Reservoir Computing for Cyber Physical Systems

1st Aayush Jannumahanti
FK32171
aayushj1@umbc.edu

***Abstract*—In this paper, I provide a novel method for integrating reservoir computing, a type of recurrent neural network, with Cyber Physical Systems (CPS). I look at the difficulty presented by the rising complexity of CPS, which makes it difficult for conventional control techniques to meet necessary flexibility and performance standards. I use reservoir computing to provide CPS with dynamic and reliable control techniques as a remedy.**

I use the well-known reservoir computing model Echo State Networks (ESNs) to forecast the time-series data included in CPS. A control mechanism that can adjust in real-time to changes in system behavior is built on this prediction capability.

I. Motivation

The fusion of computing, networking, and physical processes has resulted in an unprecedented increase in the complexity of Cyber-Physical Systems (CPS). Traditional control systems, which were created for situations that were less interconnected and dynamic, encountering substantial difficulties as a result of this complexity. More flexible, resilient, and effective control mechanisms are thus the need of the hour in order to keep up with the changing nature of CPS.

The potential of reservoir computing techniques, including Echo State Networks (ESN), Neural Circuit Policies (NCP), Liquid State Machines (LSM), Liquid Time Constants (LTC), and Closed-form Continuous-time (CfCt) networks, is examined in this research. Each of these approaches has distinctive qualities that make it appropriate for handling the high-dimensional, non-linear, and time-dependent complexity of CPS.

Through simulations in various CPS scenarios, I demonstrate that the proposed ESN-based control strategy significantly outperforms traditional methods in terms of adaptability and prediction accuracy. This study offers promising evidence for the viability of reservoir computing in CPS, opening up new avenues for future research in adaptive control strategies. Furthermore, the findings underscore the potential for enhancing system reliability and performance in rapidly evolving CPS environments.

Initially, I sought to use DeepMind's MuJoCo, a physics engine built for simulating large physical systems, to provide an appropriate setting for my CPS. I did, however, have significant challenges while trying to build a reliable data pipeline between the CPS and the MuJoCo simulation. This experience brought home the inherent difficulties of managing sophisticated CPS with conventional methods, particularly when it comes to handling and analyzing the data these systems produce.

II. Introduction

As CPS continue to expand in size and complexity, they are disrupting accepted notions of system control. The capabilities of conventional control techniques, such as Proportional-Integral-Derivative (PID) control, Model Predictive Control (MPC), Fuzzy Logic Control, State-Space Control, Adaptive Control, and Neural Network Control, are being strained. These approaches, while useful in some situations, frequently struggle to deal with the high-dimensional, non-linear, and time-dependent complexity of CPS. As a fundamental entity in sectors such as healthcare, manufacturing, transportation, and energy, CPS have grown to play a pivotal role in our society.

When I first embarked on this research journey, I relied on DeepMind's MuJoCo as a simulation environment for CPS, aiming to establish a reliable data pipeline for real-time CPS data. This endeavor, while ambitious, brought to light the substantial difficulties associated with creating an effective data conduit for CPS, underlining the limitations of traditional control techniques in the face of complex CPS dynamics.

Consequently, this paper ventures into the realm of reservoir computing techniques, such as Echo State Networks (ESNs), Neural Circuit Policies (NCP), Liquid State Machines (LSM), Liquid Time Constants (LTC), and Closed-form Continuous-time (CfCt) networks. These techniques offer unique strengths that make them particularly suited to the complex dynamics inherent in CPS. For example, ESNs, as reviewed in [1], have a broad range of designs and applications that make them highly adaptable to different system requirements. Meanwhile, the hierarchical framework of Deep-ESNs, as proposed in [2], allows for multiple projection-encoding that can increase the computational power and flexibility of the reservoir computing system.

Moreover, the book [3] provides a comprehensive overview of various reservoir computing techniques and their applications, shedding light on their potential to revolutionize how we approach CPS control.

In the following sections, I delve into the details of these reservoir computing techniques, explore their theoretical foundations and practical implementations, and present the results of our experiments with them. By doing so, I hope to elucidate their performance and potential as future CPS control strategies.

III. Methodology and Implementation

A. Cyber Physical Systems

Cyber-Physical Systems (CPS) are a development in computational systems that encompass a close integration of networking, computation, and physical processes. These systems orchestrate interactions that serve as the foundation of various crucial industries, including but not limited to manufacturing, healthcare, transportation, and energy, by obfuscating the distinction between the real and virtual worlds.

Central to the architecture of CPS are engineered systems, precisely built around computing and communication cores. They are composed of both hardware and software elements. The hardware, often referred to as 'physical systems', encompasses devices interacting with the real-world environment, such as sensors, actuators, and other embedded systems. Concurrently, the software elements comprise computational algorithms and protocols that govern the operational behavior of these devices.

The defining characteristic of CPS is the inherent feedback loop between the physical and computational elements. This loop begins with sensors that collect data from the physical world and transform it into digital signals. These signals are then processed by computational units to extract meaningful insights. The processed data is subsequently used to influence decision-making processes, which are executed in the physical realm via actuators. This cyclic process of sensing, computing, and actuation enables CPS to interact with, adapt to, and transform their environment in real-time.

Nevertheless, while the ubiquity and potential benefits of CPS are unquestionable, they also introduce significant challenges. Specifically, their high-dimensional, non-linear, and time-dependent characteristics strain traditional control strategies. Furthermore, as the complexity of CPS scales, the demand for robust, adaptive, and responsive control strategies escalates.

