

The Improved GM PHD algorithm for Multi-Target Radar Tracking

Zvonko Radosavljević, Dejan Ivković and Branko Kovačević

Abstract—Gaussian mixture probability hypothesis density (GM PHD) is a modern nonlinear algorithm for tracking multiple targets in a clutter environment. It is accompanied by known problems that are primarily related to the impossibility of associating the measurement of existing targets and determining the quality of the tracks. For this purpose the automatic track initialization by known ‘two point initialization’ was introduced. Difference of successive measurements from two radar antenna revolutions is compared with the threshold, which depends on the velocity of the target. The paper proposes to improve the algorithm by introducing the probability of the existence of a target and to reject false tracks. The results of intensive simulations of tracking multiple radar targets have shown the justification for the application of the proposed algorithms.

Index Terms—Target tracking, probability hypothesis density, track while scan (TWS) radars.

I. INTRODUCTION

Multi-target tracking in clutter, assuming linear target trajectory propagation and linear target measurement equation, naturally leads to a Gaussian mixture (GM) target tracking solution. As the origin of measurements is uncertain, both true tracks (which follow targets) and false tracks (which do not) exist [1].

The random finite sets (RFS) are representations of multi-target states and multi-target measurements. The modeling of multi-target dynamics using random sets leads to algorithms which incorporate track initiation and termination, a procedure that has mostly been performed separately in tracking algorithms. The first systematic treatment of multi-sensor multi-target filtering, as part of a unified framework for data fusion using random set theory was finite set statistics (FISST). The alternative to optimal multi-target filtering is the Probability Hypothesis Density (PHD) filter [2], [3], [4]. It is a recursion propagating the 1st moment, called the intensity function or PHD, associated with the multi-target posterior.

A Gaussian mixture, consisting of a weighted sum of Gaussian PDF, each with different means and covariance's, is the natural form of the PDF of target state. Using such a structure, a mixture component is created for every possible

association, using every possible pairing of targets and measurements with the mean and covariance calculated assuming that the particular hypothesis is true, and the weight calculated to represent the probability that the particular hypothesis is true. The serious problem in multi-target tracking is the unknown association of measurements with appropriate targets [5], [6]. Moreover, the data association problem makes up the growth of the computational load in multi target tracking algorithms. Recently, multi-target tracking formulations involve explicit associations between measurements and targets

The Gaussian Mixture Probability Hypothesis Density Filter (GM-PHD Filter) provided a closed form solution to the PHD filter recursion for multiple target tracking [7]. The posterior intensity function is estimated by a sum of weighted Gaussian components whose means, weights and covariances can be propagated analytically in time. In particular, the means and covariances are propagated by the Kalman filter.

The original Gaussian Mixture PHD filter algorithm provided a means of estimating the number of targets and their states at each point in time. The method for determining the targets simply used the weights of the Gaussian components and did not take into account tempo al continuity. We show that if a target is not detected for each iteration, the Gaussian components can still track the targets in the presence of some missed detections. The trajectory of the target in the past, before it has been detected, can also be determined by keeping the trajectories of each of the Gaussian components. The original formulation of the GM PHD filter allowed targets to be spawned from existing targets [8].

The paper proposes and tests the improved GMPHD algorithm with automatic track initialization (ATI) via 'two point methodology'. Each incoming measurement (from the previous scan) is paired with each incoming measurement from the current scan in order to examine the possibility of initializing a new trace based on the knowledge of the maximum speed of movement of the targets [9].

Rest of papers is organized as follows. Second chapter is dedicated basic problem statements, which precedes Third chapter, which brief decrypted GM PHD algorithm. Results of experiments are given by Fourth chapter. Final conclusions are given by the Fifth chapter.

II. PROBLEM STATEMENTS

Consider the target tracking scenario with two dimensions. Also, consider the tracking algorithm with two parameters:

Zvonko Radosavljević and Dejan Ivković is with the Military Technical Institute, Belgrade, Ratka Resanovica bb, 11030 Belgrade, Serbia (e-mail: zvonko.radosavljevic@gmail.com and divkovic555@gmail.com).

Branko Kovačević is with the School of Electrical Engineering, University of Belgrade, 73 Bulevar kralja Aleksandra, 11020 Belgrade, Serbia (e-mail: kovacevic_b@etf.rs).

probability of detection (P_D) and clutter density. The clutter density is depending on target dynamics and characteristic of sensor. Generally, clutter is defined by number of selection measurement from size of selection gate. At begin, consider the target state $x_k \in \mathbb{R}^n$ at time interval k . The dynamic target trajectory state models at the time k are given by the:

$$x_k = Fx_{k-1} + v_k \quad (1)$$

where F is the propagation matrix, v_k is a zero mean and white Gaussian sequence with covariance R . At each scan the sensor returns a random number of random target and clutter measurements. At time k , one sensor delivers a set of measurements $z_k = \{z_{k,j}\}_{j=1}^{M_k}$ track out of which a set of measurements are selected for track update. Converted target measurement y is given by [10]:

$$y_k = Hx_k + w_k \quad (2)$$

where H is measurements matrix and the measurements noise w_k is zero mean and white Gaussian sequence. A measurements of target is present in each scan with a probability of detection P_D . Clutter measurements follow the Poisson distribution characterized at location by clutter measurements density $\rho_k(y)$ [11].

At time k a set of $m(k)$ measurements $Y(k) = \{y_i(k)\}_{i=1}^{m(k)}$ are detected, where each measurement either originate from one of n known linear measurement models or is a false detection. The sequences $v(k)$ are mutually independent and uncorrelated with the process noise.

A. Finite Sets statistics

Finite set statistics is the concept of belief-mass functions, which are non adaptive generalizations of probability mass functions and are equivalent to probability mass function on certain abstract topological spaces. The multi-target state and multi-target measurement at time k are represented as finite subsets X_k and Z_k , respectively.

The models of motion of the multi-target system using a randomly varying RFS is given by the [12]:

$$\Gamma_k = \Phi_k[X_{k-1}, V_{k-1}] \cup B_k[X_{k-1}] \quad (3)$$

where $\Phi_k[X_{k-1}]$ represents the dynamics of the existing targets, X_{k-1} is the random set of the state vectors of the random number of targets, and V_{k-1} denotes the system process noise, while $B_k[X_{k-1}]$ represents the process of target birth.

III. THE GAUSSIAN MIXTURE PHD FILTER ALGORITHM

In this section, we describe the linear-Gaussian multiple target model and the recently developed Gaussian Mixture PHD filter. The multiple target models for the PHD recursion

is described here. Each target follows a linear Gaussian dynamical model [13],

$$f_{k|k-1}(x|\xi) = N[x; F_{k-1}(\xi), Q_{k-1}] \quad (4)$$

$$g_k(y|x) = N[y; H_k(x), R_k] \quad (5)$$

where $N(\cdot; m, P)$ denotes a Gaussian density with mean m and covariance P , F_{k-1} is the state transition matrix, Q_{k-1} is the process noise covariance, H_k is the observation matrix and R_k is the observation noise covariance. The survival $p_{S_k}(x) = p_{S,k}$ and detection $p_{D_k}(x) = p_{D,k}$ probabilities are state independent. The intensities of the spontaneous birth and spawned targets are Gaussian mixtures,

$$\gamma_k(x) = \sum_{i=1}^{J_\gamma(k)} w_{\gamma_k}^i N[x; m_{\gamma_k}^i, P_{\gamma_k}^i] \quad (6)$$

$$\beta_{k|k-1}(x|\zeta) = \sum_{j=1}^{J_\beta(k)} w_{\beta_k}^j N[x; F_{\beta_{k-1}}^j(\zeta) + d_{\beta_{k-1}}^j, Q_{\beta_{k-1}}^j] \quad (7)$$

where $J_\gamma(k)$, $w_{\gamma_k}^i$, $m_{\gamma_k}^i$, $P_{\gamma_k}^i$, $i = 1, \dots, J_\gamma(k)$ are given model parameters that determine the shape of the birth intensity, similarly, $J_\beta(k)$, $w_{\beta_k}^j$, $F_{\beta_{k-1}}^j$, $d_{\beta_{k-1}}^j$ and $Q_{\beta_{k-1}}^j$, $j = 1, \dots, J_\beta(k)$ determine the shape of the spawning intensity of a target with previous state.

A. Algorithm steps

1 Prediction step

Under the assumptions that each target follows a linear Gaussian dynamical model, the survival and detection probabilities are constant, the intensities of the birth and spawned targets are Gaussian mixtures, and that the posterior intensity at time $k-1$ is a Gaussian mixture of the form [14]

$$D_{k-1|k-1}(x) = \sum_{i=1}^{J(k-1)} w_{i,k-1}^i N[x; m_{i,k-1}^i, P_{i,k-1}^i] \quad (11)$$

Then the predicted intensity to time k is also a Gaussian mixture, and is given by

$$D_{k|k-1}(x) = D_{S_{k|k-1}}(x) + D_{\beta_{k|k-1}}(x) + \gamma_k(x) \quad (12)$$

where $D_{S_{k|k-1}}(x)$ is the PHD of existing targets, $D_{\beta_{k|k-1}}(x)$ is the PHD for spawned targets, and $\gamma_k(x)$ is the PHD of spontaneous birth targets.

The density for existing targets, $D_{S_{k|k-1}}(x)$, is determined from the linear Gaussian model using the Kalman prediction equations,

$$D_{S_k|k-1}(x) = P_{S_k} \sum_{j=1}^{J(k-1)} w_{k-1}^j N[x; m_{S_k|k-1}^j, P_{S_k|k-1}^j] \quad (13)$$

where

$$m_{S_k|k-1}^j = F_{k-1} m_{k-1}^j \quad (14)$$

$$P_{S_k|k-1}^j = Q_{k-1} + F_{k-1} P_{k-1}^j F_{k-1}^T \quad (15)$$

and similarly for the spawned target density,

$$D_{\beta_k|k-1}(x) = \sum_{j=1}^{J(k-1)} \sum_{l=1}^{J_\beta(k)} w_{k-1}^j w_{\beta}^l N[x; m_{\beta_k|k-1}^{j,l}, P_{\beta_k|k-1}^{j,l}] \quad (16)$$

where

$$D_{k|k}(x) = [1 - p_{D_k}] D_{k|k-1}(x) + \sum_{y \in Y^k} \sum_{j=1}^{J(k|k-1)} w_{k-1}^j(y) N[x; m_{k|k}^j(y), P_{k|k}^j] \quad (20)$$

where the weights are calculated according to the closed form PHD update equation,

$$w_{k|k}^j(y) = \frac{p_{D_k} w_{k|k-1}^j N[y; H_k m_{k|k-1}^j, R_k + H_k P_{k|k-1}^j H_k^T]}{K_k(y) + p_{D_k} \sum_{l=1}^{J(k|k-1)} w_{k|k-1}^l N[y; H_k m_{k|k-1}^l, R_k + H_k P_{k|k-1}^l H_k^T]} \quad (21)$$

and the mean and covariance are updated with the Kalman filter update equations,

$$m_{k|k}^j(y) = m_{k|k-1}^j(y) + K_k^j [y - H_k m_{k|k-1}^j(y)] \quad (22)$$

$$P_{k|k}^j = [I - K_k^j H_k] P_{k|k-1}^j \quad (23)$$

$$K_k^j = P_{k|k-1}^j H_k^T [H_k P_{k|k-1}^j H_k^T + R_k]^{-1} \quad (24)$$

B. GMPHD practical implementation of algorithm

At begin, we given weights, mean and covariance of the each track:

$$\{W_{k-1}^{(i)}, m_{k-1}^{(i)}, P_{k-1}^{(i)}\}_{i=1}^{J(k-1)}$$

At each scan we given measurement set $Z(k)$, from the radar sensor .

Step 1: Prediction for birth targets

$i = 0$.

for $j=1, \dots, J_\gamma(k)$

$i=i+1$.

$$m_{\beta_k|k-1}^{j,l} = F_{\beta_k-1}^l m_{k-1}^j + d_{\beta_k-1}^l \quad (17)$$

$$P_{\beta_k|k-1}^{j,l} = Q_{\beta_k-1}^l + F_{\beta_k-1}^l P_{k-1}^j (F_{\beta_k-1}^l)^T \quad (18)$$

2 Update step

Under the above assumptions, and that the predicted intensity to time t is a Gaussian mixture of the form [15]

$$D_{k|k-1}(x) = \sum_{i=1}^{J(k|k-1)} w_{k|k-1}^i N[x; m_{k|k-1}^i, P_{k|k-1}^i] \quad (19)$$

Then the posterior intensity at time k is also a Gaussian mixture, and is given by

$$w_{k|k-1}^{(i)} = w_{\gamma_k}^{(j)},$$

$$m_{k|k-1}^{(i)} = m_{\gamma_k}^{(j)}$$

$$P_{k|k-1}^{(i)} = P_{\gamma_k}^{(j)}$$

End

for $j=1, \dots, J_\beta(k)$

for $l=1, \dots, J(k-1)$

$i=i+1$.

$$W_{k|k-1}^{(i)} = W_{k-1}^{(l)} W_{\beta_k}^{(j)}$$

$$m_{k|k-1}^{(i)} = d_{\beta_k-1}^{(j)} + F_{\beta_k-1}^{(j)} m_{k-1}^{(l)}$$

$$P_{k|k-1}^{(i)} = Q_{\beta_k-1}^{(j)} + F_{\beta_k-1}^{(j)} P_{k-1}^{(l)} (F_{\beta_k-1}^{(j)})^T$$

end
end

Step 2: Prediction for existing targets

for $j=1, \dots, J(k-1)$

$i=i+1$.

$$W_{k|k-1}^{(i)} = P_{S_k} W_{\gamma_k}^{(j)},$$

$$m^{(i)}_{k|k-1} = F_{k-1} m^{(j)}_{k-1}$$

$$P^{(i)}_{k|k-1} = Q_{k-1} + F_{k-1} P^{(j)}_{k-1} (F_{k-1})^T$$

end

$$J(k|k-1) = i$$

Step 3: Construction of PHD update components

for $j=1, \dots, J(k|k-1)$

$$\eta^{(j)}_{k|k-1} = H_k m^{(j)}_{k|k-1}$$

$$S^{(j)}_k = R_k + H_k P^{(j)}_{k|k-1} H_k^T$$

$$K_k = P^{(j)}_{k|k-1} H_k^T [S^{(j)}_k]^{-1}$$

$$P^{(j)}_{k|k} = [I - K^{(j)}_k H_k] P^{(j)}_{k|k-1}$$

end

Step 3: (Update)

for $j=1, \dots, J(k|k-1)$

$$w^{(j)}_k = (1 - p_{D,k}) w^{(j)}_{k|k-1},$$

$$m^{(j)}_k = m^{(j)}_{k|k-1},$$

$$P^{(j)}_k = P^{(j)}_{k|k-1}$$

end

$l=0.$

for each $y \in Y^k$

$l=l+1,$

for $j=1, \dots, J(k|k-1)$

$$w^{[lJ(k|k-1)+j]}_k =$$

$$p_{Dk} w^{(j)}_{k|k-1} N(y; \eta^{(j)}_{k|k-1}, S^{(j)}_k)$$

$$m^{[lJ(k|k-1)+j]}_k =$$

$$= m^{(j)}_{k|k-1} + K^{(j)}_k [y - \eta^{(j)}_{k|k-1}]$$

$$P^{[lJ(k|k-1)]}_k = P^{(j)}_{k|k}$$

End

$$w^{[lJ(k|k-1)+j]}_k = \frac{w^{[lJ(k|k-1)+j]}_k}{K^{(y)}_k + \sum_{i=1}^{J(k|k-1)} w^{[lJ(k|k-1)+i]}_k}, \quad j = 1, \dots, J(k|k-1)$$

end

$$J(k) = lJ(k|k-1) + J(k|k-1)$$

Output $\{w^{(i)}_k, m^{(i)}_k, P^{(i)}_k\}_{i=1}^{J(k)}$

C. Output Calculation

Finally, we can calculate the output state estimate and covariance (for output purpose only):

$$\hat{x}_{k|k} = \sum_{l=1}^{Np} w_k^l x_k^l \quad (25)$$

$$P_k = \left(\sum_{l=1}^{Np} w_k^l \cdot x_k^l \cdot x_k^{lT} \right) - \hat{x}_k \cdot \hat{x}_k^T \quad (26)$$

IV. RESULTS OF EXPERIMENT

For the purpose of simulations, we propose a two-dimensional scenario (Fig. 1) with an unknown crossing and time varying number of targets in clutter over the region $[-500; 500] \times [-500; 500]$. The state $x_k = [x_k \ v_{xk} \ y_k \ v_{yk}]^T$, of each target consists of position (x_k, y_k) and velocity (v_{xk}, v_{yk}) , while the measurement is a noisy version of the position [16].

Each target has survival probability $p_{S;k} = 0,9$, detection probability $p_{D;k} = 0,99$ and follows the linear Gaussian dynamics. Each simulation experiment consists of a number of simulation runs. In each simulation run, targets will repeat their trajectories. The measurements are generated independently. Each algorithm uses the same set of measurements. False tracks may be initiated using target measurements, either in a conjunction with a clutter measurement, or by using measurements from different targets in different scans. The sampling period of radar sensor is $T=1s$. Duration of the scenario is 70 scans. The implemented GM-PHD is evaluated by Monte Carlo (MC) simulations over representative 2-dimensional test scenario. A target motion scenario (Fig.2) includes non-maneuvering flights modes. Dimension of terrain surveillance is $x=500m$ and $y=500m$.

Transition matrix and process noise matrix are given by:

$$F = \begin{bmatrix} 1 & T & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & T \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad Q_k = q \begin{bmatrix} T^3/3 & T^2/2 & 0 & 0 \\ T^2/2 & T & 0 & 0 \\ 0 & 0 & T^3/3 & T^2/2 \\ 0 & 0 & T^2/2 & T \end{bmatrix} \quad (27)$$

respectively. Measurements matrix and measurements noise matrix is given by:

$$H = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}, R_k = \begin{bmatrix} \sigma_x^2 & 0 \\ 0 & \sigma_y^2 \end{bmatrix} \quad (28)$$

respectively.

For the purpose of performance testing tracking targets, we propose Wasserstein distance diagram.

A. Wasserstein distance:

The Wasserstein distance from theoretical statistics was adopted as a means of defining a performance metric for multi target distances which penalizes when the estimate of the number of targets is incorrect. When the number of targets is estimated correctly, the Wasserstein distance is the same as the Hausdor distance but the Hausdor does not penalise for incorrectly estimating the number of targets. This metric has been used for assessing the performance of the PHD filter.

Let X_t and Y_t be the finite sets of target states and estimated target states at time interval k . The L^P Wasserstein distance between the two sets is defined as:

$$d_p^W(X_t, Y_t) = \inf_C \left(\sum_{x_i \in X_t} \sum_{\hat{x}_j \in \hat{X}_t} C_{ij} d(x_i, \hat{x}_j)^p \right)^{1/p} \quad (29)$$

where C is an $|X_t| \times |\hat{X}_t|$ matrix $\{C_{ij}\}$ such that $\forall i = 1 \dots |X_t|, j = 1 \dots |\hat{X}_t|$:

$$\sum_{i=1}^{|X_t|} C_{ij} = \frac{1}{|\hat{X}_t|}, \sum_{j=1}^{|\hat{X}_t|} C_{ij} = \frac{1}{|X_t|}, C_{ij} \geq 0 \quad (30)$$

The L^∞ Wasserstein distance is defined as:

$$d_\infty^W(X_t, \hat{X}_t) = \inf_C \max_{x_i \in X_t, \hat{x}_j \in \hat{X}_t} C_{ij} d(x_i, \hat{x}_j) \quad (31)$$

Where $\tilde{C}_{ij} = 1$ if $C_{ij} > 0$ and $\tilde{C}_{ij} = 0$ if $C_{ij} = 0$

B. Two point differencing initializations

Initialization with the difference of successive observations (Two point differentiation initializing) uses measurements located in the 'window' of the trace from two successive scans to initialize the trace. This procedure is repeated for all measurements from the scan $k-1$. Consider such a measurement $z_{k-1,j}$. The new trace is initialized by measurement $z_{k-1,j}$ and each selected measurement $z_{k,i}$, forming a Gaussian probability density function with the mean value given as:

$$\hat{x}_k^{(2)} = \begin{bmatrix} z_{k,i} \\ \frac{z_{k,i} - z_{k-1,j}}{\Delta T_k} \end{bmatrix} \quad (32)$$

where ΔT_k one is the period of rotation of the radar antenna. Since there is no a priori knowledge of the target speed, it can be modeled through a uniform distribution of a priori pdf measurements. At the moment k , the N_p particles of the mean value $\hat{x}_k^{(2)}$ are initialized, as well as the symmetric and semi-finite covariance matrices $P_{(2/2)}$ which correspond to the normal distribution of several variables (Figure 1):

$$x_k^p = N[\hat{x}_k^{(2)}; P_{(2/2)}] \quad (33)$$

where the initial covariance error of the condition is calculated under the assumption that there is no process noise [24]:

$$P_{(2/2)} = \begin{bmatrix} R_k & R_k / T \\ R_k / T & 2R_k / T^2 \end{bmatrix} \quad (34)$$

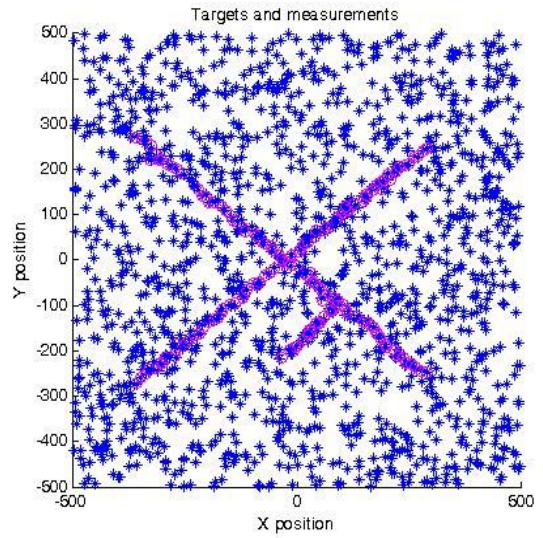


Fig. 1. Simulation scenario: targets and measurements .

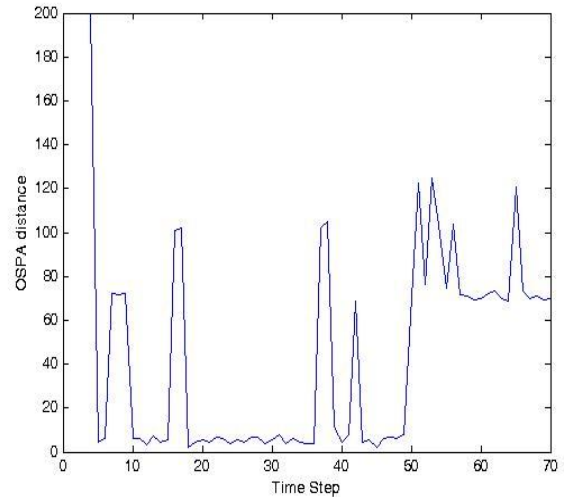


Fig. 2. Wasserstein distance metric (OSPA diagram)

Simulation results (OSPA diagram- Fig. 2) show good performance for tracking two crossing targets with one spawned targets, from scan 50 (Fig. 1). Blue dots (measurements) and magenta circle (targets) show good initializing and tracking targets in heavy clutter environments.

V. CONCLUSION

An improved GM PHD algorithm for radar sensor approaches has been presented for tracking multiple targets in high clutter density which has the ability to estimate the number of targets, track the trajectories of the targets over time, operate with missed detections and give the trajectories of the targets in the past once a target has been identified. It has been shown to outperform the ability to operate in clutter with fewer false tracks and can initiate and eliminate targets more accurately.

Future research should better determine the association of radar received measurements with existing targets as well as automatic initialization of targets. The theoretical constraints of the proposed tracking algorithm have been discussed in the case of crossing targets. It is anticipated that the problem of retaining the correct target identity in this scenario can be resolved by considering the previous trajectories of targets.

Especially, the proposed algorithm will be tested in practice on the example of video tracking [17, 18].

REFERENCES

- [1] S. Blackman, *Multiple Target Tracking with Radar Applications*. Artech House, Norwood, 1986.
- [2] Challa S., Evans R., Morelande M., Mušicki D.: *Fundamentals of Object Tracking*, Cambridge University Press 2011
- [3] R. Mahler, "Multitarget Bayes filtering via first-order multitarget moments," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 39, no. 4, pp. 1152–1178, Oct. 2003.
- [4] O. Erdinc, P. Willet, and Y. Bar-Shalom. *Probability Hypothesis Density Filter for Multitarget Multisensor Tracking*. Proc. FUSION 2005.
- [5] B. Vo and W. K. Ma. *The Gaussian Mixture Probability Hypothesis Density Filter*. *IEEE Transactions on Signal Processing*, 2006.
- [6] B. Vo, S. Singh, and A. Doucet, "Sequential Monte Carlo methods for multi target filtering with random finite sets," *IEEE Trans. Aerosp. Electron.Syst.*, vol. 41, no. 4, pp. 1224–1245, Oct. 2005.
- [7] D. Clark and J. Bell, "Convergence results for the particle PHD filter," *IEEE Trans. Signal Process.*, vol. 54, no. 7, pp. 2652–2661, Jul. 2006.
- [8] R. Mahler. *Multi-target Bayes filtering via first order multi target moments*. *IEEE Transactions on Aerospace and Electronic Systems*, 39, No.4:1152-1178, 2003.
- [9] K. Panta, B. Vo, S. Singh, and A. Doucet. *Probability hypothesis density filter versus multiple hypothesis tracking*. *Proceedings of SPIE, Volume 5429 Signal Processing, Sensor Fusion, and Target Recognition XIII*, Ivan Kadar, Editor, August 2004, pp. 284-295.
- [10] Z. Radosavljević, T.L. Song, B.Kovacevic, *Linear Multi-Target IPF Algorithm for Automatic Tracking*, *Scientific Technical Review*, 2016, Vol.66, No.1, pp.3-10, Belgrade 2016.
- [11] Z. Radosavljević, D. Mušicki, B. Kovačević, W. C. Kim, and T. L. Song, *Integrated particle filter for target tracking in clutter*, *IET proceedings on Radar Sonar and Navigation*, DOI: 10.1049/iet-rsn.2014.0341 , Print ISSN 1751-8784, May, 2015.
- [12] T. L. Song, D. Mušicki, D.S. Kim and Z.Radosavljević, *Gaussian mixtures in multi-target tracking: a look at Gaussian mixture probability hypothesis density and integrated track splitting*, *IET proceedings on Radar Sonar and Navigation*, Vol 6, no 5, pp. 359-364, June 2012.
- [13] B. Kovačević, Z. Radosavljević and D. Ivković, "The Interacting Multiple Model GM PHD for Single Target Tracking", *Proc. of 64. International Conference IcETTRAN 2020*, Belgrade, September 28.-29., 2020.
- [14] Z. Radosavljevic, D. Musicki, B. Kovacevic, W.C. Kim and T. L. Song, 'Integrated Particle Filter for Target Tracking', *Proceedings of International Conference on Electronics, Information and Communications (ICEIC)*. Kota Kinabalu, Malesya, Januar 2014.
- [15] Z. Radosavljevic, D. Paunovic, D. Ivkovic, D. Nikolic, *An Efficient Solid State VHF Band Radar Transmitter*, 6th International Scientific Conference on Defensive Technologies OTEH 2014, Belgrade, 2014.
- [16] Z. Radosavljevic, D. Musicki, *Limits of target tracking in heavy clutter*, *ASIA-Pacific International Conference of Synthetic Aperture Radar APSAR 2011*, Seoul, Republic of Korea., 2011.
- [17] Z. Fu, F. Angelini, J. Chambers, and S. M. Naqvi, "Multi-Level Cooperative Fusion of GM-PHD Filters for Online Multiple Human Tracking," *IEEE trans. Multimedia*. (Early Access), Mar. 2019.
- [18] N. L. Baisa and A. Wallace, "Development of a n-type GM-PHD filter for multiple target, multiple type visual tracking." *J. Vis. Commun. Image Represent.*, vol. 59, pp. 257–271, 2019

Four-Stage Recursive Least Squares Algorithm for CARARMA Systems

Nasar Aldian Ambark Shashoa⁽¹⁾, Senior Member, IEEE, Ahmed J. Abougarair⁽²⁾, Abdulhakim Amer Agil⁽³⁾ and Abdurrezag S. Elmezughi⁽⁴⁾

(1) Department of Electrical and Computer Engineering, Libyan Academy

(2) Department of Electrical and Electronic Engineering, University of Tripoli

(3) Department of Mechanical and Industrial Engineering, University of Tripoli and Embry-Riddle Aeronautical University Worldwide

(4) Department of Computer Engineering, Azzaytuna University

Abstract—this paper derives four -stage recursive least squares parameter estimation algorithm for Controlled Autoregressive Autoregressive Moving Average (CARARMA) systems. By applying the decomposition technique, (CARARMA) system decompose into four subsystems, which contain one parameter vector each. Compared with the recursive generalized least squares algorithm, the proposed algorithm improves the accuracy of estimated parameters and decrease the computational burden. The simulation example is given to indicate the efficiency of the proposed algorithm.

Keywords— parameter estimation, decomposition technique, computational burden, (Q-Q) plot, estimation error.

I. INTRODUCTION

The modeling of manufacturing systems plays a critical function in system analysis and control. System identification is fundamental for structuring the systems mathematical models and parameter estimation is the basis of system identification [1]. In the past decades, researchers have done a lot of research work on parameter estimation methods such as the iterative algorithms, the maximum likelihood methods, the filtering technique, multiple-stage algorithms and so on. Multi-stage least squares identification algorithm is recently employed for parameter estimation of various system models. The main contribution in this technique is to reduce the computational load. The computational load is a significant feature of the recursive algorithms. Low computational load can help the recursive algorithms realize best performances in the systems. The computational efficiency in least squares [2]. The main idea is based on the decomposition technique that can split the main identification system into few small-size subsystems, so as to enhance the computational efficiency and easy to solve. Four-Stage Recursive Least Squares Algorithm is proposed for CARARMA Systems. The decomposition technique is using in this algorithm to reduce the Computational burden. The layout of the rest of this article is arranged as follows. Section 2 derives four-stage recursive identification algorithm. Section 3 presents recursive generalized extended least squares algorithm. Section 4, a illustrative example is given to demonstrate the performance of this algorithm. Finally, the conclusions are offered in section 5.

