**PROJECT REPORT**

**ON**

***ATOMIC CLOCK OSCILLATION CORRECTION USING ANN***

**Submitted By:**

**AYUSH NEGI**

**(BTECH CSE,IPEC)**

**Under the guidance of:**

**Mr. PANKAJ KASHYAP**

**(Scientist)**



**Acknowledgement**

**We would like to express our sincere gratitude to our supervisor “Mr. PANKAJ KASHYAP”, “Scientist”, Solid State Physics Laboratory (DRDO), for giving us the opportunity to work on this topic. It would never be possible for us to take this project to this level without their innovative ideas and their relentless support and encouragement.**

**AYUSHNEGI**

**Declaration**

**We hereby declare that the project report titled “Atomic Clock Oscillation Correction Using ANN”, submitted by Saksham, Anisha, and Ayush Negi, is a genuine record of our original work. The work has been carried out under the guidance and supervision of Mr. Pankaj Kashyap, Scientist, Solid State Physics Laboratory (DRDO). We confirm that this report has not been submitted or presented elsewhere for any other purpose.**

**AYUSHNEGI**

**TABLE OFCONTENTS**

**Acknowledgement**

**i**

**Declaration**

**ii**

**Certificate**

**iii**

**Certificate from Industry/from Course**

**iv**

**Table of Contents**

**v**

**List of Figures**

**vi**

**List of Tables**

**vii**

**Abstract**

**viii**

**Chapter**

1. **Introduction**

**I. Company overview**

**II. Literature Review**

**II.1. Introduction**

**II.2. What are atomic clocks?**

**II.3.Significance ofAtomic Clocks**

**II.4.Goal**

**II.5.Survey**

**II.6.Why we used cesium-133 instead of hydrogen masers?**

**II.7.Relevant Studies and Frameworks**

**III. Design and Implementation**

**III.1. Introduction**

**III.2.Different Design Options**

**III.3.Functional Decomposition**

**III .4 Proposed Flow**

1. **Simulation Platform**
2. **Code**
3. **Output**
4. **Implementation Details**

**IV. Conclusion**

**IV.1. Summary of work done**

**IV.2.Learning Outcomes**

**IV.3.Challenges and Solutions**

**IV.4. Future Work**

**IV.5. Final Thoughts**

**References/Bibliography**

**LIST OF FIGURES**

**Figures**

**fig I.1: DRDO, Ministry of defence, Govt. of India logo**

**fig II.1:Atomic clock**

**fig II.2: Diff. between cesium-133 and hydrogen maser**

**fig II.3: Feedforward ANN**

**fig III.1: Jupyter notebook**

**fig III.2: Code**

**fig III.2: Code**

**fig III.3: Code**

**fig III.4: Code**

**fig III.5: Code**

**fig III.6: Output**

**fig III.7: Output**

**fig III.8: Output**

**ABSTRACT**

**This report presents the findings and experiences from our industrial training at Defence Research and Development organisation(DRDO), Ministry of defence, Govt. of India, conducted from 3rd July 2024 to 31st August 2024.**

**The primary objective of this training was to gain practical exposure and hands-on experience in Python development field, complementing our academic knowledge.**

**We engaged in project and tasks, utilizing python basics and advanced concepts (like TensorFlow and Matplotlib, etc). Our responsibilities included joining the live training sessions held throughout the time frame and completing assigned task.**

**The training provided valuable insights into market and live development, highlighting the importance of communication and knowledge of the market.**

**In conclusion, this industrial training has significantly enhanced our technical and professional skills, providing a robust foundation for our future career. The experience has not only broadened our understanding of development and service sector but also reinforced our commitment to pursuing a career in this domain.**

**CHAPTER I: INTRODUCTION**



**fig I.1: DRDO, Ministry of defence, Govt. of India logo**

**Defence Research and Development organisation(DRDO), Ministry of defence, Govt. of India is the R&D wing of Ministry of Defence, Govt of India, with a vision to empower India with cutting-edge defence technologies and a mission to achieve self-reliance in critical defence technologies and systems, while equipping our armed forces with state-of-the-art weapon systems and equipment in accordance with requirements laid down by the three Services. DRDO's pursuit of self-reliance and successful indigenous development and production of strategic systems and platforms such as Agni and Prithvi series of missiles; light combat aircraft, Tejas; multi-barrel rocket launcher, Pinaka; air defence system, Akash; a wide range of radars and electronic warfare systems; etc., have given quantum jump to India's military might, generating effective deterrence and providing crucial leverage.**

