International Journal of Modern Engineering
IJME

WWW.IJME.US
Print ISSN: 2157-8052
Online ISSN: 1930-6628

INTERNATIONAL JOURNAL OF MODERN ENGINEERING

Spring/Summer 2024
Volume 24, No. 2

The Leading Journal of Engineering, Applied Science and Technology

THE LEADING JOURNAL OF ENGINEERING, APPLIED SCIENCE, AND TECHNOLOGY

International Journal of Modern Engineering

www.ijme.us

Print ISSN: 2157-8052
Online ISSN: 1930-6628

International Association of Journals and Conferences

www.iajc.org

INTERNATIONAL JOURNAL OF MODERN ENGINEERING

ABOUT IJME:

- IJME was established in 2000 and is the first and official flagship journal of the International Association of Journal and Conferences (IAJC).
- IJME is a high-quality, independent journal steered by a distinguished board of directors and supported by an international review board representing many well-known universities, colleges and corporations in the U.S. and abroad.
- IJME has an impact factor of **3.00**, placing it among the top 100 engineering journals worldwide, and is the #1 visited engineering journal website (according to the National Science Digital Library).

OTHER IAJC JOURNALS:

- The International Journal of Engineering Research and Innovation (IJERI)
  For more information visit www.ijeri.org
- The Technology Interface International Journal (TIIJ).
  For more information visit www.tiij.org

IJME SUBMISSIONS:

- Manuscripts should be sent electronically to the manuscript editor, Dr. Philip Weinsier, at philipw@bgsu.edu.

  For submission guidelines visit
  www.ijme.us/submissions

TO JOIN THE REVIEW BOARD:

- Contact the chair of the International Review Board, Dr. Philip Weinsier, at philipw@bgsu.edu.

  For more information visit
  www.ijme.us/ijme_editorial.htm

INDEXING ORGANIZATIONS:

- IJME is currently indexed by 22 agencies. For a complete listing, please visit us at www.ijme.us.

Contact us:

Mark Rajai, Ph.D.

Editor-in-Chief
California State University-Northridge
College of Engineering and Computer Science
Room: JD 4510
Northridge, CA 91330
Office: (818) 677-5003
Email: mrajai@csun.edu

www.tiij.org

www.ijeri.org

INTERNATIONAL JOURNAL OF MODERN ENGINEERING

The INTERNATIONAL JOURNAL OF MODERN ENGINEERING (IJME) is an independent, not-for-profit publication, which aims to provide the engineering community with a resource and forum for scholarly expression and reflection.

IJME is published twice annually (fall and spring issues) and includes peer-reviewed articles, book and software reviews, editorials, and commentary that contribute to our understanding of the issues, problems, and research associated with engineering and related fields. The journal encourages the submission of manuscripts from private, public, and academic sectors. The views expressed are those of the authors and do not necessarily reflect the opinions of the IJME editors.

EDITORIAL OFFICE:

Mark Rajai, Ph.D.
Editor-in-Chief
Office: (818) 677-2167
Email: ijmeeditor@iajc.org
Dept. of Manufacturing Systems
Engineering & Management
California State University-
Northridge
18111Nordhoff Street
Northridge, CA 91330-8332

THE INTERNATIONAL JOURNAL OF MODERN ENGINEERING EDITORS

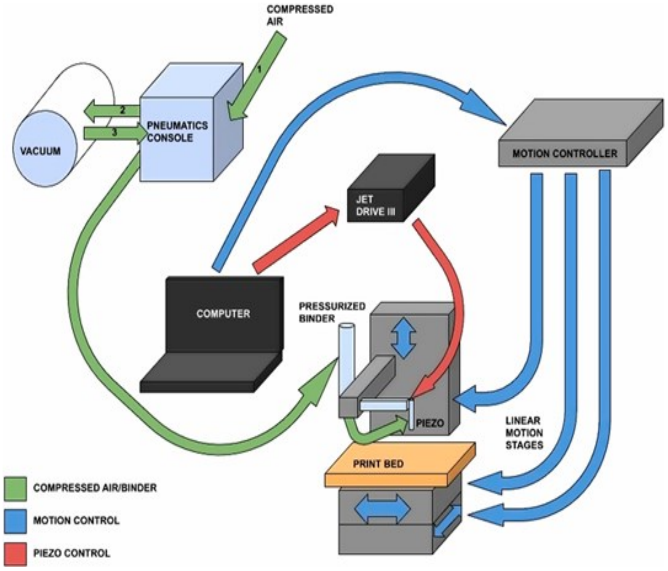
Table of Contents

In This Issue (p.20)
Binder Jetting and Additive Manufacturing

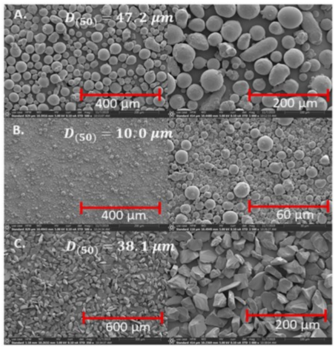
Philip Weinsier, IJME Manuscript Editor

Binder jetting (BJ), as Colton and Crane (https://doi.org/10.1016/j.addma.2020.101711) describe the process, is a low-cost additive manufacturing (AM) process that uses inkjet technology to selectively bind particles in a powder bed. BJ relies on the ability to control not only the placement of binder on the surface but also its imbibition into the powder bed. This is a complex process in which pico-liter-sized droplets impact powder beds at velocities of 1–10 m/s. However, the effects of printing parameters, such as droplet velocity, size, spacing, and inter-arrival time on saturation level (a fraction of pore space filled with binder) and line formation (merging of droplets to form a line) are unknown. Prior attempts to predict saturation levels with simple measurements of droplet primitives and capillary pressure assume that droplet/powder interactions are dominated by static equilibrium and neglect the impact of printing parameters.

Colton and Crane analyzed the influence of these parameters on the effective saturation level and conditions for line formation, when printing single lines into powder beds of varied materials (316 stainless steel, 420 stainless steel, and alumina) and varied particle size (d_{50} = 10–47 µm). Results showed that increasing droplet velocity or droplet spacing decreased effective saturation, while droplet spacing, velocity, and inter-arrival time affected line formation. The following diagram shows the experimental apparatus and connectivity between units. The Newport linear stages are controlled in unison to trigger the JetDrive III to actuate the print head and deposit lines with controlled droplet spacing. The authors varied the inkjet properties (droplet velocity, droplet size) using the JetDrive III waveform controls.

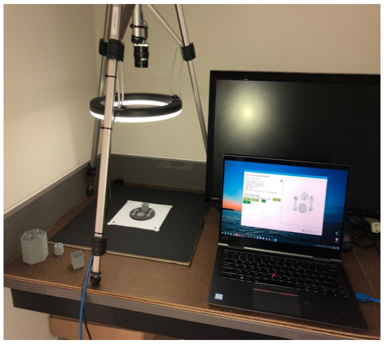


[This image: SEM images of the powder morphology]



In another study by Scime, Haley, Halsey, Singh, Sprayberry, Ziabari, and Paquit [Oak Ridge National Laboratories, September 2020, Pub146907.pdf (ornl.gov)], the authors proved that image-based defect detection coupled with time-series data analysis in binder jet technology is a very reliable methodology for in situ quality control. Additionally, the authors developed a computer vision-based solution to reinforce traceability of the items entering the CVI furnace.

Using the software tool named Raven, then, those authors tracked and took multiple pictures of the silicon carbide parts as they were loaded onto the furnace platform and throughout the CVI process. Preliminary experiments suggested that the location of a part in the CVI furnace affects the silicon carbide infiltration rate and, therefore, the amount of silicon carbide deposited. All of this directly impacts the density and material properties of the final part. These images, analyzed by Raven using computer-vision algorithms, allowed technicians to link each part within the database. The following figure shows the prototype imaging setup.



In this issue of IJME (p.20), the authors of this featured article developed a model to optimize material structure design, and showed that the capability of a feed-forward-back propagation neural network was a good technique for determining the compressive strength of BJ samples. Experimental design techniques were used to design and create three structures: solid circular, circular with 1 mm lattice, and circular with 1 mm cubical. The model predicted the compressive strength given over the range of layer thickness, sintering time, and sintering temperature parameters, and could serve as a framework to set up the process design parameters to achieve the desired output characteristics.

Editorial Review Board Members

Simulation Study of a New Type of Learning Automata

Prateek S. Aggarwal, Boston University; Chenhui Liu, Boston University; Lev Levitin, Boston University

Abstract

In this paper, the authors present numerical simulations of a Probabilistically-Switch-Action-on-Penalty learning automaton (PSAPA) in various stationary and non-stationary environments. The PSAPA is a novel, fixed-structure stochastic automaton (FSSA) framework whose analytical model was presented in detail in a previous paper on PSAPA (Aggarwal, Liu & Levitin, 2022). The key differentiating feature of this automaton is that it allows action switching in every state. It was anticipated that this feature would attribute PSAPA dynamic properties that make certain aspects of its performance superior to other FSSA that do not possess this property.

In this study, then, simulations of the PSAPA in comparison with other FSSA and variable structure stochastic automata (VSSA) were considered in two types of environments: a stationary environment (with fixed penalty probabilities) and a non-stationary environment, where the penalty probabilities would change in time periodically as a sinusoidal function. In both cases, the simulation demonstrated a dramatic difference in performance for these types of learning automata. The PSAPA showed its huge advantage in adaptability that could lead to better performance for the length of the simulation from 30,000 to 150,000 steps. Only for very long stationary conditions do Tsetlin automata outperform the PSAPA. Also, the PSAPA model responded quickly to drastic changes in the environment, unlike the other automata. In the case of sinusoidal modulations, the PSAPA tremendously outperformed other types of FSSAs for all modulation frequencies and for all depths ($D > 3$). The performance of the PSAPA does not deteriorate with increasing modulation frequency, while other FSSA are not resilient to that increase. Also, the PSAPA is more adaptive, where it has to select an action from more than two actions.

Introduction

The design of the Probabilistically-Switch-Action-on-Penalty Automata (PSAPA) in the study by Aggarwal, Liu, and Levitin (2022) was motivated by the need for design of fixed-structure stochastic automata (FSSA) that can possibly perform better than Star-automaton (Economides & Kehagias, 2002) and Tsetlin $L_{2,2N}$ (Tsetlin, 1962) in non-stationary environments. As shown by Aggarwal et al. (2022), the PSAPA achieve ε-optimality with increasing depth according to a power law, rather than exponentially, unlike most learning automata. Thus, the convergence rate of the performance as a function of the depth is larger for Tsetlin and Star automata than for the PSAPA. In other words, for a stationary environment, the minimum number of states required to achieve ε values lower than any given value, will be smaller for Tsetlin and Star automata than for PSAPA. However, one can argue that the PSAPA design allows it to demonstrate better adaptability and, hence, performance in other, possibly non-stationary, environments, as well as in stationary environments for not extremely long periods of time. A detailed description of the design and performance analysis in stationary environments, along with a discussion of the applications of learning automata, can be found in the PSAPA study by Aggarwal et al. (2022).

An FSSA that is ε-optimal tends to occupy the terminal states of each branch, and an FSSA that is not ε-optimal tends to occupy the initial states in each branch. Thus, the behavior of a non- ε-optimal FSSA will not be affected by an increase of the depth beyond a certain value since the automaton will not fully utilize the newly added states. Therefore, the automata used in all the simulations are reformulated so that they are ε-optimal and that their performance consistently improves with an increase in depth. The reformulation is performed by using the penalty-filtering technique presented in previous studies (Narendra & Thathachar, 2012), where a certain fraction of the penalties, selected randomly, are ignored and deemed as rewards. These universally ε-optimal versions of pessimistic PSAPA, ambivalent PSAPA, and Tsetlin automata, are hereby referred to as univ-pess-PSAPA, univ-amb-PSAPA, and univ-Tsetlin automaton, respectfully, for the purpose of conciseness. The optimistic PSAPA is referred as opt-PSAPA.

To assess the performance of different learning automata in stationary and non-stationary environments, previous researchers primarily considered environments where the penalty probabilities were stationary, and the automata were reset at different states at the start of the simulation or environments with periodically modulating or switching penalty probabilities (Narendra & Thathachar, 2012). The average penalty and rapidness of response of the different LAs were then compared. Simulations were also presented where penalty probabilities changed periodically from the highest to the lowest mean values. Following a similar approach, the aim in this current study was to quantify the performance of the PSAPA for two cases: (1) for fixed values of penalty probabilities, and (2) for sinusoidal modulation (within a broad range of frequencies) of the penalty probabilities. This approach was expected to aid in understanding the relationship between the parameters of the different FSSA, and their performance in both stationary and non-stationary environments. First, the transient and steady-state behaviors of univ-Tsetlin automata and PSAPA were compared for constant values of penalty probabilities. Then, for the univ-Tsetlin, PSAPA, and univ-STAR automata, the relationship between the depths of these automata and their performance in environments with periodically changing penalty probabilities was inspected.

An appropriately designed learning automaton should have its performance improve with increasing depth in any environment. Note that for all the FSSA discussed here, the optimal behavior was achieved only when the depth was infinite. From a practical computational standpoint, it is quite reasonable to simulate an automaton with a 32-bit variable. The number of different states in such an automaton is on the order of 10^9. Thus, simulating a nearly infinite state FSSA is not difficult for practical applications. The real challenge is in designing a learning automaton whose performance consistently improves with increasing depth in any environment. From this viewpoint, a set of experiments was performed to understand the transient and steady-state characteristics of the different FSSA. All simulations were performed in the MATLAB computing environment.

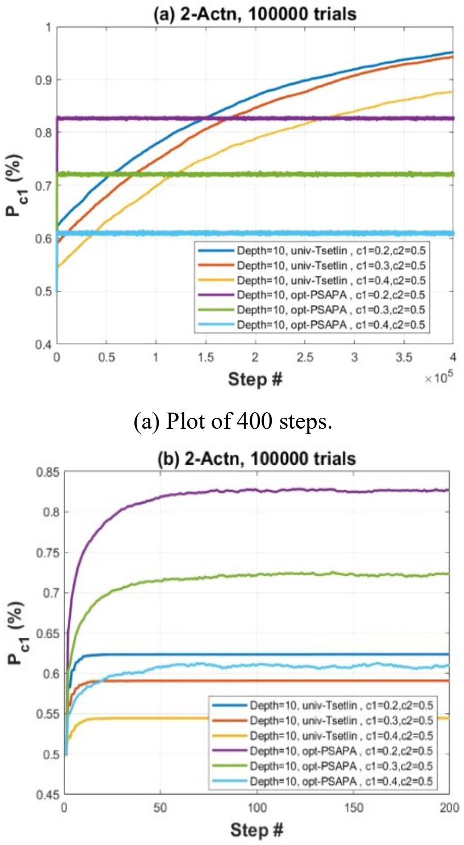
Stationary Environments: Simulation and Analysis

Two different setups were used in the case of fixed-penalty probabilities. In the first setup, the automata performed two different actions; at the start of the simulation each automaton occupied the first (initial) state of the branches with equal probabilities. This is referred to as the equiprobable action simulation. In the second setup, a two-action environment, all automata initially occupied the final (the deepest) state of the branch, which had a higher penalty probability. This means that the automata started in the most disadvantaged position. This is referred to as the position reversal simulation. The goal in both cases was to study the performance of the automata in both the transient and the steady-state stages of the simulations.

In particular, the second setup allowed sufficient evaluation time (number of steps) needed for an automaton to switch its action in the worst-case scenario. To compare the performance of different FSSA, all simulations were conducted for the same depth level, D = 10. All simulations were in a two-action environment with fixed penalty probabilities c_1 and c_2 , where $c_2 > c_1$. Each simulation included 100,000 trials. The fraction of trials $P_1(n)$ at any given step, n, when the automaton performed the action with the smaller penalty probability c_1 was used as the measure of the automaton's performance.

Results for the Equiprobable Action Simulation

Denote the probability of being at state (i,d) (i.e., the d^{th} state in the i^{th} branch), at instant n as $\pi_{(i,d)}(n)$, and the probability of taking the action corresponding to branch i at instant n as $P_i(n)$. For fixed-penalty probabilities simulation, the FSSA were reset to occupy the initial state in each arm with equal probabilities so that $\pi_{1,1}(0) = \pi_{2,1}(0) = ½$ and $P_1(0) = P_2(0) = ½$. Figure 1 shows the simulation results for values of c_1 = 0.2, 0.3, 0.4, and c_2 = 0.5. Results are plotted for different trial lengths to visualize the dynamics at different time scales. A step window of 400,000 steps was used in Figure 1(a), while a step window of 200 steps was used in Figure 1(b).



(a) Plot of 400 steps.

(b) Plot of 200 steps at a higher resolution.

Figure 1. Probability, P_1, of selection of action 1 in a two-action environment with $c_2 = 0.5$. Both plots contain graphs for univ-Tsetlin automaton and opt-PSAPA with a depth of D = 10.

The results revealed a dramatic difference in the behavior of opt-PSAPA and univ-Tsetlin automata. It took about 50 steps for opt-PSAPA to converge to the steady state, while univ-Tsetlin automata did not reach steady state even at 400,000 steps. In fact, Figure 2 shows that univ-Tsetlin automata did not even reach equilibrium (i.e., steady state) at 10^6 steps. As a result, opt-PSAPA outperformed univ-Tsetlin automata for lengths up to 35,523 steps for c_1 = 0.4, 89,909 for c_1 = 0.3, and 143,960 for c_1 = 0.2. Thus, opt-PSAPA outperformed univ-Tsetlin for all realistic number of trials. It is very hard to find real-life scenarios where the decision system can wait to process more than

35,000 steps. So, for all practical purposes, opt-PSAPA is better suited to deliver satisfactory results for real-life scenarios.

Figure 1(b) reveals another interesting facet of the behavior of the two FSSA. Note that the 200 steps in the plot of Figure 1(b) are just a high-resolution depiction of the behavior of two types of FSSA in the first 200 steps of Figure 1 (a). The plot may wrongly give the impression that the univ-Tsetlin automaton converges to equilibrium faster than the corresponding opt-PSAPA, but at a much lower performance level. However, Figure 1(a) shows that, in fact, the univ-Tsetln automaton is quite far from the steady state for this range of steps, and continues to converge and eventually surpass opt-PSAPA, albeit after an enormously larger number of steps. It highlights the point that it is never advisable to assume convergence without thoroughly evaluating the full scope of the simulation results. Figure 1(b) clearly indicates, though, that the convergence of opt-PSAPA requires approximately 50 steps. Figure 2 is the same simulation as in Figure 1, but plotted up to 10^6 steps. The plot with such a large step window reveals that the convergence of the univ-Tsetlin automaton with a depth of D = 10 required approximately one million steps versus the 50 steps required by opt-PSAPA.

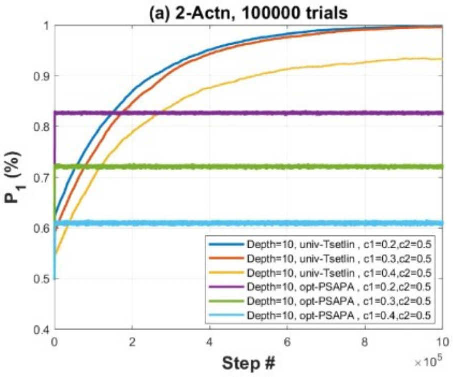


Figure 2. Probability, P_1, of selection of action 1 in a two-action environment with $c_2 = 0.5$. The plot corresponds to 1 million steps and contains graphs for univ-Tsetlin automaton and opt-PSAPA with a depth of $D = 10$.