II. FOUR-STAGE RECURSIVE IDENTIFICATION ALGORITHM

Consider linear, time-invariant, discrete-time system [3] as shown in fig. 1, described by (CARARMA), which can be expressed as:

$$A(z)y(k) = B(z)u(k) + \frac{D(z)}{C(z)}v(k) \quad (1)$$

where

$\{u(k)\}$ and $\{y(k)\}$ are the input and output, $\{v(k)\}$ is the white noise sequence with zero mean and unit variance σ^2 [9] and $A(z), B(z), C(z)$ and $D(z)$ are polynomials in the unit backward shift operator z [4].

$$A(z) = 1 + \sum_{i=1}^{n_a} a_i z^{-i}$$

$$B(z) = \sum_{i=1}^{n_b} b_i z^{-i}$$

$$C(z) = 1 + \sum_{i=1}^{n_c} c_i z^{-i}$$

$$D(z) = 1 + \sum_{i=1}^{n_c} d_i z^{-i}$$

The equation (1) can be rewritten as:

$$y(k) = [1 - A(z)]y(k) + B(z)u(k) + w(k) \quad (2)$$

$w(k)$ is defined as

$$w(k) = [1 - C(z)]w(k) + D(z)v(k) \quad (3)$$

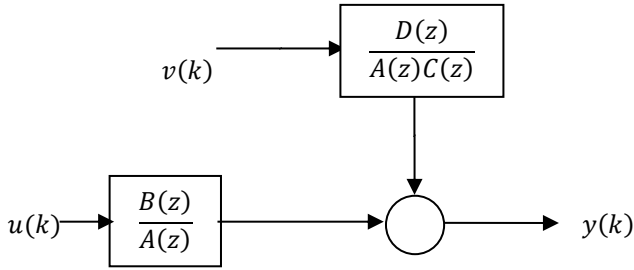


Fig. 1. CARARMA Model.

The parameter vector $\{\theta_1, \theta_2, \theta_3, \theta_4\}$ and the information vectors $\{\varphi_1(k), \varphi_2(k), \varphi_3(k), \varphi_4(k)\}$ are defined as:

$$\theta_1 := [a_1, a_2, \dots, a_{n_a}]^T$$

$$\theta_2 := [b_1, b_2, \dots, b_{n_b}]^T$$

$$\theta_3 := [c_1, c_2, \dots, c_{n_c}]^T$$

$$\theta_4 := [d_1, d, \dots, d_{n_d}]^T$$

$$\varphi_1(k) = [-y(k-1), -y(k-2), \dots, -y(k-n_a)]^T$$

$$\varphi_2(k) = [u(k-1), u(k-2), \dots, u(k-n_b)]^T$$

$$\varphi_3(k) = [-w(k-1), -w(k-2), \dots, -w(k-n_c)]^T$$

$$\varphi_4(k) = [v(k-1), v(k-2), \dots, v(k-n_d)]^T$$

Then,

$$y(k) = \varphi_1^T(k)\theta_1 + \varphi_2^T(k)\theta_2 + w(k) \quad (4)$$

$$w(k) = \varphi_3^T(k)\theta_3 + \varphi_4^T(k)\theta_4 + v(k) \quad (5)$$

Using equation (5), equation (4) can be written as [5].

$$y(k) = \varphi_1^T(k)\theta_1 + \varphi_2^T(k)\theta_2 + \varphi_3^T(k)\theta_3 + \varphi_4^T(k)\theta_4 + v(k)$$

$$y(k) = \varphi^T(k)\theta + v(k) \quad (6)$$

$$\varphi^T(k) = [\varphi_1^T(k) \ \varphi_2^T(k) \ \varphi_3^T(k) \ \varphi_4^T(k)]$$

$$\theta = [\theta_1; \theta_2; \theta_3; \theta_4]$$

The basic principle in this work is to decomposed the identification system in equation (6) into four subsystems, and the parameters of each subsystem are estimated separately [6].

Define four intermediate output variables:

$$y_1(k) = y(k) - \varphi_2^T(k)\theta_2 - \varphi_3^T(k)\theta_3 - \varphi_4^T(k)\theta_4, \quad (7)$$

$$y_2(k) = y(k) - \varphi_1^T(k)\theta_1 - \varphi_3^T(k)\theta_3 - \varphi_4^T(k)\theta_4, \quad (8)$$

$$y_3(k) = y(k) - \varphi_1^T(k)\theta_1 - \varphi_2^T(k)\theta_2 - \varphi_4^T(k)\theta_4, \quad (9)$$

$$y_4(k) = y(k) - \varphi_1^T(k)\theta_1 - \varphi_2^T(k)\theta_2 - \varphi_3^T(k)\theta_3, \quad (10)$$

This system can be decomposed into the following four subsystems:

$$y_i(k) = \varphi_i^T(k)\theta_i + v(k), \quad i = 1,2,3,4. \quad (11)$$

Define the cost functions:

$$J_i(\theta_i) := \sum_{j=1}^k [y_i(j) - \varphi_i^T(j)\theta_i]^2, \quad i = 1,2,3,4.$$

Let $\hat{\theta}(k) = [\hat{\theta}_1(k); \hat{\theta}_2(k); \hat{\theta}_3(k); \hat{\theta}_4(k)]$ be the estimate of $\theta = [\theta_1; \theta_2; \theta_3; \theta_4]$. Also, let the partial derivatives of $J_i(\theta_i), i = 1,2,3,4$ with respect to θ_i be zero

$$\frac{\partial J_i(\theta_i)}{\partial \theta_i} = -2\varphi_i(j) \sum_{j=1}^k [y_i(j) - \varphi_i^T(j)\hat{\theta}_i(k)] = 0 \quad (12)$$

Then, the following RLS algorithm can be obtained:

$$\hat{\theta}_i(k) = \hat{\theta}_i(k-1) + L_i(k)[y_i(k) - \varphi_i^T(k)\hat{\theta}_i(k-1)] \quad (13)$$

$$L_i(k) = P_i(k-1)\varphi_i(k)[1 + \varphi_i^T(k)P_i(k-1)\varphi_i(k)]^{-1},$$

$$P_i(k) = [I - L_i(k)\varphi_i^T(k)]P_i(k-1),$$

$$P_i(0) = p_0I, \quad i = 1,2,3,4$$

Substituting equations (9, 10, 11 and 12) into equation (13) [7].

$$\hat{\theta}_1(k) = \hat{\theta}_1(k-1) + L_1(k)[y(k) - \varphi_2^T(k)\theta_2 - \varphi_3^T(k)\theta_3 - \varphi_4^T(k)\theta_4 - \varphi_1^T(k)\hat{\theta}_1(k-1)] \quad (14)$$

$$\hat{\theta}_2(k) = \hat{\theta}_2(k-1) + L_2(k)[y(k) - \varphi_1^T(k)\theta_1 - \varphi_3^T(k)\theta_3 - \varphi_4^T(k)\theta_4 - \varphi_2^T(k)\hat{\theta}_2(k-1)] \quad (15)$$

$$\hat{\theta}_3(k) = \hat{\theta}_3(k-1) + L_3(k)[y(k) - \varphi_1^T(k)\theta_1 - \varphi_2^T(k)\theta_2 - \varphi_4^T(k)\theta_4 - \varphi_3^T(k)\hat{\theta}_3(k-1)] \quad (16)$$

$$\hat{\theta}_4(k) = \hat{\theta}_4(k-1) + L_4(k)[y(k) - \varphi_1^T(k)\theta_1 - \varphi_2^T(k)\theta_2 - \varphi_3^T(k)\theta_3 - \varphi_4^T(k)\hat{\theta}_4(k-1)] \quad (17)$$

Although the all RLS sub-algorithms are derived for calculating $\hat{\theta}_1(k), \hat{\theta}_2(k), \hat{\theta}_3(k)$ and $\hat{\theta}_4(k)$. They are unattainable to be completed, because the right-hand sides all RLS sub-algorithms include the unknown parameters $\theta_1, \theta_2, \theta_3$ and θ_4 . In order to handles this issue; the parameters $\theta_i, i = 1,2,3,4$ are replaced with $\hat{\theta}_i(k-1)$ [8].

Then, $\hat{\theta}_1(k), \hat{\theta}_2(k), \hat{\theta}_3(k)$ and $\hat{\theta}_4(k)$ can be calculated by

$$\begin{aligned} \hat{\theta}_1(k) &= \hat{\theta}_1(k-1) + L_1(k) [y(k) - \varphi_2^T(k)\hat{\theta}_2(k-1) - \varphi_3^T(k)\hat{\theta}_3(k-1) - \varphi_4^T(k)\hat{\theta}_4(k-1) - \varphi_1^T(k)\hat{\theta}_1(k-1)] \\ &= \hat{\theta}_1(k-1) + L_1(k)[y(k) - \varphi^T(k)\hat{\theta}(k-1)] \end{aligned} \quad (18)$$

$$\begin{aligned} \hat{\theta}_2(k) &= \hat{\theta}_2(k-1) + L_2(k)[y(k) - \varphi_1^T(k)\hat{\theta}_1(k-1) - \varphi_3^T(k)\hat{\theta}_3(k-1) - \varphi_4^T(k)\hat{\theta}_4(k-1) - \varphi_2^T(k)\hat{\theta}_2(k-1)] \end{aligned}$$

$$= \hat{\theta}_2(k-1) + L_2(k)[y(k) - \varphi^T(k)\hat{\theta}(k-1)] \quad (19)$$

$$\hat{\theta}_3(k) = \hat{\theta}_3(k-1) + L_3(k)[y(k) - \varphi_1^T(k)\hat{\theta}_1(k-1) - \varphi_2^T(k)\hat{\theta}_2(k-1) - \varphi_4^T(k)\hat{\theta}_4(k-1) - \varphi_3^T(k)\hat{\theta}_3(k-1)],$$

$$= \hat{\theta}_3(k-1) + L_3(k)[y(k) - \varphi^T(k)\hat{\theta}(k-1)] \quad (20)$$

$$\hat{\theta}_4(k) = \hat{\theta}_4(k-1) + L_4(k)[y(k) - \varphi_1^T(k)\hat{\theta}_1(k-1) - \varphi_2^T(k)\hat{\theta}_2(k-1) - \varphi_3^T(k)\hat{\theta}_3(k-1) - \varphi_4^T(k)\hat{\theta}_4(k-1)]$$

$$= \hat{\theta}_4(k-1) + L_4(k)[y(k) - \varphi^T(k)\hat{\theta}(k-1)] \quad (21)$$

Notice that the right-hand sides of equations (18)–(21) include the unknown information vectors $\varphi_3(k)$ which contains the unknown inner variables $w(k-i)$ and $\varphi_4(k)$ which contains unknown inner variables $v(k-i)$, so that, this algorithm cannot be performed [9]. Replacing $w(k-i)$ and $v(k-i)$ with their estimates $\hat{w}(k-i)$ and $\hat{v}(k-i)$, and $\hat{\varphi}_3(k)$, $\hat{\varphi}_4(k)$ are written as

$$\hat{\varphi}_3(k) = [-\hat{w}(k-1), -\hat{w}(k-2), \dots, -\hat{w}(k-n_c)]^T \quad (22)$$

$$\hat{\varphi}_4(k) = [\hat{v}(k-1), \hat{v}(k-2), \dots, \hat{v}(k-n_d)]^T \quad (23)$$

$$\hat{\varphi}(k) = [\varphi_1, \varphi_2, \hat{\varphi}_3, \hat{\varphi}_4]^T$$

Where,

$$\hat{w}(k) = y(k) - \varphi_1^T(k)\hat{\theta}_1 - \varphi_2^T(k)\hat{\theta}_2 \quad (24)$$

$$\hat{v}(k) = \hat{w}(k) - \hat{\varphi}_3^T(k)\hat{\theta}_3 - \hat{\varphi}_4^T(k)\hat{\theta}_4 \quad (25)$$

Thus, Four-Stage Recursive Least Squares Algorithm for CARARMA Systems is summarize as:

$$\hat{\theta}_1(k) = \hat{\theta}_1(k-1) + L_1(k)[y(k) - \hat{\varphi}^T(k)\hat{\theta}(k-1)] \quad (26)$$

$$L_1(k) = P_1(k-1)\varphi_1(k)[1 + \varphi_1^T(k)P_1(k-1)\varphi_1(k)]^{-1} \quad (27)$$

$$P_1(k) = [I - L_1(k)\varphi_1^T(k)]P_1(k-1) \quad (28)$$

$$\varphi_1(k) = [-y(k-1), -y(k-2), \dots, -y(k-n_a)]^T \quad (29)$$

$$\hat{\theta}_2(k) = \hat{\theta}_2(k-1) + L_2(k)[y(k) - \hat{\varphi}^T(k)\hat{\theta}(k-1)] \quad (30)$$

$$L_2(k) = P_2(k-1)\varphi_2(k)[1 + \varphi_2^T(k)P_2(k-1)\varphi_2(k)]^{-1} \quad (31)$$

$$P_2(k) = [I - L_2(k)\varphi_2^T(k)]P_2(k-1) \quad (32)$$

$$\varphi_2(k) = [u_1(k-1), u_1(k-2), \dots, u_1(k-n_1)]^T \quad (33)$$

$$\hat{\theta}_3(k) = \hat{\theta}_3(k-1) + L_3(k)[y(k) - \hat{\varphi}^T(k)\hat{\theta}(k-1)] \quad (34)$$

$$L_3(k) = P_3(k-1)\hat{\varphi}_3(k)[1 + \hat{\varphi}_3^T(k)P_3(k-1)\hat{\varphi}_3(k)]^{-1} \quad (35)$$

$$P_3(k) = [I - L_3(k)\hat{\varphi}_3^T(k)]P_3(k-1) \quad (36)$$

$$\hat{\varphi}_3(k) = [-\hat{w}(k-1), -\hat{w}(k-2), \dots, -\hat{w}(k-n_c)]^T \quad (37)$$

$$\hat{\theta}_4(k) = \hat{\theta}_4(k-1) + L_4(k)[y(k) - \hat{\varphi}^T(k)\hat{\theta}(k-1)] \quad (38)$$

$$L_4(k) = P_4(k-1)\hat{\varphi}_4(k)[1 + \hat{\varphi}_4^T(k)P_4(k-1)\hat{\varphi}_4(k)]^{-1} \quad (39)$$

$$P_4(k) = [I - L_4(k)\hat{\varphi}_4^T(k)]P_4(k-1) \quad (40)$$

$$\hat{\varphi}_4(k) = [\hat{v}(k-1), \hat{v}(k-2), \dots, \hat{v}(k-n_d)]^T \quad (41)$$

III. THE RECURSIVE GENERALIZED LEAST SQUARES ALGORITHM

In order to demonstrate the features of the Four-Stage -RLS algorithm, the following discuss recursive generalized extended least squares (RGELS) algorithm [10]. As we all know, (RGELS) algorithm can identify CARARMA Systems. The main issue is to replace $w(k-i)$ and $v(k-i)$ with their estimates in the information vector. Finally, (RGLS) algorithm can obtained and the estimated parameter vectors of CARARMA model can be calculated as [11].

$$\hat{\theta}(k) = \hat{\theta}(k-1) - L(k)[y(k) - \hat{\varphi}^T(k)\hat{\theta}(k-1)] \quad (42)$$

$$L(k) = \frac{P(k-1)\hat{\varphi}(k)}{1 + \hat{\varphi}^T(k)P(k-1)\hat{\varphi}(k)} \quad (43)$$

$$P(k) = [I - L(k)\hat{\varphi}^T(k)]P(k-1). \quad (44)$$

$$\hat{\varphi}(k) = [\varphi_s(k) \ \varphi_n(k)]^T, \ \hat{\theta}(k) = [\hat{\theta}_s(k); \ \hat{\theta}_n(k)] \quad (45)$$

$$\varphi_s^T(k) = [-y(k-1), -y(k-2), \dots, -y(k-na), u(k-1), u(k-2), \dots, u(k-n_b)] \quad (46)$$

$$\varphi_n^T(k) = [-w(k-1), -w(k-2), \dots, -w(k-n_c), v(k-1), v(k-2), \dots, v(k-n_d)] \quad (47)$$

$$\hat{w}(k) = y(k) - \varphi_s^T(k)\hat{\theta}_s(k) \quad (48)$$

$$\hat{v}(k) = y(k) - \hat{\varphi}^T(k)\hat{\theta}(k) \quad (49)$$

$$\hat{\theta}_s(k) = [\hat{a}_1(k), \hat{a}_2(k), \dots, \hat{a}_{n_a}(k), \hat{b}_1(k), \hat{b}_1(k), \dots, \hat{b}_{n_b}(k)] \quad (50)$$

$$\hat{\theta}_n(k) = [\hat{c}_1(k), \hat{c}_2(k), \dots, \hat{c}_{n_c}(k), \hat{d}_1(k), \hat{d}_1(k), \dots, \hat{d}_{n_d}(k)] \quad (51)$$

IV. SIMULATION RESULTS

The effectiveness of Four-Stage -RLS algorithm has been checked on samples generated by the following (CARARMA) model:

$$A(z)y(k) = B(z)u(k) + \frac{D(z)}{C(z)}v(k)$$

$$A(z) = 1 + 0.72z^{-1} + 0.81z^{-2}$$

$$B(z) = 0.22z^{-1} + 0.33z^{-2}$$

$$C(z) = 1 + 0.65z^{-1}$$

$$D(z) = 1 + 0.55z^{-1}$$

$$\begin{aligned} \theta &= [a_1, a_2, b_1, b_2, c_1, d_1]^T, \\ &= [0.72, 0.81, 0.22, 0.33, 0.65, 0.55]^T, \end{aligned}$$

The input $u(k)$ adopts a persistent excitation sequence with $m = 0$ and $\sigma = 1$, while $v(k)$ is a white noise sequence with $m = 0$ and $\sigma = 0.3^2$.

Quantile-quantile (Q-Q) plot and estimation error have been used in order to evaluate the performance of the proposed algorithm.

A. Quantile-quantile (Q-Q) plot

Quantile-quantile (Q-Q) plot is a diagnostic tool, which is widely used to assess the distributional similarities and differences between two independent univariate samples. It is also used to determine how well a specified probability distribution fits a given sample [12].

The following steps create a Quantile-Quantile-plot (Q-Q-plot)

1. Rank your data in ascending order: x_1, \dots, x_n
2. For each data point x_i compute the empirical percentiles ρ_i of data points smaller than or equal to x_i , i.e

$$\rho_i = \frac{i}{n}$$

3. Define a theoretical matching normal distribution as the normal distribution with the same mean and variance as the sample.

4. For each percentile ρ_i find the corresponding quantile y_i in the theoretical normal distribution.

5. Plot y_i against x_i

6. If $x_i = y_i$ for all i we get a straight line indicating that x_i 's are normally distributed [13].

B. The estimation error

The estimation error is determined as [14].

where

$$\delta = \sqrt{\frac{\|\hat{\theta}_1 - \theta_1\|^2 + \|\hat{\theta}_2 - \theta_2\|^2 + \|\hat{\theta}_3 - \theta_3\|^2 + \|\hat{\theta}_4 - \theta_4\|^2}{\|\theta_1\|^2 + \|\theta_2\|^2 + \|\theta_3\|^2 + \|\theta_4\|^2}} \quad (40)$$

As it's known, when the parameter estimation errors become smaller with the sequences increasing, the accuracy of the estimated parameters is increased.

Four-Stage -RLS algorithm and (RGELS) algorithm have been applied to estimate the parameters of the system and the estimation errors for these algorithms versus sequences (k) are shown in Fig. 2.

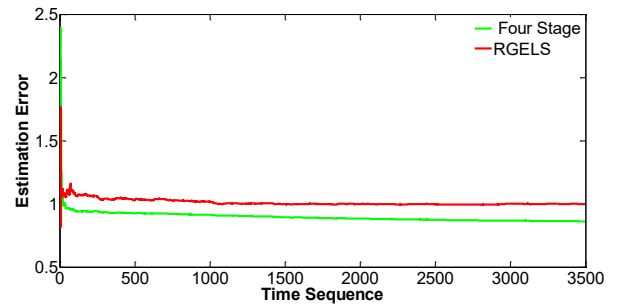


Fig. 2. Four-Stage -RLS and (RGELS) estimation errors versus k

The figure demonstrates that Four-Stage -RLS algorithm can achieved more accurate estimates than the (RGELS) algorithm.

Fig. 3 shows Q-Q plots for a residuals generation of Four-Stage -RLS algorithm (blue color) and (RGELS) algorithm (red color) which are tested in comparison with the standard normal distribution (black color) to determine the extent of the quantiles match.

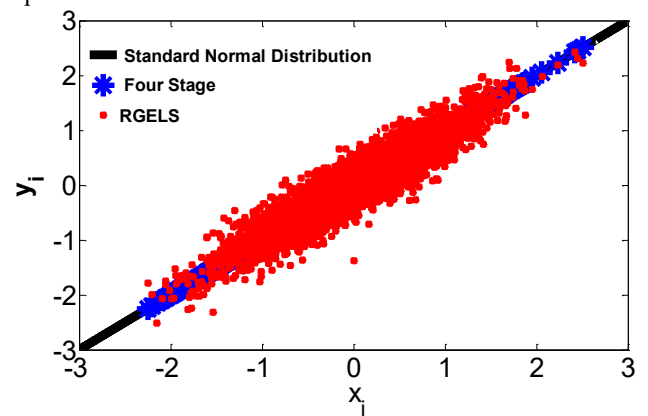


Fig. 3 Q-Q plots for a residuals generation of Four-Stage -RLS algorithm " blue color" and (RGELS) algorithm" red color".

The figure offers that, the result of the Four-Stage -RLS algorithm matches more closely along standard normal distribution than the result of the (RGELS) algorithm. This result indicates that the proposed algorithm is more effective than (RGELS) algorithm.

The estimated output and the true output of Four-Stage -RLS algorithm are plotted in Fig. 4.

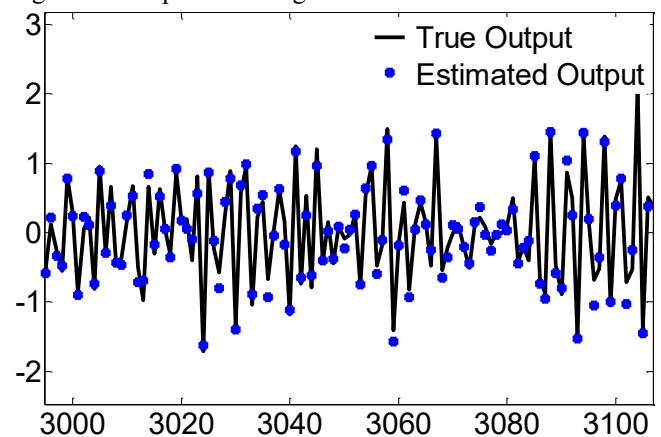


Fig. 4 The estimated output and the true output of Four-Stage -RLS algorithm

The figure illustrate that estimated output is close to the true output, which means the estimated model can fit the validation data well [10].

V. CONCLUSIONS

In this paper, Four-Stage recursive identification algorithm is derived. Through a simulation results, it is shown that the proposed algorithm has some advantages, in comparison with (RGLES) algorithm. Four-Stage -RLS improves the accuracy for estimated parameters, and has higher computational efficiency than (RGLES) algorithm. The proposed technique in this article can be extended to study the parameter estimation algorithm of (MISO) and (MIMO) systems with colored noise.

REFERENCES

- [1] Y. Ji and Z. Kang "Three-stage forgetting factor stochastic gradient parameter estimation methods for a class of nonlinear systems," *Int J Robust Nonlinear Control*, 2020.
- [2] M. Junxia, C.Jing, D. Feng "Filtering Based Multi-Stage Recursive Least Squares Parameter Estimation Algorithm for Input Nonlinear Output-Error Autoregressive Systems," 35th Chinese Control Conference, July 27-29, 2016, Chengdu, China.
- [3] F. Fraccaroli, A. Peruffo, and M. Zorzi, "A New Recursive Least-Squares Method with Multiple Forgetting Schemes," *arXiv*, pp.1503.07338v1 [math.OA], 2015.
- [4] F.Ding, YanjiaoWang and J.Ding "Recursive least squares parameter identification algorithms for systems with colored noise using the filtering technique and the auxiliary model," *Digital Signal Processing*, 2014.
- [5] W. Wang, F. Ding and J.Dai "Maximum likelihood least squares identification for systems with autoregressive moving average noise," *Applied Mathematical Modelling*, pp.1842–1853, 2012.
- [6] M. A. Arwin and N. A. Shashoa, "Extended Three-Stage Recursive Least Squares Identification Algorithm for multiple-input single-output CARARMA Systems," 2021 IEEE International IOT, Electronics and Mechatronics Conference (IEMTRONICS). Canada, April, 2021.
- [7] Y. Guo, L. Wan, L. Xu, F. Ding, A. Alsaedi, and T. Hayat "Two-stage Recursive Least Squares Parameter Estimation Algorithm for Multivariate Output-error Autoregressive Moving Average Systems," *International Journal of Control, Automation and Systems*. pp. 1547-1557, 2019.
- [8] H. Xu¹, F. Ding, M.Gan and E Yang "Two-stage recursive identification algorithms for a class of nonlinear time series models with colored noise," *Int J Robust Nonlinear Control*, 2020.
- [9] H. Duan, J. Jia, M.Gan and R. Ding "Two-stage recursive least squares parameter estimation algorithm for output error models," *Mathematical and Computer Modelling*, 2011.
- [10] F. Chen and F. Ding, "Recursive Least Squares Identification Algorithm for Multiple-Input Nonlinear Box-Jenkins Systems Using Maximum Likelihood Principle", *Journal of Computational and Nonlinear Dynamics* Vol. 11, 2016.
- [11] D. Meng and F. Ding, "Model Equivalence-Based Identification Algorithm for Equation-Error Systems with Colored Noise", *Algorithms*. pp. 280-291, 2015.
- [12] S. Dhar, P. Chaudhuri and B. Chakraborty, Multivariate quantile-quantile plots and related tests using spatial quantiles. *Theoretical Statistics and Mathematics Unit, Indian Statistical Institute, Calcutta and School of Mathematics, University of Birmingham*.
- [13] H. Støvring, Q-Q plots, Biostatistics Department of Public Health, Aarhus University, spring 2010.
- [14] B.Baoa, Y. Xua, J. Shengb and R. Ding "Least squares based iterative parameter estimation algorithm for multivariable controlled ARMA system modelling with finite measurement data," *Mathematical and Computer Modelling*, 2011.

Consensus on the Auxiliary Variables in Distributed Gradient-Based Temporal Difference Algorithms

Miloš S. Stanković, Marko Beko, Nemanja Ilić and Srdjan S. Stanković

Abstract—In this paper we discuss important properties of two novel distributed algorithms for iterative multi-agent off-policy learning of linear value function approximation in Markov Decision Processes (MDP). The algorithms are derived using the off-policy Gradient Reinforcement Learning (GRL) methodology, together with linear dynamic consensus iterations over an underlying inter-agent communication network represented by directed graphs. The proposed algorithms are entirely decentralized, offering new possibilities for choosing different behavior policies while evaluating one single target policy. The presented algorithms formally differ only in the way of applying consensus iterations to the so-called auxiliary variables. The presented proof of weak convergence of both algorithms represents a firm basis for deriving relevant conclusions concerning the role of the consensus iterations. It is shown that the algorithm utilizing consensus on the auxiliary variables shows slightly inferior asymptotic properties, but can provide a higher convergence rate. The figure of merit of each of the algorithms is presented and discussed using the theoretical results obtained under generally nonrestrictive assumptions.

I. INTRODUCTION

Decentralized multi-agent decision making algorithms have recently gained much popularity due to their high effectiveness in dealing with uncertain and dynamic environments typical for the emerging areas of Cyber-Physical Systems (CPS) and Internet of Things (IoT). Numerous distributed estimation, optimization and adaptation methods have been successfully developed using recursive collaborations aimed at achieving a consensus on variables of interest (e.g. [1]–[15] and references therein).

Reinforcement learning (RL) is a general methodology for decision making in uncertain environments based on models in the form of Markov Decision Process (MDP) based on utilization of approximate dynamic programming [16], [17]. A very important issue in this domain is the problem of approximation of the value function under very large state space and the presence of a discrepancy between the behavior policy of an agent and a policy that is currently targeted for evaluation (off-policy learning, e.g. [18]). Recently, in [19]–[22] several

fast gradient-based algorithms for temporal-difference (TD) learning have been proposed. Distributed and multi-agent RL methods have become very popular very recently (see, e.g. [1], [23]–[25] and references therein). Different setups have been adopted in a number of recent works [26]–[32].

In this paper we shall present and discuss two distributed algorithms for *iterative multi-agent off-policy learning* of linear approximation of the value function in MDPs [1]. The algorithms represent generalizations of the recently proposed single agent off-policy gradient algorithm GTD2(λ) [1], [19]–[21], incorporating a distributed consensus scheme operating over a network of typically sparsely connected agents. Another important property of the algorithms is that the local recursions of each agent can be based on eligibility traces [20], [21], where each agent may choose different λ parameters. We provide a firm theoretical background in the form of a proof that the parameter estimates *weakly converge* to consensus points [1], [19], [28], [29], [31], [33], under nonrestrictive connectivity assumption on the topology of the underlying digraph and on the state-visiting distributions of the agents (their behavior policies). The main focus of this paper is placed on the dilemma whether or not to apply consensus to the *auxiliary variables* in the DGTD2(λ)-type algorithms with one-time-scale (see [19], [28], [31]). Notice that the paper [33] deals with the basically two-time-scale algorithms of DGTDC(λ)-type. The given analysis will be exclusively concerned with the limit points of the mean asymptotic ODEs: in this sense the behavior of the estimates for large, but finite t , including the derivation of the corresponding ODEs, can be found in [1], [33]. The limit sets are analyzed by formulating appropriate Lyapunov functions, following the line of thought of [21]. A discussion on role of convexification of the auxiliary variables is provided apart, showing that the two algorithms converge to the same limit points only in special cases. Application of consensus to the auxiliary variables causes, in principle, inferior asymptotic performance, having in mind that the implicitly imposed constraint increases the achievable estimation error. On the other hand, introduction of consensus can contribute to the overall convergence rate at the global level; however, the global convergence rate depends largely on the network connectivity.

The paper is organized as follows. In Section II we formulate the problem and define the algorithms. In Section III a rigorous weak convergence analysis is presented focused on the limit points, while Section IV is devoted to a general discussion on the application of consensus to the auxiliary variables in the DGTD2(λ) algorithms.

M. S. Stanković is with Singidunum University, Belgrade, Serbia; and Vlatacom Institute, Belgrade, Serbia; e-mail: milstank@gmail.com.

M. Beko is with Instituto de Telecomunicações, Instituto Superior Técnico, Universidade de Lisboa, Lisbon, Portugal; and Faculty of Information Technology and Engineering, University Union Nikola Tesla, Belgrade, Serbia; e-mail: beko.marko@gmail.com.