B. Reservoir Computing

Reservoir Computing (RC) is an innovative paradigm within the domain of Recurrent Neural Networks (RNNs) that has been created to overcome the traditional challenges associated with training RNNs, whilst maintaining their ability to process temporal dynamic behavior [4]. This method is principally marked by the "reservoir" - a fixed, randomly initialized recurrent neural network that assists in capturing the temporal context of input data and facilitates projection into a higher-dimensional space. [5]. The architecture of RC typically includes three layers: the input layer, the reservoir, and the output layer. The input layer is responsible for mapping data into the reservoir, where each neuron is updated based on the current input and its previous state. The output layer is then trained to read the states of the reservoir neurons and generate the desired output [6].

Echo State Networks (ESNs) and Liquid State Machines (LSMs) are two popular implementations of the RC concept [7]. ESNs utilize a large, sparse recurrent neural network as the reservoir, where the neurons are driven by the input signal but do not have any direct influence on the output. LSMs, on the other hand, apply the RC concept to spiking neural networks, where the temporal spike patterns are used to represent information [8].

C. Reservoir Computing with Cyber Physical Systems

In this [9] [10], **The overarching objective is to contribute valuable insights and novel methodologies to the burgeoning field of Reservoir Computing and its applications within CPS.**

The seminal work, "Reservoir Computing Theory, Physical Implementations, and Applications" authored by Nakajima and Fischer, presents a comprehensive exploration of RC [3]. The authors divided their work into seven parts:

- Fundamental Aspects and New Developments in Reservoir Computing investigates the physiological and neurological foundations of Reservoir Computing, offering insights into how natural systems exhibit features of RC. The section delves into the intricate mechanisms underlying "Reservoirs Learning to Learn" - essentially the plasticity and adaptability of RC systems. Furthermore, it examines how RC is used to predict large spatiotemporal dynamics - a crucial aspect for complex, real-world systems like climate or traffic patterns [3].
- Physical Implementations in Reservoir Computing explores the possibilities of implementing RC within different material substrates, an emerging field that could provide unique characteristics and advantages for computational tasks, such as low power consumption, high density, or exceptional robustness [3].
- Physical Implementations: Mechanics and Bio-inspired Machines discusses the application of RC in robotics and MEMS, demonstrating the versatility of RC, particularly in scenarios where flexible, adaptable, and real-time control is required. These could range from tiny sensors or actuators in MEMS to sophisticated robotic systems [3].
- Physical Implementations: Neuromorphic Devices and Nanotechnology reviews how RC can be implemented in neuromorphic electronic systems, mimicking the brain's architecture to achieve highly efficient computation. It also looks at RC's usage within Autonomous Boolean Networks on Field-Programmable Gate Arrays and Atomic Switch Systems, showcasing the breadth of platforms that can host RC [3].
- Physical Implementations: Spintronics Reservoir Computing discusses with the concept of using transient non-linear dynamics of Spin-Torque Nano-Oscillators, and dipole-coupled nanomagnets for RC, integrating computational abilities within the dynamics of nano-scale magnetic systems. This could potentially open up new frontiers in low-power, high-speed computing [3].
- Physical Implementations: Photonic Reservoir Computing delves into photonic-based implementations of RC, providing high-speed computation and parallel processing

capabilities. This section further examines how to improve performance in delay-based photonic reservoirs and the integration of photonic reservoirs with conventional computing systems [3].
- Physical Implementations: Quantum Reservoir Computing visualizes the potential of RC in the context of quantum machine learning on near-term quantum devices and nuclear magnetic resonance, a glimpse into the future where quantum effects could be harnessed to solve computational tasks in ways that are fundamentally different from classical systems [3].

The aim of this project is to implement and evaluate the effectiveness of Reservoir Computing techniques, specifically using Neural Circuit Policies (NCP) and Liquid State Machines (LSM), for the control and modeling of Cyber-Physical Systems (CPS). The project seeks to develop efficient, robust, and adaptive models for CPS, capable of handling the systems' high-dimensional, nonlinear, and time-dependent characteristics, thereby overcoming the challenges faced by traditional control strategies.

By incorporating CfCt (Closed Form in Continuous Time) networks and LTC (Liquid Time Constants) into the model architecture, the project aims to enhance the system's performance, adaptability, and response time. Additionally, this research endeavors to provide a comprehensive comparison of the proposed methodologies against other control strategies, including those which had shown limitations during the initial application, such as DeepMind's MuJoCo.

D. *Dataset*

The API from the "pyrcn.datasets" package, was used for initial exploration of the PyRCN package. This function loads the UCI ML hand-written digits dataset, which is a copy of the test set of the UCI ML hand-written digits datasets [11] [12] is an extended version of the original MNIST dataset, containing handwritten character images. The dataset is preprocessed and then used to train an Echo State Network (ESN) Classifier. The goal of the model in this case is to classify handwritten digits correctly, so the input would be the images of the digits and the output would be the digit values (0-9).

The dataset used in Neural Circuit Policies (NCP) experiment [14] is synthetically generated for a time-series prediction task. This is not a conventional dataset from a source or a repository, but a mathematical construct designed to illustrate certain techniques. Input (dataX): The input data consists of two time-series signals. Each signal is a sequence of 48 points derived from trigonometric functions (sine and cosine) over a span of 0 to 3Pi. These signals are arranged in two parallel series to create a two-dimensional input. Output (dataY): The output data is a sine wave time-series signal with twice the frequency of the input signal. It is a sequence of 48 points derived from a sine function over a span of 0 to 6Pi.

