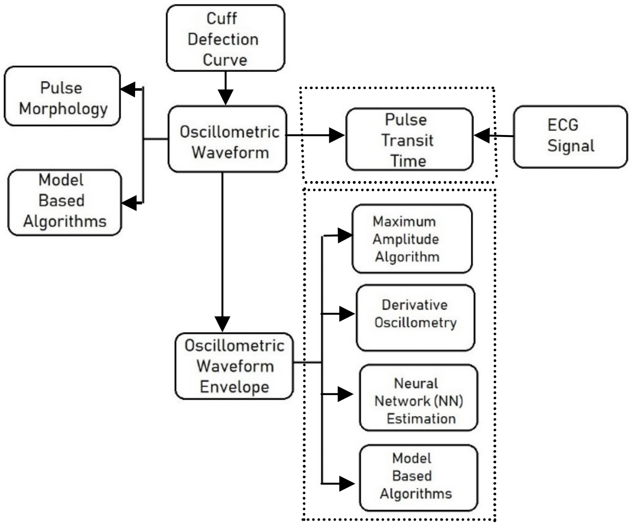


Figure 9. Classification of BP estimation algorithms based on stage at which BP estimation algorithm is applied.

This paper focuses on the comparison of various AI techniques, which are commonly employed for BP estimation applications. Section V focuses on the various AI based BP estimation techniques from OMWE in oscillometric method of BP measurement.

## V. ARTIFICIAL INTELLIGENCE (AI) TECHNIQUES FOR BP ESTIMATION

In the recent years, artificial intelligence and deep learning algorithms have been employed by various researchers for blood pressure estimation from the OMWE. The most common AI technique employed in BP estimation is the artificial neural networks. The Neural Networks (NNs) are suitable for nonlinear physiological systems in the biomedical or instrumentation sector [34]-[37].

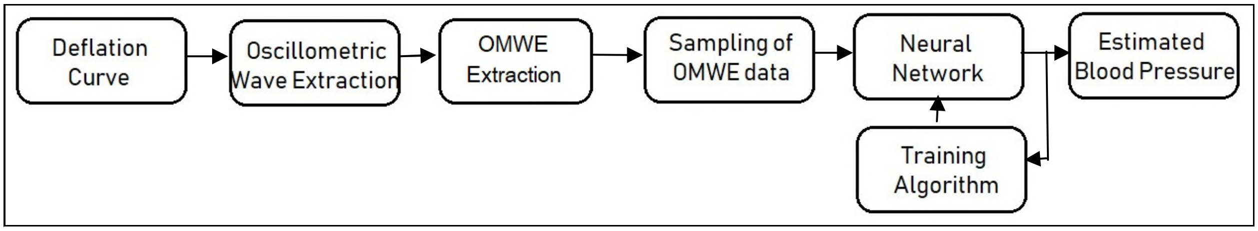


Figure 10. Feed Forward Neural Network (FFNN) based BP estimation using raw OMWE data.

The various NN algorithms that have been employed by researchers in the recent years in the field of BP estimation are listed and explained below.

### *A. Feed Forward Artificial Neural Network for BP estimation using raw OMWE data*

The Artificial Neural Network (ANN) is a set of interconnected artificial neurons arranged in the form of layers forming a system, capable of learning in training mode, and providing the desired output according to the applied input in testing stage. The FFNN can be trained based on the nature and size of sample data and, once trained, the ANN can be tested by presenting different data sets for validating the results. The ANN can be trained and tested to estimate the SBP and DBP by presenting the raw OMWE data sampled at specific increments [38] [39].

Figure 10 shows the methodology to embed neural networks in the oscillometric BP estimation process. There are certain limitations of neural network based BP estimation technique, as mentioned below:

1. The performance of ANN is purely based on the nature of the data presented to the ANN when trained. The effective representation of data can lead to improved learning and the generalization of the network to be employed for estimation purposes. Since the neural network has to be trained for the specific OMWE data, it leads to a poor generalization network [40].
2. The redundant input data leads to a larger number of hidden layers in the neural network for better accuracy [41].
3. As the number of weights in the neural network increases, a large data set is required to train the neural network [43]. However, the collection of a large data set is time consuming and expensive.

Therefore, the raw sampled OMWE data leads to a neural network having more hidden layers and more weights and thus, it may lead to a large ANN network design. Therefore, the size of the ANN can be reduced by reducing the sample size of the OMWE or parameters that capture the essential features of the signal. Hence, the Principal Component Analysis (PCA) approach was introduced, which involves pre-processing of OMWE raw data before applying it to the ANN for training, as discussed in the following subsection.

### *B. PCA based Feed Forward Neural Network (PCA-FFNN) for BP estimation using raw OMWE data*

The various features can be extracted from the envelope of oscillometric waveform, such as, the amplitude of the OMW pulses or height of OMWE, its derivative, width, etc. The basic principle behind the Principal Component Analysis (PCA) based FFNN approach for BP estimation was to reduce the dimensionality of the OMWE by discarding low-variance components that mainly reflect the noise. The compression of data set presented to the FFNN for training can be reduced by feature extraction. A subset of the extracted feature data set is used to train the ANN in PCA based approach, thus ensuring the requirement of small sized ANN for BP estimation since the input data set gets reduced because of the compression of input training data set [41]-[45]. The PCA based ANN approach implementation has been shown in the form of block diagram in Figure 11. The feature set of OMWE data is normalized before applying it to the ANN so that the data set should lie within the specified range, which reduces the chances of getting stuck at local minima [47]. The PCA based ANN approach derives a reduced feature set of OMWE instead of using the features for training the ANN. The optimal parameters of the Gaussian functions can be obtained by minimizing the least squares error between the model signal and the true signal by using the *Levenberg–Marquardt algorithm* (LMA) [45].

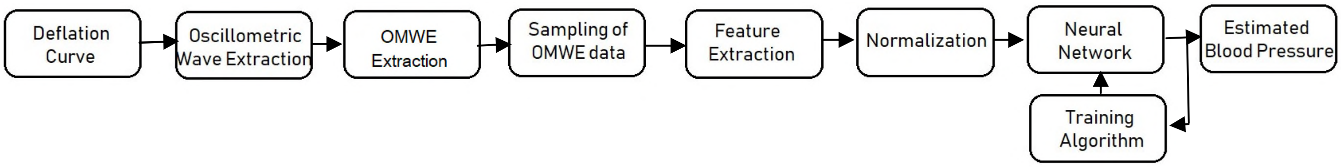


Figure 11. PCA based ANN approach implementation.

ISBN: 978-1-61208-646-0

These optimal parameters have been considered as the features for training two separate Feed Forward Artificial Neural Networks using Resilient Backpropagation (RBP) learning algorithm [46] for estimation of SBP and DBP. The RBP algorithm introduced in year 1994 has been used to train the ANN instead of steepest decent algorithm [41] or steepest descent algorithm with momentum [39] introduced in year 1995 because the RBP has diverse advantages of fast learning rate, small learning data set, do not get stuck at local minima and is robust to noise. In [47], the PCA based ANN approach was implemented for BP measurement from the radial artery and the system was trained and tested using 425 BP readings (5 readings from each of 85 persons) and the range of BP data set ranged from 42-99 mm Hg for DBP and 78-147 mm Hg for SBP.

The number of hidden layers in the FFNN were chosen iteratively for minimum Standard Deviation Error (SDE) and Mean Absolute Error (MAE) by performing the experiments for minimal errors in the output during the training. It was observed that, by reducing the number of inputs from 48 to 5 and hidden layer neurons from 4 to 2, the first layer weight is reduced from 192 to 10. Figure 12 and Figure 13 compare the performance of PCA based SBP and DBP FFNN when tested with raw data for the first time and principal feature based data in terms of number of input layers, number of output layers required, hidden neurons, weight of first layer and weight of second layer [47].

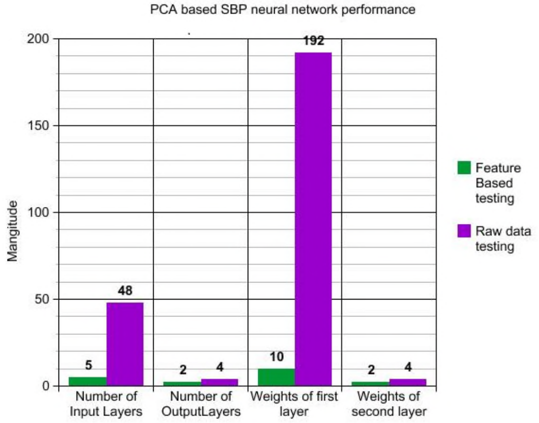


Figure 12. PCA based SBP FFNN performance when tested with raw and feature based data [47].

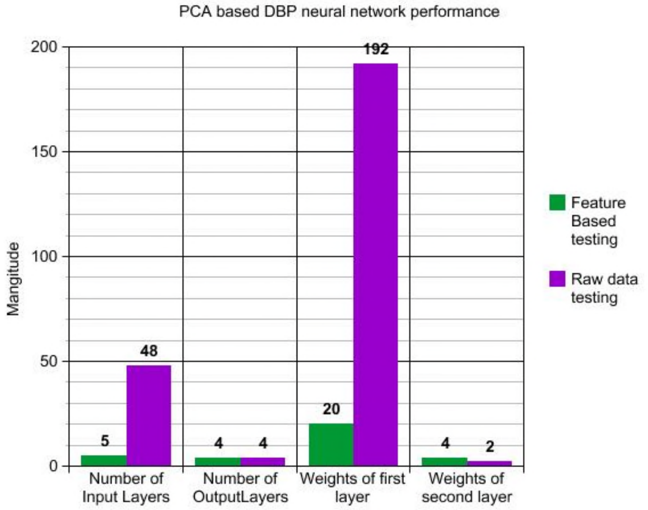


Figure 13. PCA based DBP FFNN performance when tested with raw and feature based data [47].

It can be concluded from Figure 12 and Figure 13 that principle feature based testing of feed forward neural networks[47]:

(a) employs a smaller number of input layers in both SBP and DBP estimation,
(b) employs fewer output layers in SBP estimation,
(c) employs lower weight magnitudes in the first layer in the network in SBP and DBP estimation, and
(d) employs lower weight magnitudes in the second layer in the network in SBP and DBP estimation,

### *C. Adaptive NeuroFuzzy Inference System for BP estimation*

The ANFIS approach consists of three stages, as shown below in Figure 14. In the first stage, the oscillation amplitudes (OAs) of the oscillometric waveforms (OMW) has represented as a function of the cuff pressure. In the second stage, the Principal Component Analysis (PCA) has been utilized to reduce the size of the input training data set by extracting the most effective features from the oscillation amplitudes. In the final stage, the ANFIS has been employed to perform the BP estimation.

The proposed method was tested on the data feature set derived from the 85 patients in 1994 and the results of ANFIS approach was compared with the conventional maximum amplitude algorithm (MAA). It was found that the ANFIS achieved lower values of standard deviation of error (SDE) and Mean Absolute Error (MAE) as compared to MAA approach [48]. The ANFIS system employed the advantages of both ANN and fuzzy logic (FL) for BP estimation. It was observed by B. Kosko in 1994 that, under proper conditions, ANFIS can be used as an universal approximator [48].

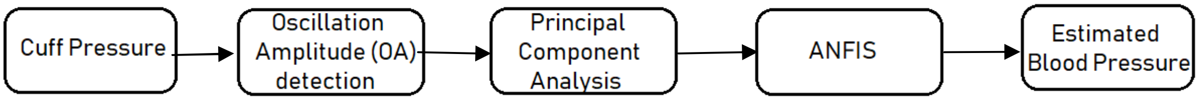

Cuff Pressure
Oscillation Amplitude (OA) detection
Principal Component Analysis
ANFIS
Estimated Blood Pressure

Figure 14. ANFIS approach based BP estimation process.

ISBN: 978-1-61208-646-0

Figure 15 and Figure 16 show the performance of the ANFIS based SBP and DBP networks when tested with raw data and principal feature based data in terms of number of input layers and number of membership functions [47].

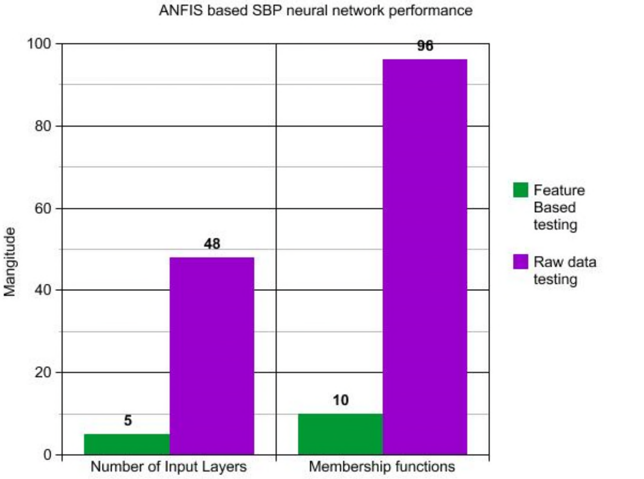


Figure 15. ANFIS based SBP network performance when tested with raw and feature based data [5].

It is to be noted that two separate ANFIS networks were designed for Systolic Blood Pressure (SBP) and Diastolic Blood Pressure (DBP) estimation. The number of hidden layers was chosen by performing the random experiments for minimal Standard Deviation Error (SDE) and Mean Deviation Error (MAE).

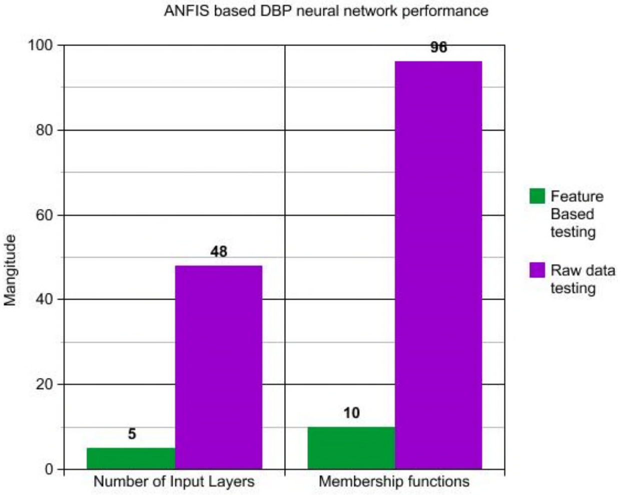


Figure 16. ANFIS based DBP network performance when tested with raw and feature based data [49].

It can be concluded from Figure 15 and Figure 16 that ANFIS employs:

(a) less number of input layers in both SBP and DBP estimation, and
(b) less number of fuzzy membership functions in both SBP and DBP estimation.

### *D. PCA based Cascade Forward Neural Network (CFNN) for BP estimation*

In 2010, the PCA based Cascade Forward Neural Network (PCA-CFNN) for BP estimation was employed for BP measurement which was similar to the PCA based Feed Forward Neural Network discussed in sub-section B above but the difference lies in the fact that there existed a weight connection from input to each layer and from each layer to the successive layer. The experimentation was performed on 85 subjects and five readings were taken for each subject leading to 425 reading set and the principle parameter set was prepared from the subject data in order to constrain the input data train data set used to train the cascade forward neural network using gradient decent algorithm with momentum.

## VI. COMPARISON OF VARIOUS AI BASED BP ESTIMATION TECHNIQUES

Figure 17 and Figure 18 show the comparison of various AI based BE estimation techniques discussed above in terms of the MAE and SDE [47]-[50].

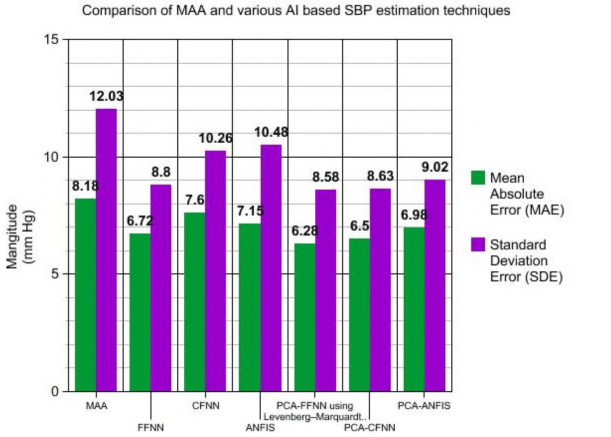


Figure 17. Comparison of various AI based SBP estimation techniques

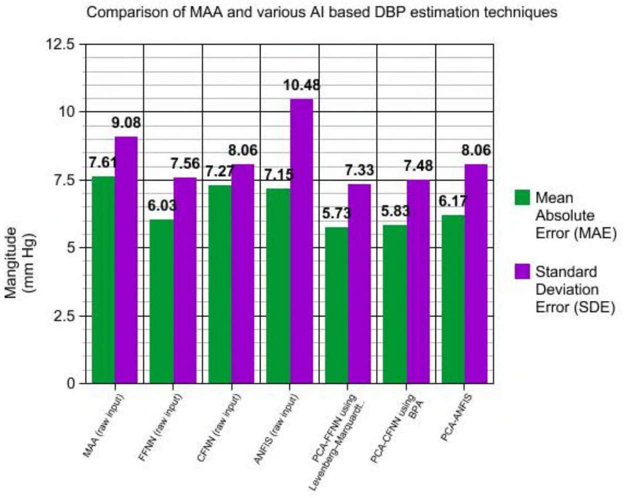


Figure 18. Comparison of various AI based DBP estimation techniques

It can be concluded from the above Figure 18 that the PCA-FFNN using Levenberg–Marquardt algorithm (LMA) has the least MAE and SDE followed by the PCA-CFNN using Back Propagation algorithm with momentum. It can also be observed that the using the PCA approach leads to lower MAE and SDE.

## VII. CONCLUSION

In this review paper, the performance of various AI based BP estimation techniques have been analyzed and compared in terms of SDE and MAE. The effect of employing the PCA with the ANNs has also been reviewed in this paper. It has been concluded that the feature based testing of PCA-FFNN employs:

a) less number of input layers in both SBP/DBP estimation,
b) less number of output layers in SBP estimation,
c) lower weights in the first layer in the network for SBP/DBP estimation, and
d) lower weights in the second layer in the network for SBP/DBP estimation in comparison to the raw testing of the feed forward neural testing.

Therefore, it has been concluded that the complexity of the system gets reduced when using the principle features. It has also been analyzed and concluded from the above discussion that using the PCA approach with ANNs leads to a reduction in MAE and SDE. Further, the PCA-FFNN using Levenberg–Marquardt algorithm (LMA) has the least MAE and SDE in comparison with the other AI based algorithms.

## REFERENCES

[1] P. Lindsay, S. Conner Gorber, M. Joffres, R. Birtwhistle, D. McKay, L. Cloutier, “Recommendations on screening for high blood pressure in Canadian adults,” Can. Fam. Physician, pp. 927-933, 2013.

[2] N.R.C Campbell, B.W. Culleton, D.W. McKay, “Misclassification of blood pressure by usual measurement in ambulatory physician practices,” AM J Hypertens, pp. 1522-1527, 2005.

[3] D.W. Jones, L.J. Appel, S.G. Sheps, E.J. Rocella, Lenfant, “Measuring blood pressure accurately: New and persistant challenges,” JAMA, pp. 1027-1030, 2003.

[4] D.W. McKay, N.R. Campbell, L.S. Parab, A. Chockalingam, JG Fodor, “Clinical assessment of blood pressure,” J. Hum Hypertens, pp. 639-645, 1990.

[5] R.T. Netea, J.W. Lenders, P. Smits, T. Thien, “Arm position is important for blood pressure measurement,” J. Hum Hypertens, pp. 105-109, 1999.

[6] G.L. Peters, S.K. Binder, N.R. Campbell, “The effect of crossing legs on blood pressure: a randomized single-blind cross-over study,” blood Press Monitoring, pp. 97-101, 1999.

[7] F.A. McAlister and S.A. Straus, “Measurement of blood pressure: an evidence based review,” BMJ, pp. 908-911, 2001.