N. Ilić is with College of Applied Technical Sciences, Kruševac, Serbia; and Vlatacom Institute, Belgrade, Serbia; e-mail: nemiliexp@yahoo.com.

S. S. Stanković is with School of Electrical Engineering, University of Belgrade, Serbia; e-mail: stankovic@etf.rs.

This research was supported by the Science Fund of the Republic of Serbia, Grant #6524745, AI-DECIDE.

II. DISTRIBUTED GRADIENT TEMPORAL DIFFERENCE ALGORITHMS

A. Problem Formulation. Definition of the Algorithm

Consider N autonomous agents learning linear approximation to the state value function for a given policy in an MDP (denoted as MDP⁽⁰⁾), using observations of sample transitions in additional N independent MDPs, denoted as MDP⁽ⁱ⁾, $i = 1, \dots, N$. Assume a finite state space $\mathcal{S} = \{1, \dots, M\}$, and that MDP⁽⁰⁾ has the transition matrix P , and MDP⁽ⁱ⁾ the transition matrices $P^{(i)}$, $i = 1, \dots, N$; these chains are induced by π and $\pi^{(i)}$, and referred to as the *target policy* and *behavior policies*, respectively. We are, therefore, dealing with a *cooperative off-policy learning* problem [1], [16], [18], [31].

We introduce the one-stage reward function $r_\pi : \mathcal{S} \rightarrow \mathcal{R}$, specifying the expected reward at each state $s \in \mathcal{S}$, where \mathcal{R} is the set of real numbers [16], [21]. The associated discounted total reward criterion (value function), with the state dependent discount factors $\gamma(s) \in [0, 1]$, $s \in \mathcal{S}$, is given by

$$v_\pi(s) = E_s^\pi \{ r_\pi(S_0) + \sum_{n=1}^{\infty} \gamma(S_1)\gamma(S_2) \cdots \gamma(S_n) \cdot r_\pi(S_n) \}, \quad (1)$$

where $E_s^\pi \{\cdot\}$ indicates the expectation w.r.t. to the Markov chain $\{S_n\}_{n>0}$ induced by π , with the initial state $S_0 = s$. Denote by Γ the $M \times M$ diagonal matrix with $\gamma(s)$ as diagonal entries and $v_\pi = [v_\pi(s_1) \cdots v_\pi(s_M)]^T$.

We assume the following [21]:

(A1) a) P is such that $I - P\Gamma$ is nonsingular; b) $P^{(i)}$ is irreducible and for all $s, s' \in \mathcal{S}$ $P_{ss'}^{(i)} = 0 \Rightarrow P_{ss'} = 0$, $i = 1, \dots, N$.

By the MDP theory [1], [16], [21], [34], v_π uniquely satisfies the *Bellman equation* $v_\pi = r_\pi + P\Gamma v_\pi$ (see e.g. [21], [34]). Within the framework of the *temporal-difference* (TD) algorithms, it is usual to consider the Bellman equation depending on the so-called λ -parameters, procedurally introduced by the so-called *eligibility traces*. In this sense, $v_\pi = T^{(\lambda)} v_\pi$ is considered as a *generalized Bellman equation*, where $T^{(\lambda)} v = r_\pi^\lambda + P^{(\lambda)} v$, $\forall v \in R^{|\mathcal{S}|}$, is the *generalized Bellman operator* for a vector $r_\pi^{(\lambda)}$ and a substochastic matrix $P^{(\lambda)}$ [16], [21].

Let $\phi : \mathcal{S} \rightarrow R^p$ be a function that maps each state to a p -dimensional feature vector $\phi = [\phi_1 \cdots \phi_p]^T$; let the subspace spanned by feature vectors ϕ be \mathcal{L}_ϕ . In general, TD algorithms look for some function $v \in \mathcal{L}_\phi$ that satisfies $v \approx T^{(\lambda)} v$. We assume that the approximation functions are parameterized as $v(s) = \phi(s)^T \theta$, $s \in \mathcal{S}$ using parameters $\theta \in R^p$, so that the algorithms learn the vector θ . If we define the $M \times p$ matrix Φ as a matrix composed of p -vectors $\phi(s)$ as row vectors, we have $v_\theta = \Phi \theta$.

In order to construct a distributed algorithm for finding an approximation function $v_\theta \in \mathcal{L}_\phi$ by using observations from MDP⁽ⁱ⁾, $i = 1, \dots, N$, we define the following *global objective function*

$$J(\theta) = \sum_{i=1}^N q_i J_i(\theta) = \frac{1}{2} \sum_{i=1}^N q_i \|\Pi_{\xi_i}(T^{(\lambda_i)} v_\theta - v_\theta)\|_{\xi_i}^2, \quad (2)$$

where $J_i(\theta)$ are the *local objective functions*, $q_i > 0$ the *a priori* defined weighting coefficients, λ_i is the local λ -parameter and Π_{ξ_i} denotes the projection onto the subspace \mathcal{L}_ϕ w.r.t. the weighted Euclidean norm $\|v\|_{\xi_i}^2 = \sum_{s \in \mathcal{S}} \xi_{i,s} v(s)^2$ for a positive M -dimensional vector ξ_i with components $\xi_{i,s}$ (see [21], [31]). In accordance with [21], [34], we take ξ_i to be the invariant probability distribution for the local Markov chain MDP⁽ⁱ⁾, with the transition matrix $P^{(i)}$ induced by $\pi^{(i)}$ ($\xi_i^T P^{(i)} = \xi_i^T$). It follows that

$$\nabla J(\theta) = \sum_{i=1}^N q_i (\Phi^T \Xi_i (P^{(\lambda_i)} - I) \Phi)^T w_i(\theta), \quad (3)$$

where Ξ_i is an $M \times M$ diagonal matrix with the components of ξ_i on the diagonal, and $w_i(\theta)$ the unique solution (in w_i) of the equation

$$\Phi w_i = \Pi_{\xi_i}(T^{(\lambda_i)} v_\theta - v_\theta), \quad (4)$$

assuming that $w_i \in \text{span}\{\phi(S)\}$.

In the off-policy scenario, we introduce the local *importance sampling ratios* $\rho_i(s, s') = P_{ss'}^i / P_{ss'}$ for $s, s' \in \mathcal{S}$, $i = 1, \dots, N$; denote $\rho_i(n) = \rho_i(S_n, S_{n+1})$, as well as $\gamma(n) = \gamma(S_n)$ [21], [34]. The local *temporal-difference term* is defined by

$$\delta_i(v_\theta; n) = \rho_i(n)(R(n+1) + \gamma(n+1)v_\theta(S_{n+1}) - v_\theta(S_n)). \quad (5)$$

The local *eligibility trace vectors* $\{e_i(n)\}$ are generated by

$$e_i(n) = \lambda_i(n)\gamma(n)\rho_i(n-1)e_i(n-1) + \phi(S_n), \quad (6)$$

where $\lambda_i(n) \in [0, 1]$ are the local λ -parameters, $i = 1, \dots, N$ [21], [34].

The distributed algorithms for learning linear approximation to the state value function for a given policy π we are going to analyze consist of two main parts: 1) *local parameter updates* based on the *gradient descent* methodology using local state transition observations from MDP⁽ⁱ⁾, and 2) interchange of the current parameter estimates aimed at achieving *consensus* between the agents. The local parameter updates are defined by

$$\theta'_i(n) = \theta_i(n) + \alpha_i(n)q_i\rho_i(n)(\phi(S_n) - \gamma(n+1)\phi(S_{n+1}))e_i(n)^T w_i(n) \quad (7)$$

$$w'_i(n) = w_i(n) + \beta_i(n)(e_i(n)\delta_i(v_{\theta_i(n)}; n) - \phi(S_n)\phi(S_n)^T w_i(n)) \quad (8)$$

where $v_{\theta_i(n)} = \Phi\theta_i(n)$; $\theta_i(0)$ is chosen arbitrarily, while for $w_i(0)$ and $e_i(0)$ we have $w_i(0), e_i(0) \in \text{span}\{\phi(S)\}$ [21]. Notice that the algorithm incorporates the *auxiliary variables* $w_i(n)$ and $w'_i(n)$; their role is essential for this paper [19], [21].

The second, communication part of the algorithm performs the following convexification w.r.t. the approximation parameter θ , leaving local auxiliary parameters unchanged, *i.e.*,

$$\theta_i(n+1) = \sum_{j=1}^N a_{ij}(n)\theta'_j(n), \quad w_i(n+1) = w'_i(n), \quad (9)$$

where $a_{ij}(n)$ are random variables, elements of a random matrix $A(n) = [a_{ij}(n)]$ [11], [31], [35]. If one adopts that the

available N MDP's are connected by communication links in accordance with a directed graph $\mathcal{G} = (\mathcal{N}, \mathcal{E})$, where \mathcal{N} is the set of nodes and \mathcal{E} the set of arcs, then matrix $A(n)$ has zeros at the same places as the graph adjacency matrix A_G and is *row-stochastic*, i.e., $\sum_{j=1}^N a_{ij}(n) = 1, i = 1, \dots, N, \forall n \geq 0$. The algorithm (7), (8) incorporating consensus only w.r.t. θ according to (9) will be denoted as AlgA.

We also consider a modification of AlgA, denoted as AlgB, obtained by applying convexification to both θ_i and $w_i, i = 1, \dots, N$, in such a way that the second relation in (9) becomes

$$w_i(n+1) = \sum_{j=1}^N a_{ij}(n)w'_j(n). \quad (10)$$

A comparative analysis of AlgA and AlgB is in the main focus of this paper.

III. CONVERGENCE ANALYSIS

A. Prerequisites

1) *Choice of λ -parameters*: The results given below are applicable to both *state-dependent* and *history-dependent* λ_i . In the first case, we have simply $\lambda_i(n) = \lambda_i(S_n)$ for a given function $\lambda_i : \mathcal{S} \rightarrow [0, 1]$, while in the second case $\lambda_i(n) = \lambda(y_i(n), e_i(n-1))$, $y_i(n) = f(y_i(n-1), S_n)$, where $y_i(n), n \geq 0$, is a memory state summarizing the history of the past states up to time n (see [21], [34]).

Different choices of λ_i lead to different generalized Bellman operators. For example, in the case of state-dependent λ_i , we have:

$$T^{(\lambda_i)}v = (I - P\Gamma\Lambda_i)^{-1}r_\pi + (I - P\Gamma\Lambda_i)^{-1}P\Gamma(I - \Lambda_i)v, \quad (11)$$

where Λ_i is an $M \times M$ diagonal matrix with entries $\lambda_i(s)$; therefore, we have $P^{(\lambda_i)} = (I - P\Gamma\Lambda_i)^{-1}P\Gamma(I - \Lambda_i)$. For the history-dependent λ_i -parameters, the formulation is more complex [21], [34] (the details are out of the scope of this paper).

2) *Properties of the State-Trace Processes*: Under the behavior policies $\pi^{(i)}$, the *state-trace processes* are defined as $\{S_n, e_i(n)\}$. These state-trace processes are Markov chains with the weak Feller property [21], [34]. Let $Z_i(n) = (S_n, e_i(n), S_{n+1})$. We have the following important result [21], [34]: a) the state-trace weak Feller-Markov chain process $Z_i(n)$ has a unique invariant probability measure ζ_i ; for each initial condition the *occupation probability measure* converges weakly to ζ_i [21, Theorem 2.1(i)]; b) if E_{ζ_i} denotes the expectation of the stationary state-trace process with initial distribution ζ_i , then $E_{\zeta_i}\{\|f(Z_0)\|\} < \infty$ and $\frac{1}{n}\sum_{j=0}^{n-1} f(Z_i(j))$ converges to $E_{\zeta_i}\{f(Z_i(0))\}$ in mean and a.s., where $f(z)$ is a Lipschitz continuous function in the trace variable e [21, Theorem 2.1(ii)].

The results a) and b) are used to prove the following:

- 1) $E_{\zeta_i}\{\phi(S_0)\phi(S_0)^T\} = \Phi^T \Xi_i \Phi$;
- 2) $E_{\zeta_i}\{e_i(0)\delta_i(v; 0)\} = \Phi^T \Xi_i (T^{(\lambda_i)}v - v), \forall v \in \mathcal{R}^M$
- 3) $E_{\zeta_i}\{e_i(0)\rho_i(0)(\phi^j(S_0) - \gamma(1)\phi^j(S_1))\} = \Phi^T \Xi_i (I - P^{(\lambda_i)})\Phi_j, 1 \leq j \leq p$;
- 4) $E_{\zeta_i}\{e_i(0)\rho_i(0)(1 - \lambda_i(1))\gamma(1)\phi^j(S_1)\} = \Phi^T \Xi_i P^{(\lambda_i)}\Phi_j, 1 \leq j \leq p$;

where $\phi^j(\cdot)$ is the j -th component of $\phi(\cdot)$ and Φ_j the j -th column vector of Φ [21, Proposition 2.1].

Under (A1), the results from [21, Proposition 2.2] also show that the sequences of traces $\{e_i(n)\}$ satisfy the condition $E\{\|e_i(n) - \hat{e}_i(n)\|\} \leq c(n)$, where $c(n) \rightarrow 0$ when $n \rightarrow \infty$, while $\{e_i(n)\}$ and $\{\hat{e}_i(n)\}$ are obtained using the same trajectory of states, but with different initial conditions $e_i(0)$ and $\hat{e}_i(0)$. Also, $\{e_i(n)\}$ is uniformly integrable, and, consequently, the random variables $\{Z_i(n)\}, n \geq 0$, are tight [36].

Let

$$g_i(\theta_i, w_i, Z_i) = \rho_i(s, s')(\phi(s) - \gamma(s')\phi(s'))e_i^T w_i \quad (12)$$

and

$$k_i(\theta_i, w_i, Z_i) = e_i \bar{\delta}_i(s, s', v_{\theta_i}) - \phi(s)\phi(s)^T w_i, \quad (13)$$

where $\bar{\delta}_i(s, s', v_{\theta_i}) = \rho_i(s, s')(r(s, s') + \gamma(s')v_{\theta_i}(s') - v_{\theta_i}(s))$. We also have:

$$\begin{aligned} \bar{g}_i(\theta_i, w_i) &= E_{\zeta_i}\{g_i(\theta_i, w_i, Z_i(0))\} \\ &= (\Phi^T \Xi_i (I - P^{(\lambda_i)}) \Phi)^T w_i, \end{aligned} \quad (14)$$

$$\begin{aligned} \bar{k}_i(\theta_i, w_i) &= E_{\zeta_i}\{k_i(\theta_i, w_i, Z_i(0))\} \\ &= \Phi^T \Xi_i (T^{(\lambda_i)} v_{\theta_i} - v_{\theta_i}) - \Phi^T \Xi_i \Phi w_i, \end{aligned} \quad (15)$$

and

$$\bar{g}_i(\theta_i, w_i(\theta_i)) = (\Phi^T \Xi_i (I - P^{(\lambda_i)}) \Phi)^T w_i(\theta_i). \quad (16)$$

Comparison with (3) shows that $\bar{g}_i(\theta_i, w_i(\theta_i)) = -\nabla J_i(\theta_i)$.

Based on the above definitions and the results from [21], we have the following important ergodic properties:

Lemma 1 ([21]): Under (A1), the following holds for each θ_i and w_i and each compact set $D_i \subset \mathcal{Z}_i$:

- a) $\lim_{m, n \rightarrow \infty} \frac{1}{m} \sum_{s=n}^{n+m-1} E_n\{k_i(\theta_i, w_i, Z_i(s+1)) - \bar{k}_i(\theta_i, w_i)\} I(Z_i(n) \in D_i) = 0$ in mean,
- b) $\lim_{m, n \rightarrow \infty} \frac{1}{m} \sum_{s=n}^{n+m-1} E_n\{g_i(\theta_i, w_i, Z_i(s+1)) - \bar{g}_i(\theta_i, w_i)\} I(Z_i(n) \in D_i) = 0$ in mean,
- c) $\lim_{m, n \rightarrow \infty} \frac{1}{m} \sum_{s=n}^{n+m-1} E_n\{g_i(\theta_i, w_i(\theta_i), Z_i(s+1)) - \bar{g}_i(\theta_i, w_i(\theta_i))\} I(Z_i(n) \in D_i) = 0$ in mean,

where $E_n\{\cdot\}$ denotes the conditional expectation given the history $(Z_i(0), \dots, Z_i(n))$ and $I(\cdot)$ denotes the indicator function.

B. Global Model

Let $X(n) = [\Theta(n)^T:W(n)^T]^T, \Theta(n) = [\theta_1(n)^T \dots \theta_N(n)^T]^T, W(n) = [w_1(n)^T \dots w_N(n)^T]^T$;

similarly, $X'(n) = [\Theta'(n)^T:W'(n)^T]^T$, together with the corresponding vector components. Then, we have for AlgA the following global model at the network level

$$\begin{aligned} X'(n) &= X(n) + \Gamma(n)F(X(n), n), \\ X(n+1) &= \text{diag}\{(A(n) \otimes I_p), I_{Np}\}X'(n), \end{aligned} \quad (17)$$

where \otimes denotes the Kronecker's product, while $\Gamma(n) = \text{diag}\{\alpha_1(n), \dots, \alpha_N(n), \beta_1(n), \dots, \beta_N(n)\} \otimes I_p$,

$$F(X(n), n) = [F^\theta(X(n), n)^T: F^w(X(n), n)^T]^T,$$

$$\begin{aligned} F^\theta(X(n), n) &= [q_1 g_1(\theta_1(n), w_1(n), Z_1(n))^T \dots \\ &\quad q_N g_N(\theta_N(n), w_N(n), Z_N(n))^T]^T, \end{aligned}$$

$$\begin{aligned}
 F^w(X(n), n) &= [k_1(\theta_1(n), w_1(n), Z_1(n))^T \\
 &\quad + e_1(n)^T \omega_1(n+1) \cdots \\
 &\quad k_N(\theta_N(n), w_N(n), Z_N(n)) \\
 &\quad + e_N(n)^T \omega_N(n+1)]^T
 \end{aligned}$$

with $g_i(\cdot)$ defined by (12).

Introduce dummy variables $X = [\Theta^T; W^T]^T$, together with $\bar{F}(X) = [\bar{F}^\theta(\Theta, W)^T; \bar{F}^w(\Theta, W)^T]^T$, $F^\theta(\Theta, W) = [q_1 \bar{g}_1(\theta_1, w_1)^T \cdots q_N \bar{g}_N(\theta_N, w_N)^T]^T$, with $\bar{g}_i(\cdot, \cdot)$ defined by (14), $F^w(\Theta, W) = [\bar{k}_1(\theta_1, w_1)^T \cdots \bar{k}_N(\theta_N, w_N)^T]^T$, with $\bar{k}_i(\cdot, \cdot)$ defined by (15).

In the case of AlgB, we have a slightly modified model (17): instead of $\text{diag}\{(A(n) \otimes I_p), I_{Np}\}$ in the second relation in (17), we have $\text{diag}\{(A(n) \otimes I_p), (A(n) \otimes I_p)\}$, as a consequence of consensus w.r.t. w_i .

C. Communication Part of the Algorithm

The result of this subsection is a slight generalization of the results in [5], based on [11].

Define $\Psi(n|k) = A(n) \cdots A(k)$ for $n \geq k$, $\Psi(n|n+1) = I_N$. Let \mathcal{F}_n be an increasing sequence of σ -algebras such that \mathcal{F}_n measures $\{X(k), k \leq n, A(k), k < n\}$.

(A2) There is a scalar $\alpha_0 > 0$ such that $a_{ii}(n) \geq \alpha_0$, and, for $i \neq j$, either $a_{ij}(n) = 0$ or $a_{ij}(n) \geq \alpha_0$.

(A3) There are a scalar $p_0 > 0$ and an integer n_0 such that $P_{\mathcal{F}_n}\{\text{agent } j \text{ communicates to agent } i \text{ on the interval } [n, n+n_0]\} \geq p_0$, for all n and $i = 1, \dots, N, j \in \mathcal{N}_i$.

(A4) The digraph \mathcal{G} is strongly connected.

According to [5], [11], it is possible to show that (A2)-(A4) imply that $\Psi(k) = \lim_n \Psi(n|k)$ exists w.p.1; moreover, its rows are equal and $E\{|\Psi(n|k) - \Psi(k)|\}$, $E_{\mathcal{F}_n}\{|\Psi(n|k) - \Psi(k)|\} \rightarrow 0$ geometrically as $n - k \rightarrow \infty$, uniformly in k and ω (w.p.1). In addition, $E_{\mathcal{F}_n}\{\Psi(n|k)\}$ converges to $\Psi(k)$ geometrically, uniformly in ω and k , as $n \rightarrow \infty$.

D. Convergence Proofs

(A5) Sequence $\{A(n)\}$ is independent of the processes in MDP i , $i = 1, \dots, N$.

(A6) There is a $N \times N$ matrix $\bar{\Psi}$ such that $E\{|E_k\{\Psi(n)\} - \bar{\Psi}|\} \rightarrow 0$ as $n - k \rightarrow \infty$, which, under the conditions of Lemma 1, has the form

$$\bar{\Psi} = \begin{bmatrix} \bar{\psi}_1 & \cdots & \bar{\psi}_N \\ \bar{\psi}_1 & \cdots & \bar{\psi}_N \\ \vdots & & \vdots \\ \bar{\psi}_1 & \cdots & \bar{\psi}_N \end{bmatrix} = \begin{bmatrix} \hat{\Psi} \\ \vdots \\ \hat{\Psi} \end{bmatrix},$$

where $\sum_i \bar{\psi}_i = 1$ ($|\cdot|$ denotes the infinity norm).

(A7) Sequence $\{X(n)\}$ is tight.

1) AlgA):

Theorem 1: Let (A1)–(A7) hold. Let $X^\alpha(n)$ be generated by AlgA, (7), (8), (9), with $\beta_i(n) = \alpha_i(n) = \alpha$ and define for $t \geq 0$, $t \in \mathcal{R}$, $X^\alpha(t) = X(n)$ for $t \in [(n - n_\alpha)\alpha, (n - n_\alpha + 1)\alpha)$. Let $w_i^\alpha(0) = w_{i,0}^\alpha$, $e_i(0) = e_{i,0} \in \text{span}\{\phi(S)\}$. Then, for any integers n'_α such that $\alpha n'_\alpha \rightarrow \infty$ as $\alpha \rightarrow 0$, there exist

positive numbers $\{T_\alpha\}$ with $T_\alpha \rightarrow \infty$ as $\alpha \rightarrow 0$ such that for any $\epsilon > 0$

$$\limsup_{\alpha \rightarrow 0} P\{(X^\alpha(n'_\alpha + k)) \notin N_\epsilon(\bar{\Sigma}) \text{ for some } k \in [0, T_\alpha/\alpha]\} = 0, \quad (18)$$

$i = 1, \dots, N$, where $N_\epsilon(\cdot)$ denotes the ϵ -neighborhood, while $\bar{\Sigma} = \bar{\Sigma}_\theta \times \cdots \times \bar{\Sigma}_\theta \times \bar{\Sigma}_{w_1} \times \cdots \times \bar{\Sigma}_{w_N}$ is the set of points $\bar{\theta}, \dots, \bar{\theta}, \bar{w}_1, \dots, \bar{w}_N$ satisfying

$$\begin{aligned}
 \sum_{i=1}^N \bar{\psi}_i q_i G_i^T \bar{w}_i &= 0, \\
 G_1 \bar{\theta} + b_1 - H_1 \bar{w}_1 &= 0, \\
 &\vdots \\
 G_N \bar{\theta} + b_N - H_N \bar{w}_N &= 0,
 \end{aligned} \quad (19)$$

where $G_i = \Phi^T \Xi_i (P^{(\lambda_i)} - I) \Phi$, $b_i = \Phi^T \Xi_i r_\pi^{(\lambda_i)}$, $r_\pi^{(\lambda_i)}$ is a constant M -vector in the affine function $T^{(\lambda_i)}(\cdot)$, while $H_i = \Phi^T \Xi_i \Phi$, $i = 1, \dots, N$.

Proof: The proof is based on [1], [33] and the general results from [5], [11], [36]. In order to apply the proof of Theorem 3.1 in [5], it is essential to verify whether the basic assumptions from [5] concerning $F(X(n), n)$ hold in our case. We can easily conclude that Lemma 1 implies that the assumptions (C3.2) and C(3.3') from Section 3 in [5] hold. Following further [5], it follows that the Skorokhod embedding implies that we have the limit process $X^\alpha(\cdot) \rightarrow X(\cdot)$, where $\dot{X} = \text{diag}\{\bar{\Psi} \otimes I_p, I_{Np}\} \bar{F}(X)$ [5]. By Lemma 1 and (A6), all the rows of $\bar{\Psi}$ are equal. Consequently, $\Theta(\cdot) = [\theta(\cdot)^T \cdots \theta(\cdot)^T]^T$, $\forall \theta(\cdot) \in \mathcal{R}^p$, implying that $\theta = \bar{\psi}_1 q_1 \bar{g}_1(\theta, w_1) + \cdots + \bar{\psi}_N q_N \bar{g}_N(\theta, w_N)$; we also have $\dot{w}_1 = \bar{k}_1(\theta, w_1), \dots, \dot{w}_N = \bar{k}_N(\theta, w_N)$, having in mind that consensus is not applied to the auxiliary variables.

In order to prove (18), we study the limit set

$$\begin{aligned}
 E &= \bigcap_{\tau \geq 0} \text{cl}\{\theta(t), w_1(t), \dots, w_N(t) | \theta(0), w_1(0), \dots, w_N(0) \\
 &\quad \in \mathcal{R}^{(N+1)p}, t \geq \tau\}.
 \end{aligned} \quad (20)$$

where $\text{cl}\{\cdot\}$ denotes the closure of a given set. Following [21] (Proposition 4.1), we introduce the Lyapunov function

$$V(\theta, w_1, \dots, w_N) = \frac{1}{2} \|\theta - \bar{\theta}\|^2 + \frac{1}{2} \sum_{i=1}^N q_i \bar{\psi}_i \|w_i - \bar{w}_i\|^2, \quad (21)$$

where $\bar{\theta}$ and \bar{w}_i are given by (19). We have directly

$$\begin{aligned}
 \dot{V}(\theta, w_1, \dots, w_N) &= \langle \theta - \bar{\theta}, - \sum_{i=1}^N q_i \bar{\psi}_i G_i^T w_i \rangle \\
 &\quad + \sum_{i=1}^N q_i \bar{\psi}_i \langle w_i - \bar{w}_i, G_i \theta + \bar{g}_i - \bar{H}_i w_i \rangle \\
 &= - \sum_{i=1}^N q_i \bar{\psi}_i \langle w_i - \bar{w}_i, H_i (w_i - \bar{w}_i) \rangle.
 \end{aligned} \quad (22)$$

Therefore, $\dot{V}(\theta, w_1, \dots, w_N) < 0$ for $w_i \in \text{span}\{\phi(S)\}$ and $w_i \neq \bar{w}_i$, implying that $\hat{w}_i = \bar{w}_i$ if $[\hat{\theta}^T \hat{w}_1^T \cdots \hat{w}_N^T]^T \in$

E and $\hat{w}_i \in \text{span}\{\phi(S)\}$, $i = 1, \dots, N$. Similarly, if $[\hat{\theta}^T \bar{w}_1^T \dots \bar{w}_N^T]^T \in E$, then $\hat{\theta} = \bar{\theta}$. In such a way we conclude that for initial conditions $w_i(0) \in \text{span}\{\phi(S)\}$ the limit set E is indeed the set $\bar{\Sigma}$ of points satisfying (19).

The steps remaining to prove (18) are standard for the stochastic approximation theory (see [1], [21], [36]). ■

2) AlgB):

Theorem 2: Let (A1)–(A7) hold. Let $X^\alpha(n)$ be generated by AlgB (7), (8), (9) and (10), with $\beta_i(n) = \alpha_i(n) = \alpha$, and let both $w_i^\alpha(0) = w_{i,0}^\alpha$ and $e_i(0) = e_{i,0} \in \text{span}\{\phi(S)\}$. Then, for any integers n'_α such that $\alpha n'_\alpha \rightarrow \infty$ as $\alpha \rightarrow 0$, there exist positive numbers $\{T_\alpha\}$ with $T_\alpha \rightarrow \infty$ as $\alpha \rightarrow 0$ such that for any $\epsilon > 0$

$$\limsup_{\alpha \rightarrow 0} P\left\{ \begin{array}{l} \theta_i^\alpha(n'_\alpha + k) \\ w_i^\alpha(n'_\alpha + k) \end{array} \right\} \notin N_\epsilon(\bar{\Sigma}) \quad \text{for some } k \in [0, T_\alpha/\alpha] = 0, \quad (23)$$

$i = 1, \dots, N$, where $N_\epsilon(\cdot)$ denotes the ϵ -neighborhood, while $\bar{\Sigma} = \bar{\Sigma}_\theta \times \bar{\Sigma}_w$ is the set of points $\bar{x} = [\bar{\theta}^T \bar{w}^T]^T \in \mathcal{R}^{2p}$ satisfying

$$\bar{G}\bar{\theta} + \bar{g} - \bar{H}\bar{w} = 0, \quad \bar{G}^T \bar{w} = 0, \quad (24)$$

where $\bar{G} = \sum_{i=1}^N \bar{\psi}_i q_i \Phi^T \Xi_i (P^{(\lambda_i)} - I) \Phi$, $\bar{b} = \Phi^T \sum_{i=1}^N \bar{\psi}_i q_i \Xi_i r_\pi^{(\lambda_i)}$, $r_\pi^{(\lambda_i)}$ is a constant M -vector in the affine function $T^{(\lambda_i)}(\cdot)$, while $\bar{H} = \sum_{i=1}^N \bar{\psi}_i q_i \Phi^T \Xi_i \Phi$.