**"Balasya Mulam Vigyanam"—the source of strength is science-drives the nation in peace and war. DRDO has firm determination to make the nation strong and self-reliant in terms of science and technology, especially in the field of military technologies. DRDO was formed in 1958 from the amalgamation of the then already functioning Technical Development Establishment (TDEs) of the IndianArmy and the Directorate of Technical Development & Production (DTDP) with the Defence Science Organisation (DSO). DRDO was then a small organisation with 10 establishments or laboratories.**

**Over the years, it has grown multi-directionally in terms of the variety of subject disciplines, number of laboratories, achievements and stature.**

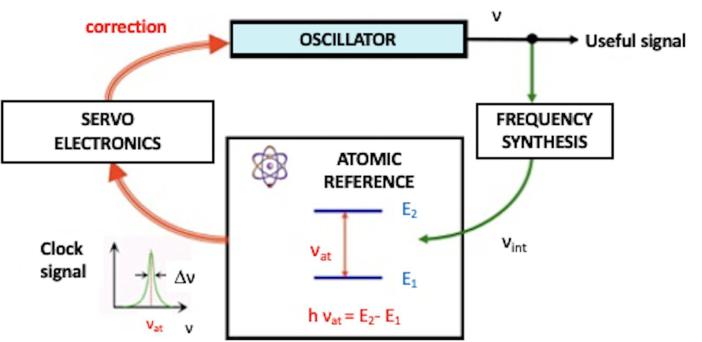
**Today, DRDO is a network of around 41 laboratories and 05 DRDO Young Scientist Laboratories (DYSLs) which are deeply engaged in developing defence technologies covering various disciplines, like aeronautics, armaments, electronics, combat vehicles, engineering systems, instrumentation, missiles, advanced computing and simulation, special materials, naval systems, life sciences, training, information systems and agriculture. Several major projects for the development of missiles, armaments, light combat aircrafts, radars, electronic warfare systems etc are on hand and significant achievements have already been made in several such technologies.**

**CHAPTER II: LITERATURE REVIEW**

**Introduction**

**This chapter provides an overview of the existing literature relevant to atomic clocks and artificial neural networks. It sets the context for the project by summarizing the current understanding and advancements in these areas.**

**What are atomic clocks?**



**fig II.1:Atomic clock**

**Atomic clocks are highly precise timekeeping devices that use the vibrations of atoms to measure time. Unlike traditional clocks that rely on mechanical movements or the oscillation of quartz crystals, atomic clocks use the consistent frequency of electromagnetic waves absorbed or emitted by atoms to keep accurate time.**

**Significance ofAtomic Clocks**

**Atomic clocks have revolutionized timekeeping, allowing for unprecedented precision and stability. They have enabled advances in technology, navigation(), and science, underpinning many systems that modern society depends on.**

**Goals and Objectives**

**Goals and Objectives:**

**The primary goal of this project is to correct atomic oscillations in simulated atomic clocks using artificial neural networks (ANNs). The objectives include enhancing the precision and stability of atomic clocks, which are critical for advanced technological applications such as navigation, communication, and scientific research.**

**Motivation:**

**The motivation behind this project is to address the inherent noise and inaccuracies in atomic clock oscillations. By leveraging ANN, the project aims to develop a method to improve the accuracy and reliability of atomic clocks, which is essential for precise timekeeping in various high-stakes applications.**

**Method:**

**The project was carried out by simulating atomic oscillations of cesium-133 and generating synthetic datasets with intentional noise. An ANN was then trained to distinguish between noisy and pure oscillations, learning the underlying patterns necessary for accurate correction.**

**Overview of the TechnicalArea**

**Atomic clocks are highly precise timekeeping devices that rely on the oscillations of atoms, typically cesium or rubidium. These oscillations are subject to various forms of noise and inaccuracies, which can affect the clock's precision. Artificial Neural Networks (ANNs) are a type of machine learning model capable of learning complex patterns and making predictions based on data. By applyingANNs to atomic clock oscillations, it is possible to improve their accuracy and reliability.**