[8] W. Nichols, M. F. O’Rourke, C. Vlachopoulos, “McDonald’s blood Flow in Arteries: Theoretical, Experimental and Clinical Principles,” 6th ed. London, U.K: Hodder Arnold Publishers, 2011.

[9] L. A. Geddes, “The Direct and Indirect Measurement of blood pressure.” Chicago, IL, USA: Year Book Medical Publishers, 1970.

[10] M. Ward and J. A. Langton, “Blood Pressure Measurement,” Contin Educ Anaesth. Crit Care Pain, vol. 7, pp. 122–126, 1997.

[11] E. O’Brien, “Replacing the mercury sphygmomanometer — requires clinicians to demand better automated devices,” Brit. Med. J., vol. 320, pp. 815–816, 2000.

[12] J. Penaz, “Photoelectric measurement of blood pressure, volume and flow in the finger,” Int. Conf. Medical Biological Engineering, Dresden, Germany, pp. 104, 1973.

[13] H. H. Rachmat, "Comparison of radialis sphygmomanometer in evaluating the blood pressure of healthy volunteers," 1st International Conference on Biomedical Engineering (IBIOMED), Yogyakarta, pp. 1-4, 2016.

[14] M. G. Myers and M. Godwin, “Automated office blood pressure,” Can. J. Cardiol., vol. 28, pp. 341–346, 2012.

[15] D. Sahu and M. Bhaskaran, “Palpatory method of measuring diastolic blood pressure,” J. Anaesthesiol. Clin. Pharmacol, vol. 26, pp. 528–530, 2010.

[16] M. Nitzan, “Automatic noninvasive measurement of arterial blood pressure,” IEEE Instrum. Meas. Mag., vol. 14, no. 1, pp. 1094–6969, Feb. 2011.

[17] W. Meyer-Sabellek, M. Anlauf, R. Gotzen, L. Steinfeld, “Blood Pressure Measurements: New Techniques in Automatic and in 24-hour Indirect Monitoring,” New York, NY, USA: Springer, 2011.

[18] T. G. Pickering, J. E. Hall, L. J. Appel, B. E. Falkner, J. Graves, M. N. Hill, D. W. Jones, T. Kurtz, S. G. Sheps, E. J. Roccella, “Recommendations for blood pressure measurement in humans and experimental animals–Part 1: blood pressure measurement in humans: A statement for professionals from the subcommittee of professional and public education of the American Heart Association council on high blood pressure research,” Hypertension, vol. 45, pp. 142–161, Dec. 2005.

[19] M. Ramsey, “Noninvasive automatic determination of mean arterial pressure,” Med. Biol. Eng. Comput., vol. 17, pp. 11–18, 1979.

[20] L. A. Geddes, M. Voelz, C. Combs, D. Reiner, C. F. Babbs, “Characterization of the oscillometric method for measuring indirect blood pressure,” Ann. Biomed. Eng., vol. 10, pp. 271–280, 1982.

[21] H. Sorvoja, R. Myllyl, P. Krj-Koskenkari, J. Koskenkari, M. Lilja, Y. Antero, “Accuracy comparison of oscillometric and electronic palpation blood pressure measuring methods using intra-arterial method as a reference,” Mol. Quant. Acoust., vol. 26, pp. 235–260, 2005.

[22] J. N. Amoore, “Extracting oscillometric pulses from the cuff pressure: Does it affect the pressures determined by oscillometric blood pressure monitors?” Blood Pressure Monitoring, vol. 11, pp. 269–279, 2006.

[23] V. Jazbinsek, J. Luznik, Z. Trontelj, “Non-invasive blood pressure measurements: Separation of the arterial pressure oscillometric waveform from the deflation using digital filtering,” in Proc. Eur. Med. Biomed. Eng. Conf., Prague, Czech Republic, Nov. 2005, pp. 1–4.

[24] S. Ahmad, S. Chen, K. Soueidan, I. Batkin, M. Bolic, H. Dajani, V. Groza, “Electrocardiogram-assisted blood

pressure estimation," IEEE Trans. Biomed. Eng., vol. 59, no. 3, pp. 608–618, Mar. 2012.

[25] G. W. Mauck, C. R. Smith, L. A. Geddes, J. D. Bourland, "The meaning of the point of maximum oscillations in cuff pressure in the indirect measurement of blood pressure—Part II," J. Biomech. Eng., vol. 102, pp. 28–33, 1980.

[26] F. K. Forster and D. Turney, "Oscillometric determination of diastolic, mean and systolic blood pressure—A numerical model," J. Biomech. Eng., vol. 108, pp. 359–364, Nov. 1986.

[27] W. T. Link, "Techniques for obtaining information associated with an individual's blood pressure including specifically a stat mode technique," U.S. Patent 4 697 596 A, Oct. 13, 1987.

[28] G. Drzewiecki, R. Hood, H. Apple, "Theory of the oscillometric maximum and the systolic and diastolic detection ratios," Ann. Biomed. Eng., vol. 22, pp. 88–96, Jan. 1994.

[29] M. Ursino and C. Cristalli, "A mathematical study of some biomechanical factors affecting the oscillometric blood pressure measurement," IEEE Trans. Biomed. Eng., vol. 43, no. 8, pp. 761–778, Aug. 1996.

[30] M. Ursino and C. Cristalli, "Techniques and applications of mathematical modeling for noninvasive blood pressure estimation," in Biomechanical Systems Techniques and Applications, Volume II: Cardiovascular Techniques, C. Leondes, Ed. Boca Raton, FL, USA: CRC Press, 2000.

[31] V. Jazbinsek, J. Luznik, S. Mieki, Z. Trontelj, "Influence of different presentations of oscillometric data on automatic determination of systolic and diastolic pressures," Biomedical Engineering, vol. 38, pp. 774-787, 2010.

[32] G. Gersak, V. Batagelj, J. Drnovsek, "Oscillometric virtual instrument for blood pressure measurement," in Proc. 18th Imeko World Congr., Rio de Janeiro, Brazil, pp. 1–5, 2006.

[33] A. Ball-llovera, "An experience in implementing the oscillometric algorithm for the non-invasive determination of human blood pressure," in Proc. 25th Annu. Int. Conf. IEEE Eng. Med. Biol. Soc., Cancun, Mexico, pp. 3173–3175, Sep. 2003.

[34] C. Alippi and V. Piuri, "Experimental neural networks for prediction and identification," IEEE Trans. Instrum. Meas., vol. 45, no. 2, pp. 670–676, Apr. 1996.

[35] D. De La Mata-Moya, M. P. Jarabo-Amores, M. Rosa-Zurera, J. C. N. Borge, F. Lopez-Ferreras, "Combining MLPs and RBFNNs to detect signals with unknown parameters," IEEE Trans. Instrum. Meas., vol. 58, no. 9, pp. 2989–2995, Sep. 2009.

[36] S. G. Mougiakakou, I. K. Valavanis, N. A. Mouravliansky, K. S. Nikita, A. Nikita, "DIAGNOSIS: A telematics-enabled system for medical image archiving, management and diagnosis assistance," IEEE Trans. Instrum. Meas., vol. 58, no. 7, pp. 2113–2120, Jul. 2009.

[37] A. Malhi and R. X. Gao, "PCA-based feature selection scheme for machine defect classification," IEEE Trans. Instrum. Meas., vol. 53, no. 6, pp. 1517–1525, Dec. 2004.

[38] P. D. Baker, J. A. Orr, D. R. Westenskow, T. P. Egbert, "Method for determining blood pressure utilizing a neural network," U.S. Patent 5 339 818, Aug. 23, 1994.

[39] S. Narus, T. Egbert, T. Lee, J. Lu, D. Westenskow, "Noninvasive blood pressure monitoring from the supraorbital artery using an artificial neural network oscillometric algorithm," J. Clin. Monit., vol. 11, no. 5, pp. 289–297, Sep. 1995.

[40] C. M. Bishop, "Neural networks and their applications," Rev. Sci. Instrum., vol. 65, no. 6, pp. 1803–1832, Jun. 1994.

[41] W. S. Sarle, "Neural Network FAQ," Periodic Posting to the Usenet Newsgroup comp.ai.neural-nets, Jun. 1, 2010. [Online]. Available: ftp://ftp.sas.com/pub/neural/FAQ.html

[42] G. Dorffner and G. Porenta, "On using feedforward neural networks for clinical diagnostic tasks," Artif. Intell. Med., vol. 6, no. 5, pp. 417–435, Oct. 1994.

[43] M. Forouzanfar, H. R. Dajani, V. Z. Groza, M. Bolic, S. Rajan, "Adaptive neuro-fuzzy inference system for oscillometric blood pressure estimation," in Proc. IEEE Int. Workshop MeMeA, Ottawa, Canada, pp. 125–129, Apr./May 2010.

[44] M. Mohamed-Saleh and S. Hoyle, "Improved neural network performance using principal component analysis on Matlab," Int. J. Comput. Internet Manage.,vol. 16, pp. 1-8, May 2008.

[45] G. A. F. Seber and C. J. Wild, Nonlinear Regression. Hoboken, NJ: Wiley-Interscience, 2003.

[46] M. Riedmiller and H. Braun, "A direct adaptive method for faster backpropagation learning: The RPROP algorithm," in Proc. IEEE Int. Conf. Neural Netw., San Fransisco, CA, pp. 586–591, Mar. 1993.

[47] M. Forouzanfar, H. R. Dajani, V. Z. Groza, M. Bolic, S. Rajan, "Feature-Based Neural Network Approach for Oscillometric blood pressure Estimation," IEEE Transactions on Instruments and Measurements, vol. 60, no. 8, Aug. 2011.

[48] B. Kosko, "Fuzzy systems as universal approximators," IEEE Trans. Comput., vol. 43, no. 11, pp. 1329–1333, Nov. 1994.

[49] M. Forouzanfar, H. R. Dajani, V. Z. Groza, M. Bolic, "Adaptive Neuro-Fuzzy Inference System for Oscillometric blood pressure Estimation", IEEE International Workshop on Medical Measurements and Applications, pp. 125-129, 2010.

[50] M. Forouzanfar, H. R. Dajani, V. Z. Groza, M. Bolic, "Oscillometric blood pressure Estimation Using Principal Component Analysis and Neural Networks," IEEE Toronto International Conference Science and Technology for Humanity (TIC-STH), pp. 981-986, 2009.

# Component Models for Embedded Systems in Industrial Cyber-Physical Systems

Luis Neto*†, Gil Gonçalves*†
*SYSTEC-FoF, Research Center for Systems and Technologies - Factories of the Future
†FEUP, Faculdade de Engenharia, Universidade do Porto
Rua Dr. Roberto Frias, s/n 4200-465, Porto, Portugal
Email: {lcneto,gil}@fe.up.pt

***Abstract*—Component Based Software Engineering is a traditional methodology that has significant advantages: reduction of production cost, code reuse, code portability, fast time to market, systematic approach to system construction and guided system design by formalization and domain specific modelling languages. This methodology is used in frameworks for enterprise systems, user interfaces, Web applications, embedded systems for Industrial Cyber Physical Productions Systems (ICPPS) and Industrial Internet of Thing's (IIoT). In this work, we surveyed component model solutions and literature applied to Industrial Cyber Physical Systems (ICPS). By conducting a reproducible systematic mapping study, we search and select results of interest. Research Questions (RQs) are formulated and addressed by applying classification schemes to the results. Finally, the classification results allow us to come up with a state-of-the art in this domain and to draw some conclusions about design considerations and research trends.**

***Keywords–Component Based Software Engineering; Component Models; Embedded Systems; Industrial Cyber-Physical Systems.***

## I. Introduction

In this paper addresses a systematic mapping study in component models for embedded systems in industrial environments. All iterations of the systematic mapping process are based in [1] and are detailed throughout the document, finishing with results that consider the research questions shown below. Heineman and Councill [2] provide a clear and unambiguous definition of *software component*, *component model* and *software component infrastructure* that we will use as reference throughout this paper.

1) **RQ1.** Which component models exist whose scope of application is ICPS and whose targets are embedded systems?
2) **RQ2.** What are the similarities and design considerations among them?
3) **RQ3.** How has research in this topic been evolving?
4) **RQ4.** What kind of contribution is given by particular papers?

A software architecture designed using the component model solution is developed as ”a composite of sub-parts rather than a monolithic entity” [3]. The advantages of this approach address many objectives of software industry, such as: reduction of production cost, code reuse, code portability, fast time to market, systematic approach to system construction and guided system design by formalization and domain specific modelling languages.

The component model is the foundation of a component based design. It defines, briefly, the composition standard: how components are assembled into larger pieces, how and if they can be composed at design and/or runtime phases of component life-cycle, how they interact, how the component repository (if any) is managed, and the runtime environment that contains the assembled application. Because of all of this, component models are hard to build. Some problems like achieving determinism and real-time, parallel flows of component and system development, maintaining components for reuse, different levels of granularity [4] and portability problems [5] may occur.

Our study focuses on component models whose target are factory shop floor systems, and component models that allow to compose solutions for discrete or continuous control and automatic reasoning, the so called component-based industrial automation applications. Component based design architectures, as classified in Vyatkin's work [6], are part of traditional software engineering methodologies. From the key areas of software engineering [6], we will focus our attention on software design and construction, configuration management, tools and methods.

We are interested in covering various levels of components, from those that represent lower level parts of embedded systems (such as drivers and system kernels) to higher level (such as algorithms and services). Component models can be characterized by their capability to assemble components. These can be composed using wrapping, static and dynamic linking, and ”plug-and-play” methods. Component models are, typically, thin layers that operate on top of an operating system (OS) or runtime environment (RTE), which brings portability and reuse issues. Because of the advent of IIoT and ICPS, many hardware vendors are providing heterogeneous solutions that require OS and RTE independent solutions.

The rest of this paper is organized as follows: Section II provides details of the search and selection process for articles; Section III discusses some of the results found to provide the reader with support for better interpretation of the mapping process, later explained in Section IV. Section V concludes the paper with a final discussion.

## II. Primary Search

In [1], the authors present a series of steps that show how to perform a systematic mapping study. These steps are illustrated in Figure 1. The information sources for the first iteration of the study were only databases of reference: *SCOPUS, IEEExplore* and *ACM Digital Library*. The initial search string used clearly reflects the research questions: ***( TITLE-ABS-***

***KEY ( component model ) AND TITLE-ABS-KEY ( industry ) AND TITLE-ABS-KEY ( embedded systems ) )***

Following the systematic mapping process, we did a first review of abstracts and selected a first set of documents based on the criteria of Table I. Because every research topic has a specific terminology unknown to the unfamiliar reader, new keywords of interest (e.g., *Software Component Framework, Component-Based Software, Component Life Cycle, Component Syntax, Component Semantics, Component Composition*) to the RQs were identified to increase the search efficiency.

TABLE I. INCLUSION AND EXCLUSION CRITERIA

| Inclusion | Exclusion |
|---|---|
| • Books and papers reporting final solutions, methodologies and evaluation of component models for embedded systems in industrial scenarios.<br>• Available and existing solutions (both commercial and academic) with documentation reporting experiments, validation and use cases.<br>• Opinion, survey, taxonomy and classification frameworks, and philosophical findings on component models for embedded systems in industrial scenarios. | • Books and papers with less than 10 references will be excluded.<br>• Any finding that does not discuss the three main keywords in the abstract and introduction ”component model”, ”embedded systems” and ”industry” will be excluded.<br>• Component frameworks with exclusive application to enterprise systems, user interfaces, web-applications and others rather embedded systems for industrial domain. |

### A. Search and Screening

To the original set of steps - the blocks represented with a contiguous outline - were added those outlined by dashed lines of Figure 1. The first search query was built combining the most frequent words of the accepted papers and the RQs. To produce this set of frequent words the keywords and abstracts of all accepted documents were gathered in a spreadsheet and parsed. *RapidMiner Studio* [7] was the text analyser tool used to count frequent words. Sequentially, this tool also allowed to use an English *stopwords filter* and a *n-Grams* operator, which allows to make combinations of *n* keywords, to count frequencies of up to 4 consecutive words. After processing, the resulting set of keywords contained 44 keywords of interest. performing combinations with this set was a time consuming process, so we tried to query the selected databases with the entire set at a time but none of them accepted such a long query. After that, we decided to try the *Google Scholar* search engine. This option was viable because it accepted the long set of keywords and this resulted in very accurate preliminary results. The results from the two queries were merged to obtain an extended set of papers. At that point, we decided that to perform a pragmatic application of criteria. The number of citations considered could not be the same because *Google Scholar* takes in account citations from a wider set of sources than the other databases. To solve this issue, each individual paper of the first set was searched in *Google Scholar*. Then, for each paper found, the multiplicative factor between the number of citations in the first and second set was calculated. Finally, the average of all multiplicative factors was calculated. This average value was used to replace the minimum number of references considered in Table I. For a *Google Scholar* paper in the second set to be considered it must have a minimum number of 36 citations.

The application of the exclusion and inclusion criteria specified in Table I drastically reduced the number of documents considered in this study, as can be observed in Table II. The final set of documents was used to conduct the evaluation. For that, a specific classification scheme was combined with the mapping process. This is detailed in the next chapters.

## III. Mapping Process

The following works, after a complete reading were the ones of major interest for this study and will be used throughout the mapping process: *PECOS* [8], Timing Definition Language (TDL) [9], *FORMULA* [10], *Bold Stroke* [11], *Rubus* [12], Real-Time-Linux-Based Framework enhanced with *IEC 6144*9 [13], *IEC 61449* model [14], Programming Temporally integrated distributed embedded systems (PTIDES) [15], *Kevoree* [5], [16], Automatic Reasoning [17], Critical Scenario simulation using *IEC 61449* [18] and Component Design to tackle safety analysis [19]. Some of the papers analysed did not provide enough details to fill rigorously the classification schemes adopted, but all were of the highest interest to provide insight in this study.

Figure 2 gives a concise overview of a component model. It shows two main phases, from a component creation to its usage. In the first stage, the component is built in a builder environment, which can be a code editor (mostly when developing from scratch) or in a graphical editor (mostly when using reusing built components to produce a composite component, these are normally represented by graphic shapes or diagrams). The design phase ends with the developer sending the component to a repository. In some cases, when there is no repository, the component can be directly sent to a RTE. In the deployment phase, components are fetched from the repository, composed in a graphical or code environment and finally sent to the RTE.

### A. Classification Schemes

Four classification schemes will be taken into account to perform the mapping of papers found. The first classification scheme divides the results in: opinion, survey, taxonomy and classification frameworks, and philosophical findings. The second scheme is based on available and existing solutions (both commercial and academic) providing documentation reporting: experiments, validation and use cases. The third classification, which is based on previous ones, specifically addresses RQ3. The last classification scheme addresses RQ4 and the categories used are based in [1].

Some classification categories can not be applied to some papers from the extended set of relevant works. For examples theoretical and other survey papers does not apply to the choose taxonomy for component models. For this reason the number of references in the classification tables is not consistent.

*1) Taxonomy Based:* There exists literature [20, 21, 22, 23] that propose classification schemes specifically for component based software engineering. In [22], the authors provide a formal and comprehensive framework of classification that will

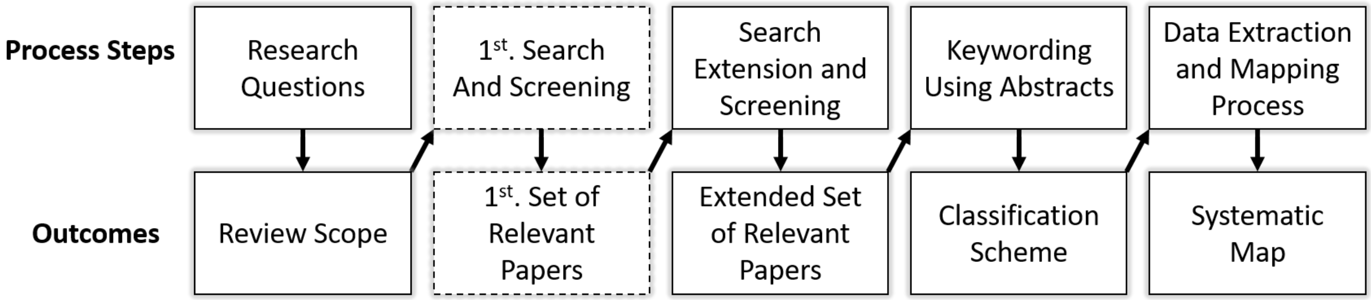

Process Steps
Research Questions
1st. Search And Screening
Search Extension and Screening
Keywording Using Abstracts
Data Extraction and Mapping Process
Outcomes
Review Scope
1st. Set of Relevant Papers
Extended Set of Relevant Papers
Classification Scheme
Systematic Map

Figure 1. Modified Systematic Mapping Process.