Proof: AlgA differs from AlgB only in the communication part of the algorithm. Formally, the procedure of the proof remains the same as in Theorem 1, after replacing $\text{diag}\{(A(n) \otimes I_p), I_{Np}\}$ by $\text{diag}\{(A(n) \otimes I_p), (A(n) \otimes I_p)\}$. This implies that, asymptotically, instead of $\text{diag}\{(\hat{\Psi} \otimes I_p), I_{Np}\}$ we have now $\text{diag}\{(\hat{\Psi} \otimes I_p), (\hat{\Psi} \otimes I_p)\}$. In this sense, we obtain $X(\cdot) = [\theta(\cdot)^T \dots \theta(\cdot)^T w(\cdot)^T \dots w(\cdot)^T]^T$, where $\theta(\cdot)$ and $w(\cdot)$ satisfy the following ODE:

$$\begin{bmatrix} \dot{\theta} \\ \dot{w} \end{bmatrix} = \bar{\psi}_1 q_1 \begin{bmatrix} \bar{g}_1(\theta, w) \\ \bar{k}_1(\theta, w) \end{bmatrix} + \dots + \bar{\psi}_N q_N \begin{bmatrix} \bar{g}_N(\theta, w) \\ \bar{k}_N(\theta, w) \end{bmatrix} \quad (25)$$

The limit points (24) follow from (25), according to Theorem 1. Namely, we define the Lyapunov function

$$V(\theta, w_1, \dots, w_N) = \frac{1}{2} \|\theta - \bar{\theta}\|^2 + \frac{1}{2} \|w - \bar{w}\|^2, \quad (26)$$

where $\bar{\theta}$ and \bar{w} are given by (24) and obtain for the derivative that

$$\begin{aligned} \dot{V}(\theta, w) &= \langle \theta - \bar{\theta}, \bar{G}^T w \rangle + \langle w - \bar{w}, \bar{G}\theta + \bar{b} - \bar{H}w \rangle \\ &= -\langle w - \bar{w}, \bar{H}(w - \bar{w}) \rangle. \end{aligned} \quad (27)$$

Therefore, $\dot{V}(\theta, w) < 0$ for $w_i \in \text{span}\{\phi(S)\}$ and $w \neq \bar{w}$, implying that $\hat{w} = \bar{w}$ if $[\hat{\theta}^T \hat{w}^T]^T \in E$ and $\hat{w} \in \text{span}\{\phi(S)\}$. In the same way, if $[\hat{\theta}^T \bar{w}^T]^T \in E$, then $\hat{\theta} = \bar{\theta}$. Consequently, for the initial conditions $w_i(0) \in \text{span}\{\phi(S)\}$, the limit set of ODE (25) is the set $\bar{\Sigma}$ satisfying (24). ■

IV. DISCUSSION

The preceding section has been devoted to the weak convergence of the proposed distributed temporal difference learning algorithms. The role of convexification of w_i remains to be

clarified. It is clear, from the definition of the criterion function (2) and the algorithm construction, that AlgA follows from the basic local relations (4), providing for all i unique solutions $w_i(\theta)$ for all θ . However, AlgB is based on the introduction of an additional constraint that $w_1(\theta) = \dots = w_N(\theta) = w(\theta)$, where $w(\theta)$ is the unique solution of

$$\Phi^T \left(\sum_{i=1}^N \bar{\psi}_i q_i \Xi_i \right) \Phi w(\theta) = \sum_{i=1}^N \bar{\psi}_i q_i \Pi_{\xi_i} (T^{(\lambda_i)} v_\theta - v_\theta). \quad (28)$$

It is straightforward to observe from (28) that we have for any given θ

$$\Phi^T \left(\sum_{i=1}^N \bar{\psi}_i q_i \Xi_i \right) \Phi w(\theta) = \sum_{i=1}^N \bar{\psi}_i q_i \Phi \Xi_i \Phi w_i(\theta). \quad (29)$$

This property is verified by the above theorems.

Consequently, convergence points of θ are different for these two cases. In order to get a clearer insight, assume that $\det H_i \neq 0$. Then, for AlgA we have

$$\sum_{i=1}^N \bar{\psi}_i q_i G_i^T w_i(\theta) = \sum_{i=1}^N \bar{\psi}_i q_i G_i^T [H_i^{-1}(G_i \theta + b_i)] = 0, \quad (30)$$

resulting in

$$\sum_{i=1}^N \bar{\psi}_i q_i G_i^T H_i^{-1} G_i \theta = \sum_{i=1}^N \bar{\psi}_i q_i G_i^T H_i^{-1} b_i = 0, \quad (31)$$

while for AlgB we obtain

$$\bar{G}^T \bar{H}^{-1} \bar{G} \theta = \bar{G}^T \bar{H}^{-1} \bar{b}. \quad (32)$$

Notice that in the case of equal λ -parameters and equal behavior policies for all the agents, both algorithms provide the same solution.

It is difficult to make any general conclusion about the relative advantage of one of the two presented algorithms. It is to be noticed that this issue has not been directly treated in the literature; all the examples of distributed TD algorithms subsume that the consensus operator is applied to both θ and w , without mentioning any alternative (e.g., [28] with the references therein). As far as the limit points of the corresponding ODEs are concerned, it should be noticed that a better approximation could be, in general, expected when consensus is not applied to w . Our experience confirms this statement; however it does not show any significant difference from this point of view. In some cases it could be expected that the application of consensus to w may improve the convergence rate of the algorithm. It is hard to draw any definite conclusion, in general, having in mind that connectedness of the underlying network can play an essential role from this point of view. A comprehensive Monte Carlo analysis could practically resolve the remaining dilemmas. It would be also interesting to analyze the discussed problem in the two-time-scale cases (see [1]).

V. CONCLUSION

In this paper we have presented and discussed two novel algorithms for distributed off-policy gradient-based value function approximation within a collaborative multi-agent reinforcement learning setting. The algorithms are based on an integration of linear dynamic consensus schemes into local gradient-based recursions, involving the so called *auxiliary variables*. We presented rigorous proofs that, under nonrestrictive assumptions, the parameter estimates weakly converge to consensus. Based on these proofs, a discussion is provided of the incorporation of consensus w.r.t. auxiliary variables, defining clearly the figure of merit of the alternative approaches.

REFERENCES

- [1] M. S. Stanković, M. Beko, and S. S. Stanković, "Distributed value function approximation for collaborative multi-agent reinforcement learning," *IEEE Trans. Control Networked Systems*, vol. (Early Access), 2021.
- [2] M. S. Stanković, S. S. Stanković, and D. M. Stipanović, "Consensus-based decentralized real-time identification of large-scale systems," *Automatica*, vol. 60, pp. 219–226, 2015.
- [3] J. N. Tsitsiklis, "Problems in decentralized decision making and computation," Ph.D. dissertation, Dep. Electrical Eng. Comput. Sci., M.I.T., Cambridge, MA, 1984.
- [4] J. N. Tsitsiklis, D. P. Bertsekas, and M. Athans, "Distributed asynchronous deterministic and stochastic gradient optimization algorithms," *IEEE Trans. Autom. Control*, vol. 31, pp. 803–812, 1986.
- [5] H. J. Kushner and G. Yin, "Asymptotic properties of distributed and communicating stochastic approximation algorithms," *SIAM J. Control Optim.*, vol. 25, pp. 1266–1290, 1987.
- [6] A. Nedić and A. Ozdaglar, "Distributed subgradient methods for multi-agent optimization," *IEEE Trans. Autom. Control*, vol. 54, pp. 48–61, 2009.
- [7] P. Bianchi, G. Fort, and W. Hachem, "Performance of a distributed stochastic approximation algorithm," *IEEE Trans. Inf. Theory*, vol. 59, pp. 7405–7418, 2013.
- [8] M. S. Stanković, S. S. Stanković, and K. H. Johansson, "Distributed time synchronization for networks with random delays and measurement noise," *Automatica*, vol. 93, pp. 126 – 137, 2018.
- [9] A. Nedić and A. Olshevsky, "Distributed optimization over time-varying directed graphs," *IEEE Trans. Autom. Control*, vol. 60, pp. 601 – 615, 2015.
- [10] S. Tu and A. Sayed, "Diffusion strategies outperform consensus strategies for distributed estimation over adaptive networks," *IEEE Trans. Signal Process.*, vol. 60, pp. 6217–6233, 2012.
- [11] M. S. Stanković, N. Ilić, and S. S. Stanković, "Distributed stochastic approximation: Weak convergence and network design," *IEEE Trans. Autom. Control*, vol. 61, no. 12, pp. 4069–4074, 2016.
- [12] N. Ilić, S. S. Stanković, M. S. Stanković, and K. H. Johansson, "Consensus based distributed change detection using generalized likelihood ratio methodology," *Signal Processing*, vol. 92, no. 7, pp. 1715 – 1728, 2012.
- [13] M. S. Stanković, S. S. Stanković, K. H. Johansson, M. Beko, and L. M. Camarinha-Matos, "On consensus-based distributed blind calibration of sensor networks," *Sensors*, vol. 18, no. 11, 2018.
- [14] M. S. Stanković, K. H. Johansson, and D. M. Stipanović, "Distributed seeking of Nash equilibria with applications to mobile sensor networks," *IEEE Trans. Autom. Control*, vol. 57, no. 4, pp. 904–919, 2012.
- [15] S. S. Stanković, M. Beko, and M. S. Stanković, "Nonlinear robustified stochastic consensus seeking," *Systems & Control Letters*, vol. 139, p. 104667, 2020.
- [16] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. MIT press Cambridge, 1998.
- [17] D. P. Bertsekas and J. Tsitsiklis, *Neuro-Dynamic Programming*. Athena Scientific, 1996.
- [18] D. Precup, R. S. Sutton, and S. Dasgupta, "Off-policy temporal-difference learning with function approximation," in *Proc. 18th Int. Conf. on Machine Learning*, 2001, pp. 417–424.
- [19] R. S. Sutton, H. R. Maei, D. Precup, S. Bhatnagar, D. Silver, C. Szepesvári, and E. Wiewiora, "Fast gradient-descent methods for temporal-difference learning with linear function approximation," in *Proc. 26th Int. Conf. on Machine Learning*, 2009, pp. 993–1000.
- [20] M. Geist and B. Scherrer, "Off-policy learning with eligibility traces: A survey," *Journal of Machine Learning Research*, vol. 15, pp. 289–333, 2014.
- [21] H. Yu, "On convergence of some gradient-based temporal-differences algorithms for off-policy learning," *arXiv:1712.09652*, 2017.
- [22] B. Dai, A. Shaw, L. Li, L. Xiao, N. He, J. Chen, and L. Song, "SBED: Convergent reinforcement learning with nonlinear function approximation," *arXiv:1712.10285*, 2017.
- [23] L. Busoniu, R. Babuska, and B. De Schutter, "A comprehensive survey of multiagent reinforcement learning," *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, vol. 38, no. 2, pp. 156–172, 2008.
- [24] J. K. Gupta, M. Egorov, and M. Kochenderfer, "Cooperative multi-agent control using deep reinforcement learning," in *Autonomous Agents and Multiagent Systems*, G. Sukthankar and J. A. Rodriguez-Aguilar, Eds. Cham: Springer International Publishing, 2017, pp. 66–83.
- [25] A. OroojlooyJadid and D. Hajinezhad, "A review of cooperative multi-agent deep reinforcement learning," *arXiv:1908.03963*, 2019.
- [26] A. Mathkar and V. S. Borkar, "Distributed reinforcement learning via gossip," *IEEE Trans. Autom. Control*, vol. 62, no. 3, pp. 1465–1470, 2017.
- [27] S. Kar, J. M. Moura, and H. V. Poor, "QD-Learning: A collaborative distributed strategy for multi-agent reinforcement learning through consensus + innovations," *IEEE Trans. Signal Proc.*, vol. 61, no. 7, pp. 1848–1862, 2013.
- [28] S. V. Macua, J. Chen, S. Zazo, and A. H. Sayed, "Distributed policy evaluation under multiple behavior strategies," *IEEE Trans. Autom. Control*, vol. 60, no. 5, pp. 1260–1274, 2015.
- [29] D. Lee, H. Yoon, and N. Hovakimyan, "Primal-dual algorithm for distributed reinforcement learning: Distributed GTD," in *IEEE Conf. Decision and Control*, 2018, pp. 1967–1972.
- [30] Y. Zhang and M. M. Zavlanos, "Distributed off-policy actor-critic reinforcement learning with policy consensus," *arXiv:1903.09255*, 2019.
- [31] M. S. Stanković and S. S. Stanković, "Multi-agent temporal-difference learning with linear function approximation: Weak convergence under time-varying network topologies," in *2016 American Control Conference (ACC)*, 2016, pp. 167–172.
- [32] T. Doan, S. Maguluri, and J. Romberg, "Finite-time analysis of distributed TD(0) with linear function approximation on multi-agent reinforcement learning," in *Proc. Int. Conf. Machine Learning*, 2019, pp. 1626–1635.
- [33] M. S. Stanković, M. Beko, N. Ilić, and S. S. Stanković, "Distributed multi-agent reinforcement learning algorithm based on gradient correction," in *Proc. IcETRAN Conference*, 2020.
- [34] H. Yu, A. Mahmood, and R. Sutton, "On generalized Bellman equations and temporal-difference learning," *Journal of Machine Learning Research*, vol. 19, pp. 1–49, 2019.
- [35] S. S. Stanković, M. S. Stanković, and D. M. Stipanović, "Decentralized parameter estimation by consensus based stochastic approximation," *IEEE Trans. Autom. Control*, vol. 47, pp. 531–543, 2011.
- [36] H. J. Kushner and G. Yin, *Stochastic Approximation and Recursive Algorithms and Applications*. Springer, 2003.

Arduino based online laboratory platform for digital control systems analysis and design

Vladimir Mitić, Vladimir Sibinović, Snežana Đorđević, Boban Veselić

Abstract— This paper presents a laboratory platform with remote access for exercising digital control systems analysis and design. The laboratory environment utilizes three types of standard control systems: positional servosystem, velocity servosystem and temperature regulation system. It also has online access through a standard Web browser that enables remote execution, visualization and data acquisition. Students can perform different variety of experiments, ranging from bachelor to master level studies. They can work independently or collaborate within a group with multiuser access to the platform. The laboratory platform is designed a way to be applicable to any regulation system with implemented communication protocols.

Index Terms— Laboratory platform, remote access, digital control theory, Arduino.

I. INTRODUCTION

The previous two years, because of the pandemic outbreak, introduced remote learning to all levels of education. While this type of learning is not new, it was never before used at such an extent and for a large number of topics.

Remote laboratories were always popular in engineering education, but their popularity has grown rapidly in recent years. With the development of different technologies, e.g., Internet-of-things, more and more researchers turn to developing remote laboratories for different topics.

The results presented in this paper stemmed from the efforts, started a long time ago, trying to modernize old laboratory teaching platforms that were used to explain control system fundamentals to students. The possibility of remote access to the laboratory setup was a priority from a very beginning. Remote experiments offer more flexibility both for teachers and students, and some experiments were performed at the Faculty of Electronic Engineering in Niš more than 10 years ago [1].

Gaining experience in that field over the years, the authors started developing a new laboratory platform for teaching control system fundamentals. With flexibility and simplicity, as the main guiding principles, a new platform has been designed for teaching digital control systems analysis and design. Remote access to the platform requires from a student to only have a device with internet access. On the other hand, the platform can be built using off the shelf components that are inexpensive.

The paper is organized as follows. Section II presents the hardware system design, while Section III describes the software architecture of the system Section IV shows the

educational possibilities and potentials of the realized laboratory platform and demonstrates its ease of use through some examples of laboratory exercises.

II. HARDWARE ARCHITECTURE

Arduino Due development board has been chosen to be the key component of the laboratory platform. Arduino is an open-source development environment that has become ubiquitous in education, from the primary schools all the way to the universities. Arduino platform provides different inexpensive development boards, that can be used for teaching different topics like programming [2], robotics [3], physics [4], etc. Arduino Due has an Atmel SAM3X8E Arm Cortex-M3 microcontroller, which uses a 32-bit RISC processor operating at the maximum frequency of 84 MHz. This gives the platform adequate processing power and enables further development of the system. It also has plenty of general-purpose input/outputs (GPIOs) for connecting other parts of the system. The main disadvantage is that this system is 3.3V transistor-transistor logic (TTL) and requires logic level converters to work with components having 5V TTL.

In order to be able to create a set of different laboratory exercises, the three standard control systems have been implemented: positional and velocity servosystems using DC motors, and temperature regulation system using heating elements.

A. Positional and velocity servosystem

The positional and velocity servosystems were realized using Couzet 82830010 DC motor that has the nominal power output of 33W. A gearbox with the ratio of 12.25 is coupled with the motor for the positional servosystem, for more accurate angular position control. An absolute magnetic encoder with 4096 increments per revolution is chosen for position measuring. These measurements are also used for the angular speed estimation of the motor shaft [5]. This was chosen as the most inexpensive solution with the highest resolution. Since the encoder is absolute, a lot of attention has been dedicated to detecting the zero-crossing condition, especially when operating on high speeds. The motor speed is estimated using simple Euler derivative approximation, which means that for small sampling periods the speed signal can be noisy. According to the predicted motor current requirements, Pololu VNH5019 motor driver has been chosen. A custom printed circuit board (PCB) was designed

Vladimir Mitić is with the Faculty of Electronic Engineering, University of Niš, Aleksandra Medvedeva 14, 18000 Niš, Serbia (e-mail: vladimir.mitic@elfak.ni.ac.rs)

Vladimir Sibinović is with the Faculty of Electronic Engineering, University of Niš, Aleksandra Medvedeva 14, 18000 Niš, Serbia (e-mail: vladimir.sibinovic@elfak.ni.ac.rs)

Snežana Đorđević is with the Faculty of Electronic Engineering, University of Niš, Aleksandra Medvedeva 14, 18000 Niš, Serbia (e-mail: snezanadjordjevic@elfak.rs)

Boban Veselić is with the Faculty of Electronic Engineering, University of Niš, Aleksandra Medvedeva 14, 18000 Niš, Serbia (e-mail: boban.veselic@elfak.ni.ac.rs)

and manufactured for easy component interconnection and system replication. All the required mechanical components were designed and 3D printed with the special care for safety and transferability. A photo of the laboratory setup is displayed in Fig. 1.

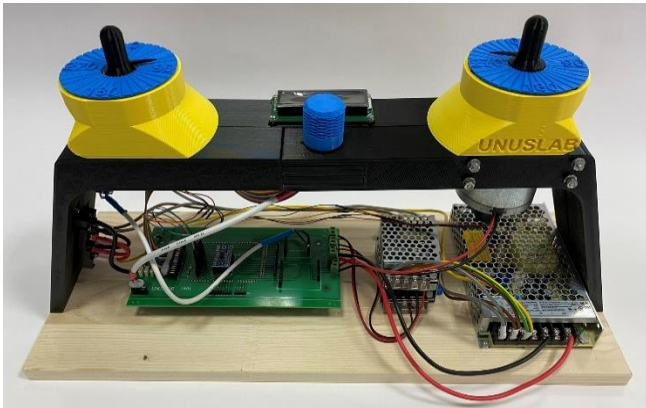


Fig. 1. Velocity servosystem laboratory platform.

A power supply port, an USB port for communication and an USB port for Arduino programming are located in the left-hand side of the platform. The power supplies are integrated into the platform, in the right-hand side, so only a standard PC power cable is needed for powering the system. The laboratory setup was designed to operate both in the stand-alone mode (controlled using the knob and display in the middle) and with usage of a computer. The referent speed/position is selected using the pointer on the left, and the motor rotation is observed using the pointer on the right.

B. Temperature regulation of a thermal process

Temperature regulation system consists of a ceramic resistor with 2.7 Ω resistance and maximum power output of 10 W. It has the role of a control object whose temperature needs to be regulated. A strong MOSFET transistor serves as the actuator of the system, and it allows current switching through the resistor. The feedback is realized using an analog temperature sensor that is amplified to achieve the highest resolution for the temperature range of 0 $^{\circ}\text{C}$ – 100 $^{\circ}\text{C}$. A photo of the temperature regulation platform is shown in Fig. 2.

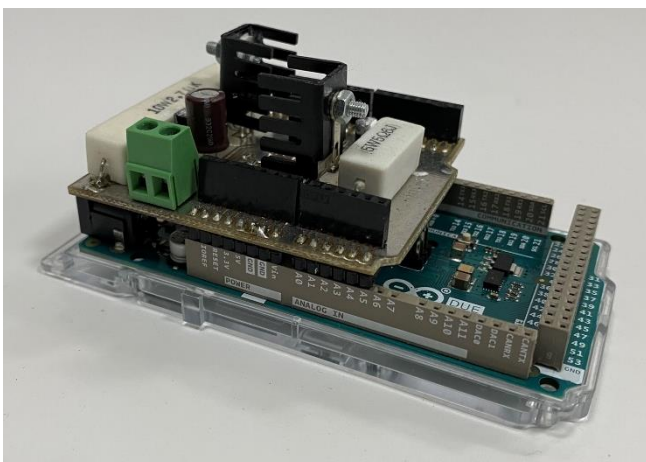


Fig 2. Temperature regulation system

The system also has a secondary ceramic resistor, that has the resistance of 5.6 Ω and maximum power dissipation of 5 W. The resistors represent two different heating elements,

that can be utilized either separately or together. This provides flexibility to the platform to be able to emulate different thermal processes.

III. SOFTWARE ARCHITECTURE

The software architecture consists of three parts: (i) real-time control system implemented on the Arduino DUE, (ii) server that generates the user interface (UI) and (iii) a bridge application that connects the previous two. The system block diagram is given in Fig. 3.

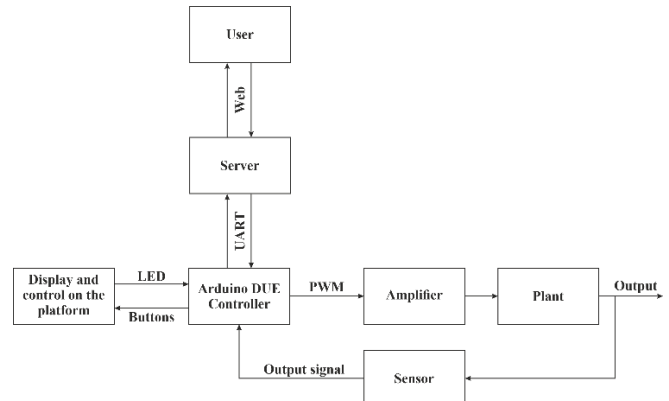


Fig 3. Block diagram of the system.

A. Communication protocols

The real-time control system communicates with the bridge application using universal asynchronous receiver-transmitter (UART) protocol. UART messages are optimized for the minimization of number of bytes, which means that they need to be decoded and passed to the server. This is implemented because of the limited transmission speed of the protocol with really large amount of data (e.g., plot data). The bridge application decodes the UART data and generates JavaScript object notation (JSON) messages that are transferred to the server via WebSocket protocol. The communication is organized using a master-slave type of protocol where the server is the master, and the Arduino is the slave. Every message from the server generates a response from the Arduino so any communication error can be easily recognized. If no response is generated retransmission feature has been implemented as well. This provides reliable communication between system components.

B. Real-time control system

The real-time control system consists of three main parts: 1) control loop, 2) communication handler and 3) display handler.

1) Control loop

The control loop must have a real-time component, executed every 4 ms by generating the timer interrupt. It handles the sensor readings for the reference and the output signal, calculate the control signal according to the applied control algorithm, converts that signal from voltage to PWM and forwards it to the selected driver. The control loop can be set to be executed with a longer sampling period, which is the integer product of 4 ms. This enables changing the sampling period of the system. This feature is useful for the experiments that analyze the effects of the sampling period size on a digital control system, as well as for the systems that

are considerably slower, such as the temperature regulation system. Several digital control laws can be employed for the control signal calculation. The implemented standard digital controllers are: PD, PID, PID AW, I-P, I-PD and universal controller. PD controller can be used for realization of P controller, by setting the derivative gain to zero. The same goes for PID controller that can be used for the realization of PI controller. PID AW represents the standard PID controller with anti-windup mechanism. I-P and I-PD controllers denote controllers with relocated P and PD actions, respectively. The implemented universal controller can represent any second-order (or less) discrete-time transfer function described by

$$G(z) = \frac{b_2z^2 + b_1z + b_0}{a_2z^2 + a_1z + a_0} \quad (1)$$

This means that virtually any linear digital controller (up to the second-order) can be tested on the given system.

It is also envisaged that the control signal can be formed using state feedback control approach. In order to implement the state feedback control loop, it is necessary to use the state-space representation of the control plant. The state variables required for the state feedback control can be obtained directly by measuring, or by using state estimation. For this reason, a predictive state observer, a current state observer and a Kalman estimator (filter) were also implemented. The software is realized in a such manner that the real-time simultaneous switching between the controllers can be done. This offers the students the possibility to compare the efficiencies of various controllers.

The desired reference signal can be obtained manually from the user or as the output of the built-in signal generator. The implemented signal generator can generate real-time step, ramp, parabolic, pulse, sine and sawtooth signals. The signal parameters can be arbitrary set and the switching between signals can be done in real-time.

2) *Communication handler*

The communication handler manages receiving a message, its decoding and generating and sending the corresponding response. The communication handler has various types of error checking, starting from incorrect number of bytes to incorrect data or data types. The number of system parameters that can be changed through the communication handler is 70. This means that every controller gain, every signal generator parameter and every other control loop parameter can be changed by the user through the Web UI. The communication handler is responsible for generating and sending the plot data, which consists of the timestamp, referent signals, output signals, and control signals. The time period of the plot data sending is defined as a user settable parameter, enabling the user to appropriately adjust the plot figure as needed.

3) *Display handler*

The display handler controls the information presented on the display and the knob that is used for moving through the menus. The display information is useful when performing experiments in the stand-alone mode (not connected to a computer). Still, for the full functionality of the system the Web UI should be used.

C. *Web UI*

The Web UI is accessible through any Web browser with no special requirements. When granted the access to the platform by the system administrator, the user establishes the

connection to the Web UI by typing in the Web address as shown in [6]. Multiple users can access the platform at the same time which enables working in groups, where the members don't need to be in the same location. This has proven to be a useful option, since the pandemic has separated students from each other. The first batch of students that used the laboratory had positive feedback on the look and usability of the Web UI. When accessing the provided Web page, a login screen firstly appears. After entering the correct login information, the control page is shown. The control page is dynamically changing the displayed parameters based on the selected options. Fig. 4. shows the overview of the control page.



Fig. 4. The control page of the Web UI.

The control page consists of the 1) control panel on the left, 2) control chart on the right and a 3) message panel on the bottom.

1) *Control panel*

The control panel is used for starting and stopping the system and for system parameter manipulation. It is divided into four sections. The first section consists of the *Start/Stop* button and a system status indicator. The possible system statuses are: Unknown, Stopped, Running and Disconnected. The statuses are described in Table 1.

TABLE I
SYSTEM STATUS DESCRIPTION

Status	Description
Unknown	The bridge application is not responding.
Stopped	Control loop is disabled, and plot data sending is stopped.
Running	Control loop is enabled, and new plot data is being displayed.
Disconnected	Connection with the server is terminated, the page should be refreshed

The second section is intended for reference signal control and output signal display. A dropdown list is used for selecting if the reference signal is entered manually or if it is connected to the signal generator. If a signal generator is selected, the corresponding signal parameters are displayed and can be changed. When the state feedback control is enabled, two input fields are displayed. One for each referent

state, and two display fields for each measured state variable. A *Show trend* checkbox displays three new fields, showing the current error, the error derivative and the error integral.

The third section is intended for setting the control loop. The first adjustable parameter is the system sampling period. A dropdown list is used for selecting the loop type and controller type. Loop type can be the standard negative feedback or the state feedback. Controller type is used for selecting one of the implemented digital controllers, explained in the previous section. When selecting the desired controller, corresponding parameters of that controller are displayed and can be tuned. Finally, there is a check box for selecting open or closed control loop. This is mostly useful in system identification exercises.

The fourth section consists of two buttons, one for refreshing of all parameters and one for resetting the all parameters to their default values. Below the control panel, there is a button that is used for displaying the webcam image. It opens a new tab that will display a livestream of the laboratory platform that is obtained via a webcam, which is mounted above the platform.

2) Control chart

The control chart is an interactive chart that can be manipulated in various ways by the user. Signal manipulation can be accomplished on the chart itself. Every signal on the chart can be hidden or shown by clicking on the signal name. Every point on the chart can be displayed in detail by hovering the mouse pointer over it. Additional controls for manipulating the chart are located directly below it. They consist of the following buttons: *Zoom in*, *Zoom out*, *Reset view* buttons and *Auto-scroll* checkbox for closer observations of the response, *Save graph* and *Save data* buttons for storing the response and an *Advanced settings* checkbox. *Save graph* button stores the presented response as an "Graph.png" file while the *Save data* button stores the response in a "data.csv" file. Every row in the file represents one timeseries, and every column one signal. This makes the data easily readable, and importable into various software packages (like Matlab) for further detailed analysis. The *Advanced settings* checkbox displays three new input fields: *Plot acquisition period*, *Number of seconds to store* and *Number of samples to store*. *Plot acquisition period* defines the period at which the platform sends the data to the user. This period can be different to the control sampling period if we do not need so much data collected and displayed. *Number of seconds to store* defines the timespan that is shown on the chart and that will be stored with the save button. *Number of samples to store* represents the number of timeseries points that is displayed and stored. Because these two fields affect the same property (timespan of the plot), only one of them should be changed.

With this kind of interactive control chart, it is obvious that it alone is sufficient for any standard system analysis, which is done when executing a laboratory exercise. This enables the students to observe the experiment, notify and immediately correct any unexpected system behavior.

3) Messaging system

The messaging system is a feature that is designed for student groups that are scattered in different places. Using the toggle button on the top of the messaging panel, different types of messaging channels can be selected. Firstly, there is

the *User-device* channel that is used for verifying that all the parameters are set correctly. Secondly, there is the *User-user* channel which is organized in a standard chat form and enables students to collaborate more effectively while performing the given task.

IV. LABORATORY EXERCISES

The realized laboratory platforms have a lot of teaching potential for the students, and they can provide practical engagement for students to better learn and understand control theory. First, students can learn about system identification techniques for different types of control plants. For example, students can record the step response of the system by removing the feedback loop and setting the digital controller to PD with the gains $k_p = 1$, $k_d = 0$. In this way, the value of the referent signal is directly passed to the plant. They can save the response data and use Matlab for system identification. By performing this on the servosystems, they learn about first- and second-order transfer function identification. In case of the thermal plant, they can learn about identifying a system with transport delay.

The second set of experiments are devoted to time-discretization process and signal reconstruction in digital control systems. Students can observe the effects of the sampling period selection on a system performance. They can experimentally determine critical gain and critical frequency when system reaches stability margin, Fig. 5. Also, they can calculate steady state errors to different input signals and experimentally validate them.

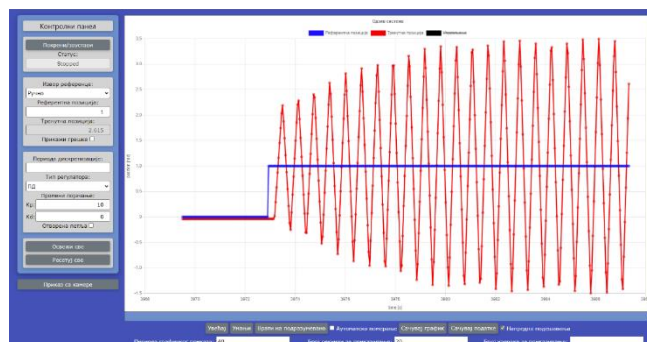


Fig 5. Experimentally determining the critical gain and frequency of a closed loop system.