**Survey**

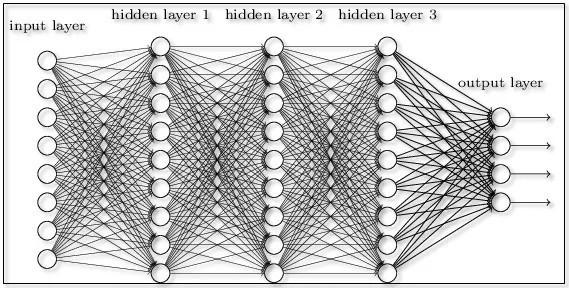
**A comprehensive survey of relevant research papers, articles, and previous projects was conducted. The survey included studies on the principles and mechanisms of atomic clocks, sources of noise and inaccuracies, and existing techniques for noise correction. Additionally, research on the application of ANNs in various fields of signal processing was reviewed to understand how similar methodologies could be adapted for atomic clock oscillation correction.**

**Why we used cesium-133 instead of hydrogen masers?**



**fig II.2: Diff. between cesium-133 and hydrogen maser**

**WhichAnn we are using and why?**



**fig II.3: FeedforwardANN**

**In the given code, we are using a feedforwardArtificial Neural Network (ANN) for the task of correcting noisy atomic oscillations. Below is an explanation of the ANN used and the rationale behind it:**

**ANNArchitecture:**

1. **Input Layer:**
   * **The input layer takes the noisy oscillation data. The shape of the input is determined by the number of data points in the noisy oscillation.**
2. **Hidden Layers:**
   * **First Hidden Layer:**
     + **Dense(128, activation='relu'):Adense layer with 128 neurons and ReLU activation function.**
   * **Second Hidden Layer:**
     + **Dense(128, activation='relu':Another dense layer with 128 neurons and ReLU activation function.**
   * **Third Hidden Layer:**
     + **Dense(128, activation='relu'): Yet another dense layer with 128 neurons and ReLU activation function.**
3. **Output Layer:**
   * **Dense(X\_train.shape[1]):Adense layer with the same number of neurons as the input data points. This ensures the output shape matches the input shape.**

**WhatAre Epochs?**

* **Epochs:An epoch is one complete pass through the entire training dataset. During each epoch, the model processes all the training samples and adjusts the weights of the neural network based on the error of the predictions. Increasing the number of epochs allows the model to learn from the data more thoroughly, potentially leading to better performance. However, training for too many epochs can lead to overfitting, where the model performs well on the training data but poorly on unseen data.**

**Relevant Studies and Frameworks**

**1. NumPy**

**Description: NumPy is a fundamental package for scientific computing in Python. It provides support for large, multi-dimensional arrays and matrices, along with a collection of mathematical functions to operate on these arrays.**

**Usage in Code:**

* + **np.linspace: Generates linearly spaced values, used here to create a time vector t.**
  + **np.sin: Computes the sine of an array, used to generate the pure oscillation signal.**
  + **np.random.normal: Generates random values from a normal (Gaussian) distribution, used to add noise to the pure oscillation.**
  + **np.array: Converts lists into NumPy arrays for efficient computation and data manipulation.**

1. **TensorFlow**

**Description: TensorFlow is an open-source deep learning framework developed by the Google Brainteam.**

**It is widely used for building and training machine learning models.**

**Usage in Code:**

* + **tf.keras.models.Sequential: Alinear stack of layers used to build theANN model.**
  + **tf.keras.layers.Dense: Afully connected layer in the neural network. The code uses multiple dense layers with ReLU activation functions.**
  + **tf.keras.optimizers.Adam: An optimizer that implements theAdam algorithm, used for updating the model parameters during training.**
  + **tf.function: Adecorator that compiles a Python function into a TensorFlow graph for better performance.**

1. **Matplotlib**

**Description: Matplotlib is a plotting library for the Python programming language and its numerical mathematics extension NumPy. It provides an object-orientedAPI for embedding plots into applications.**

**Usage in Code:**

* + **plt.figure: Creates a new figure for plotting.**
  + **plt.plot: Plots data points. In this code, it is used to plot the pure oscillation, noisy oscillation, and corrected oscillation.**
  + **plt.legend: Adds a legend to the plot.**
  + **plt.show: Displays the plot.**

1. **time**

**Description: The time module provides various time-related functions. In this code, it is used to get the current time in nanoseconds since the epoch.**

**Usage in Code:**

* **time.perf\_counter\_ns: Returns the value (in nanoseconds) of a performance counter, used to obtain high-resolution timing.**