TABLE II. DOCUMENTS AFTER CRITERIA

| | *SCOPUS* | *IEEExplore* | *ACM Digital Library* | *Google Scholar* | Duplicates |
|---|---|---|---|---|---|
| Initial (Duplicates) | 390 | 206 | 135 | 913 | 133 |
| >10 References (Duplicates) | 42 | 17 | 14 | 71 | 10 |
| Text Analysis (Duplicates) | 5 | 4 | 4 | 5 | 1 |
| Final Set | 18 | | | | |

not used because of the superficial nature of the reviewing process in systematic mapping approaches. The taxonomy that we will address is proposed in [20] and it classifies component models using the following three characteristics.

- **Component Syntax**: The syntax of components is the *component definition language*. In some cases it is a programming language, but if the solution is required to be more flexible it can be a specific language defined by the component model. In the last case, a compiler can generate code in various programming languages and make the components more versatile. Table III shows the syntax of the component models analysed.
- **Component Semantics**: The semantics of a component is what it is meant to be: it can be an object (in the sense of object oriented languages), it can be a plain piece of business logic code and also to be manipulated by a manager instance created by the container at deployment phase. In this sense, the semantic is given by the run-time environment and defined by the component model. Table IV shows the semantics of the component models analysed.
- **Composition**: Process in which components are assembled together to create new components or systems. This process can happen in two phases (Figure 2) of the software component life-cycle: at deployment phase, the builder environment is able to retrieve existing components from the repository and use them to create a new one, that in the end packaged, catalogued and sent to repository; at deployment phase, existing components in the repository can be assembled and later instantiated in a run-time environment.

TABLE III. COMPONENT SYNTAX

| Component Syntax | Component Model |
|---|---|
| Object Oriented Programming Language | |
| IDL (interface definition language) | [12, 5] |
| Architecture Description Languages | [8, 9, 10, 13, 14] |

TABLE IV. COMPONENT SEMANTICS

| Component Semantics | Component Model |
|---|---|
| Classes | [12] |
| Objects | [13, 14, 5] |
| Architectural Units | [8, 9, 10] |

TABLE V. COMPOSITION CLASSIFICATION

| Category | Component Models | Characteristics | | | | |
|---|---|---|---|---|---|---|
| | | **DR** | **RR** | **CS** | **DC** | **CP** |
| 1 | [8, 10] | x | x | x | ✓ | x |
| 2 | [12], | x | x | ✓ | x | x |
| 3 | [9] | x | x | x | x | ✓ |
| 4 | [13, 14] | ✓ | ✓ | x | ✓ | x |
| 5 | [5] | x | x | ✓ | x | ✓ |

Regarding the composition classification, the original taxonomy in [20] defines 5 characteristics of composition. These characteristics were mapped into categories for this study. The characteristics are: **DR**, In design phase new components can be deposited in a repository; **RR** In design phase components can be retrieved from the repository; **CS**: Composition is possible in design phase; **DC**, in design phase composite components can be deposited in the repository; **CP**, composition is possible in deployment phase. Table V shows the composition classification for the component models analysed.

2) *Design Considerations:* A component system capable of performing real time was a characteristic perceived as of the highest importance when reading through the chosen papers. This characteristic also introduce some concerns that are typical from the high performance computing domain. Parallelism, (a)synchronism, worst case execution time, events, threads, the mix of hard, soft and non real-time constraints are characteristics that concern to industrial control applications and that are hard to achieve altogether in component models. Integrating technologies from multiple vendors is challenging and often results in fragile tool chains that requires a considerable effort to maintain. This also touches the domain of granularity: a single component can emulate an entire system (coarse grained), benefiting from the reliability and efficiency,

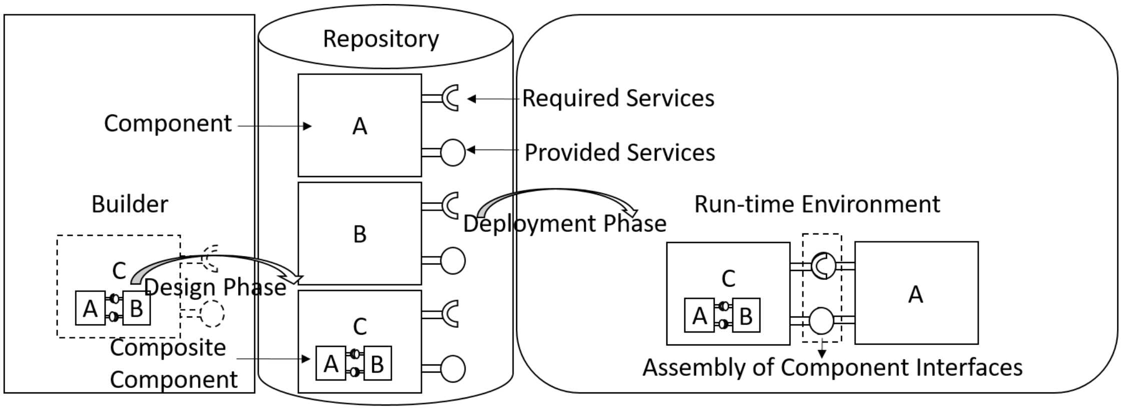


Figure 2. Component Model Overview.

but having a reduced capability of reuse. The footprint of components and its container (the run-time environment) is a recurrent concern when developing to embedded systems, which are typically very constrained. System communication refers to the application of component models to distributed systems. In scenarios where several nodes in a network are distributed physically over a production plant, the component model should be capable of making this nodes interact as components of a monolithic system.

TABLE VI. DESIGN CONSIDERATIONS

| Design Consideration | Component Model |
|---|---|
| Component Granularity | [5] |
| Intelligent Reasoning | [17] |
| Real-time | [12, 21, 9, 11, 13, 14, 15, 18] |
| Security | [5, 19] |
| Footprint | [12, 5] |
| Portability | [10, 10, 13, 14] |
| Component Reuse | [11, 13, 14] |
| System Communication | [21, 9, 14, 15, 5] |
| Systematic Design | [12, 10, 10, 14, 16] |

*3) Design Considerations Over Time:* The graph of Figure 3 shows the evolution of design considerations over the years. Despite the small population used to trace the graph, some conclusions can be drawn. This classification addresses RQ3.

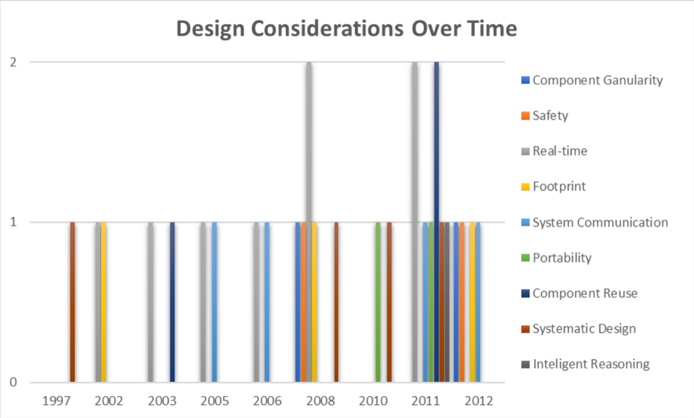


Figure 3. Design Considerations Over Time.

*4) Type of Research and Contribution:* According with the research type facet defined in [1], Table VII shows a classification of works presented in the previous sub chapters. This table addresses RQ4.

## IV. Results and Discussion

In this section we discuss some of the findings with more relevance to the topic. The objective of this analysis and the present discussion is to gain some insight about component models and about some details in the solutions found. There are some design considerations typical of industrial scenarios that this research addresses and are important to retain.

According to Lau et al. [3], components can be divided into 2 main classes, 1) objects, as in OO languages; 2) architectural units, that together compose a software architecture. According to the authors, there are no standard criteria for what constitutes a component model. Components syntax, is the language used to component definition and which may be different from implementation language. Typically the component containers and runtime environments are general purpose server computers. In this case we are interested in a particular kind of architecture in which a centralized general purpose server holds the component repository and the runtime environment is contained in physically distributed embedded systems. The taxonomy that Lau et al. [3] work defines will be used to describe the results found in the systematic mapping study. The authors conclude that a theory that supports component model process in the whole life-cycle does not exist and that a perfect component model should allow composition at design and runtime phases. A component should be deployed along with a complete information of its provided and required interfaces [2]. To enable reuse and interconnection of components, component producers and consumers must agree on a set of interfaces before the components are designed. These agreements can lead to standardized interfaces.

The authors of [21] present a survey of component frameworks for embedded systems, they point out two main difficulties in the development of component systems. The authors also present the evaluation criteria for a real-time component model for embedded systems and compare the frameworks presented against the given criteria. Component frameworks for industrial domain are also presented: *THINK* [24], *MIND* [25] (based in *THINK*) and *SOFA HI* [26]. The classification

TABLE VII. TYPE OF RESEARCH AND CONTRIBUTION CLASSIFICATION

| Contribution Facet | | | | | |
|---|---|---|---|---|---|
| Metric | Tool | Model | Method | Process | |
| | [12, 9, 10, 11] | [12, 10, 11] | [12, 9, 11, 13, 17] | [10, 13, 16, 19] | |
| Research Facet | | | | | |
| Evaluation Research | Validation Research | Philosophical Paper | Experience Research | Opinion Paper | Solution Proposal |
| [12, 13, 17] | [9, 10, 13] | [11, 16] | [12, 9, 10, 11, 16, 19] | | [12, 9, 10, 13, 17] |

criteria and review of the frameworks are very enlightening in the sense that reading this work provides a great deal of insight into component frameworks from various perspectives of application.

Authors in [13] consider component based development as a key promising technology in embedded research domain. Here the authors point out the differences that make component model solutions for general purpose computers not viable to embedded systems. A series of component models for embedded systems in industry (both based in software engineering and control theory best practices) are pointed out. From our experience in recent European projects, industrial component models need to look into disciplines, such as IIoT and machine learning. Beyond control, embedded systems of today smart factories must analyse data, communicate with vendor independent hardware (sensors, machines, actuators, cloud systems and HMI devices) and take actions.

*Rubus* [12] is a component model for embedded systems. This work regards industrial requirements that were elicited considering mixed timing and resource constrained requirements. The components in this solution also have a set of modes and/or a set of states that allows the components to execute distinct code for different system states.

Authors in [8] present a good list of reasons that motivates a component model specific for field devices. A case study in which a single board computer containing the *PECOS* solution and controlling a motor speed was developed in their work. This involved a component for representing a speed sensor and others to encapsulate control algorithms that were specifically developed for this case. The board had both web-access protocols (HTTP over TCP/IP) and an interface for an industrial protocol(*ModBus*). This solution show how components can be passive, in the sense that they are invoked by a scheduler or other components; or they can be active, own a thread to process asynchronous events or perform long computations in background.

## V. Conclusion

To draw more realistic conclusions commercial and other academic and non-academic solutions, which were of our knowledge, but not found during the search phase, should be considered in the evaluation and mapping. Some of them are *Matlab/Simulink* [27], *Node-RED* [28], *Scade* [29], *OSGi* [30] and *4DIAC* [31]. In addition, to make the study reproducible, it is important to mention that intuitive findings (such as when analysing papers and consulting other informal search engines and databases) were not included.

Some interesting conclusions can be taken from the design considerations over time in the graph of Figure 3. There are only two papers considering security issues, the second one [5] is about a component model designed for cyber-physical systems, in which security is a hot-topic. In the same classification line, real-time considerations are shown to prevail over the years. This finding can somewhat confirm that this is a hard subject to tackle in component architectures. Intelligent reasoning is an emergent topic of nowadays, we decided to include that design consideration in the classification scheme of Table VI, exactly to make readers perceive that only in most recent paper of interest [5] it was addressed. This also could mean that security and artificial intelligence are open topics of research in the software engineering component models domain. As we have seen, there are multiple works using *IEC 61499*, it seems to be the de facto standard for component syntax and semantics in industrial automation. Other concerns that seems to prevail are the communication, design and portability of components. Last but not least, apart from commercial and other non-academic solutions, it seems that this topic is not evolving in the recent years. This can also be a signal that the emergent software engineering methodologies for industrial automation [6] are capturing a lot of attention from the academic community.

## References

[1] K. Petersen, R. Feldt, S. Mujtaba, and M. Mattsson, “Systematic mapping studies in software engineering.” in EASE, vol. 8, 2008, pp. 68–77.

[2] B. Councill and G. T. Heineman, “Definition of a software component and its elements,” Component-based software engineering: putting the pieces together, 2001, pp. 5–19.

[3] K.-K. Lau and Z. Wang, “Software component models,” IEEE Transactions on software engineering, vol. 33, no. 10, 2007, pp. 709–724.

[4] C. Maga, N. Jazdi, and P. Göhner, “Reusable models in industrial automation: experiences in defining appropriate levels of granularity,” IFAC Proceedings Volumes, vol. 44, no. 1, 2011, pp. 9145–9150.

[5] F. e. a. Fouquet, “A dynamic component model for cyber physical systems,” in Proceedings of the 15th ACM SIGSOFT symposium on Component Based Software Engineering. ACM, 2012, pp. 135–144.

[6] V. Vyatkin, “Software engineering in industrial automation: State-of-the-art review,” IEEE Transactions on Industrial Informatics, vol. 9, no. 3, 2013, pp. 1234–1249.

[7] RapidMiner, Inc. Rapidminer studio. Last accessed 2018.05.04. [Online]. Available: https://rapidminer.com/products/studio/

[8] T. Genßler, A. Christoph, M. Winter, O. Nierstrasz, S. Ducasse, R. Wuyts, G. Arévalo, B. Schönhage, P. Müller, and C. Stich, “Components for embedded software: the pecos approach,” in Proceedings of the 2002 international conference on Compilers, architecture, and synthesis for embedded systems. ACM, 2002, pp. 19–26.

[9] E. Farcas, C. Farcas, W. Pree, and J. Templ, “Transparent distribution of real-time components based on logical execution time,” in ACM SIGPLAN Notices, vol. 40, no. 7. ACM, 2005, pp. 31–39.

[10] E. K. Jackson, E. Kang, M. Dahlweid, D. Seifert, and T. Santen, “Components, platforms and possibilities: towards generic automation for mda,” in Proceedings of the tenth ACM international conference on Embedded software. ACM, 2010, pp. 39–48.

[11] W. Roll, “Towards model-based and ccm-based applications for real-time systems,” in Object-Oriented Real-Time Distributed Computing, 2003. Sixth IEEE International Symposium on. IEEE, 2003, pp. 75–82.

[12] K. Hanninen, J. Maki-Turja, M. Nolin, M. Lindberg, J. Lundback, and K.-L. Lundback, “The rubus component model for resource constrained real-time systems,” in 2008 International Symposium on Industrial Embedded Systems. IEEE, 2008, pp. 177–183.

[13] G. Doukas and K. Thramboulidis, “A real-time-linux-based framework for model-driven engineering in control and automation,” IEEE Transactions on Industrial Electronics, vol. 58, no. 3, 2011, pp. 914–924.

[14] V. Vyatkin, “Iec 61499 as enabler of distributed and intelligent automation: State-of-the-art review,” IEEE Transactions on Industrial Informatics, vol. 7, no. 4, 2011, pp. 768–781.

[15] J. C. Eidson, E. A. Lee, S. Matic, S. A. Seshia, and J. Zou, “Distributed real-time software for cyber–physical systems,” Proceedings of the IEEE, vol. 100, no. 1, 2012, pp. 45–59.

[16] V. Tran, D.-B. Liu, and B. Hummel, “Component-based systems development: challenges and lessons learned,” in Software Technology and Engineering Practice, 1997. Proceedings., Eighth IEEE International Workshop on [incorporating Computer Aided Software Engineering]. IEEE, 1997, pp. 452–462.

[17] M. Khalgui, O. Mosbahi, Z. Li, and H.-M. Hanisch, “Reconfigurable multiagent embedded control systems: From modeling to implementation,” IEEE Transactions on Computers, vol. 60, no. 4, 2011, pp. 538–551.

[18] M. Khalgui, E. Carpanzano, and H.-M. Hanisch, “An optimised simulation of component-based embedded systems in manufacturing industry,” International Journal of Simulation and Process Modelling, vol. 4, no. 2, 2008, pp. 148–162.

[19] D. Domis and M. Trapp, “Integrating safety analyses and component-based design,” in International Conference on Computer Safety, Reliability, and Security. Springer, 2008, pp. 58–71.

[20] K.-K. Lau and Z. Wang, “A taxonomy of software component models,” in 31st EUROMICRO Conference on Software Engineering and Advanced Applications. IEEE, 2005, pp. 88–95.

[21] P. Hošek, T. Pop, T. Bureš, P. Hnětynka, and M. Malohlava, “Comparison of component frameworks for real-time embedded systems,” in International Symposium on Component-Based Software Engineering. Springer, 2010, pp. 21–36.

[22] I. Crnkovic, S. Sentilles, A. Vulgarakis, and M. R. Chaudron, “A classification framework for software component models,” IEEE Transactions on Software Engineering, vol. 37, no. 5, 2011, pp. 593–615.

[23] H. J. Reekie and E. A. Lee, “Lightweight component models for embedded systems,” in Published as Technical Memorandum UCB ERL M02/30, Electronics Research Laboratory, University of California at Berkeley. Citeseer, 2002.

[24] J.-P. Fassino, J.-B. Stefani, J. L. Lawall, and G. Muller, “Think: A software framework for component-based operating system kernels.” in USENIX Annual Technical Conference, General Track, 2002, pp. 73–86.

[25] MINALOGIC. Mind: Assembly technology for embedded software components. Last accessed 2018.05.04. [Online]. Available: http://www.minalogic.com/en/minalogic/about-minalogic-0

[26] M. e. a. Prochazka, “A component-oriented framework for spacecraft on-board software,” in Proceedings of DASIA. Citeseer, 2008.

[27] The MathWorks, Inc. Simulink - simulation and model-based design. Last accessed 2018.06.14. [Online]. Available: https://www.mathworks.com/products/simulink.html

[28] M. Blackstock and R. Lea, “Toward a distributed data flow platform for the web of things (distributed node-red),” in Proceedings of the 5th International Workshop on Web of Things. ACM, 2014, pp. 34–39.

[29] Esterel Technologies SA - A wholly-owned subsidiary of ANSYS, Inc. Scade suite - control software design — esterel technologies. Last accessed 2018.06.14. [Online]. Available: http://www.esterel-technologies.com/products/scade-suite/

[30] O. Alliance, “Osgi-the dynamic module system for java,” 2009.

[31] T. Strasser, M. Rooker, G. Ebenhofer, A. Zoitl, C. Sunder, A. Valentini, and A. Martel, “Framework for distributed industrial automation and control (4diac),” in Industrial Informatics, 2008. INDIN 2008. 6th IEEE International Conference on. IEEE, 2008, pp. 283–288.

