

AJSE

American Journal of Science & Engineering

Volume 2 Issue 1

April 2021



American Journal of Science & Engineering (AJSE)

Society for Makers, Artists, Researchers and Technologists (SMART)

6408 Elizabeth Ave SE, Auburn 98092, Washington, USA

ISSN: 2687-9530 (Print) and 2687-9581 (Online)

Editor-in-Chief



Dr. Chuck Easttom

University of Dallas, USA & Georgetown University, USA

Research Interest: Cryptography, Cyber Warfare, Engineering Processes and Digital Forensics.

Dr. Chuck Easttom is adjunct lecturer at Georgetown University and University of Dallas. He is the author of 31 books, including several on computer security, forensics, and cryptography. His books are used at over 60 universities. He has also authored scientific papers (over 70 so far) on digital forensics, machine learning/AI, cyber warfare, cryptography, bio-engineering, and applied mathematics. He is an inventor with 22 computer science patents. He holds a Doctor of Science (D.Sc.) in cyber security (dissertation topic: "A Comparative Study of Lattice Based Algorithms for Post Quantum Computing") and a Ph.D. in Technology focused on nanotechnology (dissertation topic: "The Effects of Complexity on Carbon Nanotube Failures"), as well as three master's degrees (one in applied computer science, one in education, and one in systems engineering). He is currently working on third doctorate, a Ph.D. in computer science with emphasis on applied mathematics from the University of Portsmouth (dissertation topic "On the application of algebraic graph theory to network forensics"). He is a Senior Member of the IEEE and a Senior Member of the ACM as well as a member of IACR (International Association of Cryptological Research) a member of APS (American Physical Society), and INCOSE (International Council on Systems Engineering). He is also a Distinguished Speaker of the ACM (Association of Computing Machinery). and a Distinguished Visitor of the IEEE Computer Society, and a frequent speaker at conferences. He also currently holds 55 industry certifications (CISSP, CASP, CEH, etc.) He is a member of IEEE Software & Systems Engineering Standards Committee. He has worked on the DevOps 2675 IEEE standards group 2017 to 2019 and currently a member of the IEEE Engineering in Medicine and Biology Standards Committee. Standard for a Unified Terminology for Brain-Computer Interfaces P2731.

From the Editor's Desk:

A thought about scientific rigor on research: As scientists, we must always be striving to produce not just more research, but better quality research. A researcher should be his or her own harshest critic. Look at your own work with a skeptical eye. Could you provide clearer data? Are your references adequate and current? Is your statistical analysis appropriate and robust? Our goal as scientists is not merely to publish research, but to produce research that is truly impactful. By constantly striving to improve the quality of our own work, we improve the entire body of work in any scientific field.

Editorial Board:

Editor-in-Chief - Dr. Chuck Easttom (University of Dallas, USA & Georgetown University, USA)

Associate Editor - Dr. Nabeeh Kandalajt (Grand Valley State University, USA)

Board Members -

- i) **Dr. Phillip Bradford** (University of-Connecticut-Stamford, USA)
- ii) **Dr. Alex "Sandy" Antunes** (Capitol Technology University, USA)
- iii) **Dr. Izzat Alsmadi** (Texas A&M, San Antonio, USA)
- iv) **Dr. Lo'ai Tawalbeh** (Texas A&M University-San Antonio, USA)
- v) **Dr. Doina Bein** (California State University, Fullerton, USA)
- vi) **Dr. Hasan Yasar** (Carnegie Mellon University, USA)
- vii) **Dr. Moises Levy** (Florida Atlantic University, USA)
- viii) **Dr. Christian Trefftz** (Grand Valley State University, USA)
- ix) **Dr. Petros Spachos** (University of Guelph, Canada)

Page No.	CONTENT
1-10	<p>Crypto-jacking Threat Detection Based on Blockchain Framework and Deception Techniques</p> <p><i>The advances in malware attacks have taken place recently, and a robust security solution is required. Most traditional security approaches, including those proposed within the hybrid approach of malware detection, are no longer effective for detecting criminal cyber-attack strategies. In this paper, we introduced a novel approach to specifically detect malware that injected webcam protocols. The approach proposed a security model to address crypto-jacking as a threat to security in the blockchain. The current blockchain framework lacks the capacity for the identification of miners and nodes attacked by crypto-jacking malware. Hence, this approach is to ensure that all the nodes on the blockchain are secured, while also ensuring greater safety for the miners. The present approach requires that the identity of the miner is known, and what the miner's crypto-jacking did, which constitute major blockchain issues. The proposed novel approach involves injecting an application into each node to detect if an unusual process is taking place when the actual miner does not have access to the system. Since the application inserted will detect the highest possible phase using the CPU will get the name of the process and give it to the cuckoo. This is the suggested solution can also extend a system to the cuckoo machine that can be used. Confusing the cryptojacker and the block can be stored in the cuckoo, but the outcome will not be returned to the miner jacking the unit. The Cuckoo is going to highlight the significant details that will shape the backbone of the blacklist, for example, the infected internet protocol and the blockchain address. When the miner's address is on the blockchain blacklist, the miner will not allow any transaction to be received. In this way, a good miner will be protected from crypto-jacking malware or any hacker. The proposed approach was tested against the threat actor and the normal user, and it demonstrated a robust methodology capable of detecting malware in a significant way.</i></p> <p>DOI: doi.org/10.15864/ajse.2101</p> <p>Haissam Badih, Yasamin Alagrash (Department of Computer Science And Engineering Oakland University Rochester, Michigan 48309, USA.)</p>
11-23	<p>A Dynamic Graph-Based Systems Framework for Modeling, and Control of Cyber-Physical Systems Typified by Buildings</p> <p><i>In this paper, we present a framework for modeling certain classes of cyber-physical systems using graph-theoretic thinking augmented by ideas from the field of behavioral systems theory. The cyber-physical systems we consider are typified by buildings. We show that the thermal processes associated with a building can be represented as a graph in which (1) the node variables (temperature and heat flows) are governed by a dynamic system and (2) interconnections between these nodes (walls, doors, windows) are also described by a dynamic system. In general, we call a collection of such nodes and interconnections a dynamic graph (dynamic consensus network). Motivated by building thermal example, we present a framework for dynamic graphs and dynamic consensus networks. This framework introduces the idea of dynamic degree, adjacency, incident, and Laplacian matrices in a way that naturally extends these concepts from the static case. From this, we can easily define equivalent concepts of the dynamic consensus networks. Then we show how a behavioral systems approach can be used to develop kernel relationships between all system variables in dynamic graphs as typified by building thermal models. We discuss how such relationships can be used to analyze these systems' properties, focusing on controllability. The ideas developed for dynamic graph theory lead to developing a controllability analysis methodology for dynamic consensus networks in conjunction with the behavioral approach. We then developed the controllability conditions for the general dynamic networks, such as identical LTI nodes with dynamic edges or even in the more general case with heterogeneous nodes.</i></p> <p>DOI: doi.org/10.15864/ajse.2102</p> <p>Fadel Lashhab (Department of Electrical Engineering and Computer Science, Howard University, Washington DC, USA), Musbah Abdulgaderz (Department of Engineering Technologies, Bowling Green State University, Bowling Green, OH, USA), and Omar A. Zargelin (Department of Electrical and Electronic Engineering, Al Zintan University, Al Zintan, Libya)</p>
24-34	<p>Distributed Constraint Optimization Problem Solving in Unstable Environments</p> <p><i>Distributed systems are widely used to share tasks in various systems which communicate to each other. One main reason to use these systems is their ability to keep each system's privacy and share only the required information. The success of these systems relies on robust communications among their nodes. Advances in network and communication technologies have led to a more robust quality solution for distributed problems as these systems heavily rely on network robustness and stability. Despite this progress, the communication problems such as delay, loss, and noise still exist in many environments that have dramatically affected the quality of distributed problem solutions. In some recent studies, these issues have been explored partially; however, the need to investigate these issues' impact specifically when combined seems necessary. This article studies the effect of message loss while there is a chance of distortion in the receiving messages. To have a better view of communication issues, both static and dynamic problems are tested. Three distributed algorithms, Distributed Stochastic Algorithm (DSA), Distributed Breakout Algorithm (DBA), and Max-Gain Message algorithm (MGM), are chosen to be tested in these environments, and their performance has been compared to each other. Test results show that all three algorithms are highly impacted by network instability, while DSA provides better results in general.</i></p> <p>DOI: doi.org/10.15864/ajse.2103</p> <p>Saeid SamadiDana (Department of Computer Science and IT Austin Peay State University Clarksville, TN, USA)</p>

35-52	<p>Study of Behaviors of Multi-Source Rauch Filters</p> <p><i>This paper presents a study of behaviors of high-order multi-source Rauch filters based on the characteristics of complex functions. The advantages of Rauch's connections are an easier selection of circuit components and a simpler design in fully differential forms and complex topologies. A general superposition formula is also introduced for deriving the transfer functions of multi-source transmission networks such as fully differential amplifiers, fully differential low-pass filters, polyphase filters, complex filters, and quadrature signal generation circuits because the shapes of any motions and the characteristics of any transmission networks are analyzed based on superposition principle. The proposed Nichols chart of the self-loop function is used to do the ringing test for high-order Rauch low-pass filters because it can be easily calculated using MATLAB, simulated using SPICE, and measured in Network Analyzers. The innovation of Nichols chart of self-loop function is a useful tool for the ringing test for high-order networks. The simulation results of the phase margin at unity gain of the self-loop function in a proposed design of the 4th-order Rauch low-pass filter are 74 degrees (over-damping region), 68 degrees (critical damping region), and 59 degrees (under-damping region). The pass-band gain and the image rejection ratio of the proposed design of a 4th-order Rauch complex filter are 12 dB and 32 dB, respectively. It's shown that complex functions play an important role in mathematical models of motion in the multi-source networks.</i></p> <p>DOI: doi.org/10.15864/ajse.2104</p> <p>MinhTri Tran, Anna Kuwana, Haruo Kobayashi (Division of Electronics and Informatics, Gunma University, Kiryu 376-8515, Japan)</p>
53-57	<p>Filter-based Denoising Methods for AWGN corrupted images</p> <p><i>Visual information transfer in the form of digital images becomes a vast method of communication in the modern scenario, but the image obtained after transmission is many a times corrupted with noise. The received image requires some processing before it can be used. Image denoising includes the manipulation of the image data to produce a visually high-quality image. In this paper a review of some existing denoising algorithms, such as filtering approach; wavelet-based approach and their comparative study has been done. Different noise models including additive and multiplicative types are discussed. It includes Gaussian noise, salt and pepper noise, speckle noise and Brownian noise. Selection of the denoising algorithm is application dependent. Hence, it is necessary to have knowledge about the noise present in the image so that one can opt the appropriate denoising algorithm. The filtering approach seems to be a better choice when the image is corrupted with salt and pepper noise. Whereas, wavelet-based techniques are suited for more detailing. In this paper denoising techniques for AWGN corrupted image has been mainly focused.</i></p> <p>DOI: doi.org/10.15864/ajse.2105</p> <p>Sumit Singh Parihar, Shailesh Khaparkar (Department of Electronics & Communication Engineering, Gyan Ganga Institute of Technology & Sciences, Jabalpur)</p>



Crypto-jacking Threat Detection Based on Blockchain Framework and Deception Techniques

Haissam Badih, Yasamin Alagrash

Department of Computer Science And Engineering

Oakland University

Rochester, Michigan 48309, USA.

{hbadih, yhalagrash}@oakland.edu

Abstract—The advances in malware attacks have taken place recently, and a robust security solution is required. Most traditional security approaches, including those proposed within the hybrid approach of malware detection, are no longer effective for detecting criminal cyber-attack strategies. In this paper, we introduced a novel approach to specifically detect malware that injected webcam protocols. The approach proposed a security model to address crypto-jacking as a threat to security in the blockchain.

The current blockchain framework lacks the capacity for the identification of miners and nodes attacked by crypto-jacking malware. Hence, this approach is to ensure that all the nodes on the blockchain are secured, while also ensuring greater safety for the miners. The present approach requires that the identity of the miner is known, and what the miner's crypto-jacking did, which constitute major blockchain issues. The proposed novel approach involves injecting an application into each node to detect if an unusual process is taking place when the actual miner does not have access to the system.

Since the application inserted will detect the highest possible phase using the CPU will get the name of the process and give it to the cuckoo. This is the suggested solution can also extend a system to the cuckoo machine that can be used. Confusing the cryptojacker and the block can be stored in the cuckoo, but the outcome will not be returned to the miner jacking the unit. The Cuckoo is going to highlight the significant details that will shape the backbone of the blacklist, for example, the infected internet protocol and the blockchain address.

When the miner's address is on the blockchain blacklist, the miner will not allow any transaction to be received. In this way, a good miner will be protected from crypto-jacking malware or any hacker. The proposed approach was tested against the threat actor and the normal user, and it demonstrated a robust methodology capable of detecting malware in a significant way.

Index Terms—Crypto-jacking, cuckoo process, smart contract, webcam security.

I. INTRODUCTION

The intellectual sphere of malware was designed to adapt to the system dynamics that they are compromising. Threat actors who developed malware variants are constantly improving their techniques and innovations to weaken the cyber defense capabilities of their targets. It is therefore important for the cyber-security community to constantly track the activities of new malware strains and find ways to mitigate their effects. Real-time anti-malware strategies must balance the introspection thoroughness with the need to avoid slowing down the performance of the endpoint. Attackers design malware to work in the blind spots that occur as a result of such

compromises. However, the malware we possess causes an enormous problem in the attacked system which provides a hint to the user.

Contribution: Since we are using the blockchain framework where all the data are safe and protected, we face a problem where the node can be infected with malware. This malware attack can be a crypto-jacking and can capture the data to hash a transaction in a different user machine or node to compromise the machine's data and I/O devices. Therefore we need to protect the blockchain network nodes from any intruder who can use any network without the owner's permission.

In this paper, we make a threefold contribution:

- We used the Cuckoo sandbox tool which is emphasized the user's behaviors and redirect malware to particular operations within a consistent set of I/O cuckoo units. In the sense of protection of the cuckoo Protocol, we set up a case study of the cuckoo consumer's behavior. Along with the cuckoo Procedures, Cuckoo I/O System Party Protective Devices Approach as a whole to allow malware to hit a cuckoo sandbox machine for immediate identification.
- An expansion of our work basis on the previous work by detection. The malware and directing it to a cuckoo driver capture specific data, such as the IP address and the mining information. We used a filter driver to redirect to the cuckoo driver, designed for sensing when network traffic is a lull. When this happens, it turns any incoming traffic to the cuckoo driver and away from the network interface card [18].
- The operating system kernel the driver for sending and receiving network packets directly to the network interface board's hardware controller. There is a whole stack of drivers that perform various aspects of network traffic transmission and receipt. The filter driver is automatically placed on top of the Miniport driver automatically, therefore, below the rest of the driver stack. All traffic moving in either direction between the Miniport driver and the rest of the driver stack can be seen and handled by the filter operator [22].
- Novelty: Crypto-jacking poses a threat to blockchain security. The blockchain framework lacks the identification of the miners and nodes that crypto-jacking malware has



attacked. We need to make all the nodes in the blockchain safe and make all miners safer. Most importantly, we needed to know who the miner was and blocking the crypto-jacking caused to the miner. This is a significant blockchain problem. Our novel idea is to inject an application into each node that will detect if an odd process occurs while the actual miner has not accessed the network.

The inserted application finds the unusual behaviors which define the block header parameters that the crypto-jacking and the IP address need to produce. We can submit the method to the cuckoo device that can deceive the crypto-jacking, and the block can be stored in the cuckoo that will not return the result to the miner jacking the system. (Therefore), the cuckoo can give the information to the attacked system; the attacked system will publish it in the blacklist and list the IP in it so that when a transaction is needed to be tacked by a miner.

The blockchain will check the list to verify if the miner's IP address is on the list. When the miner's IP resides are on the blockchain blacklist, they will not allow any transaction to be picked. In this way, through crypto-jacking, the victorious miner can be defended from any hacker. The search will be open to all nodes in the peer-to-peer network in a smart contract [10].

Novelty: We use a filter driver designed for sensing to redirect to the cuckoo driver. When the traffic on the network is minimal; if this occurs, it redirects all incoming traffic to the cuckoo driver and away from the network interface card [1]. The OS kernel has a cuckoo driver to send and receive network packets directly to the network interface board's hardware controller. There is a whole stack of drivers conducting different aspects of the transmission and reception of network traffic. The filter driver is located automatically at the top of the cuckoo driver and, therefore, below the driver stack. The user of the filter can be seen and treated.