**Summary of Usage**

* **NumPy is used for numerical operations and data manipulation, such as generating oscillation signals and noise.**
* **TensorFlow is employed to build, train, and utilize an artificial neural network for correcting noisy oscillation signals.**
* **Matplotlib is used for visualizing the oscillation signals, comparing the pure, noisy, and corrected signals.**
* **time is used for high-resolution time measurements, simulating a real-time aspect of atomic clocks.**

**These frameworks and libraries together provide the tools necessary for simulating atomic oscillations, training a neural network to correct these oscillations, and visualizing the results to evaluate the performance of the correction.**

**CHAPTER III. DESIGNAND IMPLEMENTATION**

**Introduction**

**The design and implementation of the project aim to develop an Artificial Neural Network (ANN) model for correcting oscillations in simulated atomic clocks. The primary objective is to enhance the precision and stability of atomic clocks by filtering out noise from the oscillation signals. This chapter outlines the design choices, functional decomposition, and implementation details of the system.**

**Functional Decomposition**

**The system is divided into several functional components:**

1. **Data Simulation: Generates synthetic oscillation data with added noise.**
2. **Data Preparation: Prepares the training dataset by simulating noisy and pure oscillation signals.**
3. **Model Training: Develops and trains anANN model using the prepared dataset.**
4. **Oscillation Correction: Utilizes the trainedANN model to correct noisy oscillation signals.**
5. **Integration withAtomic Clock: Integrates theANN model to simulate real-time correction of atomic clock signals.**
6. **Visualization: Plots the pure, noisy, and corrected oscillation signals for comparison.**

**Different Design Options**

**Various design options were considered for each functional component:**

1. **Data Simulation: Different levels of noise and frequencies were tested to create robust training data.**
2. **ModelArchitecture: Several neural network architectures were explored, including varying the number of layers and neurons per layer.**
3. **Activation Functions: Different activation functions (e.g., ReLU, sigmoid, tanh) were evaluated for optimal performance.**
4. **OptimizationAlgorithms: Various optimization algorithms, such as SGD, RMSprop, andAdam, were compared.**

**Proposed Flow**

**The proposed flow for the system is as follows:**

1. **Data Simulation: Simulate atomic oscillation with added noise.**
2. **Data Preparation: Generate synthetic dataset for training theANN.**
3. **Model Development: Build and compile theANN model.**
4. **Model Training: Train theANN model using the synthetic dataset.**
5. **Oscillation Correction: Correct noisy oscillation signals using the trainedANN model.**
6. **Integration and Testing: Integrate theANN model with the atomic clock function and test its performance.**
7. **Visualization: Plot and compare the results of pure, noisy, and corrected oscillations.**

**Simulation Platform**

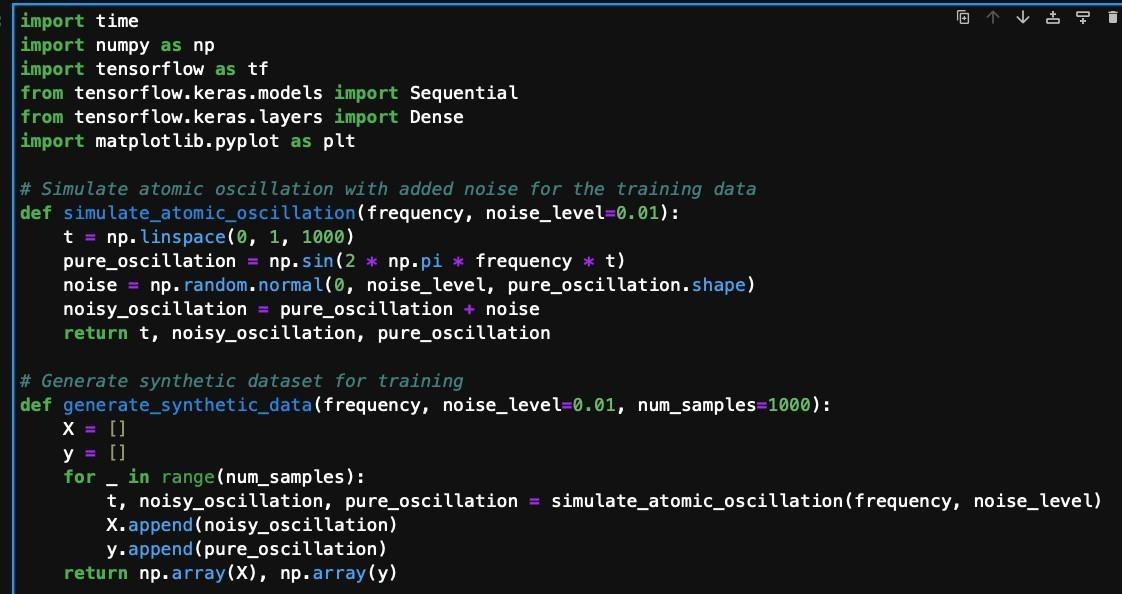
**The simulation platform used in this project includes:**



