The 2nd International Conference on Advanced Technology and Sustainable Development – 2022

PRESENTATION – CIIA 2022

November 24-26, 2022

Organized by
Industrial University of Ho Chi Minh City & Eastern International University
Vietnam

2022 International Conference on Computational Intelligence and Innovative Applications (CIIA 2022)

INDUSTRIAL UNIVERSITY OF HO CHI MINH CITY
PUBLISHING HOUSE

2022 INTERNATIONAL CONFERENCE ON COMPUTATIONAL INTELLIGENCE AND INNOVATIVE APPLICATIONS (CIIA2022)

Time		*November 25, 2022 Location: B4.5 (Building B, Floor 4)*	
		Chair: Prof. Ho Pham Huy Anh, Assoc.Prof. Nguyen Tan Luy	
10:45	11:10	**Keynote Speaker 1** (Prof. Narayan C. Debnath)	RELIABILITY OF INTELLIGENT SOFTWARE SYSTEMS: AN EXCITING CHALLENGE AHEAD ***Prof. Narayan C. Debnath, Eastern International University***
11:10	11:35	**Keynote Speaker 2** (Prof. Ho Pham Huy Anh)	ROBUST CONTROL OF UNCERTAIN NONLINEAR SYSTEMS USING ADAPTIVE REGRESSIVE NEURAL-BASED DEEP LEARNING TECHNIQUE ***Prof. Ho Pham Huy Anh, HCM City University of Technology***
11:35	12:00	**Keynote Speaker 3** (Prof. Trung Q. Duong)	EDGE INTELLIGENCE-BASED ULTRA-RELIABLE AND LOW-LATENCY COMMUNICATIONS FOR DIGITAL TWIN-ENABLED METAVERSE ***Prof. Trung Q. Duong, Queen's University Belfast (UK)***
12:00	12:25	**Keynote Speaker 4** (Dr. Antonino Masaracchia)	DIGITAL TWIN FOR OPEN RAN: TOWARDS INTELLIGENT AND RESILIENT 6G RADIO ACCESS NETWORKS ***Dr. Antonino Masaracchia (Member, IEEE)***

2022 INTERNATIONAL CONFERENCE ON COMPUTATIONAL INTELLIGENCE AND INNOVATIVE APPLICATIONS (CIIA2022)

SECTION 1: INTELLIGENT SYSTEMS, CONTROL AND APPLICATIONS

Time		*November 25, 2022 Location: B4.2 (Building B, Floor 4)*	
		Chair: Prof. Ho Pham Huy Anh, Assoc. Prof. Nguyen Tan Luy	
14:00	14:15	**ICATSD2F.103**	ADAPTIVE NONSINGULAR TERMINAL SLIDING MODE CONTROL FOR MANIPULATOR ROBOT ***Mai Thang Long, Tran Huu Toan, Tran Van Hung, Tran Ngoc Anh, Nguyen Hoang Hieu, Nguyen Thi Hong Ha***
14:15	14:30	**ICATSD2F.104**	BIOMECHANICS-BASED DEVELOPMENT OF AN UPPER LIMB REHABILITATION ROBOT ***Huu-Toan Tran, Thang-Long Mai, Van-Hung Tran, Ngoc-Anh Tran, Thanh-Hai Tran, Hoang-Hieu Nguyen***
14:30	14:45	**ICATSD2F.105**	DEVELOPMENT OF A POSITIONING SYSTEM FOR MOBILE ROBOTS USING LIDAR SENSOR IN UNDETERMINED ENVIRONMENTS ***Nguyen Van Lanh***
14:45	15:00	**ICATSD2F.107**	TELEOPERATION OF A CAR-LIKE MOBILE ROBOT WITH HAPTIC JOYSTICK USING POTENTIAL FIELD BASED FORCE FEEDBACK ***Anh Khoa Tran, Hung Hoang, Duc Thien Tran***
15:00	15:15	**ICATSD2F.108**	ADAPTIVE SLIDING MODE CONTROL FOR THE 6 DOF MANIPULATOR WITH TIME-VARYING PAYLOAD ***Ha Thanh Binh, Tong Hai Ninh, Tran Minh Phuc, Tran Duc Thien, Le Hoang Lam***
15:15	15:30	**ICATSD2F.112**	OPTIMAL TRACKING CONTROL FOR ROBOT MANIPULATORS WITH INPUT CONSTRAINT BASED REINFORCEMENT LEARNING ***Nguyen Duc Dien, Nguyen Tan Luy, Lai Khac Lai, Tran Thanh Hai***
15:30	15:45	**ICATSD2F.117**	A NEW CABLE DRIVE SOLUTION FOR TORQUE CONTROL OF EXOSKELETON'S MOTORIZED JOINTS ***Tan Hung Huynh, Minh Thong Nguyen, Hau Tran Van, Viet Anh Dung Cai, Viet Thang Nguyen, Long Triet Giang Huynh***
15:45	16:00	**ICATSD2F.124**	AN IMPROVEMENT DESIGN OF MULTI-FUNCTION CONTROLLER FOR HIGH-TECH SHRIMP FARM ***Bui Thu Cao***

2022 INTERNATIONAL CONFERENCE ON COMPUTATIONAL INTELLIGENCE AND INNOVATIVE APPLICATIONS (CIIA2022)

SECTION 2: CYBER-PHYSICAL SYSTEMS, NETWORKS AND APPLICATIONS

Time		*November 25, 2022 Location: B4.3 (Building B, Floor 4)*	
		Chair: Dr. Mai Thang Long, Dr. Nguyen Ngoc Son	
14:00	14:15	**ICATSD2F.106**	A STUDY ON DIRECT AND SIMULTANEOUS ANALYSIS OF ^{238}U AND ^{226}Ra NUCLIDES ***Vo Xuan An***
14:15	14:30	**ICATSD2F.111**	PERFORMANCE ANALYSIS OF TWO-WAY NETWORK WITH NONLINEAR ENERGY HARVESTING RELAY AND DIGITAL NETWORK CODING ***Thu- Thuy Thi Dao, Pham Ngoc Son***
14:30	14:45	**ICATSD2F.114**	A NEW APPROACH IN CALCULATING CURRENT-VOLTAGE CHARACTERISTICS OF THE QUANTUM DOT GATE FIELD EFFECT TRANSISTOR ***An Nguyen Van***
14:45	15:00	**ICATSD2F.118**	PERFORMANCE ANALYSIS 16QAM-OFDM SIGNAL EMPLOYING DVB-T2 STANDARD IN 8K MODE ***Nguyen Hoang Viet***
15:00	15:15	**ICATSD2F.119**	A NOVEL MULTILEVEL INVERTER USING SMALL CAPACITORS ***Ngo Bac Bien, Phan Xuan Dung, Trinh Ngoc Duc, Ngo Thi Kim Linh***
15:15	15:30	**ICATSD2F.122**	COOPERATIVE NOMA-ENABLED CELLULAR INTERNET-OF-THINGS SYSTEMS: PERFORMANCE ANALYSIS ***Tien-Tung Nguyen, Anh Vinh Nguyen, Hung Le Van, and Tan Loc Nguyen***

2022 INTERNATIONAL CONFERENCE ON COMPUTATIONAL INTELLIGENCE AND INNOVATIVE APPLICATIONS (CIIA2022)

SECTION 3: MACHINE VISION, MACHINE LEARNING AND APPLICATIONS

Time		*November 25, 2022*	*Location: B4.4 (Building B, Floor 4)*
		Chair: Assoc. Prof. Huynh Trung Hieu	
14:00	14:15	**ICATSD2F.101**	SEMANTIC VEHICLE SEGMENTATION FROM AERIAL IMAGES USING DEEP LEARNING ***Van Luan Tran, Huei-Yung Lin, Manh-Hung Ha***
14:15	14:30	**ICATSD2F.102**	ESTIMATING LASER IRRADIATION COORDINATES FROM TEMPERATURE DISTRIBUTION USING CONVOLUTIONAL NEURAL NETWORK ***Miki Nakaone, Tomomasa Ohkubo, Yuki Ueno, Ken Goto, Yutaka Kagawa***
14:30	14:45	**ICATSD2F.109**	FUSE DETECTION ON INDUSTRIAL PRODUCTION LINE WITH MACHINE LEARNING VISION ***Thanh Hai Diep, Thanh Son Nguyen, Ngoc Hay Nguyen, Van Luan Tran***
14:45	15:00	**ICATSD2F.115**	ATTENTION MODELS FOR COVID-19 DETECTION BASED ON LUNG ULTRASOUND IMAGES ***Hoa Thanh Le, Thao Danh Nguyen, Linh Nguyen***
15:00	15:15	**ICATSD2F.116**	HAND-WRITTEN CHARACTER RECOGNITION BY WRITING SOUND USING A ONE-DIMENSIONAL CONVOLUTIONAL NEURAL NETWORK ***Iori Tajima, Hiroaki Kurokawa***
15:15	15:30	**ICATSD2F.120**	BUILDING A QUESTION ANSWERING MODEL TO SUPPORT STUDENTS USING DEEP LEARNING ***Do Tan Hai, Dang Thi Phuc, Tran Cong Thinh, Nguyen Phuc Hung***

2022 INTERNATIONAL CONFERENCE ON COMPUTATIONAL INTELLIGENCE AND INNOVATIVE APPLICATIONS (CIIA2022)

SECTION 4: COMPUTATIONAL INTELLIGENCE AND APPLICATIONS

Time		*November 25, 2022 Location: B4.5 (Building B, Floor 4)*	
		Chair: Prof. Narayan C. Debnath, Dr. Dang Thi Phuc	
14:00	14:15	**ICATSD2F.110**	FRÉCHET MEANS IN SUB-RIEMANNIAN MANIFOLD ***Thanh-Son Trinh***
14:15	14:30	**ICATSD2F.113**	ENRICHING KNOWLEDGE GRAPH OF COMPUTING DOMAIN ONTOLOGY BY HETEROGENEOUS RESOURCES ***Ta Duy Cong Chien***
14:30	14:45	**ICATSD2F.121**	AN EFFICIENTLY METHOD DETERMINATION THE REACHABLE SET OF GEOSPATIAL DATA IN NETWORK SPACE ***Trang T.D Nguyen, Loan T.T Nguyen, L.N.Duy***
14:45	15:00	**ICATSD2F.123**	PLANT CLASSIFICATION IN SOUTHEAST ASIA USING HIGH-RESOLUTION NETWORK ***Dang Ngan Ha, Tran Hong Ngoc, Hieu Trung Huynh***

Prof. Narayan C. Debnath, Ph.D., D.Sc.

Founding Dean

School of Computing and Information Technology

Eastern International University, Vietnam

Professor Dr. Narayan C. Debnath is currently the Founding Dean of the School of Computing and Information Technology at Eastern International University, Vietnam, where he joined in 2018. He is also serving as the Head of the Department of Software Engineering at Eastern International University, Vietnam. Formerly, Dr. Debnath served as a Full Professor of Computer Science at Winona State University, Minnesota, USA for 28 years (1989-2017), and the elected Chairperson of the Computer Science Department at Winona State University for 7 years (2010-2017). Dr. Debnath has been the Director of the International Society for Computers and their Applications (ISCA), USA since 2014, and also served as the ISCA President for 2 separate terms (2005-2007 and 2011-2013).

IIC 2022
ADVANCED TECHNOLOGY AND SUSTAINABLE DEVELOPMENT

INDUSTRIAL UNIVERSITY OF HOCHIMINH CITY

EIU

CỤC CÔNG TÁC PHÍA NAM

The 2nd International Conference on Advanced Technology & Sustainable Development (ICATSD 2022)

Reliability of Intelligent Software Systems: An Exciting Challenge Ahead

Professor Narayan C. Debnath, Ph. D., D. Sc. [1,2,3]

[1] Founding Dean
School of Computing and Information Technology
Eastern International University, Vietnam,
[2] Head
Department of Software Engineering
Eastern International University, Vietnam,
[3] Director and Past President
International Society for Computers and Their Applications (ISCA)
Minnesota, USA.

CONTENTS


Winona, MN

Eastern International University (EIU)
Binh Duong Province , Vietnam

Keynote Address

- What we already know (Current Progress)
- What we do NOT know (Challenges and opportunities)

What We Really Want ?

- **Smart Home, Smart City, Smart Car, Smart Road, Smart Aircraft, Smart phone, Smart Warehouse, Smart finance, Smart devices, Smart computing power, and…**
- **Smarter Education (On-line?), Smarter Conference and Meetings (Virtual?), Smarter Industry Performance (Work from Home?), and…**
- **Intelligent communication systems**
- **Intelligent Safety and Security systems**
- **Intelligent protection against cybersecurity attack, etc. etc. ….**

What We Really Want ?