We developed an approach by adding method to cuckoo in a virtual machine (VM) on a device that can trick a crypto-jacking, and a block can be stored in a cuckoo VM that would not return the result to a miner jacking the machine. As a result, the cuckoo will send the details to the attacked system. Then the attacked one can publish it in the blacklist and list the blockchain's IP address so that every transaction needs to be tapped by the miner; the blockchain will check the list if the miner's IP and blockchain address are on the list. When the miner/node IP is on the blockchain blacklist, the node or miner will not allow any transaction to be selected. This way, by crypto-jacking, the victorious miner defends against any hacker. The search will be made open to all the nodes on the peer-to-peer network under a smart contract [10].

Organization. The remaining components of this paper are organized as follows: A related work was discussed in section III. In section IV we discuss the Blockchain and smart contracts. In section V we discussed the Cuckoo Process and cyber-deception. In section VI, we discussed Protecting Blockchain nodes from Crypto-jacking. In section VII we

discuss the Novel Malware Detection Protocol. In section VIII we discussed Testing and Validation. In section X, we summarized our work and concluded this paper.

II. HASH FUNCTION, BLOCKCHAIN ADDRESS IN PROOF-OF-WORK

A digital transaction comprises a set of key pairs, each with a private key and a public key. The private key (k) is an arbitrary number. We use the private key to create a public key with elliptical curve multiplication, a one-way cryptographic feature (K). The public key (K) is used to generate the bitcoin address using a one-way cryptographic hash function (A). The elliptic curve is used to measure the public key from the private key, which is irreversible:

$$K = k \times G(1)$$

where k is a private key, G is a constant point called the generator point, and K the corresponding public pin code is. The reverse procedure, known as the "finding of the discrete logarithm" - k calculating if you know K - is as tricky as any possible K values - i.e., the brute force search - can be checked. Blockchain uses a particular elliptical curve and a set of mathematical constants, specified by the National Institute of Norms and Technology Standards, SectP256k1 (NIST). The curve secp256k1 is set according to the function generating an elliptical curve:

$$y^2 \bmod p = (x^3 + 7) \bmod p(2)$$

The mod p (modulo prime number p) indicates that this curve is over a finite field of prime order p .

We presented with a mathematical description of the hash function used in blockchain works. The proof-of-work is a mathematical problem, the purpose of which is to create a link between two blocks. This link will be made within the header of the second block. Someone who is trying to work out a proof-of-work is called a miner. Let's consider two blocks, B_{prev} and B , and a number called bits and b . b estimates how difficult the proof-of-work is from b , and the target number can be directly computed. This target is a 64-digit hexadecimal number with multiple 0 for its left-digits, e.g.: (000000000000000000021047879c065745de75af6dcd473556dced2bcedd4qa71).

Assuming that the hash of the previous block is known, $H(B_{prev})$; we'll see how to define the hash soon after. It's a block. Solving the proof-of-work for block B , Known as mining block B , this amounts to finding a number, It was called the "nonce" such as:

$$H(H(B_{prev}) \oplus b \oplus timestamp(t) \oplus RH(B) \oplus nonce) \leq target(3)$$

Where: \oplus denotes an operation of concatenation; $timestamp(t)$ is the current time up to seconds. Since the block hash is recursively defined following the above procedure (we assumed that $H(B_{prev})$ was already known), we need an initialization: if B is the first block, there is no previous block, so $H(B_{prev})$ is a mere convention.



III. RELATED WORKS

As one of the most disruptive digital innovations in recent years, Cryptocurrency and the blockchain technology that drives it has been attracting a lot of interest. Blockchain technology offers tremendous potential for more transparent, accountable, and efficient means of storing government data and managing transactions. However, before the system can be scaled, there exist many challenges to overcome. Legal frameworks require reform to control digital currency markets and harness blockchain technology's full potential.

By integrating cuckoo processes, our work addresses blockchain challenges. The cuckoo will be used here as a trusted method to classify the invader. Many works have been suggested in many applications to leverage blockchain technology [14].

Koteska et al. To understand the contribution of the current research on the quality of the implementation of Blockchain, we examined the current quality issues in the implementation of Blockchain and identified the quality attributes of blockchain. Despite that, this topic is still immature. The results indicated that in terms of scalability, latency, throughput, cost-efficiency, authentication, privacy, and security, etc., the blockchain implementation needs to be improved [17]. Our job is special in that it provides a new tool and integration to block any invader using malware residing on a computer and was intended for attackers to access it.

This report explores many paths for future studies to foster comprehensive blockchain work that answers meaningful questions. It shows where research can benefit from multidisciplinary collaborations given the wide range of open-ended questions, and presents data sources as starting points for empirical research [3]. In the past, these consideration were ignored, but in our research, we identify the problem and establish the solution for the blockchain.

Vitalik Buterin, Ethereum's founder, a network that aims to go beyond Bitcoin to create a decentralized business blockchain, came up with a different approach. The author maintained that privately managed technology could be quicker, less costly, and better designed to enable faster interventions when repairs are needed. Nevertheless, it must be added that he acknowledged such centralized technology management will limit access rights [4]. In our work, we set out to solve the malware issues aimed at protecting miners and users.

Hong et al. provides the first systematic study of the size and effect of cryptojacking attacks. To support the automatic detection of malicious behaviors, They built CMTracker, which outperforms state-of-the-art detectors with two behavior-based runtime profilers. They've gathered 2,770 malicious samples from 853,936 popular web pages, and there subset testing manual shows that they are all true positives [19]. This paper only did studies but we did identify the problem and we introduce a solution with a real prove through our application.

Marchetto, Victor. examines crypto-jacking malware samples using both static and dynamic methods and address their

potential impact on services running on enterprise servers and operating systems, including critical infrastructure industries. The paper also lays out methods for how to identify and protect against these threats [20]. This work examines the impact of crypto-jacking and shows the impact in these industries. We show the problem and offer a solution supported by an application to do the work.

IV. BLOCKCHAIN MINING AND CONSENSUS

Digital cash was designed long before the emergence of the blockchain. In a central server system that was trusted to remove duplicate Expenditure [5]. Despite significant cryptographic advances, the lack of continuity between centralisation, anonymity and the prevention of double spending inevitably brings into question the viability of this new type of currency.

In 2008, Bitcoin gained recognition on the global marketplace by replacing the signature of the central server with a consensus system based on work evidence [6]. The novelty and enhancement of Chaum's conceptual experiment is the decentralized nature of the payment system created by blockchain technology, leading to a new age that stretches beyond global payments to corporate governance, social institutions, democratic participation and the functioning of capital markets [7].

Bitcoin was the first increase in cryptocurrencies, built based on blockchain's revolutionary technology, which previously lacked a decentralized ledger. The Bitcoin blockchain does not permit conditions to be imposed in a new block for finishing a transaction, as it includes only knowledge about the process itself. Nonetheless, the emergence of technology was the reason why smart contracts were created. Later on, the Ethereum blockchain platform allowed smart contracts to be used in operation.

There is no centralized authority for the blockchain network, as it is the real key to a decentralized system. The information in it is open to anyone to see because it is a transparent and permanent ledger. Therefore, by its very definition, everything on the blockchain is about transparency and everyone involved is responsible for their actions [15]. The blockchain has no transaction costs and is a simple but intelligent way of automatically and securely passing information from one system to the other.

The mechanism is initiated by one party to a block transaction. This one, Thousands are testing blocks, maybe millions of computers scattered around. It's the net. The verified block is connected to the chain and stored over the net. It's only making a single record, but also a single record with a specific past. Falsifying a single record would mean, in millions of cases, forging a record, the whole chain.

Blockchain technology guarantees that the dual-spending problem is addressed. The help of public-key cryptography gives each agent a private key (Maintained as a password) and a public key, exchanged with all other officers. When the future owner of the coin (or digital tokens) sends a public key to the coin. The original owner, the transaction is started. The

digital signature of a hash pushes the It's coins. Public keys are cryptographically generated addresses that are stored in the blockchain, guy.

Each coin is associated with an address and a cryptographic exchange. It's just a move of coins from one address to another. Pilkington et al, Ghost Strategically, the blockchain's striking characteristic are that the public keys are never. It is connected to the identity of the real world [8]. Although traceable, there are transactions. Allowed without the disclosure of one's identity; this is a big difference with the fiat currency transactions connected, except non-traceable cash transactions to particular economic agents of legal personality, whether human or physical morality.

Nick Szabo introduced this idea in 1994 and established a smart contract "a computerized transaction protocol is executing contract terms" [11]. Smart Contracts are scripts that are stored on the blockchain within the blockchain. History. For relational database management systems, they can be considered Slowly similar to the stored procedures [12]. They've got a particular address as They're living on the chain. When debating a deal, we are building a smart contract.

It then runs independently and automatically on each node in the network. According to the data involved in the transaction's triggering, in the stated case, the way. Smart contracts allow us to have calculations for general purposes. It was on the chain. However, when it comes to handling data-driven interactions [12] between network entities, this is where they excel.

Furthermore, when they are tasked with managing data-driven interactions [13] between network entities, they also excel. Imagine a blockchain network including Alice, Bob, and Carol, and exchange of Type X and Y digital assets. Bob deploys a smart network contract that defines: (a) a "deposit" function that allows him to deposit X amount into the contract, (b) a "trade" function that returns 1 X amount (from the contract's deposits) for each 5 Y amount he earns, and (c) a "withdrawal" function that allows Bob to withdraw all the assets held by the contract, as shown in Figure 1. We are looking at creating a smart contract in our work which can create a black list and add the IP address to the black list. When the IP address adds to the list, in the smart contract, we can create a logic to search and compare every IP in the black list. If the IP was found in the list, we can exclude any transaction from the node type and add the transaction to a block and avoid creating and transmitting any block into the ledger and adding it to the blockchain.

V. CUCKOO PROCESS AND CYBER-DECEPTION

We suggest threatening actors who can deliberately monitor user space processes and potentially leak information to users via a Skype call. It is worth noting that this concept is not completely fresh, as it had been described. It's the word pit. A pit, a cuckoo piece of data that only exists to signify a malicious entry. Take for example, a database of bank client information. In order to preserve customer protection, the bank needs to be able to monitor access to a bank records [9].

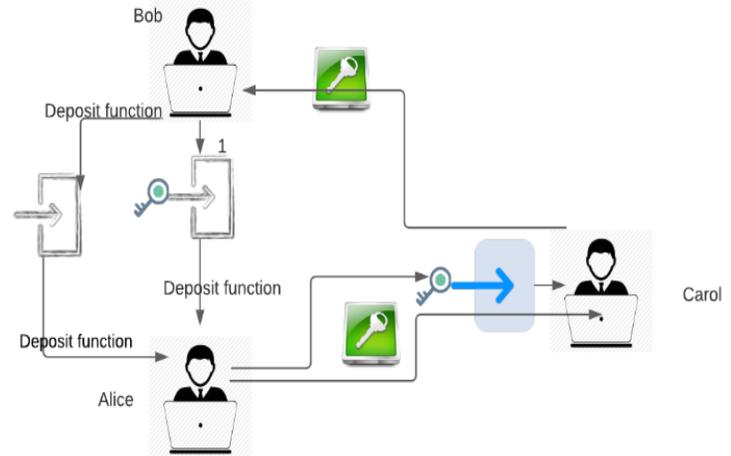


Fig. 1. Blockchain P2P transaction

In the field of network security, the notion of cuckoo systems is not new. Cuckoo Sandbox began as a Google Summer Code project in the Honeynet project in 2010. It was initially designed and built by Claudio "nex" Guarnieri, who is still the leading developer and lead developer [21]. The first beta update was released in Feb after initial work during the summer of 2010.

In one of the works, they present a TPM application protocol that detects a particular person in the middle attack where the adversary captures and replaces a legitimate computing platform with an imposter that forwards the challenge of platform authentication to the captive via a high-speed data link. This revised Cuckoo attack allows the imposter to satisfy the user's query of platform integrity by tricking the user into disclosing sensitive information to the imposter. Their protocol uses a typical smart card to verify the boot platform's integrity through TPM quote requests and verify TPM proximity by measuring TPM tick-stamp times required to respond to quotes [22].

Because we're operating in blockchain, we know the data is well-secure in the context of the blockchain. In any way, however, the blockchain may be Attacked by crypto-jacking malware that used to be restricted to download cryptocurrency program that is secretly mining unknowingly. When the machine is running, we can use the filter driver if we don't have access to our user. The filter driver supposes the machine was not in operation during times when it was not in use. Action coming from the desktop machine and designed to give a signal to the cuckoo machine that was going to start the operation. Therefore, we can redirect the crypto-jacking to the cuckoo system, deceive the malware actor, and quarantine the mining process in the cuckoo sandbox method.

VI. PROTECTING BLOCKCHAIN NODES FROM CRYPTO-JACKING

We describe the methodology of our work in this section. The logical approach and methods will be discussed. Cryp-

tocurrency, like Bitcoin, makes people rich and opens up something new that can use computers, resources, and power to help acquire money. Cryptocurrency is a former digital currency that exists only digitally instead of having printed, paper money. Units of money are made through a process called mining which uses the computer processing power to perform a complex calculation.

What happens in this mathematical calculation is complicated and it ensures that transactions between people using the cryptocurrency are documented safely and the accurate record of the ordering in which the transaction was carried out is identified and double expenditure is prevented. The user who completes this calculation will receive a reward of a small amount of the cryptocurrency, and this is how money is received. However, when used for malicious purposes, it is illegal to accept, but the cryptocurrency itself is legal. Since cryptocurrency undertakes complex calculations, it will harness the computer's power and network; it is very beneficial, however, from the miner's side, it is very expensive.

Therefore, crypto-jacking happens because the intruder wants to use someone else's machine and energy to generate better income. Therefore, they use the malware to measure the hash signature on another device, get the result to their computer, and transmit the new block to blockchain cryptocurrency. Next, they receive compensation from the blockchain cryptocurrency without using their electricity, resulting in less money to pay with more profit to gain. The crypto-jacking in the blockchain is a security threat. We need to make all the nodes in the blockchain safe in this matter and also make all miners safer. What we need to learn is who the miner is and what block the crypto-jacking miner has made.

Thus, this is a big issue in the blockchain. Our novel idea is to inject an application into each node that will detect if an odd process is occurring while the actual miner has not accessed the system. The inserted application finds the unusual behavior that can define the block header parameters that the crypto-jacking and the IP address need to produce. Then, we can submit the method to the cuckoo device which will deceive the crypto-jacker and the block can be stored in the cuckoo that will not return the result to the miner jacking the system.

Therefore the cuckoo sandbox will give the information to the attacked system, the attacked system will publish it in the blacklist, and the IP address and the blockchain address will be entered in this list. Therefore, any transaction that will have to be tackled by a miner the blockchain will search the list to verify if the miner's IP address is contained therein. If miner's IP exists in the blockchain list, no transaction will be chosen by the miner. It is in this way that the crypto-jacking protects the good miner from any hacker. This will be done using a smart contract. See Figure 2 and the displayed device consumer application in Figure 3.

Building user activity cuckoo sandbox. In this case, cuckoo dodger closely watches to determine the system's IP addresses with which the compromised computer interacts over the network. Instrumentation of the network interface alone cannot make it appear as if the compromised system

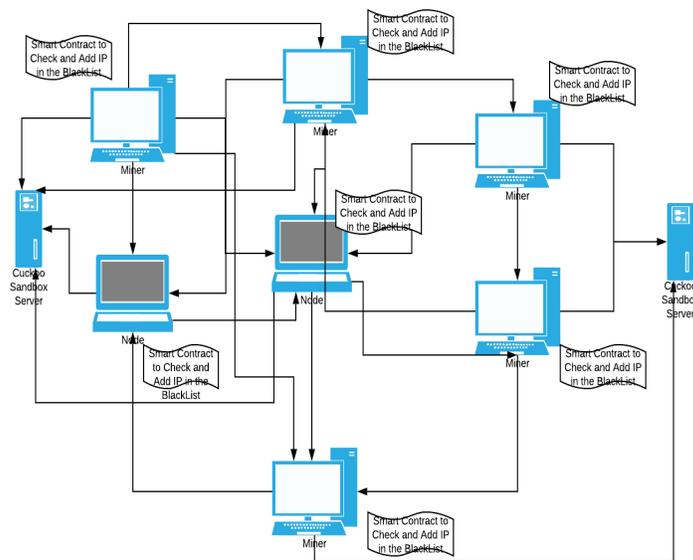


Fig. 2. Secure user in secure system

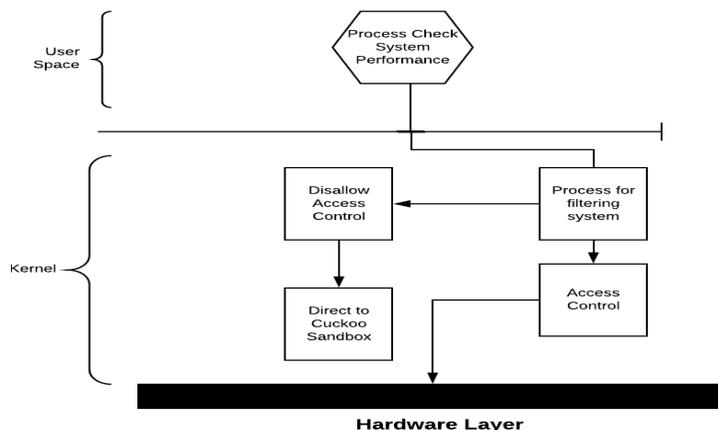


Fig. 3. System User

interacts with cuckoo sandbox. We need a mechanism in the user space, like all the network applications, to send synthetic network traffic through the driver stack, although it cannot be regarded as a real operation. This is mainly due to the high overhead of real processes, and also due to the visual appearance on the display of a real process, which definitely confuses the user and can also result in action on its graphical user interface.