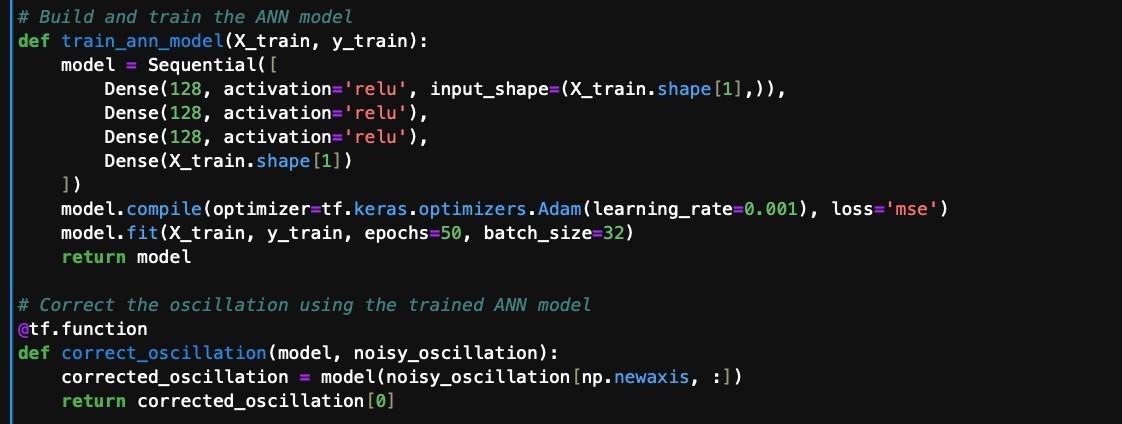
**fig III.1: Jupyter notebook**

**we have used the jupyter notebook to simulate the project.**

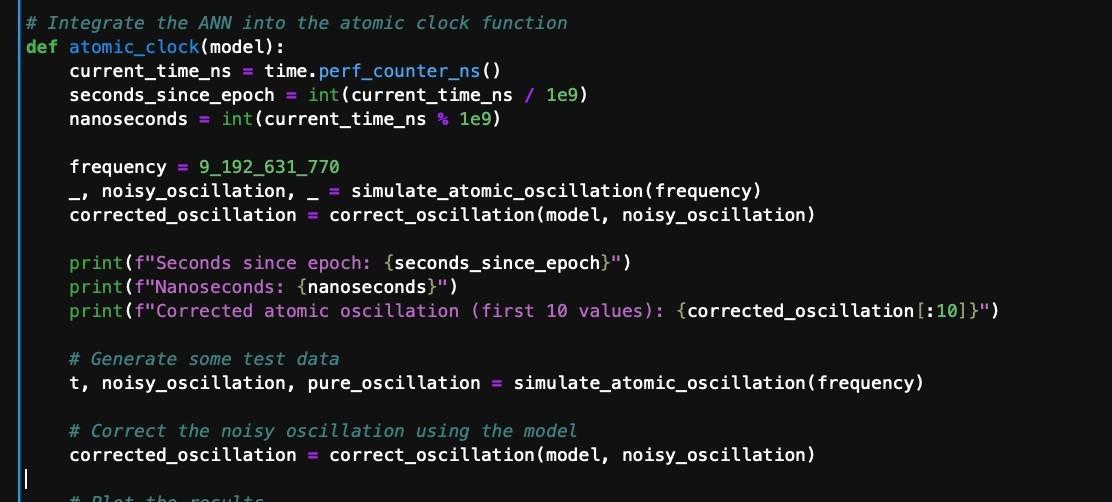
**Code:**



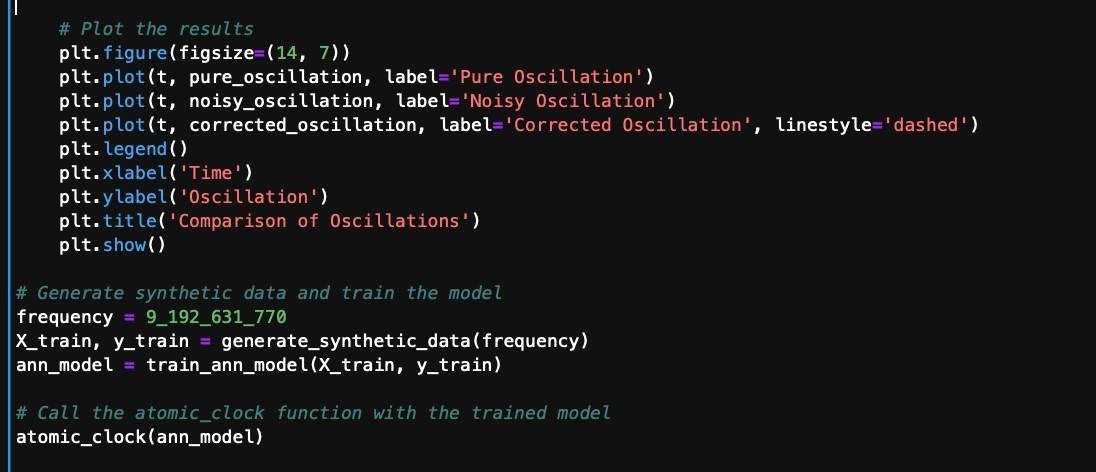
**fig III.2: Code**



**fig III.3: Code**

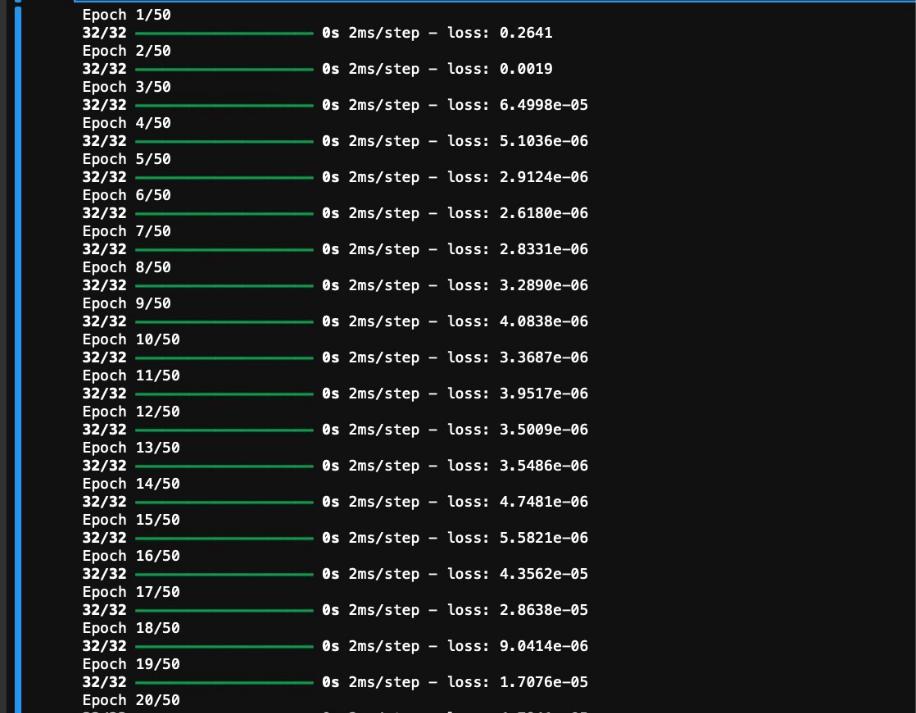


**fig III.4: Code**



**fig III.5: Code**

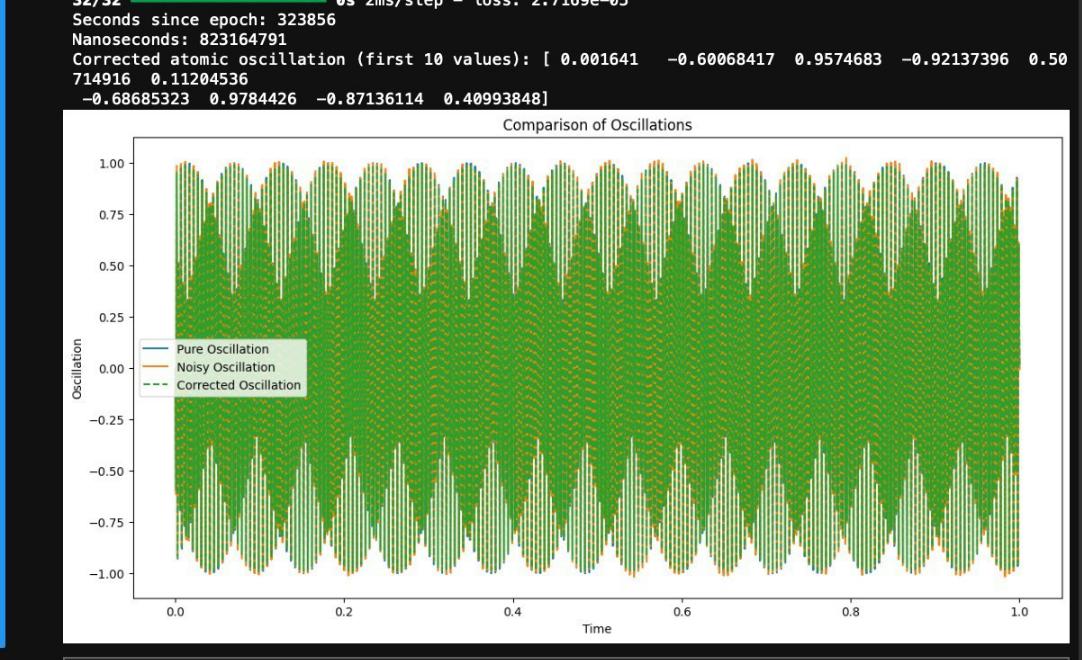
**Output:**



**fig III.6: Output**



**fig III.7: Output**



**fig III.8: Output**

* **Pure Oscillation:**
  + **The pure oscillation is a sine wave generated using a known frequency (9,192,631,770 Hz). This is the expected signal of the atomic clock without any interference or noise.**
  + **This signal is crucial as it provides a benchmark against which the noisy and corrected signals are compared.**
* **Noisy Oscillation:**
  + **The noisy oscillation is created by adding Gaussian noise to the pure oscillation signal. This noise simulates the random variations and disturbances that occur in real-world atomic clocks.**
  + **As seen in the plot, the noisy oscillation deviates from the pure oscillation, illustrating the impact of noise on the accuracy of the atomic clock.**