Simply Speaking, We Want:

<u>INTELLIGENT Devices and Applications,</u> which require and depend on <u>INTELLIGENT Software Systems</u>

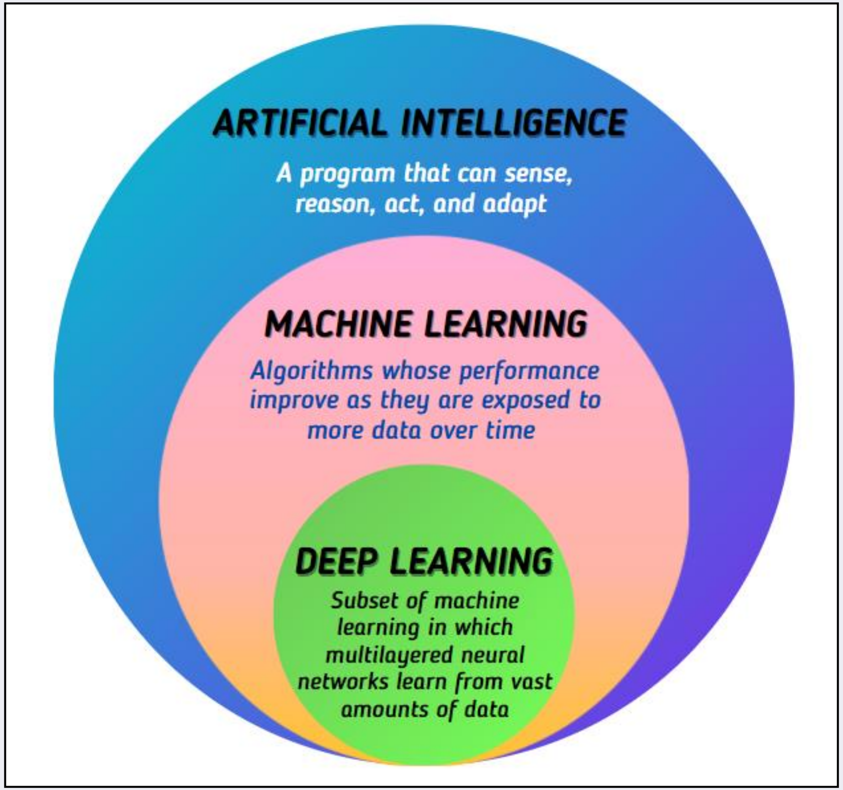
How to Achieve What we want?

- AI

Artificial Intelligence

- AI

Artificial Intelligence (AI)



Artificial Intelligence (AI)

- The function and popularity of AI are rapidly increasing by the day.

- AI alone or in combination with algorithms in Machine Learning, Deep Learning, Pattern Recognition, Computer vision, and Block chain is helping software development industries to build intelligent software in a wide range of applications.

- AI is revolutionizing the industries with its applications and helping solve complex problems.

Intelligent Software Applications

- In E-commerce
  - Personalized shopping (customer relationship)
  - AI-powered assistants using Natural Language Processing (virtual shopping assistants)
  - Fraud Prevention (credit cards, fake news)
- In Social Media
  - Instagram
  - Facebook
  - Twitter

Intelligent Software Applications

- In Education
  - Administrative tasks automated to aid educators
  - Creating smart content (digitization: video lecture, conferences, textbooks, etc.)
  - Voice assistants (to improve students learning)
- In Human Resource (AI + ML)
  - Simplify Hiring Process with intelligent software
- In Chatbots (AI + ML)
  - 24/7 Live chat and Customer Service for customer satisfaction

Intelligent Software Applications

- In Navigation
  - Improve Operational Efficiency
  - Analyze Road Traffic
  - Optimize Routes
  - Global Positioning System (GPS) technology (accurate time and improve safety)
- In Agriculture (AI + Computer vision + Robotics + ML)
  - Identify Defects and Nutrient deficiencies in the soil
  - Improve food production and improve farming

Intelligent Software Applications

- In Lifestyle (AI + ML)
  - Automotive (Electric and Autonomous Vehicle)
  - GPS, Maps, and Voice Navigation Technology
  - Spam Filter
  - Facial Recognition
  - Recommender system (User data for customized recommendation)
- In Automobiles
  - Self-Driving vehicles (emergency braking, blind-spot monitoring, driver assist steering) using intelligent sensors and powerful cameras

Intelligent Software Applications

- In Gaming and Sports
  - Smart, human-like non-player characters (NPCs) to interact with the players
  - Predict human behavior using which game design and testing can be improved
  - Man Vs. Machine
  - IBM Watson Supercomputer wins $1 million dollar in Jeopardy game on February 16, 2011
  - Most recent intelligent machine was built using rules and human intelligent, but can defeat human talent. Experts call this a technological breakthrough

Intelligent Software Applications

- In Marketing (AI + ML + Pattern recognition)
  - Highly targeted and personalized ads
- In Finance
  - 80% bank uses this benefits for Personal finance, Corporate finance, Consumer finance
  - Improves wide range of financial services
- In Telecommunication industries
  - Enhance customer experience
  - Optimize 5G Network

Intelligent Software Applications

- In Robotics
  - Carrying goods in Hospitals, Factories and Warehouses
  - Cleaning offices and large equipment
  - Inventory management
  - Human-Robot communications
  - Robot in manufacturing industry
  - Robot in Agriculture, and Mining industries
  - Robot for Healthcare

Intelligent Software Applications

- In Healthcare and Medicine (AI + ML)
  - Sophisticated machines that can detect diseases and identify cancer cells
  - Increases clinical efficiency
  - Boost diagnosis speed and accuracy
  - Improves patient outcomes
  - Use historical data and medical intelligence for discovery of new drugs

Intelligent Software Applications

- Intelligent Transportation System (ITS)
  - ITS is an advanced application which aims to provide innovative services relating to different modes of transport and traffic management, and enable users to be better informed and make safer, more coordinated and "smarter" use of transport networks.
  - AI and Machine learning applications to automobiles to railways to Aviation to Healthcare industries are helping to significantly improve reliability, services, safety, and protection to human life and lifestyle.

Computer Users vs. Computer Scientists

- Users believe we get 100% and life is really great:

  - Good News! Technology meets the expectations

- Computer scientists, software engineers, and software developers have a very different perspective:

Do We Really Get, What We Really Want?

leading to practical challenges for now and the future.

- Bad News! Yet, Exciting and Motivating with life-long opportunity to improve and contribute with future research and development.

Software Reliability

What is it ?

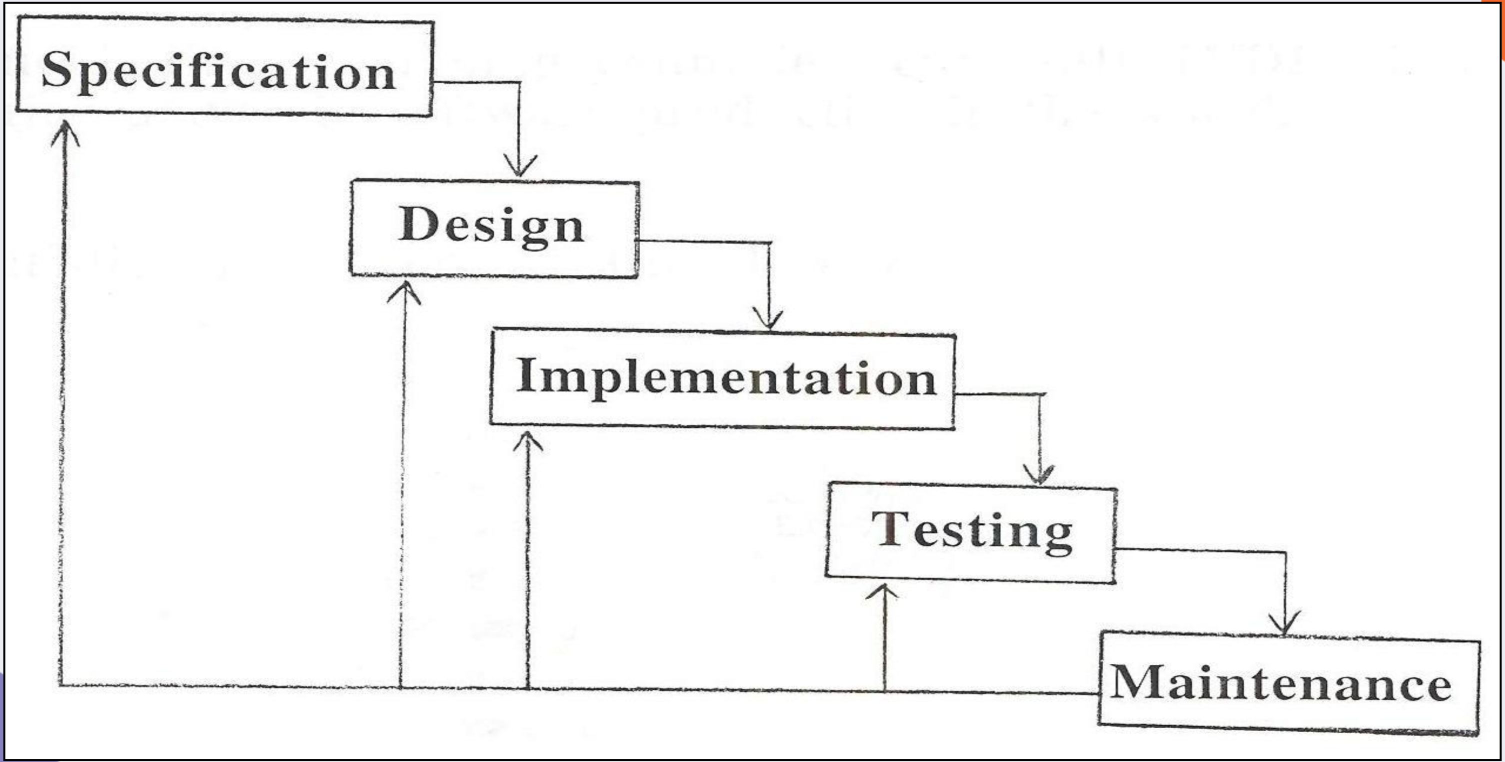
What Affects it ?

Why is it important ?

Can we achieve 100% ?

IIC 2022 ADVANCED TECHNOLOGY AND SUSTAINABLE DEVELOPMENT

Software Development Process Model



What is Software Reliability?
IEEE - Definition 1

- The IEEE defines Software Reliability as:
  - The probability that software will not cause a system failure for a specified time under specific conditions.

What is Software Reliability?
AT & T - Definition 2

- AT&T Bell Lab (John Musa) defines software reliability as:

  The probability that a given software system operates for some time period without software error, on the machine for which it was designed, given that it is used within design limit.

What is Software Reliability?
Debnath - Definition 3

- Reliability of a software system is a “measure” of how well it provides services expected of it by its users.

- The goal is to achieve 100% reliability from any software.

Why do we need software reliability?

- The demand for large, complex, intelligent software systems has increased more rapidly than the ability to design, implement, test, and maintain them.
- When the requirements and dependencies on computers increase, the possibility of crises from computer failures also increases. The impact of this failures ranges from inconvenience, economic damages, and loss of life.
- Needless to say, the reliability of computer systems (software) has become a major concern for our society.

What factors Affect Reliability

- Software Specification
- Correctness of Software Design
- Correctness of the mapping of the software design to Implementation
- Reliability of each component making up the software system

Typical Question

- Can we achieve 100% Reliability from a software system?
  - Yes
  - May be
  - No

Objective

- **FOCUS ON SOFTWARE VALIDATION**

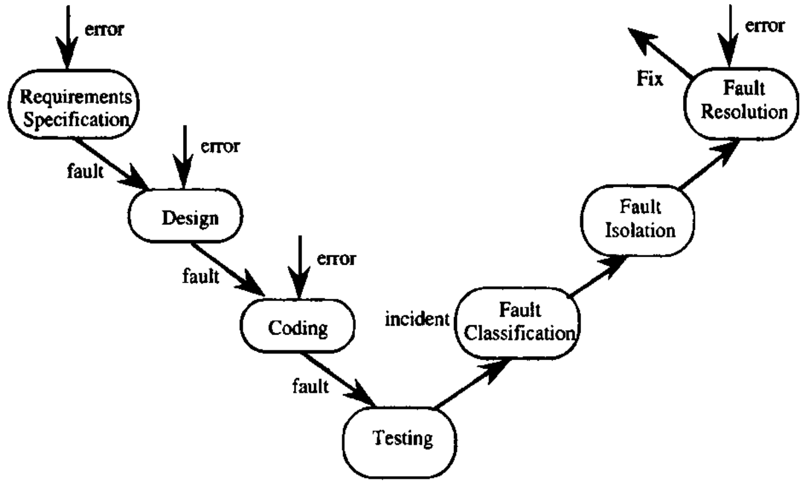
  The GOAL of validation is to establish confidence that the software should perform as intended.