When it comes to digital control systems design, the implemented digital controllers give the students a lot of possibilities. Starting from standard control loops, they can design P, PI, PD, PID controllers, lead-lag compensators and compare the system responses that they provide. Also, the students can experience some issues related to practical implementation of a control system. One of them is the saturation of the applied control signal, which can greatly affect the performance of a system. Consequently, possibilities are envisaged to implement digital controllers with relocated P and D actions that can reduce control signal peaks, as well as to enhance a control system with an anti-windup structure. The latter is crucial for temperature regulation system because of the large integrator windup produced by the dynamic properties of the system as well as the present transport delay, Fig. 6.

The state feedback control structure implementation widens even more the learning possibilities of the platform.

By designing the state feedback controller, students get a better understanding of the state-space representation of a system and its practical usage. The implemented prediction state observer, current state observer and Kalman estimator, as a special type of current state observer, offers possibilities to students to try their design and get insight into state estimation and its possible usages. An example of application of current state observer in state feedback control is given in Fig. 7.

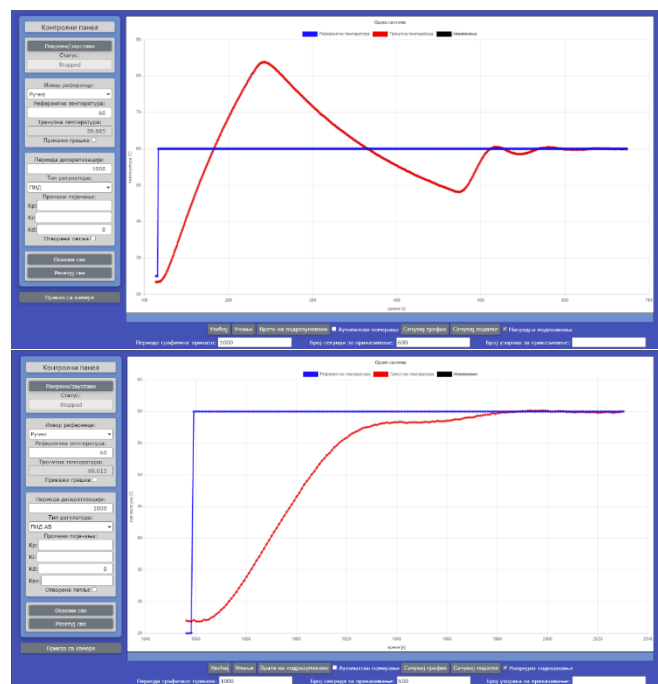


Fig. 6. Temperature regulation system without anti-windup structure (top), and with it (bottom).

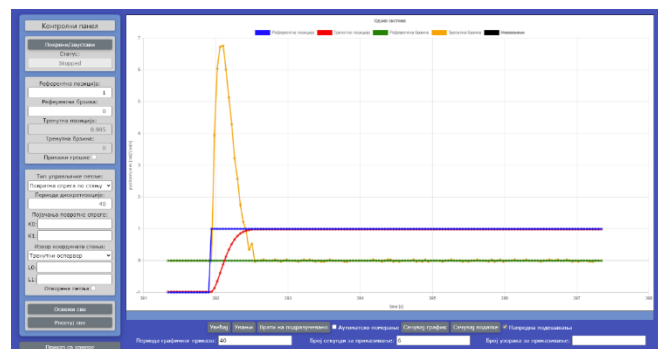


Figure 7. State feedback control implemented with current state observer.

V. CONCLUSION

The designed laboratory platform represents a universal solution for teaching of different types of digital control systems. User friendly and modern UI makes it very easily understandable for students. Its inexpensiveness makes it a viable option for a lot of educational facilities. The applied motors and drivers can be easily replaced with any motor power and driver combination, without losing any of the functionality of the exercises. The Arduino microcontroller used here enables for realization of even more complex control loops, which can be useful to students at a higher educational level. The scalability of the system makes it work with any digital system with UART protocol and this type of communications handler. The special value of this platform is the possibility of remote access and performing laboratory exercises from a distant location, which was especially useful for students in a pandemic. Future work will be directed to introduction a cooling element to the temperature regulation system, implementing force feedback loops as well as a cascaded control system with the positional outer loop and velocity inner loop for jerk reduction. This will broaden the possible systems for which students can design and test different control structures.

ACKNOWLEDGMENT

Part of this research was financed by the Serbian Ministry of Education, Science and Technological Development of the Republic of Serbia.

REFERENCES

- [1] M.B. Naumović, D. Živanović: "Remote Experiments in Control Engineering Education Laboratory", International Journal of Online Engineering, ISSN: 1861-2121, vol. 4, No. 2, 2008, pp. 48-53.
- [2] I.Perenc, T. Jaworski, P. Duch, "Teaching programming using dedicated Arduino Educational Board", Computer Applications in Engineering Education, Vol. 27. Issue 4, July 2019, pp 943-954.
- [3] F.A. Candelas, G.J. Garcia, et. al., "Experiences on using Arduino for laboratory experiments of Automatic Control and Robotics", IFAC-PapersOnLine, Vol 48, Issue 29, 2015, pp 105-110.
- [4] J Kinchin, "Using an Arduino in physics teaching for beginners", Physics Education, Vol 53, No 6, 2018
- [5] AEAT-6010/6012 Magnetic Encoder, <https://docs.broadcom.com/docs/AV02-0188EN>
- [6] http://25.49.3.248:8080/WebPSAU_Position/

Implementation of the New Curricula in Smart Products and Services Engineering

Marko T. Milojković, Dragan S. Antić, *Senior Member*, Saša S. Nikolić and Nebojša S. Jotović

Abstract— Education system proved to have a challenge to cope with today's world of accelerated development of new technologies and ever changing industry. New paradigm dictates not only new teaching contents, but whole new holistic approach to the education of engineers, especially in the field of development of smart products and services. A group of professors from the University of Niš (Faculties of Electronic and Mechanical Engineering) developed a new teaching module named "Smart Products and Services Engineering" and successfully applied for funding from European Education and Culture Executive Agency in the scope of Jean Monnet Erasmus+ project call. Raising the level of knowledge and skills of students in the field of smart products and services should further encourage the development of a modern educational system in one of the most important areas of technology.

Index Terms—Smart products, smart services, engineering education.

I. INTRODUCTION

The advancement of information and communication technologies is emerging at such a pace that they are increasingly penetrating traditional industries, changing products, processes, organizations, business models and entire value chains. Existing products are transformed into smart products [1], [2] which can intelligently communicate with other smart products and react autonomously to changes in their environment. Smart products are expanding into smart services or smart service systems. Designers today must quickly and flexibly integrate new trends and standards into their solutions. Therefore, the application of modern efficient engineering of the new generation of smart products and services, characterized by high interdisciplinarity, networking, complexity and heterogeneity, is more than necessary. Today, it is convenient to consider a holistic approach to engineering, where, in addition to the development of products and services, their complete life cycle is also taken into account. The components of holistic engineering are the models and methods used in the development process, IT tools, information models and organizational structures, united by human resources with appropriate competencies. The key competence of development engineers is creativity, because it enables successful innovations, i.e. the development of creative and market-competitive products. All these modern

tendencies dictate a new approach to education of engineers in the field of smart products and services [3], [4], [5].

With the goal to enable master students of technical faculties to acquire adequate competencies in the application of methods, models and IT tools in the efficient engineering of smart products and services, group of professors from the University of Niš successfully applied for funding from EACEA (European Education and Culture Executive Agency). Concrete project application was for Erasmus+ programme call Jean Monnet Actions in the field of Higher Education: Modules, with the project named "Smart Products and Services Engineering" (acronym SPaSE) 101047566-JMO-2021-HEI-TCH-RSCH. Project team consists of ten professors from two faculties: six from the Faculty of Electronic Engineering and four from the Faculty of Mechanical Engineering. Project duration is December 2021 - November 2023 (36 months), and during that time, project team is obligated to hold teaching module dedicated to Smart Products and Services Engineering for three generations of students. Students applicable for the module will be the students from technical faculties who already finished basic level of studies (either master students or working persons).

II. PROJECT OBJECTIVES

Main objectives of the project "Smart Products and Services Engineering" (SPaSE) are:

- Acquiring the necessary knowledge and skills by the master students of technical faculties in the field of innovation and creative techniques for the development of smart market-competitive products and services.
- Mastering the methods, models and IT tools applicable in efficient engineering of the new generation of smart products and services.
- Transfer of acquired engineering competencies in the field of smart products and services to the business entities in the region.
- More efficient use of available human resources and faster integration of the region in the application of European achievements in this area.

Project aims to contribute to the acquisition of professional competencies and innovative readiness in the field of smart product development both in educational institutions and in economic entities in the region. New student teaching module will be the focal point, where the students will learn the basics of Industry 4.0 (I4.0), application of creativity and innovation in product development, basics of digital product development, application of artificial intelligence, sensor

Marko T. Milojković, Dragan S. Antić, Saša S. Nikolić and Nebojša S. Jotović are with the University of Niš, Faculty of Electronic Engineering, Department of Control Systems, Aleksandra Medvedeva 14, 18000 Niš, Serbia (e-mails: {marko.milojkovic, dragan.antic, sasa.s.nikolic, nebojsa.jotovic}@elfak.ni.ac.rs).

technologies and business models I4.0. Special attention will be paid to the development of relevant competencies [6]: professional, methodological, and social. Students will work in teams on practical projects where they will be able to strengthen creativity and teamwork.

Besides teaching part, project team will have numerous related activities oriented toward teaching and scientific community as well as toward local business entities in the field of smart products and services. A lot of events are planned like: round tables with business, workshops, study visits, writing research papers as well as a handbook on smart products and services. All these activities will help reaching the project specific objectives:

O1: Raising the level of knowledge and skills in the field of smart products and services

O2: Efficient application of methods and IT tools for engineering a new generation of smart products and services

O3: Application of creative and innovative techniques, which provides an opportunity for multidisciplinary application of advanced technologies

O4: Strengthening human and institutional resources in the field of research and development

O5: Encouragement to development of a modern educational system in one of the most important areas of technology

O6: Encouragement to strengthen the competitiveness of the national economy

O7: Increasing the technological readiness of educational institutions and economic entities for Serbia's integration into the EU.

III. MODULE COMPOSITION

Jean Monnet Module "Smart Products and Services Engineering" is composed of 11 courses with 118 school hours of teaching in total. Module is organized in 15 weeks (3 months), two teaching days in a week, with 4 hours teaching blocks.

C1. Introduction to European integration and legislation in the field of smart engineering (6 teaching hours). The main goal is to enhance the existing theoretical understanding of legal, economic, political, and social aspects of European integration. The lectures are based on the interdisciplinary approach regarding the European integration process and its benefits, law, and future aspects with emphasize on the field of smart engineering.

C2. I 4.0: Smart products and services engineering (12 teaching hours). The main goal of the course is to master the knowledge and skills for the efficient application of engineering of the new generation of smart products and services. As part of the course, students will study basic approach in product design and development, modular principle in product development, Industry 4.0 and Reference Architecture Model for Industry 4.0 - RAMI 4.0, holistic approach in product development, smart product and service development models [7], [8].

C3. Creativity and innovation in product development (12

teaching hours). The main aim of this course is for students to master creative skills and competencies and also to present them innovative techniques which will be helpful in developing of future smart products and services [9].

C4. Digital product development (12 teaching hours). The course program will train students to independently use state of the art technologies and methods in the process of development of digital products. The acquired knowledge will enable students to create and holistically administer information, documents and resources in the digital product development process, thus fostering innovation [10].

C5. 3D CAD construction (10 teaching hours). Getting the knowledge about the basic geometrical objects and their relative positions and sections, developing surfaces, as well as the vector analysis and computing graphics. The focus of the course is on parametric design and the proper use of CAD software to produce 3D models, engineering parts, and documentation [11].

C6. Information technology (12 teaching hours). Introduction to the basic IT principles, methods, and techniques. The ultimate goal is the IT education of IT experts, capable of application of information technologies in the industry at all stages during the development of software solutions [12], [13].

C7. Artificial intelligence (12 teaching hours). The aim of course is to introduce the concepts and algorithms at the foundation of modern artificial intelligence, and explore the ideas that give rise to smart products and services. Through real world examples, students will gain knowledge to recognize and apply main tools and algorithms of AI and machine learning with the special emphasize on multilayer artificial neural networks (deep learning) [14], [15].

C8. Big data and data analytics (12 teaching hours). Upon completion of this course the students should become familiar with the principles of Big data and data analytics, gain understanding of the basic tools and techniques for data collection, processing and analysis. The desired outcome is to stimulate the students to adapt EU legislative in this area and apply main tools and algorithms of these concepts in design of simple smart products and services [16], [17].

C9. Internet and sensing technology (12 teaching hours). This course aims to provide an overview of technologies focusing on the Internet of Things and Industrial Internet of Things. Besides fundamentals and operational principles, the emphasize will be on introducing legal EU requirements and frameworks for building secure and reliable systems, opportunities and challenges, successful real-world practices, and efficient internet and sensing applications [18].

C10. Digital twins (10 teaching hours). Digital twin fundamentals, which integrate the internet of things, artificial intelligence, and software programming, will be introduced as a virtual representation of an object or system during its lifecycle, which is updated from real-time data and uses simulation, machine learning, and reasoning to help decision-making [19].

C11. Business models I4.0 (8 teaching hours). This course will be focused on analysing business model characteristics

for Industry 4.0, providing an in-depth perspective of companies' processes, structures, and tools for business model innovation. The course will also provide insight in EU good practices and competencies for business model innovation in the course of Industry 4.0 [20].

IV. CONCLUSION

This paper presented main aspects of newly designed student module at the University of Niš, named "Smart Products and Services Engineering". Module implementation is financially supported by European Education and Culture Executive Agency (EACEA) under the umbrella of Jean Monnet Erasmus+ project call. Module is jointly designed by professors from two faculties – Faculty of Mechanical Engineering and Faculty of Electrical Engineering. Module "Smart Products and Services Engineering" is composed of 11 courses with 118 school hours of teaching in total, organized during 15 weeks (3 months), two teaching days in a week, with 4 hours teaching blocks. Topics to be mastered are: Introduction to European integration and legislation in the field of smart engineering, I4.0: Smart products and services engineering, Creativity and innovation in product development, Digital product development, 3D CAD construction, Information technology, Artificial intelligence, Big data and data analytics, Internet and sensing technology, Digital twins, Business models I4.0.

Besides the main goal of the project which is acquiring the necessary knowledge and skills by the master students of technical faculties in the field of innovation and creative techniques for the development of smart market-competitive products and services and mastering the methods, models and IT tools applicable in efficient engineering of the new generation of smart products and services, project has several additional goals. Most important ones are the transfer of acquired engineering competencies in the field of smart products and services to the business entities in the region as well as more efficient use of available human resources and faster integration of the region in the application of European achievements in this area. In order to achieve all these goals, project anticipates besides teaching part, numerous related activities oriented toward teaching and scientific community as well as toward local business entities in the field of smart products and services. A lot of events are planned like: round tables with business, workshops, study visits, writing research papers as well as a handbook on smart products and services.

ACKNOWLEDGMENT

Funded by the European Union under project Smart Products and Services Engineering (SPaSE) 101047566-ERASMUS-JMO-2021-MODULE. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or EACEA.

Neither the European Union nor the granting authority can be held responsible for them.

REFERENCES

- [1] V. Miltenović and D. Antić, "Smart Products and Services Engineering," University of Niš, Niš, Serbia, 2020.
- [2] V. Miltenović and N. Marjanović, "Development of innovative and smart products - opportunities and challenges," Proceedings of the 9th International Scientific Conference-Research and Development of Mechanical Elements and Systems, Kragujevac, S. 11-21, 5-7, September, 2019.
- [3] V. Miltenović and V. Đorđević, "Excellence, relevance and efficient application of research results at university and institutes from the standpoint of economy developing," Proceedings of the International Symposium Machine and Industrial Design in Mechanical Engineering (KOD), Novi Sad, Serbia, vol. 393, 6-8 June 2018.
- [4] G. K. Fomunyan, "Education and the fourth industrial revolution: challenges and possibilities for engineering education," International Journal of Mechanical Engineering and Technology, vol. 10, no. 6, pp. 271–284, 2019.
- [5] S. Coskun, Y. Kayıkcı, and E. Gençay, "Adapting engineering education to Industry 4.0 vision," Technologies, vol. 7, no. 1, pp. 1–10, 2019.
- [6] A. Miltenović, M. Banić, and V. Miltenović, "Importance and role of competence in professional career of product develop engineers," Proceedings of the 7th International Conference on Manufacturing Science and Education - MSE 2015, Sibiu, Romania, June 3-6, 2015.
- [7] Plattform Industrie 4.0: Reference Architectural Model Industrie 4.0 (RAMI 4.0). Available online: https://www.plattform-i40.de/I40/Redaktion/DE/Downloads/Publication/Rami40-eine-einfuehrung.pdf?__blob=publicationFile&v=10 (Stand: 3.5.2018).
- [8] M. Abramovici, Engineering smarter Produkte und Services Plattform Industrie 4.0. Studie, Deutsche Akademie der Technikwissenschaften. München, 2018.
- [9] V. Miltenović, R. Mitrović, N. Burkardt, S. Stefanov, A. Miltenović, M. Banić, and M. Tica, Innovation management (in Serbian). University of Niš, Faculty of Mechanical Engineering, (2017).
- [10] European Commission, 2021. 2030 Digital Compass: the European way for the Digital Decade. Available online: <https://eur-lex.europa.eu/legal-content/en/TXT/?uri=CELEX%3A52021DC0118>
- [11] Z. Jeli B. Popokstantinovic, and Stojicevic M., "Usage of 3D Computer Modelling in Learning Engineering Graphics," In Virtual Learning, InTechOpen, pp. 49–70.
- [12] G. Gay and R. Blades, Information Technology, Oxford Information Technology for CSEC, 3rd edition, 2019.
- [13] Introduction to Computers and Information Technology, Prentice Hall, 2011.
- [14] S. J. Russell and P. Norvig, Artificial Intelligence: A Modern Approach, 3rd edition, 2015.
- [15] European Commission, 2020. On Artificial Intelligence - A European approach to excellence and trust. Available online: https://ec.europa.eu/info/publications/white-paper-artificial-intelligence-european-approach-excellence-and-trust_en
- [16] European Commission, 2020. A European strategy for data. Available online: <https://eur-lex.europa.eu/legal-content/EN/TXT/?qid=1593073685620&uri=CELEX:52020DC0066>
- [17] B. Marr, Big Data: Using SMART Big Data, Analytics and Metrics To Make Better Decisions and Improve Performance, John Wiley & Sons, 2015.
- [18] G. Veneri and A. Capasso, Hands-On Industrial Internet of Things: Create a powerful Industrial IoT infrastructure using Industry 4.0, Packt Publishing Ltd, 2018.
- [19] F. Tao, M. Zhang, and A.Y.C. Nee, Digital Twin Driven Smart Manufacturing, 1st Edition, Academic Press, 2019.
- [20] T. Döbler, B. Sniderman, M. Mahto, and C. Ahrens, Driving innovation and new business models through Industry 4.0, Deloitte insights, 2020.

Denoising the open-loop step response using an encoder-decoder convolutional neural network

Natalija Đorđević, Nenad Džamić, Aleksa Stojić, Goran Kvašček

Abstract—Denoising signals is used as a preprocessing step for all signal processing. Encoder-decoder neural networks are often proposed as a method of denoising 1D and 2D signals, because of their ability to extract essential features from the signal and then recreate it without noise. In this paper we propose a simple architecture of a convolutional neural network for denoising step responses of systems with different open-loop transfer function. The network is trained on synthetic data with added noise of different distributions, then tested on a portion of synthetic data and real-life step responses.

Index Terms— denoising, encoder-decoder, convolutional neural network, step response

I. INTRODUCTION

Signal denoising is an important preprocessing step in every type of signal processing. Even though many observations can be made on generated pure signals, real-life signals always come with a certain amount of noise. For any signals measured on electrical circuits, noise cause can be the imperfect design or layout of the circuit itself, faulty components, close proximity to other electrical equipment, environmental causes etc. However, even if causes are known, the behavior of noise is unpredictable and rarely fits into a specific probability density function. Having a denoising method that would be applicable to different types of signals and noises would be of great importance for signal processing.

In the paper [1], authors used deep recurrent denoising neural networks to denoise ECG signals and improve signal-to-noise ratio of signal from -8.82dB to 7.71dB. They used a synthetic dataset with added noise. Medical signals such as ECG are of great importance, which shows in the number of papers that deal with their denoising. Once again, a convolutional neural network proves efficient with ECG signals in [2]. Seismic data is one of the representatives of noisy real-life data, and effective use of deep convolutional neural networks for denoising of synthetic seismic data can be seen in [3]. Using a convolutional network for denoising images with Gaussian noise is also proposed in [4]. There is not a single specific architecture for denoising convolutional networks that is superior, and a useful comparison of different kinds is given in [5]. However, one structure seems to be mentioned for multiple purposes, and it is

Natalija Đorđević is a teaching associate at the Signals and systems department at the School of Electrical Engineering, University of Belgrade, 73 Bulevar kralja Aleksandra, 11020 Belgrade, Serbia (e-mail: natalija.djordjevic@etf.bg.ac.rs)

Nenad Džamić is with the School of Electrical Engineering, University of Belgrade, 73 Bulevar kralja Aleksandra, 11020 Belgrade, Serbia (e-mail: dn213336m@student.etf.bg.ac.rs)

the encoder-decoder structure. The authors in [6] suggest the use of an autoencoder for medical image denoising, which served as an initial inspiration to use encoder-decoder networks for 1D signals. Application of the encoder-decoder structure on denoising micro seismic signals is shown in [7].

In this paper we have trained an encoder-decoder neural network to denoise a step response of an open-loop system. This particular structure seemed promising and in various papers was proven to perform well in denoising for different purposes. Its great advantage is that it does not have strict limitations on the results it can produce. There are however many things that have an effect on its performance, such as the architecture, hyperparameters, the dataset etc. In this paper we attempted to make a structure that is not complicated, but performs well, as well as to generate a dataset that is informative enough for our neural network. The reason that the step response signal was chosen is its importance in observing how industrial processes react to a change in reference value. Without diving into the details of which processes each system represents, we will define their behavior only by their open-loop transfer functions.

II. DATASET SYNTHESIS

Collecting data for neural networks can be expensive and time-consuming. In order for our dataset to be as diverse as possible and simultaneously to represent many industrial processes, we used equation (1) as a general form for our open-loop transfer function. [8].

$$G(s) = \frac{e^{-\theta_d s}(1-\alpha s)}{(s+1)^n} \quad (1)$$

Parameter θ_d represents transport delay which is the amount of time that our system needs to react to a change in reference value. The measure $1/\alpha$ represents the position of a zero in the right half-plane. The system in which α is not equal to zero is called the non-minimum phase system and in its step response we can see an initial dip in value before a rise towards the reference value. An example is shown in Fig.1. Finally, the parameter n refers to the order of the system which, in general, dictates the dynamic of the system.

Aleksa Stojić is a teaching associate at the Signals and systems department at the School of Electrical Engineering, University of Belgrade, 73 Bulevar kralja Aleksandra, 11020 Belgrade, Serbia (e-mail: aleksa.stojic@etf.bg.ac.rs)

Goran Kvašček is a associate professor at the Signals and systems department at the School of Electrical Engineering, University of Belgrade, 73 Bulevar kralja Aleksandra, 11020 Belgrade, Serbia (e-mail: kvascev@etf.bg.ac.rs)

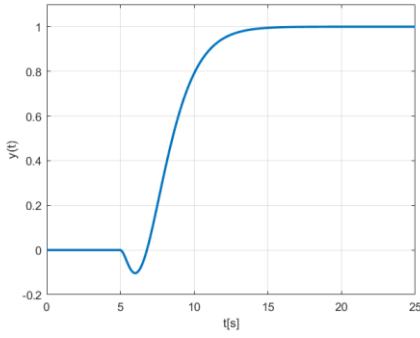


Fig. 1. Step response for a non-minimum phase system with parameters: $\theta_d = 5$, $n = 3$, $\alpha = 1$.

Convolutional neural networks are known for needing a large training set to perform well. These parameters can drastically change the output of the system which means that, in order to get a diverse dataset, we should try as many combinations as possible. We generated pure step responses for transfer functions which include parameter values given in Table I. Each signal has a duration of 25s and is sampled with a step size of 0.1s, making it 251 samples long.

TABLE I
TRANSFER FUNCTION PARAMETER VALUES

	Minimum value	Maximum value	Step size
θ_d	0	10	0.2
α	0	2	0.05
n	1	8	1

To every pure signal we generated, we added three types of noise. Given that real-life noise rarely fits into one specific probability density function, feeding different types of noise to our neural network seemed as a good way to have it generalize well after it is trained. The three types of noise added are Gaussian white noise, uniform noise and noise with Rayleigh distribution. All three types of noise have a standard deviation of 0.05. Probability density functions are given in equations (2), (3) and (4), respectively and shown in Fig.2.

$$p(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-0.5\left(\frac{x}{\sigma}\right)^2}; \sigma = 0.05 \quad (2)$$

$$p(x) = \begin{cases} 10/\sqrt{3}, & x \in \{-0.05\sqrt{3}, 0.05\sqrt{3}\} \\ 0, & x \notin \{-0.05\sqrt{3}, 0.05\sqrt{3}\} \end{cases} \quad (3)$$

$$p(x) = \frac{x}{\sigma^2} e^{-x^2/(2\sigma^2)}, \sigma = 0.05 \sqrt{\frac{2}{4-\pi}} \quad (4)$$

The expected value of the Rayleigh distribution given in equation (4) is a positive non-zero value, so to avoid having an offset in our noise we subtracted the expected value.

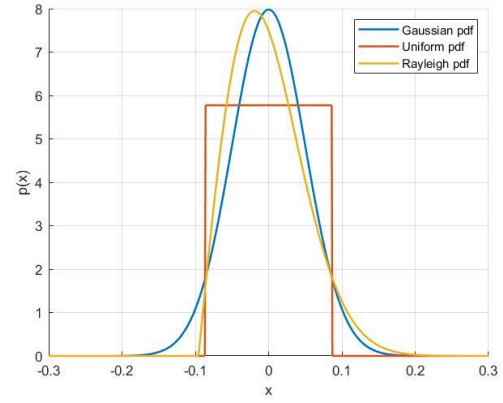


Fig. 2. Comparing three different noise probability density functions.

Final dataset has 50184 samples and 25% percent of it is intended for testing, while the rest is used for training. The test set consists of all step responses for systems of order 4 and 5, while all other orders belong to the training set. This will allow us to test our network on system orders that it has not encountered during training.

III. ARCHITECTURE AND TRAINING OF NEURAL NETWORK

When the term ‘denoising’ is mentioned in the context of neural networks, the autoencoder neural networks are usually suggested as an appropriate architecture. The idea behind them is to have an encoder part of the network, whose role is to extract features from the given input, followed by the decoder part, which tries to recreate the input from the features. Autoencoder networks are a type of unsupervised learning, because the desired output is the same as the input, therefore the input/output pair is not explicitly given during training. These networks served as an inspiration for this paper and creating the encoder-decoder architecture.

Encoder-decoder neural networks fall into the category of convolutional networks and the term ‘convolutional’ refers to the method of feature extraction. Let us consider that the input is a 2D matrix, such as a photograph. The dimensions of the photograph tend to be quite large and if we were to feed it to a regular fully-connected neural network, we would have too many trainable parameters. This would imply a huge amount of time to train and a lot of unnecessary computational resources. Also, each pixel itself might not carry as much information about the photograph as their relations do. So instead of treating individual pixels as separate entities, the idea behind convolutional neural networks is to have filters in the form of kernels (matrices with smaller dimensions) whose parameters are to be learned, which in convolution with the input would produce something more informational. Convolution is simply ‘sliding’ the kernel across the photo and multiplying it with appropriate submatrix of the photo. This way, instead of trying to learn a million neurons, the network has to learn filters which would extract meaningful features. The general advantage of neural networks is that it is not necessary for us to have insight into exactly what these filters are and what features they extract, but only how well the output is decoded using those features.

Convolution can also be done with 1D inputs, such as various forms of signals. The principal of convolution stays the same and the kernel is now a vector instead of a matrix. As previously mentioned, the aim of this project is to use a convolutional encoder-decoder network to denoise a step response of an open-loop system.

For this purpose we used *python* and the *tensorflow* library. It allows users to build a sequential neural network model by adding desired layers one-by-one.

For the encoder part of the network, we followed a standard structure of convolutional networks which implies alternatively adding convolutional layers and pooling layers. Convolutional layers learn kernel parameters and produce a convolution of the input with the kernel. It is common practice to have kernels with small dimensions, so in every convolutional layer we used kernels of size 5 (vectors of length 5). Every convolutional layer we added is followed by a *max-pooling* layer, whose role is to cut down the dimensions. It takes subvectors (of the specified size) of the input and produces an output which is the maximum value of the subvector. If the size is properly defined, we get the effect of having less computation needed, without any crucial loss in information.

The decoder part of the network has the inverse structure of the encoder part. Therefore, all convolutional layers are replaced with transposed convolutional layers. They perform deconvolution, which is an inverse operation to convolution. Opposite of *max-pooling* layers are *upsampling* layers. They once again make the dimensions larger by repeating the values of their input a specified amount of times. It is easy to notice that the output of the *upsampling* layers will not be exactly the same as the input of *max-pooling* layers.

The final architecture we chose is the simplest one that performed well. Details of its structure are shown in Table II. and Fig. 3.

TABLE II
ARCHITECTURE OF NEURAL NETWORK

Layer type	Kernel size	Output shape
Convolutional	5	(1, 247, 128)
Max-pooling	2	(1, 124, 128)
Convolutional	5	(1, 120, 64)
Max-pooling	2	(1, 60, 64)
Transposed convolutional	5	(1, 64, 64)
Upsampling	2	(1, 128, 64)
Transposed convolution	5	(1, 132, 128)
Upsampling	2	(1, 264, 128)
Convolutional	5	(1, 264, 1)
Cropping	/	(1, 251, 1)

The activation function for all layers is ReLU, except for the output layer whose activation function always depends on what the network is trying to predict. Since in our case we are doing regression, the adequate activation function is linear. The metric used to access the performance is mean-squared error, also suitable for regression.

When training a neural network it is necessary to make sure not to overfit it to the training data. Overfitting is a term used to describe the behaviour of a network that performs very well on training data, but has problems with data that it has not encountered during training, i.e. it generalizes poorly. One method to prevent overfitting we already implicitly implemented by making a large training dataset. Having a diverse dataset with many samples decreases the chances of overfitting. We also added a kernel constraint with value 3 to every layer, which stops parameters from rising above the given value. Popular method to avoid overfitting is also adding *dropout* layers after convolutional layers. Their role is to remove certain neurons with a specified rate and have them not impact the output. Adding *dropout* layers did not make the network perform better, so they were left out. 20% of samples in the dataset were used for validation. Finally, the network was trained for 7 epochs in mini-batches of size 32.