Results for Position Reversal Simulation

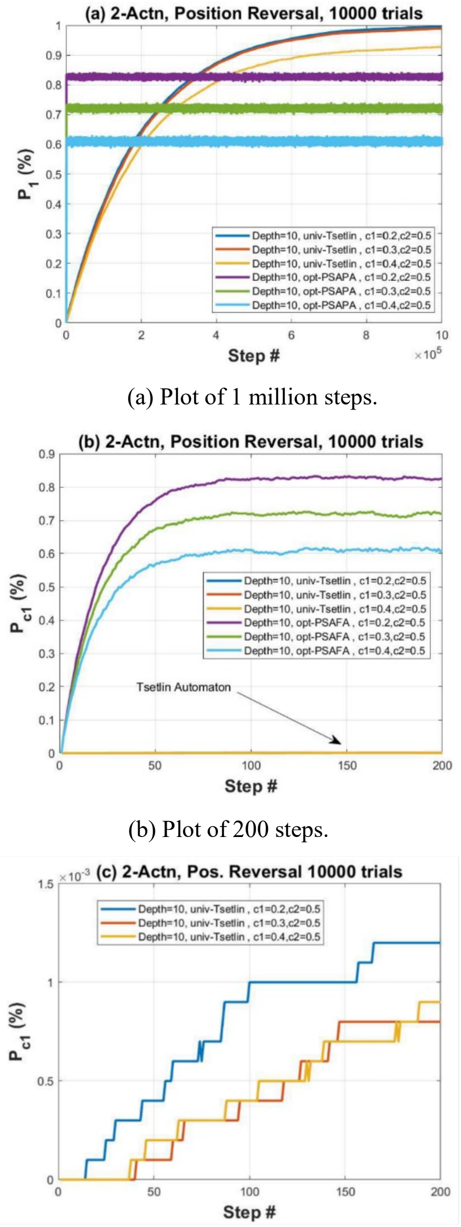
As described earlier, for the second fixed-penalty probability simulation, $\pi_{2,10}(0) = 1$ and $P_2(0) = 1$. Then, just as with the previous simulation, c_2 was set to 0.5 for all simulations and different values of c_1 were selected for different simulations. This simulation was pursued in order to understand how the action-switching capability of opt-PSAPA plays a role in determining its performance. Figure 3 shows the performance results for the two types of FSSA in the position-reversal simulation. Again, P_1 is plotted for different trial lengths. The results clearly show that the choice of the most disadvantaged initial position has little effect on the convergence of the PSAPA: it achieves the steady state after approximately 80 steps. In contrast, this “reversal” initial position has a devastating effect on the univ-Tsetlin automaton. Its probability of obtaining a larger reward (minimum penalty) P_1 remained very close to zero for the first 200 steps [Figure 1(b)], and its approach to equilibrium was also substantially delayed.

Plots of Figures 3(a) and 3(b) contain graphs for univ-Tsetlin automaton and opt-PSAPA at a depth of D=10 for automata that start with $P_2(0) = 1$ and $\pi_{2,10}(0) = 1$ in. It is clear that the time constant for the opt-PSAPA is of the order of 30 steps and the time constant for univ-Tsetlin is of the order of 200k steps for D=10. In Figure 3(c), the same plot of univ-Tsetlin is shown as in Figure 3(b), but with a greater P_1 resolution, spanning the range of 0 to $1.5\text{x}10^{-3}$. Figure 3(c) reveals the dynamics of univ-Tsetlin automaton during the 200 steps. It can be seen that, for the first 17 steps in any of the 10,000 trials, none of the univ-Tsetlin automata switched to action 1. This is the minimum time taken by the univ-Tsetlin automaton to traverse branch 2 before reaching the initial state of branch 1. Even after this initial interval, the rise in the fraction of automata switching to action 1 remained insignificant—below 0.12% for the 200 steps. Thus, action switching remains a challenge for univ-Tsetlin, as compared to opt-PSAPA.

Non-Stationary Environment: Simulation and Analysis

An important characteristic of a non-stationary environment that affects the performance of any LA is how fast the penalty probabilities are changing in time. To evaluate that effect, the most elementary situation was considered when the penalty probabilities oscillated harmonically within a broad range of different frequencies. In the non-stationary environment selected in this study, the penalty probabilities for two of the multiple actions varied sinusoidally, and at opposite phases to each other, around an average value, while the penalty probabilities of other actions remained constant at the same average value. This allowed assessment of the effect of different frequencies and different modulation depths of the sinusoidal variation, as well as to compare the performance in two-action environments with the performance in environments with a larger number of actions.

The simulations were done for three different PSAPAs, univ-Tsetlin, and univ-STAR automata with several values of depth D. Performance measures used in these simulations were defined based on the concepts of Perfect Predictor and Pure-Chance automata. The Pure-Chance automaton was introduced previously as a benchmark for comparison of the behavior of different automata in a stationary environment (Economides & Kehagias, 2002). The Pure-Chance automaton is defined as an automaton that chooses one of the available actions randomly with equal probabilities at each step. The Pure-Chance automaton provides a lower bound on the performance of any intelligent algorithm.



(a) Plot of 1 million steps.

(b) Plot of 200 steps.

(c) Plot of 200 steps at a higher resolution.

Figure 3. Probability, P_1, of selection of action 1 in a two-action environment with $c_2 = 0.5$.

The Perfect Predictor automaton (PPA) is defined as an automaton that has complete knowledge about the environment and can select the action with the least penalty probability at every step. The only hindrance in it getting rewarded for 100% of the trials is the stochastic nature of the environment. PPA sets an upper bound on the best possible performance possible for a learning automaton. In fact, this model is usually a very weak upper bound on the performance of any FSSA, but it does give insight into the limits of how well the different designs can perform.

Simulation Details for Sinusoidally Varying Penalty

Consider a two-action environment where the penalty probabilities vary according to Equation 1:

$$\begin{aligned} c_1(k) &= c + \eta \sin(2\pi kf / N) \\ c_2(k) &= c - \eta \sin(2\pi kf / N) \end{aligned} \qquad (1)$$

where, k is the step number in a sequence of trials (k=1, 2, …, N), and f is the frequency (i.e., the number of periods of sinusoidal variation during the entire sequence of trials, consisting of N steps).

This two-action model is a specific case of a more general r-action model. In the r-action model, the penalty probabilities of the first two actions vary sinusoidally and in opposite phases around an average value, c, and the penalty probabilities of the other actions remain constant at c. Note that the environment model contains two actions with sinusoidally varying probabilities to ensure that the set of penalty probabilities remains the same for any two steps that are one-half period apart. This condition ensures that the values of the output of the experiment, such as the probability of selecting the best action at any given step, and the probability of selecting a given action at a given step, can be compared and interpreted with other steps that are separated by multiples of half a period from the given one. Selection of an environment with only a single sinusoidally varying penalty probability would not provide this feature to the simulation results. Each of the simulations is an ensemble of Z = 10,000 independent trial sequences. The performance of each LA is evaluated as follows.

Denoted by $\rho(k, f, z)$ the reward obtained by an automaton in the trial z (z = 1,2, …, Z) at step k (k = 1, 2, …, N) of the experiment performed with modulation frequency f. In fact, $\rho(k, f, z)$ is a binary random variable that takes on values {0,1} with probabilities c_i and 1-c_i , respectively, if the LA is in branch i at step k. Consider the average given by Equation 2:

$$\rho(f) = \frac{1}{NZ} \sum_{k=1}^{N} \sum_{z=1}^{Z} \rho(k, f, z) \qquad (2)$$

The empirical value of $\rho(f)$, obtained in a simulation, experiment can be used as an estimate of the expected value of the reward E[$\rho(f)$] = 1–$M(f)$, where $M(f)$ is the average penalty obtained by a given automaton in a non-stationary environment with periodically changing probabilities of penalty, according to Equation 1. Note that the empirical values $\rho(k, f, z)$ fluctuate quite broadly, depending on the entire history of automaton-environment behavior and interaction in any trial of the simulation. Therefore, each experiment was repeated Z = 10,000 times with the view to make the estimate of $\rho(f)$, as given by Equation 2, closer to its expected value 1-$M(f)$.

Denoted by $\rho_{\mathrm{PP}}(f)$ the average reward obtained by the Perfect Predictor Automaton and by $\rho_{\mathrm{PC}}(f)$ the average reward obtained by the Pure-Chance Automaton. Then, the performance measure $\Omega(f)$ for an automaton obtaining the average reward $\rho(f)$ is defined by Equation 3:

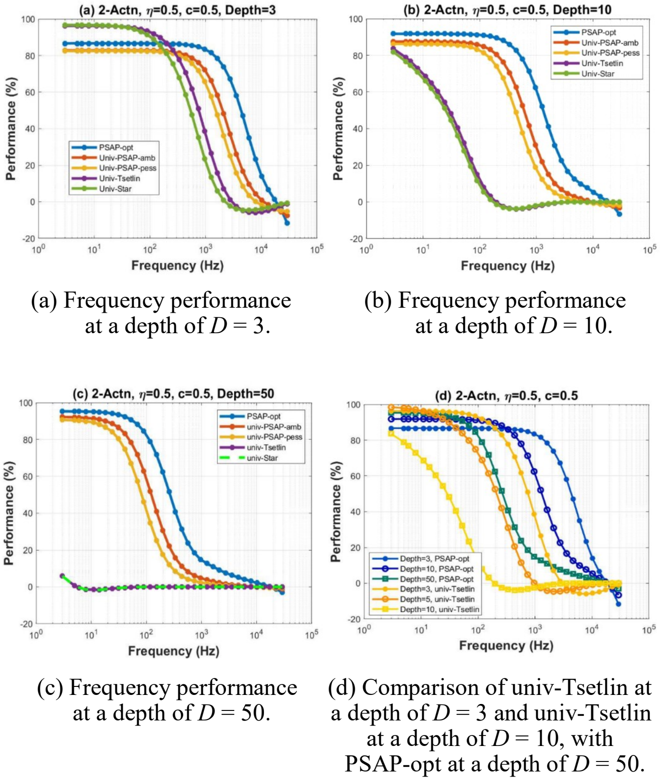
$$\Omega(f) = \frac{\rho(f)-\rho_{PC}(f)}{\rho_{PP}(f)-\rho_{PC}(f)} \tag{3}$$

The simulations were performed for a two-action environment with the value of $c = 0.5$. The frequency f was varied from 3 to 30,000, with 35 approximately regularly spaced steps on a log scale, such that f was always an integer. The number of steps was $N = 100{,}000$ in each of the experiments. The performance, $\Omega(f)$, was calculated as a function of f, the frequency of penalty probability variation, using Equations 2 and 3. To determine the effect of magnitude on variation of the penalty probabilities, $\Omega(f)$ was computed for two values of η, namely, for $\eta = 0.5$ and for $\eta = 0.1$. Comparison of the performance for these two cases should indicate the efficacy of the learning automata to track small changes in penalty probabilities. In a different experiment, simulations for ten-action environments were performed, where the penalty probabilities of the other eight actions (i.e., c_3, c_4 … c_{10}) were held constant at $c = 0.5$. The goal was to determine how the presence of multiple (possibly, non-optimal) actions confounds the performance level of the different learning automata. Although FSSA that are ε-optimal remained so, irrespective of the number of actions, their performance in a non-stationary environment may deteriorate with an increasing number of possible actions.

Thus, the performance of each automaton was characterized in three different types of environments; namely, (a) a two-action environment with sinusoidally varying penalty probabilities; (b) a two-action environment where the penalty probabilities vary within a smaller range; and, (c) a ten-action environment. One may expect that, if an FSSA design is effective for these simple non-stationary environments, its performance should consistently improve with increasing depth D.

Two-Action Environments: Performance versus Frequency and Depth

Simulation results for $\Omega(f)$ in a two-action environment, as described above, are shown for opt-PSAPA and univ-pess-PSAPA, univ-amb-PSAPA, and univ-Tsetlin automata, for the depths $D = 3$ [Figure 4(a)], $D = 10$ [Figure 4(b)], and $D = 50$ [Figure 4(c)]. Assume, for convenience, that the total time of an experiment was one second and f was measured in Hertz. It can be seen that the univ-Tsetlin automaton for $D = 3$ [Figure 4(a)] performed better than others in the low-frequency range ($f < 200$ Hz). This value of f corresponds to 500 steps per modulation cycle period. However, the univ-Tsetlin performance dropped below 80% of its maximum performance ($\Omega = 77\%$) by 337 Hz (i.e., 297 steps/cycle). At the same time, opt-PSAP automata performance dropped to $\Omega = 78.4\%$, only by 1,628 Hz (61 steps/cycle). In the same environment, a Pure-Chance automaton would always have a percentage reward ρ of 50%, irrespective of the penalty modulation frequency value. The cutoff frequency (i.e., the frequency at which an automaton performs worse than Pure-Chance automaton) of univ-Tsetlin $L_{2,6}$ was about 1,200 Hz (83.33 steps/cycle), while that of opt-PSAP automata was more than 22,400 Hz (4.46 steps/cycle). Thus, the PSAPA demonstrated superior performance, compared to univ-Tsetlin automata and univ-STAR automata for all frequencies $f > 200$ Hz, and at a small depth $D = 3$.



(a) Frequency performance at a depth of $D = 3$.

(b) Frequency performance at a depth of $D = 10$.

(c) Frequency performance at a depth of $D = 50$.

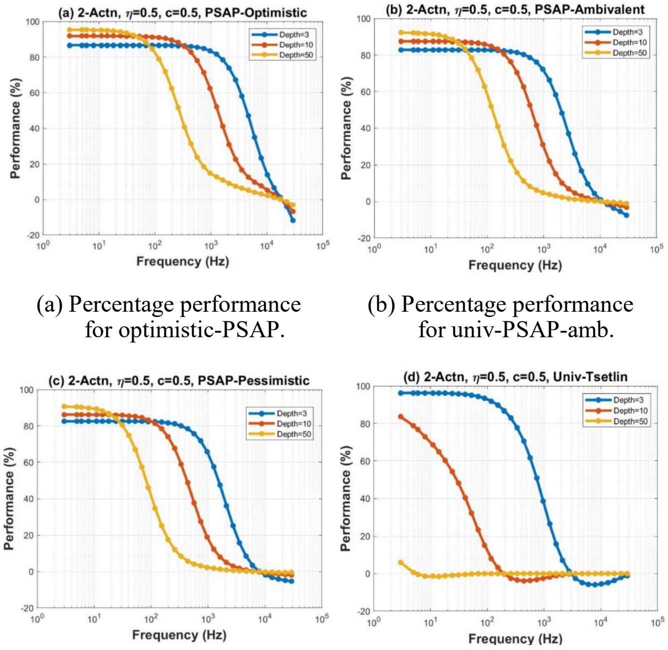
(d) Comparison of univ-Tsetlin at a depth of $D = 3$ and univ-Tsetlin at a depth of $D = 10$, with PSAP-opt at a depth of $D = 50$.

Figure 4. Performance, $\Omega(f)$, as a function of frequency.

The drastic difference In behavior between PSAPA and other FSSA becomes even more profound for larger depths. The depth increase to $D = 10$ had a positive effect on PSAPA, but ruined the performance of the univ-Tsetlin and univ-STAR automata. The cutoff frequency for univ-Tsetlin and univ-STAR dropped to 200 Hz (500 steps/cycle), while for opt-PSAPA it remained above 20,000 Hz (5 steps/cycle), For depth $D = 50$, Tsetlin and STAR automata performed about the same as the Pure-Chance automaton for all frequencies $f > 7$ Hz, while opt-PSAPA kept the cutoff point at about 20,000 Hz. In fact, the performance of opt-PSAPA at low frequencies (up to $f = 200$ Hz) improved with the increase of depth D.

Figure 4(d) shows that, while the opt-PSAPA with a depth of $D = 3$ has a higher working frequency range than all univ-Tsetlin automata of any depth, the opt-PSAPA of depth $D = 50$ is superior to univ-Tsetlin automata with a depth of $D \geq 5$ for low as well as high frequencies. Thus, the opt-PSAPA design is wholly superior to the univ-Tsetlin design in a non-stationary environment.

Figure 5 further illustrates the relationship between frequency response and depth for different universally ε-optimal automata. One can see from this figure that, while the univ-Tsetlin automaton showed superior performance at a depth of $D = 3$, its performance deteriorated considerably as depth increased. Thus, the univ-Tsetlin automaton design does not fully capitalize on the increase in number of states in the automaton, while the opt-PSAPA design is more conducive to penalty-modulated environments. Another observation was that, for the PSAPA design, the cutoff frequency remained almost unchanged with increasing depth, which seemed to be a distinctive property specific to the PSAPA design.



(a) Percentage performance for optimistic-PSAP.

(b) Percentage performance for univ-PSAP-amb.

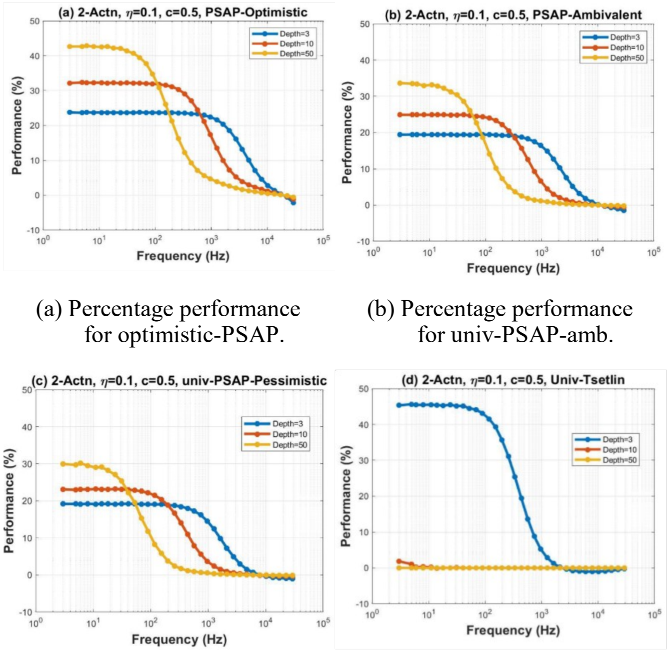
(c) Percentage performance for univ-PSAP-pess.

(d) Percentage performance for univ-Tsetlin.

Figure 5. Percentage performance as a function of frequency for the different ε-optimal automata.

Two-Action Environment: 10% Modulation Amplitude

Figure 6 shows simulation results for a two-action environment with $\eta = 0.1$: for the opt-PSAPA [Figure 6(a)], the univ-amb-PSAPA [Figure 6(b)], the univ-pess-PSAPA [Figure 6(c)], and the univ-Tsetlin [Figure 6(d)] automata for depths $D = 3$, $D = 10$, and $D = 50$. It can be seen that opt -PSAPA again had the best performance, while univ-Tsetlin had much worse performance characteristics. In fact, a univ-Tsetlin automaton with a depth of $D = 10$ already had a nearly Pure Chance performance. This performance level was substantially worse than the case for $\eta = 0.5$ [Figure 5(d)] in the previous section. It becomes clear that the univ-Tsetlin automaton performance worsened in an environment producing small and slowly varying changes in penalty probabilities. Thus, the utilization of additional states by the univ-Tsetlin automaton design was even poorer for such environments. In this regard, the opt-PSAPA design was more conducive to penalty-modulating environments with low-modulation amplitudes, since its performance continued to show improvement similar to the kind of improvement it showed in a high-penalty modulation-amplitude environment with increasing depth. Again, the PSAPA design ensured that the cutoff frequency of the frequency response remained unchanged with increasing depth, irrespective of the penalty-modulation amplitude.