# Process Modeling and Parameter Optimization for Machine Calibration in Smart Manufacturing for Laser Seam Welding

João Reis, Gil Gonçalves

Research Center for Systems & Technologies (SYSTEC)
Faculty of Engineering University of Porto
Porto, Portugal
Email: {jpcreis, gil}@fe.up.pt

***Abstract*—One of the main challenges towards a smart factory is the automation of processes and inclusion of personnel experience in those systems. One of these challenges is related to advances in artificial intelligence that have already been proven to be effective in solving real world problems in the last decade. The problem addressed in this paper is finding the most suitable machine parameters of a laser seam welding process. Once new quality requirements are defined by the customer, normally, a machine calibration phase is required in order to find the proper parameters that yield the desired quality of the product. To address this problem, first a modeling phase was performed to create a suitable model using Artificial Neural Networks (ANNs) that map process parameters onto the observed product quality, and second, the Basin-Hopping search algorithm was used to find the machine parameters needed to achieve a target quality. In order to demonstrate the robustness of the presented approach, three datasets were used that represent three different pairs of materials used for welding in the same machine. The results demonstrate that ANNs are a flexible and robust technique to be used in industry for process modeling and the calibration phase can be minimized.**

***Keywords–Process Modeling; Process Parameter Optimization; Artificial Neural Networks; Smart Manufacturing; Machine Learning.***

## I. INTRODUCTION

The increasing number of product variations as a result of Mass Production to Mass Customization paradigm shift [1] has been leading to the necessity of knowing in detail the machine process dynamics. This is due to the quick change between product variations being produced in a small-lot fashion, or to the introduction of new machines in the shop-floor. This happens mainly because manufacturing companies are getting closer and closer to the end-customer, allowing for customized products composed of multiple options and combinations, and consequently leading to a high number of product variations. This forces the manufacturing companies to be much more responsive to the market needs as a way to increase their market share and create new competitive advantages. However, in order to achieve this level of competitiveness, smarter and innovative ways to explore equipment capabilities and reconfiguration are required. Given the machine operations heterogeneity and shorter production cycles, there is a demand for new techniques that intelligently can operate machinery according to new and diverse product requirements, and rapidly respond and react to these requirement changes, ultimately leading to the automation of the manufacturing process.

Normally, the operation of a certain machine is guided by a set of process parameters that influence process quality that dictate the final result of a certain product. In order to achieve that, the correct process parameters need to be chosen that would yield the correct process quality subject to a set of process conditions. Hence, there is an implicit relation between the influence of machine parameters in the final quality of the product. This way, a good understanding of how process parameters influence the process quality is peremptory for process automation. Normally, the exploration of these relations is made by a set of experiments by performing a Design of Experiment (DoE) - Full Factorial Design or Fractional Factorial Design - to know how of the process parameters map into the process quality. From these experimental findings, normally a dataset is built and machine learning techniques can be used to build process models, which is a simplified version of the real world dynamics - also known as surrogate model. However, as referred before, for the selection of the most suitable process parameters according to certain process quality, this model is necessary but not sufficient. Additionally, an optimization problem is normally formulated to explore the machine parameter feature space that minimizes the distance between the desired process quality and the ones yielded by the process model.

Such an approach is being widely used as a way to perform process optimization as presented in several works reported in the literature. Some examples of such works are [2] and [3] where they use an ANN to model the process using experimental data, and use the concept of Inverse ANN to optimize, using Nelder-Mead algorithm, the process parameters for COD removal in the aqueous treatment of alazine and for energy processes, correspondingly. Another example is presented in [4] where the authors used an ANN to model a thermoplastic joining process and use Genetic Algorithms to find the most suitable process parameters for joining. Moreover, [5] compared Symbolic Regression via Genetic Algorithms with ANN on the modeling and optimization of a controlled drug release of pharmaceutical formulation. For a more thorough understanding of the subject, [6] presents a good review of the High-Dimensional, Expensive (computationally) and Black-box (HEB) problems, presenting multiple examples on a variety of disciplines.

The rest of the paper is organized as follows. Section II details the laser seam welding manufacturing process where this research is focused. Sections III and IV explain how the process modeling and process parameter optimization was performed in this context, leading to Section V where the main

TABLE I. Process Conditions, Process Parameters, Process Quality and Number of Experiments

| | Process 1 | Process 2 | Process 3 |
|---|---|---|---|
| Upper Thickness | 1.5 | 0.6 | 1.2 |
| Lower Thickness | 1.5 | 1.2 | 1.5 |
| P (kW) | 4676.2±666.8 | 4408.3±742.1 | 4594.6±702.6 |
| F (mm) | -0.6±12.6 | -0.3±14.0 | -0.1±12.4 |
| V (mm/s) | 104.8±26.5 | 154.1±32.0 | 120.9±28.8 |
| D (mm) | 0.6±0.3 | 0.6±0.2 | 0.6±0.3 |
| W (mm) | 0.9±0.1 | 1.0±0.1 | 0.9±0.1 |
| Experiments | 188 | 260 | 220 |

results are depicted and discussed. Finally, Section VI draws some conclusions about the performed work.

## II. Laser Seam Welding Scenario

To better understand the presented scenario of process parameter optimization, a description of the process will be given. The laser seam welding process is composed of laser head mounted in a robotic arm with the goal of welding two metal sheets by issuing radiation from the laser head to a local area where the materials need to be joined. Thus, it creates a melting zone around the laser focus in both sheets, which solidifies once the the laser beam is moved through the desired welding area. This produces a continuous welding seam while the beam is moved along the overlapping sheets at a controlled speed. In this particular scenario, the process parameters that can be changed are described by 3 independent variables: Laser Head Power (P); Focal Distance (F) from the surface and Robotic Arm Velocity (V). The observed process quality is described by the Weld Width (W) and Penetration Depth (D) of the welded area. For this work, 3 different datasets are used representing 3 different welding processes in the same machine, where different pairs of materials with different properties and thicknesses were used. These pairs are namely DC04-HC380LA (Process 1), HC260LA-HC420LA (Process 2) and HC420LA-HC380LA (Process 3). Table I presents a summary of the 3 datasets used. If the influence of process parameters over process objectives is explicit in a dataset, machine learning techniques can be used to model this relation, building up a process model.

On top of this information, the process conditions define in which context the process model is valid. For example, if the a process model is trained using the process parameters and quality of two metal sheets, both with 1.5mm of thickness as in Process 1, such a process model becomes obsolete if these thicknesses change, mainly because the relation between process parameters and quality also change. In this context, different thicknesses represent different product variations. As a consequence of that, if a new product variation is introduced in the manufacturing process, this process parameter and quality relation needs to be discovered and detailed as a dataset, so the proper techniques can be used for Process Parameter Optimization. If one wants to explore the relation between already known processes and the conditions that describe the new unseen processes, different machine learning techniques must be applied. Transfer Learning is an emerging research area that is yielding good results in multiple domains, and can be applied to solve the presented problem of learning a new process of a new product variation. In the recent years, the Hyper-Model approach is also being applied to the manufacturing context, which is named as Hyper-Process Modeling [7], to deal with such an issue. However, the details of how these techniques operate are out of the scope of this paper. In the next sections we will present the approach for modeling and optimization in the presented scenario of laser seam welding.

## III. Process Modeling

Since we are modeling a predictor for continuous variables, the presented problem is classified as regression. Hence, the well known Multi-Layer Perceptron (MLP) was used to model an ANN to map machine parameters onto the observed quality data for the laser seam welding process. The concept of artificial neuron is a generalization and simplification of the biological neuron, which is nothing more than a mathematical representation of information processing [8]. This way, the same principle observed in biological systems is then used in the concept of ANNs, where multiple layers of neurons are stacked and connected to perform pattern recognition and predictions. This results in feedforward ANN that proved already of great practical value in solving difficult and specific real life problems.

As its name indicates, for the MLP there are multiple layers of fully connected neurons, meaning that all the neurons of a layer are connected to each neuron of the subsequent layer. These connections are often called weights and dictate how much significance a neuron has to one another. The first layer is called the input layer, the last layer is called output layer, and the remaining in between are called hidden layers. This means that we should have at least three layers to have an MLP, and multiple topologies since these networks can grow by number of hidden layers and number of neurons by hidden layer. Normally, the input and output layers are fixed and correspond to the number of features used for the prediction (independent variables) and number of features that compose the prediction (dependent variables). Based on this, the input of each neuron is composed by the sum of the output of $M$ neurons from the precedent layer and the corresponding weight, and is represented as follows:

$$a_j = \sum_{i=0}^{D} w_{ji}^{(n)} x_i \tag{1}$$

where $j$ is the corresponding layer, $D$ is the number neurons connected to the subsequent layer $j$ plus 1 considering the bias, $w$ is the weight of the corresponding neuron, $n$ is the current layer and finally $x$ is the output of the corresponding neuron. The values of the variable $w$ are called the model parameters. Based on this, the input of a neuron in a subsequent layer can be calculated based on each neuron output ($x$) of the current and its influence ($w$). However, this is simply a linear transformation of data, and no nonlinear dynamics of the system can be grasped. Hence, the calculated input normally is transformed using a nonlinear activation function $h(.)$. This dictates the final form of a neuron output based on the neurons in the previous layer:

$$z_j = h(a_j) \tag{2}$$

Normally, the chosen nonlinear functions are sigmoid or hyperbolic tangent.

Based on this, we have trained our ANN with Adaptive Subgradient Methods for weight optimization [9]. P, F and V specify the inputs feature space $X$ and D and W define the output feature space $Y$, leading to 3 neurons for the input layer and 3 neurons for the output layer. All the neurons from both input and output layers have a linear activation function, while in the hidden layers the sigmoid activation function was used. The number of hidden layers and neurons was obtained experimentally through a trial and error process of all combination of number of layers $L = \{2, 3\}$ and number of neurons per layer $M = \{4, 6, 8, 10\}$. An adaptive learning rate was used starting at 0.5 and decreased once two consecutive epochs fail to decrease the training loss by at least 1e-8, or fail to increase validation score by the same value. For the purposes of training, the input values were normalized between 0 and 1. All the network topologies assessed are depicted in Table II together with the MSE and $R^2$ to evaluate, which one should be used in order to minimize the overfitting effect. As for the training process, a 5-fold Cross-Validation was used for each topology, meaning that 80% of the data was used for the training set, and the remaining was kept aside to assess the generalization capability of the networks. In the training routine of the ANN, the number of epochs was set to 30000, and 10% of the data was used as a validation set during training. After the training process, the network was evaluated in the test set.

Instead of the usual Early Stopping where the training is stopped when the error of the validation set stops decreasing representing overfitting and loss of generality, a Model Checkpoint technique was used. The reason behind not using the Early Stopping lies in the difficulty of specifying a reasonable patience value - number of epochs that the method should wait to stop training once the validation error stops decreasing [10]. On one hand, if the value of patience is set too low, the training might stop before the network converges to a suitable parameter solution, and on the other hand, if the patience is too high, the validation error might increase quickly and model generalization is lost. Both cases depict a situation that we consider not fair to compare networks in terms of performance. The Model Checkpoint just keeps track of the best parameter set regarding the validation error and once the network is trained, the best parameters are returned. This way, we consider this approach to be the most fair for network comparability. However, the main drawback of such an approach is longer periods for training the network due to constant storage and comparability of the best parameters regarding the current parameter set of the ANN. If the cost per minute for training is not a constraint, we strongly encourage to use such an approach.

As main training results, and as can be seen from Table II, for process DC04-HC380LA the best topology regarding the minimization of MSE is 6-6-6, not considering the network input and output layers, where the lower MSE is 0.0086 and $R^2$ of 0.918. This means that the network will have a total of 5 layers, being 2 the input and output layers, together with these 3 hidden layers. As for the HC260LA-HC420LA process, the best topology is 10-10 where the lowest MSE is 0.0063 and a $R^2$ of 0.926. Finally, for the last process HC420LA-HC380LA the minimum MSE found was 0.0084 for a topology of 10-10-10, leading to a $R^2$ of 0.916. Once found these topologies, we need to finalize the models so they can be ready for the following optimization step. For this case, the whole dataset was used to train a ANN with the topology that minimizes the MSE on the test set on 5-fold cross validation, and therefore is the topology that maximizes the generalization of the ANN.

TABLE II. ANN TOPOLOGY ASSESSMENT IN ORDER TO FIND THE MOST SUITABLE MODEL FOR EACH PROCESS.

| Process | ANN Topology | MSE | $R^2$ |
|---|---|---|---|
| DC04 - HC380LA | [4,4] | 0.0120 | 0.883 |
| | [6,6] | 0.0091 | 0.913 |
| | [8,8] | 0.0100 | 0.898 |
| | [10,10] | 0.0110 | 0.897 |
| | [4,4,4] | 0.0100 | 0.904 |
| | **[6,6,6]** | **0.0086** | **0.918** |
| | [8,8,8] | 0.0097 | 0.908 |
| | [10,10,10] | 0.0093 | 0.910 |
| HC260LA - HC420LA | [4,4] | 0.0065 | 0.922 |
| | [6,6] | 0.0065 | 0.923 |
| | [8,8] | 0.0063 | 0.926 |
| | **[10,10]** | **0.0063** | **0.926** |
| | [4,4,4] | 0.0070 | 0.916 |
| | [6,6,6] | 0.0070 | 0.919 |
| | [8,8,8] | 0.0065 | 0.924 |
| | [10,10,10] | 0.0065 | 0.924 |
| HC420LA - HC380LA | [4,4] | 0.0087 | 0.912 |
| | [6,6] | 0.0088 | 0.912 |
| | [8,8] | 0.0088 | 0.913 |
| | [10,10] | 0.0092 | 0.909 |
| | [4,4,4] | 0.0085 | 0.914 |
| | [6,6,6] | 0.0085 | 0.915 |
| | [8,8,8] | 0.0086 | 0.916 |
| | **[10,10,10]** | **0.0084** | **0.916** |

In order to better evaluate the generalization of the trained ANNs, Figure 1 presents the MSE prediction histograms for all the presented welding processes. As can be seen, most of the samples are between the range of 0 and 0.02, being the most of them around 0. Thus, this supports the presented results in Table II where a good performance was achieved with the ANN training using the real datasets provided.

## IV. PROCESS OPTIMIZATION

As the main purpose of training such models is to perform process parameter optimization, we will now assess the performance of the model by providing a set of process quality values from the dataset, and by using optimization algorithms, the best process parameters should be found. This optimization process simulates what could happen in a real scenario when a shop-floor operator needs to know the most suitable machine parameterization in order to meet the customer specifications. In this context, the process quality parameters defined by the customer are the weld width and depth yielding more robust or fragile welds in the final product. Different customers might have different requirements depending on the product application. One might only want to join metal sheets for aesthetics, where not a strong joining is required when compared with a car chassis that should be as strong and robust as possible in the automotive industry. Therefore, based on these quality values, the process parameter optimization should return the parameters to be used in the machine.

More concretely, the process models provide a prediction from a certain $x$ (process parameters) finding the most suitable $\hat{y}$ (process quality). Contrary to this, in the process parameter optimization, the idea is to provide the desired process quality $y$ in order to find the best process parameterization $\hat{x}$. This means that we can specify the customer requirements and

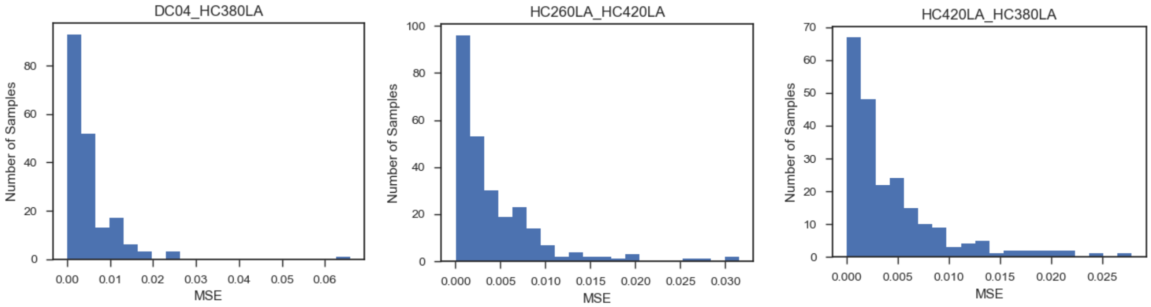


Figure 1. MSE Prediction histograms for the three ANNs trained for each of the processes.

obtain the optimal, or near optimal, machine parameterization. Based on this, a set of optimization routines was made using the trained models to assess how close the parameterization found is from the ground truth. For this test, the whole dataset was used to assess the robustness of the model in the wider range of shop-floor conditions.

For the process parameter optimization the Basin-Hopping algorithm [11] was used to find the most suitable machine parameters by minimizing the difference between the target $y$ and optimized process quality $\hat{y}$. The Basin-Hopping (BH) algorithm was first introduced by Wales and Doye in 1997 to study the lowest-energy structures of Lennard–Jones clusters consisting of up to 110 atoms, and is based on the Monte-Carlo algorithm and gradient-based local search. It is therefore a stochastic algorithm aiming to find the global minimum of a certain function (in this case a loss function) and is mainly based on the following steps: 1) Random perturbation of the coordinates to be tested in the provided function; 2) Step towards the local minimization of the solution; 3) Reject or Accept the proposed coordinates based on the minimization step. As for the acceptance test, the Metropolis criterion is used from the Monte-Carlo application. For this algorithm an initial Temperature of 20 was set to cause large jumps in the loss function value, a number of 20000 iterations for the optimization process and stop after 1000 iterations of no solution improvement. As for the optimization process, the process models are used to iteratively assess a set of process quality values according to a certain process parameters produced by the optimization algorithm. Since these process models have used a normalized dataset between 0 and 1, we have constrained the solution search by the algorithm also between 0 and 1. As an initial guess of a solution, we have set the value to 0 for each of the parameters to be optimized.

Regarding the problem formulation, we aim to minimize the difference between the real process quality (here called target) and the solution generated by the algorithm. For that purpose, the loss function used was simply the MSE to assess these differences. Therefore, 3 defines the minimization problem:

$$\begin{aligned} \hat{x} &= \arg\min_{x} L(\hat{y}, y) \\ &= \arg\min_{x} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 \end{aligned} \tag{3}$$

where $\hat{x}$ is the machine parameterization, $y$ is the target process quality to be achieved, $x$ is the tested input and $\hat{y}$ is the process quality predicted by the process model trained with experimental data.

## V. Results

In the present section, the best optimizations will be depicted as a main result of this paper. However, we must first clarify what is a good or bad optimization process in this context. Intuitively, one might think that a good optimization process is just to find a certain process parameterization that yields the closest process quality considering a defined target. The objective is to minimize a loss function that calculates the distance between what the model produces and the provided target. Hence, as this distance is close to 0, the best. However, in practice, this might not be useful if the difference from the ground truth of process parameters $x$ is too far from the solution found from the algorithm $\hat{x}$.

Hence, Table III depicts the best solutions that minimize the distance between the target quality and the optimized one for all three processes, along with the process parameters to be suggested to the operator in a real application. Additionally, both MSE for process parameters found and resulting process quality are depicted. As can be seen, the obtained MSE for the process quality is very low, meaning that the algorithm used for the optimization process is very effective in finding the global optimum solution. Complementarily, Figure 2 presents the histogram for each process with number of samples in relation to the MSE between target and optimized quality. It can be seen that most of the MSE samples are near the value 0 regarding the total samples present in each dataset of the 3 processes.

However, as previously discussed, this is not very useful if the solution minimizes the distance from the target but the real parameterization is not close to the real application, or if it is even out of the parameterization bounds. This can be observed in Table III in some parameterizations suggested on Process 2 (Opt.), which are not very close to the real parameterization used. Thus, we need to ensure that this is an exception and not the rule.

In order to correctly evaluate the process parameter optimization using the trained process models, not only this distance from the target should be considered, but also the difference between process parameters and the ground truth. Therefore, Table IV presents the 3 best solutions that minimize

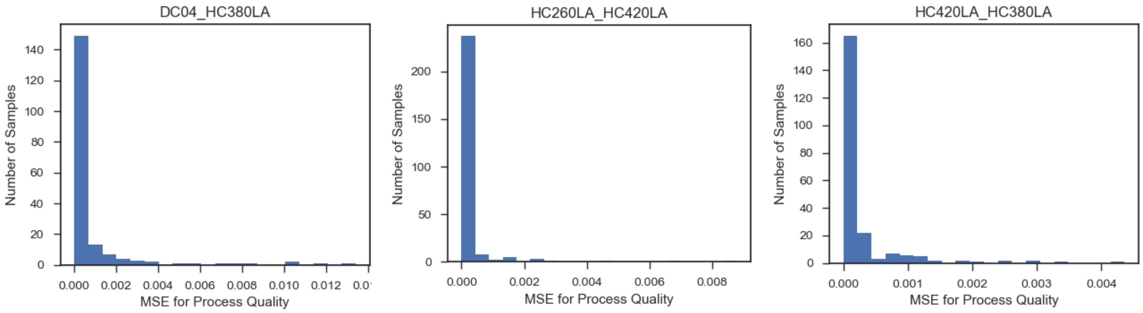


Figure 2. Histogram depicting the number of samples in relation to the MSE for the target and optimized quality.