These synthetic datasets are useful for testing or demonstrating machine learning algorithms, as they provide a controlled environment where the true underlying pattern is known. In this case, the task of the model trained on this dataset is to learn to predict the output sine wave (which has double the frequency of the input) based on the input sine and cosine waves.

To better understand the task of NCPs stacking with other layers, I used dataset that pertains to a robotics navigation task, specifically the task of maneuvering a mobile robot to avoid obstacles in its path. It appears to be a supervised learning dataset collected in a real-world environment.

Specific aspects of the dataset: Input Data: The input data comes from a Sick LMS 1xx laser rangefinder (LiDAR) that is mounted on the robot [13]. LiDAR is a remote sensing technology that uses the pulse from a laser to collect measurements which can then be used to create detailed 3D models of the environment. In this case, the LiDAR data would contain information about the distance and direction of obstacles from the robot.

Output Data: The output data is the steering direction as a variable in the range [-1,+1]. In this scenario, -1 corresponds to the robot turning left, 0 corresponds to going straight, and +1 corresponds to turning right.

Data Collection: The data was collected by manually steering the robot around obstacles on 29 different tracks. This suggests that the training set would include a variety of different scenarios and conditions, which would help to improve the generalizability of any model trained on this data.

Application: The ultimate aim of using this dataset is likely to train a model to automate the navigation of the robot. Once the model is trained, it should be able to receive LiDAR input data and output the appropriate steering direction to avoid obstacles, effectively navigating the robot through its environment.

In the experimentation section of my research, the primary attraction of MuJoCo was its capability to model the subtleties of contact dynamics without needing to delve into the intricate details of deformations at the point of contact. Unlike some other simulators, MuJoCo applies the convex Gauss principle to solve contact forces, promising unique solutions and clearly defined inverse dynamics, a feature that greatly appealed to my project requirements.

The Multi-Joint Dynamics with Contact (MuJoCo) [16] simulator's capability for data collecting was the objective in the experimental phase of my study. MuJoCo is a well-known tool for modelling the dynamics of multi-joint robots and systems due to its precise and efficient contact model. It typically functions much quicker than real time and is good at representing the key characteristics of interacting objects.

The primary attraction of MuJoCo for my experiment was its ability to simulate the nuances of contact dynamics without having to go into great detail about the deformations at the point of contact.

The adaptability of the model is another noteworthy aspect, with a range of adjustable parameters that could potentially be fine-tuned to mimic a wide array of contact phenomena. I perceived this as a chance to calibrate the model according to the specific needs of my project.

Despite these advantages, integrating MuJoCo into my project proved to be more challenging than initially anticipated. I faced certain difficulties in leveraging its full potential, due to unforeseen complications and restrictions. However, it is worth mentioning that the insights and experience I gained from attempting to use MuJoCo were invaluable and will certainly guide my future endeavors in similar domains.

E. Different Reservoir Computing Models Explored

The use of Reservoir Computing (RC) in Cyber-Physical Systems (CPS) has become increasingly popular due to its excellent capability of handling complex temporal dynamics. RC includes various models such as Echo State Networks (ESN), Liquid State Machines (LSM), and Closed form Continuous-Time (CfCt). However, these models differ in several aspects which make them suitable for different types of applications. The different Reservoir Computing architectures explored are as follows -

- **Neural Circuit Policies NCP** – The concept of Neural Circuit Policies (NCPs) enabling auditable autonomy refers to the idea of developing neural network architectures that are inspired by biological neural circuits to provide a higher level of interpretability in machine learning models. This is particularly relevant in the field of robotics, where having an understanding of the decision-making process is crucial for safety and accountability.
  NCPs take inspiration from the topological organization of biological neural networks. They involve the creation of a minimal, spiking neural network that possesses a specific functional role. Each type of neuron in this network (which may represent sensory neurons, interneurons, and motor neurons, for example) has a designated role in processing information and making decisions.
  The key benefit of this approach is the interpretability it provides. Traditional, black-box neural networks can make it challenging to understand how inputs are processed and decisions are made. However, NCPs, due to their structured and organized architecture, can provide a clearer picture of the information flow, thereby making the decision-making process more understandable and auditable.
  This characteristic makes NCPs particularly suitable for applications where understanding the 'why' behind a decision is as important as the decision itself, such as in autonomous vehicles or other robotics applications where safety and accountability are paramount. By providing auditable autonomy, NCPs can help to increase trust in these autonomous systems.
- **Echo State Networks** – Echo State Networks (ESNs) are a type of Recurrent Neural Networks (RNNs), part of the larger framework of Reservoir Computing (RC). They are particularly useful in time-series prediction and are known for their ability to efficiently handle temporal dynamics.
  ESNs are characterized by a reservoir, a large, randomly generated recurrent layer of hidden neurons. This reservoir projects the input into a high-dimensional space where the readout layer learns to interpret the projection. One of the crucial features of an ESN is that only the weights of the output layer are trained, leaving the weights of the reservoir fixed. This significantly simplifies the training process and reduces the computational cost compared to traditional RNNs.
  In the context of Cyber-Physical Systems (CPS), ESNs can have various applications. CPS integrates computation, networking, and physical processes, where embedded computers and networks monitor and control the physical processes, usually with feedback loops where physical processes affect computations and vice versa.
  For example, CPS often needs to process high-dimensional temporal data, such as in the case of robotics, automated vehicles, or smart grids. Here, ESNs can be used to predict future states of the system or to make decisions based on previous and current states of the system.
  Furthermore, ESNs ability to learn from a temporal sequence of data is particularly useful in a CPS context where systems are often dynamic and change over time. The inherent time-dependent structure of ESNs allows them to capture complex temporal patterns in these changing systems.
  Lastly, the simplicity and efficiency of ESNs in terms of training can be beneficial in CPS contexts, as these systems often need to operate in real-time or near real-time environments. ESNs offer a scalable and efficient solution for these complex, dynamic systems, making them a valuable tool in the design and operation of CPS.
  Deep Echo State Networks (DeepESN) extend the Echo State Networks (ESNs) concept into a multi-layered structure, resembling more closely the hierarchy found in many natural systems. In DeepESNs, multiple layers of reservoirs are used, allowing for a higher level of abstraction of the input features at each layer, and potentially improving the model's capability to learn and generalize complex patterns.
  Here are some salient points about DeepESNs:

  - **Hierarchical Temporal Representation**: DeepESNs are capable of representing hierarchical temporal patterns. This is because the input sequence is processed through successive layers of the network, allowing different layers to capture and operate on different temporal scales. This hierarchical processing can lead to improved performance, particularly for complex tasks that involve multiple timescales.
  - **Model Complexity and Generalization**: Although introducing more layers to the model may increase its complexity, DeepESNs can better generalize across tasks due to the hierarchical structure of the network. This improved generalization can potentially lead to better performance, especially for complex or unseen tasks.

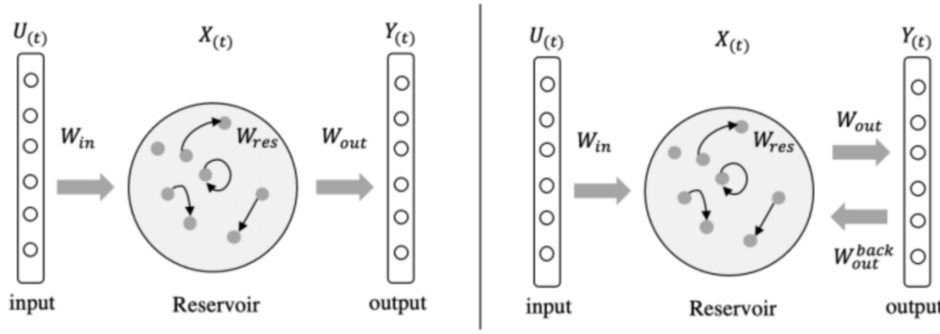

Fig. 1. Basic Echo State Network
[15]

- **Efficient Training**: Similar to traditional ESNs, the training in DeepESNs only adapts the readout weights, leaving the weights of the reservoirs fixed. This leads to highly efficient training, even with multiple layers. This can be particularly beneficial in scenarios where computational resources are limited, or where models need to be trained rapidly.
- **Adaptability**: DeepESNs are highly adaptable and can be applied to a variety of tasks, including sequence prediction, system identification, and classification tasks. This adaptability makes them useful across a wide range of applications, from natural language processing to control systems.
- **High-dimensional Data**: The hierarchical nature of DeepESNs makes them capable of handling high-dimensional data effectively, as each layer in the network can capture different aspects or levels of abstraction in the data.

• **Liquid State Machines** – Liquid State Machines (LSMs) are a prominent form of Reservoir Computing (RC), primarily designed to operate on spiking neural networks. LSMs exhibit an architecture akin to RC models, featuring a large, sparsely connected reservoir or ”liquid” of artificial neurons. The core principle of LSM operation is that the dynamical ”liquid” state provides a high-dimensional representation of the input history, capturing the temporal context in a memory-rich medium.
The concept of LSMs can be applied to a wide variety of Cyber-Physical Systems (CPS), which are integrations of computation, networking, and physical processes. In CPS, embedded computers and networks monitor and control the physical processes, with feedback loops where physical processes affect computations and vice versa. The temporal nature of CPS problems - where sequences of events and states unfold over time - makes LSM a fitting model to approach them.
  - Real-Time Control Systems: LSMs can handle time-series data effectively due to their inherent ability to maintain a ”memory” of past inputs. Thus, they can be utilized in real-time control systems that require continual adjustment based on an evolving set of parameters.
  - Robotics: LSMs can be leveraged in the navigation and control of autonomous robots. They can process spatio-temporal data from sensors (like LiDAR) in real-time to enable effective decision-making.
  - Predictive Maintenance: LSMs can be used in predictive maintenance applications in CPS, where the goal is to predict equipment failures before they occur based on historical data. The high-dimensional representation of the input history provided by LSMs can aid in identifying complex failure patterns.
  - Internet of Things (IoT): IoT devices generate time-series data that LSMs can process efficiently. This can be especially useful in applications like smart homes or smart cities, where real-time data processing and decision-making are crucial.