(a) Percentage performance for optimistic-PSAP.

(b) Percentage performance for univ-PSAP-amb.

(c) Percentage performance for univ-PSAP-pess.

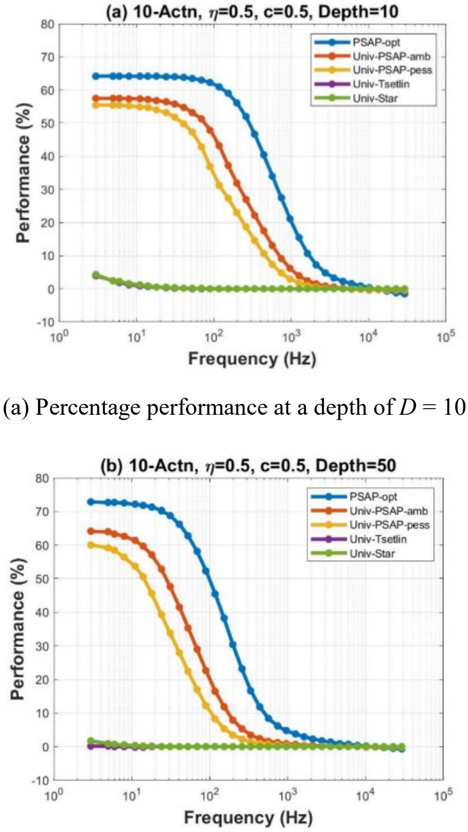
(d) Percentage performance for univ-Tsetlin.

Figure 6. Percentage performance as a function of frequency for depth values of $D = 3$, 10, and 50, while the penalty probabilities modulate with $\eta = 0.1$.

Ten-Action Environment: Performance versus Frequency and Depth

Figure 7 shows the simulation results for a ten-action environment, as described earlier: for Æ-opt, univ-pess-PSAPA, univ-amb-PSAPA, and univ-Tsetlin automata for

D = 10 [Figure 7(a)] and *D* = 50 [Figure 7(b)]. Note that the performance of the Perfect Predictor automaton should not be affected by the number of actions available.



(a) Percentage performance at a depth of *D* = 10.

(b) Percentage performance at a depth of *D* = 50.

Figure 7. Percentage performance as a function of frequency for the different ε-optimal automata in a ten-action environment, as described in section 4C.1.

The presence of many actions affected the LA performance negatively. Nevertheless, it can be seen that performance of PSAPA improved with increasing depth for the range of low frequencies, while keeping the cutoff frequency almost unchanged at about *f* = 10,000 Hz. The Tsetlin automaton performance deteriorated to a Pure Chance level for almost the entire range of frequencies. Thus, the PSAPA design continued to capitalize on increasing depth in the ten -action environment, while the performance of univ-Tsetlin automaton versus depth got worse in the ten-action environment, when compared to the two-action environment.

Conclusions and Future Work

In non-stationary environments, simulation results of two-action and ten-action environments were presented in this paper. Note that the state of the automaton prior to entering the stationary environment was carefully selected to reflect the impact of penalty switching on the behavior of the automaton. Penalty switching simulations revealed the fast convergence rate as well as the quick-action switching property of PSAPA in comparison to other FSSA. These simulations clearly established the benefits of using PSAPA in real-life situations, since they require an incomparably smaller number of steps to converge to the steady state than the univ-Tsetlin automaton, and generally perform better than univ-Tsetlin automaton for a large range of steps. In fact, univ-Tsetlin automaton took an unrealistic number of steps to outperform PSAPA.

Position reversal simulations presented in this paper further accentuated the difference in the convergence speed of PSAPA versus univ-Tsetlin automata. The motivation for position reversal simulations was also derived from real-life, human decision-making behavior. In many scenarios, one finds that people who are extremely sure about a particular decision can struggle when the environment suddenly changes and they need to switch their decision. On the other hand, other people find switching their decisions when faced with a changing environment to be much less taxing and they evolve quickly in a transient environment. Thus, learning automata models discussed in this paper could play a significant role in explaining differences in behaviors of different individuals in changing environments.

The sinusoidally modulated penalty-probability simulations indicated that the performance of univ-STAR and univ-Tsetlin automata deteriorated substantially in simple, non-stationary environments with the increase in depth. On the other hand, the PSAP automata showed a nearly constant high-frequency response for increased depth, and an improved low-frequency response. Thus, although the response of PSAPA may be worse than Tsetlin automata at very low depths, PSAPA is a more favorable candidate for an automaton that capitalizes on the increase of the number of states to improve performance.

This trend in performance is even more pronounced when the modulation amplitude fluctuations are small, or when environment provides many actions to choose from. In the multi-action environment, although the opt-PSAPA performance deteriorated slightly with the maximum operating frequency dropping from 300 Hz in a two-action environment to 100 Hz in a ten-action environment, the deterioration was not as severe as for the univ-Tsetlin automaton, which stopped responding to the modulations in the penalty probabilities entirely. All these observations indicate the strength of the PSAP framework in non-stationary environments. Another strength of the PSAPA design is the degree of flexibility in the assignment of the action-switching probabilities. For the classical FSSA, action switching happens only in a few of the automata states and, therefore, action-switching probabilities are hard to control, since they are only implicitly related to the state-transition probabilities. However, the explicit attribution of one state-switching probability to action switching for every state helps the automaton designer control the action-switching probability in the PSAPA framework. This flexibility creates many more design possibilities for future research.

There exists a general notion in FSSA research that the parameter D can be fine-tuned to provide the best tradeoff between optimality and speed of response to non-stationary environments. This notion is promulgated by the observation that the performance of classical FSSA designs in non-stationary environments degrades drastically with increasing depth (Economides & Kehagias, 2002). Based on simulations in this study, non-stationary environment performance degradation with depth is not necessary for every FSSA design. Further research in the field of FSSA should focus on identifying FSSA designs that do not suffer from degradation in performance with the depth D, especially since, as mentioned in the previous paragraph, it is possible to simulate infinite-state FSSA that are optimal for stationary environments by taking into consideration the finiteness of the number of steps in real-life scenarios.

The action-switching capability in every state for a PSAPA makes it comparable to VSSA in terms of its action -switching capability. In a VSSA, action switching can occur at any time, since the VSSA design does not commit to any one action but instead picks the next action from the basket using probability-based weighting. The performance of PSAPA will be compared with different VSSA designs in future studies. It needs to be determined whether the simulations in simple, non-stationary environments presented in this paper can be used to comprehend the behavior of these different learning automata in complex, non-stationary as well as game-theoretic environments.

In the past, simulation results of opt-PSAPA working in scheduling of multiple devices using a single channel in an ad-hoc network (Aggarwal & Liu, 2005) were presented. With a clearer understanding of the behavior of PSAPA through the two papers presented here, investigation of further applications of PSAPA are planned in the future. In future studies, the authors intend to apply these novel designs of LA to deep learning and real-life challenging problems, such as training deep neural networks and clustering (Guo, Li, Qi, Guo & Xu, 2020; Hasanzadeh-Mofrad & Rezvanian, 2018), financial portfolio management (Sbruzzi, Leles & Nascimento, 2018), adaptive recommender systems (Ghavipour & Meybodi, 2016), resource-efficient cloud computing and cost-efficient resource allocations (Yazidi & Hammer, 2018), wireless network design and management (Misra, Chatterjee & Guizani, 2015), stochastic queuing systems (Vahidipour & Esnaashari, 2018), machine vision (Damerchilu, Norouzzadeh & Meybodi, 2016), and optimization of cooperative tasks (Zhang, Wang & Gao, 2021), where efficient LA with fast convergence rates and assured ε-optimality or real-time reaction for each iteration are required.

References

Aggarwal, P. S., Liu, C., & Levitin, L. (2022). *A New Class of Learning Automata: The Probabilistically-Switch-Action-on-Penalty Automaton: Part 1 – Analytical Models*. https://www.techrxiv.org/articles/preprint/A_New_Class_of_Learning_Automata_The_Probabilistically-Switch-Action-on-Failure_Automaton_Part_1_Analytical_Models/19425761/1

Aggarwal, P. S., & Liu, C. (2005). Coordination within multiple learning automata agents: A novel distributed permission switching protocol. In C. Thompson, & H. Hexmoor (Eds.), *Integration of Knowledge Intensive Multi-Agent Systems*. [Conference presentation]. International Conference on integration of knowledge intensive multi-agent systems, 53-58. https://doi.org/10.1109/KIMAS.2005.1427052

Damerchilu, B., Norouzzadeh, M. S., & Meybodi, M. R. (2016). Motion estimation using learning automata. *Machine Vision and Application, 27*(7), 1047-1061. https://doi.org/10.1007/s00138-016-0788-0

Economides, A. A., & Kehagias, A. (2002). The STAR automaton: expediency and optimality properties. *IEEE Transactions on Systems, Man and Cybernetics, Part B (Cybernetics), 32*(6), 723-737. https://doi.org/10.1109/TSMCB.2002.1049607

Ghavipour, M., & Meybodi, M. R. (2016). An adaptive fuzzy recommender system based on learning automata. *Electronic Commerce Research & Applications, 20*, 105-115. https://doi.org/10.1016/j.elerap.2016.10.002

Guo, H., Li, S., Qi, K., Guo, Y., & Xu, Z. (2020). Learning Automata Based Competition Scheme to Train Deep Neural Networks. *IEEE Transactions on Emerging Topics in Computational Intelligence*, *4*(2), 151-158.

Hasanzadeh-Mofrad, M., & Rezvanian, A. (2018). Learning automata clustering. *Journal of Computational Science, 24,* 379-388.

Misra, S., Chatterjee, S. S., & Guizani, M. (2015). Stochastic learning automata-based channel selection in cognitive radio/dynamic spectrum access for WiMAX networks. *International Journal of Communication Systems, 28*(5), 801-817. https://doi.org/10.1002/dac.2704

Narendra, K. S., & Thathachar, M. A. L. (2012). *Learning Automata: An introduction*. Englewood Cliffs, NJ: Dover Publications.

Sbruzzi, E. F., Leles, M. C. R., & Nascimento, C. L. (2018). Introducing learning automata to financial portfolio components selection. *Proceedings of the SysCon Conference*. Paper presented at 2018 Annual IEEE International Systems Conference (SysCon), Vancouver, BC, Canada. https://doi.org/10.1109/SYSCON.2018.8369522

Tsetlin, M. L. (1962). On the behavior of finite automata in random media. *Automatika I Telemechanika, 22,* 1210-1219.

Vahidipour, M., & Esnaashari, M. (2018). Priority assignment in queuing systems with unknown characteristics using learning automata and adaptive stochastic Petri nets. *Journal of Computational Science, 24*, 343-357. https://doi.org/10.1016/j.jocs.2017.08.000

Yazidi, A., & Hammer, H. L. (2018). Solving stochastic nonlinear resource allocation problems using continuous learning automata. *Applied Intelligence (Boston), 48*(11), 4392-4411. https://doi.org/10.1007/s10489-018-1201-7

Zhang, Z., Wang, D., & Gao, J. (2021). Learning Automata-Based Multiagent Reinforcement Learning for Optimization of Cooperative Tasks. *IEEE Transactions on Neural Networks and Learning Systems, 32*(10), 4639-4652. https://doi.org/10.1109/TNNLS.2020.3025711

Biographies

PRATEEK S. AGGARWAL received his PhD in biomedical engineering from Boston University in 2003. He has been working as a data scientist in different companies since 2005, including Oracle Retail, Adobe, and JP Morgan & Chase. He is currently employed as a senior data scientist with MasterCard. He is a member of IEEE. Dr. Aggarwal may be reached at prateekji@gmail.com

CHENHUI LIU works as a Principal Consultant in the Data Analytics group at Slalom Consulting, helping many corporate clients accelerate business outcomes with rapid insights backed by AI/ML. Her research interests include learning automaton design, efficient encoding and decoding of learning processes, AI and its applications to portfolio optimization and wealth management, default and bankruptcy prediction, and optimal design of learning paths with interactive cross-media systems. She received her PhD in Computer Engineering from Boston University in 2006. She is a member of IEEE. Dr. Liu may be reached at dr.chenhui.liu@ieee.org

LEV LEVITIN is the Distinguished Professor of Engineering Science with the Department of Electrical and Computer Engineering at Boston University. He has published over 200 papers, presentations, and patents. His research interests include information theory, quantum communication systems, physics of computation, quantum computing, quantum theory of measurements, mathematical linguistics, theory of complex systems, coding theory, theory of computer hardware testing, reliable computer networks, and bioinformatics. He is a life fellow of IEEE, a member of the International Academy of Informatics and other professional societies. Prof. Levitin may be reached at levitin@bu.edu

O(1) for Amplitude Amplification in Grover's Algorithm and Its Quantum Circuit

Ying Liu, Savannah State University

Abstract

Grover's algorithm is a category of quantum algorithms that can be applied to many problems through the exploitation of quantum parallelism. The amplitude amplification in Grover's algorithm is $T = O(\sqrt{N})$. In this paper, a quantum circuit will be constructed for amplitude amplification such that the time complexity is $T = O(1)$.

Introduction

Quantum computing is a field of computing that leverages the principles of quantum mechanics (Nielsen & Chuang, 2010; Rieffel & Polak, 2011; Johnston, Harrigan & Gimeno -Segovia, 2019; Hidary, 2019; Aaronson, 2013). Traditional computers use bits as the fundamental unit of information, which can exist in one of two states: 0 or 1. Quantum computers, on the other hand, use qubits, which can exist in multiple states simultaneously. Quantum computing has strange phenomena known as superposition and entanglement.

Grover's algorithm (Grover, 1996) is one of the most famous quantum algorithms and provides a quadratic increase in speed over the best classical algorithms for unstructured search problems [i.e., from T = O(N) to $T = O(\sqrt{N})$]. It was proposed by Grover (1996) and is a fundamental algorithm in the field of quantum computing. Grover's algorithm can be applied to find a specific item in an unsorted list, and its efficiency arises from the exploitation of quantum parallelism and quantum interference. Furthermore, it is a category of algorithms that can be applied to many problems, such as SAT (Berti, 2022) and Subset Sum (Shirgure, 2024). Both SAT and Subset Sum stop in exponential time with respect to the size of the problem in classic computers. The latest system, run by Google, has a total of 70 operational qubits (Nield, 2024).

Grover's algorithm (Grover, 1996) is:

- Initialization
- Oracle for
- Amplitude Amplification $\left(i = 0; i < O\left(\sqrt{N}\right), i++\right)$
- Measurement

where, specifically:

- Initialization means start with a superposition of all possible states then, if there are $N = 2^n$ possible solutions, where n is the number of qubits, the superposition is created over N states.
- Oracle means introduce an Oracle gate that identifies the target solution.
- Amplitude amplification means apply a series of quantum operations that amplify the amplitude of the marked state and suppress the amplitudes of the other states.
- Repeat amplification means that amplifications are repeated for a certain number of iterations.
- Measurement means that the quantum state is measured and, with high probability, the correct solution is obtained.

In Grover's algorithm, the key is Oracle and amplitude amplification. Amplitude amplification is the constructive and destructive interference that occurs during the amplitude amplification step. The amplitudes of incorrect states experience destructive interference, reducing their probabilities, while the amplitude of the correct state experiences constructive interference, increasing its probability. There have been many attempts to improve Grover's algorithm. In a study by Brassard, Hoyer, Mosca, and Tapp (2022), the authors let α be the probability of a marked state in the initial state, and let the amplitude amplification process be composed of building an initial state, measuring the output, and checking the validity of the marked state, then the process was expected to repeat 1/α times, on average, before a solution was found.

Clearly, if the initial state was restricted to an equal superposition of all states, then α = 1/N and the algorithm had O(N), thus the algorithm had to modify the initial state. The restriction of an equal superposition of all states implies that the initial superposition had a time complexity of O(1); without this condition, the initial superposition can have a much higher time complexity (Liu, 2024). Soni and Rasool (2021) made the comparative analysis between Grover's algorithm and its variants; all the variants had $T = O(\sqrt{N})$. Boyer, Brassard, Hoeyer, and Tapp (1998) proposed a simple formula for the probability of success after any given number of iterations of Grover's algorithm (Boyer et al., 1998). Zalka (1999) showed that Grover's algorithm cannot be parallelized better by assigning different parts of the search space to independent quantum computers. This current study differs from the others noted above, as the proposed algorithm was T = O(1).

In this paper, then, the author will: 1) construct a quantum circuit for amplitude amplification using Grover's algorithm such that the time complexity is T = O(1); 2) introduce X = $\{0,1\}^d$ space and the notation for superposition vectors; 3) describe the well-known Oracle assumption; and, 4) list the well-known gate equations used in this paper. A simple example will be used to show the design.

Basic Notation

Throughout this paper:

n is the number of qubits;
$N = 2^n$ is the number of states;
$L \leq N$ is the number of items in an unsorted list;
M is the unsorted list.

Standard notations will be used, meaning the symbol, X, will be overloaded for both the Instance Space name and Gate name. An Instance Space, X = {0, 1}^{n}, is a set of all instances given in Equation 1:

$$X = \{0\ldots00, 0\ldots01, 0\ldots10, 0\ldots11, \ldots\} \tag{1}$$

An instance of X is x = 00 … 0, or, 0 … 01, … , and |X| = 2^n. An instance, x, is a binary string that can be converted into a decimal number using Equation 2:

$$X = \{0, 1, 2, 3, \ldots N-1\} \tag{2}$$

An instance, x, can be the qubits, and can appear as a binary string or a decimal number. An example is given in Table 1.

Table 1. n = 3 qubit states, which could be either binary or decimal.

x_2	x_1	x_0	$\lvert x_2\, x_1 x_0\rangle = \lvert x\rangle$
0	0	0	$\lvert 000\rangle = \lvert 0\rangle$
0	0	1	$\lvert 001\rangle = \lvert 1\rangle$
…			
1	1	1	$\lvert 111\rangle = \lvert 7\rangle$

The classical reference for quantum Grover's algorithm is the linear search with a time complexity given by Equation 3:

$$T = O(N) = O(2^n) \tag{3}$$

In Grover's algorithm, the starting superposition of all possible states by Equation 4 and is always represented by Equation 5:

$$\left|\psi\right\rangle = \sum_{x=0}^{N-1} a(x)\left|x\right\rangle \tag{4}$$

$$\left|\psi\right\rangle = \frac{1}{\sqrt{N}} \sum_{x=0}^{N-1} \left|x\right\rangle \tag{5}$$

Here, $a(x)$ in Equation 4 are the amplitudes. Equation 5 comes from the product in Equation 6:

$$\left|\psi\right\rangle = \left|x_{n-1}\right\rangle \otimes \ldots \left|x_1\right\rangle \otimes \left|x_0\right\rangle \tag{6}$$

where, the qubits are given by Equation 7:

$$\begin{aligned} \left|x_0\right\rangle &= \frac{1}{\sqrt{2}}\left(\left|0\right\rangle + \left|1\right\rangle\right) = \left|+\right\rangle = H\left|0\right\rangle \\ \left|x_1\right\rangle &= \frac{1}{\sqrt{2}}\left(\left|0\right\rangle + \left|1\right\rangle\right) = \left|+\right\rangle = H\left|0\right\rangle \\ &\ldots \end{aligned} \tag{7}$$

Equation 5 is the starting superposition state (i.e., at the beginning of circuits in this paper, the following section in Equation 8 will be omitted in the design):

$$\left|+\right\rangle = H\left|0\right\rangle \tag{8}$$

where, H is the Hadamard gate.