- **Methods of Validation**
  - **Formal Proof of Correctness**
  - **Software Reliability Analysis**
  - **Software Testing**

Software Testing

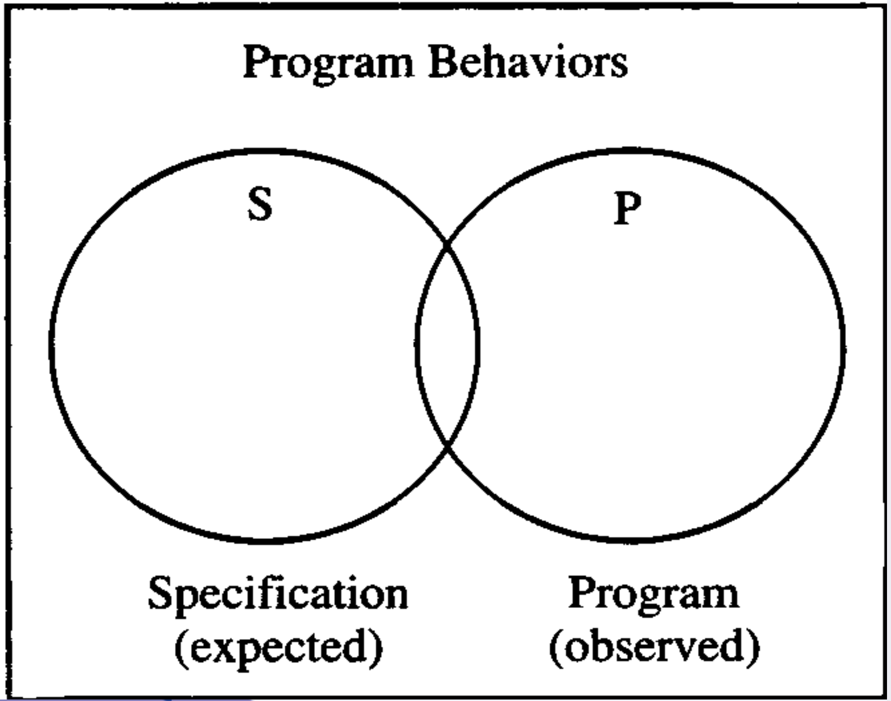
- Software Testing is the most common, widely accepted and practiced method of validating computer software.
- However, just because it is widely practiced does not imply that it is being done any more effectively than other possible validation methods.
- Researchers have been trying to come up with efficient testing tools and techniques.
- Progress is much slower than expected.

Testing Lifecycle



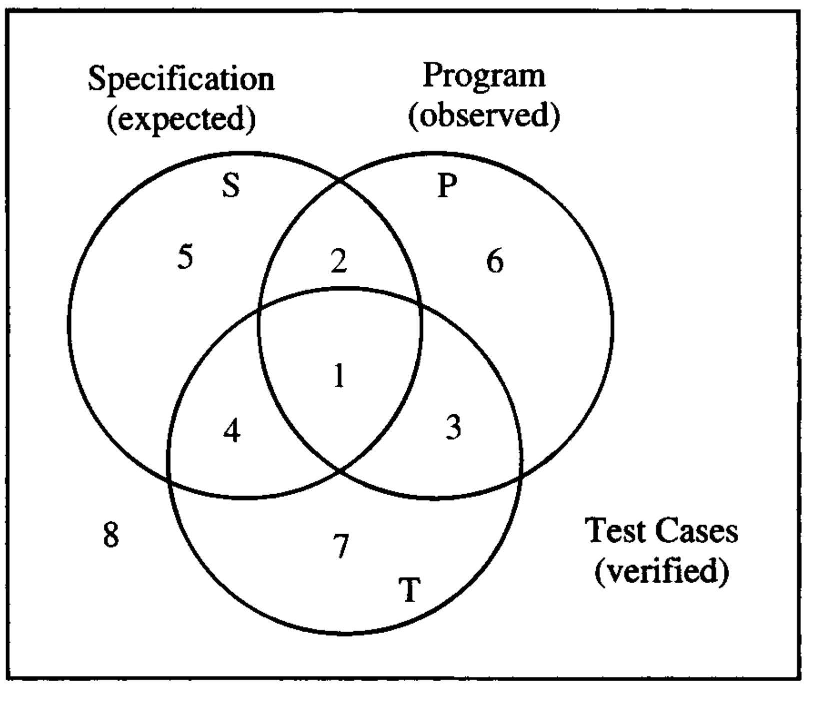
- Software testing goes hand in hand with the software development cycle, as shown above.
- In the first three phases of the development cycle, there are opportunities to introduce errors to the software, resulting in faults that propagate through the remainder of the development process.

Insights From a Venn Diagram



Specified and implemented program behaviors

Insights From a Venn Diagram



Specified, implemented, and tested behaviors

Identifying Test Cases

- There are Two Fundamental Approaches:

1. Functional Testing (Black Box)

2. Structural Testing (White Box)

Major Difficulties in Functional Testing

- **Functional testing creates difficulties in terms of**
  - **GAPS of untested software**
  - **REDUNDANCY in test cases**

- **These problems can be reduced or eliminated in some cases with sophisticated testing methods**

Major Difficulties in Structural Testing

- Structural Testing creates difficulties in terms of
  - Potentially Infinite number of paths (iterations)
  - Infeasible paths (predicates or conditionals)
  - Phenomena like
    - Coincidental Correctness (nature of data/operations)
    - Missing paths (lack of trivial cases)
  - These problems are <u>impossible</u> to detect/eliminate in most cases

Faults classified by severity

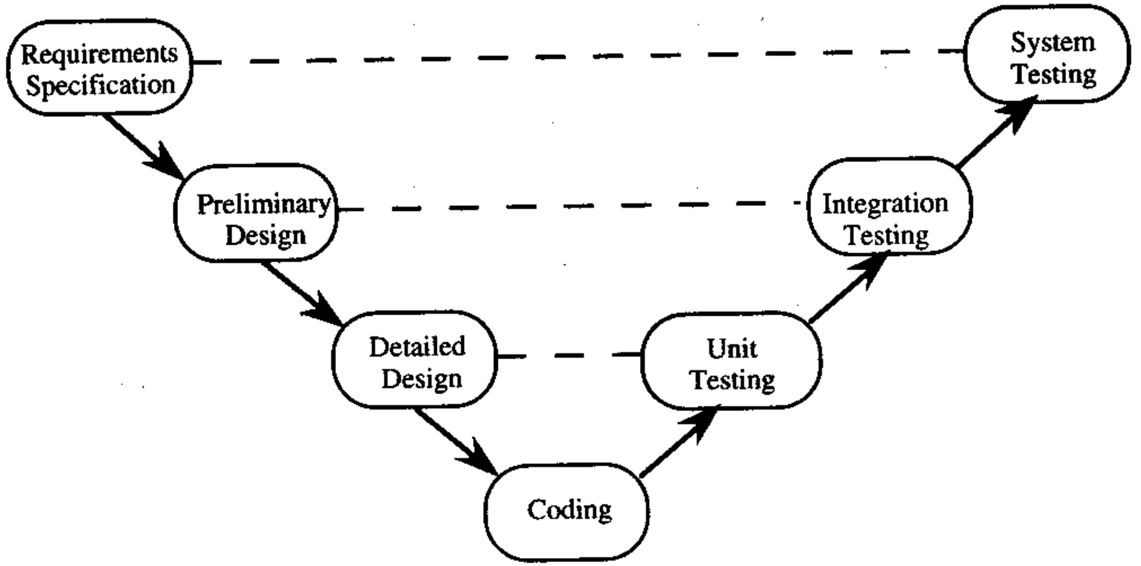
• Mild	Misspelled word
• Moderate	Misleading or redundant information
• Annoying	Truncated names, bill for $0.00
• Disturbing	Some transaction(s) not processed
• Serious	lose a transaction
• Very Serious	Incorrect transaction execution
• Extreme	Frequent “Very Serious” errors
• Intolerable	Database corruption
• Catastrophic	System shutdown
• Infectious	Shutdown that spreads out to others

Fault Classification

- Input and Output Faults
- Logic Faults
- Computation Faults
- Interface Faults
- Data Faults

Note: Within each of these categories of faults, there are many subcategories. So, Faults classification and Faults Isolation becomes very difficult.

Levels of Testing: Correspondence between Testing and Design levels



Levels of abstraction and testing in the Waterfall Model

Should We Be Concerned ?

- Probably YES

- Software Testing, especially large and complex intelligent systems
    - Difficult
    - No single acceptable method
    - 100% testing impossible

Should We Be Encouraged ?

- Of course, YES

- Most of the theory of software testing has been proposed in the areas of
  - Functional
  - Structural

- Also, they are mostly applied to conventional (imperative) languages.

Final Observation

- **Can we achieve 100% Reliability for large and complex intelligent software systems ?**

  The answer is: NO

- **In general, it is impossible to get 100% Reliability, and so the goal is to maximize it as much as we can based on available resources.**

Concluding Comments

- The field of software testing is wide open leading to new research opportunities.
- The research progress in software testing is slower than expected due to many challenges.
- There are always new challenges and exciting opportunities in the field of Computational Intelligence and Innovative Applications:
  - Software Engineering researchers and professionals (university or industry) have always constant opportunities to improve and contribute in the field.
  - Employment opportunities for software engineering professionals, primarily in testing area, will exist forever.

Concluding Comments

- National and international research collaboration opportunities (industry-academic) are highly encouraging.
- Continue to Gain Knowledge, Experience, and Training to Have Fun with
  - Highly RELIABLE and well-engineered, intelligent software systems
- Through active international professional collaboration make the world from Smart to even Smarter….

Thanks for your Attention
And
Enjoy the conference!

IIC2022
ADVANCED TECHNOLOGY AND SUSTAINABLE DEVELOPMENT

The 2nd International Conference on Advanced Technology & Sustainable Development - 2022 (ICATSD 2022)

Robust Control of Uncertain Nonlinear Systems Using Adaptive Regressive Neural-based Deep Learning Technique

PROF. HO PHAM HUY ANH
FEEE, HCM City University of Technology, VNU-HCM

Prof. Dr. Ho Pham Huy Anh

Advanced lecturer,
professor in the FEEE,
HCM City University of Technology, Viet Nam

Prof. Dr. Ho Pham Huy Anh received the B.S. and the M.Sc. degrees from HCM City University of Technology (HCMUT) in 1987 and in 1993, respectively. He received the Ph.D. degree from University of Ulsan, S. Korea in 2008. He is currently an Advanced lecturer, professor in the FEEE, HCM City University of Technology, Viet Nam. Up to now, he authored and co-authored 5 books and published over 180 papers on national and international journals and conference proceedings. His current research interests include intelligent control, robotics, sustainable energy applications, modeling and identification of uncertain MIMO system, computational intelligence.

Content

1. Introduction
2. Preliminaries of Imp-GRNN controller
3. Formulation and Design of Proposed Controller
4. The Weight Update Laws of the proposed controller
5. Stability Analysis of Closed-loop System using proposed controller
6. Simulation Results and Analysis
7. Conclusions

1. Introduction

- From a practical point of view, most of the real-world plants are nonlinear in nature. For instance, the systems in industry, space, power systems, robotics, and autonomous vehicles exhibit nonlinear behaviors.
- In order to deal with those requirements, nowadays, neural controllers have been improvingly applied to control the nonlinear dynamic systems.
- Liu et al. **(Aiqin Liu 2021)** introduced an adaptive neural-based approach for manipulator tracking control.
- Wang et al. **(Wang 2021a)** suggested an adaptive neural network method for controlling the steer-by-wire plants containing disturbance observer.
- Zhang et al. **(Zhang 2021)** introduced an neural-based controller for a specific uncertain dynamic plants.
- Yang et al. **(Yang 2021)** successfully applied a neural-based adaptive controller for deep-space spacecraft formation.
- The disadvantages of those studies related to the fact that their complicated neural-based structure along with strict bounded ranges for input/output variables which dramatically restrict the performance of neural networks controllers in practice.

1. Introduction

- As to improve the performance of neural-based controllers, numerous recent researchers have paid the attention to efficiently combine neural model with advanced control techniques, such as sliding mode, back-stepping, optimal PID, iterative learning control, etc.
- Liu et al. **(Xiao-Fang Liu 2021)** applied the distributed differential evolution (DDE) optimization technique for the neural controller reinforcement learning used in power-electronics scheme control.
- The main drawback of those studies concerns the huge computational burden which hinders their availability in practical applications.
- Recently the integration of neural structure in various advanced machine learning techniques has been increasingly applied.
- Jeyaraj et al. **(Jeyaraj 2022)** recently introduced the real-time data-driven PID controller for multivariable systems using neural-based deep machine learning technique.
- Slama et al. **(Slama 2021)** proposed a tuning neural-based learning method using PSO optimization approach for highly uncertain plants.
- Ait Abbas **(Ait Abbas 2021)** introduced an adaptive deep neural–based learning technique based on sparse auto-encoder for the antilock braking plant subject to high restrictions.
- The main drawback of these neural–based machine learning techniques focused on the fact that those neural-based learning methods efficiently operate mostly in offline identification.

1. Introduction

- With deep learning technique, the hidden weighting values are initialized in random and hold unchanged during the training procedure regardless iteratively tuning.
- Theoretical researches have demonstrated that although randomly initiated hidden weighting values, deep learning-based neural model ensures the out-performing universal approximating ability.
- The main drawback of such deep learning methods concerns the fact that deep learning-based identification process was only offline estimated.
- Moreover the neural-based deep learning structure may still present a slow convergence if initiated values for the hidden layer nodes are not properly selected.
- Furthermore, another drawback is that the system stability requirement has not yet satisfactorily resolved.
- Eventually the deep learning-based neural model shows hard for allowing an efficient adaptation of the identifier in the presence of significant internal/external disturbances.

1. Introduction

- As to overcome these disadvantages, recent researchers have improved novel neural-based machine learning techniques which have effectively applied in online operation to identify and control of highly nonlinear MIMO dynamic systems.
- Wang et al. **(Wang 2022)** proposed a new neural-based real-time cascading deep learning technique in order to diagnose anomalies based on online data pools.
- Hu et al. **(Hu 2021)** introduced an online COVID-19 diagnosis from X-Ray images using deep CNN for extreme learning machines.
- Yu et al. **(Helong Yu 2021)** suggested an enhanced butterfly optimizer-configured neural-based extreme learning machine for fault diagnosis.
- Ye et al. **(Ye 2021)** successfully used the neural-based online deep learning for person re-identification, among them.