TABLE III. PROCESS PARAMETER OPTIMIZATION FOR THE BEST 3 SOLUTIONS IN THE TEST SET THAT MINIMIZES THE DISTANCE BETWEEN OPTIMIZED AND REAL QUALITY.

| Process | Case | Parameterization | | | Quality | | MSE | MSE |
|---|---|---|---|---|---|---|---|---|
| | | P | F | V | Depth | Width | Param. | Quality |
| Process 1 | Real | 3500 | -10 | 100 | 0.14 | 0.67 | 0.362 | 1.828e-7 |
| | Opt. | 5500 | -20 | 127.18 | 0.14 | 0.67 | | |
| | Real | 5000 | -10 | 80 | 1.1 | 1.05 | 0.079 | 5.99e-7 |
| | Opt. | 5100 | 8.97 | 61.72 | 1.09 | 1.048 | | |
| | Real | 5000 | 5 | 160 | 0.15 | 0.72 | 0.021 | 6.003e-7 |
| | Opt. | 5385 | 11.45 | 153.46 | 0.149 | 0.718 | | |
| Process 2 | Real | 3500 | -20 | 80 | 0.67 | 1.25 | 0.0711 | 3.848e-10 |
| | Opt. | 4264 | -19.96 | 124.13 | 0.67 | 1.25 | | |
| | Real | 4500 | -20 | 130 | 0.65 | 1.2 | 0.016 | 3.911e-10 |
| | Opt. | 4934 | -20 | 138.86 | 0.65 | 1.2 | | |
| | Real | 4000 | 15 | 170 | 0.21 | 0.71 | 0.414 | 1.495e-9 |
| | Opt. | 5333 | -18.79 | 220 | 0.21 | 0.709 | | |
| Process 3 | Real | 4000 | -15 | 90 | 0.87 | 1.07 | 0.023 | 2.11e-9 |
| | Opt. | 3500 | -15.41 | 75.85 | 0.869 | 1.069 | | |
| | Real | 4500 | 15 | 90 | 0.97 | 1.2 | 0.345 | 1.058e-7 |
| | Opt. | 3500 | -20 | 65.13 | 0.97 | 1.199 | | |
| | Real | 4500 | 15 | 120 | 0.47 | 1.12 | 0.281 | 1.968e-7 |
| | Opt. | 5500 | -15.38 | 142.98 | 0.469 | 1.12 | | |

TABLE IV. PROCESS PARAMETER OPTIMIZATION FOR THE BEST 3 SOLUTIONS IN THE TEST SET THAT MINIMIZES THE DISTANCE BETWEEN OPTIMIZED AND REAL PARAMETERIZATION.

| Process | Case | Parameterization | | | Quality | | MSE | MSE |
|---|---|---|---|---|---|---|---|---|
| | | P | F | V | Depth | Width | Param. | Quality |
| Process 1 | Real | 3500 | -10 | 80 | 0.69 | 1.01 | 9.688e-5 | 1.349e-5 |
| | Opt. | 3500 | -9.77 | 82.73 | 0.689 | 1.004 | | |
| | Real | 5500 | 20 | 120 | 0.12 | 0.73 | 1.655e-4 | 2.405e-4 |
| | Opt. | 5500 | 20 | 123.78 | 0.138 | 0.718 | | |
| | Real | 4500 | 5 | 120 | 0.44 | 0.94 | 4.657e-4 | 1.613e-4 |
| | Opt. | 4437 | 4.36 | 122.28 | 0.422 | 0.937 | | |
| Process 2 | Real | 3500 | -20 | 100 | 0.57 | 1.2 | 3.285e-5 | 1.743e-4 |
| | Opt. | 3500 | -19.99 | 101.68 | 0.573 | 1.181 | | |
| | Real | 5500 | 10 | 220 | 0.48 | 0.86 | 4.501e-5 | 4.66e-4 |
| | Opt. | 5500 | 10.46 | 220 | 0.46 | 0.883 | | |
| | Real | 3500 | -20 | 120 | 0.41 | 1.07 | 8.348e-5 | 3.267e-4 |
| | Opt. | 3500 | -20 | 122.69 | 0.43 | 1.05 | | |
| Process 3 | Real | 3500 | 20 | 60 | 0.93 | 1.21 | 1.55e-6 | 2.095e-5 |
| | Opt. | 3500 | 20 | 59.63 | 0.934 | 1.215 | | |
| | Real | 3500 | -20 | 70 | 0.76 | 1.15 | 3.778e-6 | 3.86e-6 |
| | Opt. | 3500 | -20 | 70.56 | 0.762 | 1.148 | | |
| | Real | 3500 | 20 | 90 | 0.23 | 0.64 | 2.075e-5 | 3.883e-5 |
| | Opt. | 3500 | 20 | 91.34 | 0.222 | 0.645 | | |

the MSE for process parameterization, where a more balanced trade-off between MSEs is achieved. We can see that the presented solutions are near in both process parameters and process quality, being the ideal case in a practical application where a shop-floor operator can truly rely on what the system advises him to do. Hence, in order to understand if these results are consistent throughout the entire dataset, Figure 3 depicts the histogram for each process with the MSE between desired $x$ and optimized process parameters $\hat{x}$. It can be seen that the majority of the samples are around 0, meaning that the process model, together with the optimization technique, are capable of indicating a suitable machine parameterization according to a given process quality. Although, there are some samples with higher errors, also revealing that the process model, for a very small amount of data points is not capable of providing a good indication of machine parameters.

## VI. CONCLUSION

As discussed in the present paper, the process automation is one of the key challenges to be addressed in this fourth industrial revolution, and can be tackled using machine learning. Hence, we will conclude this paper by wrapping up with the pros and cons related with the approach of process parameter optimization and also some future work and research directions.

As for the pros, the first and most obvious is the automation of finding the most suitable machine parameters of a certain process model, or at least give a good initial guess for the machine calibration phase. Moreover, we must also highlight the suitability of ANNs in the context of process modeling, referring its flexibility, robustness and versatility when compared to the difficult process of analytical modeling by experts defining a set of equations that define the process dynamics. Additionally, we should also refer that search algorithms for global optimum are good candidates to address the problem of process parameter optimization and quickly find a close parameterization to the one used in the machine. All together, these factors are of great importance for manufacturing companies that are willing to explore the benefits of key enabling technologies associated with Industry 4.0.

Regarding the cons of such approach, we should refer to the constraint associated with most machine learning techniques of data availability. In order to train a model that should perform well in real world applications, a fair amount of data is required, which is often very difficult in manufacturing systems since these data come from machine experiments and require high material and personnel costs. Moreover, a good understanding of the machine learning algorithms to be used is also required to achieve fair results, otherwise results might not be the most satisfactory for real world scenarios and or even incorrect. Related with this topic, we should highlight approaches to address overfitting, where k-fold cross validation is one of the most widely used approaches when finding hyperparameters for the model, where a wide range

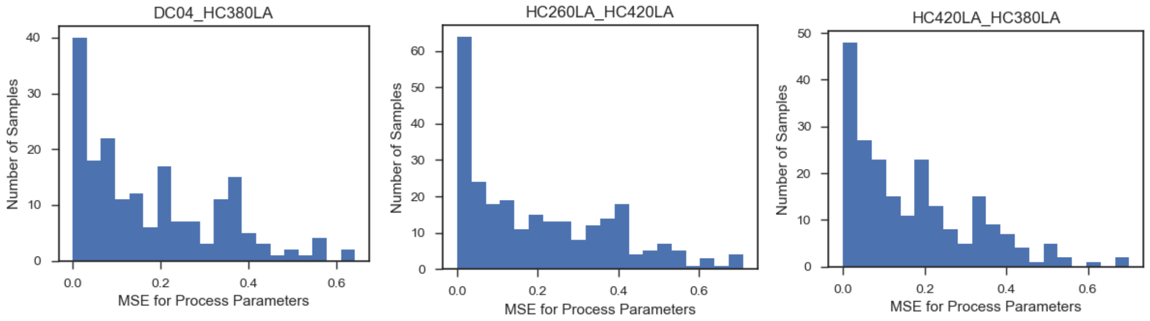


Figure 3. Histogram depicting the number of samples in relation to the MSE for the desired and optimized parameters.

of values need to be tested. Also, one should be aware of model finalization where the model with the parameters that maximize generalization should be trained with the whole dataset, and not only with training set. This is one of the most critical points that should be understood once machine learning models are used in real life applications and not only in scientific papers. Ultimately, as these techniques might tend to increase the complexity once optimizing all its parameters, it is also very important to have experience dealing with such techniques.

As for the future work, there are at least two challenges that we should discuss in the context of manufacturing systems. One of them is the topic of Transfer Learning in manufacturing systems [12] [13]. As one of the presented cons is the amount of data required for modeling, this issue can be tackled with Transfer Learning where the main goal is to improve the learning process of a new task using little amount of data, based on already existing models. In the context of manufacturing systems, this could represent training a process model with a small amount of experiments of a new machine or a new process in an already deployed machine.

Last, but not least, is the topic of Adaptive Learning where the process model is updated during time. It is known that unforeseen events and the inherent degradation of machine components forces to maintenance activities and replacement for new parts that are no longer the same as the initial state of the machine. Complementary Learning System (CLS) theory [14] has brought new promising methods that address the update of a machine learning algorithm as a stream of data is available. The CLS proposed the organization of a learning system in two different parts: 1) Hippocampus as a quick learner of new information with volatile properties and seen as short term memory, and 2) Neocortex, as a high level structural learner with a long term memory [15]. This architecture, which has its roots in neuroscience, have inspired a set of new works that recently tackle the problem of adaptive learning or continuous learning for machine learning systems.

As a conclusion, there are very interesting opportunities for machine learning to enter into manufacturing systems, and help to improve the efficiency and effectiveness of processes through the use of techniques like the ones presented in this work, and many others that still lack the validation in industry.

## REFERENCES

[1] S. Wang, J. Wan, D. Li, and C. Zhang, “Implementing smart factory of industrie 4.0: an outlook,” International Journal of Distributed Sensor Networks, 2016.

[2] Y. E. Hamzaoui et al., “Optimal performance of {COD} removal during aqueous treatment of alazine and gesaprim commercial herbicides by direct and inverse neural network,” Desalination, vol. 277, no. 1–3, 2011, pp. 325 – 337.

[3] J. A. Hernández et al., “Inverse neural network for optimal performance in polygeneration systems,” Applied Thermal Engineering, vol. 50, no. 2, 2013, pp. 1399–1406.

[4] X. Wang et al., “Modeling and optimization of joint quality for laser transmission joint of thermoplastic using an artificial neural network and a genetic algorithm,” Optics and Lasers in Engineering, vol. 50, no. 11, 2012, pp. 1522–1532.

[5] P. Barmpalexis, K. Kachrimanis, A. Tsakonas, and E. Georgarakis, “Symbolic regression via genetic programming in the optimization of a controlled release pharmaceutical formulation,” Chemometrics and Intelligent Laboratory Systems, vol. 107, no. 1, 2011, pp. 75–82.

[6] S. Shan and G. G. Wang, “Survey of modeling and optimization strategies to solve high-dimensional design problems with computationally-expensive black-box functions,” Structural and Multidisciplinary Optimization, vol. 41, no. 2, 2010, pp. 219–241.

[7] J. Reis, G. Gonçalves, and N. Link, “Meta-process modeling methodology for process model generation in intelligent manufacturing,” in IECON 2017 - 43rd Annual Conference of the IEEE Industrial Electronics Society, Oct 2017, pp. 3396–3402.

[8] F. Rosenblatt, “Principles of neurodynamics: Perceptrons and the theory of brain mechanisms.” Washington: Spartan Books, 1962.

[9] J. Duchi, E. Hazan, and Y. Singer, “Adaptive subgradient methods for online learning and stochastic optimization,” Journal of Machine Learning Research, vol. 12, no. Jul, 2011, pp. 2121–2159.

[10] L. Prechelt, “Early stopping—but when?” in Neural networks: tricks of the trade. Springer, 2012, pp. 53–67.

[11] D. J. Wales and J. P. Doye, “Global optimization by basin-hopping and the lowest energy structures of lennard-jones clusters containing up to 110 atoms,” The Journal of Physical Chemistry A, vol. 101, no. 28, 1997, pp. 5111–5116.

[12] S. J. Pan and Q. Yang, “A survey on transfer learning,” IEEE Transactions on knowledge and data engineering, vol. 22, no. 10, 2010, pp. 1345–1359.

[13] K. Weiss, T. M. Khoshgoftaar, and D. Wang, “A survey of transfer learning,” Journal of Big Data, vol. 3, no. 1, 2016, pp. 1–40.

[14] J. L. McClelland, B. L. McNaughton, and R. C. O’reilly, “Why there are complementary learning systems in the hippocampus and neocortex: insights from the successes and failures of connectionist models of learning and memory.” Psychological review, vol. 102, no. 3, 1995, pp. 419–457.

[15] R. C. O’Reilly, R. Bhattacharyya, M. D. Howard, and N. Ketz, “Complementary learning systems,” Cognitive science, vol. 38, no. 6, 2014, pp. 1229–1248.

ISBN: 978-1-61208-646-0

# Adaptability in Smart Manufacturing Systems

Gil Gonçalves, João Reis, Rui Pinto
SYSTEC, Research Center for Systems and Technologies
Faculty of Engineering of University of Porto,
Porto, Portugal
email: {gil, jpcreis, rpinto}@fe.up.pt

Michael Peschl
Harms & Wende GmbH
Hamburg, Germany
email: michael.peschl@harms-wende.de

***Abstract*—Adaptability and reconfigurability of the production system are two key enablers to address global competition and a constantly evolving demand. Adaptive and smart manufacturing systems, realized by a variable number of heterogeneous production Smart Components with specialized capabilities, is one promising approach to guarantee a high degree of adaptability to ever changing demand. This paper presents a realization of a smart manufacturing system based on a multi-agent system approach, discusses its values and drawbacks, and presents possible improvements on the conceptual realization.**

***Keywords—smart manufacturing systems; production smart components; adaptability; reconfigurability.***

## I. INTRODUCTION

Rapid changing product portfolios and continuously evolving process technologies require manufacturing systems that are themselves easily upgradeable, into which new technologies and new functions can be readily integrated [1]. This demands increased productivity through highly optimized production processes, creating the need for novel manufacturing control systems able to cope with the increased complexity required to manage product and production variability and disturbances, effectively and efficiently [2], and to implement agility, flexibility and reactivity in mass customized manufacturing.

Increasingly, traditional top-down and centralized process planning, scheduling, and control mechanisms are becoming insufficient to respond to constant changes in these high-mix low-volume production environments [3]. These traditional centralized hierarchical approaches limit the adaptability [4], contribute to reduce the resilience of the system, as well as to reduce the flexibility of planning and contribute to a corresponding increase in response overheads [5]. The ability of a manufacturing system, at all of the functional and organizational levels, to reconfigure itself in order to quickly adjust production capabilities and capacities in response to sudden changes in the market or in the regulatory environment is nowadays a major requirement.

This paper presents a realization of a smart manufacturing system based on a multi-agent system framework to implement the concept of adaptive and reconfigurable factory. Its contributions and limitations are discussed, along with the roadmap for future improvements.

The paper is structured as follows. After presenting the motivation and objectives, Section 2 frames the problem and presents related work. In Section 3, the overall approach is presented and Section 4 presents the multi-agent system-based realization. Section 5 discusses the results, as well as future improvements, and Section 6 presents the conclusions.

## II. RELATED WORK

The manufacturing enterprises of the 21st century are in an environment in which market demand is frequently changing, new technologies are continuously emerging, and competition is global. Manufacturing strategies should therefore shift to support global competitiveness, new product innovation and customization, and rapid market responsiveness. The next generation of manufacturing systems will thus be more strongly time-oriented (or highly responsive), while still focusing on cost and quality. Such manufacturing systems will need to satisfy a number of fundamental requirements, including [6]: Full integration of heterogeneous software and hardware systems within an enterprise, or across a supply chain; Open system architecture to accommodate new subsystems (software, hardware, *peopleware*) or dismantle existing subsystems “on the fly”; Efficient and effective communication and cooperation among different elements (units, lines, cells, equipment) within an enterprise and among enterprises; Embodiment of human factors into manufacturing systems; Quick response to external order changes and unexpected disturbances from both internal and external manufacturing environments; Fault tolerance both at the system level and at the subsystem level so as to detect and recover from system failures and minimize their impacts on the overall performance. Some possible approaches to fulfil these requirements are presented in the next sections.

### *A. Networked Factories and equipment virtualization*

Modern Industries have a continuous need to satisfy their markets at better costs in order to keep their competitive edge. This simple fact creates the continuous need for new products, new production lines and new control methodologies. The FleXible PRoduction Experts for reconfigurable aSSembly technology (XPRESS) project [7], a cooperative European project involving industry and academia, studied this issue in order to define a new flexible production concept. This concept, based on specialized intelligent process units, called *manufactrons*, was able to integrate a complete process chain, and included support for production configuration, multi-variant production lines and 100% quality monitoring [26]. The concept was demonstrated for the automotive, aeronautics and electrical component industries, but it can be transferred to nearly all production processes.

The latest trends in intelligent manufacturing are related with shop-floor equipment virtualization, fostering the easy access to machine information, allowing collaboration among shop-floor equipment and task execution on demand. The *manufactron* concept was further developed under the project called Intelligent Reconfigurable Machines for Smart Plug&Produce Production (I-RAMP[3]). The goal was to shorten the ramp-up phase time and manage the scheduled and unscheduled maintenance phase time. This goal was achieved by the development of the NETwork-enabled DEVices (NETDEVs), which acted as a technological shell to all the industrial equipment, converting it into an agent-like system and tackling the existing gaps between hardware and software [23]. NETDEVs are intelligent agent-based production devices that are responsible to equip the conventional manufacturing equipment - both complex machines, such as industrial PCs or PLC, and sensors & actuators - with standardized communication skills, along with intelligent functionalities for inter-device negotiation and process optimization. By wrapping equipment components with the NETDEV shell, they become equipped with built-in intelligence. This is at the base of the Smart Component concept [24], which will be further explored in Section 3.

### B. *Reconfigurable manufacturing systems*

Reconfigurability has been an issue in computing and robotics for many years. In general, reconfigurability is the ability to repeatedly change and rearrange the components of a system in a cost-effective way. Koren *et al.* [8] define a Reconfigurable Manufacturing Systems (RMS) as being “[..] designed at the outset far rapid change in structure, as well as in hardware and software components, in order to quickly adjust production capacity and functionality [..] in response to sudden changes in market or in regulatory requirements”. Merhabi *et al.* [9] complemented this definition with the notion that “reconfiguration allows adding, removing or modifying specific process capabilities, controls, software, or machine structure to adjust production capacity in response to changing market demands or technologies [..] provides customised flexibility [..] so that it can be improved, upgraded and reconfigured, rather than replaced”.

RMS are seen as a cost-effective response to market changes, that try to combine the high throughput of dedicated production with the flexibility of flexible manufacturing systems (FMS), and are also able to react to changes quickly and efficiently. For this to be accomplished, the system and its machines have to be adapted for an adjustable structure that enables system scalability in response to market demands and system/machine adaptability to new products. RMS are composed of reconfigurable machines and open architecture reconfigurable control systems to produce a variety of parts with family relationships. The structure of these systems may be adjusted at the system level (e.g., adding/removing machines) and at the machine level (changing machine hardware, control software or parameters).