In their previous work, they explored cuckoo processes [18]. We extend the principle of cuckoo process in this research to produce observable, cuckoo sandbox user activity that sends the synthetic network traffic we need for our cuckoo tactics. The authors found two main factors related to cuckoo processes, namely the dynamics of life and the efficiency of run-time. Through instrumenting with the data structures

used by the OS kernel to handle processes and threads, they make a cuckoo process noticeable. Consequently, in their performance, the task manager method, the task-list command, and the ps command all showed an entry for the cuckoo operation.

A cuckoo process' run-time performance dynamics are generated by instrumenting with the output counter data structures in the OS kernel. A performance data provider in the OS kernel collects these data structures, which in turn makes performance counter-data available in user space to users, including potential malware. Such performance counter-data relates to multiple components, including the cuckoo sandbox mechanism, network interface and diagnostics of TCP/IP performance.

Integration with the interface of the network. The binary instrumentation module generates a placeholder that is allocated a set of virtual memory addresses to the cuckoo method. The cuckoo process usually does not run and therefore does not use CPUs, physical memory frames, secondary storage, networking, and other resources. The binary instrumentation module places synthetic network packets and a small block of code in the memory of the cuckoo process when the filter driver finds a time window with low or missing network activity and decides to switch to the cuckoo network interface. This can work in TCPREPLAY method [9]

Because the synthetic network packets are already presented in the memory, they are simply sent over the network by the code block. Since it is the currently active cuckoo network interface, the synthetic network packets reach the cuckoo I/O module, which generates replies and sends them back through the driver stack to the cuckoo process. Responses are network packets that also tend to originate in sandbox. The cuckoo process stops at this stage and gives its physical memory frames back to its state of 0-resource usage. The physical network interface is reinstalled by the filter driver.

Impact of the outcome. A network sniffer reveals a two-way communication on the compromised computer between a device and honeypots. These honeypots seem to be active in the network. Once again, they are indistinguishable from the production equipment. To prevent legitimate users from seeing fake network traffic in their network sniffing devices, we rely on a monitor filter driver that eliminates all cuckoo data bound for the monitor before showing those data [16]. As seen in Figure 4.

Most information can be collected by the network emulator. Additionally, some of the details that the emulator requires is the invader's IP address to pass to the output of the process control system. This information is required to identify the invader and to be able to publish the IP address for protection so that all nodes are aware of the illegal work done by the bad node or miner, as shown in Figure 4.

Smart contract process for blacklist. This method tests system performance after obtaining the emulator's IP address and blockchain address that invades the crypto-jacking program. The output of the process control system would submit the required data to a blacklist of smart contracts. This smart

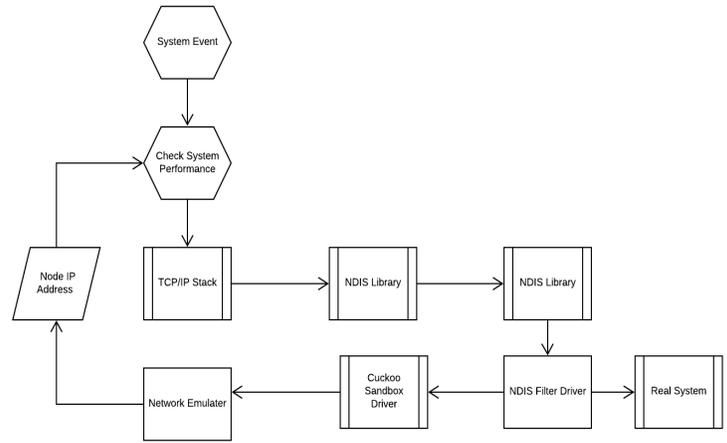


Fig. 4. Cuckoo Sandbox process

contract will get the address of the blockchain and the IP address and add it to the blacklist. This blacklist functions to verify if any node or miner has performed illegal activities, such as using the crypto-jacking and attempting to mine in another network or system. This is because they rely on other schemes to use the resources to reduce costs and make the profit illegal. If this information is saved in the blacklist, it will be forbidden if they need to get any blockchain transactions to block it and post the block to the blockchain. This way, if we have achieved security of good miners and nodes, we won't be allowed to do that, as seen in Figure 5.

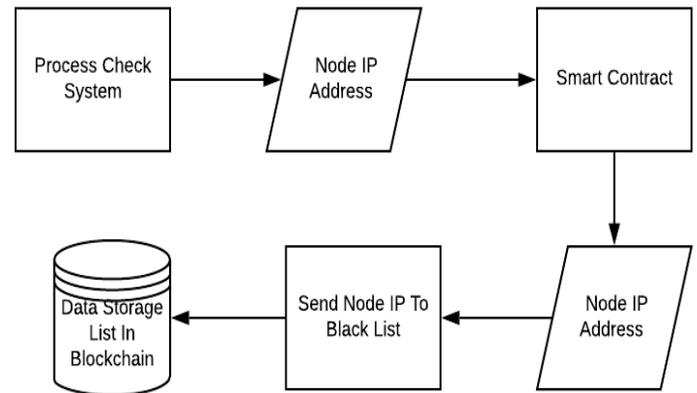


Fig. 5. Smart Contract Process

What is a smart contract? Smart contracts are computer programs that only operate upon the fulfillment of specific conditions. Simply put, it's a protocol that implements a contract's terms. Like a conventional contract, the conditions and penalties around an arrangement are specified by a smart contract. Unlike traditional contracts, such obligations are executed automatically by a smart contract, often through an



escrow-like account. Also, unlike a conventional contract, they can not be changed once the terms of a smart contract have been documented on the blockchain.

Creating Smart Contracts. The Ethereum Virtual Machine (EVM) offers a decentralized virtual machine for developers to create multiple smart contract applications in Ethereum. Thought of the EVM as a machine that runs all smart contracts globally. Because of this smart contract, our goal will be to use it to define the blacklist in our research and make it available for blockchain transactions to mine a block and publish the block in the blockchain. And the aim of this contract is to make it accessible to all nodes and miners as a protocol in the blockchain as a decentralized application as can be seen in Figure 6.

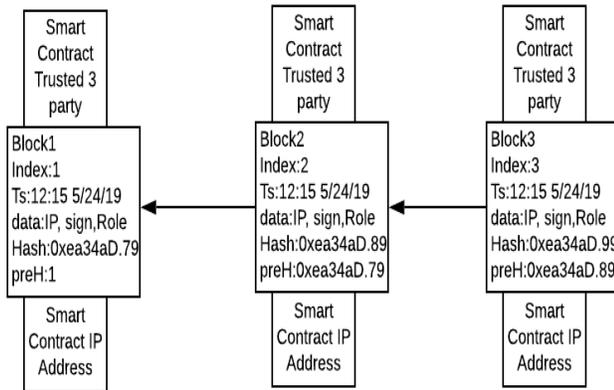


Fig. 6. Smart Contract in Blockchain

Algorithm 1: How to Get the miner invader

```

Input: Performance and get if system in use
Result: Performance and System in use
// This is code in the algorithm 1 will show how the
process will get the performance, and base on the
performance and the if system in use we can
identify whether the system can reach the control
access in the kernel of direct the execution for
mining to the cuckoo sandbox emulator.
while While performance value >30percent do
    if performance value >30percent And System in use then
        System permitted to control access;
    else
        System goes to cuckoo;
    end
end
    
```

Algorithm 2: How to add Address of a miner invader into blacklist

```

Result: Cuckoo capture mine invader
Input: IP and blockchain address.
// This code is intended to contain the invader
mechanism in cuckoo, capture IP and blockchain
address if the correct IP invader allows us to
obtain both addresses and transfer them to a smart
contract and then add them to a blacklist
while Address Invader not null do
    Address received by Agent Process check performance;
    Send address to Smart Contract;
    Smart Contract Add Address to blacklist;
    Exit;
end
    
```

VII. NOVEL MALWARE DETECTION PROTOCOL

Before we test our model and recognize any possibility of malware attack we consider this model. As the user working on the computer doing Skype meetings or conferences, we started to test our model in line with what is depicted in Figure 7. Who going to discover the crypto-jacking in web browser? In the point we have two scenarios of solutions.

- Base on our previous work second order detection we have cuckoo I/O assuming bob operate cuckoo I/O, which means is going to discover the crypto-jacking through cuckoo I/O.
- Our scenario Bob doing Skype call using microphone and webcam also cuckoo is activated for every signal of request send from Bob machine.

This work is a continuous to our previous work second order detection. In this paper we do deep investigation based on Cuckoo Security Protocol. We develop an approach as another support for cuckoo based on most common usage scenario like transfer money using blockchain over Skype. Our solution based on the following as shown in Figure 7, we applied our solution to the blockchain technique to check malware possibility in the system. Since we want the system to be secured and we don't want to overload the user's computer, we turn the cuckoo system on and redirect the malware to it. More specifically we trace our model from user request. The request will be through the application called CpuProcessUsage. What this application does is that it will capture every process running in the system with CPU usage percentage. This will then add all the processes name and usage into the dictionary object list with typed string and float as showing in Figure 8. Also we can capture all the URLs from the user infected and transfers them to cuckoo.

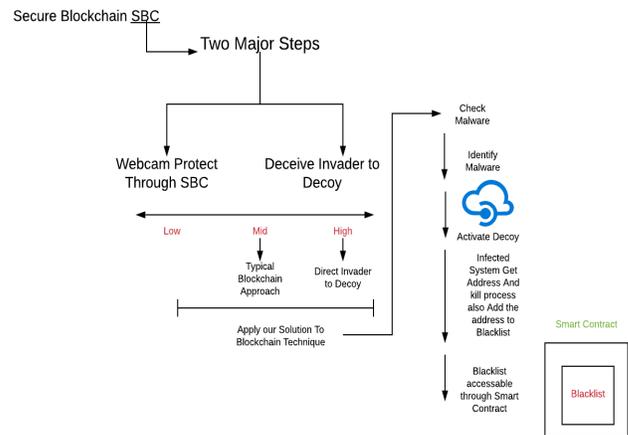


Fig. 7. Malware Scenario.

VIII. TESTING AND VALIDATION

Based on the sequence of our work we introduced Blockchain's trust-based smart contracts to access a webcam in a network that provides better performance and robust



system security most safely with less overhead. Our prototype provides more secure webcam sessions that are used in Windows Operating System user and kernel mode. The kernel-mode controllable strategy has been developed to record any video based on the user's role that will be authorized by smart contracts. Our Blockchain solution allows the Windows operating system kernel to access the Webcam.

In the second approach, we proposed that cuckoo userspace activity, together with a case study, namely a cuckoo security protocol, involves malware in a series of interactions that lead them to access a cuckoo I/O device, i.e. a cuckoo webcam in this case. In practical terms, cuckoo user space activities are carried out by a coherent set of cuckoo I/O devices and cuckoo processes. This work makes cuckoo user space activity and cuckoo's I/O indistinguishable from their actual counterparts and thus increases the uncertainty of malware operations on a compromised machine to benefit from their detection.

In the current approach, this paper set out to secure blockchain (SBC) in two fundamental major steps:

- The webcam protects the user through the SBC by using the blockchain framework.
- The cuckoo process deceive the invader to cuckoo by using the second order webcam.

As shown in Figure 7, we applied our solution to the blockchain technique to check malware possibility in the system. Since we want the system to be secured and we don't want to overload the user's computer, we turn the cuckoo system on and redirect the malware to it. More specifically we trace our model from user request. The request will be through the application called CpuProcessUsage. What this application does is that it will capture every process running in the system with CPU usage percentage. This will then add all the processes name and usage into the dictionary object list with typed string and float as showing in Figure 8.

```

File Edit View Project Build Debug Team Tools Architecture Test Analyze Window Help
NuGet: CpuProcessUsage Program.cs x CpuProcessUsage Program
class Program
{
    static void Main()
    {
        var counterList = new List<PerformanceCounter>();
        while (true)
        {
            var procDict = new Dictionary<string, float>();
            Process.GetProcesses().ToList().ForEach(p =>
            {
                using (p)
                {
                    if (counterList
                        .FirstOrDefault(c => c.InstanceName == p.ProcessName) == null)
                        counterList.Add(
                            new PerformanceCounter("Process", "% Processor Time",
                                p.ProcessName, true));
                }
            });
        }
    }
}
    
```

Fig. 8. Cpu Process Usage.

Figure 9 shows the analysis of the event process, in the CPU. After, that we add them to a list base on the performance counter object. The value will be calculated over the base line of the number of the logical CPU's multiply by 100, so this is going to be a calculated over the baseline of more than 100 as seen in Figure 10.

When we get the counter list for performance processes we get the list in descending order and this order will help us to simply get the highest process with CPU usage and get selected as shown in Figure 11 and the result for the process

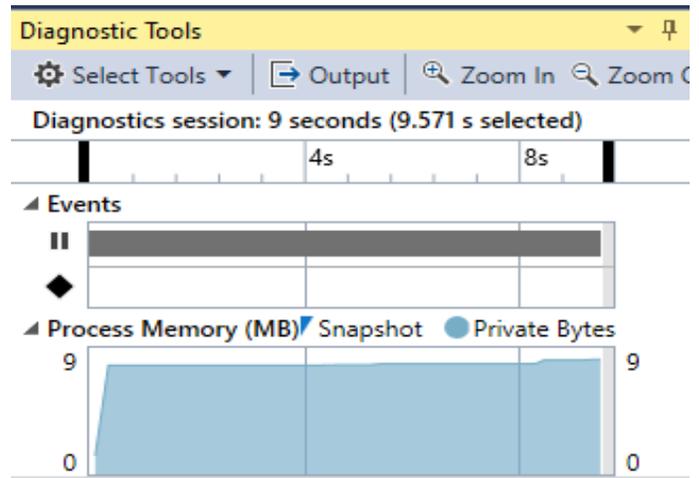


Fig. 9. CPU Usage and analysis.

```

counterList.ForEach(c =>
{
    try
    {
        // This value is calculated over the base line of
        // (No of Logical CPUS * 100), So this is going to be a
        // calculated over a baseline of more than 100.
        var percent = c.NextValue() / Environment.ProcessorCount;
        if (percent == 0)
            return;
        // Uncomment if you want to filter the "Idle" process
        //if (c.InstanceName.Trim().ToLower() == "idle")
        //    return;
        procDict[c.InstanceName] = percent;
    }
    catch (InvalidOperationException) { /* some will fail */ }
});
    
```

Fig. 10. CPU Processes Usage in the system.

showing the selected process with high CPU is depicted in Figure 12. When the process name is selected we can kill it after capturing the URLs list from the browser and redirecting the list of URLs to cuckoo is depict in Figure 12. The cuckoo will execute the URLs until it captures the infected website. We can run the infected website from cuckoo and deceive the malware and capture it through the static analysis tools. We can get the most important data which are the IP and the blockchain addresses because the malware we are targeting is the crypto-jacking, which is a malware that will run as a mining process to do mining with high calculation process which can exhaust the CPU. Because of that, we redirect the process to cuckoo and we can capture the IP and the blockchain addresses.

```

myKeylogger.cs BrowserHistory.cs Program.cs x Object Browser
CpuProcessUsage Program.cs x CpuProcessUsage Program
namespace CpuProcessUsage
{
    class Program
    {
        static void Main()
        {
            var val = like.First();
            Console.WriteLine(val.Key);
            namespace myKeylogger
            {
                class processClass = new ManagementClass("Win32_Process");
                ManagementObjectCollection classObjects = processClass.GetInstances();
                foreach (ManagementObject classObject in classObjects)
                {
                    string name = classObject.GetPropertyValue("Name").ToString();
                    if (name.Contains(val.Key.ToString()))
                    {
                        Console.WriteLine(name);
                        Process[] proc = Process.GetProcessesByName(name);
                        proc[0].Kill();
                    }
                }
            }
            //ArrayList theProcessToRun = history.ChromeBrowserHistory();
            string irpEvent = keylogger.IrpCapture(@ngab);
        }
    }
}
    
```

Fig. 11. CPU Process Usage to transfer process to cuckoo.

