International Journal of Engineering and Advanced Technology (IJEAT)
ISSN: 2249-8958 (Online), Volume-13 Issue-1, October 2023

Performance Analysis of an Improved Particle Swarm Optimization and the Standard Particle Swarm Optimization

Patrick O. M. Ogutu, Nicholas Oyie, Winston Ojenge

Abstract*: Many industries employ different modes of control when it comes to PID parameter tuning. Previous authors have addressed the problem of tuning a control system for linear and nonlinear systems; however, the level of error reduction in system performance has not been achieved effectively. Hence, this study focuses on improving particle swarm optimisation using an improved Algorithm for PID parameter tuning. This paper addresses the optimisation of PID parameters using an improved PSO algorithm for nonlinear systems. Particle swarm optimisation is used to tune the PID parameters, ensuring an enhanced system response and operation. The PSO was deployed in a nonlinear system for application and validation of results achieved through PID tuning of the standard parameters on the MATLAB Simulink platform. The study ensured that the PID parameters were effectively tuned by applying an improved PSO Algorithm to the plant process. The research employed a standard nonlinear system to depict a real-life situation and an Improved Particle Swarm Optimisation Algorithm to analyse and compare the improved behaviour in the MATLAB/Simulink toolbox, as applied to the PID parameters. Finally, it was logically realised that the enhanced PSO Algorithm system response was significantly better compared to the non-PSO tuned system. The simulation was performed on the plant transfer function using the MATLAB and Simulink platforms, with various parameter choices and situations, and realisations were made from the obtained data. As the iteration was increased from 10, 50, and 100, there was a significant reduction in ITAE error from 0.054806 to a minimum of 0.01900, which is far better than the SPSO algorithm. SPSO reduces the error from 0.065143 to 0.020476. It was noted that the system behaviour was significantly improved in terms of settling time and peak overshoot for IPSO.

Keywords: MATLAB Simulink, PID parameters, Iteration, nonlinear system, Particle swarm optimization Algorithm

I. INTRODUCTION

Particle Swarm Optimisation (PSO) is a popular optimisation algorithm that has been widely applied to various nonlinear systems.

Manuscript received on 06 September 2023 | Revised Manuscript received on 13 September 2023 | Manuscript Accepted on 15 October 2023 | Manuscript published on 30 October 2023.

*Correspondence Author(s)
Patrick O. M. Ogutu*, Department of Electrical and Electronic Engineering, Murang'a University, Nairobi, Kenya. E-mail: patogutu2002@yahoo.com, ORCID ID: 0000-0003-4194-1985
Dr. Nicholas Oyie, Department of Electrical and Electronic Engineering, Murang'a University, Nairobi, Kenya. E-mail: noyie@mut.ac.ke , ORCID ID: 0000-0001-8829-0281
Dr. Winston Ojenge, Department of Electrical and Electronic Engineering, Murang'a University, Nairobi, Kenya. E-mail: ojenge@tukenya.ac.ke., ORCID ID: 0000-0002-1576-2126

© The Authors. Published by Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP). This is an open access article under the CC-BY-NC-ND license http://creativecommons.org/licenses/by-nc-nd/4.0/

However, the traditional PSO algorithm can be limited in its ability to solve complex optimization problems effectively. Therefore, an improved version of PSO, known as Improved Particle Swarm Optimization (IPSO), has been developed to overcome some of these limitations. IPSO is a population-based optimization algorithm that uses a swarm of particles to search for the optimal solution to a problem. Like traditional PSO, IPSO starts by initializing a population of particles, each representing a potential solution to the problem. However, IPSO introduces several modifications to the PSO algorithm that enable it to explore the search space better and converge to a global optimum.

One modification that IPSO introduces is the use of a dynamic inertia weight. The inertia weight is a parameter that controls the impact of the particle's previous velocity on its current velocity. In IPSO, the inertia weight is adjusted dynamically during the optimisation process, enabling a more efficient search of the search space. [5]

Another modification is the use of a hybrid mutation operator. The mutation operator introduces random perturbations to the particles' positions, promoting the exploration of the search space. In IPSO, the mutation operator is combined with a local search method to ensure that the perturbations lead to improved solutions.