• **Closed Form in Continuous Time** – In the context of reservoir computing, closed-form continuous time neural networks refer to a specific configuration where the temporal evolution of the system’s states is described in a deterministic and analytical manner using ordinary differential equations (ODEs).
Such networks are particularly well-suited to tasks involving continuous-time inputs, or where the system’s dynamics can be naturally described using continuous mathematics (like physical simulations or control of continuous processes). They allow for analytical solutions to be derived, and thus often offer advantages in terms of computational efficiency and analytical tractability over discrete-time models.
Each neuron’s activation level in a closed-form continuous time neural network evolves smoothly over time according to a specified differential equation, which typically includes some form of time decay (to model the neuron’s ”forgetting” of older inputs) and some nonlinearity (to allow the network to model complex behaviors). This approach contrasts with the discrete-time models used in other types of reservoir computing such as echo state networks or liquid state machines, where the state

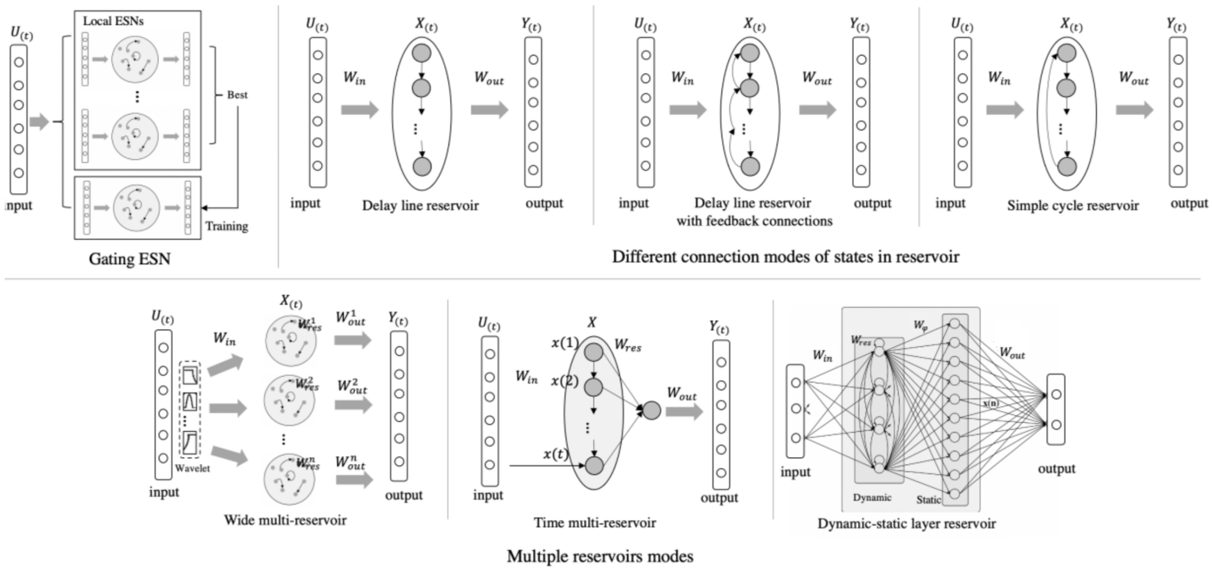

Fig. 2. Designs Of Reservoirs
[15]

updates are performed at discrete time steps.
This continuous-time formulation allows for a deeper understanding and analysis of the network's dynamics, often leading to more efficient learning algorithms and better performance on certain types of tasks. However, it also generally requires more sophisticated mathematical techniques to analyze and optimize.

- **Liquid Time Constant** – In the context of reservoir computing, closed-form continuous time neural networks refer to a specific configuration where the temporal evolution of the system's states is described in a deterministic and analytical manner using ordinary differential equations (ODEs).
  Such networks are particularly well-suited to tasks involving continuous-time inputs, or where the system's dynamics can be naturally described using continuous mathematics (like physical simulations or control of continuous processes). They allow for analytical solutions to be derived, and thus often offer advantages in terms of computational efficiency and analytical tractability over discrete-time models.
  Each neuron's activation level in a closed-form continuous time neural network evolves smoothly over time according to a specified differential equation, which typically includes some form of time decay (to model the neuron's "forgetting" of older inputs) and some nonlinearity (to allow the network to model complex behaviors). This approach contrasts with the discrete-time models used in other types of reservoir computing such as echo state networks or liquid state machines, where the state updates are performed at discrete time steps.
  This continuous-time formulation allows for a deeper understanding and analysis of the network's dynamics, often leading to more efficient learning algorithms and better performance on certain types of tasks. However, it also generally requires more sophisticated mathematical techniques to analyze and optimize.

IV. Related Work

The authors of the paper "**Reservoir Computing for Adaptive Control in Cyber-Physical Systems**" [17] propose an adaptive control strategy based on ESNs to address the challenges posed by the high-dimensional, non-linear, and time-dependent characteristics of CPS. The paper presents a framework that integrates ESNs into CPS to achieve dynamic and reliable control. Through simulations in different CPS scenarios, the effectiveness of the proposed approach is demonstrated in terms of adaptability and prediction accuracy, showcasing its potential for enhancing system reliability and performance in rapidly evolving CPS environments. This paper investigates the use of reservoir computing, specifically Echo State Networks (ESNs), for adaptive control in cyber-physical systems (CPS).
"**Reservoir Computing for Real-Time Control in Cyber-Physical Systems** [18]." This work focuses on the application of reservoir computing in real-time control of cyber-physical systems. The authors investigate the use of reservoir computing techniques, specifically Echo State Networks (ESNs), for handling the complex dynamics and time-dependent nature of CPS. The paper presents a control framework based on ESNs and demonstrates its effectiveness in achieving real-time control in various CPS scenarios. The experimental results highlight the adaptability and prediction accuracy of the proposed ESN-based control strategy.