1. Introduction

- Inspired from the results above-mentioned, in this research, an improved global regressive neural GRNN model, namely Imp-GRNN, is suggested to handle the multivariable uncertain dynamic plants.
- The Imp-GRNN retains the original characteristics of GRNN, and it provides an improved development to the precision compared of GRNN as an online controller.
- The Imp-GRNN is enhanced in various key aspects.
- Firstly, a new method is developed to adapt the input-hidden weighting values of GRNN based on the regressive stochastic tools of the inputs.
- Secondly, an output layer is newly introduced. Also, novel adaptive weighting values between input to output layer are introduced.
- Besides, a new smoothen coefficient is proposed to eradicate the necessity for choosing it in precedent. Moreover, the stability of the controller is demonstrated via Lyapunov concept applied for discrete-time plants.
- Eventually the superiority of the controller is verified via various benchmark tests and in comparison to the original GRNN to confirm its performance.

2. Preliminaries of Imp-GRNN controller

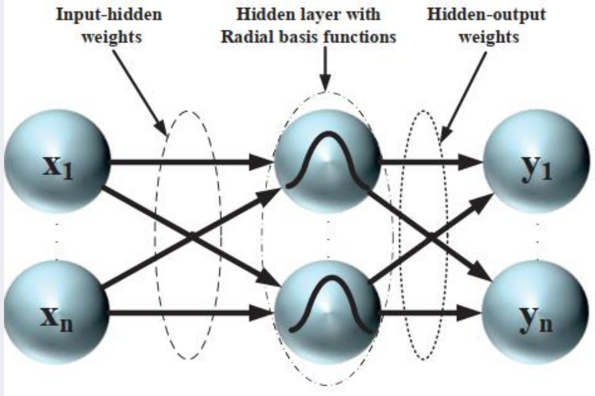


Figure 1: Structure of GRNN

$$E[y \mid X] = \hat{Y}(X) = \frac{\int_{-\infty}^{\infty} y f(X,y)dy}{\int_{-\infty}^{\infty} f(X,y)dy}$$

$$\hat{f}(X,Y) = \frac{1}{2\pi^{(p+1)/2}\sigma^{(p+1)}} \frac{1}{2}\sum_{i=1}^{n} e^{-\frac{(X-X^i)^T(X-X^i)}{2\sigma^2}} e^{-\frac{(Y-Y^i)^2}{2\sigma^2}}$$

$$\hat{Y}(X) = \frac{\sum_{i=1}^{n} y_i e^{-\frac{(X-X^i)^T(X-X^i)}{2\sigma^2}}}{\sum_{i=1}^{n} e^{-\frac{(X-X^i)^T(X-X^i)}{2\sigma^2}}}$$

Let D_i^2 be defined as $D_i^2 = \left(X - X^i\right)^T \left(X - X^i\right)$

$$\hat{Y}(X) = \frac{\sum_{i=1}^{n} y_i e^{\frac{-D_i^2}{2\sigma^2}}}{\sum_{i=1}^{n} e^{\frac{-D_i^2}{2\sigma^2}}}$$

As the basic GRNN is a single-pass learning network, its training process is different from the typical algorithm of the back-propagation of the error. The GRNN is trained as follows:

* Input-hidden weights are selected as the input data samples.
* Hidden-output weighting values will be selected as the target data samples.
* is selected before the training process, and it should be positive and greater than zero.

For every distinct training sample, a new node is added. In this training process, GRNN record every distinct data sample that will be used later for calculating its outputs for similar data samples.

2. Preliminaries of Imp-GRNN controller

Proposed Imp GRNN control method

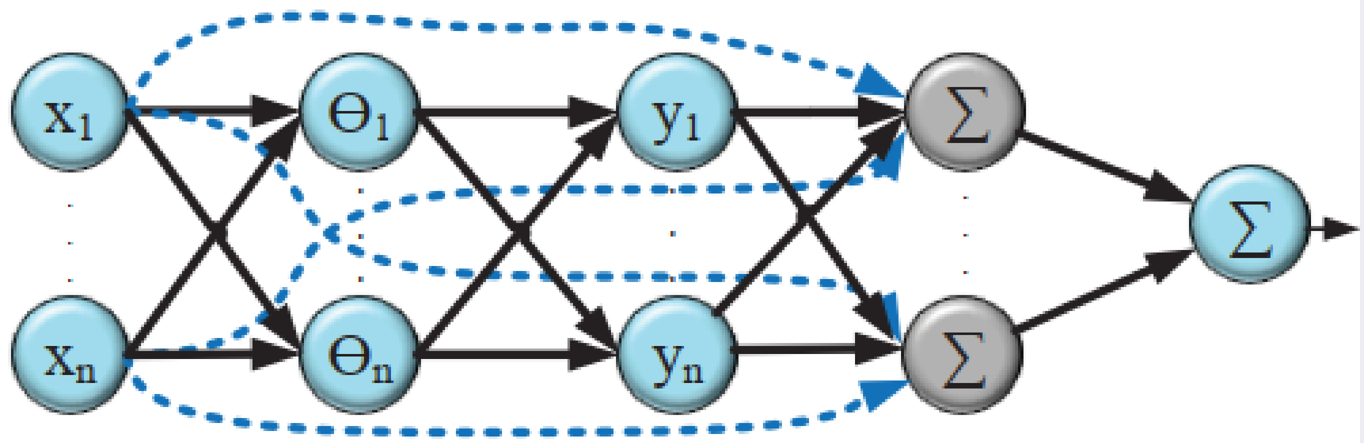


Figure 2: Imp-GRNN with the dotted lines represents the novel suggested weighting values and the gray neurons represent the novel proposed output ones.

2. Preliminaries of Imp-GRNN controller

a. ***Input layer***: The input variables x_i will be considered. Given N input variables x_i provided to the neural model, L represents the number of hidden nodes, S denotes the amount of output variables of the network and C represents the scope of the interval of the smoothing parameter σ. The input-hidden weighting matrix $M \in \mathbb{R}^{L\times N}$ contains the statistical means of the current inputs as follows:

$$M = \begin{pmatrix} \mu_{11}(k) & \cdots & \mu_{1N}(k) \\ \vdots & \ddots & \vdots \\ \mu_{L1}(k) & \cdots & \mu_{LN}(k) \end{pmatrix} \qquad (11)$$

The Euclidean vector $D \in [0,\infty)$ represents the distance of the present inputs vector $P(k)$ $\in \mathbb{R}^N$ with the present matrix M is determined as:

$$D(k) = d\left(M, P(k)\right) \qquad (12)$$

The average value of each input is re-calculated regarding to the novel input based on (13):

$$\mu_i(k) = \frac{(k-1)}{k}\mu_i(k-1) + \frac{1}{k}p_i(k) \qquad (13)$$

$i = 1,\ldots,N$ and $k = 1,2,\ldots$

2. Preliminaries of Imp-GRNN controller

a. ***Input layer***: The input variables x_i will be considered. Given N input variables x_i provided to the neural model, L represents the number of hidden nodes, S denotes the amount of output variables of the network and C represents the scope of the interval of the smoothing parameter σ. The input-hidden weighting matrix $M \in \mathbb{R}^{L\times N}$ contains the statistical means of the current inputs as follows:

$$M = \begin{pmatrix} \mu_{11}(k) & \dots & \mu_{1N}(k) \\ \vdots & \ddots & \vdots \\ \mu_{L1}(k) & \cdots & \mu_{LN}(k) \end{pmatrix} \qquad (11)$$

The Euclidean vector $D \in [0,\infty)$ represents the distance of the present inputs vector $P(k) \in \mathbb{R}^N$ with the present matrix M is determined as:

$$D(k) = d\left(M, P\left(k\right)\right) \qquad (12)$$

The average value of each input is re-calculated regarding to the novel input based on (13):

$$\mu_i(k) = \frac{(k-1)}{k}\mu_i(k-1) + \frac{1}{k}p_i(k) \qquad (13)$$

$i = 1,\dots,N$ and $k = 1,2,\dots$

2. Preliminaries of Imp-GRNN controller

a. ***Input layer***: The input variables x_i will be considered. Given N input variables x_i provided to the neural model, L represents the number of hidden nodes, S denotes the amount of output variables of the network and C represents the scope of the interval of the smoothing parameter σ. The input-hidden weighting matrix $M \in \mathbb{R}^{L\times N}$ contains the statistical means of the current inputs as follows:

$$M = \begin{pmatrix} \mu_{11}(k) & \cdots & \mu_{1N}(k) \\ \vdots & \ddots & \vdots \\ \mu_{L1}(k) & \cdots & \mu_{LN}(k) \end{pmatrix} \qquad (11)$$

The Euclidean vector $D \in [0,\infty)$ represents the distance of the present inputs vector $P(k) \in \mathbb{R}^N$ with the present matrix M is determined as:

$$D(k) = d\left(M, P(k)\right) \qquad (12)$$

The average value of each input is re-calculated regarding to the novel input based on (13):

$$\mu_i(k) = \frac{(k-1)}{k}\mu_i(k-1) + \frac{1}{k}p_i(k) \qquad (13)$$

$i = 1,\ldots,N$ and $k = 1,2,\ldots$

2. Preliminaries of Imp-GRNN controller

b. ***The 1st hidden layer***: In this layer, the fired vectors of the outputs of the hidden layer $\theta(k) \in (0, \infty)$ for every element of the interval $\sigma \in (0, \infty)$ are calculated. The size of these vectors combined together is $C \times L$. The following equation is used to calculate these vectors:

$$\theta_j(k) = e^{\frac{-D^2(k)}{2\sigma_j^2}}, j = 1, \ldots, C \quad (14)$$

The stochastic means of the fired vectors denoted by $\Theta(k) \in (0, \infty)$, is calculated as follows:

$$\Theta(k) = \frac{\sum_{j=1}^{c} \theta_j(k)}{C} \quad (15)$$

c. ***The second hidden layer***: In this layer, the statistical means of the firing vectors from the previous step ($\Theta(k)$) is multiplied by the vector of the weights $\Phi(k) \in \mathbb{R}^{S \times L}$ and divided by the summation of $\Theta(k)$ to produce $U_1(k)$ which is the output of the Imp GRNN before the new layer:

$$U_1(k) = \frac{\Phi(k)^T \Theta(k)}{\sum_{i=1}^{N} \Theta(k)_i} \tag{16}$$

Now we let $\Theta_n(k)$ described as (17):

$$\Theta_n(k) = \frac{\Theta(k)}{\sum_{i=1}^{N} \Theta(k)_i} \tag{17}$$

Based on (17), (16) can be rewritten as follows:

$$U_1(k) = \Phi(k)^T \Theta_n(k) \tag{18}$$

2. Preliminaries of Imp-GRNN controller

d. ***The output layer***: Here, the input-output weighting values $\Psi \in \mathbb{R}^{S \times N}$ will be suggested in which the eventual output of proposed controller is determined as (19):

$$\begin{aligned} U_f(k) &= \Psi^T(k)P(k) + U_1(k) \\ U_f(k) &= \Psi^T(k)P(k) + \Phi^T(k)\Theta_n(k) \end{aligned} \qquad (19)$$

3. Formulation and Design of Proposed Imp-GRNN Controller

Investigate a multivariable nonlinear discrete-time plant described in (20):

$$X(k+1) = F(X(k), U(k)) + G(X(k))U(k)$$

$$y_i(k) = x_{1i}(k)$$

$$U(k) = [u_1(k), \ldots, u_n(k)]$$

$$X = \begin{bmatrix} x_{1,1}, \ldots, x_{1,n} \\ x_{2,1}, \ldots, x_{2,n} \\ \ldots \\ x_{n,1}, \ldots, x_{n,n} \end{bmatrix} \qquad (20)$$

The tracking erroneous term $\tilde{y}(k) \in \mathbb{R}^n$ will be defined:

$$\tilde{y}(k) = y_d(k) - y(k) \qquad (21)$$

3. Formulation and Design of Proposed Imp-GRNN Controller

Consider a suggested Imp-GRNN controller containing L hidden nodes, N denotes the amount of inputs, S denotes the amount of outputs. The proposed controller output will be described in (22):

$$U_{ImpGRNN}(k) = \Psi^T(k)P(k) + \Phi^T(k)\Theta_n(k) \qquad (22)$$

where $\Psi \in \mathbb{R}^{S\times N}$ is the matrix of the novel suggested weighting values, $P(k) \in \mathbb{R}^N$ represents the inputs, $\Phi(k) \in \mathbb{R}^{S\times L}$ and $\Theta_n(k) \in (0,\infty)$ of size $1\times L$.