The time complexity for Equation 5 is O(1). Note that although there are n qubits and each qubit will go through Equation 8, the n steps proceed in parallel so that the time for all qubits together in Equation 7 is O(1). For the same reason, the superposition of ψ in Equation 6 from n qubit, $\{\ldots, x_2, x_1, x_0\}$, is also O(1).

The Well-known Oracle Assumption

In Grover's algorithm the Oracle example means to search through the unsorted list and amplify the probability of finding the target state(s) upon measurement. In this particular example, the Oracle's role is to mark the target number, which is given. For example, let n = 3, and let a search target be, |5⟩ = |101⟩, from an arbitrary random list, M = {3, 4, 6, 0, 1, 2, 7, 5}. Since the state |101⟩ is given, the Oracle simply marks the |101⟩ state. Grover's algorithm is a category of quantum algorithms that can be applied to many problems. In general, the Oracle does not need to "know" the solution in the classical sense (Nielsen & Chuang, 2010; Rieffel & Polak, 2011; Johnston et al., 2019; Hidary, 2019; Aaronson, 2013). Instead, it is provided with a black-box function that can identify the solution(s) without explicitly knowing what they are. This black box evaluates whether a given state represents a solution and marks it accordingly.

The basic assumption of the Oracle is not to reveal the solution but rather to provide a quantum operation that efficiently marks the solution state(s) within the superposition of states. This marking process is essential for subsequent steps in amplitude amplification, which enhances the probability of measuring the solution state upon measurement. Again, the significance of the Oracle lies in its ability to mark the solution state(s) within a superposition of states efficiently, without explicitly knowing the solution beforehand. In other words, the definition of the Oracle operation is to efficiently identify the solution state(s) within the quantum superposition without requiring explicit knowledge of the solution beforehand.

Consider the following example: you have a box of 100 different types of candies (superposition), which are equally likely to appear (1%). There is a cavity (Oracle gates) in the bottom of the box. If you open this box (measurement), you could get one candy of a random type (i.e., 1% probability for each type). You begin to shake the box (quantum parallelism) and a small candy (oracle-vector) falls out of the box from the cavity. This particular type of candy can pass through the cavity, because of its shape and size (oracle gates). For the same reason, other types of candies cannot fall through. Even if it is out of the box, the candy (oracle-vector) is still entangled with the other 99 types of candies, so that if you open the box (measurement), the chance for this candy that previously fell out (oracle-vector) to appear would still be 1%.

In other words, the black box can identify the solution(s) without explicitly knowing what they are, because of low probability. Therefore, the Oracle's goal is not to reveal the solution but rather to provide a quantum operation that marks the solution state(s) within the superposition of states efficiently. At this point, you would have to spray red ink on the candy that fell out (oracle-vector) to mark it, at which point the Oracle's job ends. In this paper, Equation 19 represents this candy (oracle-vector).

The Gate Equations

All of the gate equations used in this paper are well known (Nielsen & Chuang, 2010; Rieffel & Polak, 2011; Johnston et al., 2019; Hidary, 2019; Aaronson, 2013). The gates are X, Z, Controlled Z, and Hadamard (optional). The Pauli gate X is represented by Equation 9. Figure 1 shows this gate.

$$X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \tag{9}$$

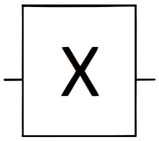


Figure 1. Pauli X Gate (Not Gate, and Bit-flip Gate). Reprinted with permission.

Equation 10 indicates that $|+\rangle$ is an eigenvector of X; in other words, $|+\rangle$ is measurable along the X direction with a probability of 100% for the eigenvalue of 1. Similarly, $|-\rangle$ is measurable along the X direction with a probability of 100% for the eigenvalue −1.

$$X|+\rangle = |+\rangle, X|-\rangle = -|-\rangle \tag{10}$$

The Pauli X gate is also a "Not" gate in the Pauli Z direction, represented by Equation 11:

$$X|0\rangle = |1\rangle, X|1\rangle = |0\rangle \tag{11}$$

Equation 11 will be used for flipping the control qubits. The Pauli gate Z will be used to rotate a qubit from $|+\rangle$ to $|-\rangle$ in Equation 12:

$$Z|+\rangle = |-\rangle \tag{12}$$

The Hadamard gate (H gate) can be used to change base vectors in Equation 13:

$$\begin{aligned} H|+\rangle = |0\rangle\, H|-\rangle = |1\rangle \\ XH|+\rangle = |1\rangle X\, H|-\rangle = |0\rangle \end{aligned} \tag{13}$$

Throughout this paper, for simplicity, the measurable variables are the Pauli X matrix (i.e., the measurement is along the X-direction). This measurement direction can be changed from X to Z by Equation 13. The Controlled Z gate is given by Equations 14-16:

$$CZ = |0\rangle\langle 0| \otimes I + |1\rangle\langle 1| \otimes Z \tag{14}$$

$$CZ|0\rangle|x\rangle = |0\rangle|x\rangle \tag{15}$$

$$CZ|1\rangle|x\rangle = |1\rangle \otimes Z|x\rangle \tag{16}$$

Equations 14, 15, and 16 are the main equations used in this paper. Figure 2 shows the Controlled Z gate (CNot, Controlled Not).

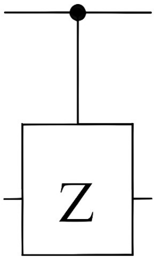


Figure 2. Controlled Z gate. Reprinted with permission.

T = O(1) Circuit Design

Given an unsorted list in Equation 17, and assuming that this list can be encoded by *n* qubits, the measurement-vector is defined by Equation 18. The Oracle black box marks the target number with *n* qubits, the oracle-vector, which is the output of the black box in equation 19:

$$M = \{m_0, m_1, \ldots, m_{L-1}\} \tag{17}$$

$$x = \{x_{n-1}, \ldots, x_1, x_0\} \tag{18}$$

$$o = \{o_{n-1}, \ldots, o_1, o_0\} \tag{19}$$

The circuit will have 2n qubits: *n* qubits in the oracle-vector, which come out of the Oracle black box, and *n* qubits in the measurement-vector. The oracle-vector in Equation 19 represents the control qubits. The measurement-vector in Equation 18 represents the measurement qubits. Equation 20 pairs the two groups of qubits so that the control qubits can be applied to Z gates to control measurement qubits.

$$ox = \left\{\left(o_{n-1}, x_{n-1}\right), \ldots, \left(o_1, x_1\right), \left(o_0, x_0\right)\right\} \qquad (20)$$

The general design will be given in the next section. In this section, a simple example will be used to show the circuit design. The design is given in Figure 3.

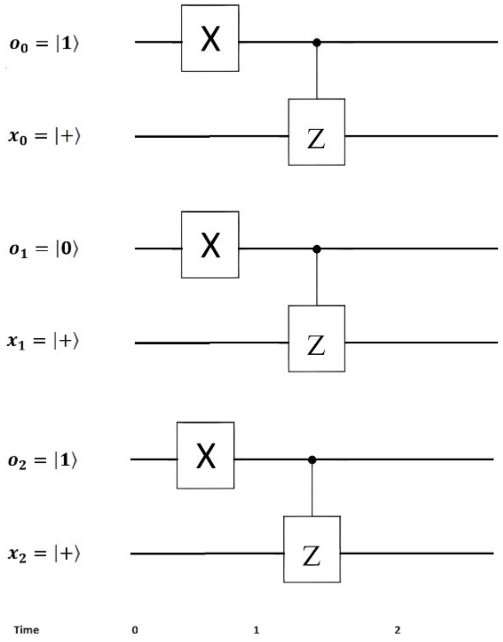


Figure 3. An example of O(1) Circuit Design. The picture is made by the author with basic gate components taken from Wikipedia with permission.

A simple example is finding a target, 5, from M = {3, 4, 6, 0, 1, 2, 7, 5}. The Oracle will mark target, |5⟩ = |101⟩, which is given in this example. The amplitude amplification is to apply a series of quantum operations that amplify the amplitude of the marked state and suppress the amplitudes of the other states. In the following, the measurement direction is Pauli X. At Step 0 in Figure 3, the oracle-vector is:

$$o = \left\{o_2 = |1\rangle, o_1 = |0\rangle, o_0 = |1\rangle\right\}$$

The superposition qubits, based on Equation 5, are:

$$x = \left\{x_2 = |+\rangle, x_1 = |+\rangle, x_0 = |+\rangle\right\}$$

At Step 1 (after X gate in Figure 3), using Equation 11, the oracle-vector is:

$$o = \left\{o_2 = |0\rangle, o_1 = |1\rangle, o_0 = |0\rangle\right\}$$

Note that the X gates flipped the control qubits from 101 to 010. This step selects the control qubit so that only the middle qubit is chosen. This will prepare the next step, which will change the measurement-vector from + + + to + − +, where 7 (+ + +) is the initial state and 5 (+ − +) is the target state. Again, Pauli X is the measurement direction. At Step 2 (after Controlled Z gate in Figure 3), using Equations 14-16, the Z gate only changed the middle qubit. From Equation 12, the measurement qubits are:

$$x = \left\{x_2 = |+\rangle, x_1 = |-\rangle, x_0 = |+\rangle\right\}$$

At this point, note that Z gate rotated the x_1 qubit from $x_1 = |+\rangle$ to $x_1 = |-\rangle$. All three qubits are the eigenvectors of the Pauli X matrix, see Equation 10; so, if the measurement-vector is measured at this point, the only outcome is 101. At Step 3, take a measurement; the probability P (|101⟩) =1 and the target, 5, is identified. Using another example, assume that the problem is finding a target, 2, from M = {3, 4, 6, 0, 1, 2, 7, 5}.

At Step 0,

$$o = \left\{o_2 = |0\rangle, o_1 = 0|1\rangle, o_0 = |0\rangle\right\}$$
$$x = \left\{x_2 = |+\rangle, x_1 = |+\rangle, x_0 = |+\rangle\right\}$$

At Step 1,

$$o = \left\{o_2 = |1\rangle, o_1 = |0\rangle, o_0 = |1\rangle\right\}$$

At Step 2,

$$x = \left\{x_2 = |-\rangle, x_1 = |+\rangle, x_0 = |-\rangle\right\}$$

At Step 3, take a measurement; the probability P (|010⟩) = 1 and the target, 2, is identified.

- Comment 1: Although the quantum state in Equation 18 is measured with a high probability to obtain the correct solution, if the quantum state in Equation 19 is measured, it has a very low probability of obtaining the correct solution. In fact, Equation 19 has a distribution of Equation 5. Thus, amplitude amplification is necessary. Oracle and amplitude amplification fulfill different roles.
- Comment 2: Equation 14 allows a change in the measurement direction from X to Z by inserting H gates at the end of Figure 3 for *x* so that the measurement direction is not important theoretically, although it could be very important in implementation of the circuit.

- Comment 3: The amplitude amplification time is significantly reduced, at the cost of *n* additional qubits in Equation 19, as compared to Grover's algorithm.

General Design and Modified Search Algorithm

The algorithm proposed in this paper is:

- Initialization
- Oracle
- O(1) Amplitude Amplification
- Measurement

Figure 3 can be repeated for an arbitrary number of qubits, which is Figure 4. The input for Figure 4 is Equation 20, which consists of the oracle-vector in Equation 19 and the measurement-vector in Equation 18. The oracle-vector is the output of the Oracle black box, which is in the state of superposition and entanglement and, thus, its correct measurement probability is small.

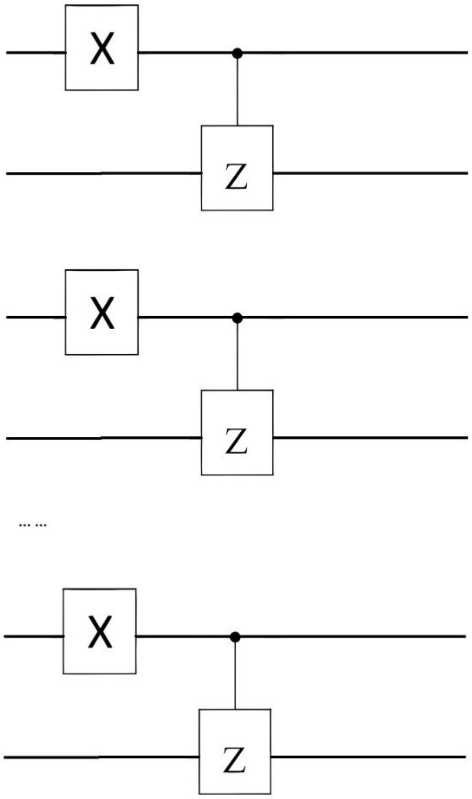


Figure 4. The O(1) circuit design for amplitude amplification. The input is Equation 20, which consists of the oracle-vector in Equation 19 and the measurement-vector in Equation 18. The measurement is along the Pauli X direction. The image was created by the author with basic gate components taken from Wikipedia, with permission.

The time complexity for the design in Figure 4 is clearly O(1), because of the parallel gates. Let the above algorithm be applied to the following unsorted search:

- the initialization given in Equation 5 is O(1);
- the Oracle is O(1), because it simply passes the target through the Oracle's black box;
- the amplitude amplification is O(1), which is the purpose of this paper; and,
- the measurement is O(1).

So, the unsorted-list search has a time complexity of O(1).

Discussion

In computer science and computational complexity theory (Arora & Barak, 2009; Papadimitriou, 1994; Garey & Johnson, 1979; Sipser, 2012), P and NP are classes of decision problems. P = Polynomial Time and NP = Nondeterministic Polynomial Time. P = NP is one of the most famous unsolved problems in computer science and mathematics. There is no proof that P = NP, and the prevailing belief among most experts is well known: P < NP.

Quantum computing leverages the principles of superposition and entanglement. An O(1) step in a quantum computer can be simulated by a classical computer in O(2^n) steps, where *n* is the number of qubits. This is because of Equation 5, the classical time complicity is O(N) = O(2^n). Conversely, in some situations, O(2^n) steps in a classical computer can be simulated in a quantum computer in O(1) steps. This provides a potential for an O(2^n) increase in speed in time complexity. This is the potential power of quantum parallelism.

If an O(2^n) speed could be applied to the NP computation class, then the time complexity of an NP problem would run in polynomial time. This would have significant implications, as it would imply that many computationally difficult problems—such as the traveling salesman problem, the Boolean satisfiability problem, etc.—could be solved in a quantum computer with polynomial times. If the algorithm (summarized in the section on General Design and Modified Search Algorithm) is applied to an NP problem, the time complexity of NP problems is reduced to the time complexity of the Oracle black box:

- the Initialization is O(1);
- the amplitude amplification is O(1), which is the purpose of this paper; and,
- the measurement is O(1).

In terms of implementation of the algorithms, the current limitations in quantum computing hardware are on the order of 100 qubits. The latest system, run by Google, has a total of 70 operational qubits (Nield, 2024). This means that the applicability of the proposed algorithms for solving various problems is quite possible.

Conclusions

In this paper, the author proposed a quantum circuit for the amplitude amplification in Grover's algorithm such that the time complexity was T = O(1), resulting in a significant reduction in the amplitude amplification time.

Acknowledgments

The author would like to thank Gina Porter for proofreading this paper.

References

Aaronson, S. (2013). Quantum Computing Since Democritus. *Cambridge University Press, 2013.*

Arora, S., & Barak, B. (2009). Computational Complexity: A Modern Approach. *Cambridge University Press*.

Berti, A. (2022). Behind Oracles: Grover's Algorithm. https://towardsdatascience.com/behind-oracles-grovers-algorithm-amplitude-amplification-46b928b46f1e

Boyer, M., Brassard, G., Hoeyer, P., & Tapp, A. (1998). Tight bounds on quantum searching. *Fortsch. Phys. 46*, 493-506. DOI: https://doi.org/10.48550/arXiv.quant-ph/9605034

Brassard, G., Hoyer, P., Mosca, M., & Tapp, A. (2022). Quantum Amplitude Amplification and Estimation, Quantum Computation and Quantum Information. *AMS Contemporary Mathematics*, *305*, 53-74. DOI: https://doi.org/10.1090/conm/305/05215

Garey, M. R., & Johnson, D. S. (1979). Computers and Intractability: A Guide to the Theory of NP-Completeness. *W. H. Freeman & Co*.

Grover, L. K. (1996). A fast quantum mechanical algorithm for database search. *Proceedings of the twenty-eighth annual ACM Symposium on Theory of Computing, STOC 96,* 212-219.

Hidary, J. D. (2019). Quantum Computing: An Applied Approach. *Springer, 2019.*

Johnston, E. R., Harrigan, N., & Gimeno-Segovia, M. (2019). Programming Quantum Computers: Essential Algorithms and Code Samples. *O'Reilly Media, 2019.*

Liu, Y. (2024). Time Complexity of the Oracle Phase in Grover's Algorithm. *American Journal of Computational Mathematics*, *14*, 1-10. doi: 10.4236/ajcm.2024.141001

Nield, D. (2024). Google Quantum Computer Is "47 Years" Faster Than #1 Supercomputer. *Ascience Alert.* https://www.sciencealert.com/google-quantum-computer-is-47-years-faster-than-1-supercomputer

Nielsen, M. A., & Chuang, I. L. (2010). Quantum Computation and Quantum Information. *Cambridge University Press.*

Papadimitriou, C. H. (1994). Computational Complexity. *Addison-Wesley.*

Rieffel, E., & Polak, W. (2011). Quantum Computing: A Gentle Introduction. *The MIT Press, 2011.*

Shirgure, S. (2024). Solving the subset sum problem on a quantum computer. https://sumeetshirgure.github.io/assets/pdfs/presentations/ee_520.pdf

Sipser, M. (2012). Introduction to the Theory of Computation. *Cengage Learning*.

Soni, K. K., & Rasool, A. (2021). Quantum Computation Based Grover's Search Algorithm and its Variations. *Recent Patents on Engineering*, *15*(4), 45-54.

Zalka, C. (1999). Grover's quantum searching algorithm is optimal. *Phys. Rev. A, 60*, 2746-2751. DOI: 10.1103/PhysRevA.60.2746

Biographies

YING LIU is an Associate Professor of Computer Science Technology at Savannah State University. He received his master's degree and PhD in physics from Carnegie-Mellon University, and a master's degree in computer science from the University of South Carolina. Dr. Liu has multiple Microsoft Certifications, including MCSE (Microsoft Certified System Engineer) and MCDBA (Database Administrator). He has published over 60 research papers, and holds one patent and over 30 software copyrights. Dr. Liu has extensive experience in software research and development in image recognition. Dr. Liu may be reached at liuy@savannahstate.edu

Material Design Process-Property Optimization Using Neural Network Modeling of Additive Manufacturing

Sairam Vangapally, Minnesota State University; Shaobiao Cai, Minnesota State University

Abstract

Among additive manufacturing processes, such as binder jet 3D printing, manufacturing can fabricate complex geometrical parts with no support structures without employing heat during the part building process. It is of great interest to many engineering applications—such as biomedical, aerospace, automobile industries. However, the mechanical properties of printed materials vary depending on process parameters and conditions; hence, there is a need to tune the process parameters and conditions for optimal characteristics. In this current study, the authors developed a feed-forward-back propagation artificial neural network model to quantify the relationship between three parameters and compressive strength. The model was developed based on experimental data and was validated with known data. The authors feel that the model can be a useful tool for predicting and increasing design and manufacturing efficiency.