Furthermore, IPSO uses a novel boundary-handling technique to address the problem of particles moving beyond the search space. This technique helps to prevent premature convergence and improve the algorithm's ability to find global optima.

The effectiveness of IPSO has been demonstrated in various non-linear system optimisation problems, including neural network training, control system design, and feature selection in machine learning. The IPSO algorithm has been shown to outperform traditional PSO and other popular optimization algorithms in terms of convergence speed, accuracy, and robustness.

In summary, IPSO is an enhanced version of the PSO algorithm that addresses some of the limitations of traditional PSO in solving nonlinear system optimisation problems. Its dynamic inertia weight, hybrid mutation operator, and novel boundary-handling technique make it a robust optimisation algorithm that can efficiently search for optimal solutions in a variety of complex optimisation problems.

II. LITERATURE REVIEW AND MATHEMATICAL MODELLING

A. Literature Review

The author [2] studied the influence of energy and

© Copyright: All rights reserved.

humidity control but did not effectively address the performance aspect. ACO and PSO techniques were employed in designing a plant control system, where ACO was found to be slightly superior to PSO. The distancing function technique was implemented, and the method was found to be easy, very fast, and easily implementable. However, [1] worked on the latest PID adjustment technique by using PSOSCALF as a form of optimisation, which resulted in the Overshoot decreasing from 25% to 10%, respectively. The optimal tuning of a PID controller gain based on the PSO method, using the ITSE performance index, and the efficacy of the controller are tested with a tri-loop error at different transient conditions [1].

A Modified Particle Swarm Optimisation Algorithm for the economic dispatch problem was proposed, and a study on the use of PSO for solving the economic dispatch problem in power systems was conducted. The outcome revealed that the proposed method performed poorly compared to other optimization algorithms [8].

A performance comparison of Various Evolutionary Algorithms for Optimal Reactive Power Dispatch was investigated, and the authors compared the performance of PSO and other evolutionary algorithms in solving the reactive power dispatch problem in power systems. Their results showed that PSO performed poorly in terms of convergence speed and solution quality compared to other algorithms. [6]

A new approach to fuzzy tuning using the distancing function is a paper that proposes a novel method for tuning fuzzy systems based on the use of a distancing function. The approach is shown to have several advantages over traditional fuzzy tuning methods and is validated through simulation experiments on several benchmark problems. [3]

Design of Symmetrical FIR Digital Filters Using Particle Swarm Optimisation: A Method for Digital Filter Design. However, their results showed that PSO performed poorly compared to other optimization algorithms for the same problem. [10]

These are just a few examples of authors who have studied PSO optimization with limited progress. It's worth noting that the performance of PSO [4] can be highly dependent on the specific problem being solved, the choice of parameters, and the quality of the fitness function. Therefore, it's essential to carefully evaluate the performance of PSO and compare it with other optimization algorithms before drawing any conclusions about its effectiveness

A. Mathematical Modeling of the Process

The circuit diagram, as shown in Fig. 1, displays the arrangement of several components that will yield results for analysis when run on MATLAB/Simulink with both the standard and improved particle swarm algorithms. The circuit consists of two operational amplifiers that will be injected with a step signal, a PID[7] block from which the initial proportional-integral-derivative parameters shall be set, an integrator and a derivative circuit connected to achieve ITAE, ISE, and IAE, a display unit to indicate the output values during the iterations, a multiplexer (mux) which is a digital circuit that selects one of several inputs and forwards the selected input to a single output line, and a CRO which deploys the output response of the signals.