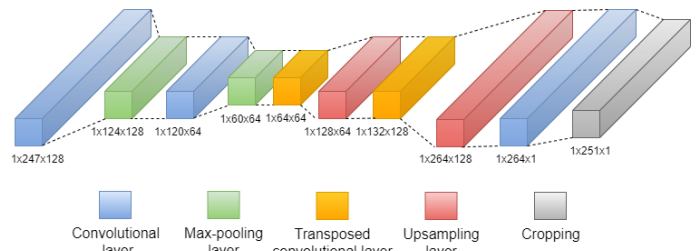


Fig. 3. Neural network architecture

The optimizer used is *Adam* optimizer, which is a type of a stochastic gradient descent algorithm. It is known for keeping track of the first and second momentum to provide quicker and more stable convergence. Its primary hyperparameters are learning rate, β_1 and β_2 . We used stochastic grid-search to find optimal hyperparameter values and they are: learning rate = 0.0005, $\beta_1 = 0.9$ and $\beta_2 = 0.99$.

IV. RESULTS

The network was first tested on a generated test set mentioned in chapter II. Average mean-squared error (mse) on the entire test set and on separate noise types are shown in Table III. In Fig. 5 the average values and standard deviations of mse for each of the noises are shown. Gaussian noise has the highest error, however all three have values bellow $9e^{-5}$. One more metric to asses the denoising process is comparing the signal-to-noise ratios (SNR) of signals before and after denoising. The average SNR for all types of noise was 53.5dB in the beginning. Signal-to-noise ratios after denoising are given in Table III.

TABLE III
MEAN-SQUARED ERROR FOR DIFFERENT NOISE TYPES

	MSE	SNR (dB)
Gaussian noise	$8.84e^{-5}$	87.8
Uniform noise	$8.14e^{-5}$	88.6
Rayleigh noise	$7.21e^{-5}$	89.9
Entire test set	$8.06e^{-5}$	88.8

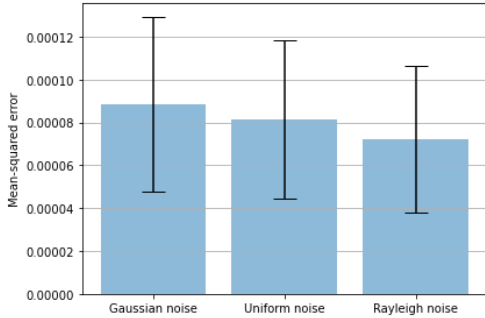


Fig. 4. Means and standard deviations of mean-squared errors for each noise

In Fig 5A. there are three representative step responses: one for each of the three types of noises. Based on observing, the output of the network seems to follow the signal dynamic well. Judging by the mean-squared error and Fig. 5A, Gaussian noise is the most difficult to remove, while Rayleigh noise seems the easiest to remove.

Next step was to test our network in a real-world environment to see how it handles a step response with noise which does not specifically match any distribution. The system we tested on was a handmade dryer that consists of a chamber on a metal surface and it has three platinum temperature sensors and one sensor which measures the airflow. The chamber is 1m long and has a heater of 400W and a ventilator on one side. Since airflow is not steady and turbulence occurs, the signal we captured is very noisy. In Fig 5B. there are three different original signals of airflow, as well as their denoised version. The results are satisfactory considering the amount of noise and the fact that the exact transfer function was not included in the training set.

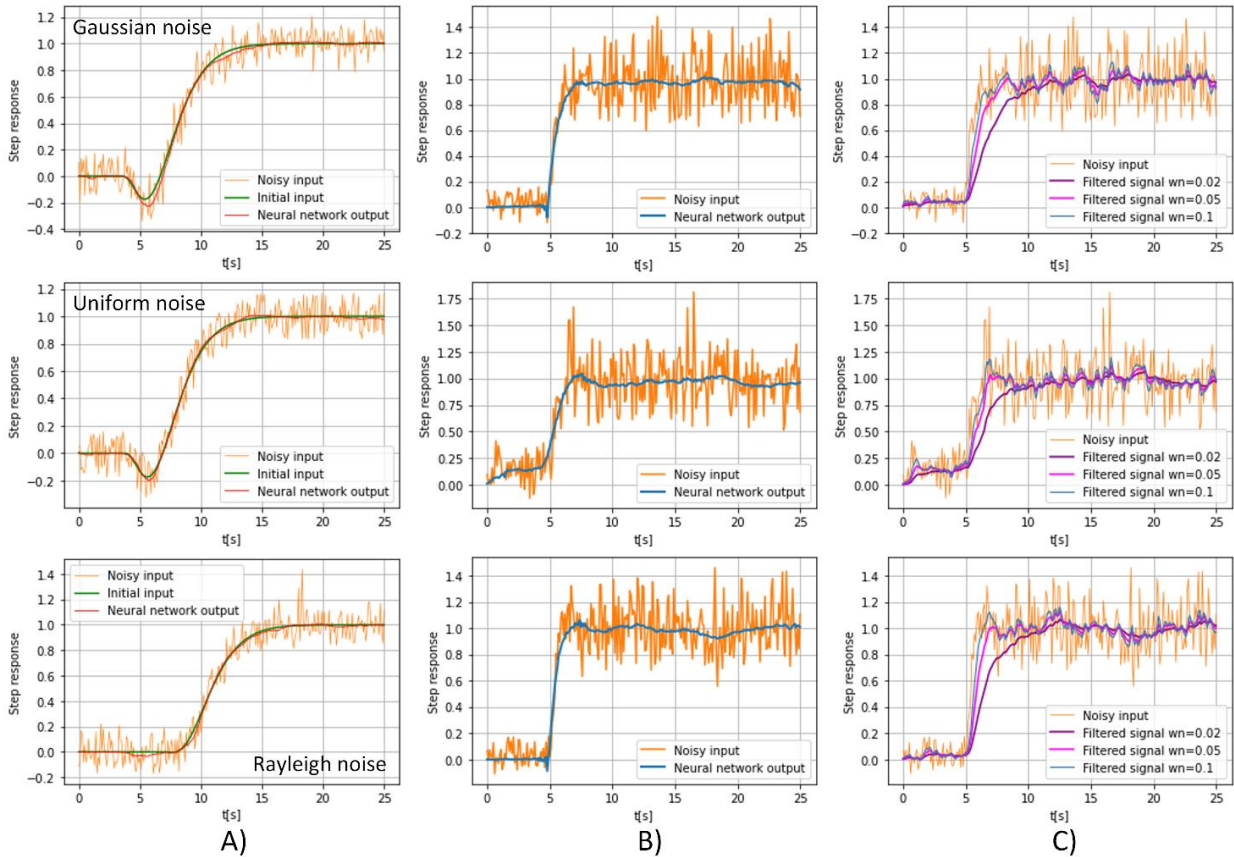


Fig. 5. Step responses before and after denoising. A) synthetic data, B) airflow signal after neural network, C) airflow signal filtered with Butterworth filter

To compare our results with classical filtering methods, we give the same noisy inputs but filtered with a first order Butterworth filter with different normalized frequencies in Fig. 5C. We see that in order to not compromise the dynamic of the signal, we have to settle for less noise removal. The neural

network we trained gives better results in terms of removing noise, but still following the dynamic. The comparison we did gives us an insight into why using a neural network for denoising could be a justified option. Conventional filtering methods often have to make sacrifices in terms of degrading

some other signal characteristics. Even though the structure of the neural network is not necessarily simple for all purposes, it is very flexible, and if we feed it the right amount of input data to learn from, the results it yields can be satisfactory in many ways.

V. CONCLUSION

The architecture of neural network used in this paper shows great potential for signal denoising. The proposed general form of transfer function in equation (1) also proved to be enough for the network to be able to follow the step response dynamic well. The next step in improving performance and extending it to other signals can move in a couple of directions. Having a training set with more noise distributions would also help the network generalize even better. Next possible extension could be having step responses with added disturbances at different moments in time in our dataset, as well as creating open-loop transfer functions in a more generalized form.

ACKNOWLEDGMENT

The paper was co-funded by the Ministry of Education, Science, and Technological Development of the Republic of

Serbia. This year's contract number is 451-03-68/2022-14/200103.

REFERENCES

- [1] Antczak, Karol. "Deep recurrent neural networks for ECG signal denoising." *arXiv preprint arXiv:1807.11551* (2018).
- [2] Arsene, Corneliu TC, Richard Hankins, and Hujun Yin. "Deep learning models for denoising ECG signals." *2019 27th European Signal Processing Conference (EUSIPCO)*. IEEE, 2019.
- [3] Zhu, Weiqiang, S. Mostafa Mousavi, and Gregory C. Beroza. "Seismic signal denoising and decomposition using deep neural networks." *IEEE Transactions on Geoscience and Remote Sensing* 57.11 (2019): 9476-9488.
- [4] Murali, Vineeth, and P. V. Sudeep. "Image denoising using DnCNN: an exploration study." *Advances in Communication Systems and Networks*. Springer, Singapore, 2020. 847-859.
- [5] Thakur, R. S., Yadav, R. N., & Gupta, L. (2019). State-of-art analysis of image denoising methods using convolutional neural networks. *IET Image Processing*, 13(13), 2367-2380.
- [6] Gondara, Lovedeep. "Medical image denoising using convolutional denoising autoencoders." *2016 IEEE 16th international conference on data mining workshops (ICDMW)*. IEEE, 2016
- [7] Zhang, Hang, et al. "Microseismic signal denoising and separation based on fully convolutional encoder-decoder network." *Applied Sciences* 10.18 (2020): 6621.
- [8] Goran Kvaščev. "Dalji razvoj i uporedna analiza procedura za eksperimentalno projektovanje i podešavanje industrijskih regulatora" Magister's thesis, 2005.

Single Screw Extruder Temperature Control Using PLC and HMI in Cable Production Process

Igor Kocić, Saša S. Nikolić, Aleksandra Milovanović, Darko Mitić, Petar Đekić and Nikola Danković

Abstract— In this paper it is developed and described the control software for the temperature regulation of the extruder zones with the mutual influence of zones. Extruder zone temperature control realized using Siemens PID_Temp block FB1132. PID_Temp block does not take into account the interaction of zones. This effect can only be taken into account if all zones are adjusted at the same time. The temperature process of the extruder was identified with an emphasis on the mutual influence of the zones. After that, a simulation was performed in Matlab and Simulink, and the results were experimentally verified using Siemens' LSim_LIB_V3_0_0 library and in a real process.

Index Terms— Extrusion, zone temperature identification, PID Temp controller, PID tuning.

I. INTRODUCTION

Extrusion [1, 8] of plastics and rubber is basically a continuous process. The starting raw material is usually in powder or granules, while in the case of rubber extruders it can also be in the form of a belt. The raw material moves from the basket under the action of gravity towards the feeder, and then enters the cylinder. A screw is placed in the cylinder, which transports the mass in advance under the action of the drive, the mass is further heated, homogenized and finally formed into the desired shape by passing through the tool. The machine in which this process takes place is called an extruder. Single-screw extruders have a huge application in polymer processing, they are simple constructions, have good technical performance, they are cheaper than multi-screw extruders.

Extruder temperature control is a problem that occurs very often in the industry. The main goal of temperature control is to maintain the set temperature. The extruder [1] consists of three basic units:

1. feed zone,
2. melting (compression) zone, and
3. pumping zone

In Section II we analysis the mutual influence of zones in

Igor Kocić, Saša S. Nikolić, Aleksandra Milovanović, Darko Mitić, and Nikola Danković are with the University of Niš, Faculty of Electronic Engineering, Department of Control Systems, Aleksandra Medvedeva 14, 18000 Niš, Serbia (e-mails: {igor.kocic, sasa.s.nikolic, aleksandra.milovanovic, darko.mitic, nikola.dankovic}@elfak.ni.ac.rs)

Petar Đekić is with The Academy of Applied Technical and Preschool Studies-Niš, Aleksandra Medvedeva 20, Niš, Serbia, e-mail: petar.djekic@akademijanis.edu.rs

terms of heat transfer from zone to zone on real extruder. Based on the recorded characteristics for zones 1, 2 and 3, a model of the mutual influence of the zones was made. Based on the zone influence model, the synthesis of the temperature controller was performed in Section III using the Siemens PID Temp controller and software was written in SCL for the simultaneous setting of the PID parameters of the zone controller. In chapter four, a check of the controller operation was performed using the Siemens simulation library LSim_LIB_V3_0_0.

II. EXTRUDER ZONE TEMPERATURE IDENTIFICATION

Figure 1 shows a diagram of a single-screw extruder [1] with nine zones. The basic parts of each extruder are hopper, barrel, screw and head with tool. Each zone of the extruder has its own heating heater, if necessary a cooling fan and a sensor that measures the current temperature value. In addition to measuring the temperature, the melting pressure is also measured at the place where the tool is located. Zones 1 to 5 have heater and cooler, zones 6 to 9 have only heater, cooling is done by the influence of ambient temperature.

It is noticed that zones 2, 3, ..., 8 are inner zones, and zones 1 and 9 are outer zones, and therefore the character of heat dissipation is different.

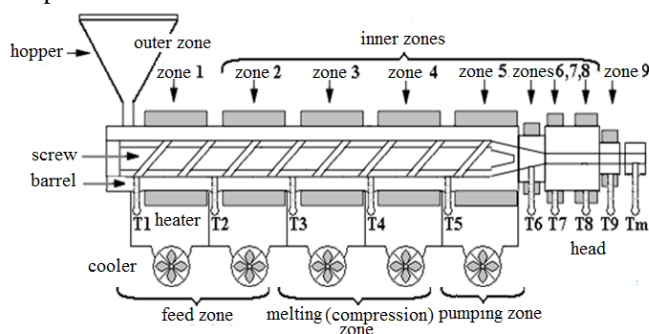


Fig. 1. Diagram of a single-screw PVC extruder

In determining the mutual influence of the zones, we started from the assumption that the control object in this case is the extruder cylinder is homogeneous and isotropic, which is correct because it is made of the same material. The influence of the heater and fan transmission function has not been considered.

In the general case, the temperature field of the extruder cylinder is non-stationary and can be described by the

following equation [4, 8]:

$$T = f(x, y, z, t) \quad (1)$$

When heating, the temperature at the measuring points changes over time (2), for isotropic materials, Fourier's law [8] of heat conduction applies (3), where Q is the amount of heat, S is the area, dQ/dS is the specific heat flux, and λ is the heat conduction coefficient:

$$\frac{\partial T}{\partial t} \neq 0 \quad (2)$$

$$dQ = -\lambda \nabla(T) dS dt. \quad (3)$$

If we take into account that the temperature changes over time, which is the case, we get a partial differential equation of parabolic type:

$$\frac{\partial T}{\partial t} = a \frac{\partial^2 T}{\partial x^2} \quad (4)$$

For further simplification, it is assumed that the temperature at the measuring points is equal to the temperature of the entire section, the temperature of the section is considered to be the temperature of the extruder zone. In this way, we consider sections to be isothermal surfaces. For the of further simplification, we will consider that the temperature is transferred only along the x axis ($x \gg y, z$ in practice the ratio of length and diameter of the cylinder in PVC extruders is large) Fig. 2 and that it is a stationary temperature field. At the time of temperature measurement at the zone boundaries, we can consider it constant.

If we now observe the part of the cylinder between the temperature measurement points of zone 1 (T1) and zone 2 (T2) and consider that these temperatures are constants, and then the following applies to the temperature field Eqs. (4) and (5) reduces to (6) where a is thermal diffusion coefficient:

$$\frac{\partial T}{\partial t} = a \left(\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} + \frac{\partial^2 T}{\partial z^2} \right) \quad (5)$$

$$\frac{\partial T}{\partial t} = 0, \frac{\partial T}{\partial y} = 0, \frac{\partial T}{\partial z} = 0 \quad (6)$$

$$a \frac{\partial^2 T}{\partial x^2} = 0 \quad (7)$$

$$\frac{\partial T}{\partial x} = const. \quad (8)$$

By integrating (7) and taking the boundary conditions $T(0)=T_1, T(L)=T_2$, where L is the distance from the temperature measurement point T1 to T2, we get that the temperature in x coordinates changes according to (8):

$$T(x) = T_1 - \frac{T_1 - T_2}{L} x \quad (9)$$

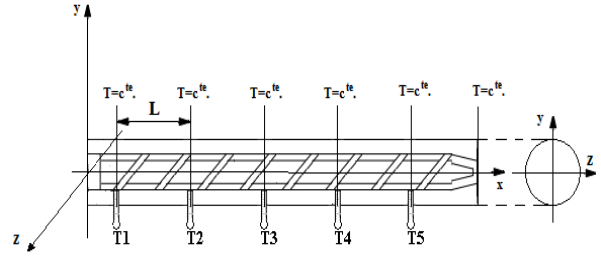


Fig. 2. Extruder cylinder (barrel) in x, y, z coordinate system observed along the x axis

When performing the above equation, we started from the assumption that $T_1 > T_2$, that only zone 1 is heated, and zone 2 is not. In order to determine the mutual influence of the zones, measured was applied on real Royle 4 1/2" extruder during heating and cooling, heating of zone 1 was done first, where the temperatures in zones 1 and 2 were measured, zone 2 was not heated. The ambient temperature at which the experiment was performed is $T_a = 17^\circ\text{C}$. Fig. 3 shows the influence of zone 1 on zone 2 as well as the temperature difference between zones 1 and 2.

The extruder was then cooled to ambient temperature. Only zone 2 was heated and the characteristics of zone 2 were recorded. Then the extruder was cooled, and zones 1, 2 and 3 were heated, and the temperature characteristics of zones 1, 2 and 3 were recorded at the same time so that the mutual influence of zones could be considered. Fig. 5.a shows the characteristics of only zone 2 which is heated by itself and Fig. 5.b shows the characteristics of zones 1, 2 and 3 which are heated at the same time. The extruder is heated to a temperature of 100°C , which corresponds to a time of about 25 min. Ambient temperature $T_a = 17^\circ\text{C}$.

The Matlab software package was used to draw the temperature characteristics of the characteristic zones.

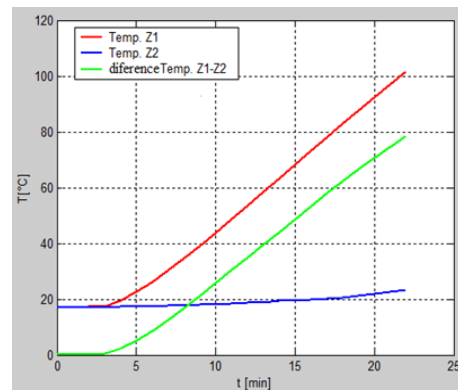


Fig. 3. Influence of Zone 1 to Zone 2

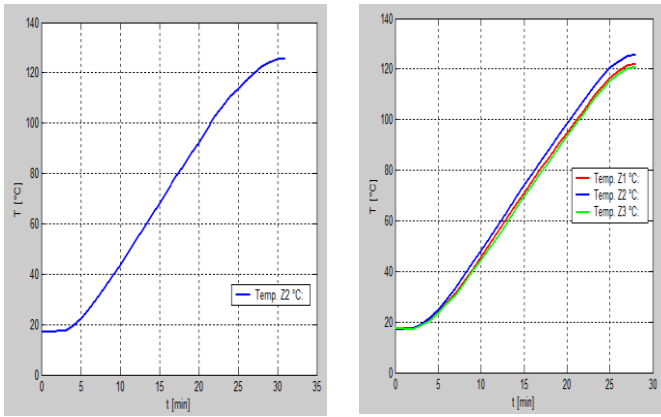


Fig. 4. Characteristics of heated

From Fig. 4 it is clearly seen that the heat transfer from zones 1 and 3 directly affects the temperature of zone 2. In Fig. 5.b we notice that the temperature of zone 1 is slightly higher than the temperature of zone 3, the reason is that zone 1 is the outer zone, and its adjacent zone is the feeder zone whose mass is less than zone 4. Zone 4 is not heated, so part of the heat from zone 3 passed to it.

In identifying and determining the process model, we started from the FOPDT model (9), which approximated the process, starting from the assumption that the transfer function of the object that controls astatism is zero. The parameters of the model are determined on the basis of the bounce response in the open coupling. The moment when the bounce trigger is brought is considered as the zero moment. Based on the recorded characteristics, the dead time T_u , the time constant T_u and the gain K were determined using the Kűpfműller method [10], so the model of the heating process for all three zones is described by the transfer function:

$$G_{ob}(s) = \frac{k_{ob}}{1 + sT_{ob}} e^{-sT_u} \quad (10)$$

To determine the parameters T_u and T_{ob} , a program was written in Matlab. Table gives the values first only for zone 2 when only it is heated, and then for zones 1, 2 and 3 that are heated at the same time.

TABLE I
HEATING ZONE 2 ONLY, AND ZONE 1, 2, 3

	only zone 2 is heated	zone 1, 2, 3 are heated at the same time		
	zone 2	zone 1	zone 2	zone 3
T_u [min]	5.5882	5.05769	4.214	5.117
T_{ob} [min]	21.098	16.654	16.214	17.153
K_{ob}	0.869	0.912	0.945	0.902

Based on the measurements, it is clear that the times T_u and T_{ob} for zone 2 are higher when only zone 2 is heated. It can also be seen that when zones 1, 2, 3 are heated, that zone 2 reaches the temperature the fastest, where the impact of zone 1 and zone 3 on zone 2 is observed.

FOPDT models of zones 1, 2 and 3 were used for the initial analysis of the process. For further analysis, an attempt was made to determine the PT_2 model based on the Streitz approximation through two points [3]. The PT_2 process model is given in (11) where K are the gain of the object, and T_2 are the time constants of the PT_2 model.

$$G_{ob}(s) = \frac{k_{ob}}{(1 + sT_1)(1 + sT_2)} \quad (11)$$

If the analysis of the ratio T_{ob}/T_u from table I according to table II ($T_{ob}/T_u < 9,65$) is performed, it is concluded that it is impossible to determine the characteristics of the process described by the PT_2 model according to Streitz's identification method [3]. After that, models were determined for all three zones based on the PT_N model (12), where T is the process time constant, n is the order of the PT_N model, T_{ob} is the object gain taken from the FOPDT model [3]:

$$G_{ob}(s) = \frac{k_{ob}}{(1 + sT)^n} \quad (12)$$

TABLE II
RELATIONSHIP BETWEEN FOPDT AND PT_N MODEL

n	1	2	3	4	5	6	7
T_{ob}/T_u		9.65	4.59	3.13	2.44	2.03	1.75
T_{ob}/T	1	2.718	3.695	4.463	5.119	5.699	6.226
T_u/T	0	0.282	0.805	1.425	2.1	2.811	3.549

Based on that, a model of the third order process is made for all three zones. To calculate the time constant of the PT_N model, a program was written in Matlab. The time constants T for the sweater zones are given in Table III.

TABLE III
VALUE OF THE TIME CONSTANT OF THE PT_N MODEL

	only zone 2 is heated	zone 1,2,3 are heated at the same time		
	zone 2 (n=3)	zone 1 (n=3)	zone 2 (n=3)	zone 3 (n=3)
T [min]	5.710	4.507	4.3887	4.643

Figure 5a shows the temperature difference $T_{z2} - T_{z1}, T_{z2} - T_{z3}$ and Fig. 5b analyzes the influence of zone 1 on zone 2 according to the equation $(T_{z2} - T_{z1})/T_{z2}$ and the influence of zone 3 on zone 2 according to the equation $(T_{z2} - T_{z3})/T_{z2}$.

The mutual influence of the zones during the simultaneous heating of zones 1, 2 and 3 is the greatest during the transition process when it moves up to 8%, after that it decreases and moves in up to 4% Fig. 5.

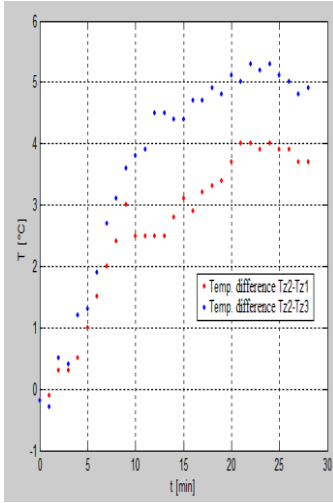


Fig. 5a.

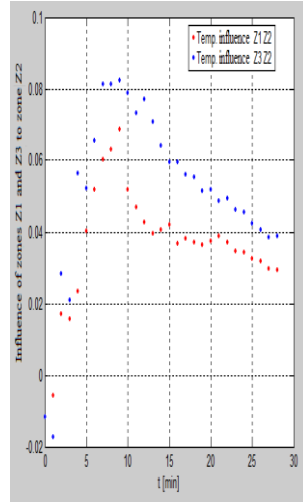


Fig. 5b.

Fig. 5. Influence of zones Z1 and Z3 on zone Z2

It is noticed that the temperature difference between zone 2 and zone 3 is greater than between zone 2 and zone 1, the reason is that zone 1 is an external zone, and adjacent to zone 3 is zone 4 which is not heated, so part of the heat is removed from zone 3 to zone 4. Fig. 6 shows a model of the mutual influence of zones.

The equation (13) [9, 10] applies to the simulation model of the impact of zones 1 and 3 on zone 2:

$$T_{z2}(s) = G_{obz2} (u_2 + k_{z12} T_{z1} + k_{z32} T_{z3}) \quad (13)$$

In general, applies equation (14), where $(i = 1, \dots, n)$ is denoted by observed zone:

$$T_{zi}(s) = G_{obzi} (u_i + k_{z(i-1)i} T_{z(i-1)} + k_{z(i+1)i} T_{z(i+1)}) \quad (14)$$

When it is taken into account that zone 2 affects zones 1 and 3, can applies equation (15) [9,10] where the coefficient K_{z2z2} takes into account that part of the energy of zone 2 is spent on heating zones 1 and 3.

$$T_{z2}(s) = G_{obz2} (u_2 + k_{z12} T_{z1} + k_{z32} T_{z3} - k_{z2z2} T_2) \quad (15)$$

When simulating the mutual influence of zone heating, all zones are identified separately. This means that it is necessary to heat only the observed zone, make its model, cool it and its neighboring zones. Repeat the procedure for all zones observed. In this way, the coefficient K_{z22} can be determined by comparing the influences when the heated zones are individually and all at once, so model (14) is applied for the simulation.

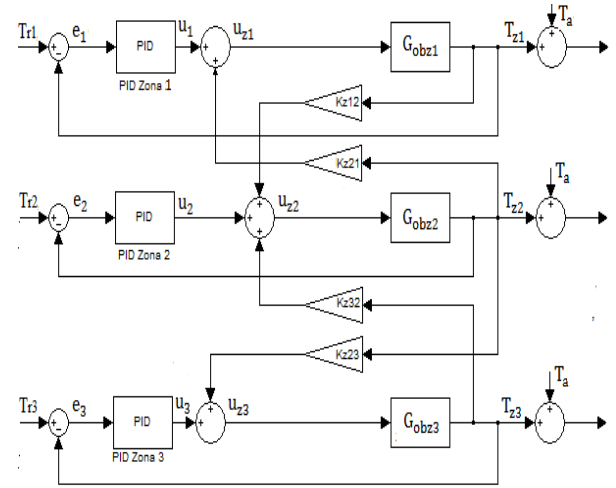


Fig. 6. Model of mutual influence of zones

III. SYNTHESIS TEMPERATURE CONTROLLER

Zone temperature controller is realized using a Siemens PID_Temp block FB1132 [10] with anti windup function and adjustable weight coefficients for proportional and integral action which is a continuous PID controller designed for heating and cooling applications, where y is output of PID controller, K_p proportional gain, b weight coefficient of proportional action, w set value, x measured value, T_i time constant of integral action, T_d time constant of differential action, and time delay differential action, c weighting coefficient of differential action.

$$y = k_p \left[(bw - x) + \frac{1}{T_i s} (w - x) + \frac{T_d s}{aT_d s + 1} (cw - x) \right]. \quad (16)$$

The PIDT controller (Siemens PID temperature controller block) (16) is integrated within the PID_Temp block [10]. The configuration and algorithm for setting the parameters of the PID_Temp block parameters is performed using the TIA Portal, where one instance is created for each control loop. For zone i , InstPIDTemp(i) is assigned a data block specifically DB(i) within which all settings for that instance are stored. Instances are called from the OB30 block (cyclic interrupt) Fig. 7 whose call time is set to 0.1s.

The PID_Temp block has two setup algorithms: pretuning and Fine Tune. For both tuning algorithms from TIA Portal software Siemens does not give a closer explanation of which criterion it uses to obtain parameters. Specifies only the necessary conditions for executing tuning algorithms.

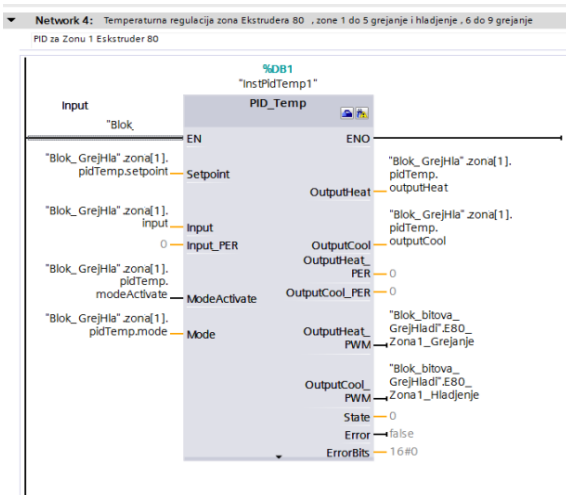


Fig. 7. Calling InstPIDTemp1 from OB30 block for zone 1

The disadvantage of individual adjustment of zones [11, 14] is reflected in the fact that the mutual influence of zones through thermal coupling is not fully taken into account. For this reason, software has been developed to simultaneously enable the setting of all zones from the HMI panel.

Before switching on the pre-tuning, the temperatures of the operating points of all zones are set, which must be higher than the current temperature, it is best to start from the ambient temperature. The inactive PID mode is selected for all zones. When performing pre-tuning, the heating pre-tuning is performed first, the cooling of the zones is temporarily switched off so that one of the zones does not adjust the heating parameters and its adjacent zones have completed the adjustment and switched to cooling mode. This creates a false temperature coupling. When the setting of the heating parameters is completed, the setting of the cooling parameters for all zones is switched on.

IV. EXPERIMENTAL RESULTS

Experimental results were obtained using Royle 4_1/2" extruder for PVC insulation and Siemens library LSim_LIB_V3_0_0 [15], Siemens PLC type S7 1500 3PN/DP, HMI 1500 Comfort series. The simulation of the interaction of zones 1, 2 and 3 was implemented in the software using the model of the interaction of zones by (14), Fig. 8.