Introduction

Binder jet additive manufacturing technology was originally developed at MIT in 1990 and commercialized in 2010 (Gibson & Stucker, 2010). This technology is capable of printing a variety of materials including metals, sand, and ceramics. Binder jetting is an additive manufacturing process in which a liquid binding agent is selectively deposited on powder particles. The print head strategically drops layers of the binder into powder, which are then bonded together to form a 3D product. The process involves binding, curing, de-powdering, sintering, and finishing. The main technique of manufacturing using binder jet additive manufacturing is as follows: a) the CAD file is sliced into layers and an STL file is generated; b) each layer begins with a thin distribution of powder spread over the surface of a powder bed; c) using a technology similar to ink-jet printing, a binder material selectively joins particles where the object has to be formed; d) a piston, that supports the powder bed and part in progress, lowers so that the next powder layer can be spread and selectively joined; e) this layer-by-layer process is repeated until the part is completed; f) following a heat treatment, unbound powder is removed and the metal powder is sintered together.

In this study, the authors considered the output characteristics of compressive strength along with radial and longitudinal shrinkage rates. These were chosen from the binder jetting application perspective in bone scaffold engineering, as the complex bone structure produced should be dimensionally accurate with compressive strength. The fish-bone diagram of Figure 1 shows the process parameters that affect the output characteristics of samples of binder jet additive manufacturing. The parameters include powder size, layer thickness during binding, part orientation in the bed, drying time during binding, heater power, roller speed, curing temperature, curing time, sintering time, sintering temperature, and sintering atmosphere. Any variation in these parameters changes the output properties, which make the relationship between input process parameters and output properties very complicated. Hence, there is a need to tune the process parameters to achieve a controlled and stable process.

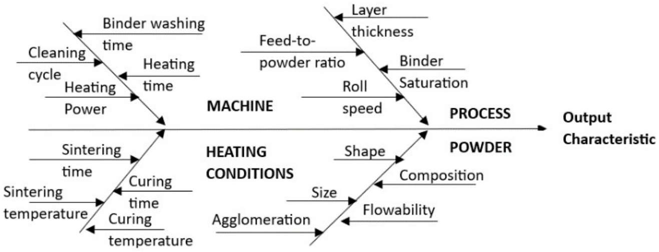


Figure 1. Fishbone diagram representing various parameters involved in the process.

Some experimental work in process parameters and property relationships have been presented in the literature. Torres, Sandback, and Cai (2018) studied some of the building design parameters, such as the effect of temperature to the tensile strength of PLA material for solid samples. To reflect the structure effects in material design, Vangapally, Agarwal, Sheldon, and Cai (2017) studied the effect of lattice arrangement and process parameters on dimensional and mechanical properties. Due to the complexity of the influence of a large number of building factors to the final material properties, studies on different factors are also available. For example, a number of studies were found on the binder setting saturation value, layer thickness, and the location of made-up parts (Yao & Tseng, 2002; Hsu & Lai, 2010; Suwanprateeb, Thammarakcharoen, Wasoontararat & Suvannapruk, 2012).

Also discerned from these prior studies was the fact that practical parts can be made with different materials; in addition to polymer and plasters, metals and alloys were among the popular choices. Investigations into the effects of similar design parameters with metal alloys were carried out as well. In the study by Tang, Zhou, Hoff, Garon & Zhao

(2016), the authors focused on the mechanical properties of SS316 samples made by binder jetting with default process parameters, whereas Cai, Vangapally, and Zhao (2019) compared process parameters such as temperature and particle size to the product properties of SS316 recently used for controlled operations. Research on other alloys, such as bronze infiltrated stainless steel, was presented by Doyle, Agarwal, Sealy, and Schull (2015) related to the effects of layer thickness and orientation of the parts using binder jetting.

Apart from studying the process-property relationship, it is very important to establish a quantitative relationship between process parameters and properties to minimize the cost of experiments. Physics-based modeling is almost impossible to utilize in 3D printing, as it involves powder-binder reaction, curing and sintering. Numerical models can, however, be effective in finding the appropriate parameters with respect to desired output characteristics. Artificial neural networks are a well-known method for use as a numerical model based on experimental data; hence, such a numerical model was developed in this current study for the 3D additive printing process. Figure 2 shows the schematic representation of a neural network generating output values based on input parameters being fed into it. An artificial neural network is a system of mathematical equations working on data and attempting to approximate the human brain. Neural networks consist of neurons connecting each other with respective weights and passing the information.

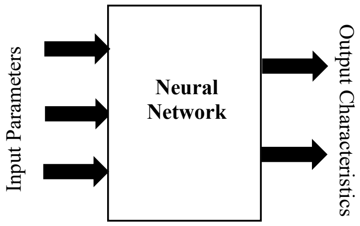


Figure 2. Neural network schematic representation.

The neural network method has been applied in the study of many relatively complicated systems, such as in the study by Cundari and Moody (1997) in which the authors compared neural network models to quantum mechanical models for predicting the mechanical properties of inorganic systems and concluded that neural networks give more accurate predictions. In neural network methods, the feed-forward-back propagation network is a popular choice. For example, Asada, Nakada, Matsumoto, and Uesaka (1997) used a feed-forward-back propagation network to predict the superconducting transition temperature of material as a function of chemical composition. Vermeulen, Morris, deWeijer, and VanderZwaag (1996) used it to predict the finishing temperature of rolling mill as a function of processing parameters. In addition, other researchers have used it to study different materials; for example, to predict the fatigue life of unidirectional composites (Al-Assaf & Kadi, 2001), the use of ceramic materials as a function of composition (Scott, Coveney, Kilner, Rossiny & Alford, 2007).

Though many studies considered neural networks in manufacturing processes, few used them for the additive manufacturing process. Little was done on neural network modeling of additive manufacturing processes. In this current study, the authors aimed to develop a predictive model using the feed-forward-back propagation artificial neural network for the jet binding manufacturing process. A predictive model was designed to define the relationship between process parameters and compressive strength using the experimental data. The feed-forward-back propagation neural network was used to develop a predictive model that established the relationship between process parameters and desired output characteristics.

Experimental Design

In the experimental design, stainless steel 316 powder with a particle size of 30 µm was used. The material was obtained from Ex-One and used with no further treatment. Table 1 shows the chemical composition of stainless steel powder.

Table 1. Chemical composition of SS31 (wt%).

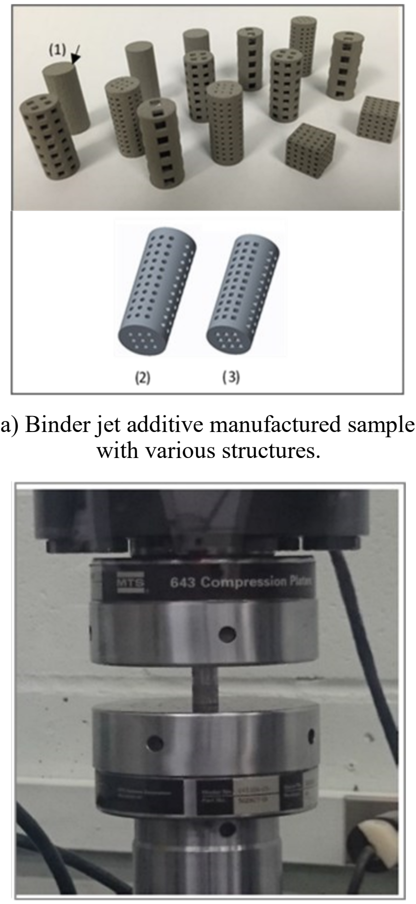
C	Mn	P	S	Si	Cr
0.08 max	2.00 max	0.045 max	0.03 max	0.75 max	16.00-18.00

A full factorial design of experiments was used to test all the possible combinations in current research with three parameters and two levels each, 23 = 8 experiments to be conducted. Table 2 represents the total experiments considered in the study.

Table 2. Full factorial experimental plan: low level is represented as 0 and high level as 1. A (low- 50 µm, high- 100 µm), B (low-2 hours, high- 4 hours), C (low- 1120°C, high- 1180°C).

Experiment	Layer Thickness(A)	Sintering Time (B)	Sintering Temperature (C)
1	0	0	0
2	0	0	1
3	0	1	0
4	0	1	1
5	1	0	0
6	1	0	1
7	1	1	0
8	1	1	1

Figure 3 shows the samples that were created following the design plan. Three types of solid structures were used in this current study: a) circular; 2) 1 mm circular lattice; and, 3) 1 mm cubical lattice structured. All eight experiments for each type of structure were conducted; Figure 3(b) shows the setup. The desired output characteristics for compressive strength were recorded as input to the numerical neural network model. (More details can be found in the previous work done by Cai et al., 2019.)



(a) Binder jet additive manufactured samples with various structures.

(b) Sample testing MTS setup.

Figure 3. Three types of solid structure samples created following the study's design plan and which were used in the study.

Neural Network Model Design

Artificial neural networks (ANN) are preferable tools compared to other available data modeling tools, as they are capable of mapping complex, non-linear relationships between input factors and output characteristics. With training, a neural network with known data can provide approximate output results with unseen data, which makes the technique useful for predictive applications. In the neural network method family, the feed-forward-back propagation neural network is the simplest and most efficient ANN in use today, and is found in applications for developing predictive experimental models. The feed-forward-back propagation neural network with sigmoid activation function was considered in this current study for designing the model.

Figure 4 represents the architecture of the neural network used in this study, where A is layer thickness, B is sintering time, C is sintering temperature, O is compressive strength, Σ represents summation, F(x) is the activation function, and b1 and b2 are bias. There are three different layers in neural networks. The left-most layer is the input layer. Input parameters are fed into the neural network through this layer. The middle is the hidden layer. This layer connects the input and output layers; it is called hidden, as its values are not observed in the training set. The right-most layer is the output layer. In this layer, all of the hidden neurons produce an output. In feed forward, neurons in the input layer are connected to neurons in the hidden layer, whereas neurons in the hidden layer are connected to the output layer. Back propagation is a training method in which neurons adjust their weight to achieve the target output. The network contains three layers with a total of 8 nodes, of which 4 are hidden nodes, 3 are input nodes, and 1 is an output node. Other symbols/parameters defined in the model are listed below.

Xi = input values fed to neural network through input node *i*
$W1_{ij}$ = weights connecting input-hidden nodes, where *i* represents input node and j represents hidden node
$W2_{jk}$ = weights connecting hidden-output nodes, where *j* represents hidden node and k represents output node
b1 = bias at hidden node
b2 = bias at output node
F(x) = activation function
δ_k = error information at output node
δ_j = error information at hidden node
ΔW1 = delta weights at input-hidden layer
ΔW2 = delta weights at hidden-output layer
Z_j = hidden node
Y_k = output node
The sigmoid function is used as the activation function for this model: F(x) =1/(1+*e*^(-x))

For the training, the feed-forward-back propagation process was used. In the feed-forward process, the random-numbered weights for the hidden input and hidden output layers are initialized first. Then the inputs are transferred to nodes in the hidden layers, where the summation of input values with respective node weights take place and are then transferred to the next layer and applying the activation function, as shown in Equations 1 and 2:

$$Z_{in} = \sum X_i * W1_{ij} + b1_i \tag{1}$$

$$Y_{in} = F\left(Z_{in}\right) \tag{2}$$

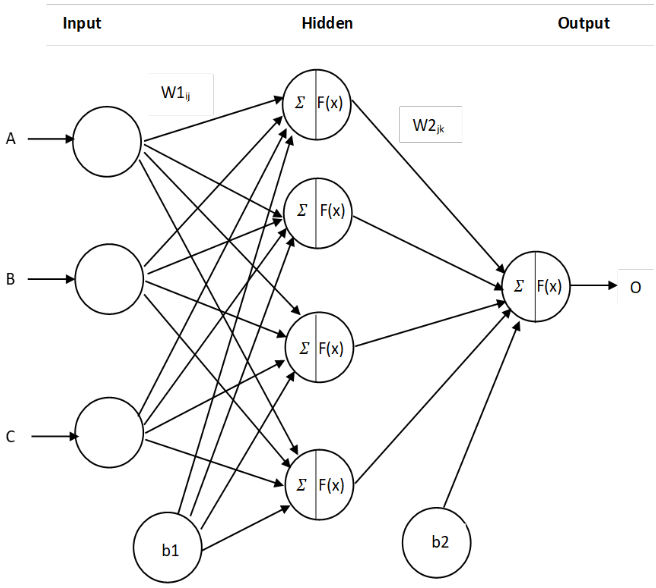


Figure 4. Neural network schematic representation.

The values at the hidden nodes get transferred to the output nodes, where they get multiplied by the respective weights before applying the activation function to produce an output using Equations 3 and 4:

$$Y_{out} = \sum Y_{in} * W2_{jk} + b2_{j} \tag{3}$$

$$Output = F\left(Y_{out}\right) \tag{4}$$

In the feed-forward process, errors and weights at different notes are calculated and updated. The error or margin is calculated by comparing the target value with the output value of the developed model by using Equation 5:

$$e = \left(\text{Target-Output}\right)^{2} \tag{5}$$

The error information at the output is found using Equation 6:

$$\delta_{k} = eF'\left(Y_{out}\right) \tag{6}$$

The error information at the hidden unit is calculated using Equation 7:

$$\delta_{j} = F'\left(Z_{in}\right) * \sum\nolimits_{k} \delta_{k} W2_{jk} \tag{7}$$

The weights, updating at the hidden input layer, are calculated using Equations 8 and 9:

$$\Delta W1 = \delta_{k} Z_{j} \tag{8}$$

$$\left(W1\right)_{new} = \left(W1\right)_{old} + \Delta W1 \tag{9}$$

The weights, updating the hidden output layer, are found using Equations 10 and 11:

$$\Delta W2 = \delta_{j} X_{i} \tag{10}$$

$$\left(W2\right)_{new} = \left(W2\right)_{old} + \Delta W2 \tag{11}$$

The compressive strength value is normalized so that all the values are in the range of 0 to 1, using the Equation 12:

$$O_{i} = \frac{Y_{i} - \min\left(Y\right)}{\max\left(Y\right) - \min\left(Y\right)} \tag{12}$$

where, Y_i represents the compressive strength of each experimental run i (1 to 8).

Figure 5 shows the training process and parameters involved at each step. The neural network was trained such that error between the desired output and the actual output was less than 0.05.

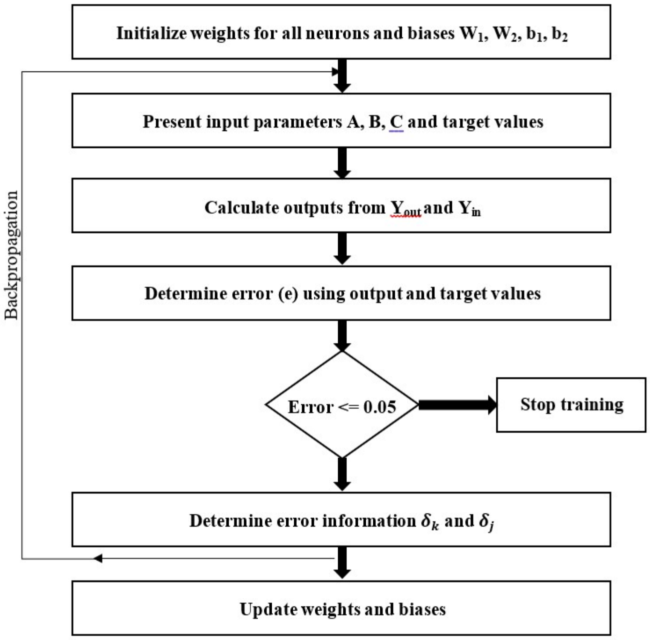

Initialize weights for all neurons and biases W₁, W₂, b₁, b₂
Present input parameters A, B, C and target values
Calculate outputs from Yout and Yin
Determine error (e) using output and target values
Error <= 0.05
Stop training
Determine error information δk and δj
Update weights and biases
Backpropagation

Figure 5. Flowchart showing the entire training process and the parameters involved.

After successful training, the network was tested using new data sets for its performance. Then the value obtained using the network was denormalized to find the predicted value as determined by Equation 13. With it, the difference between the predicted and actual values could be analyzed.

$$Y_{predicted} = \left[Y_{network\ value} * \left(\max\left(Y\right) - \min\left(Y\right)\right)\right] + \min\left(Y\right) \tag{13}$$

Results and Discussion

Experimental analysis was performed to determine the effect and significance of layer thickness, sintering time, and sintering temperature on compressive strength. The collected experimental data were used to the developed a neural network model for predicting compressive strength, given the inputs of layer thickness, sintering time, and sintering temperature. Three cases were conducted. In the first case, the data used for the neural network were from compression testing of a solid structure. In the second case, the data used were from compression testing of a 1-mm circular lattice structure. In the third case, the data used were from compression testing of a 1-mm cubical lattice structure. The neural network used a back-propagation algorithm for training, a sigmoid function as activation function, where the network had one hidden layer.

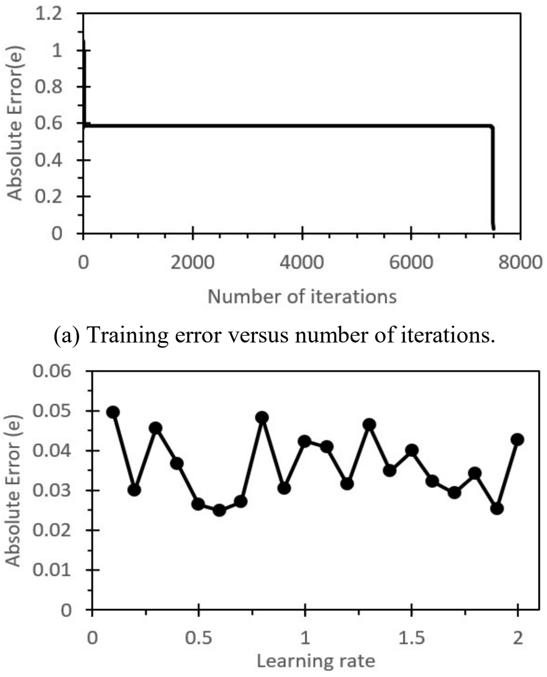
Solid Circular Structure

Table 3 displays the data for the solid circular structure; the data indicate that the compressive strength of the samples fabricated using the eight experimental settings were normalized. The feed-forward-back propagation network was trained with seven datasets, leaving behind the one data set of experiment 7 for testing network performance.

Table 3. Inputs A (layer thickness), B (sintering time), C (sintering temperature), along with the normalized predicted output of compressive strength in the range of 0 to 1 for the solid circular structure.