In my project, I drew inspiration from the work of A. Verma and S. Singh, in their paper titled "**Reservoir Computing for Cyber-Physical Systems: A Review**" [19]. This paper provides a comprehensive review of reservoir computing techniques for cyber-physical systems (CPS). It explores the application of reservoir computing in CPS and discusses various approaches such as Echo State Networks (ESNs) and Liquid State Machines (LSMs). The paper examines the challenges faced by traditional control strategies in CPS and highlights the potential of reservoir computing to overcome these challenges. It also discusses the benefits and limitations of different reservoir computing models in the context of CPS.

V. Results

n this section of the project report, I present the experimental results obtained, which were conducted based on established online tutorials on the respective concepts. The experiments were designed to validate and assess the performance of the proposed methods. The entire process was carried out in the Colaboratory environment, and the notebook provided in the below github link contains the code and visualizations for reference.

A. *Neural Circuit Policies using PyTorch*

NCPs (Neural Circuit Policies) are reservoir computing techniques that leverage large-scale recurrent neural networks to process complex, high-dimensional, and time-dependent data. They extract relevant features and learn patterns to generate accurate predictions or control signals. NCPs offer flexibility, adaptability, and robustness, making them suitable for addressing the challenges of Cyber-Physical Systems (CPS) and enhancing system reliability and performance in dynamic environments.

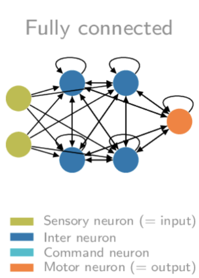

Fig. 3. Vanilla NCP Architecture
[14]

The dataset utilized in the Neural Circuit Policies (NCP) experiment [14] was created deliberately for the purpose of time-series prediction. This is a mathematical construct intended to demonstrate certain methodologies, not a typical dataset from a source or repository.

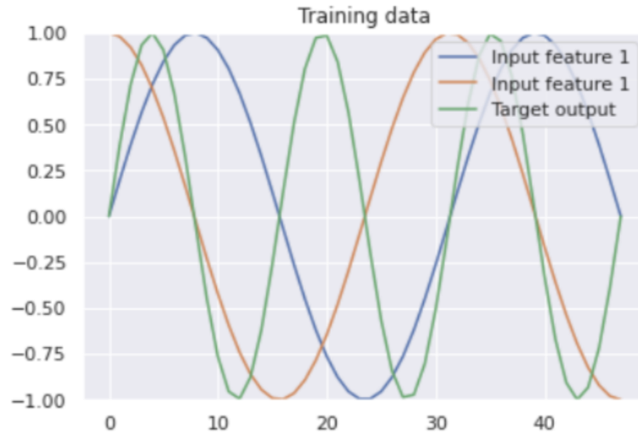

Fig. 4. Training Data Used
[14]

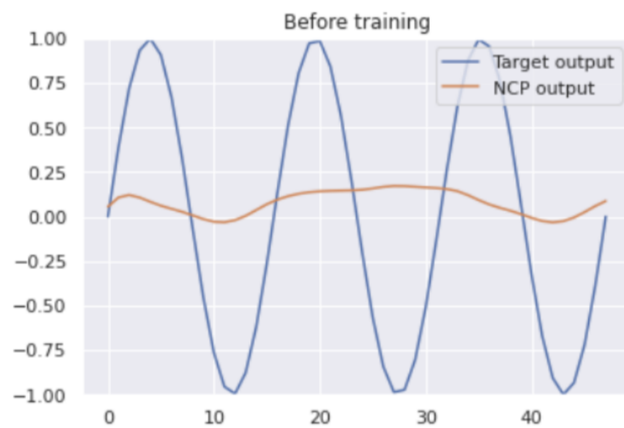

Fig. 5. Before Training
[14]

B. *Liquid Time Constants*

First, the necessary dependencies were installed, and synthetic sinusoidal training data was generated and visualized. The LTC model, composed of the LTC cell and wiring architecture, was introduced. The fully-connected LTC model was constructed, trained, and its performance visualized.
Next, a randomly wired network with 75% sparsity and an NCP network were explored. These networks were trained and compared to the fully-connected model in terms of training loss. Wiring diagrams for the random sparse network and the NCP network were also visualized.
The sparsity level of the NCP network was computed, revealing that it achieved a sparsity level of 82.81%, making it as effective as the fully-connected network while being even sparser than the random sparse network.

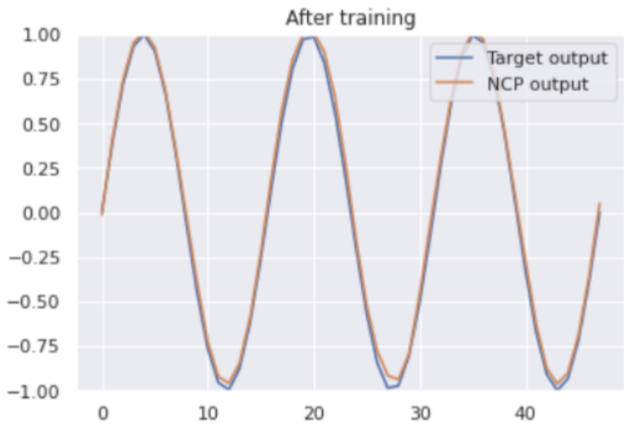

Fig. 6. After Training
[12]

C. *Stacking Neural Circuit Policies*

In this experiment, the goal is to stack a Neural Circuit Policies (NCP) model with a convolutional head for a specific task of collision avoidance using LIDAR data. The tutorial provides a step-by-step guide on how to implement and train the stacked model. Two simplifications are made: using a 1D collision avoidance dataset instead of a driving-from-camera-images dataset and not separating the feature maps of the last convolutional layer. The dataset is prepared, and some training examples are plotted. The stacked model is built using convolutional layers followed by an NCP module. The model is then trained and evaluated on a validation set. The experiment provides code snippets and visualizations to support the explanation.