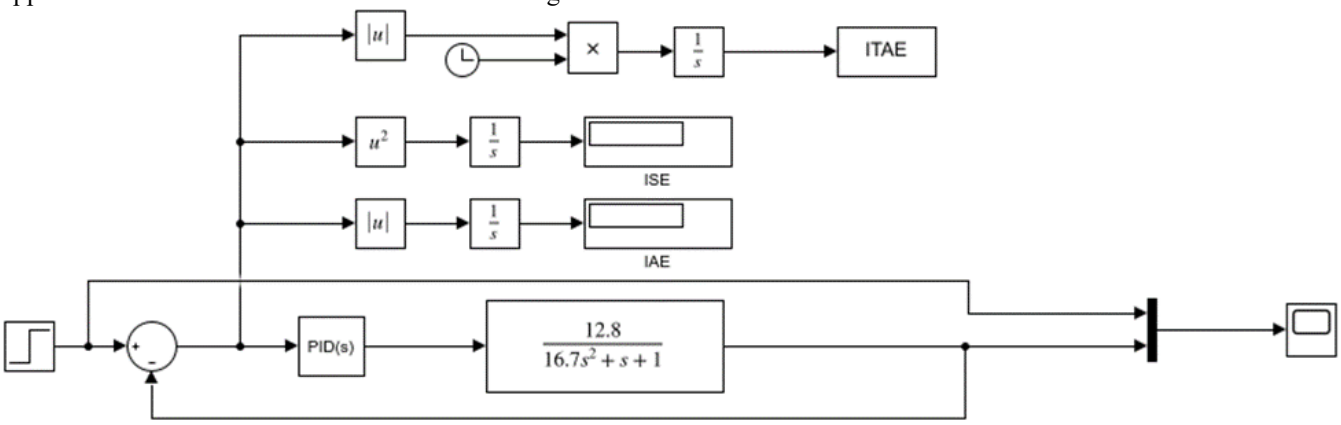

Fig. 1 Circuit Layout

B. Transfer Function

The plant mathematical model is arrived at after a series of calculations described below. Consider the Transfer function [9]

$$T.F = \frac{G}{1+GH} = \frac{Y(s)}{U(s)} = \frac{12.8}{16.7s^2+s+1} \quad (1)$$

Obtained from x. Using a state-space representation, the state vector can be defined as X = [x1, x2]T, where x1 and x2 are the state variables. The function can be expressed in state space.

$$\frac{dx}{dt} = Ax + Bu \quad (2)$$

$$y = Cx + Du \quad (3)$$

Where

A, B, C, and D are matrices that explain the dynamic behaviour of the system. To obtain the matrix, the following parameters are used:-

© Copyright: All rights reserved.

A= [0 1;-1/16.7 -1/16.7]

B=[0;12.8/16.7]

C= [10]

D= 0

$$\frac{dx1}{dt} = x_2 \quad (4)$$

$$dx2/dt-= -(1/16.7)x1 - (1/16.7)x2 + (12.8/16.7)u \quad (5)$$

$$y = x_1 \quad (6)$$

The State equation describes the dynamic behaviour of the system, and the output equation relates the output to the state vector. As mentioned above, the state equation representation can be easily transferred to MATLAB.

C. PSO Tuning Process

The PID Controller: The PID controller generates the control signal sent to the actuator. The controller calculates the error signal by comparing the set point with the current process value and generates the control signal based on the proportional, integral, and derivative terms.PSO Algorithm: The PSO algorithm is employed to optimise the parameters of the PID controller. The PSO algorithm searches for the optimal set of parameters that minimizes the error between the set point and the process value. The PSO algorithm iteratively updates the position and velocity of each particle in the search space to find the optimal solution.

Fitness Function: The fitness function evaluates the performance of the PID controller with a particular set of parameters. The fitness function calculates the error between the set point and the process value over a specified period.

Optimization: The PSO algorithm evaluates the fitness function for each set of parameters and updates the particles' positions and velocities based on the best-performing particles in the search space. The optimization process continues until the stopping criterion is met, such as reaching a maximum number of iterations or achieving a certain level of convergence.

Actuator: The actuator receives the control signal from the PID controller and applies it to the process to control its behavior. The actuator could be a valve, a pump, or any other mechanical or electrical component that can adjust the process's behavior. In summary, the block diagram of a process controlled by a PID controller optimized using PSO consists of a process, a sensor, a PID controller, a PSO algorithm, a fitness function, an optimization process, and an actuator. The PSO algorithm searches for the optimal set of parameters for the PID controller to minimise the error between the set point and the process value. The actuator then applies the control signal to the process to achieve the desired behaviour.