The Imp GRNN focuses on forcing the plant outputs $y(t)$ to precisely track the ideal $y_d(t)$

Assumption 1: Let us assume there are weighting matrix Ψ^* & Φ^* which helps the proposed control accurately track the desired control vector $U_d(k)$ which guide the plant outputs $y(k)$ to the ideal $y_d(k)$ regarding to the erroneous value $\varepsilon(k)$ described in (23):

$$U_d(k) = \Psi^{*T}P(k) + \Phi^*(k)\Theta_n(k) + \varepsilon(k) \qquad (23)$$

As the term $\tilde{U}(k) = U_d(k) - U(k)$ is unavailable because of $U_d(k)$ often uninformed, it could be linked to the tracking erroneous term using the following assumption:

Assumption 2: The defined term $\tilde{U}(k) = U_d(k) - U(k)$ is less than or equal to the output erroneous term $\tilde{y}(k)$ as follows:

$$\tilde{U}(k) \le \tilde{y}(k) \qquad (24)$$

3. Formulation and Design of Proposed Imp-GRNN Controller

Remark 1: Assumption 2 is essential for the design of the controller. It is utilized to design the update laws of the weights, as depicted in (41). As the target control is not given to the proposed controller, whose quality can be measured by the accuracy of tracking the desired trajectories. Thus, the weighting values of proposed controller will be adapted using the tracking error.

Assumption 3: Assume that the output of the first hidden layer in the Imp-GRNN is limited as (25):

$$\left\|\Theta_n(k)\right\| \le \xi \qquad (25)$$

with $\xi \in [0, L]$ for the Imp-GRNN, Θ_n(k) denotes the standardized output of the first hidden layer of Imp-GRNN (see (17)) and L is the number of the nodes in the first hidden layer of the Imp-GRNN.

3. Formulation and Design of Proposed Imp-GRNN Controller

Lemma 1: For a given vector a, the following relation is applied:

$$a^T a = \|a\|^2 \tag{26}$$

Lemma 2: The tracking erroneous value $\tilde{y}(k+1)$ is determined as:

$$\tilde{y}_i(k+1) \leq \tilde{y}_i(k) - \beta_i \tag{26a}$$

with $\tilde{y}(k) = y_d(k) - y(k)$ represents the deviation, $i = 1, \ldots, n$ represents the amount of the plant outputs, β denotes an unchanged value.

The equation (26a) under vector form is described as:

$$\tilde{y}(k+1) \leq \tilde{y}(k) - B$$

where $\tilde{y} = [\tilde{y}_1, \ldots, \tilde{y}_n]^T$, $B = [\beta_1, \ldots, \beta_n]^T$

3. Formulation and Design of Proposed Imp-GRNN Controller

Proof:

The deviation $\Delta\tilde{y}_i(k)$ of $\tilde{y}_i(k)$ is described as:

$$\Delta\tilde{y}_i(k) = \tilde{y}_i(k+1) - \tilde{y}_i(k) \qquad (27)$$

with i = 1,…, n represents the amount of the plant outputs. In case (27) is re-edited, $\tilde{y}_i(k+1)$ is expressed as:

$$\tilde{y}_i(k+1) = \tilde{y}_i(k) - \Delta\tilde{y}_i(k) \qquad (28)$$

Since $\tilde{y}_i(k)$ is able to be described in series over time, $\tilde{y}_i(k+1)$ will be described with respect to $\tilde{y}_i(k)$ as:

$$\tilde{y}_i(k+1) = f(\tilde{y}_i(k)) \qquad (29)$$

In which $f(.)$ represents a function which is approximately used an enable regressive technique.

In case a linear approximating function is used to evaluate $f(.)$ based on $\tilde{y}(k)$, hence (30) is derived:

$$\tilde{y}_i(k+1) = \tilde{y}_i(k) - \eta_i \qquad (30)$$

with η_i represents an approximated error limited within $\|\eta_i\| \le \beta_i$ and $\beta_i \ge 0$.

3. Formulation and Design of Proposed Imp-GRNN Controller

Replacing (30) to (28), regarding $\|\eta_i\| \le \beta_i$, that leads,

$$\Delta\tilde{y}_i(k) \le -\beta_i \qquad (31)$$

Replacing (31) to (28), it leads to:

$$\tilde{y}_i(k+1) \le \tilde{y}_i(k) - \beta_i \qquad (32)$$

Then the approximating expression of $U(k)$ via proposed controller is described as:

$$U(k) = \hat{\Psi}^T(k)P(k) + \hat{\Phi}^T(k)\Theta_n(k) \qquad (33)$$

Thus the control erroneous value $\tilde{U}(k)$ will be expressed in (34):

$$\tilde{U}(k) = \Psi^{*T}P(k) + \Phi^{*T}\Theta_n(k) + \varepsilon(k) - \hat{\Psi}^T(k)P(k) - \hat{\Phi}^T(k)\Theta_n(k) \qquad (34)$$

The weight erroneous vectors $\tilde{\Phi}(k), \tilde{\Psi}(k)$ are defined in (35):

$$\tilde{\Phi}(k) = \Phi^* - \hat{\Phi}(k)$$

$$\tilde{\Psi}(k) = \Psi^* - \hat{\Psi}(k) \qquad (35)$$

Based on (35), (34) can be rewritten as:

$$\tilde{U}(k) = \tilde{\Psi}^T(k)P(k) + \tilde{\Phi}^T(k)\Theta_n(k) + \varepsilon(k) \qquad (36)$$

3. Formulation and Design of Proposed Imp-GRNN Controller

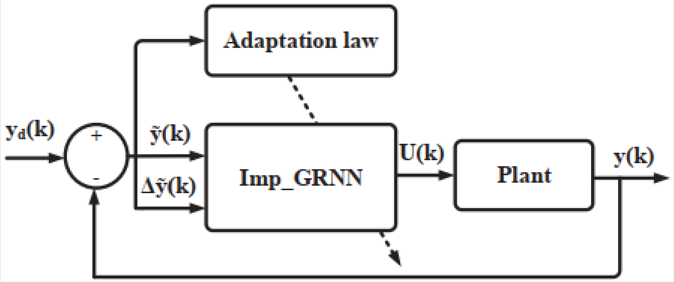


Figure 3a: Block-diagram of proposed Imp-GRNN algorithm

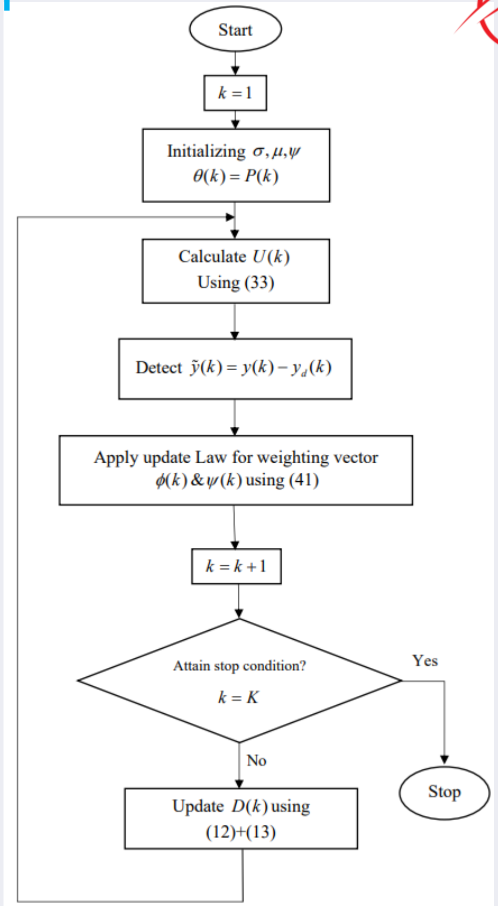


Figure 3b: Flowchart of proposed Imp-GRNN algorithm

4. The Weight Update Laws of the proposed controller

Here, the new update laws of proposed controller are derived. In order to implement an optimum adaptable NN-based tracking algorithm for plant (20), the weighting values of proposed controller will be optimally updated in term of an objective function $J(k)$ which is expressed as:

$$J(k) = \frac{1}{2}\left(\tilde{U}(k)^T \tilde{U}(k)\right) \qquad (37)$$

Thus, using gradient descent GD approach, the followed updated rules of proposed controller for weighting values are implemented

$$\begin{aligned} \Phi(k+1) &= \Phi(k) + \gamma_1 \frac{\Delta J}{\Delta \Phi(k)} \\ \Psi(k+1) &= \Psi(k) + \gamma_2 \frac{\Delta J}{\Delta \Phi(k)} \end{aligned} \qquad (38)$$

4. The Weight Update Laws of the proposed controller

Here, the new update laws of proposed controller are derived. In order to implement an optimum adaptable NN-based tracking algorithm for plant (20), the weighting values of proposed controller will be optimally updated in term of an objective function $J(k)$ which is expressed as:

$$J(k) = \frac{1}{2}\left(\tilde{U}(k)^T \tilde{U}(k)\right) \tag{37}$$

Thus, using gradient descent GD approach, the followed updated rules of proposed controller for weighting values are implemented

$$\begin{aligned} \Phi(k+1) &= \Phi(k) + \gamma_1 \frac{\Delta J}{\Delta \Phi(k)} \\ \Psi(k+1) &= \Psi(k) + \gamma_2 \frac{\Delta J}{\Delta \Phi(k)} \end{aligned} \tag{38}$$

4. The Weight Update Laws of the proposed controller

To determine $\frac{\Delta J(k)}{\Delta \Phi(k)}$ and $\frac{\Delta J(k)}{\Delta \Psi(k)}$, the chain rule is applied to the cost function $J(k)$, it gives:

$$\begin{aligned}\frac{\Delta J(k)}{\Delta \Phi(k)} &= \frac{\Delta J(k)}{\Delta U(k)} \frac{\Delta U(k)}{\Delta \Phi(k)} \\ &= -\tilde{U}(k)\Theta_n(k)\end{aligned} \tag{39}$$

$$\begin{aligned}\frac{\Delta J(k)}{\Delta \Psi(k)} &= \frac{\Delta J(k)}{\Delta U(k)} \frac{\Delta U(k)}{\Delta \Psi(k)} \\ &= -\tilde{U}(k)P(k)\end{aligned} \tag{40}$$

By incorporating Assumption 2, the final update laws are formulated as:

$$\begin{aligned}\Phi(k+1) &\leq \Phi(k) - \gamma_1 \tilde{y}(k)\Theta_n(k) \\ \Psi(k+1) &\leq \Psi(k) - \gamma_2 \tilde{y}(k)P(k)\end{aligned} \tag{41}$$

5. Stability Analysis of Closed-loop System using proposed controller

Here, the system stability will be investigated based on Lyapunov principle. Consider a candidate $V(k)$ described as

$$V(k) = \sum_{i=1}^{3} V_i(k) \tag{42}$$

In which the 1st candidate $V_1(k)$ is expressed as:

$$V_1(k) = \tilde{y}(k)^T \tilde{y}(k) \tag{43}$$

The derivation of V_1 is determined in (44):

$$\Delta V_1(k) = \tilde{y}(k+1)^T \tilde{y}(k+1) - \tilde{y}(k)^T \tilde{y}(k) \tag{44}$$

Via combining Lemma 2 with inequality Schwarz presented in **(Bresch-Pietri 2012)**, the following can be written:

$$\Delta V_1(k) \le \left(\tilde{y}(k) - B\right)^T \left(\tilde{y}(k) - B\right) - \tilde{y}(k)^T \tilde{y}(k) \tag{45}$$

$$\begin{aligned} \Delta V_1(k) &\le 2\tilde{y}^T(k)\tilde{y}(k) - 2B^T B - \tilde{y}(k)^T \tilde{y}(k) \\ &\le \tilde{y}^T(k)\tilde{y}(k) - 2B^T B \end{aligned} \tag{46}$$

$$\Delta V_1(k) \le \left\|\tilde{y}(k)\right\|^2 - 2\left\|B\right\|^2 \tag{47}$$

5. Stability Analysis of Closed-loop System using proposed controller

IIC 2022 ADVANCED TECHNOLOGY AND SUSTAINABLE DEVELOPMENT

Here, the system stability will be investigated based on Lyapunov principle. Consider a candidate $V(k)$ described as

$$V(k) = \sum_{i=1}^{3} V_i(k) \qquad (42)$$

In which the 1st candidate $V_1(k)$ is expressed as:

$$V_1(k) = \tilde{y}(k)^T \tilde{y}(k) \qquad (43)$$

The derivation of V_1 is determined in (44):

$$\Delta V_1(k) = \tilde{y}(k+1)^T \tilde{y}(k+1) - \tilde{y}(k)^T \tilde{y}(k) \qquad (44)$$

Via combining Lemma 2 with inequality Schwarz presented in **(Bresch-Pietri 2012)**, the following can be written:

$$\Delta V_1(k) \le \left(\tilde{y}(k) - B\right)^T \left(\tilde{y}(k) - B\right) - \tilde{y}(k)^T \tilde{y}(k) \qquad (45)$$

$$\begin{aligned} \Delta V_1(k) &\le 2\tilde{y}^T(k)\tilde{y}(k) - 2B^T B - \tilde{y}(k)^T \tilde{y}(k) \\ &\le \tilde{y}^T(k)\tilde{y}(k) - 2B^T B \end{aligned} \qquad (46)$$

$$\Delta V_1(k) \le \left\|\tilde{y}(k)\right\|^2 - 2\left\|B\right\|^2 \qquad (47)$$