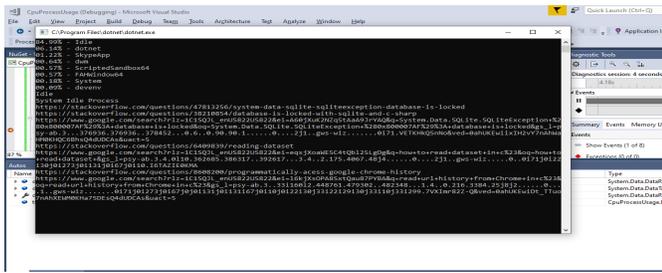


Fig. 12. Check Perform Processes.

Since we have identified the malware process as shown in Figure 12, we can connect to cuckoo. To connect to a Windows cuckoo, we can use A ConnectionOptions to connect to a remote computer with default connection options. For protection, we can use the credential bypassing the IP address username and password, with these parameters we can make remote connections for WMI v1 through the ManagementScope object as showing in figure 11, and then we can run the process from the invoking method in the cuckoo framework. Then cuckoo will do read the list of websites and get the infected website caused the crypto-jacking, through this can get the IP address from the mechanism by using Wireshark or another method that can collect network traffic or static analysis tools.

Wireshark is a free and open-source package analyzer. It is used for network troubleshooting, research, software and communication protocol creation, and education. So we can get the IP address via Wireshark, because the crypto-jacking would allow heavy contact from the IP network to another IP, which is the IP invader because we recognize the IP of our system that we can catch with which we communicate.

Since cuckoo possesses both addresses including the IP and the blockchain, therefore, we want to send the information to the invaded system. The reason is that we want to insert the information into the blacklist. The blacklist will generate through a smart contract and this smart contract is code written in solidity language which can reside in remix Ethereum. Within the Ethereum, a network of thousands of computer processes is used every time a program is used. Contracts written within smart contract-specific programming languages are compiled into bytecodes that can be interpreted and executed by the ethereum virtual machine (EVM) feature. Therefore, we created a smart contract to be inserted in the invader's addresses into the blacklist as depicted in Figure 13.

Now that we have all the applications existing, we create a JavaScript program through which we will be able to access the smart contract and add the bad miner addresses into the blacklist. To be able to do that we use the plugin for blockchain. Since we are using ethereum blockchain we use Web3.js plugin which is a series of libraries that allow you to communicate with a local or remote ethereum node through an HTTP or IPC connection. The JavaScript web3 library communicates with the Ethereum blockchain as depicted in Figure 14.

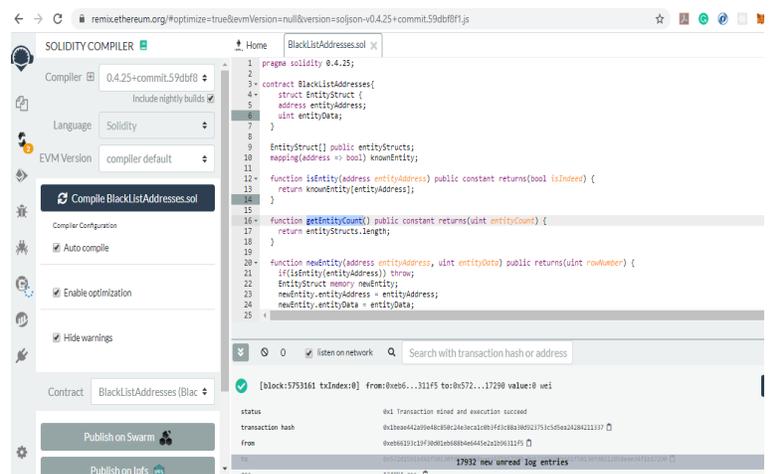


Fig. 13. Blacklist Smart Contract.

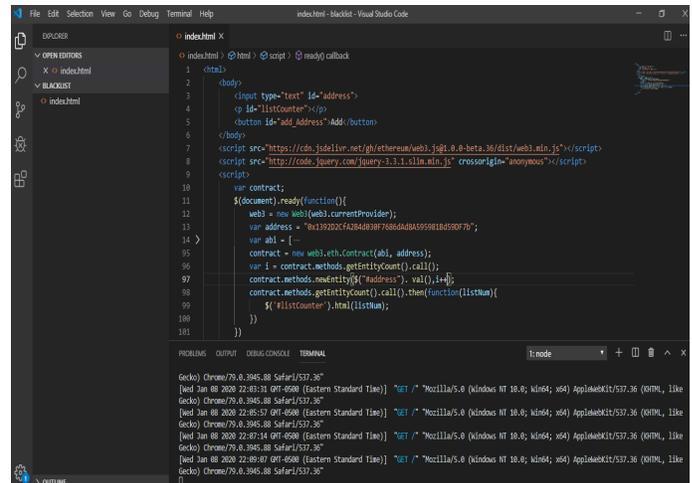


Fig. 14. web3 library communicates with the Ethereum.

IX. PERFORMANCE EVALUATION

We perform a The maximum deviation (D) in the cumulative is measured by the KS test of crypto-jacking scripts to analyze the performance and complexity of their code. Static analysis reveals standard code-specific features that provide a more in-depth insight into information flow to code execution. For static analysis, we collected crypto-jacking scripts from all major crypto-jacking service providers found in our datasets, such as Coinhive, JSEcoin, Crypto-Loot, Hashing, and Deep-Miner. We noted that all service providers had unique codes specific to their platform. In other words, the websites using Coinhive's services had the same JavaScript code template for all of them. As a result, 80 percent of our dataset websites used the same JavaScript template for crypto-jacking. Similarly, all JSEcoin-based websites used the same standard template for mining. However, each service provider's code template was different from each other, which led us to believe that each script had unique static features. With all this in mind, we have been performing a static analysis on the crypto-jacking;



the websites and the results were compared with other standard JavaScript for a baseline comparison.

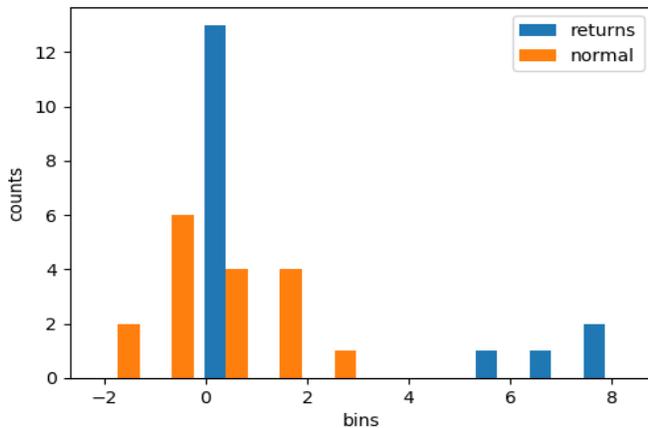


Fig. 15. The maximum deviation (D) in the cumulative is measured by the KS test.

X. CONCLUSION

This paper demonstrates a cyber-deception insight into the blockchain framework to identify the finger prints of the threat actor activities. Our approach keeps webcam protocol safer and more robust in the blockchain technology. The central component of our approach entailed a novel idea which is to inject into each node an application that can detect if an unusual process is going on while the actual miner does not have access to the system. Since the inserted application detects an unusual behavior that can determine the parameters of the block header that the crypto-jacking needs to create.

We can apply the method to the cuckoo device that can circumvent the crypto-jacking and the block can be stored in the cuckoo that does not return the result to the miner who is jacking the system. The cuckoo will highlight the significant information which is going to be a backbone for the blacklist such as the infected internet protocol and the blockchain address. This information is conveyed to the invaded node. Therefore, the invaded node will get important information and insert them into a blacklist.

Also, any transaction that needs to be tacked by a miner or node the blockchain will check the blacklist through the significant data such as IP and blockchain addresses of the miner is on the list. When node IP and blockchain addresses reside in the blockchain blacklist, the miner will not allow for any transaction to be picked. The solution that we have identified therefore assists in our understanding how to defend the successful miner from any crypto-jacking or hacker. The search will be open to being implemented in all nodes in the peer-to-peer network in a smart contract.

One type of webcam malware is clickjacking, which manipulates the website rendering making it invisible. We are entirely sure that we have resolved the malware as well.

REFERENCES

- [1] J. Rrushi, "HoneyPot Evader: Activity-guided Propagation versus Counter-evasion via Decoy OS Activity", In Proceedings of the 14th IEEE International Conference on Malicious and Unwanted Software, Nantucket, Massachusetts, USA, October 2019.
- [2] J. Yuill, M. Zappe, D. Denning, and F. Feer. "Honeyfiles: Deceptive Files for Intrusion Detection", IEEE Workshop on Information Assurance, West Point, NY, USA, pp. 116-122, June 2004.
- [3] M. Risius, and K. Spohrer, "A Blockchain Research Framework: What We (don't) Know, Where We Go from Here, and How We Will Get There. Business and Information Systems Engineering, 59(6), 385-409." (2017)
- [4] Giovanni P. "BLOCKCHAIN: IS SELF-REGULATION SUFFICIENT?." (2017)
- [5] D. Chaum "Blind signatures for untraceable payments." Advances in cryptology. Springer, Boston, MA, 1983.
- [6] A. Back, M. Corallo, L. Dashjr, M. Friedenbach, G. Maxwell, A. Miller, and P. Wuille "Enabling blockchain innovations with pegged sidechains." URL: <http://www.opensciencereview.com/papers/123/enablingblockchain-innovations-with-pegged-sidechains> 72 (2014).
- [7] A. Wright, and P. De Filippi "Decentralized blockchain technology and the rise of lex cryptographia." Available at SSRN 2580664 (2015).
- [8] M. Pilkington "11 Blockchain technology: principles and applications." Research handbook on digital transformations 225 (2016).
- [9] S. Jajodia, V. Subrahmanian, V. Swarup, and C. Wang "Cyber deception.", Springer, 2016.
- [10] Y. Yuan, and F. Y. Wang "Towards blockchain-based intelligent transportation systems." 2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC). IEEE, 2016.
- [11] N. Szabo, Smart Contracts. [Online]. Available: <http://szabo.best.vwh.net/smart.contracts.htm> (1994).
- [12] MySQL Reference Manual—Using Stored Routines (Procedures and Functions), accessed on Mar. 15, 2016. [Online]. Available: <http://dev.mysql.com/doc/refman/5.7/en/stored-routines.html>
- [13] Eris Industries Documentation—Smart Contracts, accessed on Mar. 15, 2016. [Online]. Available: <https://docs.erisindustries.com/explainers/smartcontracts/>
- [14] C. Stoll, "The Cuckoo's Egg". New York: Doubleday, 1989.
- [15] B. Liu, X. Yu, S. Chen, X. Xu, and L. Zhu, "Blockchain based data integrity service framework for IoT data." In 2017 IEEE International Conference on Web Services (ICWS), pp. 468-475. IEEE, 2017.
- [16] J. Rrushi, "Dnic architectural developments for 0-knowledge detection of opc malware," IEEE Transactions on Dependable and Secure Computing, September 2018
- [17] B. Koteska, E. Karafiloski, and A. Mishev "Blockchain Implementation Quality Challenges: A Literature." SQAMIA 2017: 6th Workshop of Software Quality, Analysis, Monitoring, Improvement, and Applications. 2017.
- [18] H. Badih, Y. Alagrash, J. Rrushi A Blockchain and Defensive Deception Co-design for Webcam Spyware Detection. In 2020 IEEE Intl Conf on Dependable, Autonomic and Secure Computing, Intl Conf on Pervasive Intelligence and Computing, Intl Conf on Cloud and Big Data Computing, Intl Conf on Cyber Science and Technology Congress (DASC/PiCom/CBDCCom/CyberSciTech) (pp. 593-600). IEEE. August 2020.
- [19] G. Hong, Z. Yang, S. Yang, L. Zhang, and Y. Nan "How you get shot in the back: A systematical study about cryptojacking in the real world." Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security. ACM, 2018.
- [20] V. Marchetto "An Investigation of Cryptojacking: Malware Analysis and Defense Strategies." Journal of Strategic Innovation and Sustainability 14.1 (2019): 66-80.
- [21] D. Oktavianto Digit, and I. Muhandianto. Cuckoo malware analysis. Packt Publishing Ltd, 2013.
- [22] R. A. Fink, A. T. Sherman, A. O. Mitchell, and D. C. Challener "Catching the cuckoo: Verifying tpm proximity using a quote timing side-channel." In International Conference on Trust and Trustworthy Computing, pp. 294-301. Springer, Berlin, Heidelberg, 2011.



A Dynamic Graph-Based Systems Framework for Modeling, and Control of Cyber-Physical Systems Typified by Buildings

Fadel Lashhab*, Musbah Abdulgader[‡], and Omar A. Zargelin[¶]

* *Department of Electrical Engineering and Computer Science, Howard University, Washington DC, USA.*

[‡] *Department of Engineering Technologies, Bowling Green State University, Bowling Green, OH, USA*

[¶] *Department of Electrical and Electronic Engineering, Al Zintan University, Al Zintan, Libya.*

Email: *fadel.lashhab@howard.edu, [‡]mabdulg@bgsu.edu, and [¶]zargelin@uoz.edu.ly

Abstract—In this paper, we present a framework for modeling certain classes of cyber-physical systems using graph-theoretic thinking augmented by ideas from the field of behavioral systems theory. The cyber-physical systems we consider are typified by buildings. We show that the thermal processes associated with a building can be represented as a graph in which (1) the node variables (temperature and heat flows) are governed by a dynamic system and (2) interconnections between these nodes (walls, doors, windows) are also described by a dynamic system. In general, we call a collection of such nodes and interconnections a dynamic graph (dynamic consensus network). Motivated by building thermal example, we present a framework for dynamic graphs and dynamic consensus networks. This framework introduces the idea of dynamic degree, adjacency, incident, and Laplacian matrices in a way that naturally extends these concepts from the static case. From this, we can easily define equivalent concepts of the dynamic consensus networks. Then we show how a behavioral systems approach can be used to develop kernel relationships between all system variables in dynamic graphs as typified by building thermal models. We discuss how such relationships can be used to analyze these systems' properties, focusing on controllability. The ideas developed for dynamic graph theory lead to developing a controllability analysis methodology for dynamic consensus networks in conjunction with the behavioral approach. We then developed the controllability conditions for the general dynamic networks, such as identical LTI nodes with dynamic edges or even in the more general case with heterogeneous nodes.

Keywords—Dynamic graphs, dynamic consensus networks, dynamic Laplacian, Controllability analysis; Behavioral approach.

I. INTRODUCTION

In this paper, we present a vision for the analysis and design of a class of cyber-physical systems using graph-theoretic ideas and introducing the perspective of behavioral systems theory. We are motivated by the energy-efficient control of buildings.

Our fundamental view of a building is of overlapping, interacting networks, as shown in Fig. 1. In this diagram, we depict the dominant phenomenon that contributes to the energy use of a building as networks (or graphs). The networks are made up of nodes that each represent a distinct subsystem.

For example, in the thermal or human networks, the nodes may represent rooms, while in the control network, a node is a sensor, actuator, or computational unit. The links between nodes indicate variable sharing, such as the flow of people in the human network between rooms through hallways and doors or the flow of heat between rooms in the thermal network through walls and doors. Smaller circles in Fig. 1 indicate links between networks. Note that typical graph-based networks assume links that are in some way constant, but as we will see, in some networks, such as the building thermal network, links between nodes may be dynamic.

Control of distributed systems, such as shown in Fig. 1 is a currently active area of research within the field of control systems. By a distributed system, we mean one with many inputs and outputs, possibly spatially-distributed dynamics, and a decentralized decision and control architecture, with restrictions on communication between computational nodes. The current state of the art has focused primarily on homogeneous systems. However, a building may be viewed as a composite system where a physical process (the structure itself) has been augmented with a hardware infrastructure (sensors and actuators) and a cyber-infrastructure (communication and decision nodes). Such overlaid heterogeneous systems with constrained connectivity and interaction between the different layers present challenges and system optimization and control opportunities. What is needed are ways to reason about discrete, multi-attribute heterogeneous entities (such as cyber-systems) and continuous, heterogeneous processes (such as physical phenomena) operating on a hierarchy of layered graphs related to each other through a set of mappings or transformations.