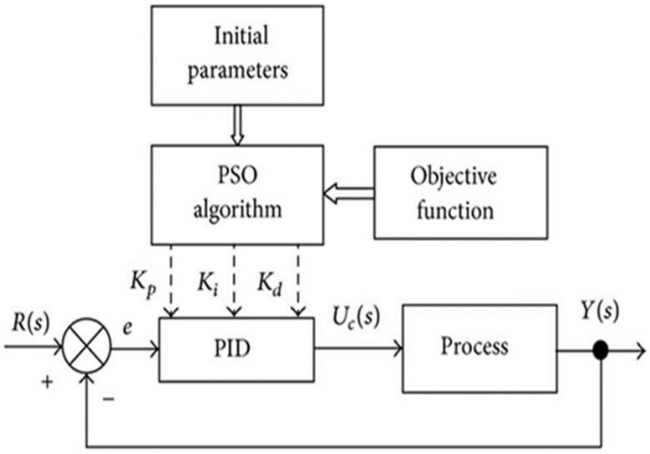

Fig. 2 PID and PSO Optimized Block Diagram.

III. MATERIALS & METHODS

The components used to accomplish the paper objectives are: ITAE, ISE, IAE Blocks, step function, CRO, MATLAB Simulink software, and MUX. The process is then designed using MATLAB simulation software, which operates on the Simulink model design platform. The same design is connected to the PID controller and then tuned using the PID tuning platform in MATLAB Simulink, as shown in Fig. 1. The created Algorithm is inserted and stored in the MATLAB workspace, then run according to the number of iterations and particle optimisation. The sequence of activities is repeated, with the PID parameters adjusted, and the process is repeated.

IV. RESULTS & DISCUSSIONS

This section presents the simulation results obtained using MATLAB/Simulink software for the system with a PID controller tuned using the Ziegler-Nichols and PSO methods. Furthermore, the results obtained using different performance evaluation indices for the PSO-tuned PID controller are shown and compared [1].

The optimisation process of a PID controller involves tuning the proportional, integral, and derivative gains to achieve the desired system response. IPSO and SPSO are optimisation techniques used to automate this tuning process. IPSO employed a hybrid approach that combines the standard PSO update equation with the gradient-based update equation, whereas SPSO incorporated a self-organising mechanism and a mutation operator. Both IPSO and SPSO result in faster convergence and improved performance compared to the standard PSO algorithm and PID, as shown in Table 1.

Table 1: Simulation 1: 10 Iterations and 30 Particles

ITEM	K_P	K_I	K_D	ITAE	ISE	IAE
PID	2	0.170800	2	2.795100	0.31840	0.92270
SPSO	100	37.10776	100	0.061543	0.01141	0.03459
IPSO	100	15.26855	100	0.054806	0.01155	0.03159

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)
© Copyright: All rights reserved.

Table 1. The above shows that the Integral Time Absolute Error (ITAE) is very high for PID, with a value of 2.7951, and very low for Improved Particle Swarm Optimisation (IPSO), at 0.0054866. However, the Standard Particle Swarm Optimisation (PSO) is also more effective in terms of error reduction compared to the conventional PID. For the case of ISE, PID performed dismally with a value of 031840, while Standard PSO performed better compared to IPSO. For IAE, the error is further reduced as the system runs from PID to SPSO and finally to IPSO.The system response curve, as shown in Fig. 3, clearly indicates that IPSO outperforms SPSO and the conventional PID. The magnified caption reveals that the overshoot is considerably higher for SPSO compared to IPSO. This directly demonstrates that the behaviour of IPSO has improved the system's behaviour as a result of the enhanced algorithm. For 10 iterations, convergence is achieved faster with IPSO compared to SPSO. IPSO settles much quicker, while SPSO takes longer, as can be seen in the graphic representation in Fig. 4