	A (Layer thick-ness)	B (Sintering time)	C (Sintering Temp)	Y (Comp. Strength, MPa)	Output (O)
1	0	0	0	745.5	0.351
2	0	0	1	1780.5	0.899
3	0	1	0	1811.0	0.915
4	0	1	1	1972.0	1.000
5	1	0	0	82.9	0.000
6	1	0	1	879.5	0.422
7	1	1	0	978.5	0.474
8	1	1	1	1083.5	0.530
Maximum value in Y column				1972.0	
Minimum value in Y column				82.9	

Figure 6 shows the error graph and the different learning rates for the neural network model used during training; Figure 6(a) show the error graph using the error values obtained from each iteration, where the necessary number of iterations required to reach the goal was approximately 7500—high iterations signify the acuteness of the carried calculations. From 15 to 7500 iterations, the error was changing in decimal places. A method was developed such that the neural network stopped training once the error between network output and actual output was less than an absolute value of 0.05. Different learning rates were presented to the network from 0.1 to 2; Figure 6(b) shows training error plotted against learning rate. The maximum error in training allowed was an absolute value of 0.05. The optimum learning rate for minimum error was found to be 0.6 in the training phase for the network. The network was tested for a target value of 0.470 and the value obtained from the network was 0.488, which indicates good performance prediction.



(a) Training error versus number of iterations.

(b) Performance of network architecture for different learning rates.

Figure 6. Neural network model during training of the solid circular structure.

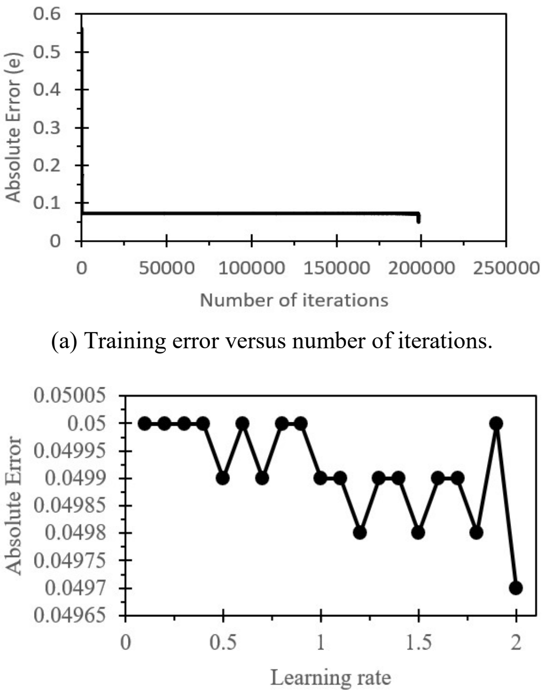
Circular Lattice Structure

The same feed-forward-back propagation network used for predicting the compressive strength of solid was used for training and testing the compressive strength of the 1-mm circular lattice structure. Table 4 shows that the compressive strengths of the samples fabricated using the eight experimental settings were normalized. Data obtained from experiment two were used for network testing, while other remaining data were used to train the network.

Figure 7 shows a plot of the error graph and the different learning rates for the model during training. Figure 7(a) shows the error graph of the values obtained in each iteration, where the necessary number of iterations to reach the goal was approximately 200,000. From 100 to 199,000 iterations, the error was changing in decimal places; hence, the straight line. Figure 7(b) shows the presentation of the different learning rates to the network from 0.1 to 2, where training error is plotted against learning rate. The maximum error in training allowed had an absolute value of 0.05. The optimum learning rate for minimum error was found to be 2 in the training phase for the network. The network was tested for a target value of 0.650, and the value obtained from the network was 0.678.

Table 4. Inputs A (layer thickness), B (sintering time), C (sintering temperature), along with the normalized predicted output of compressive strength in the range of 0 to 1 for the 1-mm lattice structure.

	A (Layer thick-ness)	B (Sintering time)	C (Sintering Temp)	Y (Comp. Strength, MPa)	Output (O)
1	0	0	0	160.3	0.150
2	0	0	1	654.3	0.651
3	0	1	0	739.9	0.738
4	0	1	1	998.6	1.000
5	1	0	0	12.8	0.000
6	1	0	1	303.3	0.295
7	1	1	0	229.3	0.220
8	1	1	1	545.5	0.540
Maximum value in Y column				998.6	
Minimum value in Y column				12.8	



(a) Training error versus number of iterations.

(b) Performance of network architecture for different learning rates of the 1 mm circular lattice structure.

Figure 7. Neural network model during training of the circular lattice structure.

Cubical Lattice Structure

Table 5 shows that the compressive strength of the 1-mm cubical lattice-structured samples fabricated using the eight experimental settings were normalized along with their inputs. Data obtained from experiment six were used for testing the network, while remaining data were used to train the network, as in the previous case.

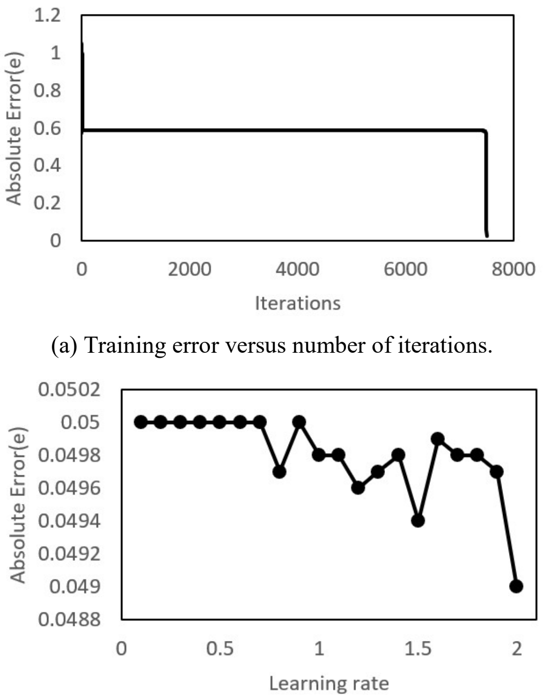
Table 5. Inputs A (layer thickness), B (sintering time), C (sintering temperature), along with the normalized predicted output of compressive strength in the range of 0 to 1 for the cubical 1-mm lattice structure.

	A (Layer thick-ness)	B (Sintering time)	C (Sintering Temp)	Y (Comp. Strength, MPa)	Output (O)
1	0	0	0	154.5	0.151
2	0	0	1	545.2	0.564
3	0	1	0	612.3	0.635
4	0	1	1	958.3	1.000
5	1	0	0	11.6	0.000
6	1	0	1	250.9	0.253
7	1	1	0	253.2	0.255
8	1	1	1	326.4	0.333
Maximum value in Y column				958.3	
Minimum value in Y column				11.6	

Figure 8 shows a plot of the error graph and the different learning rates for the model using the cubical structure during training. Figure 8(a) shows the error graph of the values obtained in each iteration, where the necessary number of iterations to reach the goal was approximately 7500. From 75 to 7500 iterations, the error was changing in decimal places; hence, the straight line. Figure 8(b) shows the presentation of the different learning rates to the network from 0.1 to 2, where training error is plotted against learning rate. The maximum error in training allowed had an absolute value of 0.05. The optimum learning rate for minimum error was found to be 2 in the training phase for the network. The network was tested for a target value of 0.250, and the value obtained from the network was 0.276.

Model Validation

The authors compared this current model with similar studies found in the literature; the results are presented here. In the validation, the study used was the neural network model for predicting the hardness of shielded-metal arc-welded joints, given the inputs of current, voltage, welding speed, and magnetic field (Singh, Gupta & Sarkar, 2013). Table 6 provides a comparative analysis of predictive resultant data to the experimental results, along with work found in the literature. The difference between the experimentally tested HRC hardness and predicted hardness using the current model is also represented in the table. The authors observed that the current neural network model agreed very well with the work and experimental data found in the literature. The minimum difference between the experimental value and the value predicted using current model was 0.04 % and the maximum difference was 8.9%. The simple average among all seven data sets was 3.82%, which demonstrated some improvement to the referenced 4.95%. The small difference was believed to be the different architecture used by the reference model and the current neural network model.



(a) Training error versus number of iterations.

(b) Performance of network architecture for different learning rates of the 1 mm cubical lattice structure.

Figure 8. Neural network model during training of the cubical lattice structure.

Table 6. Comparison and validation of the current neural network model.

	Experiment Hardness (HRC)	Prediction value from the literature (Singh et al., 2013)	Prediction value using current model	Difference %
1	91	85.6	90.96	0.04
2	86	85.1	93.7	8.9
3	89	85.4	93.84	5.4
4	89	85.2	87.6	1.54
5	81	84.8	82.64	2.02
6	78	84.6	82.8	6.1
7	79	83.9	81.04	2.58
Average Difference			4.95	3.82

Model Prediction and Potentials

The data obtained for the above three experimental structure cases were used as inputs to the current neural network to predict compressive strength. The feed-forward-back propagation network was trained with seven datasets; one data set of experiment result was used for prediction comparison analysis for testing the network performance. Upon reaching the normalized prediction, meeting the defined computational accuracy, the normalized compressive strength was denormalized for each of the three cases. The prediction was further compared with the experimental measured results. Table 7 shows the experimental compressive strength of the three different structures obtained from physical experimentation and the predicted value output from the numerical neural network model.

Table 7. Neural network results for different structures.

Structure	Experimental Compressive Strength (MPa)	Predicted Compressive Strength (MPa)	Difference (%)
Solid	978.5	1005.5	2.75
1mm Circular	654.3	681.5	4.15
1mm Cubical	250.9	273.9	9.16

The predicted values were found to be in good agreement with the experimental resultant data, with the maximum difference being 2.75%. The values obtained using the neural network were slightly greater than the values obtained using experimentation. It should also be noted that the performance of the network could be somewhat more, perhaps even quite a bit more robust with additional sets of experimental training input data. The percentage difference between a target value and its prediction can be even closer, as the network gets more data for training. Also, the performance of the model may be improved further either by changing the number of hidden nodes used in the study or by using a different architecture in future studies. The current neural network model was expected to be helpful in identifying the set of parameters to achieve the desired compressive strength, thereby eliminating the extensive time and effort required in experimentation.

The model may be used to develop feed forward artificial neural networks using a back propagation training algorithm, if needed. Data normalization prior to training is crucial for obtaining good results as well as to increase the speed of calculations (Nguyen, Yang, Bae & Choi, 2009). Network architecture can have a significant effect on prediction results. Thus, it would be a good practice to rely on the number of input and output parameters to decide the architecture of a feed forward neural network, the number of hidden layers, and hidden nodes in the structure. Once the architecture has been chosen, a training algorithm and an activation function need to be selected, which is similar to this current study, which used a back propagation algorithm with a sigmoid activation function. With the normalized input values, target values, network architecture, and activation function, the model can be developed according to the training algorithm.

Conclusions

Experimental design techniques were used to design and create the three structures in this study: solid circular, circular with 1 mm lattice, and circular with 1 mm cubical. MTS material testing was performed and experimentation data were corrected. A feed-forward-back propagation neural network with sigmoid activation function was developed. The neural network, with one hidden layer, used a back propagation algorithm as the training algorithm and a sigmoid function as an activation function. The model was validated using data published in literature. Furthermore, experimental data of the three different structures were used as input to the model. Performance analysis of the developed neural network model using the experimental data was performed. The authors found that the model prediction agreed with the experimental results. The developed model predicted the compressive strength given over the range of layer thickness, sintering time, and sintering temperature parameters, and could serve as framework to set up the process design parameters to achieve the desired output characteristics.

The model was developed for the purpose of material structure design optimization. Due to the complexity of multi-parameter multi-level effects on material properties, conducting a large number of experiments to get meaningful results was a challenge. The study showed, though, that the capability of a feed-forward-back propagation neural network was a good technique for determining the compressive strength of binder jetting samples. The analysis revealed the application of neural networks in material science and engineering, particularly in additive manufacturing. The authors saw additional benefits and applications; for example, it can be very efficient, if many factors are involved in a complex process in which the factors play roles at different levels. Broad applications of the model, in additional to material structure designs, can be found, especially for those involving multi-parameter multi-level complex studies in areas such as medical, engineering, or even business. In those studies, a large number of trial-and-error data were typically needed to get meaningful results. However, neural networks would help with better design solutions, providing direction and better results, thereby saving large investments of effort, time, and cost needed for conducting large arrays of experiments.

References

Al-Assaf, Y., & Kadi, H. (2001). Fatigue life prediction of unidirectional glass fiber/epoxy composite laminae using neural networks. *Composite Structures, 53*(1), 65 -71.

Asada, Y., Nakada, E., Matsumoto, S., & Uesaka, H. (1997). Prediction of t-c for yba2cu3oz doped with ca using neural network. *Journal of Superconductivity, 10* (1), 23-26.

Cai, S., Vangapally, S., & Zhao, Y. (2019). Experimental Design Analysis of 3D Printing Processes to the Optimization of Strength of Material. *International Journal of Modern Engineering, 20*(1), 5-11.

Cundari, T., & Moody, E. (1997). A comparison of neural networks versus quantum mechanics for inorganic systems. *Journal of Chemical Information and Computer Sciences, 37*(5), 871-875.

Doyle, M., Agarwal, K., Sealy, K., & Schull, K. (2015). Effect of layer thickness and orientation on mechanical behavior of binder jet stainless steel 420+ bronze parts. *Procedia Manufacturing, 1*, 251-262.

Gibson, D. R., & Stucker, B. (2010). *Additive manufacturing technologies*. Springer.

Hsu, T., & Lai, W.(2010). Manufacturing parts optimization in the three-dimensional printing process by the Taguchi method. *Journal of Chinese institute of Engineers, 33*(1), 121-130. https://doi.org/10.1080/02533839.2010.9671604

Nguyen, T., Yang, Y., Bae, K., & Choi, S. (2009). Prediction of deformation of steel plate by artificial neural network in forming process with induction heating. *Journal of Mechanical Science and Technology, 23*, 1211-1221.

Scott, D., Coveney, P., Kilner, J., Rossiny, J., & Alford, N. (2007). Prediction of the functional properties of ceramic materials from composition using artificial neural networks. *Journal of the European Ceramic Society, 27* (16). DOI:10.1016/j.jeurceramsoc.2007.02.212.

Singh, R., Gupta, R., & Sarkar, S. (2013). Application of artificial neural network for prediction of hardness of shielded metal arc welded joints under the influence of external magnetic field. *International Journal of Engineering Research and Development. 5*(7), 84-88.

Suwanprateeb, J., Thammarakcharoen, F., Wasoontararat, K., & Suvannapruk, W. (2012). Influence of printing parameters on the transformation efficiency of 3D-printed plaster of paris to hydroxyapatite and its properties. *Rapid Prototyping Journal, 18*(6), 490-499.

Tang, Y., Zhou, Y., Hoff, T., Garon, M., & Zhao, Y. (2016). Elastic modulus of 316 stainless steel lattice structure fabricated via binder jetting process. *Material Science Technology, 32*, 648-656.

Torres, P., Sandback, T., & Cai, S. (2018). Parametric Analysis of Building Parameters to Maximize Strength of Material Using Additive Manufacturing. *International Refereed Journal of Engineering and Science (IRJES), 7*(1), 25-34.

Vangapally, S., Agarwal, K., Sheldon, A., & Cai, S. (2017). Effect of lattice design and process parameters on dimensional and mechanical properties of binder jet additively manufactured stainless steel 316 for bone scaffolds. *Procedia Manufacturing, 10*, 750-759.

Vermeulen, W., Morris, P., deWeijer, A., & VanderZwaag, S. (1996). Prediction of martensite start temperature using artificial neural networks. *Ironmaking Steelmaking, 23*(5), 433-437.

Yao, A., & Tseng, Y. (2002). A robust process optimization for a powder type rapid prototype. *Rapid Prototyping Journal, 8*, 180-189.

Biographies

SAIRAM VANGAPALLY is a graduate student in the Department of Mechanical and Civil Engineering at Minnesota State University Mankato. Mr. Vangapally may be reached at sairam.vangapally@mnsu.edu

SHAOBIAO CAI is an associate professor in the Department of Mechanical and Civil Engineering at Minnesota State University Mankato. His areas of expertise and research interests include interfacial contact mechanics and tribology, mechanical failure of materials, layered medium design, and the application of DFMA principles in the development of robust design solutions. He is a registered professional engineer in the State of Minnesota. He is a member of the American Society of Mechanical Engineers, the Society of Tribologists and Lubrication Engineers and American Society of Engineering Education. Dr. Cai may be reached at shabiao.cai@mnsu.edu

The Impact of Printing Direction and Raster Angle on the Part Density and Mechanical Properties of 316L Stainless Steel 3D-Printed Parts Using FDM/ FFF Technology

Suleiman Obeidat, Sam Houston State University; Iftekhar Basith, Sam Houston State University;
Ulan Dakeev, Sam Houston State University

Abstract

Additive manufacturing using Fused Deposition Modeling/ Fused Filament Fabrication (FDM/ FFF) technology is one of the most common methods for 3D printing of parts using filaments. MakerBot Method X was used to print BASF Ultrafuse 316L stainless steel filaments. Bending and tensile testing samples were 3D printed horizontally and vertically at raster angles of 0°, 30°, and 45°. The impact of raster angle on the apparent density and the mechanical properties were studied in this work. It was preferred to print the samples vertically at 45° to get the highest value of the modulus of elasticity. The results showed that there was a significant impact of the raster angle on the apparent density of the green and sintered parts; printing horizontally improved the tensile strength of the material at different raster angles, while printing vertically gave higher flexural strength.

Introduction

Additive manufacturing is one of the operations used to make parts layer by layer instead of cutting materials to shape the part made as in machining processes. In the last two decades, industry has used additive manufacturing extensively in making complicated parts for use in different applications, such as biomedical implants, heat engines, and other applications that need complicated and strong parts with intricate details (Obeidat, Nervis & Ma, 2022). As with many other manufacturing processes, 3D printing technology entails different defective parts that impact the part's properties, such as mechanical properties. Different printing parameters need to be controlled to make parts within the standards needed. Many researchers have studied those factors. Kedziora, Decker, Museyibov, Morbach, Hohmann, Huwer, and Wahl (2022) stated that the tensile and fatigue strength of BASF Ultrafuse 316L stainless steel printed by Free Fused Fabrication (FFF) technology is lower than that of ones made using Selective Laser Melting (SLM) technology, which makes the 316L stainless steel made by FFF technology unsafe for structural applications. Tosto, Tirillò, Sarasini, Sergi, and Cicala (2022) used the design of experiments (DOE) technique to study the impact of nozzle temperature, layer thickness, and flow rate on the tensile and bending properties of the 316L stainless steel printed using FDM technology.

Ferretti, Leon-Cardenas, Ciotti, Santi, Donnici, and Frizziero (2021) studied the impact of the slicing parameters, specifically the line width on the number and volume percentage of the defects in the 3D-printed parts in the FDM technology. Quarto, Carminati, and D'Urso (2021) used Analysis of Variance (ANOVA) to investigate the effect of different printing factors, such as nozzle temperature, on the density and shrinkage of the 316L stainless steel 3D printed using FDM technology. Their results showed no effect of nozzle temperature on bulk density. Spiller, Kolstad, and Razavi (2022) investigated the impact of various printing parameters, such as layer thickness and nozzle temperature, on the mechanical properties of 316L stainless steel. They reported that the strength of the sintered parts is low compared to conventional 316L stainless steel. Ortega Varela de Seijas, Bardenhagen, Rohr, and Stoll (2023) proposed a method that employs induction current to sinter the 316L stainless steel parts produced by FDM technology. They reached a density of 99.8% with a soaking time of about six minutes. Liu, Wang, Lin, and Zhang (2020) investigated the mechanical properties of 3D-printed 316L stainless steel, using FDM technology to print the parts. Their results showed that the material's yield strength, ultimate tensile strength, and elongation at break were lower than those of the AISI and SLM counterparts. Farashi and Vafaee (2022) conducted a meta-analysis on the effects of extruding temperature and printing speed on printed materials' tensile strength and other mechanical properties.