In this paper, we consider methods for studying distributed systems that are heterogeneous and possibly spatially-varying. Though a building can be seen as a set of interconnected networks, we consider only the thermal network. We begin by showing how a building's thermal processes can be modeled as a graph whose node variables are temperature and heat flows and whose interconnections are walls, doors, windows, etc. In our graphical representation of a building, both the nodes and

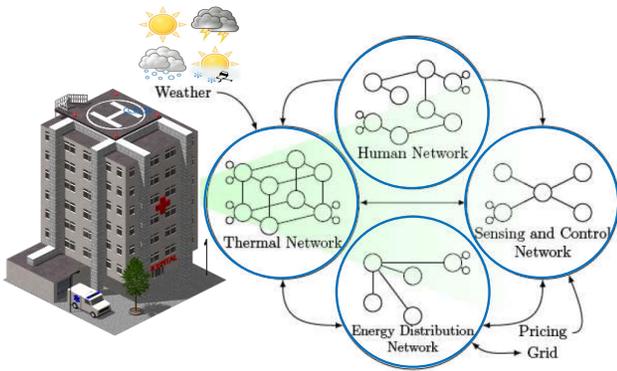


Fig. 1. A building as a collection of interacting networks.

the interconnections can be (heterogeneous) dynamic systems. We call this a *dynamic graph* (or network). For such systems, we show that the relationships between the node variables reduce to the traditional graph Laplacian in the steady-state, so that consensus variable convergence can be obtained by discussing the steady-state properties of the system. We then show how a behavioral systems approach can develop kernel relationships between all system variables in dynamic graphs typified by building thermal models. Using these kernel relationships, we consider the controllability analysis of such systems.

The idea of consensus in networking has received significant attention due to its wide array of applications in robotics, transportation, sensor networking, communication networking, biology, and physics. This paper aims to study a generalization of consensus problems whereby the weights of network edges are no longer static gains. Instead, they are dynamic systems, leading to the notion of *dynamic consensus networks*.

The network topology is static for the consensus networks, meaning that there are no dynamics in the interconnections between the nodes ($\lambda_{ij} = \text{constant} \geq 0$), and the nodes are assumed to be integrators [1]. Thus, static consensus problems can be written in the time domain for each node $i = 1, 2, \dots, n$ as

$$\dot{x}_i = \sum_{j \in \mathcal{N}_i} \lambda_{ij}(x_j(t) - x_i(t)). \tag{1}$$

The continuous time linear consensus protocol (1) can be written in matrix form as:

$$\dot{x}(t) = -Lx(t), \tag{2}$$

where $x(t) = [x_1(t), x_2(t), \dots, x_n(t)]^T$ and L , the graph's Laplacian matrix $L = [l_{ij}]$, is defined by

$$l_{ij} = \begin{cases} \sum_{j \in \mathcal{N}_i} \lambda_{ij} & i = j \\ -\lambda_{ij} & i \neq j \text{ and } (i, j) \in \mathcal{E} \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

For the multi-agent consensus problem, suppose that N agents evolve their individual beliefs $x_i \in \mathbb{R}^1$ about a so-called global consensus variable x using communications with their nearest neighbors according to the consensus protocol

(1). A key result is that the solution of $\dot{x}(t) = -Lx(t)$ gives $x_i \rightarrow x^*$ if the static graph is connected [1]. This specific fact has been the basis of much of the literature related to consensus problems.

There have been many engineering scientists in the past years involved in the controllability of dynamic consensus networks. The focus was on controlling dynamic consensus networks under the leader-follower approach, where some nodes are considered leaders, and other nodes are followers. This approach aims to transfer followers' trajectories from an initial position to the desired position (set-point) by adequately selecting the leaders' trajectory. Many authors [24], [25], [26], [27], and [28] have considered this framework by using some algebraic methods and the eigenvalues and eigenvectors of the dynamic Laplacian matrix. Other researchers also investigated the controllability using graphic tools such as the graph's equitable partition [27] and symmetry properties [26]. These graphical tools are built based on the graph's configuration and topology associated with the consensus network. The controllability investigation using the minimum energy for static consensus networks using the first-order system formulated and proposed in [29], and [30]. This paper will investigate the controllability for dynamic consensus networks with edges (links) of rational dynamical systems.

Several researchers have already studied controllability analysis for consensus networks with static topology. Most of these studies have investigated the effect of the static topology on the controllability of consensus networks. The authors [11] introduced a graph-theoretic characterization of static networks' structural controllability with a single leader. They showed that a static network with a switching topology is structurally-controllable if the union graph of the underlying static topologies is connected. In [12], the controllability investigated using the graph's size and connectivity. Controllability for leader-based, multi-agent systems analyzed in [13], and [31] based on connectivity and the null space of the leader and followers' incidence matrices. Controllability using the graph symmetry, and equitable partition properties addressed in [14]. The paper [23] formulated an equivalent data-driven Hautus-type test for a general input/output system that assumes no knowledge of the system's state. The authors' work proposed in this paper also provided an algorithm for data-driven verification of controllability of the system. They used the singular value decomposition of the Hankel matrix. A multi-vehicle system's consensus problem was proposed and analyzed by [32] with a time-varying reference state. Under the condition, only a portion of the vehicles can access the reference state in this problem. Those vehicles might not have the ability to share the information with the other vehicles in the team. Although their paper focused on developing an algorithm for investigating the consensus conditions for a directed fixed information-exchange topology, so it is useful to extend this algorithm to directed switching information-exchange topologies. In our article, the topology (edges) that describes the interconnections between nodes is considered time-varying rational transfer functions. Investigating the consensus conditions and the controllability for a multi-vehicle system might be one of the motivating application of this work.

The consensus protocol such as in (1) and its variants have been studied extensively in the literature. They have been applied in many areas, notably for formation motion control when the agents are mobile. However, the consensus paradigm is restrictive in several ways. Notice that we have interpreted the consensus problem as having integrating nodes and static weights. One might ask: what if the weights were also transfer functions? What if the nodes are more than integrators? In the next section, we have considered this question, motivated by modeling heat transfer in buildings [2]–[5]. By the notation "dynamic systems," we mean that linear ordinary differential equations (LODEs) are described as relationships between the system variables. We call such networks *dynamic consensus networks* because all the node variables converge to a common value called a consensus value under some conditions.

The paper is organized as follows: In Section II, we present a general framework for a dynamic consensus network. We present a detailed study of modeling thermal processes in buildings as directed, dynamic graphs, beginning with a simple two-room model and transitioning to a model with multiple interconnected rooms. Section III presents a framework for dynamic graphs and dynamic consensus networks. This framework introduces the idea of dynamic degree, adjacency, incident, and Laplacian matrices in a way that naturally extends these concepts from the static case. From this, we can easily define equivalent concepts of dynamic interconnection matrices and dynamic consensus networks. In Section IV, we use the established aspects and properties of the defined dynamic graph theory in conjunction with the behavioral approach to developing a controllability-analysis methodology for dynamic networks. Because of scalability in such dynamic networks, the controllability conditions based on node and network topology for the overall dynamic networks were developed in Section V.

II. MODELING A THERMAL PROCESS IN A BUILDING AS A DIRECTED DYNAMIC GRAPH

This Section first presents examples showing how a dynamic graph can arise in applications and then give a general framework for a dynamic consensus network. We present a detailed study of modeling thermal processes in buildings as directed, dynamic graphs, beginning with a simple two-room model and transitioning to a model with multiple interconnected rooms. Motivated by this example, we then present a framework for dynamic graphs and dynamic consensus networks. This framework introduces the idea of dynamic degree, adjacency, incident, and Laplacian matrices in a way that naturally extends these concepts from the static case. From this, we can easily define equivalent concepts of dynamic interconnection matrices and dynamic consensus networks.

Historically, there has always been a recognized need to model the energy processes within buildings. Typical examples of this modeling application are sizing HVAC equipment, determining energy usage performance, and optimizing energy management in a building through persistent control. Current state-of-the-art methods include modeling packages, such as Energy Plus [6], that allows users to specify a building's

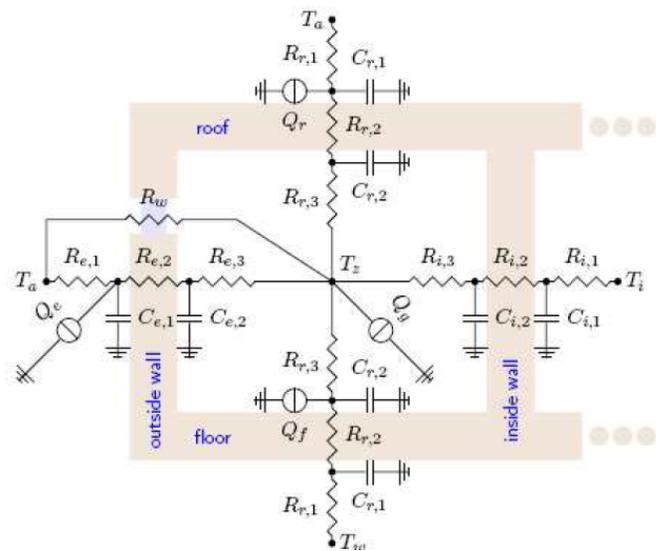


Fig. 2. Thermal model of a room.

geometry, equipment, orientation, materials, and usage patterns, which are then simulated using first principles models and simulated weather data. Though undoubtedly useful for design, these computationally-complex systems may suffer from certain limitations once a building has been constructed due to significant deviations in construction, occupant use, and other specifications that cause the actual building's behavior to be quite different from the model.

At the opposite extreme, so-called black box models have been developed from observational data. Though these models can be utilized to predict future values of particular variables, they do not incorporate any structural information about the system when gathering data, resulting in the need for large amounts of data to train and suffering from the difficulty of extracting relevant information about internal physical parameters that may be of interest.

Semi-physical models resulting in an intermediate level of modeling are known as gray-box modeling. Simple modeling elements containing parameters identified using observational data are chosen and connected based upon physical insight to represent the system's actual configuration. This is commonly the modeling technique used for thermal networks, which have been used to study load-shifting and peak-reducing control in buildings [7], [8]. A typical thermal network model for a single room is shown in Fig. 2, which was adapted from [8]. These networks of (analogous) thermal resistors and capacitors model different building elements. To date, this has typically been performed at a very coarse level, sometimes by combining multiple rooms into one practical room per zone. In [8], a gray-box model for an experimental building was created by utilizing measurements of weather, room temperature, and room air supply and flow. This model was used to predict the effects of a demand-limiting control strategy that was later validated experimentally.

This section uses a single-room model as shown in Fig. 2 from [8] as the basis for a node and its interconnections to

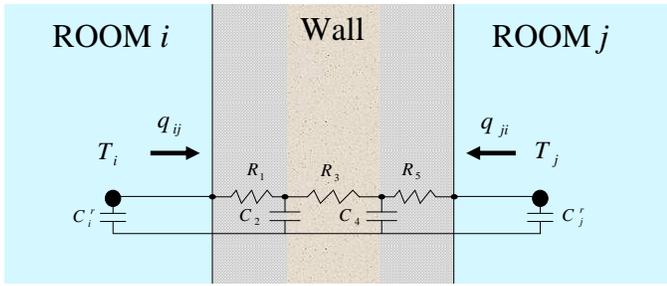


Fig. 3. Two rooms connected by a wall using the 3R2C model.

other nodes to build up a dynamic graph representation of a building’s thermal processes. First, we consider two rooms connected by a wall. We then illustrate how several such nodes may be interconnected, using the example of a hypothetical four-room building, with analysis provided of the resulting model that motivates the generalization in the next section.

Before proceeding, we note that the initial interest in modeling thermal processes in a building comes from viewing a building as a group of overlapping, interacting networks. In the thermal network, the nodes may represent rooms, while in the control network, a node is a sensor, actuator, or computational unit. The links between nodes indicate variable information sharing, such as the heat flow between rooms through walls and doors in the thermal network. The typical, graph-based networks assume links that are in some way constant; some networks, such as a building’s thermal network, may have dynamic links between nodes, as we will see in the next section.

A. Two Rooms Connected by a Wall

Fig. 3 depicts what is called a 3R2C model in the literature [9]. We identify a room i as a node with node variable T_i , the lumped room temperature, Q_i^{in} , the input heat flow (a manipulated variable, not shown), and q_{ij} , the heat flow out of the room through walls or doors or windows (of course with this convention, if $q_{ij} < 0$ then this is heat flow into the room). The parameter C_i^r is the thermal capacity (mass) of the room i .

The interconnection between the two rooms is a wall, which is represented analogously by an electrical circuit with three resistors and two capacitors, a simplification of the model in Fig. 3. The capacitors C_2 and C_4 can be thought of as the heat storage capacity of the wall’s materials, which could be different on each side of the wall. A complete model adds a capacitor to represent the insulation properties in addition to that of the wall’s board materials. For the resistors, R_3 represents the heat dissipation inside the wall, while R_1 and R_5 represent the heat dissipation from each room to the inside of the wall.

As shown in [9], the heat flows in Fig. 3 can be written as:

$$\begin{bmatrix} q_{ij} \\ q_{ji} \end{bmatrix} = \frac{1}{B_{ij}(s)} \begin{bmatrix} A_{ij}(s) & -D_{ij}(s) \\ -D_{ij}(s) & A_{ji}(s) \end{bmatrix} \begin{bmatrix} T_i \\ T_j \end{bmatrix} \quad (4)$$

where

$$\begin{aligned} A_{ij}(s) &= 1 + a_1^{ij}s + a_2^{ij}s^2 \\ A_{ji}(s) &= 1 + a_1^{ji}s + a_2^{ji}s^2 \\ B_{ij}(s) &= b_0^{ij} + b_1^{ij}s + b_2^{ij}s^2 \\ D_{ij}(s) &= 1 + d_1^{ij}s + d_2^{ij}s^2 \end{aligned}$$

and

$$\begin{aligned} a_1^{ij} &= C_4R_5 + C_2R_3 + C_2R_5 \\ a_2^{ij} &= C_4C_2R_5R_3 \\ a_1^{ji} &= C_4R_1 + C_4R_3 + C_2R_1 \\ a_2^{ji} &= C_4C_2R_3R_1 \\ b_0^{ij} &= R_5 + R_3 + R_1 \\ b_1^{ij} &= C_4R_5R_1 + C_2R_3R_1 + C_2R_5R_1 + C_4R_5R_3 \\ b_2^{ij} &= C_4C_2R_5R_3R_1 \\ d_1^{ij} &= d_2^{ij} = 0 \end{aligned}$$

Here, s is the independent variable of the Laplace transform, which can be interpreted as $s \doteq \frac{d}{dt}(\cdot)$. As noted in [9], we can interpret $G_x(s) = A_{ij}(s)/B_{ij}(s)$ as the external conduction of the wall, $G_y(s) = D_{ij}(s)/B_{ij}(s)$ as the cross-conduction of the wall, and $G_z(s) = A_{ji}(s)/B_{ij}(s)$ as the internal conduction of the wall.

From (3), the nodal equation can be written as:

$$C_i^r \frac{dT_i}{dt} = Q_i^{in} - q_{ij} \quad (5)$$

Combining (4) and (5) gives:

$$\begin{bmatrix} C_i^r s & 0 \\ 0 & C_j^r s \end{bmatrix} \begin{bmatrix} T_i \\ T_j \end{bmatrix} = \begin{bmatrix} Q_1^{in} \\ Q_2^{in} \end{bmatrix} - \frac{1}{B_{ij}(s)} \begin{bmatrix} A_{ij}(s) & -D_{ij}(s) \\ -D_{ij}(s) & A_{ji}(s) \end{bmatrix} \begin{bmatrix} T_i \\ T_j \end{bmatrix} \quad (6)$$

To motivate later analysis, notice that in the absence of any external heat inputs (i.e., $Q_i^{in} = 0$), we can rewrite the previous equation as:

$$\begin{bmatrix} sC_i^r T_i(s) \\ sC_j^r T_j(s) \end{bmatrix} = -\frac{1}{B_{ij}(s)} \begin{bmatrix} A_{ij}(s) & -D_{ij}(s) \\ -D_{ij}(s) & A_{ji}(s) \end{bmatrix} \begin{bmatrix} T_i \\ T_j \end{bmatrix}, \quad (7)$$

which defines the relationship between the temperatures in two rooms using the 3R2C model.