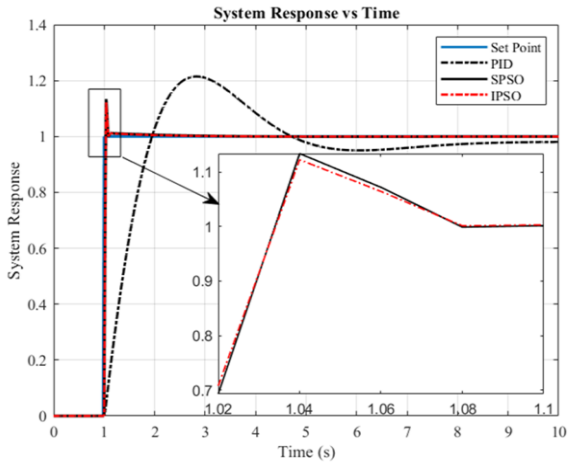

Fig. 3. System response for 10 iterations

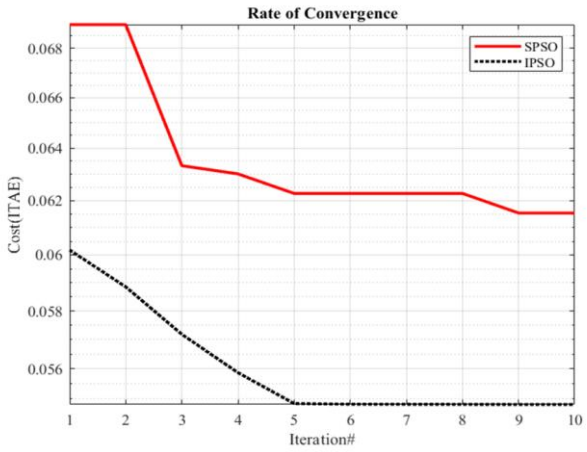

Fig. 4. Rate of convergence

Table 2: Simulation 2: 50 Iterations and 30 Particles

ITEM	K_P	K_I	K_D	ITAE	ISE	IAE
PID	2	0.170800	2	7.326000	0.32080	1.14900
SPSO	100	15.66876	100	0.054696	0.01140	0.03200
IPSO	100	15.23772	100	0.054806	0.00155	0.03158

Table 2 above shows that the Integral Time Absolute Error (ITAE) is very high for the PID controller, with a value of 7.326000, and very low for the standard particle swarm optimisation, at 0.054696. However, the improved particle swarm optimisation is more effective in terms of error reduction compared to the conventional PID, with a value of 0.054806.

For the case of ISE, PID performed dismally with a value of 0.32080. However, the improved PSO performed better than SPSO, with values of 0.00155 and 0.01140, respectively. For IAE, the error is further reduced as the system is run from PID to SPSO and finally to the improved particle swarm optimization with values of 1.14900, 0.03200, and 0.03158 respectively. Overall, the standard SPSO yields better results compared to PID and IPSO, particularly in terms of ITAE. However, for IAE, IPSO performs well in comparison to the other methods.

The system response curve, as shown in Fig. 5, clearly indicates that IPSO outperforms SPSO and the conventional PID. The magnified caption reveals that the overshoot is significantly higher for SPSO compared to IPSO. This directly demonstrates that the behaviour of IPSO has improved the system's behaviour as a result of the enhanced algorithm.

For 50 iterations, convergence is achieved faster with IPSO compared to SPSO. IPSO settles much quicker than SPSO, as shown in the graphical representation in Fig. 6.