5. Stability Analysis of Closed-loop System using proposed controller

Here, the system stability will be investigated based on Lyapunov principle. Consider a candidate $V(k)$ described as

$$V(k) = \sum_{i=1}^{3} V_i(k) \tag{42}$$

In which the 1st candidate $V_1(k)$ is expressed as:

$$V_1(k) = \tilde{y}(k)^T \tilde{y}(k) \tag{43}$$

The derivation of V_1 is determined in (44):

$$\Delta V_1(k) = \tilde{y}(k+1)^T \tilde{y}(k+1) - \tilde{y}(k)^T \tilde{y}(k) \tag{44}$$

Via combining Lemma 2 with inequality Schwarz presented in **(Bresch-Pietri 2012)**, the following can be written:

$$\Delta V_1(k) \le \left(\tilde{y}(k) - B\right)^T \left(\tilde{y}(k) - B\right) - \tilde{y}(k)^T \tilde{y}(k) \tag{45}$$

$$\begin{aligned} \Delta V_1(k) &\le 2\tilde{y}^T(k)\tilde{y}(k) - 2B^T B - \tilde{y}(k)^T \tilde{y}(k) \\ &\le \tilde{y}^T(k)\tilde{y}(k) - 2B^T B \end{aligned} \tag{46}$$

$$\Delta V_1(k) \le \left\|\tilde{y}(k)\right\|^2 - 2\left\|B\right\|^2 \tag{47}$$

5. Stability Analysis of Closed-loop System using proposed controller

IIC2022 ADVANCED TECHNOLOGY AND SUSTAINABLE DEVELOPMENT

Here, the system stability will be investigated based on Lyapunov principle. Consider a candidate $V(k)$ described as

$$V(k)=\sum_{i=1}^{3}V_i(k) \qquad (42)$$

In which the 1st candidate $V_1(k)$ is expressed as:

$$V_1(k)=\tilde{y}(k)^T\tilde{y}(k) \qquad (43)$$

The derivation of V_1 is determined in (44):

$$\Delta V_1(k)=\tilde{y}(k+1)^T\tilde{y}(k+1)-\tilde{y}(k)^T\tilde{y}(k) \qquad (44)$$

Via combining Lemma 2 with inequality Schwarz presented in **(Bresch-Pietri 2012)**, the following can be written:

$$\Delta V_1(k)\le\left(\tilde{y}(k)-B\right)^T\left(\tilde{y}(k)-B\right)-\tilde{y}(k)^T\tilde{y}(k) \qquad (45)$$

$$\begin{aligned}\Delta V_1(k)&\le 2\tilde{y}^T(k)\tilde{y}(k)-2B^TB-\tilde{y}(k)^T\tilde{y}(k)\\ &\le \tilde{y}^T(k)\tilde{y}(k)-2B^TB\end{aligned} \qquad (46)$$

$$\Delta V_1(k)\le\left\|\tilde{y}(k)\right\|^2-2\left\|B\right\|^2 \qquad (47)$$

5. Stability Analysis of Closed-loop System using proposed controller

Here, the system stability will be investigated based on Lyapunov principle. Consider a candidate $V(k)$ described as

$$V(k) = \sum_{i=1}^{3} V_i(k) \tag{42}$$

In which the 1st candidate $V_1(k)$ is expressed as:

$$V_1(k) = \tilde{y}(k)^T \tilde{y}(k) \tag{43}$$

The derivation of V_1 is determined in (44):

$$\Delta V_1(k) = \tilde{y}(k+1)^T \tilde{y}(k+1) - \tilde{y}(k)^T \tilde{y}(k) \tag{44}$$

Via combining Lemma 2 with inequality Schwarz presented in **(Bresch-Pietri 2012)**, the following can be written:

$$\Delta V_1(k) \le \left(\tilde{y}(k) - B\right)^T \left(\tilde{y}(k) - B\right) - \tilde{y}(k)^T \tilde{y}(k) \tag{45}$$

$$\begin{aligned} \Delta V_1(k) &\le 2\tilde{y}^T(k)\tilde{y}(k) - 2B^T B - \tilde{y}(k)^T \tilde{y}(k) \\ &\le \tilde{y}^T(k)\tilde{y}(k) - 2B^T B \end{aligned} \tag{46}$$

$$\Delta V_1(k) \le \|\tilde{y}(k)\|^2 - 2\|B\|^2 \tag{47}$$

5. Stability Analysis of Closed-loop System using proposed controller

The function V_1 seems stable in case $\|\tilde{y}(k)\|$ within the set Ω_1, which is described as:

$$\Omega_1 = \|\tilde{y}(k)\| \le \pm\sqrt{2}\|B\| \qquad (48)$$

Then the 2nd candidate V_2 is chosen as:

$$V_2(k) = \tilde{\Phi}(k)^T\tilde{\Phi}(k) \qquad (49)$$

The first difference of V_2 is defined as:

$$\Delta V_2(k) = \tilde{\Phi}(k+1)^T\tilde{\Phi}(k+1) - \tilde{\Phi}(k)^T\tilde{\Phi}(k) \qquad (50)$$

Simplifying (50) further and substituting (41), it leads to:

$$\Delta V_2(k) \le \left(\tilde{\Phi}(k) - \gamma_1\tilde{y}(k)\Theta_n(k)\right)\left(\tilde{\Phi}(k) - \gamma_1\tilde{y}(k)\Theta_n(k)\right) - \tilde{\Phi}(k)^T\tilde{\Phi}(k) \quad (51)$$

By incorporating the inequality of Cauchy-Schwarz **(Bresch-Pietri 2012)**, (51) can be rewritten as:

$$\Delta V_2(k) \le 2\left(\tilde{\Phi}^T(k)\tilde{\Phi}(k) - \left(\gamma_1\tilde{y}(k)\Theta_n(k)\right)^T\left(\gamma_1\tilde{y}(k)\Theta_n(k)\right)\right) - \tilde{\Phi}(k)^T\tilde{\Phi}(k) \qquad (52)$$

Simplifying (52) further and incorporating *Assumption 3,* it leads to:

$$\Delta V_2(k) \le \|\tilde{\Phi}(k)\|^2 - 2\gamma_1^2\xi^2\|\tilde{y}(k)\|^2 \qquad (53)$$

5. Stability Analysis of Closed-loop System using proposed controller

$V_2(k)$ proves stable in case γ_1 within the set Ω_2 which is described in (54):

$$\Omega_2 = \gamma_2 \geq \pm \frac{\left\|\tilde{\Phi}(k)\right\|}{\sqrt{2\xi}\left\|\tilde{y}(k)\right\|} \tag{54}$$

Combining Assumption 2, (54) is also expressed as:

$$\Omega_2 = \gamma_1 \geq \pm \frac{\left\|\tilde{\Phi}(k)\right\|}{\sqrt{2\xi}\left\|\tilde{U}(k)\right\|} \tag{55}$$

The third Lyapunov's function $V_3(k)$ is selected as:

$$V_3(k) = \tilde{\Psi}(k)^T \tilde{\Psi}(k) \tag{56}$$

The first difference of V_3 is written as:

$$\Delta V_3(k) = \tilde{\Psi}(k+1)^T \tilde{\Psi}(k+1) - \tilde{\Psi}(k)^T \tilde{\Psi}(k) \tag{57}$$

Simplifying (57) further and substituting (41) leads to:

$$\Delta V_3(k) \leq \left(\tilde{\Psi}(k) - \gamma_2 \tilde{y}(k) P_n(k)\right)\left(\tilde{\Psi}(k) - \gamma_2 \tilde{y}(k) P(k)\right) - \tilde{\Psi}(k)^T \tilde{\Psi}(k) \tag{58}$$

5. Stability Analysis of Closed-loop System using proposed controller

By incorporating the inequality of Cauchy-Schwarz **(Bresch-Pietri 2012)** and simplifying further, (58) will become:

$$\Delta V_3(k) \leq \left\|\tilde{\Psi}(k)\right\|^2 - 2\gamma_2^2 \left\|\tilde{y}(k)P(k)\right\|^2 \qquad (59)$$

V_3 (k) shows stable in case γ_2 was within the set Ω_3, which is described in (60):

$$\Omega_3 = \gamma_2 \geq \pm \frac{\left\|\tilde{\Psi}(k)\right\|}{\sqrt{2}\left\|\tilde{y}(k)P(k)\right\|} \qquad (60)$$

By incorporating Assumption 2, (60) becomes:

$$\Omega_3 = \gamma_2 \geq \pm \frac{\left\|\tilde{\Psi}(k)\right\|}{\sqrt{2}\left\|\tilde{U}(k)P(k)\right\|} \qquad (61)$$

Using the stability demonstration above-mentioned, the requirements in (48), (55) and (61) will be completely responded, the system stability is proven.

Benchmark Test

We investigate a MIMO non-affine discrete nonlinear system **(Ge 2004)**.

The system dynamics is expressed as:

$$x_{1,1}(k+1) = f_{1,1}(x_{1,1}(k)) + g_{1,1}(x_{1,1}(k))$$
$$x_{1,2}(k+1) = f_{1,2}(x_{1,2}(k)) + g_{1,2}(x_{1,1}(k))u_1(k) + d_1(k)$$
$$x_{2,1}(k+1) = f_{2,1}(x_{2,1}(k)) + g_{2,1}(x_{2,2}(k))$$
$$x_{2,2}(k+1) = f_{2,2}(x_{2,2}(k), u_1(k)) + g_{2,2}(x_{2,2}(k))u_2(k) + d_2(k)$$

$$y_1(k+1) = x_{1,1}(k)$$
$$y_2(k+1) = x_{2,1}(k)$$

The initial states are $[x_{1,1}(0), x_{1,2}(0), x_{2,1}(0), x_{2,2}(0)] = [0,0,0,0]$.

$$f_{1,1}(x_{1,1}(k)) = \frac{x_{1,1}^2(k)}{1 + x_{1,1}^2(k)}$$
$$g_{1,1}(x_{1,1}(k)) = 0.3$$
$$f_{1,2}(x_{1,2}(k)) = \frac{x_{1,1}^2(k)}{1 + x_{1,2}^2(k) + x_{2,1}^2(k) + x_{2,2}^2(k)}$$
$$g_{1,2}(x_{1,1}(k)) = 1$$
$$d_1(k) = 0.1\cos(0.05k)\cos(x_{1,1}(k))$$
$$f_{2,1}(x_{2,1}(k)) = \frac{x_{2,1}^2(k)}{1 + x_{2,1}^2(k)}$$
$$g_{2,1}(x_{2,2}(k)) = 0.2$$
$$f_{2,2}(x_{2,2}(k), u_2(k)) = \frac{x_{2,1}^2(k)u_2^2(k)}{1 + x_{1,1}^2(k) + x_{2,1}^2(k) + x_{2,2}^2(k)}$$
$$g_{2,2}(x_{2,2}(k)) = 1$$
$$d_2(k) = 0.1\cos(0.05k)\cos(x_{2,1}(k))$$