It is useful to separate (7) as

$$\begin{aligned} T_i(s) &= -\frac{1}{C_i^r s} \left[\frac{A_{ij}(s)}{B_{ij}(s)} T_i(s) - \frac{D_{ij}(s)}{B_{ij}(s)} T_j(s) \right] \\ &= -\frac{1}{s} \left[\lambda_{ij}^S(s) T_i(s) - \lambda_{ij}^C(s) T_j(s) \right], \end{aligned} \quad (8)$$

where $\lambda_{ij}^S(s) = \frac{A_{ij}(s)}{B_{ij}(s)}$ is a self-correction weight and $\lambda_{ij}^C(s) = \frac{D_{ij}(s)}{B_{ij}(s)}$ is a cross-correction weight. Note that we assume C_i^r in (8) is equal to the unity for simplicity. Comparing this to static-weights consensus networks, we see that this appears similar to the equation of a two-node consensus network, in which the nodes are integrators. The difference is that there is a separate weighting on the terms $T_i(s)$ and $T_j(s)$, and these weights are dynamic.

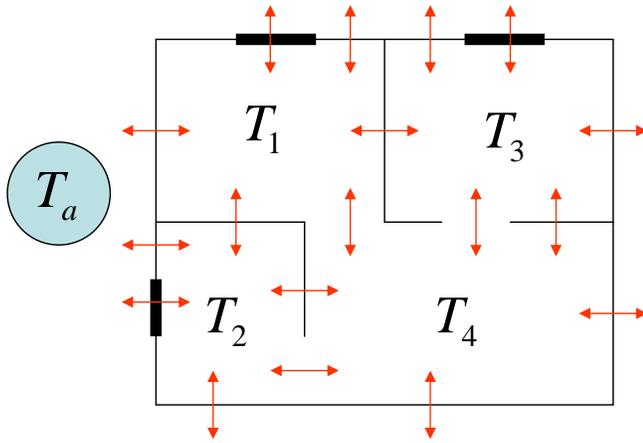


Fig. 4. A hypothetical four-room example.

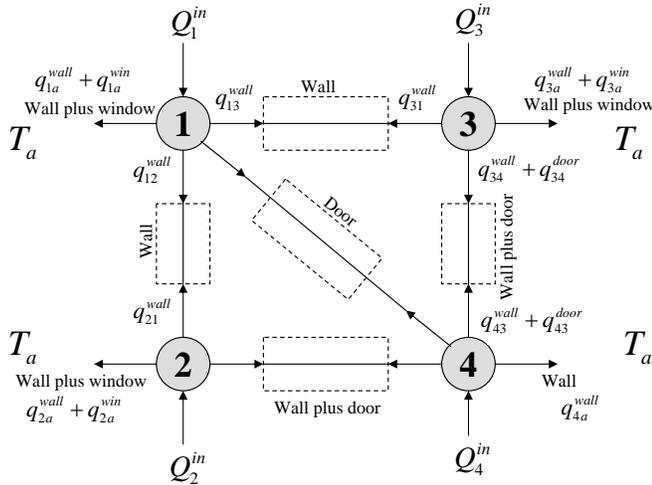


Fig. 5. Heat flow network corresponding to the four-room example.

B. Several Interconnected Rooms

This subsection uses the previous subsection’s expressions to develop a building model with several interconnected rooms with different possible pathways between each room and the outside environment. Ideas are developed for the specific hypothetical four-room building shown in Fig. 4 in which each room has several neighbors with which it is interconnected. One such neighbor is always the external environment whose variable is denoted T_a with ‘a’ referring to the ambient. Pathways include walls, doors, and windows. The corresponding graph for this example is shown in Fig. 5. There are several heat flows that are not shown, including $q_{14} = q_{14}^{door}$, $q_{41} = q_{41}^{door}$, $q_{24} = q_{24}^{wall} + q_{24}^{door}$ and $q_{42} = q_{42}^{wall} + q_{42}^{door}$.

In developing a model for this system, we modify (5) to sum the energy losses through all pathways connected to a node, resulting in:

$$C_i^r \frac{dT_i}{dt} = Q_i^{in} - \sum_{j \in \mathcal{N}_i} \sum_{k_j \in \mathcal{P}_j} q_{ij}^{k_j} \tag{9}$$

where \mathcal{N}_i is the set of neighbors to which a node i is connected and \mathcal{P}_j is the set of pathways k_j associated with any neighbor j of node i . We note that $q_{ij} = \sum_{k_j \in \mathcal{P}_j} q_{ij}^{k_j}$.

For building thermal analysis, there may be several different types of interconnection elements, though they will all have the basic format

TABLE I
HYPOTHETICAL FOUR ROOM EXAMPLE

Node	Neighbors	Paths	Coefficients
1	2	1-wall	A_{12}, B_{12}
		2-wall	A'_{1a}, B'_{1a}
	3	1-window	R_{w1a}
		1-wall	A_{13}, B_{13}
2	1	1-door	$R_{d14} = B_{14}$
		1-wall	D_{12}, B_{12}
	a	2-wall	A'_{2a}, B'_{2a}
3	a	1-window	$R_{w2a} = B_{2a}$
		2-wall	A'_{24}, B'_{24}
	4	1-door	$R_{d24} = B_{24}$
4	1	1-wall	D_{13}, B_{13}
		2-wall	A'_{3a}, B'_{3a}
	a	1-window	$R_{w3a} = B_{3a}$
		2-wall	A'_{34}, B'_{34}
2	2-wall	1-door	$R_{d34} = B_{34}$
		1-door	$R_{d41} = B_{41}$
	3	1-wall	A'_{1a}, B'_{1a}
4	2	2-wall	D_{24}, B_{24}
		1-door	$R_{d42} = B_{42}$
	3	2-wall	D'_{34}, B'_{34}
		1-door	$R_{d43} = B_{43}$

of (4). Because there is negligible energy storage in doorways and windows, when these are the sole interconnection elements between rooms, we use a single R model, so that (4) is expressed with $b_0^{ij} = R$ with all other variables being set to zero. To make the notation a bit more uniform, for the common case when a door or window is in parallel with a wall, the interconnection transfer function matrix is the sum of the single R model and the 3R2C model. That is, the model would be given by A_{ij}, A_{ji}, B_{ij} , and D_{ij} that satisfies:

$$\begin{bmatrix} \frac{A_{ij}(s)}{B_{ij}(s)} & -\frac{D_{ij}(s)}{B_{ij}(s)} \\ -\frac{D_{ij}(s)}{B_{ij}(s)} & \frac{A_{ji}(s)}{B_{ij}(s)} \end{bmatrix} = \begin{bmatrix} \frac{A'_{ij}(s)}{B'_{ij}(s)} & -\frac{1}{B'_{ij}(s)} \\ -\frac{1}{B'_{ij}(s)} & \frac{A'_{ji}(s)}{B'_{ij}(s)} \end{bmatrix} + \begin{bmatrix} \frac{1}{R} & -\frac{1}{R} \\ -\frac{1}{R} & \frac{1}{R} \end{bmatrix}, \tag{10}$$

where the primed variables represent the 3R2C model, and the unprimed variables represent the resulting parallel connection. In the expressions below, we assume that this computation has been done and the resulting unprimed coefficients can be easily calculated and are thus omitted here. Note that in the case of a door or window that is parallel to a wall, the coefficients d_1^{ij} and d_2^{ij} associated with the unprimed variables are non-zero.

Table I summarizes the neighbors for each node and the pathways between each node and each of its neighbors for this example. The table also identifies the coefficients used in the transfer matrix describing the interconnection between each pair of neighbors where the various polynomials A_{ij}, B_{ij}, D_{ij} (when there is no parallel door or wall) or $A'_{ij}, B'_{ij}, D'_{ij}$ (before being combined in parallel with any doors or windows) are defined as above in (4).

Combining (4) and (9) for the configuration shown in Fig. 5 with the parameters shown in Table I and defining the vectors

$$T(s) = [T_1(s) \ T_2(s) \ T_3(s) \ T_4(s)]^T, \\ Q^{in}(s) = [Q_1^{in}(s) \ Q_2^{in}(s) \ Q_3^{in}(s) \ Q_4^{in}(s)]^T,$$

we can easily show that:

$$sT(s) = Q^{in}(s) - L(s)T(s), \tag{11}$$

where the matrix $L(s) = [L_{ij}(s)]$ is given as:

$$L_{ij}(s) = \begin{cases} \sum_{j \in \mathcal{N}_i} \lambda_{ij}^S(s) & i = j \\ -\lambda_{ij}^C(s) & i \neq j \text{ and } (i, j) \in \mathcal{E} \\ 0 & \text{otherwise,} \end{cases} \tag{12}$$



or $L(s)$ is given as (13). We will refer to $L(s)$ defined in this way as a *dynamic Laplacian matrix*. For the graph topology shown in Fig. 5, the dynamic Laplacian matrix has the form shown in (13). In the previous work [2], We have shown that when the weight matrices $\lambda_{ij}(s)$ satisfy certain assumptions, $-L(s)$ can be viewed as a *dynamic interconnection matrix*, allowing the demonstration of consensus.

Notice that we can redraw Fig. 5 as shown in Fig. 6, where

$$\lambda_{ij}(s) = [\lambda_{ij}^S(s) \quad -\lambda_{ij}^C(s)] \quad (14)$$

$$= \left[\left(\frac{A_{ij}}{B_{ij}} \right)_w + \frac{1}{R_{ij}^d} \quad - \left(\frac{1}{B_{ij}} \right)_w - \frac{1}{R_{ij}^d} \right], \quad (15)$$

The graph shown in Fig. 6 will be referred to as a *dynamic graph* or a *dynamic consensus network*. Then applying the definition of the dynamic Laplacian (12) for the dynamic graph Fig. 6 we get

$$L(s) = \begin{bmatrix} \sum_{j=2,3,4} \lambda_{1j}^S(s) & -\lambda_{12}^C(s) & -\lambda_{13}^C(s) & -\lambda_{14}^C(s) \\ -\lambda_{21}^C(s) & \sum_{j=1,4} \lambda_{2j}^S(s) & 0 & -\lambda_{24}^C(s) \\ -\lambda_{31}^C(s) & 0 & \sum_{j=1,4} \lambda_{3j}^S(s) & -\lambda_{34}^C(s) \\ -\lambda_{41}^C(s) & -\lambda_{42}^C(s) & -\lambda_{43}^C(s) & \sum_{j=1,2,3} \lambda_{4j}^S(s) \end{bmatrix}, \quad (16)$$

which reduces to (13) if we insert the full expressions for $\lambda_{ij}^C(s)$ and $\lambda_{ij}^S(s)$ defined in (14). This leads us to consider the idea of dynamic consensus networks.

Figure II-B shows a simple simulation of (9) for the case when $Q_i^{in} = 0$ and the scalar $T_a = 80$ for a nominal set of parameters available from the authors upon request (omitted here in the interest of space). As intuitively expected, all temperatures converge to the ambient temperature, as the external environment has infinite capacity and there is no energy input or removal. This can also be seen by examining (9) at steady-state with $Q_{in} = 0$. Further noting that $L(0)$ has the form of a classic graph Laplacian matrix, with row sum equal to zero (and in this case column sum equal zero as well), we can argue that $T^{ss} = T_a$ is a unique solution.

We also [10] consider another example that motivated a generalization of the static consensus problem (1), modeling the load frequency control (LFC) network of an electrical power grid as a dynamic consensus network. We consider the following network:

$$Y_i(s) = \frac{1}{s} \sum_{j \in \mathcal{N}_i} G_i(s) a_{ij} (Y_j(s) - Y_i(s)), \quad (17)$$

$i = 1, \dots, N$, which can be viewed as a single-integrator consensus network with dynamic interconnection coefficients $G_i(s) a_{ij}$. In the grid's LFC network, each system's output is the phase of its voltage, which is the integration of the angular velocity. The interconnection is power exchanges among the individual systems through transmission lines dependent on phase differences.

Based on the dynamics of a network's nodes and their topology, several consensus problems can be specified. This paper focuses on two types of dynamic consensus networks: directed and undirected. The dynamic consensus networks studied are:

- **Dynamic Network 1:** Directed dynamic networks with integrator nodes and dynamic edges:

$$\dot{x}_i(t) = - \sum_{j \in \mathcal{N}_i} [\lambda_{ij}^S(t) * x_i(t) - \lambda_{ij}^C(t) * x_j(t)],$$

or,

$$x_i(s) = -\frac{1}{s} \sum_{j \in \mathcal{N}_i} [\lambda_{ij}^S(s) x_i(s) - \lambda_{ij}^C(s) x_j(s)] \quad (18)$$

- **Dynamic Network 2:** Undirected dynamic networks with integrator nodes and strictly-positive-real (SPR) transfer function edges:

$$\dot{x}_i(t) = - \sum_{j \in \mathcal{N}_i} [\lambda_{ij}(t) * (x_i(t) - x_j(t))],$$

or,

$$x_i(s) = -\frac{1}{s} \sum_{j \in \mathcal{N}_i} [\lambda_{ij}(s) (x_i(s) - x_j(s))] \quad (19)$$

- **Dynamic Network 3:** Undirected dynamic networks with identical nodes and dynamic edges:

$$x_i(t) = -p(t) * \sum_{j \in \mathcal{N}_i} [\lambda_{ij}(t) * (x_i(t) - x_j(t))],$$

or,

$$x_i(s) = -p(s) \sum_{j \in \mathcal{N}_i} [\lambda_{ij}(s) (x_i(s) - x_j(s))] \quad (20)$$

- **Dynamic Network 4:** Undirected dynamic networks with heterogeneous nodes and dynamic edges:

$$x_i(t) = -p_i(t) * \sum_{j \in \mathcal{N}_i} [\lambda_{ij}(t) * (x_i(t) - x_j(t))],$$

$$x_i(s) = -p_i(s) \sum_{j \in \mathcal{N}_i} [\lambda_{ij}(s) (x_i(s) - x_j(s))] \quad (21)$$

Assumptions: For the dynamic networks (18 - 21), we make the following assumptions:

- 1) The node and edge processing in the proposed dynamic networks (18 - 21) are linear, time-invariant LTI.
- 2) The dynamic topologies consist of dynamic edges $\lambda_{ij}(s)$ modeled as transfer functions. For the second proposed dynamic network (19), we assume the edges' dynamics are strictly positive real (SPR) transfer functions.
- 3) The topology of a network can be directed or undirected. The first dynamic network (18) uses a directed topology, whereas the second dynamic network (19) uses an undirected topology.
- 4) Depending on the application, the flow is modeled differently. For instance, $[\lambda_{ij}^S(s)x_i(s) - \lambda_{ij}^C(s)x_j(s)]$ and $[\lambda_{ij}(s)(x_i(s) - x_j(s))]$ are two different ways of modeling flow, as is indicated by the previous Section. The difference between these two cases is illustrated in (18, 19). These flow models are rooted in the types of dynamic networks to be modeled per the motivation for each network. The first dynamic network (18) is based upon modeling buildings' thermal processes as directed dynamic graphs. In contrast, the second dynamic network (19) is based upon the motivation of modeling micro-grids of power systems as undirected dynamic graphs.
- 5) The nodes' dynamics can be integrators (18, 19) or more general dynamics (20, 21).
- 6) The nodes' dynamics and the edges can be identical (20) or heterogeneous (21).
- 7) These models are often autonomous, meaning no input flows into the dynamic consensus networks. However, we add inputs and disturbances to the proposed dynamic consensus networks' general forms in some problems.

III. DYNAMIC GRAPHS DEFINITIONS

The previous work [2], [10] showed how two different phenomena could be modeled in a graph whose edges are transfer functions (i.e., dynamic systems). In this section, from the models' motivation developed in the previous work, we generalize all the typical notations from (static) graph theory to the dynamic case. The development here parallels the notations in the static graph.