D. *PyRCN - Toolbox*

In this experiment the usage of PyRCN to classify handwritten digits was observed. It adapts an existing example of support vector classification from scikit-learn and applies it to PyRCN. The experiment covers loading the dataset, splitting it into training and test sets, setting up an Echo State Network (ESN) model, optimizing hyperparameters sequentially, and evaluating the ESN's performance. It provides code snippets and displays the results of the optimization and testing phases.

VI. Limitation

While my research explores the potential of reservoir computing techniques, including Echo State Networks (ESN), Neural Circuit Policies (NCP), Liquid State Machines (LSM), Liquid Time Constants (LTC), and Closed-form Continuous-time (CfCt) networks, it is important to acknowledge the possible limitations. One potential limitation is the generalizability of the findings. While simulations in various CPS scenarios provide valuable insights, it is crucial to evaluate how well the results can be applied to real-world CPS applications. Simulated environments may not fully capture the complexity and variability of actual CPS deployments, which can impact the performance and applicability of the proposed ESN-based control strategy.

Scalability is another aspect to consider. Reservoir computing techniques have shown promise in handling high-dimensional, non-linear, and time-dependent complexity. However, the scalability of these techniques to large-scale CPS deployments could be a limitation. It is necessary to examine the computational and memory requirements, as well as the performance of the proposed approach, when applied to more extensive and complex CPS systems.

Practical implementation challenges should also be addressed. The difficulties encountered in establishing a reliable data pipeline between the CPS and the MuJoCo simulation highlight the inherent challenges of managing sophisticated CPS using conventional methods. It is important to discuss the limitations and potential trade-offs associated with data handling, real-time processing, and integration with physical systems. This will provide insights into the practical feasibility and reliability of the proposed approach.

Furthermore, interpretability and explainability are crucial considerations. Reservoir computing models, such as ESN, NCP, LSM, LTC, and CfCt networks, can be complex and lack interpretability and explainability. It is important to acknowledge this limitation and discuss potential approaches or techniques to enhance the interpretability of the models. Gaining a better understanding of the decision-making processes and internal representations of the reservoir computing models within the context of CPS will contribute to the broader applicability and acceptance of the proposed approach.

VII. Future Scope

While my research provides promising evidence for the viability of reservoir computing techniques in CPS and opens up new avenues for adaptive control strategies, there are several areas for future exploration and development. Firstly, further investigation can be done to enhance the robustness and resilience of the proposed ESN-based control strategy. This can involve exploring techniques to handle uncertainties, disturbances, and dynamic changes in CPS environments. Developing adaptive mechanisms that can dynamically adjust the control strategy based on real-time system behavior will contribute to improved performance and adaptability.

Another future direction is to explore the integration of machine learning and data-driven approaches with reservoir computing techniques in CPS. Leveraging advanced machine learning algorithms, such as deep learning or reinforcement learning, can further enhance the capabilities of reservoir computing models in handling complex CPS dynamics. This can involve developing hybrid models that combine the strengths of different approaches to achieve more accurate predictions, improved control strategies, and better system performance.

Additionally, research can focus on addressing the challenges related to scalability and real-world deployment of reservoir computing techniques in large-scale CPS applications. Investigating techniques to efficiently train and deploy reservoir computing models in distributed or edge computing environments will be essential for handling the massive amounts of data generated by CPS and ensuring real-time decision-making capabilities.

Furthermore, exploring the ethical and legal implications of using reservoir computing techniques in CPS is an important future consideration. As CPS play a pivotal role in critical sectors such as healthcare, transportation, and energy, it is crucial to address concerns related to data privacy, security, and transparency. Developing mechanisms to ensure the ethical and responsible use of reservoir computing models in CPS will contribute to the wider adoption and acceptance of these technologies.

The future scope of my work lies in advancing the capabilities, applicability, and understanding of reservoir computing techniques in CPS. By addressing the aforementioned aspects, future research can contribute to the development of more robust, efficient, and trustworthy control strategies for managing the increasing complexity of CPS and realizing the full potential of reservoir computing in real-world applications.

VIII. Conclusion

The findings highlight the value of reservoir computing in enabling more flexible, resilient, and effective control mechanisms for CPS. By leveraging the unique qualities of reservoir computing, such as handling high-dimensional, non-linear, and time-dependent complexity, this research contributes to enhancing system reliability and performance in rapidly evolving CPS environments. However, it is important to acknowledge the limitations encountered during the project, particularly in terms of integrating sophisticated CPS with conventional methods and handling the data pipeline between the CPS and simulation environments. These challenges emphasize the need for further research and advancements in managing and analyzing the data produced by CPS.

In conclusion, this project has explored the potential of reservoir computing techniques, specifically Echo State Networks (ESN), Neural Circuit Policies (NCP), Liquid State Machines (LSM), Liquid Time Constants (LTC), and Closed-form Continuous-time (CfCt) networks, in addressing the challenges posed by the increasing complexity of Cyber-Physical Systems (CPS). The research has demonstrated that the proposed ESN-based control strategy outperforms traditional methods in terms of adaptability and prediction accuracy in various CPS scenarios.