6. Simulation Results and Analysis

The Results of Step Input Tracking

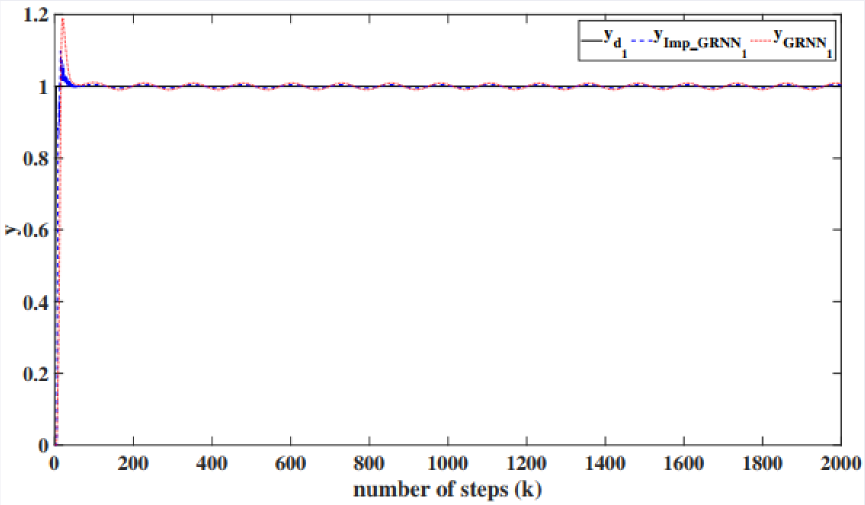


Figure 4: Tracking results for output one benchmark test 1 with step input

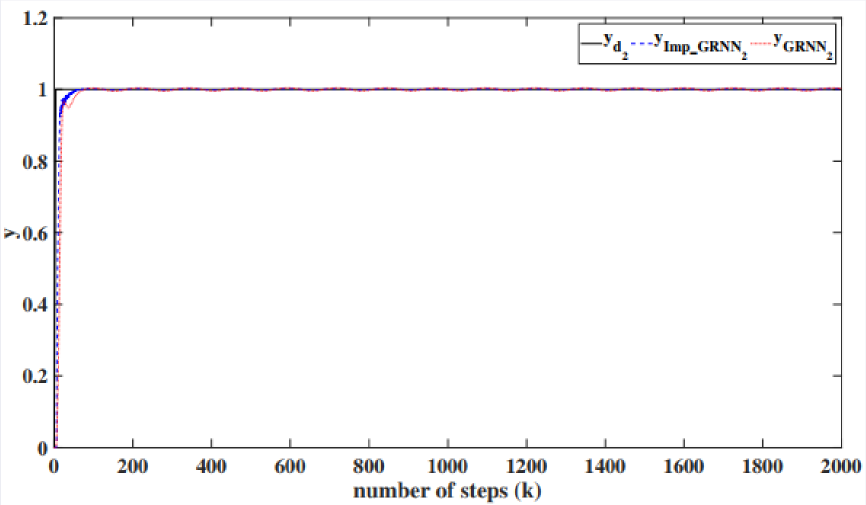


Figure 5: Tracking results for output two benchmark test 1 with step input

6. Simulation Results and Analysis

The Results of the Sum of Sine Input Tracking

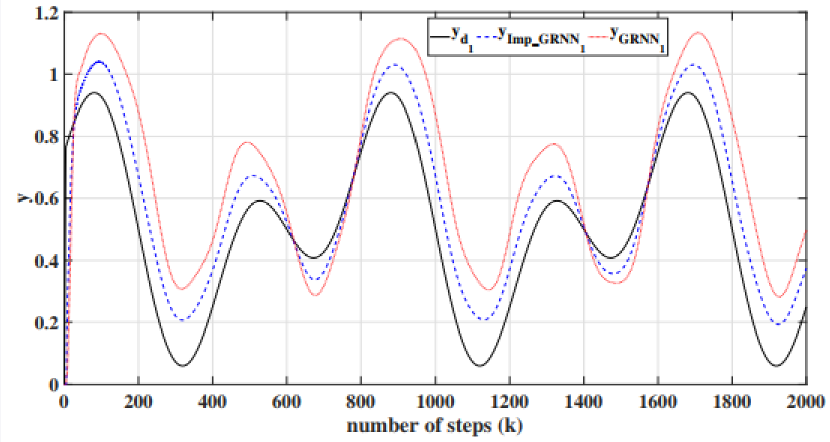


Figure 6: Tracking results for output one of example 1 for sum of sinus input

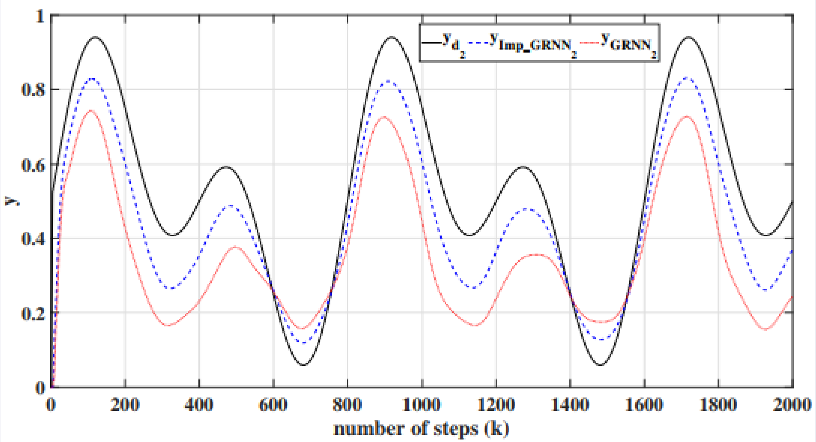


Figure 7: Tracking results for output two of benchmark test 1 for sum of sinus input

6. Simulation Results and Analysis

Table 1: Comparative GRNN versus proposed Imp-GRNN in MIMO non-affine system (step input $y_d()$)

MIMO non-affine system (step input y_d)	RMSE		MAE		SDE	
	GRNN	Proposed Imp-GRNN	GRNN	Proposed Imp-GRNN	GRNN	Proposed Imp-GRNN
First output $y_1()$	0.0589	**0.0418**	0.0435	**0.0323**	0.0663	**0.0450**
Second output $y_2()$	0.0650	**0.0472**	0.0564	**0.0381**	0.0742	**0.0494**

Table 2: Comparative GRNN versus proposed Imp-GRNN in MIMO non-affine system (sum of sinus input $y_d()$)

MIMO non-affine system (sum of sinus y_d)	RMSE		MAE		SDE	
	GRNN	Proposed Imp-GRNN	GRNN	Proposed Imp-GRNN	GRNN	Proposed Imp-GRNN
First output $y_1()$	0.2854	**0.1138**	0.2958	**0.1164**	0.3082	**0.1257**
Second output $y_2()$	0.3306	**0.1476**	0.3471	**0.1492**	0.3562	**0.1524**

6. Simulation Results and Analysis

Table 1: Comparative GRNN versus proposed Imp-GRNN in MIMO non-affine system (step input $y_d()$)

MIMO non-affine system (step input y_d)	RMSE		MAE		SDE	
	GRNN	**Proposed Imp-GRNN**	**GRNN**	**Proposed Imp-GRNN**	**GRNN**	**Proposed Imp-GRNN**
First output $y_1()$	0.0589	**0.0418**	0.0435	**0.0323**	0.0663	**0.0450**
Second output $y_2()$	0.0650	**0.0472**	0.0564	**0.0381**	0.0742	**0.0494**

Table 2: Comparative GRNN versus proposed Imp-GRNN in MIMO non-affine system (sum of sinus input $y_d()$)

MIMO non-affine system (sum of sinus y_d)	RMSE		MAE		SDE	
	GRNN	**Proposed Imp-GRNN**	**GRNN**	**Proposed Imp-GRNN**	**GRNN**	**Proposed Imp-GRNN**
First output $y_1()$	0.2854	**0.1138**	0.2958	**0.1164**	0.3082	**0.1257**
Second output $y_2()$	0.3306	**0.1476**	0.3471	**0.1492**	0.3562	**0.1524**

6. Simulation Results and Analysis

Benchmark Test 2

Consider q described as $q = [\theta_1, \theta_2]^T$, the Lagrange-Euler description of the SCARA manipulator is as:

$$M(q)\ddot{q} + C(\dot{q}, q) + g(q) = \tau$$

$$M(q) = \begin{bmatrix} (m_1 + m_2)l_1^2 + m_2 l_2^2 + 2m_2 l_1 l_2 \cos(\theta_2) & m_2 l_2^2 + m_2 l_1 l_2 \cos(\theta_2) \\ m_2 l_2^2 + m_2 l_1 l_2 \cos(\theta_2) & m_2 l_2^2 \end{bmatrix}$$

$$C(\dot{q}, q) = \begin{bmatrix} -m_2 l_1 l_2 \sin(\theta_2)(2\dot{\theta}_1\dot{\theta}_2 + \dot{\theta}_2^2) \\ -m_2 l_1 l_2 \sin(\theta_2)\dot{\theta}_1\dot{\theta}_2 \end{bmatrix}$$

$$g(q) = \begin{bmatrix} -(m_1 + m_2) g l_1 \sin(\theta_1) - m_2 g l_2 \sin(\theta_1 + \theta_2) \\ -m_2 g l_2 \sin(\theta_1 + \theta_2) \end{bmatrix}$$

$$\tau = \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}$$

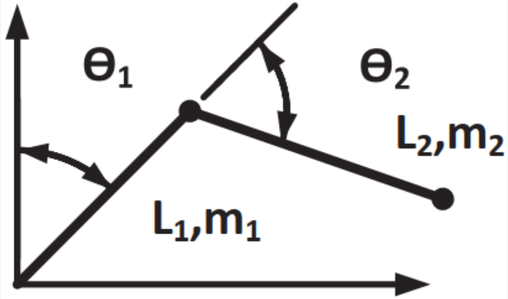


Figure 8: The 2-DOF planar robotic arm

The required trajectory shows as step form for θ_1 and θ_2. The initial values are [$\theta_1(0)$, $\theta_2(0)$] = [1, 1]. The learning rates are γ_1= [20.5, 20.5], γ_2 = [1.5, 1.5], the size of the hidden layer is ten, and σ = [0.01: 0.1: 1]. Compared of GRNN, we have the same for γ_1, the size of the hidden equal ten with $\sigma = 1$.

6. Simulation Results and Analysis

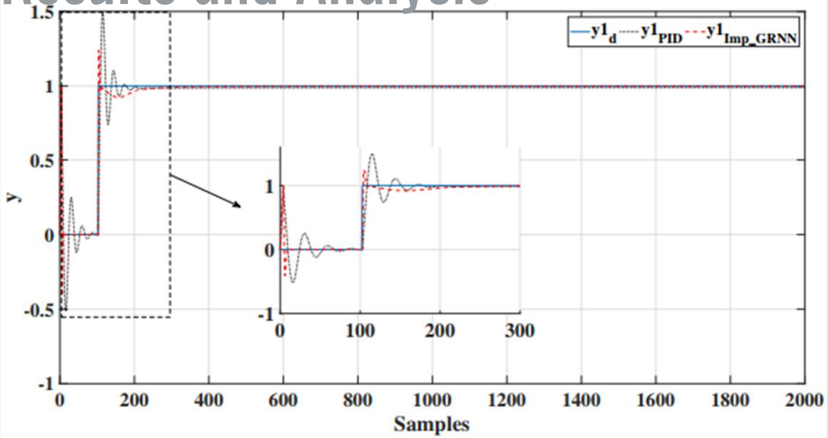


Figure 9: θ_1 tracking for robotic arm

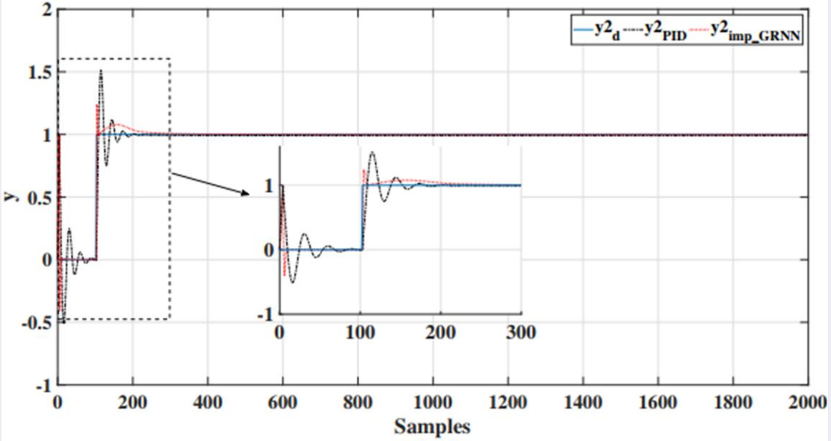


Figure 10: θ_2 tracking for robotic arm

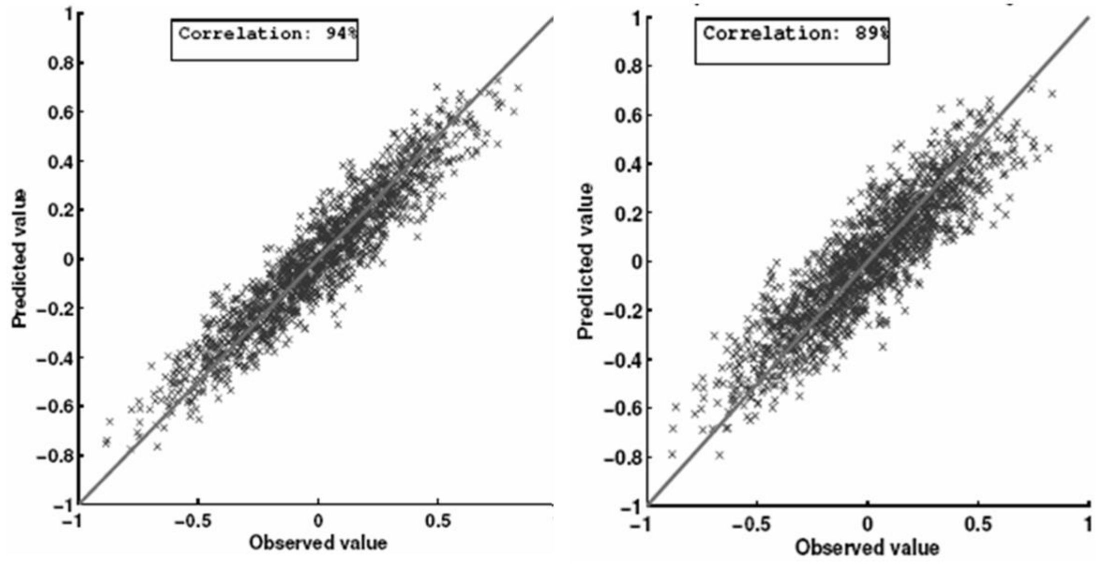
6. Simulation Results and Analysis

Table 3: Comparison of RMSE-MAE-SDE for GRNN and Imp-GRNN in benchmark test 2

Benchmark Test 2 (2-DOF robot arm)	RMSE		MAE		SDE	
	GRNN	**Proposed Imp-GRNN**	**GRNN**	**Proposed Imp-GRNN**	**GRNN**	**Proposed Imp-GRNN**
First output θ_1	0.0534	**0.0380**	0.0458	**0.0368**	0.0582	**0.0397**
Second output θ_2	0.0548	**0.0376**	0.0471	**0.0382**	0.0596	**0.0412**

6. Simulation Results and Analysis

Eventually by using statistical tests for the benchmark test 2, it is clear to see that the proposed Imp-GRNN also shows better than its GRNN counterpart in this respect as evident from the correlation coefficient between the predictions $\hat{y}_1[k]$, and the observed response $y_1[k]$. These results illustrated on Figure 11 confirms the fact the predictions of the proposed Imp-GRNN have a lower scatter around the unity line compared of the GRNN.