Consider the example of a directed, dynamic graph shown in Fig. 8. Such graphs can be described as a set of nodes (or vertices) $\mathcal{N} =$



$$L(s) = \begin{bmatrix} \frac{1}{R_{14}^d} + \sum_{j=2,3} \frac{A_{1j}}{B_{1j}} & -\frac{1}{B_{12}} & -\frac{1}{B_{13}} & -\frac{1}{R_{14}^d} \\ -\frac{1}{B_{21}} & \frac{1}{R_{24}^d} + \sum_{j=1,4} \frac{A_{2j}}{B_{2j}} & 0 & -\frac{1}{B_{24}} - \frac{1}{R_{24}^d} \\ -\frac{1}{B_{31}} & 0 & \frac{1}{R_{34}^d} + \sum_{j=1,4} \frac{A_{3j}}{B_{3j}} & -\frac{1}{B_{34}} - \frac{1}{R_{34}^d} \\ -\frac{1}{R_{41}^d} & -\frac{1}{B_{42}} - \frac{1}{R_{42}^d} & -\frac{1}{B_{43}} - \frac{1}{R_{43}^d} & \frac{1}{R_{41}^d} + \frac{1}{R_{42}^d} + \frac{1}{R_{43}^d} + \sum_{j=2,3} \frac{A_{4j}}{B_{4j}} \end{bmatrix} \quad (13)$$

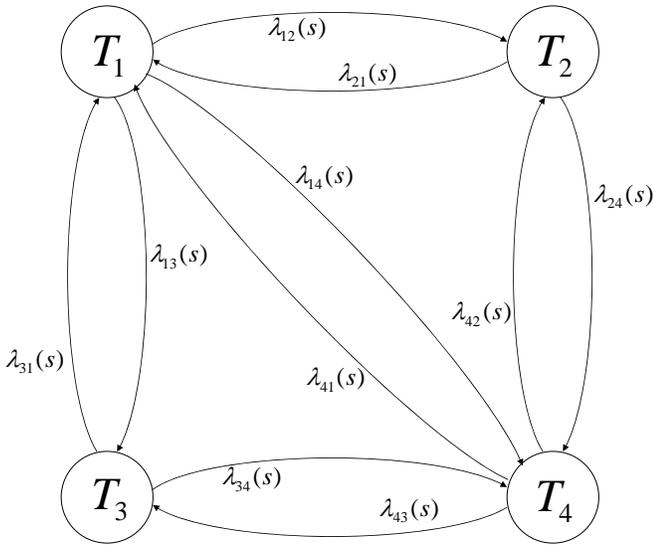


Fig. 6. A hypothetical four-room example as a dynamic consensus network.

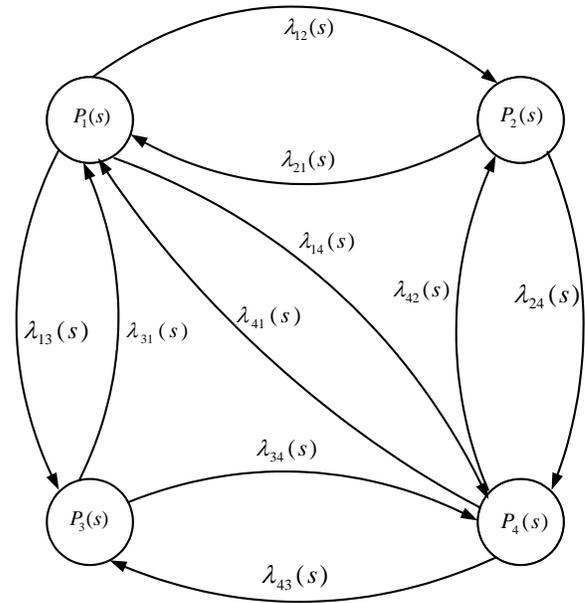


Fig. 8. Directed-Dynamic Graph.

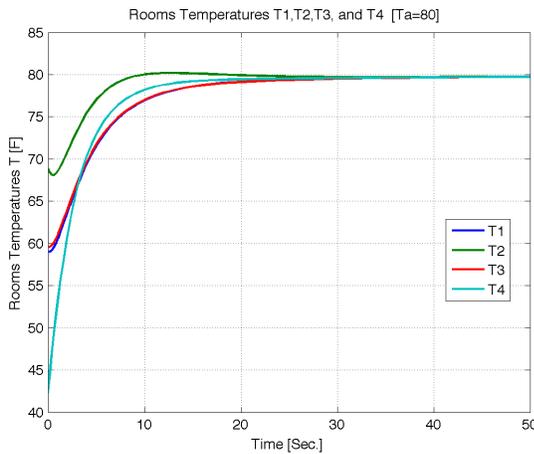


Fig. 7. Example simulation.

$\{n_i\}$ connected by a set of edges $\mathcal{E}(s) = \{(n_i, n_j) : n_i, n_j \in \mathcal{N}\}$. Each edge is modeled as a transfer function $\lambda_{ij}(s)$. More formally, we can say that each edge $\lambda_{ij}(s) \in \mathcal{R}(s)$, where $\mathcal{R}(s)$ denotes the set of all complex-valued functions analytic in the open right-half complex plane (real rational functions). We assume there are no self-loops associated with any node. If there is an edge between nodes n_i and n_j , we say these nodes are adjacent (or neighbors). We denote the neighbors of node n_i as $\mathcal{N}_i = \{j : (n_i, n_j) \in \mathcal{E}(s)\}$. A path

between two nodes is a sequence of edges by which it is possible to move along the arc sequence from one of the nodes to the other. If there is at least one node with at least one path to every other node, the graph is connected.

Later, we also view a node as implementing a transfer function that produces the node variable ($P_i(s)$ for $i = 1, 2, \dots, N$, where N is the number of nodes in the dynamic graph) by processing the incoming and outgoing flows.

As in the static case, the edges $e_{ij}(s)$ are ordered by the edge originating from node $P_i(s)$, known as the tail node, and terminating at node $P_j(s)$, known as the head node, which can be identified by the presence of an arrowhead. If the dynamics of the nodes are different, we call such dynamic networks heterogeneous. If the nodes have the same dynamics, the network is called homogeneous. A particular case is when we have integrator nodes with dynamic edges.

Each node $P_i(s)$ in a directed dynamic graph, such as Fig. 8, is associated with a dynamic degree $\nu_i(s)$ representing the total sum of the dynamic edge weights that are connected to the node i . More specifically, each node has a dynamic in-degree $\nu_i^{in}(s)$ and a dynamic out-degree $\nu_i^{out}(s)$ representing the sum of the dynamic edge weights of the incoming and outgoing edges, respectively. Clearly $\nu_i(s) = \nu_i^{in}(s) + \nu_i^{out}(s)$. From these dynamic degree definitions, we can define three different dynamic degree matrices:

- 1) The dynamic in-degree matrix $D^{in}(s) = \text{diag}(\nu_i^{in}(s))$.
- 2) The dynamic out-degree matrix $D^{out}(s) = \text{diag}(\nu_i^{out}(s))$.
- 3) The dynamic degree matrix $D(s) = \text{diag}(\nu_i(s))$.

Notice that $D(s) = D^{in}(s) + D^{out}(s)$.



If a dynamic edge $e_{ij}(s)$ exists between two nodes i and j , these nodes are considered to be adjacent and are known as neighbors, and are denoted for a node $P_i(s)$ by $N_i = \{j : (n_i, n_j) \in \mathcal{E}(s)\}$. As before, neighbors can be distinguished based upon whether they are associated with incoming or outgoing arcs. Thus, we can define three dynamic adjacency matrices:

1) The incoming dynamic adjacency matrix $A^{in}(s) = [a_{ij}^{in}(s)]$, is defined by

$$a_{ij}^{in}(s) = \begin{cases} \sum_{j \in N_i} e_{ij}(s); (\text{coming into } n_i \text{ from } n_j) & \text{if } i \neq j \\ 0 & \text{otherwise.} \end{cases}$$

2) The dynamic outgoing adjacency matrix $A^{out}(s) = [a_{ij}^{out}(s)]$, is defined by

$$a_{ij}^{out}(s) = \begin{cases} \sum_{j \in N_i} e_{ij}(s); (\text{going out from } n_i \text{ into } n_j) & \text{if } i \neq j \\ 0 & \text{otherwise.} \end{cases}$$

3) The dynamic adjacency matrix $A(s) = [a_{ij}(s)]$, is defined by

$$a_{ij}(s) = \begin{cases} \text{total } \sum_{j \in N_i} e_{ij}(s); (\text{between } n_i \text{ and } n_j) & \text{if } i \neq j \\ 0 & \text{otherwise.} \end{cases}$$

Notice that $A(s) = A^{in}(s) + A^{out}(s)$.

Another type of dynamic matrix is the dynamic incident matrix. For the incoming incident matrix, we will define two incident matrices: one indicates the direction of the edges connected to a node, where for node $P_i(s)$ the edge $e_{ij}(s)$ is given a value based upon being disconnected, incoming, or outgoing and denoted as B_S^{in} where “S” refers to static. Another matrix captures the transfer functions of the edges and is denoted as $B_D^{in}(s)$ where “D” refers to dynamic. Thus, the dynamic and static incoming incident matrices are defined as $B_D^{in}(s) = [b_{ij-D}^{in}(s)]$, $B_S^{in} = [b_{ij-S}^{in}]$, where,

$$b_{ij-D}^{in}(s) = \begin{cases} +\lambda_{ij}(s) & \text{if arc } j \text{ enters node } n_i \\ 0 & \text{otherwise.} \end{cases}$$

$$b_{ij-S}^{in} = \begin{cases} +1 & \text{if arc } j \text{ enters node } n_i \\ 0 & \text{otherwise.} \end{cases}$$

Similarly, we can define a dynamic and static outgoing incident matrices for a dynamic graph by, $B_D^{out}(s) = [b_{ij-D}^{out}(s)]$, $B_S^{out} = [b_{ij-S}^{out}]$, where,

$$b_{ij-D}^{out}(s) = \begin{cases} -\lambda_{ij}(s) & \text{if arc } j \text{ leaves node } n_i \\ 0 & \text{otherwise.} \end{cases}$$

$$b_{ij-S}^{out} = \begin{cases} -1 & \text{if arc } j \text{ leaves node } n_i \\ 0 & \text{otherwise.} \end{cases}$$

Also, we can define the dynamic and static incident matrices for a directed dynamic graph as $B_D(s) = [b_{ij-D}(s)]$, $B_S = [b_{ij-S}]$ where

$$b_{ij-D}(s) = \begin{cases} -\lambda_{ij}(s) & \text{if arc } j \text{ leaves node } n_i \\ +\lambda_{ij}(s) & \text{if arc } j \text{ enters node } n_i \\ 0 & \text{otherwise.} \end{cases}$$

$$b_{ij-S} = \begin{cases} -1 & \text{if arc } j \text{ leaves node } n_i \\ +1 & \text{if arc } j \text{ enters node } n_i \\ 0 & \text{otherwise.} \end{cases}$$

Notice that $B_D(s) = B_D^{in}(s) + B_D^{out}(s)$ and $B_S = B_S^{in} + B_S^{out}$.

We can now give the dynamic equivalent of the static Laplacian matrix with the derived dynamic degree, adjacency, and incident matrices. The dynamic Laplacian matrix has spectral properties that indicate many facts about a graph. An undirected dynamic graph, one whose dynamic edges are not directionally-fixed, has a corresponding dynamic Laplacian matrix defined with no ambiguities by $L(s) = D(s) - A(s)$. More specifically, the dynamic Laplacian matrix is defined as $L(s) = [l_{ij}(s)]$, where

$$l_{ij}(s) = \begin{cases} \sum_{j \in N_i} \lambda_{ij}(s) & i = j \\ -\lambda_{ij}(s) & i \neq j \text{ and } (i, j) \in \mathcal{E} \\ 0 & \text{otherwise} \end{cases} \quad (22)$$

We can also define the dynamic Laplacian matrix of an undirected dynamic graph as

$$L(s) = BD(s)B^T = D(s) - A(s), \quad (23)$$

where $D(s) \in \mathbb{C}^{m \times m}$ is the dynamic degree matrix formed by the dynamic degree of the m edges, $B \in \mathbb{R}^{n \times m}$ is the static incident matrix that captures the orientations of the edges, and $A(s) \in \mathbb{C}^{n \times n}$ is the dynamic adjacency matrix.

In the static case, the definition of the Laplacian matrix for a directed graph requires adopting either an incoming or outgoing convention. Likewise, we define the dynamic Laplacian matrix utilizing the dynamic degree and dynamic adjacency matrices distinguishing between incoming and outgoing conventions. Examples include: $L^{in}(s) = D^{in}(s) - A^{in}(s)$ and $L^{out}(s) = D^{out}(s) - A^{out}(s)$. With these definitions, $L(s) = L^{in}(s) + L^{out}(s)$. Particular caution must be taken in noting that while $L = BB^T$ in the static case, $L(s) \neq B(s)B(s)^T$, $L^{in}(s) \neq B^{in}(s)B^{in}(s)^T$ and $L^{out}(s) \neq B^{out}(s)B^{out}(s)^T$. To overcome this, in the sequel, we will use $L^{out}(s)$, where the outgoing dynamic Laplacian matrix can be defined using the incident matrices as follow:

$$L^{out}(s) = D^{out}(s) - A^{out}(s), \quad (24)$$

where $D^{out}(s)$ and $A^{out}(s)$ are the dynamic-outgoing degree and adjacency matrices, respectively. These matrices can be defined in a static case using the incident matrices as $D^{out} = B^{out}B^{outT}$ and $A^{out} = -B^{out}B^{inT}$. For the dynamic graphs, $D^{out}(s) \neq B^{out}(s)B^{out}(s)^T$ because the product of the dynamic incident matrices results in a matrix where its elements are the square of the edge dynamics (i.e., $\lambda_{ij}(s)^2$). To overcome this problem, we use the dynamic and static incident matrices (one will capture the edges’ orientation, and the other will capture the edges dynamic) for defining the dynamic degree and adjacency matrices. Thus, the outgoing, dynamic degree, and adjacency matrices can be defined as:

$$\begin{aligned} D^{out}(s) &= B_D^{out}(s)B_S^{outT}, \\ A^{out}(s) &= -B_D^{out}(s)B_S^{inT}. \end{aligned} \quad (25)$$

By combining (24) and (25), the outgoing, dynamic Laplacian matrix $L^{out}(s)$ can be defined as

$$\begin{aligned} L^{out}(s) &= B_D^{out}(s)B_S^{outT} + B_D^{out}(s)B_S^{inT} \\ &= B_D^{out}(s)(B_S^{outT} + B_S^{inT}) = B_D^{out}(s)B_S^T. \end{aligned} \quad (26)$$

A similar definition can be given for $L^{in}(s)$.

To illustrate, for the example shown in Fig. 8, the associated dynamic degree, adjacency, and Laplacian matrices are given by (27). Note that all $\lambda_{ij}(s)$ in (27) are transfer functions that describe the interconnections (edges) between the nodes.



$$\begin{aligned}
 D^{out}(s) &= \begin{bmatrix} \lambda_{12}(s) + \lambda_{13}(s) + \lambda_{14}(s) & 0 & 0 & 0 \\ 0 & \lambda_{21}(s) + \lambda_{24}(s) & 0 & 0 \\ 0 & 0 & \lambda_{31}(s) + \lambda_{34}(s) & 0 \\ 0 & 0 & 0 & \lambda_{41}(s) + \lambda_{42}(s) + \lambda_{43}(s) \end{bmatrix}, \\
 A^{out}(s) &= \begin{bmatrix} 0 & \lambda_{12}(s) & \lambda_{13}(s) & \lambda_{14}(s) \\ \lambda_{21}(s) & 0 & 0 & \lambda_{24}(s) \\ \lambda_{31}(s) & 0 & 0 & \lambda_{34}(s) \\ \lambda_{41}(s) & \lambda_{42}(s) & \lambda_{43}(s) & 0 \end{bmatrix}, \\
 L^{out}(s) &= \begin{bmatrix} \lambda_{12}(s) + \lambda_{13} + \lambda_{14}(s) & -\lambda_{12}(s) & -\lambda_{13}(s) & -\lambda_{14}(s) \\ -\lambda_{21}(s) & \lambda_{21}(s) + \lambda_{24}(s) & 0 & -\lambda_{24}(s) \\ -\lambda_{31}(s) & 0 & \lambda_{31}(s) + \lambda_{34}(s) & -\lambda_{34}(s) \\ -\lambda_{41}(s) & -\lambda_{42}(s) & -\lambda_{43}(s) & \lambda_{41}(s) + \lambda_{42}(s) + \lambda_{43}(s) \end{bmatrix}. \tag{27}
 \end{aligned}$$

IV. CONTROLLABILITY ANALYSIS FOR DYNAMIC CONSENSUS NETWORKS USING THE BEHAVIORAL APPROACH

This section uses the established aspects and properties of the defined dynamic graph theory in conjunction with the behavioral approach to developing a controllability-analysis methodology for dynamic networks. Additionally, we demonstrate the dynamic interconnection topology (dynamic Laplacian matrix) in analyzing node interconnections to establish controllability.

A. Behavioral Approach

This section analyzes the dynamic interconnection topologies of the proposed dynamic networks to establish controllability using the behavioral approach to control [15]. We first describe the behavioral process and then discuss the two-room example’s controllability utilizing this approach. We then give the general result.