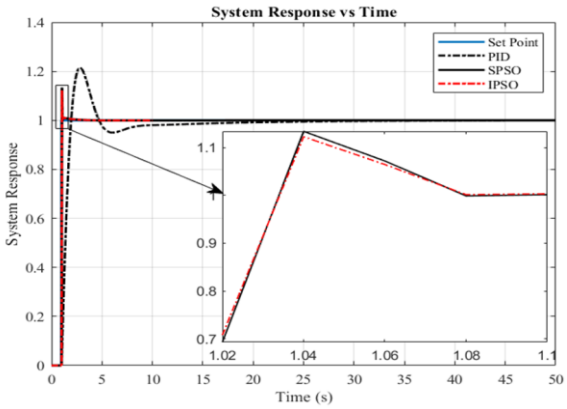

Fig. 5. System response for 50 iterations

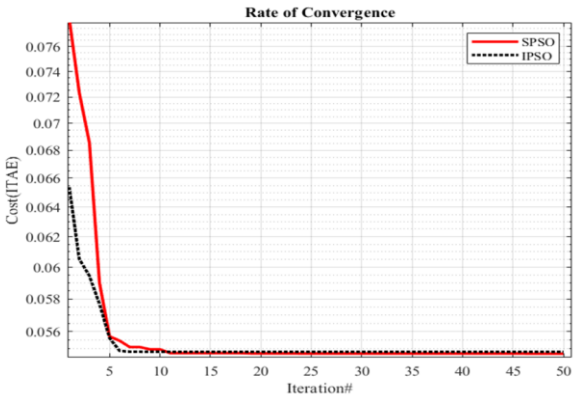

Fig. 6. Rate of Convergence 50 Iterations

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)
© Copyright: All rights reserved.

V. CONCLUSION

The paper has convincingly demonstrated that particle swarm optimisation is superior to the conventional PID controller in optimisation. Simulated results show that SPSO and IPSO have a positive effect on tuning PID controller parameters. The PID controller parameter tuning method based on the IPSO Algorithm improves the situation, whereas the traditional tuning method requires multiple attempts to select the optimum parameters. In contrast, IPSO and SPSO automatically select the parameters according to the algorithm within a short time. This paper successfully develops an AI-based automatic PID tuning scheme using an IPSO algorithm that can automatically acquire or re-tune the PID parameters during plant operation in a routine manner. The results demonstrate the effectiveness of modern optimisation techniques, such as IPSO, in control applications. The simulation was performed on the plant transfer function using the MATLAB and Simulink platforms, with various parameter choices and scenarios, and realisations were made from the obtained data. As the iteration was increased from 10 to 50 and 100, there was a significant reduction in ITAE error, from 0.054806 to a minimum of 0.01900, which is also significantly better than the SPSO algorithm. SPSO reduces the error from 0.065143 to 0.020476. When considering the ISE, it was noted that PID substantially increases the error factor; however, SPSO and IPSO reduce the error at iteration 50 from 10 and then further reduce the error at 100 iterations. For IAE, the PID error increases from 0.92270 to 1.1550; however, for SPSO and IPSO, the error factor is drastically reduced as the number of iterations increases.

It can thus be concluded that the number of iterations improves the error for both SPSO-IPSO and IPSO, and the performance of IPSO is significantly better than that of SPSO and the conventional PID. When it came to convergence, IPSO also performed better than SPSO. An increase in population from 15 to 30, while maintaining other parameters, did not affect the PID errors in terms of ITAE, ISE, and IAE. However, it has a significant impact on the SPSO-ITAE, where it reduces the error on ITAE while increasing the error on ISE and IAE. The paper found that IPSO, using the improved algorithm, can perform significantly better compared to SPSO and PID; hence, it can be applied in plant optimisation.

Future researchers can use this study as a springboard for high-quality research in areas that my study has not thoroughly explored. The work contributes to the study of PID controller tuning methods and limitations, particularly when using meta-heuristic algorithms that have not been thoroughly explored in the context of PID parameter tuning in industrial applications. The authors do hereby declare that the work was a team effort. The study does not require ethical approval or consent to participate, as it only involves simulations. No financial support was obtained. No conflict of interest is declared. Data was obtained from MATLAB/Simulink designed and Algorithm simulations.

ACKNOWLEDGEMENT

All those important people who participated in one way or another, and finally, to God be the glory—special thanks to TU-K and MUC for objective guidance.

DECLARATION STATEMENT

Funding/ Grants/ Financial Support	No, I did not receive.
Conflicts of interest	No conflicts of interest to the best of our knowledge.
Ethical Approval and Consent to Participate	No, this article does not require ethical approval or consent to participate, as it is based on evidence.
Availability of Data and Material/ Data Access Statement	Not relevant.
Authors Contributions	All authors have equal contributions to this article.