### C. *Industrial applications of agent systems*

Duffie and Piper [10] were one of the first to discuss and introduce a non-hierarchical control approach, using agents to represent physical resources, parts and human operators, and implementing scheduling oriented to the parts. Yet another manufacturing system (YAMS), introduced by Parunak *et al.* [11], applies a contract net technique to a hierarchical model of manufacturing system, including agents to represent the shop floor. The autonomous agents at Rock Island Arsenal (AARIA) [12] control a production system with the goal to fulfil incoming tasks in due time, focusing on the dynamic scheduling, dynamic reconfiguration and in the control of manufacturing systems that fulfil the delivering dates. The manufacturing resources, processes and operations are encapsulated as agents using an autonomous agent approach.

Some relevant approaches have been introduced in this domain. The product resource order staff architecture (PROSA), proposed by Brussel *et al.* [2], is a holonic reference architecture for manufacturing systems, which uses holons to represent products, resources, orders and logical activities. Gonçalves *et al.* [13] presented an approach based on co-operating agents to the reengineering production facilities. The approach focus on several aspects related to enterprise dynamic reconfiguration due to product redesign or changing demand, and on optimizing the production process or removing errors that might have emerged.

In spite of all the research described above, only a few industrial/laboratorial applications were developed and reported in the literature. Bussmann and Schild [14], as part of the *Production 2000+* project, use agent technology to design a flexible and robust production system for large series manufacturing that meet rapidly changing operations in a factory plant of DaimlerChrysler, producing cylinder heads for four-cylinder diesel engines. This agent-oriented collaborative control system, proved to be useful to control widely distributed and heterogeneous devices in environments that are prone to disruptions and where hard real-time constraints are crucial.

Cooperative Engineering concerns the application of Concurrent Engineering techniques to the design and development of products and of their manufacturing systems by a network of companies coming together exclusively for that purpose. Gonçalves *et al.* [15] presented an implementation of a framework for Cooperative Engineering based on a general framework of distributed hybrid systems and MAS. More examples of agent-based approaches in manufacturing systems can be found in [16]-[18].

## III. Adaptive Smart Manufacturing Systems

The goal of XPRESS was to realize an Intelligent Manufacturing System (IMS) and to establish a breakthrough for the factory of the future, with a new flexible assembly and manufacturing concept based on the generic idea of "specialized intelligent process units" (referred to as *manufactrons* in the context of XPRESS) integrated in cross-sectorial learning networks for customized production and flexible system organization. This knowledge-based concept integrates the complete process hierarchy, from the production planning to the assembly, the quality assurance of the produced/assembled products and the reusability of process units. Different functionalities within a factory are encapsulated in specialized intelligent process units called "Smart Components". By doing so, a single Smart Component is able to perform the assigned tasks optimally within linked networks by considering their knowledge. The mechanisms of self-learning, self-organization, knowledge acquisition (experiments), as well as the use of shared communication opportunities, which are required for performing successfully, are stored in every Smart Component.

### A. Industrial Smart Components

A Smart Component is a self-contained entity, which encapsulates expertise and functionalities, and that interacts with its environment by the exchange of standardized synchronous messages. Being self-contained, it is expected that a typical Smart Component can be included to a smart manufacturing system by just plugging an additional device (into the factory's network). Therefore, the Smart Component has to be realized as an independent component (comprising software and hardware) rather than a distributed set of parts, where a lot of different parts of the component are to be integrated into different systems of the factory – Enterprise Resource Planning (ERP), Manufacturing Execution Systems (MES), or different kinds of Programmable Logic Controller (PLC) systems [19].

The Smart Component shall not only realize a simple functionality, but also provide expertise on this functionality to the outer world. This allows the outer world to state a task to be fulfilled to the Smart Component without the need to know about every small detail associated with the task. The encapsulation of expertise is therefore the solution to demands stated by multi-variant production and flexibility in terms of production resources.

The Smart Component can be seen as an autonomous agent, able to decide the best way to reach its given goals, but not when to do it. The task execution is triggered from outside as defined by a Smart Component from a specific category, named "workflow manager", responsible for overlooking the factory level with dedicated knowledge expertise [20]. This results in a Smart Component hierarchy: "Production Smart Components" (executing basic manufacturing tasks) and "Super Smart Component" (coordinating groups of Production Smart Components); "Workflow managers" (controlling the production flow of an item) conforming the manufacturing execution system up to production planning; "Configuration Smart Components" responsible for finding an optimum production configuration and for the creation of workflow managers for different product variants or for varying production conditions.

### B. Communication

Communication between different systems is a major challenge in industrial environments. Most communication channels are particularly tailored to different systems and are often proprietary. Hence, integration of equipment requires additional engineering and makes it difficult the simple replacement of systems. On the other hand, if standard connections are used, the process slows down in most cases and finally just covers a subset of the necessary functionalities [19]. A generic understandable task description, describing the production tasks to be performed by a particular machine for a certain class of products can be a solution for this problem. The basic approach of the Smart Component communication scheme is a synchronous exchange of documents. For that, only three types of documents exist: Task Description Documents (TDD); Quality Result Documents (QRD); and Smart Component Self Description (SCSD). This approach led to the development of a uniform and standardized communication protocol for the Smart Component framework.

### C. Smart Component Networks

The Smart Components are hierarchized into three categories according to their function: Configuration Smart Components responsible for finding an optimum production configuration and for the creation of a workflow manager template that can be instantiated to produce the product variant; Workflow Manager controls the production flow of an item according to the workflow manager template; Production Smart Components responsible for executing basic manufacturing tasks and/or for coordinating groups of production Smart Components.

A major challenge of the approach is the interaction of the different components of the whole system. The communication scheme between components of the different layers (ERP, shop floor and cell level) and also within the layers must be powerful, flexible and extensible. The concept of Smart Component network comprises the Production Configuration System (PCS), the Workflow Execution System (WES), and the lower level Smart Components: Super Smart Component, Production Smart Component and Handling Smart Component.

The PCS is divided in three components: production simulation system (PSS), production execution system (PES), and finally production quality system (PQS). The PSS performs simulation tasks, using different workflows with various production Smart Components and configurations. On the other hand, the PES is responsible for receiving and selecting the best configuration from production jobs issued by external ordering systems, such as SAP, Baan or MES. Regarding PQS, this component is responsible for storing and retrieving the quality results in XML formatted files denominated quality result documents (QRDs), which are generated at the end of the production cycle and contain the complete quality information of the entire production process and the product itself.

The WES, instantiated by the PCS during the simulation phase or production phase, consists of a workflow manager (WFM) and a quality manager (QM). This component, the WES, is the mediator between the PCS and all the other production Smart Components (PMs), handling Smart Components (HMs) or super Smart Components (SMs). Each started instance of WFM or QM is responsible for the control and organization of the Smart Components related to the process. This allows the WES to suspend or to persist the Smart Components, if no activity is to be performed. It is the responsibility of every Smart Component to communicate with lower or higher level Smart Components (SMs or WES “Smart Component”). As far as the communication goes, it is via the exchange of XML data between the components and the system. The system’s communication is synchronous, therefore, each TDD sent to a Smart Component must result in a QRD. In case that the operation is not performed, a QRD containing an error message must be sent to the upper level.

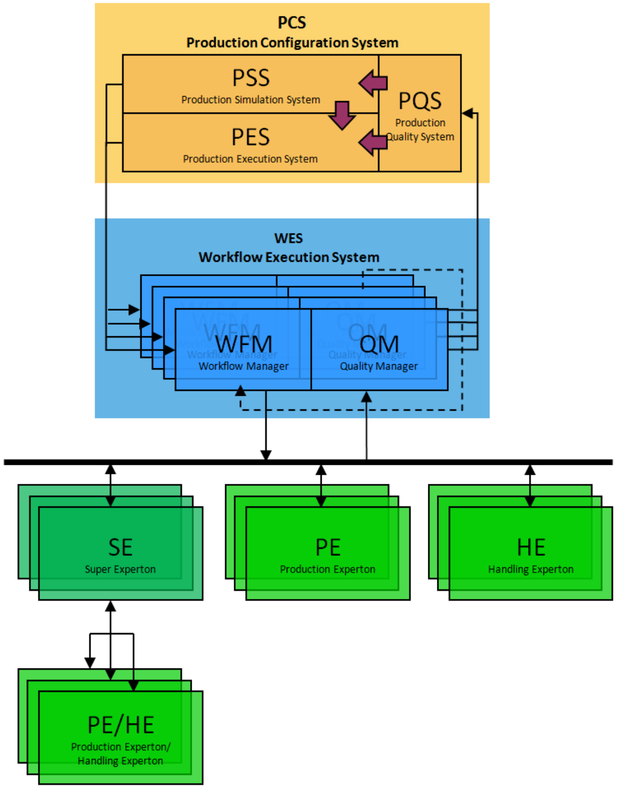


Figure 1 – Smart Component Network

A production system implemented via a Smart Component network, in which several production equipment and therefore Smart Components are considered to execute a process step, the Production Configuration System (PCS) collects the different specifications and generates a TDD. This file can then be understood by all Smart Components that are considered for the process. The structure of MSD and TDD documents is defined in such way that the integration and transformation can take place as easily and unambiguously as possible. An overview of the Smart Component architecture with the communication between layers is given in Figure 1. During production, the Workflow Execution System (WES) sends the TDD to a particular Smart Component (production equipment). Ideally, this happens simultaneously with the loading of the work piece. Due to the fact that it possesses all the necessary information, the Smart Component should now be able to execute the process step successfully. The task description is a high-level document and should not be mistaken for a batch sheet or recipe: in most cases the task description is less extensive but at the same time more flexible than a pure batch sheet specification. At the end of the process step, the product and quality data are returned to the WES simultaneously with the physical unloading of the work piece. The shape of the QRD sent to the WES is also predetermined by the MSD in order to ease the analysis of the resulting quality.

The radical innovations of the “Smart Component Networked Factory” are knowledge and responsibility segregation, trans-sectoral process learning in specialist knowledge networks. The concept is built on coordinated teams of specialized autonomous objects (Smart Components), each knowing how to do a certain process optimally. This architecture allows continuous process improvement, and therefore the system is able to anticipate and to respond to rapidly changing consumer needs, producing high-quality products in adequate quantities while reducing costs.

## IV. MULTI-AGENT IMPLEMENTATION OF THE NETFACTORY

As explained in the aforementioned sections, one of the steps forward on the reconfigurability in networked factories is the encapsulation of the equipment with software, extending it with communication capabilities and intelligent functionalities, such as negotiation. This kind of approach will allow not only the inter-equipment communication and collaboration, but also the communication between the shop-floor equipment and any software component, assuming it is also encapsulated with the same technology. This will leverage a much more flexible and effective way of equipment configuration, paving the way for the Network Factory implementation, and therefore, the shop-floor reconfigurability.

This way, a simple MAS was developed to mimic the pertinent behaviours and interactions between the most important Smart Components, and thus, analyse and predict the problems that might occur in a real industrial environment, at a collaborative and cooperative level. As can be seen in Figure 1, there are three different levels of abstraction present in the Smart Component Network, but only the first and the last ones were considered for the MAS modelling. This selection lies on the fact that only problems on the shop-floor reconfiguration will be analysed, not considering if the production is running well or not (monitoring and controlling), but instead, take into account the negotiation and collaborative abilities to verify if the requirements for fast shop-floor reconfiguration are met, in the presence of a new product variant.

Therefore, Configuration Smart Component and Production Smart Component Agents were developed, and as explained in Section 3, the first one is responsible to find the optimum production configuration according to some product requirements, and the latter one is intended to execute the basic manufacturing tasks. Hence, in terms of information flow, whenever a Production Smart Component Agent enters into the network, it should be able to generate a MSD, and send it to the already existing Configuration Smart Component Agents, so they can know how the shop-floor can be configured using the available equipment and according to some product requirements. The first step towards the production process is related with the information sent to a certain Configuration Smart Component Agent about the product specifications, and the generation of the corresponding TDD to subsequently send it to the available Production Smart Components Agent with the matching capabilities, for shop-floor operation. Furthermore, when the Production Smart Components Agents finish their operation on the production process, the next step is the generation of the QRD that is then sent to the Configuration Smart Component Agent to update and report the information about the equipment's production performance. This quality feedback will drastically influence the selection of the available Production Smart Components in the optimum production configuration, benefiting the equipment with better performances, tending, this way, to choose the most reliable and effective ones.

As previously mentioned, one of the MAS purposes is to study the problems associated with collaborative activities like the ones described earlier, when the Configuration Smart Component Agent delegates TDDs to Production Smart Component Agents to act accordingly, and subsequent feedback to report the process quality by means of QRD. However, most of the collaborative abilities can lead to a conflict situation, mainly when two different entities are trying to establish a partnership with the same third party. In the context of the Network Factory, this can occur when there are several instances of Configuration Smart Components that can include in their optimum production configuration the same Production Smart Component to operate on the shop-floor level, if this search is made concurrently. One of the techniques associated for conflict resolution is the market-based negotiation. This concept can be simply explained as the increase of a resource cost until only one "costumer" is willing to pay for the achieved price. For the implementation of this technique, *Utility*, *Cost* and *Threshold* functions were built to measure the overall usefulness of using a certain Production Smart Component on the production configuration. The first one measures how distant an equipment operation is from the ideal product specification, the second one returns a value of how much an equipment execution can cost (not its actual running cost, but only a measure representative for this problem) based on QRDs information – as much worse the equipment performance is, the higher is the cost associated to it, and the latter one is how much an agent is willing to pay, based on the utility previously calculated – if the utility is high, the threshold value will also be, and vice-versa. Hence, when the same Production Smart Component Agent is the most suitable one for different Configuration Smart Component Agents, the cost of Production Smart Component Agent's execution will be increase, until only one Configuration Smart Component Agent remains with the threshold value above the cost.

## V. Discussion and Future Work

### A. *Results from the multi-agent implementation*

The strategies presented on the previous sections regarding MAS, along with the agent paradigm and well structured communication processes (MSD, TDD and QRD), proved to be an effective and reliable approach, since some of the problems that arise from equipment collaboration were studied and successfully solved using the market-based negotiation approach. The modelled MAS represents a short step forward, but not less important, towards a flexible and extensible production reconfiguration, taking into account the complex industrial dynamics and heterogeneous environments. One of the most important advantages of the MAS characteristics is undoubtedly the decentralized approach that verifies the fault tolerant property, in case of sudden equipment failure. The networked factory will maintain its communication and collaboration activities, avoiding stopping the production process due to component non-dependency issues, minimizing costs and maximizing the network reliability. Another important concept presented in this paper is the task-driven communication, in which equipment execution on shop-floor level are specified in XML-based format, and used to delegate responsibilities for operation according to precise specifications (TDD), and receive a valuable feedback on the equipment quality execution (QRD). Comparing with manual reconfigurability, which in turn reveals to be not cost effective, this concept is an important step forward regarding the automatic reconfiguration of equipment for shop-floor operation.

### B. *Limitations and future extensions of the approach*

The main goal of the work presented in this paper is to provide methods, that can be either fully automated or an aid to the planning engineer, that selects which Smart Components to use for a specific job (new product or variant); this will answer the question: "which is the best configuration for this task?"

From the modules that build the configuration Smart Component, the Production Simulation System (PSS) is the one responsible for the creation of new configurations to answer a specific Job description. The assignment problem is a special type of linear programming problem where resources are being assigned to perform tasks [21]. There is a simple algorithm to efficiently evaluate the solution. This algorithm is known as the Hungarian Method and is able to retrieve the best set of Smart Components for a set of tasks. However, this approach is not helpful in the present context mainly due to the fact that the data made available by the Smart Component (each Smart Component provides a self description document with its typical production capabilities, times and quality levels) does not take into account the impact of working in tandem with other Smart Components. This is the main reason to include a simulation tool on the decision process. To be effective, this tool has to be able to analyse several hundreds of different line configurations. A specific data development analysis model referred to as Charnes, Cooper and Rhodes (CCR) [22] model is a fractional programming technique that evaluates the relative efficiency of homogeneous decision making units, in our case, the relative efficiency of Smart Components. The general efficiency measure, which will be referred as the cross-reference comparison, is presented in (1).

$$\mathrm{E_{ks}} = \frac{\sum_{\mathrm{y}} \mathrm{O_{sy}v_{ky}}}{\sum_{\mathrm{x}} \mathrm{I_{sx}u_{kx}}} \tag{1}$$

where: $\boldsymbol{O_{sy}}$ are the output measures $\boldsymbol{y}$ of the Smart Component $\boldsymbol{s}$; $\boldsymbol{v_{ky}}$ are the weights of the "target" Smart Component $\boldsymbol{k}$ to output $\boldsymbol{y}$; $\boldsymbol{I_{sx}}$ are the input measures $\boldsymbol{x}$ of the Smart Component $\boldsymbol{s}$; $\boldsymbol{u_{kx}}$ are the weights of the "target" Smart Component $\boldsymbol{k}$ to input $\boldsymbol{x}$; $\boldsymbol{E_{ks}}$ is the cross- efficiency of Smart Component $\boldsymbol{s}$, using the weights of "target" Smart Component $\boldsymbol{k}$.

An optimal value $\boldsymbol{E^*_{kk}}$ for the cross-reference comparison is obtained by maximizing (2):

$$\mathrm{E^*_{kk}} = \frac{\sum_{\mathrm{k}} \mathrm{O_{ky}v_{ky}}}{\sum_{\mathrm{k}} \mathrm{I_{kx}u_{kx}}} \tag{2}$$

subjet to:

$$\mathrm{E_{ks}} = \frac{\sum_{\mathrm{y}} \mathrm{O_{sy}v_{ky}}}{\sum_{\mathrm{x}} \mathrm{I_{sx}u_{kx}}} \leq 1 \;\forall\, s$$

$$v_{ky} \geq 0, u_{kx} \geq 0, and \sum_{x} I_{sx}v_{kx} = 1$$

If $\boldsymbol{E^*_{kk}}$ is equal to 1 then there is no other Smart Component which is better than Smart Component $\boldsymbol{k}$ for its optimal weights. Solving this optimization to all the Smart Components, then it is possible to select the ones that are not optimal ($\boldsymbol{E^*_{kk}}$ < 1) and remove them from the solution space. The cross reference comparison leads to Pareto optimal solutions but it is not a sufficient condition.

## VI. Conclusions

The Smart Component network concept meets the challenge to integrate intelligence and flexibility at the "highest" level of the production control system, as well as the "lowest" level of the singular machine, and precludes the shift of the production process from a resource-efficiency perspective towards knowledge-based and customer-driven approach. This networked factory approach allows the implementation of a multi-variant system making it possible to have an adequate number of production lines for the manufacturing of adequate quantities of respective goods using an adequate the number of Smart Components in order to meet the requirements of increasing product variants and producing at ever-smaller lot sizes. Due to the knowledge and responsibility segregation within the system, the various production units are easily extendable and exchangeable and thus offer an unlimited "plug & produce" functionality. Different product variants can be produced with the same assembly units (Smart Components) on the same production line. The new Smart Component concept achieves a high level of reusability of assembly equipment and is fast, flexible, reconfigurable, and modular. New developments of this concept, currently being explored include its adaptation to fast ramp-up and equipment re-use scenarios [25].

## References

[1] M. G. Mehrabi, A. G. Ulsoy, and Y. Koren, "Reconfigurable manufacturing systems: Key to future manufacturing," Journal of Intelligent Manufacturing, vol. 11, pp. 403-419, 2000/08/01 2000.

[2] H. Van Brussel, J. Wyns, P. Valckenaers, L. Bongaerts, and P. Peeters, "Reference architecture for holonic manufacturing systems: PROSA," Computers in Industry, vol. 37, pp. 255-274, 11// 1998.

[3] G. Morel, P. Valckenaers, J.-M. Faure, C. E. Pereira, and C. Diedrich, "Manufacturing plant control challenges and issues," Control Engineering Practice, vol. 15, pp. 1321-1331, 11// 2007.

[4] L. M. Sanchez and R. Nagi, "A review of agile manufacturing systems," International Journal of Production Research, vol. 39, pp. 3561-3600, 2001/01/01 2001.