a) Proposed Imp-GRNN

b) GRNN

Figure 11: Comparative correlation between the predictions $\hat{y}_1[k]$ and the observed $y_1[k]$

6. Simulation Results and Analysis

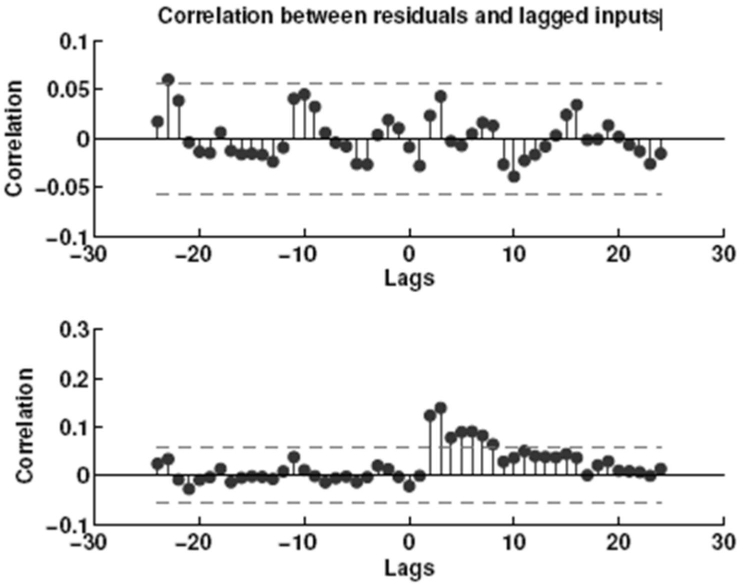


Figure 12 Correlation between residuals and lagged inputs for the GRNN (bottom) and the proposed Imp-GRNN (top)

6. Simulation Results and Analysis

Performance Comparison with other Advanced Controllers

Table 4: Comparison of Imp-GRNN with other controllers for benchmark test 1 with sum of sinus input

Comparative Controllers	**RMSE**		**MAE**		**SDE**	
	First output y_1	Second output y_2	First output y_1	Second output y_2	First output y_1	Second output y_2
Proposed Imp-GRNN	**0.1138**	**0.1476**	**0.1164**	**0.1492**	**0.1257**	**0.1524**
RBFNN	0.2236	0.2478	0.2071	0.2152	0.2562	0.2684
FFNN	0.2158	0.2496	0.1884	0.1982	0.2245	0.2314
Optimal PID	0.1506	0.1976	0.1498	0.1608	0.1662	0.1751

7. Conclusions

- **A new Imp-GRNN control algorithm which offers significant improvements compared to the original GRNN**.
- Proposed controller suggests an adaptive technique to implement the weighting values needless every optimized approaches via applying the regressive stochastic means.
- **A new smoothing parameter is innovatively applied to exploit the benefits of interval σ choosing coefficient instead of applying other computing-burden methods**.
- Moreover, novel forward input-output weighting values were proposed, along with novel output layer also used to ensure the weighting values’ adaptation.
- **Through the two benchmark tests, proposed controller obtains better tracking precision and faster convergent velocity compared of GRNN algorithm and other advanced neural-based controllers**.
- The Imp-GRNN controls well for multivariable benchmark test that guarantees its availability for various uncertain nonlinear plants.

REFERENCES

Aiqin Liu, Zhao, H., Song, T., Liu, Z., Wang, H., & Sun, D. (2021). Adaptive control of manipulator based on neural network. Neural Computing and Applications, 33(9), 4077-4085.

Ait Abbas, H. (2021). A new adaptive deep neural network controller based on sparse auto-encoder for the antilock braking system systems subject to high constraints. Asian Journal of Control, 23(5), 2145-2156.

Bresch-Pietri, D., Chauvin, J., Petit, N. (2012). Adaptive control scheme for uncertain time-delay systems. Automatica 48(8) 1536-1552.

Efe, M. Ö. (2008). Fractional fuzzy adaptive sliding-mode control of a 2-DOF direct-drive robot arm. IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics), 38(6), 1561-1570.

Ge, S., Zhang, J., Lee, T. (2004). Adaptive Neural Network Control for a Class of MIMO Nonlinear Systems with Disturbances in Discrete-Time. IEEE Transactions on Systems, Man and Cybernetics, Part B (Cybernetics) 34(4) 1630-1645.

Helong Yu, Yuan, K., Li, W., Zhao, N., Chen, W., Huang, C., ... & Wang, M. (2021). Improved butterfly optimizer-configured extreme learning machine for fault diagnosis. Complexity, 2021.

Hu, T., Khishe, M., Mohammadi, M., Parvizi, G. R., Karim, S. H. T., & Rashid, T. A. (2021). Real time COVID-19 diagnosis from X-Ray images using deep CNN and extreme learning machines stabilized by chimp optimization algorithm. Biomedical Signal Processing and Control, 68, 102764.

Jeyaraj, P. R., & Nadar, E. R. S. (2022). Real-time data-driven PID controller for multivariable process employing deep neural network. Asian Journal of Control.

Jiang, M., Ghahremani, M., Dadfar, S., Chi, H., Abdallah, Y. N., & Furukawa, N. (2021). A novel combinatorial hybrid SFL–PS algorithm based neural network with perturb and observe for the MPPT controller of a hybrid PV-storage system. Control Engineering Practice, 114, 104880.

Lee, T., Kim, I., & Baek, Y. S. (2021, January). Design of a 2-DOF ankle exoskeleton with a polycentric structure and a bi-directional tendon-driven actuator controlled using a PID neural network. In Actuators (Vol. 10, No. 1, p. 9). Multidisciplinary Digital Publishing Institute

REFERENCES

Ma, R., Zhang, B., Zhou, Y., Li, Z., & Lei, F. (2021). PID Controller-Guided Attention Neural Network Learning for Fast and Effective Real Photographs De-noising. IEEE Transactions on Neural Networks and Learning Systems.

Sabahi, F. (2018). Introducing validity into self-organizing fuzzy neural network applied to impedance force control. Fuzzy Sets and Systems, 337, 113-127.

Slama, S., Errachdi, A., & Benrejeb, M. (2021). Tuning Artificial Neural Network Controller Using Particle Swarm Optimization Technique for Nonlinear System. In Deep Learning Applications. Intech-Open.

Wang, X., Tu, S., Zhao, W., & Shi, C. (2022). A novel energy-based online sequential extreme learning machine to detect anomalies over real-time data streams. Neural Computing and Applications, 34(2), 823-831.

Wang, Y., & Wang, Y. (2021a). Discrete-time adaptive neural network control for steer-by-wire systems with disturbance observer. Expert Systems with Applications, 183, 115395.

Wang, Y., Niu, B., Wang, H., Alotaibi, N., & Abozinadah, E. (2021b). Neural network-based adaptive tracking control for switched nonlinear systems with prescribed performance: an average dwell time switching approach. Neuro-computing, 435, 295-306.

Xiao-Fang Liu, Zhan, Z. H., & Zhang, J. (2021). Resource-Aware Distributed Differential Evolution for Training Expensive Neural-Network-Based Controller in Power Electronic Circuit. IEEE Transactions on Neural Networks and Learning Systems.

Xu, Y. P., Tan, J. W., Zhu, D. J., Ouyang, P., & Taheri, B. (2021). Model identification of the proton exchange membrane fuel cells by extreme learning machine and a developed version of arithmetic optimization algorithm. Energy Reports, 7, 2332-2342.

Yang, C., Zhang, H., & Gao, Y. (2021). Analysis of a neural-network-based adaptive controller for deep-space formation flying. Advances in Space Research, 68(1), 54-70.

REFERENCES

Ye, M., Shen, J., Lin, G., Xiang, T., Shao, L., & Hoi, S. C. (2021). Deep learning for person re-identification: A survey and outlook. IEEE transactions on pattern analysis and machine intelligence, 44(6), 2872-2893.

Yewei Yu, Zhang, C., Wang, Y., & Zhou, M. (2021). Neural-Network-Based Iterative Learning Control for Hysteresis in a Magnetic Shape Memory Alloy Actuator. IEEE/ASME Transactions on Mechatronics, 27(2), 928-939.

Zhang, Y., Wang, F., & Yan, F. (2021). Fast finite time adaptive neural network control for a class of uncertain nonlinear systems subject to un-modeled dynamics. Information Sciences, 565, 306-325.

Zhao, H., Zeng, X., Chen, T., Liu, Z., & Woodcock, J. (2021). Learning safe neural network controllers with barrier certificates. Formal Aspects of Computing, 33(3), 437-455.

This research results have been published in:

Anh, H.P.H., Kien, C.V., **Nov. 2022,** "Robust Control of Uncertain Nonlinear Systems Using Adaptive Regressive Neural-based Deep Learning Technique," Journal of Expert Systems with Applications – ESWA – Elsevier, Vol. 214, March 2023, pp. 1-13 (ISI-SCI-Q1)

DOI: https://doi.org/10.1016/j.eswa.2022.119084

IIC2022

ADVANCED TECHNOLOGY AND SUSTAINABLE DEVELOPMENT

THANK YOU FOR YOUR ATTENTION

Prof. Dr. Trung Q. Duong

Chair Professor of Telecommunications
at Queen's University Belfast

Dr. Trung Q. Duong (IEEE Fellow) is a Chair Professor of Telecommunications at Queen's University Belfast, U.K. and a Research Chair of the Royal Academy of Engineering, U.K. His current research interests include optimization, signal processing, and machine learning in wireless communications. He has published more than 400 books/book chapters/journal articles/conference papers with 14,000+ citations and h-index 64. He has served as an Editor for the IEEE Transactions on Wireless Communications, IEEE Transactions on Communications, IEEE Transaction on Vehicular Technology, IEEE Communications Letters, and IEEE Wireless Communications Letters. He has received the Best Paper Award at the IEEE VTC-Spring in 2013, IEEE ICC in 2014, IEEE GLOBECOM in2016, IEEE DSP in 2017, and IEEE GLOBECOM 2019. He is the recipient of the Royal Academy of Engineering Research Fellowship (2015-2020) and the prestigious Newton Prize 2017. He is a Fellow of IEEE.

Title: 6G Ultra-Reliable and Low-Latency Communications: Achieving Industrial Automation

Abstract: Rapid advancement in wireless communication systems including Industry 4.0 machine-communications, tactile Internet, VR/AR services is moderated by deep challenges at the inflection point where machines and mission-critical applications will drive evolution in 6G networks. Ultra-reliable and low-latency communications (URLLC) have been envisioned to enable a new range of mission-critical applications and services such as self-driving vehicles, tele-surgery, autonomous factory, and industry automation. The market for URLLC service is forecast by ABI research to be worth US$18.9 billion by 2028, creating a vast opportunity for research and development to fulfil the communications needs in several vertical sectors, including healthcare, mobility and transport, and manufacturing. However, the enabling technologies for URLLC are stills in its infancy. This talk will discuss innovative paradigm of URLLC, not only basic requirements, essential definitions related to URLLC but also state-of-the-art technologies.

Dr. Antonino Masaracchia

Dr. Antonino Masaracchia (Member, IEEE) received the Ph.D. degree in electronics and telecommunications engineering from the University of Palermo, Italy, in 2016. From 2017 to 2018, he was a Post- doctoral Researcher at the Sant' Anna School of Advanced Studies, the BioRobotics Institute. Since September 2018, he has been a Research Fellow with the Centre for Wireless Innovation, Queens University Belfast, U.K. His research interests include fifth generation (5G) and beyond 5G networks (6G) oriented services, convex optimization and applied machine learning techniques to wireless communications, reconfigurable intelligent surfaces (RIS), UAV-enabled networks, and ultra-reliable and low-latency communications (URLLC). He is currently serving as Associate Editor for a section on Artificial Intelligence of Things published within Frontiers in the Internet of Things. He also serves as a Guest Editor for EAI Endorsed Transactions on Industrial Networks and Intelligent Systems and a Special Issue on Radio Frequency Energy Harvesting and Wireless Power Transfer published by Electronics (MDPI) for which he serves also as a Topic Editor. He serves as a Guest Editor for a Special Issue on Controls, Communications and Networking for Ad-Hoc Mobile Sensor Networks published by ICSES Transactions on Computer Networks and Communications. He has served as a Guest Editor for a Special Issue on Reliable Communication for Emerging Wireless Networks published by Mobile Networks and Applications (ACM/Springer).

Title: **Digital Twin for Open RAN: An Intelligent and Resilient 6G Radio Access Networks**

Abstract: The Open Radio Access Network (O-RAN) alliance has been recently established with the aim of evolving the RAN for next generation network in a way that it will result more open and smarter than previous generations. At the same time, the concept of digital twin (DT) is constantly emerging as a keystone technology for the deployment of mobile communication services envisaged for the sixth-generation (6G) networks. This keynote will provide a comprehensive vision on how DT and O-RAN represent two complementary concepts, which merged together will enable the deployment of an intelligent and resilient 6G RAN. A brief overview of both O-RAN and DT concepts is firstly provided. Subsequently, potential use cases and services delivered through a DT-based O-RAN architecture are illustrated and discussed. Finally, the most representative challenges and future research directions necessary for the successful development and deployment of this promising architecture highlighted and discussed.