After searching through 560 studies, they concluded that a high extruder temperature has a positive effect on the material, while an increased printing speed has a negative effect. Ryder, Lados, Iannacchione, and Peterson (2018) fabricated ABS-420 stainless steel system composites from scratch to conduct tests on the material. They tested ABS-420 SS with 10 wt.%, 15 wt.%, 23 wt.%, and base ABS. They conducted tensile tests at two different raster angles—0/90 and 45/-45—oriented horizontally and vertically. Their results showed that the metal concentration past 15% weakened the material's mechanical properties significantly compared to the properties of base ABS. They also learned that horizontally printed samples tested better than their vertical counterparts. Ferretti et al. (2021) investigated the correlation between line width and volume of defects in an FDM-printed component. Kurose, Abe, Santos, Kanaya, Ishigami, Tanaka, and Ito (2020) studied the influence of layer directions on 316L stainless steel parts by observing the effect of binder domains in filaments. Lebedev, Gefle, Amitov, Zhuravlev, Berchuk, and Mikutskiy (2018) tested and discussed the mechanical properties of several PLA-based composites. They discovered that the elongation at break and yield strength of the 3D-printed PLA composites decreased by 15-60% compared to the hot-pressed samples. Liu, Zhang, Li, Si, Peng, and Hu (2017) studied several samples of FDM parts using the Taguchi method. They considered deposition orientation, layer thickness, deposition style, raster width, and raster gap as the main parameters and put them against three strength indices: tensile, flexural, and impact. Santamaria, Salasi, Bakhtiari, Leadbeater, Iannuzzi, and Quadir (2022) studied the impact of the sintering process on the microstructure and mechanical properties of the 3D-printed 316L stainless steel. Kuo, Wu, Li, and Wu (2019) studied the optimal process parameters to minimize the warpage of ABS prototypes. More specifically, they tested different temperatures at which the nozzle, bed, and chamber could optimally perform. Abouzaid, Bassir, Guessasma, and Yue (2021) discussed the observations of FDM and some optimal parameters to use when printing. They discussed the mechanical properties and how certain parameters of the printing job affect the final product, such as building orientation, raster angle, layer thickness, and printing temperature.

Li, Yu, Yang, and Shen (2020) utilized acoustic emission to detect distortion in FDM samples in real-time. Novakova-Marcincinova, Novak-Marcincin, Barna, and Torok (2012) explored all the uses for FDM in the real world, discussing different methods, materials, applications, and more. They focused on the process used to create material with metal particles and how they can be densified through de-binding and sintering. Hassan, Farid, Tosi, Rane, and Strano (2021) implemented tests on sintered 316L stainless steel parts to find the effects of three printing parameters: extrusion velocity, layer height, and orientation. de Oliveira Fiuza, Ribeiro da Rocha, Torres dos Santos, Rodrigues da Silva, Augusto Couto, Camilli Bottene, and Alves dos Santos (2021) explored the effects of parameters on FDM-printed parts such as temperature, deposition speed, layer height, and infill pattern. They found that temperature was a major quality factor. Excess temperature creates excess material through fluidity, and insufficient temperature causes a lack of adhesion and lower flowability. The objective of this current study was to investigate the impact of the printing direction and raster angle on the part density and the mechanical properties of 316L stainless steel 3D-printed parts using FDM/ FFF technology.

Materials and Methods: Parts Fabrication

In this current study, the parts were made using the FFF/ FDM process along with a MakerBot Method X 3D printer. The process started by creating a CAD model using computer-aided design (CAD) software and then converting this file into an .stl format that is compatible with these types of printers. A filament comprised of a blend of 80% 316L stainless steel powder and 20% resin was used. This filament had a diameter of 1.75 mm. The authors fabricated and tested the tensile specimens in accordance with ASTM E8 guidelines and bending specimens in compliance with ASTM E290 standards. Using ADMET, model 1032FGR-50K-B equipment of 50 Klbf, tensile and bending tests were performed. The building plate was prepared by scraping any residues from previous prints and then adding Magigoo Pro Metal 3D Bed Adhesion Solution for BASF Ultrafuse 316L. Figure 1 shows that this printer had two extruders; one for building material (316L stainless steel) and the other for support material. The authors then operated a single lab extruder, primarily intended for handling metals. For the fabrication of tensile and bending samples, Figure 2a shows that they were printed in two orientations: horizontal and on their edges. The printing parameters included the following: layer thickness set at 0.2 mm and 0.1 mm, extrusion temperature at 245°C, chamber temperature maintained at 85°C, infill percentage of 100% using a linear pattern, and a travel speed of 250 mm/s. To investigate how raster angles at 0°, 30°, and 45° affect the mechanical properties and density of the printed and sintered samples, the authors employed these various raster angles. After the printing process, the parts were de-bound and sintered to test their mechanical properties.

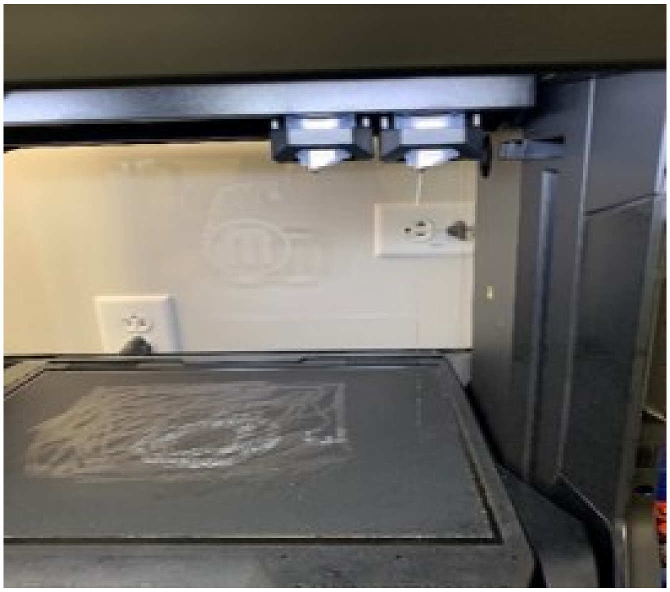
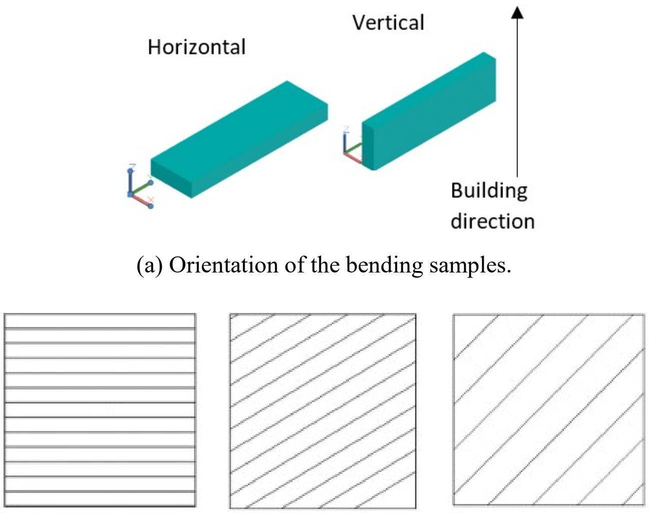


Figure 1. MakerBot Method X 3D printer.

Materials and Methods: De-Binding and Sintering Process

The samples were sent to DSH Technologies, LLC, for the thermal de-binding and sintering process. The dimensions and the densities were measured for the green parts and the sintered parts. The authors then performed tensile and bending testing on the sintered samples. Figure 3 shows some of the broken samples (after tensile testing), as well as how three samples—at different printing parameters—were printed and sintered for each geometry. In the tensile test, samples printed at 0.2 mm layer thickness and 0.1 mm layer thickness were tested. At each of those layer thicknesses, 12 samples were printed, six of which were printed horizontally with the other six printed vertically. Of the six samples, two were printed at each of the raster angles: 0°, 30°, and 45°. Figure 3 further shows how the dimensions and the densities were then measured for the green and sintered parts for each geometry. The dimensions were measured using a regular caliper, while the density was measured using the Archimedes method. A Torbal AGCN220 Internal Calibration Analytical Balance, 220g x 0.0001g, was used in the process in addition to a Torbal Density Kit for AGCN scales.



(a) Orientation of the bending samples.

(b) Raster angles (0°, 30°, and 45°).

Figure 2. Fabrication of tensile and bending samples, printed horizontally and on edge.

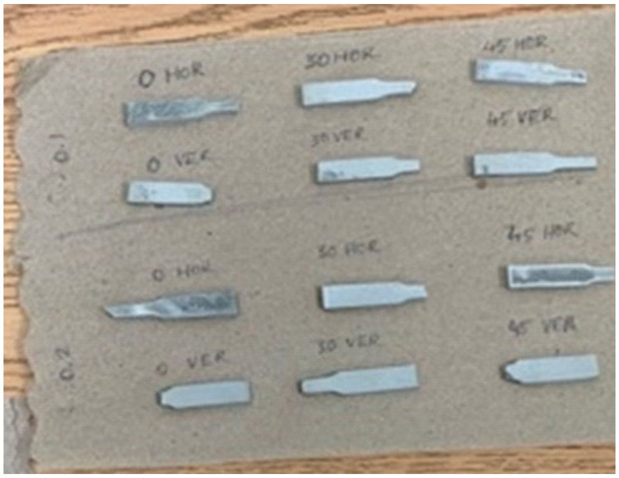


Figure 3. 316L stainless steel tensile testing samples after failure.

Results and Discussion

As mentioned previously, density was measured using the Archimedes method, while apparent density was calculated using Equation 1. Figure 4 shows the apparent density of the green parts printed horizontally and vertically (on edge) at raster angles of 0°, 30°, and 45°, and at layer thicknesses of 0.2 mm and 0.1 mm.

$$\rho_a = \rho_{wt}\frac{m_d}{m_d - m_w} \tag{1}$$

ρ_a = apparent density of the part
ρ_{wt} = water density at the measurement temperature
m_d = dry mass of the part
m_w = wet mass of the part

The results showed that there was a density difference for the green parts between all the samples, especially at a raster angle of 45° when the samples were printed at a layer thickness of 0.2 mm. Tables 1a and 1b show the green parts' density change percentages for vertical and horizontal printing at different raster angles. It was found that the range of that difference in density was between 2.6% at 0° (at a layer thickness of 0.1 mm) and -25% at 45° (at a layer thickness of 0.2 mm), when compared between vertical and horizontal printing at different layer thicknesses. To measure the density of the green samples provided in Tables 1a and 1b, six samples were printed at the 0.2 mm layer thickness and another six at the 0.1 mm layer thickness.

One sample was printed vertically and the other was printed horizontally at each layer thickness for raster angles of 0°, 30°, and 45°. The same number of samples were used at a layer thickness of 0.1 mm. Figure 4 shows the samples used to calculate the apparent density of the green parts. The same procedure was performed for the sintered samples in Tables 2a and 2b. Figure 5 shows the samples used to calculate the apparent density of the sintered samples. The authors compared the twelve sintered samples with twelve green samples for density measurement purposes.

The 95% confidence interval error bars of Figure 4 show no or low overlap between the calculated apparent density values reported for the green samples. Figure 5 shows that the apparent density of the sintered parts was affected by the raster angle. When comparing horizontal printing with vertical printing (assuming the horizontal printing was the base for comparison), the highest density difference (3.7% at a layer thickness of 0.2 mm) occurred when the raster angle was 0°. On the other hand, the highest density difference was seen at 6.18% with a raster angle of 30° when the layer thickness was 0.1 mm. Figure 5, along with Tables 2a and 2b further show that printing at raster angles of 30° and 45° vertically and horizontally impacted the apparent density of the sintered parts.

Table 1a. Density changes of the green parts with a layer thickness of 0.2 mm.

Density (g/mL) at 0.2 mm layer thickness			
Raster angle (°)	Horizontal	Vertical	Density change (%)
0	4.73	4.86	2.78
30	4.58	4.79	4.56
45	4.49	3.36	-25.15

Table 1b. Density changes of the green parts with a layer thickness of 0.1 mm.

Density (g/mL) at 0.1 mm layer thickness			
Raster angle (°)	Horizontal	Vertical	Density change (%)
0	4.62	4.74	2.62
30	4.486	4.70	4.88
45	4.53	4.51	-0.53

Table 2a. Density changes of the sintered parts with a layer thickness of 0.2 mm.

Density (g/mL) at 0.2 mm layer thickness			
Raster angle (°)	Horizontal	Vertical	Density change (%)
0	7.44	7.16	-3.74
30	7.38	7.49	1.49
45	7.26	7.39	1.80

Table 2b. Density changes of the sintered parts with a layer thickness of 0.1 mm.

Density (g/mL) at 0.1 mm layer thickness			
Raster angle (°)	Horizontal	Vertical	Density change (%)
0	7.59	7.47	-1.67
30	7.55	7.08	-6.18
45	7.33	7.21	-1.62

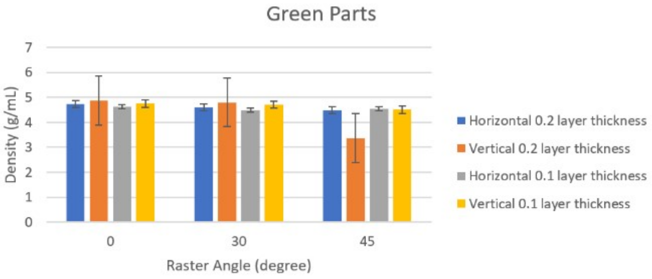


Figure 4. Apparent density of the green parts.

Figure 6 shows the tensile strength of the sintered samples printed horizontally and vertically at a layer thickness of 0.1 mm. A significant impact of the building direction (vertical or horizontal) on the tensile strength was also observed. In all of the cases, printing horizontally resulted in a higher tensile strength than printing vertically at all raster angles: 0° (17% - 27%), 30° (20% - 25%), and 45° (around 20%). This is because, when printed horizontally, the surface area at each layer was bigger, which would improve the packing of the samples on top of each other because of the high cooling rate. Equation 2 explains that by increasing the surface area, the cooling rate increases.

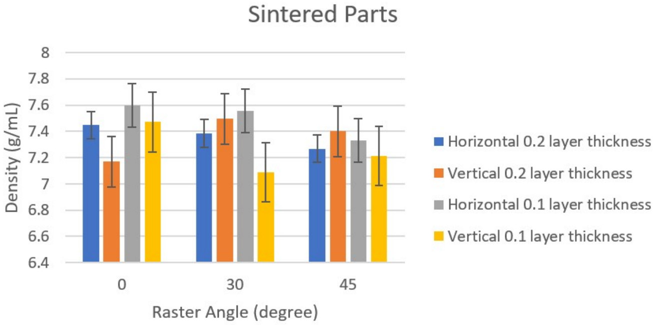


Figure 5. Apparent density of the sintered parts.

$$Q = hA / \Delta T \tag{2}$$

where, Q is the rate of heat transfer (in watts); h is the convective heat transfer coefficient $[W/(m^2 \cdot °C)]$; A is the surface area through which heat is being transferred (m^2); and, ΔT is the temperature difference between the object's surface and the surrounding air (°C).

The samples printed horizontally had a high surface area from the top (printing direction) compared to the samples printed on edge. This explains a high cooling rate when printed horizontally. Figure 6 further shows that there was an impact of the raster angle on the tensile strength of the material. The percentage decrease in the tensile strength was between 2.47% and 4.75%, when printed horizontally compared to printing at 0°. Tensile strength would increase by 4.3% at 45°, when printed vertically at a layer thickness of 0.1 mm. Figure 7 shows the impact of the building direction and raster angle on the modulus of elasticity, when printed at a layer thickness of 0.1 mm.

In all of the cases except one (at 30°), printing horizontally resulted in a higher modulus of elasticity than printing vertically at 0° (9% - 29%), 30° (-6% - 5.8%), and 45° (8% - 24 %). The results show that the modulus of elasticity increased when the samples were printed horizontally; thus, to get the highest value of the modulus of elasticity, it is preferably to print the samples horizontally at 45 to get the highest value of the modulus of elasticity. The value obtained was about 4.50 GPa, while the highest value obtained if printed vertically at 30 to get the highest value of the modulus of elasticity was about 4.35 GPa. All the values of the modulus of elasticity obtained were higher if printed horizontally at different angles except for one (at 30°). If the parts were printed horizontally, there was a smaller number of layers, which led to a lower probability of layers slipping, compared with the case if printed on the edge with a higher number of layers. That led to higher tensile strength, as explained in Figure 6, and higher modulus of elasticity, as explained in Figure 7. Figure 8 shows that printing horizontally resulted in higher tensile strength for all raster angles (up to 25% improvement at 0°, up to 46% at 30°, and up to 25% at 45°). Figure 9 shows that printing horizontally at 0° changed the modulus of elasticity by +19% at 0°, (-13% to 55% at 30°), and (2% to -42% at 45°).

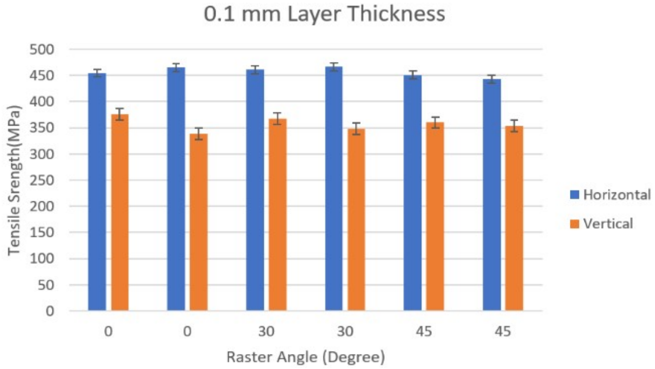


Figure 6. Tensile strength at different raster angles printed horizontally and vertically at a layer thickness of 0.1 mm.

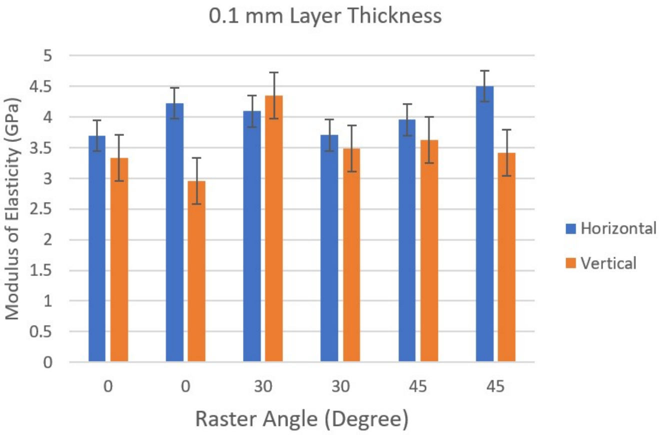


Figure 7. Modulus of elasticity at different raster angles printed horizontally and vertically at a layer thickness of 0.1 mm.

Figure 10 shows the flexural strength at different raster angles printed horizontally and vertically at a layer thickness of 0.2 mm. Printing vertically gives higher flexural strength and shows a significant difference compared to printing horizontally. The range of improvement was from 16% at 0° to 76% at 45°. Printing vertically on edge means more layers, which means when applying the bending test more interlaminar resistance to get the layers apart. That is why the flexural strength was higher in that direction.