[5] H. Yang and D. Xue, "Recent research on developing Web-based manufacturing systems: a review," International Journal of Production Research, vol. 41, pp. 3601-3629, 2003/01/01 2003.

[6] W. Shen, Q. Hao, H. Yoon, and D. Norrie, "Applications of agent-based systems in intelligent manufacturing: an update review," Advanced Engineering Informatics, vol. 20, pp. 415-431, 2006.

[7] F. Almeida, P. Dias, G. Gonçalves, M. Peschl, and M. Hoffmeister, "A proposition of a manufactronic network approach for intelligent and flexible manufacturing systems," International Journal of Industrial Engineering Computations, vol. 2, pp. 873-890, 2011.

[8] Y. Koren *et al.*, "Reconfigurable Manufacturing Systems," CIRP Annals - Manufacturing Technology, vol. 48, pp. 527-540, // 1999.

[9] M. G. Mehrabi, A. G. Ulsoy, Y. Koren, and P. Heytler, "Trends and perspectives in flexible and reconfigurable manufacturing systems," Journal of Intelligent Manufacturing, vol. 13, pp. 135-146, 2002/04/01 2002.

[10] N. A. Duffie and R. S. Piper, "Non-hierarchical control of a flexible manufacturing cell," Robotics and Computer-Integrated Manufacturing, vol. 3, pp. 175-179, 1987.

[11] H.V.D. Parunak, "An Architecture for Heuristic Factory Control", Proceedings of the American Control Conference, pp. 548-558, 1986.

[12] H. V. D. Parunak, A. D. Baker, and S. J. Clark, "The AARIA Agent Architecture: From Manufacturing Requirements to Agent-Based System Design," Integrated Computer-Aided Engineering volume 8 (1), pp. 45-58, 1998.

[13] G. Gonçalves, J. Sousa, F. Pereira, P. Dias, and A. Santos, "A framework for e-cooperation business agents: An application to the (re)engineering of production facilities" in: Jagdev, H.S., Wortmann, J.C., Pels, H.J., eds., Collaborative Systems for Production Management (Kluwer Academic Publishers), pp. 189-204, 2002.

[14] S. Bussmann and K. Schild, "An Agent-Based Approach to the Control of Flexible Production Systems," presented at the 8th IEEE International Conference on Emergent Technologies and Factory Automation (ETFA 2001), Antibes Juan-les-pins, France, 2001.

[15] G. Gonçalves, P. Dias, A. Santos, J. Sousa, and F. Pereira, "An implementation of a framework for cooperative engineering," presented at the Proceedings of the 16th IFAC World Congress, Czech Republic, 2005.

[16] H. Dyke Parunak, "What can agents do in industry, and why? An overview of industrially-oriented R&D at CEC," in Cooperative Information Agents II Learning, Mobility and Electronic Commerce for Information Discovery on the Internet. vol. 1435, M. Klusch and G. Weiß, Eds., ed: Springer Berlin Heidelberg, 1998, pp. 1-18.

[17] B. Saint Germain, P. Valckenaers, P. Verstraete, Hadeli, and H. Van Brussel, "A multi-agent supply network control framework," Control Engineering Practice, vol. 15, pp. 1394-1402, 11// 2007.

[18] V. Mařík and J. Lažanský, "Industrial applications of agent technologies," Control Engineering Practice, vol. 15, pp. 1364-1380, 11// 2007.

[19] L. Neto, J. Reis, D. Guimarães, G. Gonçalves, "Sensor cloud: Smartcomponent framework for reconfigurable diagnostics in intelligent manufacturing environments", 2015 IEEE 13th international conference on industrial informatics (INDIN), pp. 1706-1711, 2015.

[20] M. Peschl, N. Link, M. Hoffmeister, G. Gonçalves, and F. Almeida, "Designing and implementation of an intelligent manufacturing system," Journal of Industrial Engineering and Management, vol. 4, pp. 718-745, 2011.

[21] F. S. Hillier and G. J. Lieberman, Introduction to Operations Research, Eighth ed.: McGraw-Hill Primis, 2008.

[22] A. Charnes, W. W. Cooper, and E. Rhodes, "Measuring the efficiency of decision making units," European Journal of Operational Research vol. 6, p. 429.444, 1978.

[23] R Pinto, J. Reis, R. Silva, M. Peschl, G. Gonçalves, “Smart sensing components in advanced manufacturing systems”, International Journal on Advances in Intelligent Systems 9 (1&2), 181-193, 2016.

[24] L Neto, J Reis, R Silva, G Gonçalves, “Sensor SelComp, a smart component for the industrial sensor cloud of the future”, 2017 IEEE International Conference on Industrial Technology (ICIT), 1256-1261, 2017.

[25] S. Aguiar, R. Pinto, J. Reis, G Gonçalves, “A Marketplace for Cyber-Physical Production Systems: Architecture and Key Enablers”, INTELLI 2017, The Sixth International Conference on Intelligent Systems and Applications, Nice, France, 2017.

[26] M. Peschl, “An Architecture for Flexible for Flexible Manufacturing Systems based on task-driven Intelligent Agents”, Ph.D. dissertation, Uniiversiitatiis Ouluensiis, Finland, 2014.

# A Software Architecture for Transport in a Production Grid

Leo van Moergestel, Erik Puik
Institute for ICT
HU Utrecht University of Applied Sciences
Utrecht, the Netherlands
Email: leo.vanmoergestel@hu.nl, erik.puik@hu.nl

John-Jules Meyer
Intelligent systems group
Utrecht University
Utrecht, the Netherlands
Alan Turing Institute Almere, The Netherlands
Email: J.J.C.Meyer@uu.nl

***Abstract*—In this paper, a software architecture is proposed to implement transport of products being made along production units. In the classical approach, production lines are used where products all follow a similar linear path during production. New production methods require a more agile and flexible path to meet the requirements of different paths to be followed during production to enable the productions of products with different user requirements, as well as a more fault tolerant production system. Starting with results from simulation, the requirements of the software architecture are established. The architecture proposed is inspired by the architecture used in software defined networks, that play a mayor role in complex computer networks.**

***Keywords*–Agent technology; Agile manufacturing; Production software architecture.**

## I. Introduction

Today, information technology plays a major role in manufacturing as well as in other aspects of our modern society. In manufacturing, the trend is towards low-cost agile manufacturing of small batch sizes or even one product according to end-user requirements. When looking at the industrial revolutions, the first revolution was the use of steam power to facilitate production. The second revolution was the introduction of production lines based on the use of electrical energy. This resulted in economic and feasible mass production. Computer technology resulted in the third revolution, where many production tasks were automated by the use of Programmable Logic Controllers (PLCs), Distributed Control Systems (DCS) and robots. The latest revolution is the integration of information technology in the production process as a whole. This has been described by the term industry 4.0 [1] or cyber physical systems [2].

One of the ideas behind the concept is production on demand according to end-user requirements. To accomplish this, new production paradigms should be developed. One of the requirements of these new paradigms is the search for alternatives for the so-called production lines, where mass-production is realised by a linear sequence of production units or cells. Every unit offers a single production step in the sequence of steps needed to realise the final product.

In our research group, a set of cheap reconfigurable production machines called equiplets has been proposed as the production platforms that should be combined with a flexible transport system between these equiplets. This resulted in the concepts of a grid of these equiplets that should be capable to produce a variety of different products in parallel [3]. The concept fits in the concepts of Industry 4.0 or cyber physical systems. This paper will focus on the architecture to be used to implement a flexible transport of products during production. Though based on our concept of grid production using equiplets, this model can also be used in situations where a flexible transport between production units or cells is needed. The concept of grid production presented here, does not focus on a specific industry, but should be considered as a generic production concept.

The rest of this paper is organised as follows. Section II is dedicated to related work. Section III discusses the production model in more detail. In Section IV, the simulation model and implementation is presented, followed by Section V showing some results of the simulation. Section VI discusses the architecture that can be used to implement the real transport system and a conclusion will end the paper.

## II. Related work

In this section, an overview will be given on agent-based manufacturing. Especially the planning part will be given attention. Important work in the field of agent-based manufacturing has already been done. Paolucci and Sacile [12] give an extensive overview of what has been done. Their work focuses on simulation as well as production scheduling and control [13]. The main purpose to use agents in [12] is agile production and making complex production tasks possible by using a multiagent system. Agents are also proposed to deliver a flexible and scalable alternative for manufacturing execution systems (MES) [14] for small production companies. The roles of the agents in this overview are quite diverse. In simulations agents play the role of active entities in the production. In production scheduling and control agents support or replace human operators. Agent technology is used in parts or sub-systems of the manufacturing process. The planning is mostly based on the type of planning that is used in MES. This type of planning is normally based on batch production. We based the manufacturing process as a whole on agent technology. In our case, a co-design of hardware and software was the basis. The planning will be done on a single product basis and not on batch production.

Bussmann and Jennings [15][16] used an approach that compares in some aspects to our approach. The system they describe introduced three types of agents, a workpiece agent, a machine agent and a switch agent. Some characteristics of their solution are:

- The production system is a production line that is built for a certain product. This design is based on redundant production machinery and focuses on production availability and a minimum of downtime in the production process. Our system is a grid and is capable to produce many different products in parallel;
- The roles of the agents in this approach are different from our approach. The workpiece agent sends an

ISBN: 978-1-61208-646-0

invitation to bid for its current task to all machine agents. The machine agents issue bids to the workpiece agent. The workpiece agent chooses the best bid or tries again. This is what is known as the contract net protocol. In our system the negotiating is between the product agents, thus not disrupting the machine agents;

- They use a special infrastructure for the logistic subsystem, controlled by so-called switch agents. Even though the practical implementation is akin to their solution, in our solution the service offered by the logistic subsystems can be considered as production steps offered by an equiplet and should be based on a more flexible transport mechanism.

So, there are important differences between the approach of Bussmann and our approach. The solution presented by Bussmann and Jennings has the characteristics of a production pipeline and is very useful as such, however it is not meant to be an agile multi-parallel production system as presented here. Their system uses redundancy to overcome the problem that arises in pipeline-based production when one of the production systems fails or becomes unavailable. The planning is based on batch processing.

Other authors focus on using agent technology as a solution to a specific problem in a production environment. In [17], a multi-agent monitoring is presented. This work focusses on monitoring a manufacturing plant. The approach we use monitors the production of every single product. The work of Xiang and Lee [18] presents a scheduling multiagent-based solution using swarm intelligence. This work uses negotiating between job-agents and machine-agents for equal distribution of tasks among machines. The implementation and a simulation of the performance is discussed. We did not focus on a specific part of the production but we developed a complete production paradigm based on agent technology in combination with a production grid. This model is based on two types of agents and focuses on agile multiparallel production. The role of the product agent is much more important than in the other agent-based solutions discussed here. In our model, the product agent can also play an important role in the life-cycle of the product [19]. The design and implementation of the production platforms and the idea to build a production grid can be found in Puik [20].

## III. Production model

Industry 4.0 is also characterised as a cyber physical system. In this section, these two parts will be explained starting with the physical aspect.

### A. Physical aspect

As stated in the introduction, the actual production is done by so-called equiplets. An equiplet is a reconfigurable production machine [4]. Every equiplet is capable to perform one or more production steps. A definition of a production step is: *A production step is an action or group of coordinated or coherent actions on a product, to bring the product a step further to its final realisation. The states of the product before and after the step are stable, meaning that the time it takes to start the next step can be short or long for the production as a process (not for the production time) and that the product thus can be transported or temporarily stored between two steps.* A sequence of production steps should be performed to create a product. To accomplish this, a set of equiplets should be used in a certain order. This is done by moving platforms that can transport components, as well as the product itself from equiplet to equiplet. In Figure 1, this setup is shown. The arrows show a global path a product has to follow [5]. The equiplets are placed in a grid. The transport platform is first loaded with components needed for the production and will enter the grid where the components are handled by the equiplets to create a product (or product part or a product that is used as a component for the final product, a so-called half product). When this is done, the product will be finished or in case of product parts or half products, the grid can be re-entered to handle different product parts to make the final product. Different products need specific production steps in their own perhaps different order. The transport between the equiplets for a certain product will look like the path depicted by arrows in Figure 2. This particular path is actually a production line for that specific product mapped on the grid. The strength and versatility of the system is that every product can have its own path in the grid, resulting in a unique tailor made product. Complex products consisting of a set of half products can be built using the same principle. In that case, multiple paths should be followed to create the half products and the grid should be re-entered.

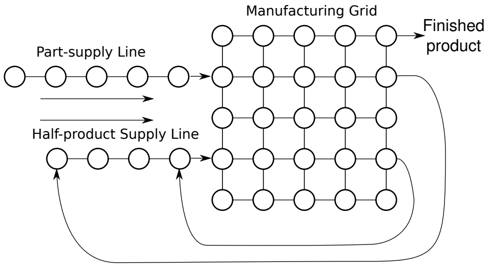


Figure 1. Grid production setup

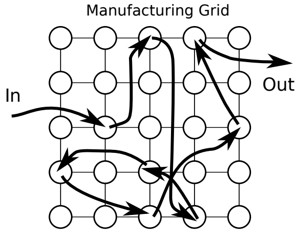


Figure 2. Path for a certain product

### B. Cyber aspect

The software entities that control the production are software agents [6]. An equiplet is represented by an equiplet agent and every product to be made is represented by its own product agent. The robot platforms or moving production platforms are part of the transport infrastructure. The architecture of this infrastructure is discussed in Section VI.

As a start to create a product, a product agent is generated. This agent knows what production steps should be taken and the components to be used. The product agents allocates a transport system and collects the components. Next, it will pass along the equiplets required to perform the production steps. The product agent can discover the equiplets needed by looking at a blackboard system where the equiplets have published the production steps they are cappable to perform. Before an equiplet is chosen, the product agent will first investigate if the equiplet is really capable to perform the specific production step needed, given the parameters involved. To do so the equiplet will run a precise simulation of the step with the parameters given to discover the possibility and the time needed to bring the production step to an end. It will then inform the product agent about success or failure. When the actual real production step is performed, the product agent will also be informed about success or failure, but also the production parameters that had been used. This might be the exact temperature, or the amount and type of adhesive used, etc. Finally, the product agent has a complete production log of the product it represents.

In our model, the creation of a product agent can be done by using a webinterface where the end-user can specify his/her product to be made [7].

## IV. Simulation model and implementation

The simulation model presented in this paper opens the possibility to explore the behaviour of the production system as a whole, taking into account, the transport as well as the time to perform a production step. The model is based on the production model described in the previous section. This means that at random times an agent enters the grid with a list of steps to perform, resulting in a list of equiplets to be visited. This situation is comparable to a group of people shopping in a shopping center, where they need to buy items available in different shops. Everybody is doing this autonomously and according to their own specific shopping list.

To make the simulation versatile, a decision was made to use a graph approach for the description of the grid. The advantage is that all kinds of interconnected nodes can be simulated including a grid so this approach is more powerful and can also be used in a grid where some of the interconnections are obstructed or impossible to use.

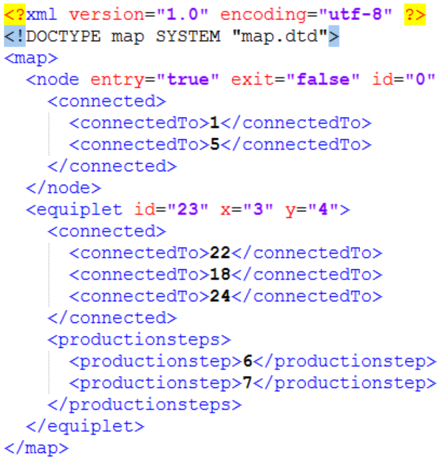
The simulation is driven by three different information files. These file are XML-files so human- and computer-readable.

1) the file maps.xml describes the structure of the grid, actually the structure of the graph;
2) the steps needed for a certain product;
3) the products to be made.

In Figure 3 an example of a map is shown. A map consists of nodes and equiplets, where an equiplet is actually a node offering production steps. Both nodes and equiplets have an unique id, an x-coordinate and y coordinate. A node can also be an entry point and/or exit point of the grid. Equiplets have a set of at least one production step. This way, all kind of production infrastructures fitting in our production model, can be expressed.

The simulation is controlled by a central clock. The simulation is not a realtime simulation, but by using this clock as the central heartbeat, a lot of concurrency problems could be prevented.

A path finding solution is in case of this particular simulation one of the challenges. The production system is based on autonomous entities, actually the product agents, that share the production grid, each having a specific goal, and each making the product it represents. The way this goal should be accomplished should fit in the common goal of the system, a versatile agile production system. The path finding solution used was based on a special map that was generated, telling for every node how far the distance to a certain production step (equiplet) was. A moving platform would choose a direction towards the production step node. If this was not possible it would choose a node having the next lowest distance to the production step node. The reason for making this choice can be found in [8].



```
<?xml version="1.0" encoding="utf-8" ?>
<!DOCTYPE map SYSTEM "map.dtd">
<map>
  <node entry="true" exit="false" id="0" x="0" y="0">
    <connected>
      <connectedTo>1</connectedTo>
      <connectedTo>5</connectedTo>
    </connected>
  </node>
  <equiplet id="23" x="3" y="4">
    <connected>
      <connectedTo>22</connectedTo>
      <connectedTo>18</connectedTo>
      <connectedTo>24</connectedTo>
    </connected>
    <productionsteps>
      <productionstep>6</productionstep>
      <productionstep>7</productionstep>
    </productionsteps>
  </equiplet>
</map>
```

Figure 3. XML content describing the grid

The simulation has been implemented as two components. First, there is the core system that actually performs the simulation. The second component is a graphical user interface (GUI) that will show in detail the working of the production system. It is possible to use the core system without the GUI if a lot of simulation runs should be made to generate data that can be studied afterwards.

Java has been used as the language for implementation. It is not considered to be the fastest language, but it fits well in modern software engineering concepts. The fact that many multiagent platform implementations are also based on Java was a second reason to use this language, because this simulation can also become part of the production software that is actually a multiagent system based on Jade. Jade is a Java-based multiagent programming environment.

## V. Some simulation results

The implementation resulted in a simulation system that can be used with or without a GUI. The grid consists of nodes that are connected in a certain way. It is actually a graph as mentioned earlier. The edges of the graph can be unidirectional or bidirectional. A node can host an equiplet, but also be empty.

The first result that will be shown is the behaviour of the grid under different loads. By load is meant:

$$LOAD_{Grid} = \frac{Numberofproductsinthegrid}{Numberofnodes} \times 100\%$$

In Figure 4, a grid is shown that is not fully connected. The grid has five equiplets (denoted by the extra square connected to the node), one entry-point at the top left corner and two exit points at the right side (top and bottom node of the group of three nodes). We used this grid to simulate the production of 20 products. In this case we have four times a similar product, thus making one specific product five times. The test was run with several different loads of the grid. The results of the test are shown in Table I. The top row shows the load and the other numbers are the time ticks for a product to complete. When the production is not possible due to deadlock in the overcrowded grid, this is denoted by $DEAD$.