* **Corrected Oscillation:**
  + **The corrected oscillation is the output of theANN model, which has been trained to reduce noise from the noisy oscillation.**
  + **The green dashed line closely follows the blue pure oscillation line, indicating that the ANN model successfully denoised the signal and approximated the pure oscillation accurately.**
  + **This correction demonstrates the effectiveness of theANN in filtering out noise and**

**enhancing the precision of the atomic clock's oscillation signal.**

**Here's a flowchart representing the flow of the code for correcting noisy atomic oscillations using anArtificial**

**Neural Network (ANN):**

**START**

**|**

**v**

**Import Libraries**

**|**

**v**

**Simulate Atomic Oscillation**

**|**

**v**

**Generate Synthetic Data**

**|**

**v**

**Build and Train ANN Model**

**|**

**v**

**Correct Oscillation Using ANN Model**

**|**

**v**

**Integrate ANN into Atomic Clock Function**

**|**

**v**

**Generate Test Data and Plot Results**

**|**

**v**

**END**

**Implementation Details**

1. **Data Simulation and Preparation:**
   * **The simulate\_atomic\_oscillation function generates a time vector t, pure oscillation signal, and noisy oscillation signal by adding Gaussian noise.**
   * **The generate\_synthetic\_data function creates a synthetic dataset of noisy and pure oscillation signals for training the ANN model.**
2. **Model Development and Training:**
   * **TheANN model is built using the Sequential class from Keras, with three hidden layers of 128 neurons each and ReLU activation functions.**
   * **The model is compiled with theAdam optimizer and mean squared error (MSE) loss function.**
   * **The model is trained for 50 epochs with a batch size of 32.**
3. **Oscillation Correction:**
   * **The correct\_oscillation function uses the trainedANN model to correct noisy oscillation signals in real-time.**
4. **Integration and Testing:**
   * **The atomic\_clock function integrates theANN model to simulate real-time correction of atomic oscillations, displaying the corrected values and plotting the results.**
5. **Visualization:**
   * **The matplotlib.pyplot library is used to plot the pure, noisy, and corrected oscillation signals, providing a visual comparison of the model's performance**