The zone transfer functions are simulated by the Siemens function to simulate the third-order process LSim_PT3. Simulation software was written in SCL. Part of the Zone 2 simulation software is shown in Fig. 8.

The transfer function (12) was realized using the LSim_PT3 block. At the input of the block Block_Zone2 are the control signals from the PID block (InstPidTemp2) of zone 2 and the coefficients of mutual influence of the zones. In the function block Block_Zone2, a program is written that realizes the equation of influence of zone 1, 3 on zone 2 (13). The value thus obtained is fed to the input of the block LSim_PT3_Zone 2, at the output of which the value for the

temperature of zone 2 is obtained.

In the theoretical part, only heating is considered, and here the results for heating and cooling are given, Table IV. The simulation program is called from the OB30 object block with a selection period of 0.1 s. In Fig. 9 shows the response ratio of the calculated PID controller according to the PT_3 model and the controller obtained using the PLC simulation library for zone 1. The time to reach the desired temperature of the extruder zones is completely satisfactory, given the mass of the extruder cylinder.

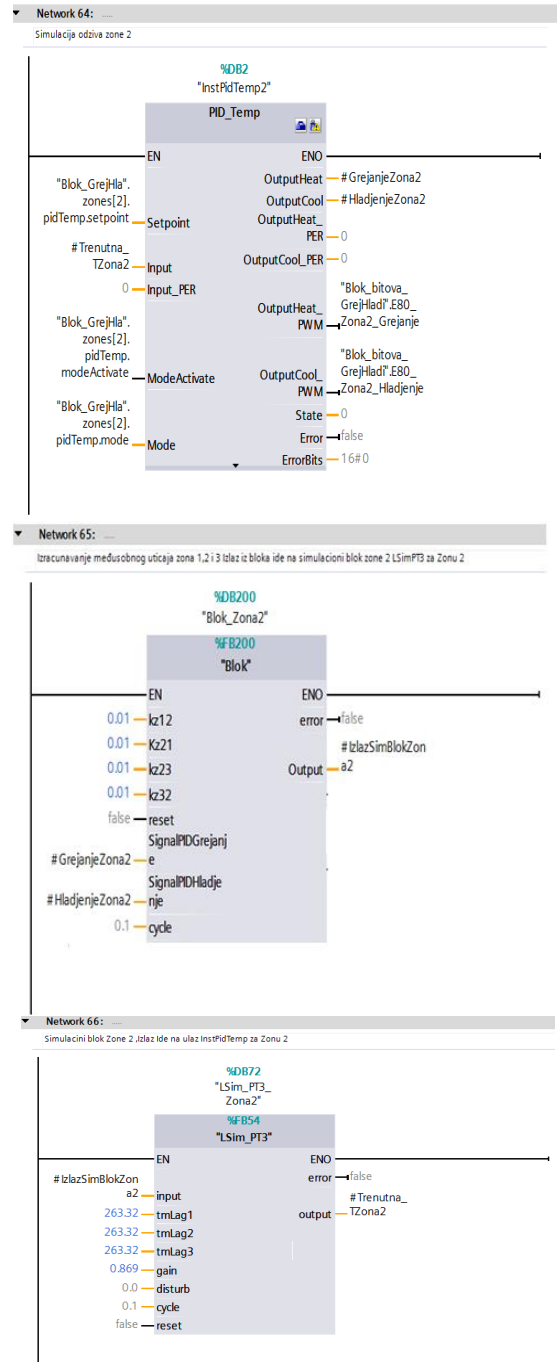


Fig. 8. Part of software for simulation of zone 2

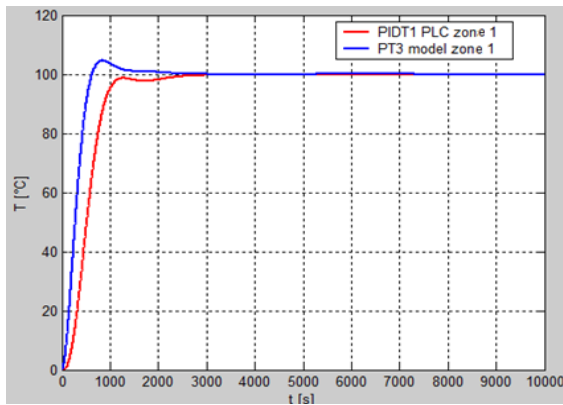


Fig. 9. Response characteristics of the calculated PID controller for PT₃ model and PIDT1 for Zone1 obtained by simulation using a PLC controller and a simulation library

TABLE IV
PID HEATING AND COOLING PARAMETERS OBTAINED USING SIMULATION ON PLC

	PID heating parameters		
	Zone 1	Zone 2	Zone 3
Kr	2.2338	2.1643	2.2562
Ti	816.98	756.866	875.896
Td	163.5282	158.118	167.8306
a	0.1	0.1	0.1
b	0.8	0.8	0.8
c	0	0	0
	PID cooling parameters		
	Zone 1	Zone 2	Zone 3
Kr	5.7454	5.7056	5.7557
Ti	317.126	273.652	359.381
Td	118.5414	114.6287	122.3443
a	0.1	0.1	0.1
b	0.8	0.8	0.8
c	0	0	0

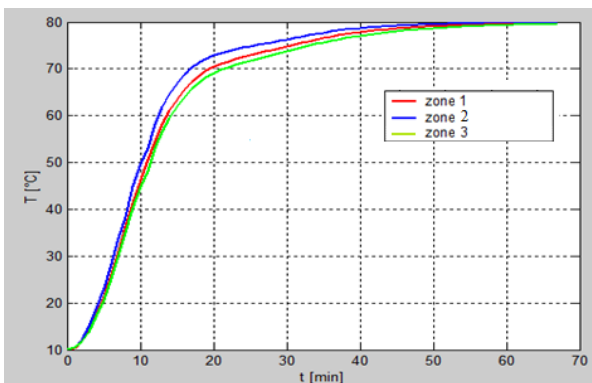


Fig. 10. Response characteristics zones 1, 2, 3 using simulation library

Figure 10 shows the real characteristics obtained by drawing

in Matlab, based on the data collected by recording the temperatures of all three zones by entering in the DB block of the PLC during the simulation using the PT₃ SIM library.

V. CONCLUSION

This paper presents zone temperature controller of extruder zones with special reference to the mutual influence of extruder zones. A model of mutual influence of zones is given, a simulation is done in TIA Portal.

It has been shown that the mutual influence of the zones during the simultaneous heating of the zones is greatest during the transition process when it moves up to 8%, after that it decreases and moves up to 4%.

The paper develops software for simultaneous adjustment of parameters of all extruder zones, as well as for adjustment of individual zones. The big advantage is that the software enables the adjustment of extruder zones from the HMI panel and not only from the TIA Portal software package.

REFERENCES

- [1] J.R. Wagner, E.M. Mount, and H.F. Giles, Extrusion: The Definitive Processing Guide and Handbook. USA, 2013.
- [2] A. Visioli, Practical PID Control, Springer, London, UK, 2006.
- [3] H. Uhbehauen and G. P. Rao, Identification of Continuous- time Systems, North-Holland Publishing Co., 1987.
- [4] B.R. Tibbetts and J.T.-Y. Wen, "Extrusion Process Control: Modeling Identification, and Optimization," IEEE Transactions on control Systems technology, vol. 6, no. 2, pp. 134-145, 1998.
- [5] J.G. Ziegler and N.B. Nichols, "Optimum settings for automatic controllers," Trans. ASME, no 64, pp 759-768, 1942.
- [6] P. Isermann, "Results on the simplification of dynamic process models," International Journal of Control, vol. 19, no. 1, pp. 149-159, 1973.
- [7] M. Maksimović, Tehnološke operacije, Tehnološki fakultet Banja Luka, 2007.
- [8] Multi-Zone Control with "PID_Temp", https://cache.industry.siemens.com/dl/files/463/109740463/att_993000/v1/109740463_PidTemp_MultiZone_DOC_V11_en.pdf
- [9] Function Manual: SIMATIC S7-1200, S7-1500 PID Control, https://support.industry.siemens.com/cs/document/100746401/pid-control-with-pid_compact-for-simatic-s7-1200-s7-1500?dti=0&lc=en-AR
- [10] S.L. Crabtree, M. A. Spalding, and C. L. Pavlicek, "Single screw extruder zone temperature selection for optimized performance," pp. 1410-1415, 2008.
- [11] J.G. Gonzales and M. Chimal, "Adaptive temperature controller for plastic extrusion process, Laboratoire d'Informatique, de Robotique et de Microélectronique de Montpellier (LIRMM), 2020.
- [12] C.-C. Yu, Auto Tuning of PID Controllers, Springer, London, 2006.
- [13] C. Abeykoon, K. Li, M. McAfee, P.J. Martin, and G.W. Irwin, "Extruder melt temperature control with fuzzy logic, IFAC Proceedings, vol. 44, no. 1, pp. 8577-8585, 2011.
- [14] Manual: 79047707_LSim_DOC_V3_0_0_en, <https://www.scribd.com/document/441305559/79047707-LSim-DOC-V3-0-0-en>
- [15] Function Manual: SIMATIC S7-1200, S7-1500 PID Control.

Upravljanje nivoom vode i pumpama upotrebom SMS poruka

Igor Kocić, Zoran Jovanović, *Elektronski fakultet u Nišu*

Apstrakt— U radu je opisano daljinsko upravljanje pumpama za dopremanje vode za potrebe proizvodnog procesa upotrebom Siemens Logo PLC kontrolera i odgovarajućih GSM modula. Radi povećanja pouzdanosti rada sistema komunikacija se obavlja dvojako upotrebom SMS poruka i žičane veze. Projektovan je i realizovan softver i hardver za obe vrste komunikacije, kao i softver za HMI panel kojim se vrši upravljanje i nadzor rada celog postrojenja.

Ključne reči— Logo PLC, CMR modul, SMS, sekvencijalno upravljanje

I. UVOD

U sistemima za upravljanje na daljinu tradicionalno se koristila prosta žičana veza. Osnovni nedostatak takvih sistema javlja se u slučaju prekida kabla u komandnom kolu.

U tom slučaju se u potpunosti gube informacije o sistemu upravljanja na udaljenoj lokaciji. U novije vreme žičana veza se menja radio vezom, čime je pouzdanost takvog sistema značajno povećana. Sa razvojem mobilne telefonije i interneta ostvarena je mogućnost upravljanja na daljinu sa velikom pouzdanošću.

Zbog potrebe povećanja pouzdanosti sistema u radu razvijen je upravljački softver i hardver za upravljanje nivoom vode u bazenima upotrebom SMS poruka. Pogodnim algoritmom sekvencijalnog upravljanja ostvareno je upravljanje pumpama i nivoom vode u rezervoaru.

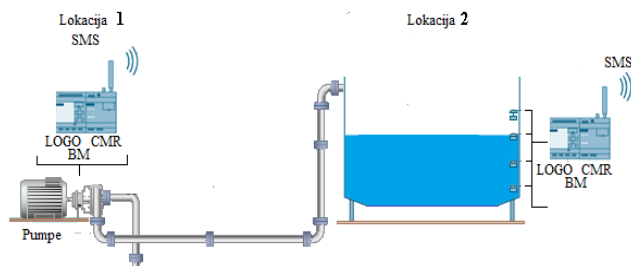
U slučaju otkaza mobilne mreže softver omogućava automatski prelaz na žičanu vezu i obrnuto. Postojeću žičanu vezu operater može uključiti ručno u svakom trenutku.

II. APLIKACIJA-OPIS HARDVERSKOG I SOFTVERSKOG REŠENJA

Na Sl.1. je dat šematski prikaz lokacije postrojenja, uređaja i opreme. Postrojenje je smešteno na dve udaljene lokacije. Glavni upravljački centar se nalazi u fabrici, a vodozahvat se nalazi na oko 2 km od fabrike. Lokacija bazena je nazvana: "Lokacija 2", a lokacija vodozahvata sa pumpnim postrojenjem nazvana je: "Lokacija 1".

U radu će se nadalje koristiti ovi nazivi. Sistem se sastoji od glavnog upravljačkog centra gde su ujedno nalaze bazeni sa sistemom za merenje nivoa vode i udaljenog pumpnog postrojenja u kome su instalirani frekvencijski regulatori kojima se obavlja pokretanje pumpi i regulacija pritiska u

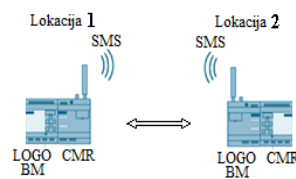
cevovodu i sistem za zaštitu od prevelikog pritiska na cevovodu.



Sl.1. Šematski prikaz procesa daljinskog upravljanja pumpnim postrojenjem

Kasnjenje upravljačkih signala nije od velikog značaja jer su bazeni za vodu velikog kapaciteta. Na obe lokacije se nalaze po jedan PLC kontroler serije LOGO sa komunikacionim modulima tip CMR2020 [2].

Na Sl.2 prikazana je topologija GSM mreže preko koje se obavlja komunikacija SMS porukama.



Sl.2. Topologija mobilne mreže, komunikacija SMS porukama

CMR modul za GSM/GPRS komunikaciju ima široku vrstu primena. Neki od mogućih slučajeva upotrebe su sledeći:

1. Mobilna bežična komunikacija bez LOGO PLC kontrolera,
2. Mobilna bežična komunikacija sa LOGO PLC kontrolerom,
3. Određivanje položaja (GPS).

U svaki CMR modul instalirane su SIM kartice i podešeni potrebni parametri dobijeni od strane mrežnog operatera. Svaki CMR modul povezan je na LOGO PLC [1]. LOGO PLC je sa CMR modulom povezan pomoću lokalne Ethernet mreže. CMR modul koristi omnidirekcionu antenu za mreže tipa GSM (2G), UMTS (3G) i LTE (4G) [2]. Time je omogućeno slanje i primanje SMS poruka povezanih sa događajima koji su programirani u LOGO PLC kontroleru, i pokretanje akcija i slanje informacija o statusu pomoću SMS poruka.

Podešavanje CMR modula obavlja se pomoću internet pretraživača unošenjem default adrese. Nakon inicijalnog podešavanja adrese su promenjene.

Igor Kocić – Elektronski fakultet, Univerzitet u Nišu, Aleksandra Medvedeva 14, 18000 Niš, Srbija (e-mail: igkocic@gmail.com).

Zoran Jovanović – Elektronski fakultet, Univerzitet u Nišu, Aleksandra Medvedeva 14, 18000 Niš, Srbija (e-mail: zoran.jovanovic@elfak.ni.ac.rs).

Podešavanje CMR modula se sastoji iz sledećih koraka:

1. Podešavanje parametara mobilne i lokalne Ethernet mreže.
2. Podešavanje parametara za SMS poruke.
3. Podešavanje korisnika i grupa korisnika.
4. Podešavanje poruka, akcija, definisanje ulaza, izlaza, događaja, alarmnih i poruka greške.
5. Definisanje poruka i akcija slanja i prijema poruka.

Na Sl.3 prikazan je glavni konfiguracioni ekran CMR modula. Svakom od modula dodeljena su imena i lokalne Ethernet adrese. Podešavanje mobilne mreže obavljeno je unosom PIN koda SIM kartice, broja kartice i aktivacijom interfejsa za komunikaciju preko mobilne mreže. Takođe moguća je komunikacija i sa uređajima koji su u drugoj zemlji uključanjem Roving opcije.



Sl.3. Glavni konfiguracioni ekran CMR modula

Primer sintakse SMS poruka dat je ispod u formi:

<Password>;LOGO=VM<Address>,<Value>,<Data type>.

Password predstavlja lozinku za komande tipa write "pisanje", čime je ostvarena dodatna zaštita od neovlašćenog slanja poruka.

LOGO=VM<Address> predstavlja lokalnu adresu u memoriji LOGO PLC, <Value> predstavlja vrednost koja se šalje ili prima, Data type je tip podatka koji se šalje ili prima.

SMS poruka komande upisa u LOGO PLC na adresu memorijske lokacije VM0 data je u formi:

SMSupis;LOGO=VM0,50,WORD.

Izvršavanjem ove komande u registar VM0 LOGO PLC kontrolera upisuje se vrednost 50.

Podešavanje korisnika i grupa korisnika obavljeno je unosom imena korisnika, lozinke, broja telefona korisnika i dozvolom za čitanje i slanje SMS poruka prema CMR modulu. Napravljene su dve grupa korisnika, prva dobija samo poruke o alarmima, a druga grupa pored alarmnih poruka ima dodeljene privilegije isključenja ili uključanja delova postrojenja u normalnom radu ili interventno.

Podešavanje poruka i akcija na događaje urađeno je dodelom ID broja poruke, imenom poruke i akcije kojom se šalje određena poruka.

Ako je nivo u bazenu nizak šalje se poruka čija je sintaksa:

SMSupis;LOGO=VM0,10,WORD.

Za slučaj greške komunikacije korisnicima se šalje poruka:

Text: ! 'Error: Stanica na Lokaciji 1 nedostupna.'

Ako je detektovana greška nivoa šalje se poruka :

Text: ! 'Error: Signal nivoa nije u redu. Senzor ili kabl je neispravan.'

Lokalne Ethernet adrese uređaja su redom:

1. PLC LOGO 8.2 Lokacija 2 bazen 192.168.0.103,
2. CMR modul 192.168.0.21,
3. PLC LOGO 8.2 Lokacija 1 vodozahvat 192.168.0.104,
4. CMR modul 192.168.0.11,
5. Touch HMI Panel KTP400 192.168.0.232

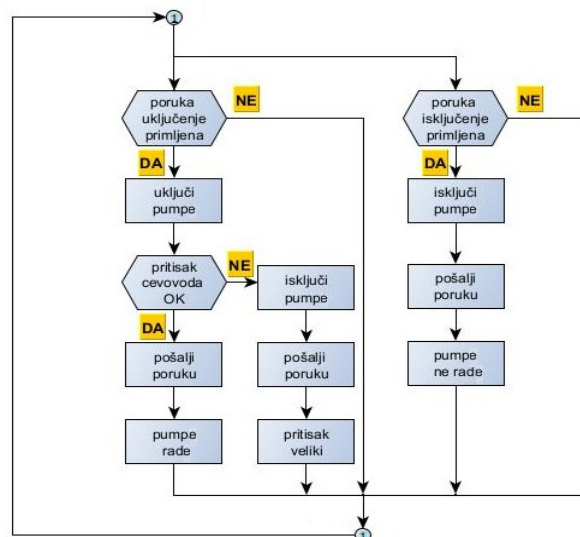
Na ovaj način omogućen je pristup mreži sa bilo koje od dve lokacije upotrebom mrežnih rutera.

Na lokaciji 2 upotrebom senzora za detekciju nivoa vrši se detekcija nivoa u više tačaka. Od interesa je minimalni i maksimalni nivo.

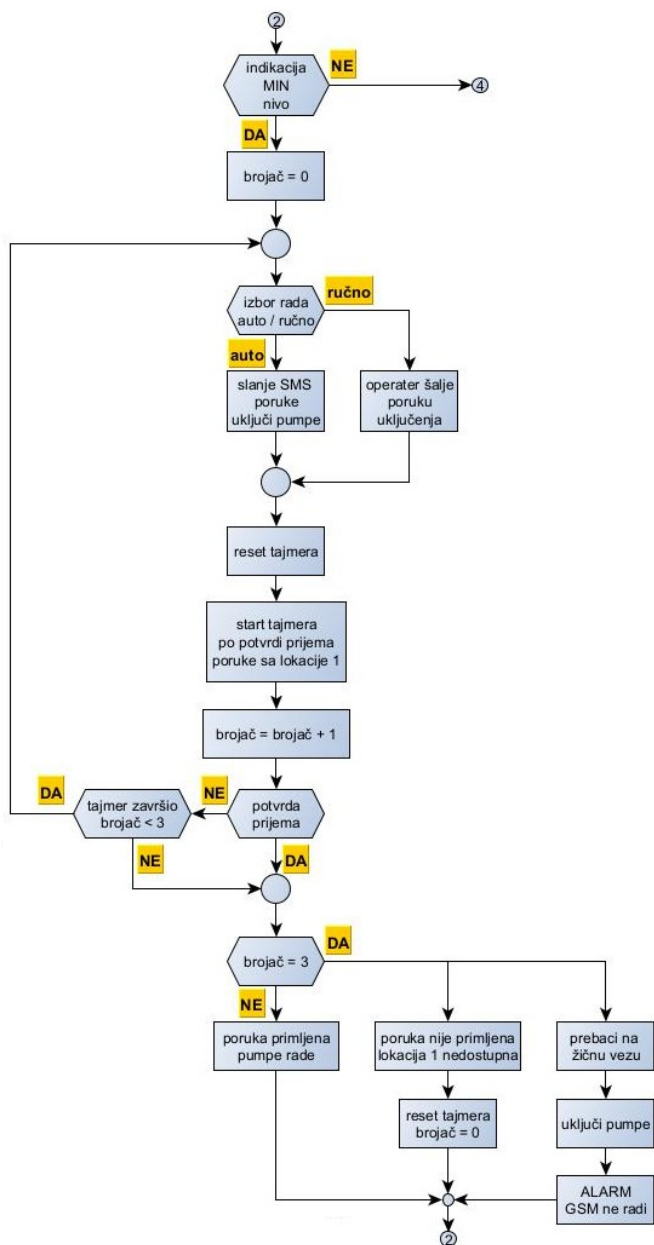
Kada nivo vode u tankovima padne ispod minimalne vrednosti vrši se uključanje pumpi za dopremanje vode.

Isključenje pumpi vrši se u dva slučaja:

1. radni uslov isključenja, kad je dostignut željeni nivo vode u tankovima,
2. havarijsko isključenje, koje se obavlja ako se pritisak u cevovodu poveća iznad određene vrednosti.



Sl.5. Deo algoritma rada udaljenog postrojenja upotrebom SMS poruka (lokacija 1)



SI.5. Deo algoritma rada postrojenja upotrebom SMS poruka pri indikaciji minimalnog nivoa (lokacija 2)

Pored automatskog rada, postrojenje može da radi i u ručnom režimu, kada se uključenje ili isključenje obavlja pomoću HMI panela i relejnom komandom ili SMS porukama koje šalju operateri koji za to imaju definisane privilegije. Postrojenje može da radi i upotrebom "stare" žičane veze, pod uslovom da je SMS komunikacija isključena.

Neki od ulaza, izlaza koji se koriste na LOGO PLC na lokaciji 2 su redom:

1. I1 minimalni nivo,
2. I2 nivo 2,

3. I3 nivo 3,
4. I4 maksimalni nivo,
5. I5 izbor rada ručno ili automatski,
6. I6 izbor SMS ili žičana veza,
7. I7 greška komunikacije,
8. Q1 signal greške komunikacije,
9. Q2 signal indikacije greške usled prevelikog pritiska

Neki od ulaza, izlaza koji se koriste na LOGO PLC na lokaciji 1 su redom:

1. I2 dozvola rada,
2. I4 senzor pritiska (presostat),
3. Q1 uključenje isključenje pumpe.

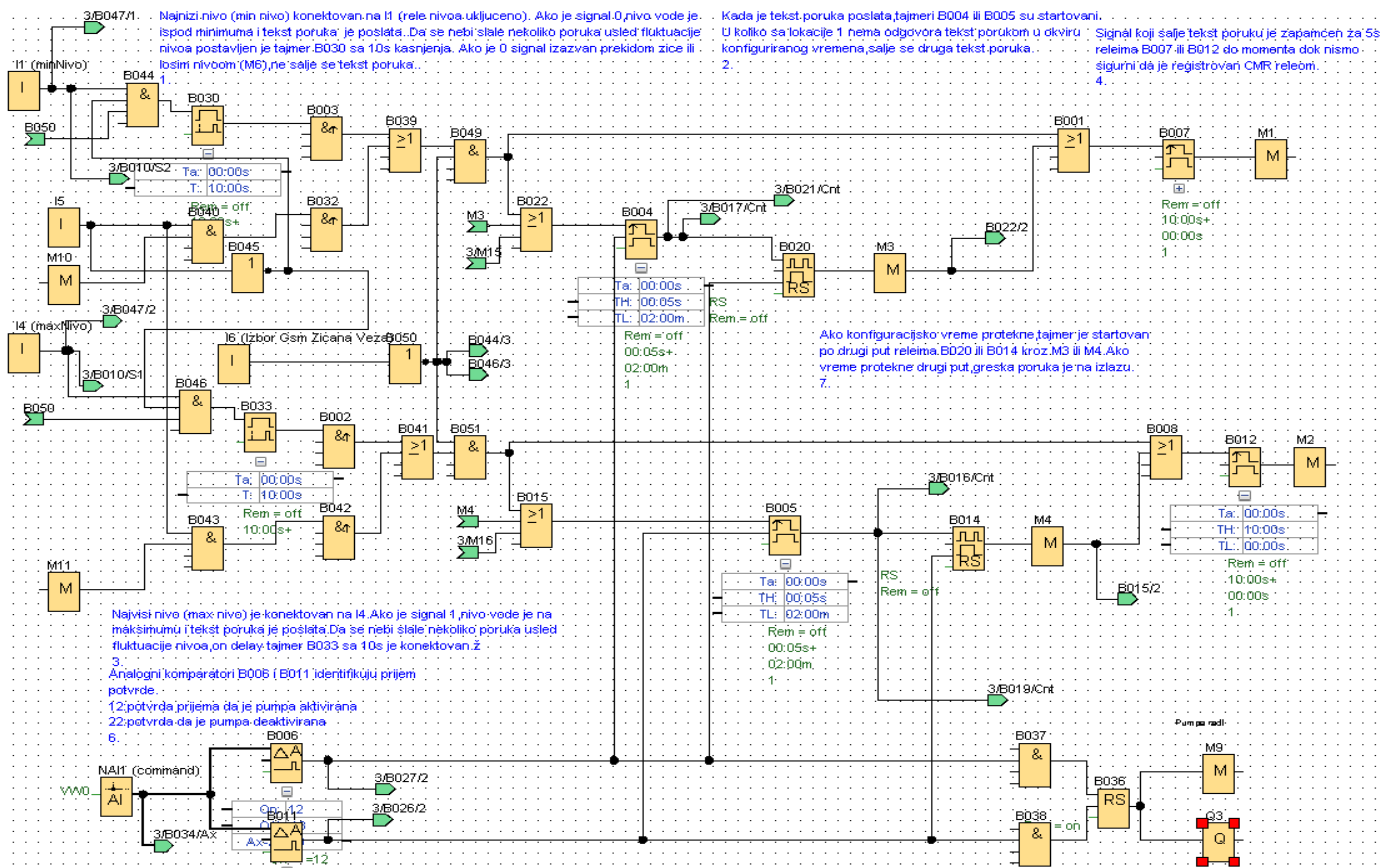
Na SI.4. prikazan je deo upravljačkog algoritma za uključenje, isključenje pumpi u udaljenom postrojenju. U slučaju povećanja pritiska u cevovodu senzor pritiska (presostat) isključuje pumpe.

Na SI.5. prikazan je deo upravljačkog algoritama postrojenja kada je nivo niži od minimalnog. U slučaju da se pojavi greška ili prekid u komunikacionom protokolu koji se trenutno koristi automatski se izvrši prelaz sa trenutnog na žičani tip komunikacije (žičanu vezu). Ako je trenutno aktivna SMS, prebacivanje se obavlja na žičanu vezu.

Softver za komunikaciju SMS porukama napisan je upotrebom Simensovog LOGO!Soft Comfort V8.2 alata za programiranje LOGO PLC kontrolera.

Na SI.6 dat je deo softvera koji se odnosi na slanje poruka za uključenje i isključenje pumpi u udaljenom pumpnom postrojenju (lokacija 1), kao i dela za prijem i obradu poruka koje šalje udaljeno postrojenje.

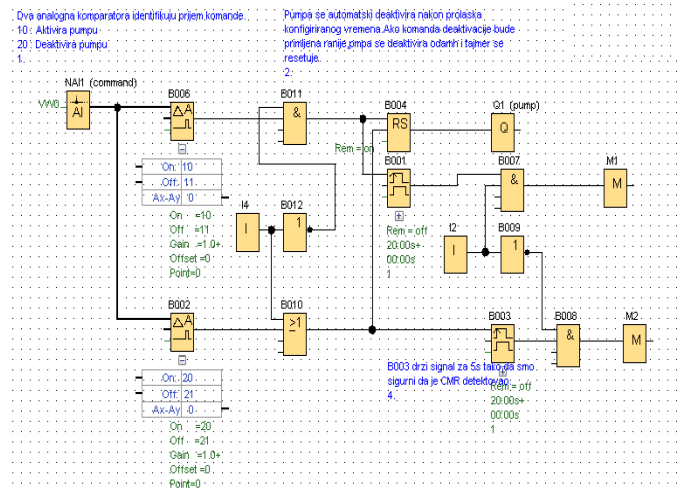
Kada nivo vode padne ispod minimalnog nivoa sa lokacije 2 šalje se prva SMS poruka. U slučaju fluktuacije nivoa usled talasanja vode a koja može uticati na sondu za signal minimalnog nivoa, signal aktivacije slanja poruke se filtrira upotrebom tajmera sa kašnjenjem od 10s instrukcije za detekciju rastuće ivice. Obezbeđeno da se svaki zahtev za slanje izvrši jedanput. Nakon slanja poruke aktivira se tajmer kojim je definisano maksimalno vreme potrebno da udaljeno pumpno postrojenje kojem je poslata poruka odgovori (60 s). Ovo vreme eksperimentalno je definisano tako da ne utiče na regulaciju nivoa. Ako vreme za odgovor istekne ista poruka se šalje i drugi put. Ukoliko odgovor sa udaljene lokacije 1 i u tom slučaju ne stigne, automatski se šalje poruka lokacija 1 je neaktivna. Poruka greške se ispisuje na HMI panel i šalje SMS poruka na brojeve koji su konfigurisani u CMR modulu. Komunikacija se automatski prebacuje na žičanu vezu. U režimu ručnog rada operater nadgleda rad postrojenja upotrebom HMI panela i odlučuje kada će da uključi ili isključi pumpe.



Sl.6. Deo softvera za slanje i prijem poruka za uključnje, isključenje i indikaciju rada pumpi pumpnog postrojenja

Deo softvera koji se koristi za uključnje isključenje pumpi na udaljenoj lokaciji 1 dat je na Sl.7. Poruka koja je primljena se poredi komparatorima B006 i B002. U koliko je primljena poruka koja se nalazi u registru VM0=10 na Logo PLC kontroleru se uključuje izlaz Q1 koji aktivira frekvencijski regulator za kontrolu rada pumpi. Odmah po prijemu vrši se slanje SMS poruke sa vrednosti '11' prema lokaciji 2 koji nosi informaciju da su pumpe startovane.