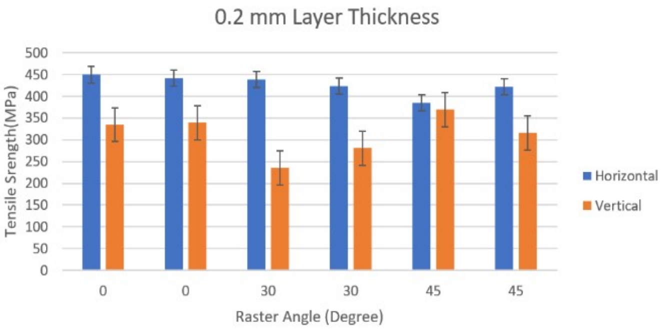


Figure 8. Tensile strength at different raster angles printed horizontally and vertically at a layer thickness of 0.2 mm.

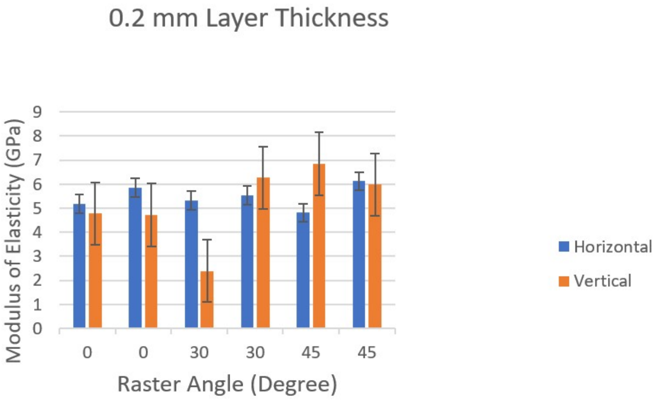


Figure 9. Modulus of elasticity at different raster angles printed horizontally and vertically at a layer thickness of 0.2 mm.

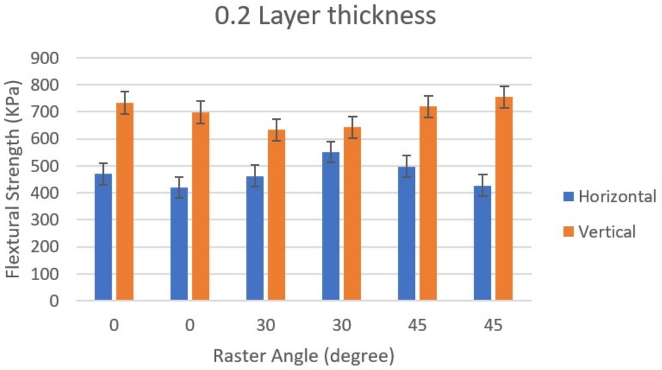


Figure 10. Flexural strength at different raster angles printed horizontally and vertically at a layer thickness of 0.2 mm.

Conclusions

In this study, the researchers studied the effect of building direction on the apparent and relative density and the mechanical properties of 3D-printed parts using FDM technology at different raster angles. Tensile strength, modulus of elasticity, and flexural strength were studied. The results showed that there was a significant effect of the printing direction (horizontal or vertical) and the raster angle on the apparent density of the sintered parts, while there was no significant effect on the apparent density of the green samples. In all cases, printing horizontally resulted in a higher tensile strength than printing vertically at all raster angles: 0° (17% - 27%), 30° (20% - 25%), and 45° (around 20%). The results showed that printing horizontally improved the modulus of elasticity of the material at different raster angles, while printing vertically resulted in higher flexural strength.

Future Research

The authors are in the process of micro-machining the same samples printed in this current study to evaluate the impact of raster angle and building direction on the roughness of the machined 316L stainless steel printed using FDM technology.

References

Abouzaid, K., Bassir, D., Guessasma, S., & Yue, H. (2021). Modeling the Process of Fused Deposition Modelling and the Effect of Temperature on the Mechanical, Roughness, and Porosity Properties of Resulting Composite Products. *Mechanics of Composite Materials*, *56* (6), 805-816.

de Oliveira Fiuza, D., Ribeiro da Rocha, J., Torres dos Santos, V., Rodrigues da Silva, M., Augusto Couto, A., Camilli Bottene, A., & Alves dos Santos, G. (2021). The influence of parameters and strategies on parts manufactured by FDM process. *Proceedings of the International Conference on Industrial Engineering & Operations Management*, 689-698. doi:10.46254/sa02.20210375

Farashi, S., & Vafaee, F. (2022). Effect of extruder temperature and printing speed on the tensile strength of fused deposition modeling (FDM) 3D printed samples: a meta-analysis study. *International Journal on Interactive Design and Manufacturing (IJIDeM)*, *16*(1), 305-316.

Ferretti, P., Leon-Cardenas, C., Ciotti, E., Santi, G. M., Donnici, G., & Frizziero, L. (2021). FDM 3D Printing Parameters Optimization: The Key Role of Line Width. *Proceedings of the International Conference on Industrial Engineering & Operations Management*, Bangalore, India, 921-929.

Hassan, W., Farid, M. A., Tosi, A., Rane, K., & Strano, M. (2021). The effect of printing parameters on sintered properties of extrusion-based additively manufactured stainless steel 316L parts. *The International Journal of Advanced Manufacturing Technology*, *114*(9-10), 3057-3067.

Kedziora, S., Decker, T., Museyibov, E., Morbach, J., Hohmann, S., Huwer, A., & Wahl, M. (2022). Strength Properties of 316L and 17-4 PH Stainless Steel Produced with Additive Manufacturing. *Materials (1996-1944)*, *15*(18), 6278-N.PAG.

Kuo, C.-C., Wu, Y.-R., Li, M.-H., & Wu, H.-W. (2019). Minimizing warpage of ABS prototypes built with low-cost fused deposition modeling machine using developed closed-chamber and optimal process parameters. *International Journal of Advanced Manufacturing Technology*, *101*(1/4), 593-602.

Kurose, T., Abe, A., Santos, M.V.A., Kanaya, Y., Ishigami, A., Tanaka, S., & Ito, H. (2020). Influence of the Layer Directions on the Properties of 316L Stainless Steel Parts Fabricated through Fused Deposition of Metals. *Materials*, *13*(11), 2493. https://doi.org/10.3390/ma13112493

Lebedev, S., Gefle, O., Amitov, E., Zhuravlev, D., Berchuk, D., & Mikutskiy, E. (2018). Mechanical properties of PLA-based composites for fused deposition modeling technology. *International Journal of Advanced Manufacturing Technology*, *97*(1/4), 511-518.

Li, F., Yu, Z., Yang, Z., & Shen, X. 2020. Real-Time Distortion Monitoring during Fused Deposition Modeling via Acoustic Emission. *Structural Health Monitoring—An International Journal, 19*(2), 412-23. doi:10.1177/1475921719849700.

Liu, B., Wang, Y., Lin, Z., & Zhang, T. (2020). Creating metal parts by Fused Deposition Modeling and Sintering. *Materials Letters, 263*, 127252.

Liu, X., Zhang, M., Li, S., Si, L., Peng, J., & Hu, Y. (2017). Mechanical property parametric appraisal of fused deposition modeling parts based on the gray Taguchi method. *International Journal of Advanced Manufacturing Technology*, *89*(5/8), 2387-2397.

Novakova-Marcincinova, L., Novak-Marcincin, J., Barna, J., & Torok, J. (2012). Special materials used in FDM rapid prototyping technology application. *Proceedings of the 2012 IEEE 16th International Conference on Intelligent Engineering Systems (INES),* 73-76.

Obeidat, S., Nervis, J. Jr., & Ma, J. (2022). The Impact of the Printed Part Geometry on the Shrinkage, and the Density of 316L Stainless Steel Parts Printed by FFF/FDM Technology. *Proceedings of the 33rd Annual International Solid Freeform Fabrication Symposium – An Additive Manufacturing Conference*, Austin, Texas, 1870-1877.

Ortega Varela de Seijas, M., Bardenhagen, A., Rohr, T., & Stoll, E. (2023). Indirect Induction Sintering of Metal Parts Produced through Material Extrusion Additive Manufacturing. *Materials (1996-1944)*, *16*(2), 885.

Quarto, M., Carminati, M., & D'Urso, G. (2021). Density and shrinkage evaluation of AISI 316L parts printed via FDM process. *Materials and Manufacturing Processes*, *36*(13), 1535-1543.

Ryder, M. A., Lados, D. A., Iannacchione, G. S., & Peterson, A. M. (2018). Fabrication and properties of novel polymer-metal composites using fused deposition modeling. *Composites Science and Technology*, *158*, 43 -50.

Santamaria, R., Salasi, M., Bakhtiari, S., Leadbeater, G., Iannuzzi, M., & Quadir, M. Z. (2022). Microstructure and mechanical behavior of 316L stainless steel produced using sinter-based extrusion additive manufacturing. *Journal of Materials Science*, *57*(21), 9646-9662.

Spiller, S., Kolstad, S. O., & Razavi, S. M. J. (2022). Fabrication and characterization of 316L stainless steel components printed with material extrusion additive manufacturing. *Procedia Structural Integrity*, *42*, 1239-1248.

Tosto, C., Tirillò, J., Sarasini, F., Sergi, C., & Cicala, G. (2022). Fused Deposition Modeling Parameter Optimization for Cost-Effective Metal Part Printing. *Polymers (20734360)*, *14*(16), 3264.

Biographies

SULEIMAN OBEIDAT is an assistant professor at Sam Houston State University. He received his PhD in Industrial Engineering from the University of Oklahoma. Dr. Obeidat's research interests include additive manufacturing and inspection of machined surfaces using coordinate machines (CMM) in addition to engineering education. Dr. Obeidat may be reached at obeidat@shsu.edu

IFTEKHAR BASITH is an associate professor at Sam Houston State University. He received his PhD in Electrical and Computer Engineering from the University of Windsor, ON, Canada. Dr. Basith's research interests include automation, and robotics in addition to engineering education. Dr. Basith may be reached at iib002@shsu.edu

ULAN DAKEEV is serving as an Associate Professor of Engineering Technology in the College of Science and Engineering Technology at Sam Houston State University. He served as an assistant professor at Texas A&M University – Kingsville, a lecturer at the University of Michigan – Flint, and a design engineer at John Deere Waterloo. Dr. Dakeev holds a BS in Industrial Engineering from the International Black Sea University, an MS in Industrial Management, and a doctorate in technology from the University of Northern Iowa. He also holds an engineering design certification from MIT. He is an active member of ATMAE, IAJC, ASEE, and Epsilon Pi Tau National Engineering Honor Society. Dr. Dakeev is a certified NCCER instructor for 21 trade skills. His research interests include virtual and augmented reality, motivation, and engagement of students and employees. Dr. Dakeev may be reached at dakeev@shsu.edu

Instructions for Authors: Manuscript Formatting Requirements

The INTERNATIONAL JOURNAL OF MODERN ENGINEERING is an online/print publication designed for Engineering, Engineering Technology, and Industrial Technology professionals. All submissions to this journal, submission of manuscripts, peer-reviews of submitted documents, requested editing changes, notification of acceptance or rejection, and final publication of accepted manuscripts will be handled electronically. The only exception is the submission of separate high-quality image files that are too large to send electronically.

All manuscript submissions must be prepared in Microsoft Word (.doc or .docx) and contain all figures, images and/or pictures embedded where you want them and appropriately captioned. Also included here is a summary of the formatting instructions. You should, however, review the sample Word document on our website (http://ijme.us/formatting_guidelines/) for details on how to correctly format your manuscript. The editorial staff reserves the right to edit and reformat any submitted document in order to meet publication standards of the journal.

The references included in the References section of your manuscript must follow APA-formatting guidelines. In order to help you, the sample Word document also includes numerous examples of how to format a variety of scenarios. Keep in mind that an incorrectly formatted manuscript will be returned to you, a delay that may cause it (if accepted) to be moved to a subsequent issue of the journal.

1. **Word Document Page Setup:** Two columns with ¼" spacing between columns; top of page = ¾"; bottom of page = 1" (from the top of the footer to bottom of page); left margin = ¾"; right margin = ¾".

2. **Paper Title:** Centered at the top of the first page with a 22-point Times New Roman (Bold), small-caps font.

3. **Page Breaks:** Do not use page breaks.

4. **Figures, Tables, and Equations:** All figures, tables, and equations must be placed immediately after the first paragraph in which they are introduced. And, each must be introduced. For example: "Figure 1 shows the operation of supercapacitors." "The speed of light can be determined using Equation 4:"
5. **More on Tables and Figures:** Center table captions above each table; center figure captions below each figure. Use 9-point Times New Roman (TNR) font. Italicize the words for table and figure, as well as their respective numbers; the remaining information in the caption is not italicized and followed by a period—e.g., "*Table 1*. Number of research universities in the state." or "*Figure 5*. Cross-sectional aerial map of the forested area."

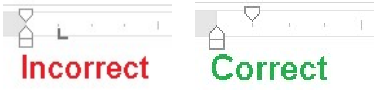
6. **Figures with Multiple Images:** If any given figure includes multiple images, do NOT group them; they must be placed individually and have individual minor captions using, "(a)" "(b)" etc. Again, use 9-point TNR.

7. **Equations:** Each equation must be numbered, placed in numerical order within the document, and introduced—as noted in item #4.

8. **Tables, Graphs, and Flowcharts:** All tables, graphs, and flowcharts must be created directly in Word; tables must be enclosed on all sides. The use of color and/or highlighting is acceptable and encouraged, if it provides clarity for the reader.

9. **Textboxes:** Do not use text boxes anywhere in the document. For example, table/figure captions must be regular text and not attached in any way to their tables or images.

10. **Body Fonts:** Use 10-point TNR for body text throughout (1/8" paragraph indention); indent all new paragraphs as per the images shown below; do not use tabs anywhere in the document; 9-point TNR for author names/affiliations under the paper title; 16-point TNR for major section titles; 14-point TNR for minor section titles.



11. **Personal Pronouns:** Do not use personal pronouns (e.g., "we" "our" etc.).

12. **Section Numbering:** Do not use section numbering of any kind.

13. **Headers and Footers:** Do not use either.

14. **References in the Abstract:** Do NOT include any references in the Abstract.

15. **In-Text Referencing:** For the first occurrence of a given reference, list all authors—last names only—up to seven (7); if more than seven, use “et al.” after the seventh author. For a second citation of the same reference—assuming that it has three or more authors—add “et al.” after the third author. Again, see the *sample Word document* and the *formatting guide for references* for specifics.

16. **More on In-Text References:** If you include a reference on any table, figure, or equation that was not created or originally published by one or more authors on your manuscript, you may not republish it without the expressed, written consent of the publishing author(s). The same holds true for name-brand products.

17. **End-of-Document References Section:** List all references in alphabetical order using the last name of the first author—last name first, followed by a comma and the author’s initials. Do not use retrieval dates for websites.

18. **Author Biographies:** Include biographies and current email addresses for each author at the end of the document.

19. **Page Limit:** Manuscripts should not be more than 15 pages (single-spaced, 2-column format, 10-point TNR font).

20. **Page Numbering:** Do not use page numbers.

21. **Publication Charges:** Manuscripts accepted for publication are subject to mandatory publication charges.

22. **Copyright Agreement:** A copyright transfer agreement form must be signed by all authors on a given manuscript and submitted by the corresponding author before that manuscript will be published. Two versions of the form will be sent with your manuscript’s acceptance email.

    Only one form is required. Do not submit both forms!

    The form named “paper” must be hand-signed by each author. The other form, “electronic,” does not require hand signatures and may be filled out by the corresponding author, as long as he/she receives written permission from all authors to have him/her sign on their behalf.

23. **Submissions:** All manuscripts and required files and forms must be submitted electronically to Dr. Philip D. Weinsier, manuscript editor, at philipw@bgsu.edu.

24. **Published Deadlines:** Manuscripts may be submitted at any time during the year, irrespective of published deadlines, and the editor will automatically have your manuscript reviewed for the next-available issue of the journal. Published deadlines are intended as “target” dates for submitting new manuscripts as well as revised documents. Assuming that all other submission conditions have been met, and that there is space available in the associated issue, your manuscript will be published in that issue if the submission process—including payment of publication fees—has been completed by the posted deadline for that issue.

    Missing a deadline generally only means that your manuscript may be held for a subsequent issue of the journal. However, conditions exist under which a given manuscript may be rejected. Always check with the editor to be sure. Also, if you do not complete the submission process (including all required revisions) within 12 months of the original submission of your manuscript, your manuscript may be rejected or it may have to begin the entire review process anew.

CUTTING EDGE JOURNAL OF RESEARCH AND INNOVATION IN ENGINEERING

www.ijeri.org

Print ISSN: 2152-4157
Online ISSN: 2152-4165

www.iajc.org

INTERNATIONAL JOURNAL OF ENGINEERING RESEARCH AND INNOVATION

ABOUT IJERI:

- IJERI is the second official journal of the International Association of Journals and Conferences (IAJC).
- IJERI is a high-quality, independent journal steered by a distinguished board of directors and supported by an international review board representing many well-known universities, colleges, and corporations in the U.S. and abroad.
- IJERI has an impact factor of **1.58**, placing it among an elite group of most-cited engineering journals worldwide.

OTHER IAJC JOURNALS:

- The International Journal of Modern Engineering (IJME)
  For more information visit www.ijme.us
- The Technology Interface International Journal (TIIJ)
  For more information visit www.tiij.org

IJERI SUBMISSIONS:

- Manuscripts should be sent electronically to the manuscript editor, Dr. Philip Weinsier, at philipw@bgsu.edu.

  For submission guidelines visit www.ijeri.org/submissions

TO JOIN THE REVIEW BOARD:

- Contact the chair of the International Review Board, Dr. Philip Weinsier, at philipw@bgsu.edu.

  For more information visit www.ijeri.org/editorial

INDEXING ORGANIZATIONS:

- IJERI is currently indexed by 16 agencies. For a complete listing, please visit us at www.ijeri.org.

Contact us:

Mark Rajai, Ph.D.

Editor-in-Chief
California State University-Northridge
College of Engineering and Computer Science
Room: JD 4510
Northridge, CA 91330
Office: (818) 677-5003
Email: mrajai@csun.edu

www.tiij.org

www.ijme.us

THE LEADING JOURNAL OF ENGINEERING, APPLIED SCIENCE AND TECHNOLOGY

The latest impact factor (IF) calculation (Google Scholar method) for IJME of 3.0 moves it even higher in its march towards the top 10 engineering journals.

IJME IS THE OFFICAL AND FLAGSHIP JOURNAL OF THE INTERNATIONAL ASSOCATION OF JOURNALS AND CONFERENCE (IAJC)
www.iajc.org

The International Journal of Modern Engineering (IJME) is a highly-selective, peer-reviewed journal covering topics that appeal to a broad readership of various branches of engineering and related technologies. IJME is steered by the IAJC distinguished board of directors and is supported by an international review board consisting of prominent individuals representing many well-known universities, colleges, and corporations in the United States and abroad.

IJME Contact Information

General questions or inquiries about sponsorship of the journal should be directed to:

Mark Rajai, Ph.D.
Editor-in-Chief
Office: (818) 677-5003
Email: editor@ijme.us
Department of Manufacturing Systems Engineering & Management
California State University-Northridge
18111 Nordhoff St.
Northridge, CA 91330