**Conclusion:**

**The plot visually confirms the effectiveness of the ANN model in correcting noisy oscillations. By comparing the pure, noisy, and corrected oscillations, it is evident that the ANN model can effectively restore the pure signal from the noisy data. This capability is crucial for improving the performance and reliability of atomic clocks in practical applications.**

**CHAPTERIV: CONCLUSION**

**Summary of the Project**

**The project involves the use of anArtificial Neural Network (ANN) to correct the atomic oscillations of a simulated atomic clock. The oscillation is simulated using a sine wave with a frequency corresponding to cesium-133 atoms, which emit radiation at a precise frequency when transitioning between energy levels. The code generates synthetic data representing these oscillations with added noise, trains anANN to learn the relationship between noisy and pure oscillations, and uses the trained model to correct new noisy oscillations. The purpose of the project is to enhance the precision of atomic clocks by correcting the noise in the oscillations**

**Learning Outcomes**

1. **Understanding ofAtomic Oscillations: Gained knowledge about the principles of atomic oscillations and their role in timekeeping.**
2. **Machine LearningApplications: Learned how to apply machine learning, specificallyANNs, to signal processing tasks.**
3. **Model Training and Optimization: Acquired skills in training neural networks and optimizing their performance using techniques such as the ReLU activation function and theAdam optimizer.**
4. **Data Generation and Noise Simulation: Developed an understanding of how to generate synthetic data and simulate real-world conditions by adding noise.**

**Challenges and Solutions**

* **Challenge: Ensuring the neural network effectively learned to correct the noisy oscillations.**

**Solution: Used ReLU activation functions and the Adam optimizer to enhance the training process, providing faster convergence and better performance.**

* **Challenge: Balancing the complexity of the model to avoid overfitting. Solution: Implemented a three-layer ANN with 128 neurons each and optimized hyperparameters such as learning rate and batch size.**

**Future Work**

* **Incorporating More Complex Models: Future work could involve exploring more sophisticated neural network architectures or other machine learning techniques.**
* **Real-World DataApplication:Applying the model to real-world atomic clock data to test its**

**efficacy outside of simulated environments.**

* **Enhanced Error Correction: Developing advanced error correction techniques to further improve the precision of atomic clocks.**

**Final Thoughts**

**The project successfully demonstrated the application of ANNs in correcting noisy atomic oscillations, showcasing the potential of machine learning in enhancing the accuracy of atomic clocks. While the simulation was simplified, it provided valuable insights into the principles of atomic timekeeping and the role of machine learning in signal processing.**

**These key points encapsulate the essence and achievements of the project, as well as the knowledge gained and the potential directions for future research**

**REFERENCES**

1. **J. Nelson, “What is anAtomic Clock?” nasa.gov. http:// www.nasa.gov/feature/jpl/what-is-an-atomic-clock.**
2. **Sruthikeerthi Nandita, R., Maharana, S., Rajathilagam, B., Subramanya Ganesh, T. and Krishnamoorthy, S., 2020. An artificial neural network model for timescale atomic clock ensemble algorithm. MAPAN, 35, pp.547-554.**
3. **López, C., 2023.Artificial intelligence and advanced materials. Advanced Materials, 35(23), p.2208683.**
4. **Fangneng, L., Yifeng, L., Jiangning, X. and Miao, W., 2023.Advances in satellite atomic clock**

**technologies for the GNSS. Measurement Science and Technology, 35(1), p.015027.**

1. **Wang, W., Wang, Y., Yu, C., Xu, F. and Dou, X., 2021. Spaceborne atomic clock performance review of BDS-3 MEO satellites. Measurement, 175, p.109075.**
2. **Bandi, T.N., 2024.Acomprehensive overview of atomic clocks and their applications. Demo Journal, 1(1), pp.40-50.**
3. **https://stackoverflow.com/**
4. **https://www.tensorflow.org/api\_docs/python/tf/all\_symbols**
5. **http://geeksforgeeks.org**
6. **https://docs.python.org/3/tutorial/index.html**