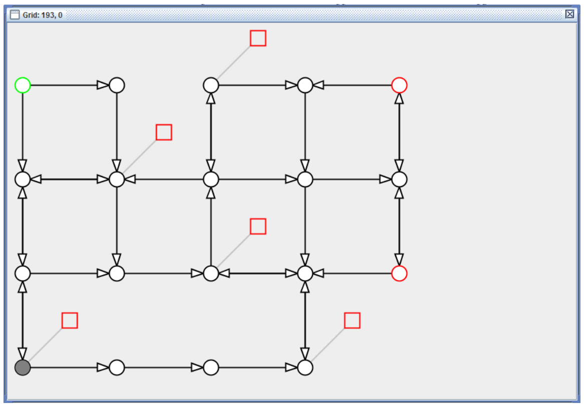


Figure 4. Example grid setup

TABLE I. INCREASING GRID LOAD

| 10 | 25 | 50 | 75 | 90 | 100 |
|---|---|---|---|---|---|
| 114 | 130 | 128 | 143 | DEAD | DEAD |
| 114 | 130 | 161 | 256 | DEAD | DEAD |
| 114 | 168 | 202 | 398 | DEAD | DEAD |
| 114 | 168 | 237 | 404 | DEAD | DEAD |
| 113 | 174 | 279 | 446 | DEAD | DEAD |
| 120 | 174 | 296 | 664 | DEAD | DEAD |
| 121 | 169 | 317 | 672 | DEAD | DEAD |
| 121 | 200 | 313 | 683 | DEAD | DEAD |
| 121 | 167 | 367 | 681 | DEAD | DEAD |
| 120 | 174 | 284 | 723 | DEAD | DEAD |
| 143 | 192 | 347 | 717 | DEAD | DEAD |
| 143 | 208 | 328 | 777 | DEAD | DEAD |
| 143 | 183 | 302 | 775 | DEAD | DEAD |
| 142 | 181 | 376 | 771 | DEAD | DEAD |
| 143 | 184 | 357 | 712 | DEAD | DEAD |
| 190 | 213 | 362 | 595 | DEAD | DEAD |
| 190 | 200 | 370 | 488 | DEAD | DEAD |
| 190 | 215 | 315 | 481 | DEAD | DEAD |
| 190 | 232 | 335 | 472 | DEAD | DEAD |
| 189 | 256 | 306 | 251 | DEAD | DEAD |

Table II is partly generated from Table I and shows the average production time for all products under a certain load. The load is shown in the first row, the average production time in the second row. The last row shows the total production time for all products. This is actually the total time of the simulation.

TABLE II. CALCULATED VALUES FROM THE SIMULATION

| 10 | 25 | 50 | 75 | 90 | 100 |
|---|---|---|---|---|---|
| 149 | 196 | 315 | 585 | 0 | 0 |
| 2879 | 1064 | 824 | 936 | 0 | 0 |

The data from the second row (average production time) in Table II are plotted in Figure 5 and Figure 6 shows the total production time (the last row in Table II). As might be expected, the average time increases when the grid is working under a heavy load. A load of 75% is still feasible. The total production time of all products will at first decrease, because a higher load means also more parallelism in the production. However, the total production time for a 75% load is higher than the time for a 50% load. This is due to the crowded grid traffic and the availability of equiplets that are working under a heavy load in the 75% load situation.

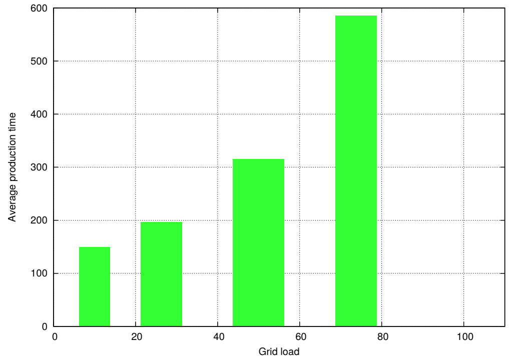


Figure 5. Average time for all products

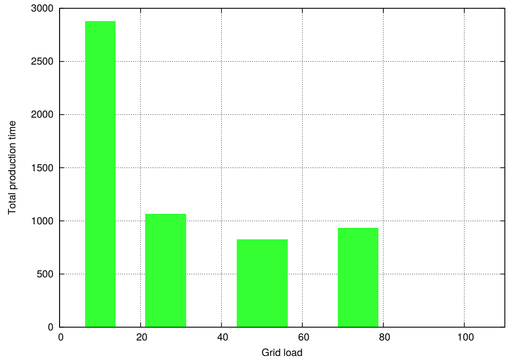


Figure 6. Total time for all products

By exporting the data to an open standard spreadsheet format, spreadsheet tools can be used to generate graphs or calculate additional data. An example of a graph is shown in Figure 7. A nice way to show the busiest node in a certain simulation. A third result shows the effect of making a modification in the path finding method. Observing the simulation, it turned out that a production platform was moving around an equiplet while the equiplet was busy with another product. If another equiplet with the same production step was available, it would be better to head for that equiplet. This was implemented and the results are shown in Table III. Three types of grids are used. They have the same paths and number of nodes, however, type A uses unidirectional paths, type B unidirectional vertical paths and bidirectional horizontal paths, while type C is using bidirectional paths. The static approach shows the time for the situation where moving to an alternative equiplet is not supported, while dynamic supports this option. In the last column the percentage of decrease in production time is shown.

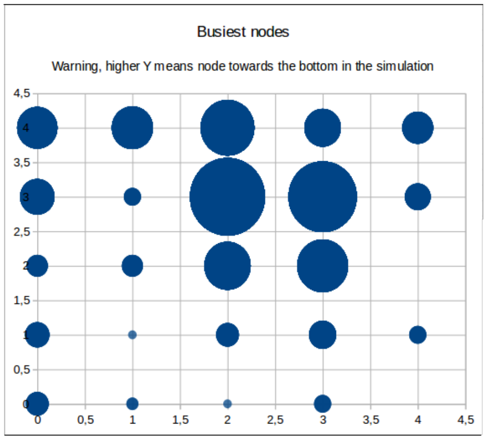


Figure 7. Simulation results showing busiest nodes

TABLE III. CALCULATED VALUES FROM THE SIMULATION

| Type | Load | Static | Dynamic | Diff. |
|---|---|---|---|---|
| A | 25% | 5473 | 4302 | -26% |
| A | 50% | 4425 | 3542 | -20% |
| A | 75% | 4237 | 3604 | -15% |
| B | 25% | 5023 | 3288 | -35% |
| B | 50% | 3676 | 2933 | -20% |
| B | 75% | 2550 | 2942 | -17% |
| C | 25% | 4796 | 3290 | -31% |
| C | 50% | 3374 | 2929 | -13% |
| C | 75% | 3297 | 2938 | -11% |

Another important result is that our simulation proves that the path finding approach works well in our production system. However, under heavy loads the system will block as shown in Table I. This means that a way finding an architecture for implementing this in the production multiagent system is the next challenge. The next section will discuss this issue.

## VI. From Simulation to an Implementation Architecture

The simulation was a tool to study the transport system and might play a role in the implementation architecture. In real life, the following situations should be taken care of:

- it is a concurrent system, so there should be a solution for the concurrency problem.
- a moving production platform could fail and block a path in the graph;
- a production step might take longer or shorter than predicted;
- a production unit might fail or become unavailable.

This means that the graph containing the paths for the transport robot will change in time and that the system should be prepared for the unexpected. In our first architecture proposal, only the first item mentioned is covered.

### A. *Pure autonomous agents based architecture*

A way to mimic the situation of the simulation and thus overcoming concurrency problems might be a token passing system where the transport is based on timeslots (comparable with the clock ticks in the simulation). A timeslot is the time needed to reach a nearby node in the grid. An overview or list of active product agents should be available. The situation of the grid $G$ at the beginning of the timeslot having $N$ active product agents can be described by:

$$G(p1(t), p2(t)...pN(t)$$

Where $p1(t)$ is the position of the product agent $p1$ at time $t$, $p2(t)$ the position at time $t$ of product agent $p2$ and so on. At the start the token is given to agent $p1$ that calculates its path according to the weighted path algorithm described in this paper. This will generate a new state for the grid, given by:

$$G(p1(t+1), p2(t)...pN(t)$$

Now the token is passed to agent $p2$ that will calculate its path based on this new state and so on until all $N$ agents have a path and the new grid state will be:

$$G(p1(t+1), p2(t+1)...pN(t+1)$$

This concept can be implemented in a multiagent system by sharing the state of the grid $G$ on a blackboard. Every agent is only interested in a small part of this information and will update only the state of two nodes, the node that becomes free and the node it will occupy at $t+1$. The overhead of communication in the distributed system will be small. There are also some disadvantages involved. There are no concepts included to overcome some of the situations mentioned in the beginning of this section. A token passing system is also vulnerable to loss of token, resulting in the whole system failing. There are solutions to this problem, like letting the token passing agents check the agent it will send the token to and a token timer to check if the token passing continues, but this requires extra overhead and complexity of the system. So an alternative solution should be investigated.

### B. *Logically centralized control*

The concepts of autonomous agents seems to fit in the pure academic view on multiagent systems, but in our situation it might also be a pitfall as described in Wooldridge [9]. A central control of the transport system might be a suitable concept, but central control could become a single point of failure. A proper solution was found in the latest developments in the realisation of complex computer networks and is known by the term *Software Defined Networks* (SDN) [10].

*1) Concepts used in Software Defined Networks:* The concepts that are used in our proposal are inspired by the concepts of software defined networks. This paragraph will explain in a nutshell these concepts that have to do with the network infrastructure used in complex computer networks as used by Internet Service Providers and Content Providers. The core of the network is based on routers that receive packets from source hosts and forward it to other routers to deliver it to the destination hosts. This is the situation as shown in Figure 8. In the classic situation, all routers had the capability to compute the output the received packet should be forwarded to, based on the destination address in the packet and the information in the routing table. The routing table is built by the cooperation of routers, sending information to each other by a routing protocol like Open Shortest Path First (OSPF) or Border Gateway Protocol (BGP). The actual situation is that a router consists of two parts: a part that forwards packets from certain inputs to certain outputs and a part that is responsible for maintaining and building the routing tables based on information received from other routers. One failing router can spoil the system

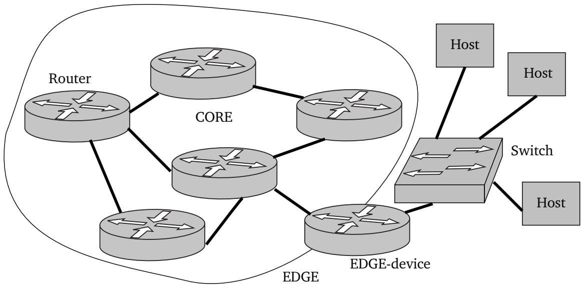


Figure 8. network core with routers

by corrupting the routing tables of other routers. All routers are also involved with two separate tasks being complex path finding algorithms as well as fast forwarding.

In the software defined network approach, routers are not involved with setting up the routing tables. They receive these tables from a server that computes the routes for them. In the SDN approach a router plays its primary role by forwarding packets based on a table containing pattern action combinations. This routing table (actually called a flow table in SDN) is not built by the router itself but by a logically centralised system that functions as the network operating system. Normally, the system is called the SDN controller. This logically centralised system can receive event messages from the routers (like the status and speed of the links it is connected to) and send messages to the routers. On the other hand as shown in Figure 9, it can also communicate with other servers to implement things like access control, routing computation and so on [11].

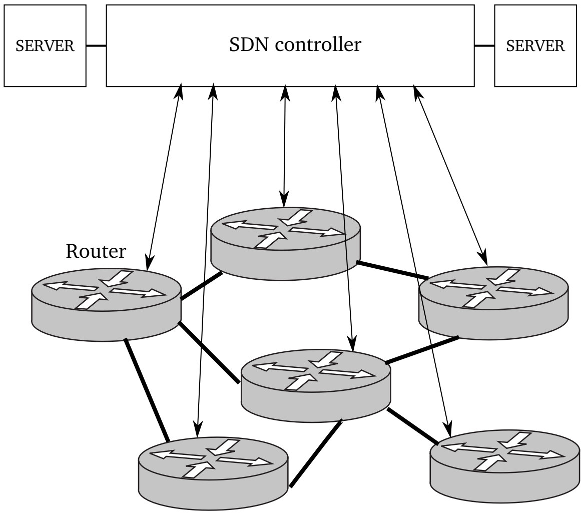


Figure 9. network with SDN controller

So summarized, one can say that there is an ”unbundling” of network functionality. The result of this unbundling is that the routers are less complicated as a system and that the behaviour of the network can be easily controlled and changed by the software in the SDN controller and its related servers. Though the name suggests that this SDN controller is a single server, in practice, to prevent a single point of failure, it will be implemented as a distributed server with fail-over and high availability capabilities.

*2) Using the SDN concept:* The lesson learned from the previous paragraph is that it might be a good solution to simplify the agent controlled robot platform and to introduce a traffic control agent or system that is logically centralized like the SDN controller. Of course a moving robot platform is not a router, but there are also similarities. A moving production platform is in the field and like a router can explore its direct neighbourhood. This information can be sent to the traffic agent, that can update its view on the production grid as a whole and inform other platforms if they need this information to reach their next destination. The advantages of this approach are quite similar to the advantages of software defined networking being:

- All information about the traffic in the grid is available at a central place.
- Easy maintenance and control is possible. If a change in path planning is needed, only the traffic agent is involved.
- Simplification of the software on board of the moving platforms.
- No direct communication between the moving platforms.
- Computing power to solve the routing is not needed on board of every moving platform, but can be done by a special server of set of servers.

Considering the fact that the approach of a multiagent-based system is still adequate, the roles of the agents and their communication should be specified. The traffic agent knows the status of the grid. That means, the available paths, the position of the equiplets and the status of the equiplets as well as the status of all moving production platforms. Based on this information, it will guide every production platform to its next destination. The knowledge about the status of the grid is kept up-to-date by information received from the field where the moving platforms and the equiplets live. The situation is depicted in Figure 10. The traffic agent will not plan the whole path for the production, but only the path between two production steps. This is done to prevent a roll-back of plans already made, if a production step takes longer or shorter than expected. The step by step planning is also used in the simulation described before. The transport agent is the software entity that lives in the production platform and its goal is to bring the product from equiplet to equiplet according to the production steps needed. To meet its goal this agent needs to know the position of the equiplets in the grid and a path to reach the next equiplet in the set of equiplets to be visited. This information will be received from the traffic agent. Events that will generate a message from transport agent to the traffic agent are:

- entering the grid at a certain entering node
- change of edge in the grid
- starting a production step at a certain equiplet
- completion of a production step
- failure of the platform
- failure to enter a certain path (an edge in the graph) because of an obstacle

Summarized: the product agent has the list of steps and will build a list of equiplets to be visited. The product agent

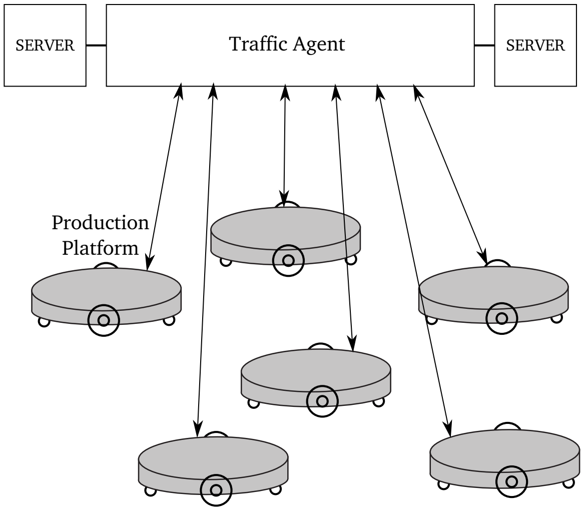


Figure 10. Traffic agent and transport platforms

hands over the list to the traffic agent. The traffic agent will allocate a production platform and guide it along the equiplets to be visited. To accomplish this, the traffic agent will tell the platform (i.e. the traffic agent in the platform) where to move to, while the platform itself is informing the traffic agent about the actual situation of its neighbourhood in the grid.

## VII. Conclusion

An important conclusion from the simulation was, that a change in path finding could result in a significant improvement of the working of the production grid. In practice with autonomous path finding software in all platforms, this would mean that all software in the production platforms should be replaced. From SDN was learned that the moving platforms could contain a simpler type of software and the path finding could be done remotely by a traffic agent and sent to the platform. The platforms could send significant information to the traffic agent. This way, the traffic agent has an accurate view on the status of the grid at a certain moment and can use this status to generate paths for the moving platforms in the grid. The simulation system developed so far can be used to implement the path finding in the traffic agent controlled production system. In that case, calculations for different production approaches can be simulated resulting in the selection of a path planning possibility with the best result for the production as a whole.

Future work will be to implement the architecture as proposed in this paper.

## References

[1] M. Brettel, N. Friederichsen, M. Keller, and M. Rosenberg, “How virtualization, decentralization and network building change the manufacturing landscape: An industry 4.0 perspective,” International Journal of Mechanical, Aerospace, Industrial, Mechatronic and Manufacturing Engineering, vol. 8, no. 1, 2014, pp. 37–44.

[2] R. Rajkumar, I. Lee, Insup, S. L., and J. Stankovic, “Cyber-physical systems: The next computing revolution,” Proceedings of the 47th Design Automation Conference (DAC), Anaheim, California, 2010, pp. 731–736.

[3] L. v. Moergestel, J.-J. Meyer, E. Puik, and D. Telgen, “Decentralized autonomous-agent-based infrastructure for agile multiparallel manufacturing,” Proceedings of the International Symposium on Autonomous Distributed Systems (ISADS 2011) Kobe, Japan, 2011, pp. 281–288.

[4] Z. M. Bi, S. Y. T. Lang, W. Shen, and L. Wang, “Reconfigurable manufacturing systems: the state of the art,” International Journal of Production Research, vol. 46, no. 4, 2008, pp. 599–620.

[5] L. v. Moergestel et al., “A simulation model for transport in a grid-based manufacturing system,” The Third International Conference on Intelligent Systems and Applications (Intelli 2014), Seville, Spain, 2014, pp. 1–7.

[6] M. Wooldridge, An Introduction to MultiAgent Systems, Second Edition. Sussex, UK: Wiley, 2009.

[7] L. v. Moergestel, J.-J. Meyer, E. Puik, and D. Telgen, “Implementation of manufacturing as a service: A pull-driven agent-based manufacturing grid,” Proceedings of the 11th International Conference on ICT in Education, Research and Industrial Applications (ICTERI 2015), Lviv, Ukraine, 2015, pp. 172–187.

[8] L. v. Moergestel et al., “A multiagent-based agile work distribution system,” Proceedings of the Intelligent Agent Technology (IAT 2013), 2013, pp. 293–298.

[9] M. Wooldridge and N. N. R. Jennings, “Software engineering with agents: Pitfalls and pratfalls,” IEEE Internet Computing, May/June 1999, pp.175–196.

[10] J. Kurose and K. Ross, Computer Networking, A Top-Down Approach, 7th ed., ISBN 978-1-292-15359-9. Pearson, 2017.

[11] D. e. Kreutz, “Software-defined networking: A comprehensive survey,” Proceedings of the IEEE, vol. 103, no. 1, 2015, pp. 14–76.

[12] M. Paolucci and R. Sacile, Agent-based manufacturing and control systems : new agile manufacturing solutions for achieving peak performance. Boca Raton, Fla.: CRC Press, 2005.

[13] E. Montaldo, R. Sacile, M. Coccoli, M. Paolucci, and A. Boccalatte, “Agent-based enhanced workflow in manufacturing information systems: the makeit approach,” J. Computing Inf. Technol., vol. 10, no. 4, 2002, pp. 303–316.

[14] J. Kletti, Manufacturing Execution System - MES. Berlin Heidelberg: Springer-Verlag, 2007.

[15] S. Bussmann, N. Jennings, and M. Wooldridge, Multiagent Systems for Manufacturing Control. Berlin Heidelberg: Springer-Verlag, 2004.

[16] N. Jennings and S. Bussmann, “Agent-based control system,” IEEE Control Systems Magazine, vol. 23, no. 3, 2003, pp. 61–74.

[17] D. Ouelhadj, C. Hanachi, and B. Bouzouia, “Multi-agent architecture for distributed monitoring in flexible manufacturing systems (fms),” ICRA 2000 proceedings, 2000, pp. 2416–2421.

[18] W. Xiang and H. Lee, “Ant colony intelligence in multi-agent dynamic manafacturing scheduling,” Engineering Applications of Artificial Intelligence, vol. 16, no. 4, 2008, pp. 335–348.

[19] L. v. Moergestel, J.-J. Meyer, E. Puik, and D. Telgen, “Embedded autonomous agents in products supporting repair and recycling,” Proceedings of the International Symposium on Autonomous Distributed Systems (ISADS 2013) Mexico City, 2013, pp. 67–74.

[20] E. Puik and L. v. Moergestel, “Agile multi-parallel micro manufacturing using a grid of equiplets,” Proceedings of the International Precision Assembly Seminar (IPAS 2010), 2010, pp. 271–282.