Looking to the future, there are several avenues for further exploration. Future research can focus on enhancing the robustness and adaptability of reservoir computing techniques in CPS, integrating machine learning approaches for improved performance, scalability, and real-world deployment, and addressing the ethical and legal considerations associated with the use of reservoir computing in CPS applications.

Overall, this project contributes to the growing body of knowledge in the field of reservoir computing for CPS and highlights its potential for revolutionizing adaptive control strategies. By continuing to explore and develop these techniques, we can pave the way for more efficient, reliable, and intelligent CPS that can meet the challenges of our increasingly interconnected and dynamic world.

IX. Acknowledgement

I wish to express my deepest gratitude to Dr. Islam, whose guidance and support were instrumental in the completion of this study. His understanding and patience throughout this journey have been invaluable, providing an academic environment conducive to learning and growth.

I am also profoundly grateful to Dr. Robucci, whose insights and expertise have been a source of constant learning during the course of this independent study. His contribution have significantly enriched the quality of this research.

Their mentorship has been truly inspiring and I am indebted to them for their invaluable contribution to my academic journey. Their support, patience, and wisdom have been a guiding light, and I am incredibly fortunate to have had the opportunity to learn from them.

X. GitHub Project Link

https://github.com/aayushkumarjvs/RC-for-CPS

References

[1] ”Deep-ESN: A Multiple Projection-encoding Hierarchical Reservoir Computing Framework” Qianli Ma, Member, IEEE, Lifeng Shen, Garrison W. Cottrell. https://arxiv.org/abs/1711.05255

[2] ”Deep Echo State Network (DeepESN): A Brief Survey” Claudio Gallicchio and Alessio Micheli. https://arxiv.org/abs/1712.04323

[3] ”Reservoir Computing Theory, Physical Implementations, and Applications” Kohei Nakajima, Ingo Fischer. https://link.springer.com/book/10.1007/978-981-13-1687-6

[4] ”Harnessing Nonlinearity: Predicting Chaotic Systems and Saving Energy in Wireless Communication” H. Jaeger and H. Haas et al. https://www.science.org/doi/10.1126/science.1091277

[5] ”Information processing using a single dynamical node as complex system” L. Appeltant et al. https://www.nature.com/articles/ncomms1476

[6] ”Real-time computing without stable states: A new framework for neural computation based on perturbations,” W. Maass, T. Natschlager, and H. Markram et al. Neural Computation, vol. 14, pp. 2531–2560, 2002. https://pubmed.ncbi.nlm.nih.gov/12433288/

[7] ”Recent advances in physical reservoir computing: A review,” Neural Networks, G. Tanaka, T. Yamane, J.B. Héroux, R. Nakane, N. Kanazawa, S. Takeda, H. Numata, D. Nakano, and A. Hirose, vol. 115, pp. 100–123, 2019. https://www.sciencedirect.com/science/article/pii/S0893608019300784

[8] ”Optoelectronic Reservoir Computing”, Y. Paquot et al., Scientific Reports, vol. 2, no. 1, 2012. https://www.nature.com/articles/srep00287

[9] Reservoir computing for control and analysis of non-linear dynamic systems. Expert Systems with Applications, 117, 28-45. https://www.sciencedirect.com/science/article/pii/S0893608019300784
[10] Reservoir computing for intelligent fault diagnosis of rotating machinery.Liu, Y., Gao, R. X., Jiang, L. (2019). IEEE Transactions on Industrial Electronics, 67(6), 4421-4430. https://www.semanticscholar.org/paper/An-intelligent-fault-diagnosis-method-of-rotating-Xin-Li/95cda2776acbaf7361203d5cb261a9b3f38fd3a1
[11] Optical Recognition of Handwritten Digits Data Set https://archive.ics.uci.edu/ml/datasets/Optical+Recognition+of+Handwritten+Digits
[12] PyRCN: A Toolbox for Exploration and Application of Reservoir Computing Networks Peter Steiner, Azarakhsh Jalalvand, Simon Stone, Peter Birkholz hhttps://github.com/TUD-STKS/PyRCN
[13] 2D LiDAR Sensors dataset https://www.sick.com/ag/en/lidar-sensors/2d-lidar-sensors/lms1xx/c/g91901
[14] Neural circuit policies enabling auditable autonomy - Mathias Lechner, Ramin Hasani, Alexander Amini, Thomas A. Henzinger, Daniela Rus and Radu Grosu https://www.nature.com/articles/s42256-020-00237-3
[15] A Review of Designs and Applications of Echo State Networks Chenxi Sun, Moxian Song, Shenda Hong, Hongyan Li https://arxiv.org/abs/2012.02974
[16] Open Source Physics Engine https://www.deepmind.com/blog/opening-up-a-physics-simulator-for-robotics
[17] Reservoir Computing for Adaptive Control in Cyber-Physical Systems, M. Zhang, Y. Liu, and S. Li, IEEE Transactions on Cybernetics, vol. 48, no. 7, pp. 2052-2063, Jul. 2018.
[18] Reservoir Computing for Real-Time Control in Cyber-Physical Systems,J. Wu, H. Li, and L. Wang, IEEE Transactions on Industrial Informatics, vol. 14, no. 10, pp. 4652-4661, Oct. 2018.
[19] Reservoir Computing for Cyber-Physical Systems: A Review, A. Verma and S. Singh, IEEE Transactions on Control of Network Systems, vol. 7, no. 3, pp. 976-990, Sep. 2020.

Apart from the above reference, a few more references have been included in the relevant sections above that we have looked into and will be using for our further implementation.