REFERENCES

1. E. Y. Bejarbaneh and A. Bagheri, "A new adjustment technique for PID-type fuzzy logic controller using PSOSCALF optimisation algorithm," 2019. https://doi.org/10.1016/j.asoc.2019.105822
2. V. Fevrier et al., "Fuzzy dynamic parameter adaptation in ACO and PSO for designing fuzzy controller –the case of water level and temperature control," 2018. https://doi.org/10.1155/2018/1274969
3. J. Dombi and A. Hissein, "A new approach to fuzzy control using the distancing function," Journal of Process Control, 2020. https://doi.org/10.1016/j.jprocont.2019.12.005
4. S. K. Rath and B. K. Pangrahi, "Performance comparison of various Evolutionary Algorithms for Operational Reactive Power dispatch," 2013.
5. P. A. Hokam et al, Lecture notes on nonlinear systems and control, Switzerland: ABB Ltd, 2018.
6. J. Iqbal et al, "Nonlinear control system," vol. 6, no. 4, pp. 301-312, 2017.
7. M. Maldini, "PID control Explained," https://maldus512.medium.com/pid-control-explained-45b671f10bc, 2018.
8. O. Wu and J. Zeng, "A modified particle swarm optimization algorithm for economic dispatch problem," 2015.
9. M.W.Berry and R.K.Wood, "Chemical Engineering Science," Pergamon Press, vol. 28, pp. 1707-1712, 1973. https://doi.org/10.1016/0009-2509(73)80025-9
10. S. Chakraborty and U. Maulik, "Design of Symmetrical FIR Digital filters using PSO," 2007.

AUTHORS PROFILE

Mr. Patrick O.M. Ogutu, an experienced and dedicated instrumentation and control technologist with over 23 years of hands-on experience in training and mentorship, is a highly skilled and competent Instrumentation and control technician who has excelled in their workplace, as revealed by the industrial liaison office. My short-term goal is to obtain an M.Tech in Control and Instrumentation, and over the next five years, to get a Ph.D. in Instrumentation Astronomy, specialising in the design, maintenance, and calibration of modern astronomical instruments. This goal is based on completing the basic DARA-sponsored training, which provided me with valuable insight and sparked my interest. I am currently in charge of the control and automation laboratory, where I serve as a reliable and honest technologist who has excelled in the field of automation. I perform duties with minimal supervision, handle work promptly, and strictly adhere to deadlines. Master of Technology (Instrumentation and Control), Murang'a University of Science and Technology (student), BPhil Technology (Automation and control), TUK. Higher Diploma in Electrical and Electronic Engineering, Mombasa Polytechnic College, Diploma in Technology

© Copyright: All rights reserved.

(Instrumentation and control), Mombasa Polytechnic College, A-Level Certificate, K.A.C.E., Shimo-la Tewa High School, Level Certificate, K.C.E., Alidina Visram High School.

Dr. Nicholas Oyie (Ph.D.) -Lecturer and Chairman of the Department of EEE, School of Engineering and Technology, Murang'a University of Technology, Kenya. PhD. in Electronic Engineering from the University of KwaZulu-Natal, South Africa. Registered Graduate Engineer with the Engineers Board of Kenya (EBK). Measurements and analysis of large-scale path loss model at 14 and 22 GHz in indoor corridor, Performance study of path loss models at 14, 18, and 22 GHz in an indoor corridor environment for wireless communications, An author of the following: A comparative study of dual-slope path loss model in various indoor environments at 14 to 22 GHz, Investigating the impact of antenna heights on path loss models in an indoor corridor environment, An empirical approach to omnidirectional path loss and line-of-sight probability models at 18 GHz for 5G networks, Propagation path loss prediction modelling in enclosed environments for 5G networks: Contact

Dr. Winston Ochieng Ojenge, (PhD), Computer Science at the T-UK, (M.Sc.) Information Systems UON (Kenya) 2008, (B.Ed.) technology, Moi University (Kenya).Lecturer, Control and Automation Engineering. He heads the Digital Economy Program. He has a PhD in Computer Science from the Technical University of Kenya. His research interests are artificial intelligence and machine learning, and IoT. He was the founder and coordinator of the Innovation Lab at the Technical University of Kenya and holds three patents in the areas of telecommunications and AI, with three other patent applications under review at the Kenya Institute of Intellectual Property. He is currently the Co-Lead for the AI4D PhD Scholarships in AI. The Contact email is

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of the Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP)/ journal and/or the editor(s). The Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP) and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.