Ukoliko je primljena poruka VM0=20 izlaz Q1 se isključuje čime se rad frekventnog regulatora a samim tim i pumpi stopira. Po prijemu te poruke prema lokaciji 2 šalje se poruka sa vrednošću '22' pumpa isključena. Ukoliko se iz nekog razloga pritisak na cevovodu poveća senzor pritiska (presostat) detektuje nedozvoljenu vrednost pritiska, obavlja se havarijsko isključenje frekvencijskog regulatora pumpi. Prema lokaciji 2 šalje se poruka sa vrednošću '13', uključuje se alarm sa informacijom da je izvršeno havarijsko isključenje pumpi, pritisak povećan pumpe isključene. Takođe se šalje i SMS poruka operateru koji dalje preduzima potrebne mere.

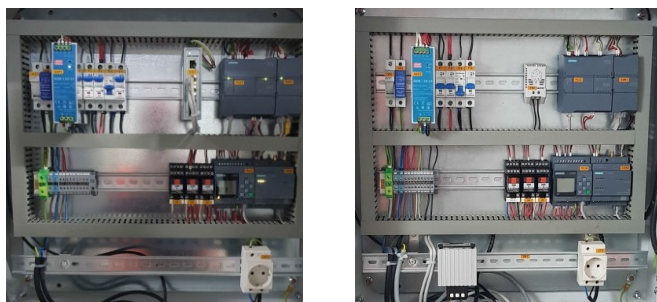


Sl.7. Deo softvera za uključnje isključenje pumpi u udaljenom pumpnom postrojenju

III. REALIZACIJA UPRAVLJAČKE APLIKACIJE ZA UPRAVLJANJE NIVOOM VODE U REZERVOARU

Sekvencijalno upravljanje nivoom vode u bazenu na lokaciji 2 obavljeno je izradom dva upravljačka ormana za prenos poruka upotrebom SMS poruka i ormana sa relejnom logikom, kao i ormanom sa frekventnim regulatorom koji upravlja radom pumpi.

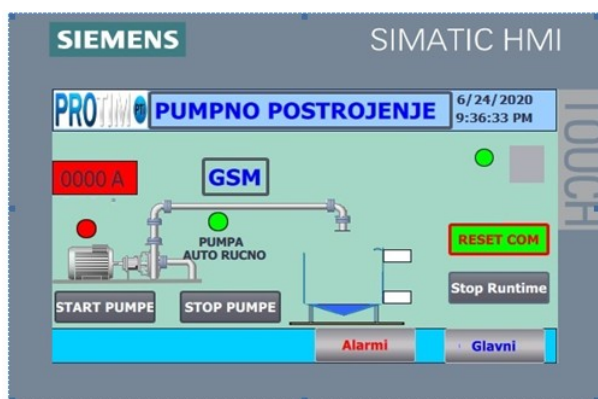
Na Sl.8 prikazan je upravljački orman na lokacijama 1 i 2 .



Sl.8. Upravljački ormani za upravljanje nivoom vode

U svakom od njih je postavljen po jedan LOGO PLC sa komunikacionim CMR modulom i S7 1200 PLC kontrolerom koji upravlja radom postrojenja pomoću žičane veze.

Na Sl.9 prikazan je HMI panel za nadzor postrojenja.



Sl.9. Prikaz HMI panela pri SMS komunikaciji

IV. ZAKLJUČAK

Za realizaciju ovakve vrste upravljačkih zadataka godinama se koristila samo žičana veza i prosta relejna logika. Izrade žičane veze za velike udaljenosti obično se realizuje u momentu izrade cevovoda, kada se paraleleno sa cevima polažu komunikacioni i napojni kablovi do udaljenije lokacije. U slučaju otkaza, oštećenja komunikacionih veza potrebno je ponovo otkopati deo trase kako bi se otkrio prekid u

komunikaciji, što iziskuje značajna sredstva. U novije vreme sve češće se koristi komunikacija SMS porukama upotrebom mreže mobilne telefonije. U radu je pored softvera i hardvera razvijenog za komunikaciju SMS porukama razvijen i deo softvera i hardver za komunikaciju preko žičane veze. Time je mogućnost otkaza sistema smanjena.

LITERATURA

- [1] Logo_e, Manal Edition 06/2003, https://www.google.com/url?esrc=s&q=&rct=j&sa=U&url=https://cache.industry.siemens.com/dl/files/461/16527461/att_82564/v1/Logo_e.pdf&ved=2ahUKEwjm-qQJz9n3AhUBlmoFHboXDvoQFnoECAkQAg&usq=AOvVaw07CoQzjU-TKxPeO7uedRlp
- [2] SIMATIC NET LOGO! - Industrial Ethernet, LOGO! CMR2020, LOGO! CMR2040, Operating instruction, 05/2015, C79000-G8976-C356-02, https://cache.industry.siemens.com/dl/files/268/103657268/att_850514/v1/BA_LOGO-CMR2020-CMR2040_76_en-US.pdf
- [3] SIMATIC, S7-1200 Programmable controller, System Manual, V4.3.0 02/2019, A5E02486680-AM, https://cache.industry.siemens.com/dl/files/129/109764129/att_974298/v1/s71200_system_manual_en-US_en-US.pdf
- [4] Hans Berger, Automating with SIMATIC S7-1200, Publicis, 2nd edition (June 10, 2013)
- [5] J.F. DiMarzio, Rutiranje, Kompjuter biblioteka, 2002
- [6] M. Fallahnejad, A new algorithm for connecting between LOGO and labview software, January 2012
- [7] G.Kozoris, M. Papoutsidakis, A. Chatzopoulos, Electropneumatic Positioning System Control With The Legendary LOGO! PLC, December 2017
- [8] L. Fang, Application of PLC technology in electrical engineering and automation control, January 2017

ABSTRACT

The paper describes the remote control of water supply pumps for the needs of the production process using Siemens Logo PLC controllers and appropriate GSM modules. In order to increase the reliability of the system, communication is performed in two ways using SMS messages and wired connection. Software and hardware for both types of communication were designed and implemented, as well as software for the HMI panel which manages and supervises the operation of the entire plant.

Water level and pump control using SMS messages

Igor Kocić, Zoran Jovanović

Application of Subtractive Clustering in Data Processing

Boris Barišić, Aleksandra Krstić, Sanja Vujnović, Željko Đurović

Abstract—The problem of data clustering is still in development and various approaches to solving it are being proposed, all of which have different success rates. One of the nonparametric clustering methods is subtractive clustering. The success of this algorithm largely depends on tuning its parameters. In this paper we give a theoretical analysis of different suggestions for choosing their values. Based on probability theory, we examined the impact of dimensionality and number of samples on the clustering radius. By conducting a controlled experiment with known sample distributions, the performance of this algorithm with suggested parameters is tested, as well as its robustness.

Index Terms—subtractive clustering, parameters tuning, classification, data processing

I. INTRODUCTION

For a large number of classification problems we do not have adequate a priori knowledge and therefore we are not able to generate an appropriate training set. Starting with Charles Darwin and his systematization of animals and plants into genera, species, families etc. up until the development of systems based on various forms of artificial intelligence, the man has attempted to improve clustering techniques.

Nonparametric clustering methods do not consider optimization criteria or data distributions. They are based on implementing different ways to locate ‘valleys’ or ‘hills’ in the probability density function of the data as a natural border between different classes.

In 1992. Yager and Filev suggested *mountain clustering* as one of the techniques [1]. The method is based on dividing the entire data space into a dense grid of small hypercubes whose vertices are potentially cluster centers. The potential i.e., the *mountain* function is then calculated for each vertex as a measure of sample density in its surrounding. Clearly the potential for the vertex to be a cluster center increases with the number of samples surrounding it. The core idea of this method is the following: after finding the first cluster center, potentials of all vertices are reduced inversely proportional to the distance from the vertex to the cluster center. For vertices closer to the center, the potential reduces more. The next cluster center is chosen as the vertex with the highest potential (after reduction). This method of finding cluster centers repeats until the potential

in all vertices falls beneath a certain threshold.

Even though it was imagined as a very simple method, its numerical complexity grows exponentially with sample dimensionality due to a large number of hypercubes in the grid. In 1994. Chiu suggested a modification of this algorithm called subtractive clustering [2].

II. THEORETICAL ANALYSIS

A. Algorithm

The idea of *subtractive* clustering is that every sample in the dataset can be a cluster center. Due to this starting assumption of considering only given samples as cluster centers, the complexity of this algorithm is practically linear.

Samples are assigned a certain density based on which the cluster centers are found during the iterative procedure. Aside from considering a smaller dataset than *mountain* clustering, calculating the density function implies the squared distance between samples, so there is no need for determining the square root.

Based on the description given in [2], the algorithm is comprised of a few steps:

Step 0: For each of N samples we calculate the value of the initial density D_i^1 according to equation (1),

$$D_i^1 = \sum_{j=1}^N e^{-\frac{\|x_i - x_j\|^2}{(r_a/2)^2}}, i = \overline{1, N} \quad (1)$$

where r_a is a positive constant called the clustering radius.

Step 1: Based on initial density values we determine the first cluster center X_c^1 :

$$X_c^1 = \arg\{\max_{i=\overline{1, N}} D_i^1\} \quad (2)$$

Step 3: Having found the first cluster center, we start the iterative procedure of finding other cluster centers. Since the first center is already found, let the iteration counter start at $k = 1$.

Step 4: We increment the counter to $k = k + 1$ and eliminate the influence of samples near the previously found center by modifying their density function:

$$D_i^k = D_i^{k-1} - D_c^{k-1} e^{-\frac{\|x_i - X_c^{k-1}\|^2}{(r_b/2)^2}}, i = \overline{1, N} \quad (3)$$

Boris Barišić – School of Electrical Engineering, University of Belgrade, Bulevar Kralja Aleksandra 73, 11020 Belgrade, Serbia (e-mail: boris.barisic@etf.bg.ac.rs).

Aleksandra Krstić – School of Electrical Engineering, University of Belgrade, Bulevar Kralja Aleksandra 73, 11020 Belgrade, Serbia (e-mail: amarjanovic@etf.bg.ac.rs).

Sanja Vujnović – School of Electrical Engineering, University of Belgrade, Bulevar Kralja Aleksandra 73, 11020 Belgrade, Serbia (e-mail: svujnovic@etf.bg.ac.rs).

Željko Đurović – School of Electrical Engineering, University of Belgrade, Bulevar Kralja Aleksandra 73, 11020 Belgrade, Serbia (e-mail: zdjurovic@etf.bg.ac.rs).

D_i^k is the new density function value, and D_i^{k-1} is the previous one. D_c^{k-1} is the maximum density from the previous iteration, and X_c^{k-1} is the cluster center found also in the previous iteration. Parameter r_b represents a new clustering radius.

Step 5: Based on the newly modified values of density functions we choose a k^{th} cluster center X_c^k :

$$X_c^k = \arg\{\max_{i=1, \dots, N} D_i^k\} \quad (4)$$

Step 6: We check whether D_c^k i.e., the density of samples in radius r_b of k^{th} cluster center satisfies:

$$D_c^k \leq \delta D_c^1 \quad (5)$$

If the given condition is not satisfied, we go back to step 4, and if it is, the algorithm ends. Parameter δ is a positive value smaller than 1 and it is called the clustering threshold.

B. Parameter tuning

The success of clustering largely depends on choosing the values for r_a , r_b and δ . Clustering radiuses r_a and r_b should in some way incorporate the information on how samples are scattered around cluster centers.

A great number of scientists and engineers have put their knowledge and experience into determining clear and straightforward recommendations for choosing the values of parameters involved in clustering. Unfortunately, most of them so far have turned out to be inefficient and inapplicable in most cases. In [3] it is suggested and in [4] analyzed that the parameter r_a should be chosen according to:

$$r_a = \left\{ \frac{1}{4} [\max\{\|X_i - X_j\|\} + \min\{\|X_k - X_l\|\}] \right\}^\beta \quad (6)$$

where $i, j, k, l = \overline{1, N}$, and N the total number of samples available. Parameter β should serve as amortization for extreme values of maxima which come from potential outliers [3].

We will perform an analysis on how the theoretical probability density function (pdf) of r_a changes according to N and dimensionality n . First, we consider a one-dimensional case ($n = 1$). Let samples from all clusters have a Gaussian joined probability density function $\mathcal{N}(m, \sigma^2)$. For the sake of simplicity, we will consider the minimum value to be neglectable in respect to the maximum value, as well as the parameter $\beta = 1$. Samples X_i and X_j then become independent identically distributed random variables, and parameter r_a becomes a random variable R_a for which the following holds:

$$R_a = \max\{|X - Y|\} \quad (7)$$

Random variable $Z = X - Y$ will then also have a Gaussian pdf with parameters $\mathcal{N}(0, 2\sigma^2)$, as it is a subtraction of two Gaussian variables. The absolute value of this variable $U = |Z|$ will have a cumulative distribution function (cdf):

$$F_U(u) = P(U \leq u) = P(|Z| \leq u) = P(-u \leq Z \leq u) u_H(u) \quad (8)$$

i.e. the following will hold:

$$F_U(u) = (2F_Z(u) - 1)u_H(u) \quad (9)$$

where $u_H(u)$ represents a Heaviside unit step function.

Now we can easily obtain the probability density function of the random variable U :

$$f_U(u) = 2f_Z(u)u_H(u) \quad (10)$$

If the total number of samples in our dataset is N , then there are $N_U = \binom{N}{2} = \frac{N(N-1)}{2}$ values which can be calculated as $U = |X - Y|$ and let these values be U_1, \dots, U_{N_U} . Let us form an array $U_{(1)}, \dots, U_{(N_U)}$, whose elements are variables U_1, \dots, U_{N_U} in a non-declining order. $U_{(1)}$ is the minimum, and $U_{(N_U)}$ the maximum calculated value. If we want to determine the cdf $F_{U_{(k)}}(u)$ for $U_{(k)}$, we will notice that the event of $\{U_{(k)} \leq u\}$ occurs if and only if the k^{th} value is not bigger than u , meaning that at least k of N_U random variables U_1, \dots, U_{N_U} have a value less than or equal to u . Imagine we have N_U Bernoullie's experiments, with every experiment testing whether the event $\{U_{(k)} \leq u\}$ (success) occurred or not [5]. The probability of success is equal to $P(U_{(k)} \leq u) = F_U(u)$, and the probability of at least k successes occurring in N_U experiments is:

$$F_{U_{(k)}}(u) = P(U_{(k)} \leq u) = \sum_{i=k}^{N_U} \binom{N_U}{i} F_U^i(u) (1 - F_U(u))^{N_U-i} \quad (11)$$

Since the object of our analysis is the maximum value of $|X - Y|$ i.e. $R_a = U_{(N_U)}$, by replacing k with N_U we get:

$$F_{R_a}(r_a) = (F_U(r_a))^{\frac{N(N-1)}{2}} \quad (12)$$

The pdf for the parameter R_a is:

$$f_{R_a}(r_a) = \frac{N(N-1)}{2} (F_U(r_a))^{\frac{N(N-1)}{2}-1} f_U(r_a) \quad (13)$$

Let us assume now that our samples are n -dimensional. Let samples X_i and X_j be independent identically distributed vectors whose pdf is $\mathcal{N}(\mathbf{M}, \mathbf{\Sigma})$. We will additionally assume that their variances along all dimensions are equal and not correlated, i.e. that the covariance matrix has the form $\mathbf{\Sigma} = \sigma^2 \mathbf{I}$, where \mathbf{I} is the identity matrix.

The Euclidian norm $V = \|\mathbf{X} - \mathbf{Y}\|$ is calculated as

$$V = \sqrt{\sum_{i=1}^n (X_i - Y_i)^2} = \sqrt{\sum_{i=1}^n Z_i^2} \quad (14)$$

where Z_i is the random variable $Z_i = X_i - Y_i$. We previously showed that the distribution of Z_i will be $\mathcal{N}(0, 2\sigma^2)$, which means that $\frac{Z_i}{\sqrt{2}\sigma}$ will have a Gaussian pdf $\mathcal{N}(0, 1)$. We can determine the distribution of random variable

$$V = \sqrt{2}\sigma \sqrt{\sum_{i=1}^n \left(\frac{Z_i}{\sqrt{2}\sigma}\right)^2} \quad (15)$$

using the results known from probability theory, which state that the square root of the sum of squares of n independent identically distributed variables with distribution $\mathcal{N}(0,1)$ will have a χ distribution with n degrees of freedom [6]. Therefore, the cdf of random variable V will be:

$$F_V(v) = \frac{\gamma\left(\frac{n}{2}, \frac{v^2}{4\sigma^2}\right)}{\Gamma\left(\frac{n}{2}\right)} u_H(v) \quad (16)$$

where Γ is the gamma function, and γ is the lower incomplete gamma function. The pdf of random variable V is:

$$f_V(v) = \frac{v^{n-1}}{2^{n-1}\sigma^n\Gamma\left(\frac{n}{2}\right)} e^{-\frac{v^2}{4\sigma^2}} \cdot u_H(v) \quad (17)$$

Results given in equations (12) and (13) also hold for the distribution of the maximum value of random variable V with arbitrary dimensionality n .

Fig. 1 shows the probability density function of random variable R_a for different dimensionalities n and different number of samples N . We can see that the clustering radius can be affected by changes in both parameters. Larger dimensionality n results in more additions when calculating the norm, which expectedly also results in larger values of clustering radius. On the other hand, having more samples (larger N) increases the chance of extrema appearing, i.e. the chance of having samples which are far away from each other, which has a bigger clustering radius as a result.

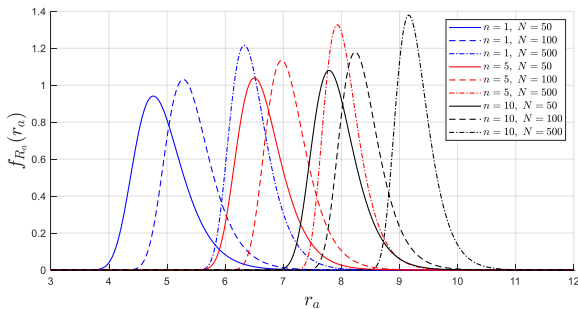


Fig. 1. Probability density function of random variable R_a which represents maximum of Euclidian norm between two samples.

It is also meaningful to analyze how the mathematical expectation m_{R_a} and variance $\sigma_{R_a}^2$ of the pdf of variable R_a change depending on sample number N and dimensionality n . Both statistics have been calculated using numerical integration of pdf in equation (13) and the results are shown in Fig. 2 and Fig. 3. Mathematical expectation and variance increase with the increase of n . On the other hand, larger number of samples increases the mathematical expectation, but decreases variance.

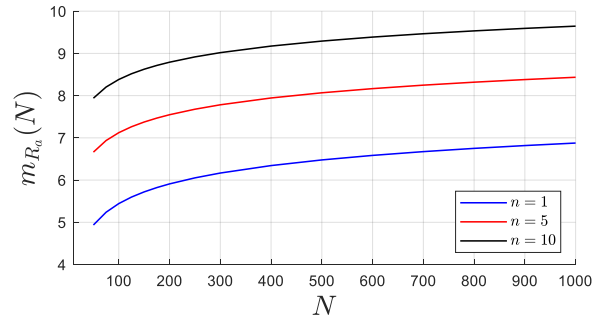


Fig. 2. Mathematical expectation m_{R_a} of random variable R_a depending on dimensionality n and number of samples N .

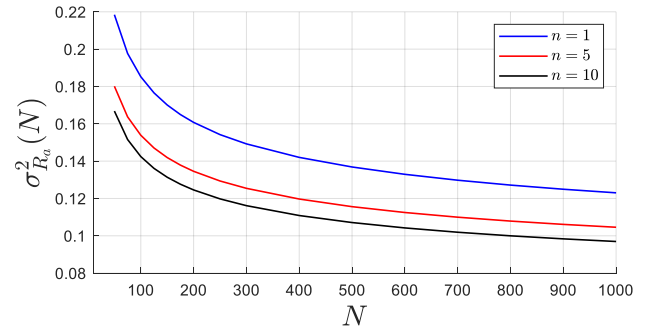


Fig. 3. Variance $\sigma_{R_a}^2$ of random variable R_a depending on dimensionality n and number of samples N .

In the beginning of this analysis, we assumed that parameter $\beta = 1$, which does not have to be the case in general. Let us introduce a random variable $W = R_a^\beta$ to show how parameter β affects the clustering radius. We can easily obtain the cdf for random variable W :

$$F_W(w) = P\left(R_a^\beta \leq w\right) = P\left(R_a \leq w^{\frac{1}{\beta}}\right) = F_{R_a}\left(w^{\frac{1}{\beta}}\right) u_H(w) \quad (18)$$

as well as its pdf:

$$f_W(w) = \frac{1}{\beta} w^{\frac{1}{\beta}-1} f_{R_a}\left(w^{\frac{1}{\beta}}\right) u_H(w) \quad (19)$$

For a constant number of samples $N = 100$ and various values of β , the pdf of W is calculated and shown in Fig.4. We can see that by changing the value of β by $\pm 30\%$ in respect to the unit value, we make a great impact on the expected value of clustering radius, as well as its variance.

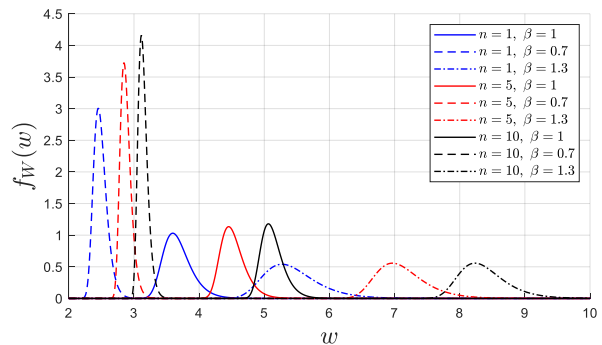


Fig. 4. Probability density function of random variable $W = R_a^\beta$ depending on dimensionality of samples n and parameter value β . W represents maximum of Euclidian norm between two samples to the power of β .

C. Classification accuracy

Performance of clustering algorithm is tested on two-dimensional samples from 5 classes, distributed normally with different covariance matrices.

The main goal of clustering is to find centers of all classes and then test how accurate the classification is. Since the dataset is synthetically generated and all classes and covariance matrices are known, we decided to determine which class a sample belongs to by calculating the statistical distance according to equation (20),

$$d_i^2 = (X - X_{c_i})^T \Sigma_{c_i}^{-1} (X - X_{c_i}) \quad (20)$$

where d_i is the statistical distance of sample X to i^{th} class, whose center is in X_{c_i} , and covariance matrix is Σ_{c_i} . For each sample the statistical distance to all found centers is calculated, and then the sample gets placed in the class for which the distance is minimal. Classification accuracy is finally measured as a percentage of accurately classified samples.

III. RESULTS AND DISCUSSION

We generated 300 samples for each of the 5 classes. As previously mentioned, each class is normally distributed. Clustering radius r_a is determined according to (6), and parameter $\beta = 0.5$ [3]. As suggested in [2], the new clustering radius r_b is $r_b = 1.5r_a$. Clustering threshold δ is manually tuned for the algorithm to detect the right number of clusters.

Figure 6. shows all classes and centers which were detected by subtractive clustering.

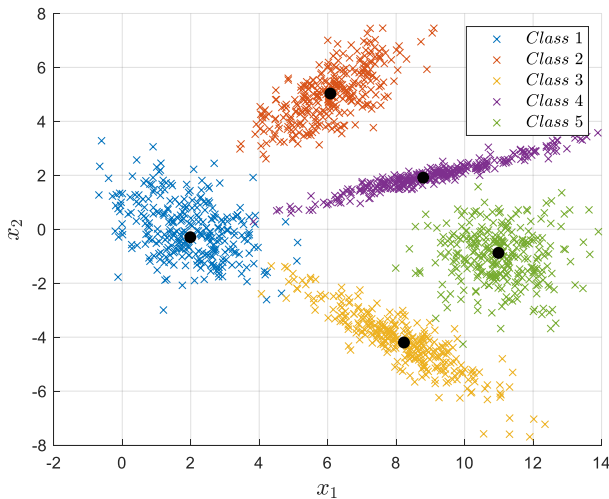


Fig. 5. Sample distribution among classes and corresponding cluster centers found.

To test the sensitivity of the algorithm (with said parameter values) the experiment was repeated 500 times. For each experiment the clustering radius r_a is determined separately. The histogram of the number of detected cluster centers in 500 experiments is given in Fig. 6.

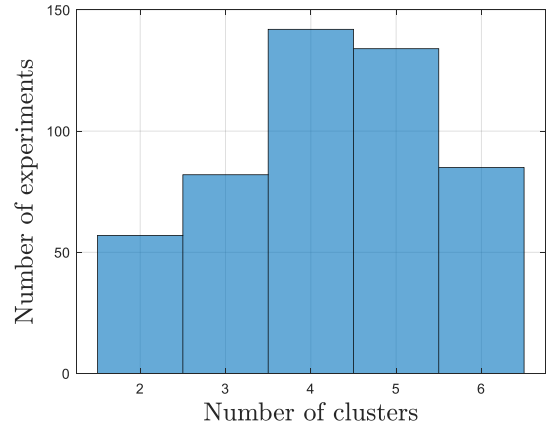


Fig. 6. Histogram of the number of detected cluster centers.

For experiments in which the detected number of centers is correct, i.e. 5, we calculated the classification accuracy as mentioned earlier. Figure 7. shows the histogram of classification accuracy.

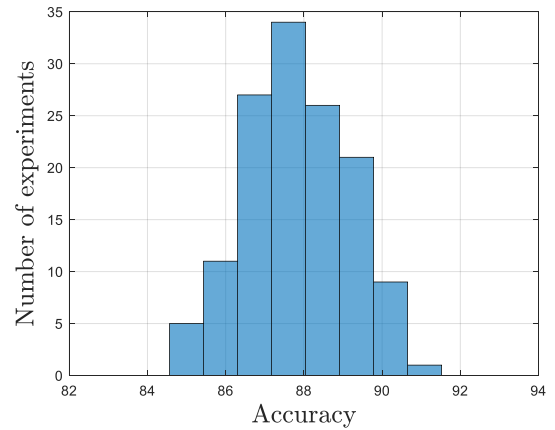


Fig. 7. Histogram of classification accuracy.

In 134 out of 500 experiments the algorithm has managed to detect all cluster centers. The average accuracy in these experiments is 88 %.

Finally, it is also interesting to see how the new cluster radius r_b affects the success of finding cluster centers. We choose r_b according to:

$$r_b = \varepsilon r_a \quad (21)$$

where ε is a positive constant called the squash factor. Various papers give different suggestions regarding the value of ε , depending on the practical use of the algorithm. Based on [7] and [8] ε should be in an interval of $\varepsilon \in [1, 1.5]$.

We generated the same number of samples with the same distribution as before, and for each of the experiments we tested the success of the algorithm by changing ε from 1 to 1.5 with the step $\Delta\varepsilon = 0.05$. The experiment was repeated one hundred times. Histogram of the number of centers found depending on ε is shown in Fig. 8.

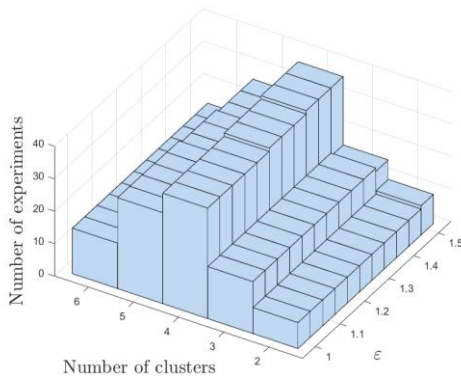


Fig. 8. Histogram of the number of cluster centers found depending on ϵ .

IV. CONCLUSION

In this paper we analyzed both theoretically and experimentally the technique of subtractive clustering, as well as suggestions for tuning its parameters. Values of parameters r_a , r_b and δ significantly affect the success of clustering. We observed that even with various suggestions on how to choose the values, the algorithm does not always perform well on different sets with the same distribution.

However, even if we assume that parameters r_a and r_b are tuned correctly, there are no theoretical propositions on choosing the threshold value δ , which has a great impact on the number of clusters found. In case of samples with dimensions $n = 1, 2, 3$ we can visualize the dataset and assess whether the cluster centers have been found correctly. Nevertheless, in most cases the dimensions will be significantly bigger and there would be no unique way to rate the success of the algorithm.

One of the methods to further enhance the algorithm and propose a uniform way for choosing parameter values would be to separately consider the maximum scattering along each of the axis. For example, if we have two-dimensional samples

where the variance along one axis is much larger than the other and classes are near to each other, determining the density in the radius as the maximum quadratic norm of two samples, would lead to poor results.

Though the simplicity of subtractive clustering is its big advantage, the results are not ideal. That being said, it can be used as a preprocessing technique for other more sophisticated methods.

ACKNOWLEDGMENT

The paper was co-funded by the Ministry of Education, Science and Technological Development of the Republic of Serbia. This year's contract number is 451-03-68/2022-14/200103.

REFERENCES

- [1] R. R. Yager and D. P. Filev, "Approximate clustering via the mountain method", *IEEE Transactions on systems, man, and Cybernetics*: pp. 1279-1284, 1994.
- [2] S. L. Chiu, "Fuzzy model identification based on cluster estimation", *Journal of Intelligent & fuzzy systems*, vol. 2, no. 3, pp. 267-278, 1994.
- [3] X. Cui, S. Liu, and L. Jia, "An improved method of semantic driven subtractive clustering algorithm", in *2015 IEEE 5th International Conference on Electronics Information and Emergency Communication*, pp. 232-235, IEEE, 2015.
- [4] M. M. Milošević, "Speaker identification in conditions of emotional speech", Ph.D. dissertation, Signals and Systems, University of Belgrade, Belgrade, Serbia, 2020.
- [5] M. Merkle, "Random variables", *Probability and statistics for engineers and engineering students* [in Serbian], Belgrade, Serbia, Akademska misao, 2016, ch. 3, sec. 3.6, pp. 81–82.
- [6] M. Merkle, "Some important distributions", *Probability and statistics for engineers and engineering students* [in Serbian], Belgrade, Serbia, Akademska misao, 2016, ch. 7, sec. 7.2, pp. 157–159.
- [7] K. Demirli, S. Cheng, and P. Muthukumar, "Subtractive clustering based modeling of job sequencing with parametric search", *Fuzzy Sets and Systems*, vol. 137, no. 2, pp. 235-270, 2003.
- [8] X.-x. Jing, L. Zhan, H. Zhao, and P. Zhou, "Speaker recognition system using the improved gmm-based clustering algorithm", in *2010 International Conference on Intelligent Computing and Integrated Systems*, pp. 482-485, IEEE, 2010.