The behavior of a dynamic system is the collection of all possible time trajectories in the system [16], [17]. In other words, the behavior is a family of trajectories, rather than a transfer function. The behavior of a dynamic system is defined as the set of solutions of a system of linear, constant-coefficient differential equations. For example, consider a system described by the following set of differential equations:

$$\begin{aligned}
 w_1 + 2 \frac{d}{dt} w_1 + \frac{d^2}{dt^2} w_1 - w_2 - \frac{d}{dt} w_2 &= 0; \\
 -w_1 - \frac{d}{dt} w_1 + 7w_2 + 5 \frac{d}{dt} w_2 + \frac{d^2}{dt^2} w_2 - 6w_3 - \\
 4 \frac{d}{dt} w_3 &= 0; \tag{28}
 \end{aligned}$$

The dynamic system (28) can be written in matrix form as:

$$M \left(\frac{d}{dt} \right) w = 0, \tag{29}$$

where

$$M \left(\frac{d}{dt} \right) = \begin{bmatrix} \frac{d^2}{dt^2}(\cdot) + 2 \frac{d}{dt}(\cdot) + 1 & -(\frac{d}{dt}(\cdot) + 1) & 0 \\ -(\frac{d}{dt}(\cdot) + 1) & \frac{d^2}{dt^2}(\cdot) + 5 \frac{d}{dt}(\cdot) + 7 & -4 \frac{d}{dt}(\cdot) - 6 \end{bmatrix},$$

and $w = [w_1, w_2, w_3]^T$. We can also describe (28) using polynomial notation as

$$\begin{bmatrix} s^2 + 2s + 1 & -(s + 1) & 0 \\ -(s + 1) & s^2 + 5s + 7 & -4s - 6 \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \\ w_3 \end{bmatrix} = 0, \tag{30}$$

From the above equations, we can define the behavior of the system (28) as all possible sets of solutions (w_1, w_2 , and w_3) that satisfy

(29) or (30). We call equation (29) or (30) a kernel representation of the dynamic system (28) [15].

We can use the behavioral system theory notation $M(\frac{d}{dt})$ to describe the relationships between the variables at a node can be represented by a kernel operator made up of differential equations. For more details, see [18]. To illustrate, given a node with the transfer function description:

$$T_i(s) = -\frac{10}{s+1} q_{i1}(s) - \frac{s+4}{s+2} q_{i2}(s) + \frac{s+5}{s+1} Q_i^{in}(s),$$

which could be rewritten as

$$T_i(s) + \frac{10}{s+1} q_{i1}(s) + \frac{s+4}{s+2} q_{i2}(s) - \frac{s+5}{s+1} Q_i^{in}(s) = 0,$$

we can describe this as $M(\frac{d}{dt})x(t) = 0$, where $x(t) = [T_i(t) \ q_{i1}(t) \ q_{i2}(t) \ Q_i^{in}(t)]^T$ and $M(\frac{d}{dt})$ is defined by (31).

With this notation, an edge can be represented as $M^{i \leftrightarrow j}(\frac{d}{dt})$, as is done in considering the two-node example shown in Fig. 9. The transfer function in the connection element is equivalently:

$$\begin{bmatrix} -(\frac{d}{dt}(\cdot) + 3) & 0 & \frac{d}{dt}(\cdot) + 1 & 10 \\ 0 & -(\frac{d}{dt}(\cdot) + 3) & 10 & \frac{d}{dt}(\cdot) + 2 \end{bmatrix} \begin{bmatrix} q_1 \\ q_2 \\ T_1 \\ T_2 \end{bmatrix} = 0. \tag{32}$$

The kernel representation for the system can then be defined by:

$$M \left(\frac{d}{dt} \right) x(t) = 0. \tag{33}$$

B. Controllability using the Behavioral Representation

Definition 4.1: [15] The system (33) is *controllable* if for any two trajectories $x_1(t)$ and $x_2(t)$ (as shown in Fig. 10) satisfying (33) there exists a time $T \geq 0$ and a third trajectory $x_3(t)$ satisfying (33) such that

$$x_3(t) = \begin{cases} x_1(t) & t \leq 0 \\ x_2(t - T) & t \geq T \end{cases}$$

In a controllable system, it is possible to utilize freely-assignable variables to switch between any legal past and future trajectories with some delay T as shown in Fig. 10. Thus, for a Controllable system, it is possible to reach any legal future trajectory regardless of its present state or to use the transition trajectory. In [19], controllability was described using the kernel representation:

Theorem 4.1:

The system (33) is controllable if and only if the rank of $M(s)$ is the same for all $s \in \mathbb{C}$.



$$M\left(\frac{d}{dt}\right) = \left[\left(\frac{d^2}{dt^2}(\cdot) + 3\frac{d}{dt}(\cdot) + 2\right) \quad 10\left(\frac{d}{dt}(\cdot) + 2\right) \quad \left(\frac{d^2}{dt^2}(\cdot) + 5\frac{d}{dt}(\cdot) + 4\right) \quad -\left(\frac{d^2}{dt^2}(\cdot) + 7\frac{d}{dt}(\cdot) + 10\right)\right] \quad (31)$$

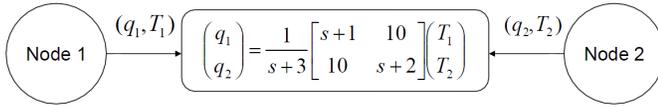


Fig. 9. Connection transfer function example.

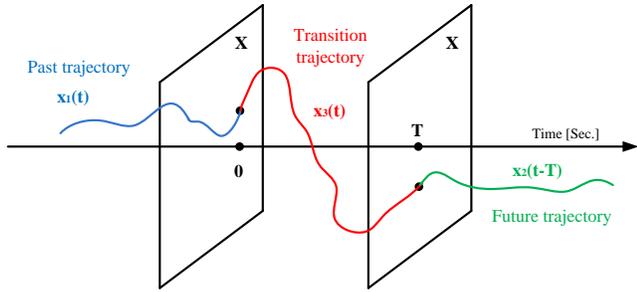


Fig. 10. Behavioral controllability.

C. Controllability of Two Rooms: An Example

As an example, we investigate the controllability of the two-room example. Our problem is first formulated by defining the behavioral representation of the model containing two rooms (For more details in the modeling of two rooms example, see Section II-A. From Fig. 3, the nodal equation can be written as:

$$\begin{bmatrix} C_i^r s & 0 \\ 0 & C_j^r s \end{bmatrix} \begin{bmatrix} T_i \\ T_j \end{bmatrix} = \begin{bmatrix} Q_1^{in} \\ Q_2^{in} \end{bmatrix} - \frac{1}{B_{ij}(s)} \begin{bmatrix} A_{ij}(s) & -D_{ij}(s) \\ -D_{ij}(s) & A_{ji}(s) \end{bmatrix} \begin{bmatrix} T_i \\ T_j \end{bmatrix} \quad (34)$$

From (34), the kernel representation of the two rooms example can be expressed by (35).

Using Theorem 4.1 and the structures of the system's matrices, the following theorem defines the controllability of the two rooms example:

Theorem 4.2: Suppose for all i, j , the roots of $B_{ij}(s)$ (in 6) are isolated. Then system (35) is uncontrollable if and only if there exists a root λ of $B_{ij}(s)$ such that

$$\begin{bmatrix} A_{ij}(\lambda) & -D_{ij}(\lambda) \\ -D_{ij}(\lambda) & A_{ji}(\lambda) \end{bmatrix} \quad (36)$$

is not full row rank.

Proof:

Based on Theorem

Case 1: λ not a root of $B_{ij}(s)$: The system is controllable because $B_{ij}(\lambda) \neq 0$, thus the rank of $M(s)$ is two.

Case 2: λ a root of $B_{ij}(s)$: In this case $B_{ij}(\lambda) = 0$, so $M(s)$ becomes

$$M(\lambda) = \begin{bmatrix} A_{ij}(\lambda) & -D_{ij}(\lambda) & 0 & 0 \\ -D_{ij}(\lambda) & A_{ji}(\lambda) & 0 & 0 \end{bmatrix}. \quad (37)$$

The last equation shows that the system is uncontrollable if and only if the non-zeros block matrix.

$$\begin{bmatrix} A_{ij}(\lambda) & -D_{ij}(\lambda) \\ -D_{ij}(\lambda) & A_{ji}(\lambda) \end{bmatrix} \quad (38)$$

is not full row rank. ■

A similar result is also possible with repeated roots where the rank condition would have to be verified for terms that share roots of the same or higher multiplicity.

V. GENERAL CONTROLLABILITY OF DYNAMIC CONSENSUS NETWORKS

The controllability conditions obtained in the previous section cannot be applied for the general dynamic networks, such as identical LTI nodes with dynamic edges or even in the more general case with heterogeneous nodes. This is because of scalability in such dynamic networks. For example, we have a similar result for the four rooms example, but this result will be challenging to apply when considering more rooms or nodes in a dynamic network. Thus, we seek to develop controllability conditions based on node and interconnection (edge) parameters that guarantee the overall dynamic network's controllability.

The block diagram for the general dynamic networks, heterogeneous nodes with dynamic edges proposed in (21) can be depicted in Fig.reffig:DynamicHeterogeneousNetwork. To propose our task, our problem is first formulated by considering a dynamic consensus network consisting of N heterogeneous LTI nodes $P_i(s)$ for all $i = 1, \dots, N$ modeled by the following state-space equations:

$$\begin{aligned} \dot{x}_{p,i}(t) &= A_{p,i}x_{p,i}(t) + B_{p,i}u_{p,i}(t) \\ y_{p,i}(t) &= C_{p,i}x_{p,i}(t), \end{aligned} \quad (39)$$

where $x_{p,i}(t) \in \mathbb{R}^{n_i}$, $u_{p,i}(t) \in \mathbb{R}^{m_i}$, $y_{p,i}(t) \in \mathbb{R}^{p_i}$ denote the state, input, and output, respectively, of node i for $i = 1, \dots, N$ and $A_{p,i}, B_{p,i}, C_{p,i}$ are constant matrices with appropriate dimensions. If we define the vectors $x_p(t) = [x_{p,1}(t)^T, \dots, x_{p,N}(t)^T]^T$, $u_p(t) = [u_{p,1}(t)^T, \dots, u_{p,N}(t)^T]^T$, and $y_p(t) = [y_{p,1}(t)^T, \dots, y_{p,N}(t)^T]^T$ we can write (39) in a matrix form as:

$$\begin{aligned} \dot{x}_p(t) &= A_p x_p(t) + B_p u_p(t) \\ y_p(t) &= C_p x_p(t), \end{aligned} \quad (40)$$

where, $x_p(t) \in \mathbb{R}^n$, $u_p(t) \in \mathbb{R}^m$, $y_p(t) \in \mathbb{R}^p$, $n = \sum_{i=1}^N n_i$, $m = \sum_{i=1}^N m_i$, $p = \sum_{i=1}^N p_i$, $A_p = \text{diag}(A_{p,i})$, $B_p = \text{diag}(B_{p,i})$, and $C_p = \text{diag}(C_{p,i})$.

Note that the input vector into the nodes $u_p(t)$ (see Fig. 11) is given by:

$$u_p(t) = u(t)^{in} + u^e(t), \quad (41)$$

where $u^{in}(t) = [u_1^{in}(t)^T, \dots, u_N^{in}(t)^T]^T$ is the input vector from the environment to the nodes in the dynamic network and $u^e(t) = [u_1^e(t)^T, \dots, u_N^e(t)^T]^T$ is the input vector from the dynamic topology (edges) to the nodes. Assume that the interconnections (edges) of the dynamic topology are modeled by transfer functions $e_{ij}(s) = \lambda_{ij}(s)$. Thus, the dynamic consensus protocol that describes the dynamic topology is given in the frequency domain by the following equations:

$$u_i^e(s) = \sum_{j \in \mathcal{N}_{p,i}} [\lambda_{ij}(s)(y_{p,j}(s) - y_{p,i}(s))], \forall i = 1 \dots N. \quad (42)$$

Note that the outgoing convention has been used for describing the dynamic topology. The dynamic consensus protocol (42) can be expressed in a matrix form as:

$$u^e(s) = -L^{out}(s)y_p(s), \quad (43)$$

where $L^{out}(s)$ is the outgoing, dynamic Laplacian matrix defined in (26) as

$$L^{out}(s) = B_D^{out}(s)B_S^{outT} + B_D^{out}(s)B_S^{inT} = B_D^{out}(s)B_S^T, \quad (44)$$



$$\underbrace{\begin{bmatrix} sC_i^r B_{ij}(s) + A_{ij}(s) & -D_{ij}(s) & -B_{ij}(s) & 0 \\ -D_{ij}(s) & sC_j^r B_{ij}(s) + A_{ji}(s) & 0 & -B_{ij}(s) \end{bmatrix}}_{M(s)|_{s=\frac{d}{dt}}} \begin{bmatrix} T_i \\ T_j \\ Q_i^{in} \\ Q_j^{in} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix},$$

$$M\left(\frac{d}{dt}\right)x(t) = 0. \tag{35}$$

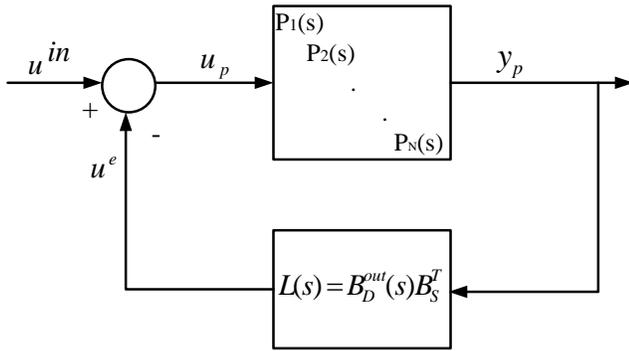


Fig. 11. Block diagram of the dynamic network with heterogeneous nodes and dynamic edges.

where $B_D^{out}(s)$ and B_S are the dynamic outgoing incident matrix and dynamic incident matrix, respectively, and these matrices are defined in Section III for the example shown in Fig. 8.

Combining (43) and (44), we get

$$u^e(s) = -B_D^{out}(s)B_S^T y_p(s), \tag{45}$$

where $B_D^{out}(s)$ matrix captures the dynamics of the edges in the dynamic topology (each column in $B_D^{out}(s)$ describes the dynamic of an edge $e_{ij}(s) = \lambda_{ij}(s)$) and B_S matrix captures the interconnections between the nodes and the edges in the dynamic network.

If the $B_D^{out}(s)$ matrix has a state space realization (A_e, B_e, C_e, D_e) , the dynamic topology model (45) can be written in the time domain as:

$$\begin{aligned} \dot{x}_e(t) &= A_e x_e(t) - B_e B_S^T y_p(t) \\ u^e(t) &= C_e x_e(t) - D_e B_S^T y_p(t), \end{aligned} \tag{46}$$

where $x_e(t) \in \mathbb{R}^{\bar{n}}$ is the state vector of the matrix B_S and A_e, B_e, C_e, D_e are constant matrices with appropriate dimensions.

Combining (40), (41) and (46), the overall representation of the dynamic consensus network with N (LTI) heterogeneous nodes and dynamic edges can be written as:

$$\begin{aligned} \begin{bmatrix} \dot{x}_p(t) \\ \dot{x}_e(t) \end{bmatrix} &= \begin{bmatrix} A_p - B_p D_e B_S^T C_p & B_p C_e \\ -B_e B_S^T C_p & A_e \end{bmatrix} \begin{bmatrix} x_p(t) \\ x_e(t) \end{bmatrix} + \begin{bmatrix} B_p \\ 0 \end{bmatrix} u^{in}(t), \\ y(t) &= [C_p \quad 0] \begin{bmatrix} x_p(t) \\ x_e(t) \end{bmatrix}. \end{aligned} \tag{47}$$

Define $\widetilde{D}_e = D_e B_S^T$ and $\widetilde{B}_e = B_e B_S^T$, the system (47) can be rewritten as

$$\begin{aligned} \begin{bmatrix} \dot{x}_p(t) \\ \dot{x}_e(t) \end{bmatrix} &= \begin{bmatrix} A_p - B_p \widetilde{D}_e C_p & B_p C_e \\ -\widetilde{B}_e C_p & A_e \end{bmatrix} \begin{bmatrix} x_p(t) \\ x_e(t) \end{bmatrix} + \begin{bmatrix} B_p \\ 0 \end{bmatrix} u^{in}(t), \\ y(t) &= [C_p \quad 0] \begin{bmatrix} x_p(t) \\ x_e(t) \end{bmatrix}. \end{aligned} \tag{48}$$

From the overall representation of the dynamic network (48), the kernel representation of this system can be expressed by:

$$\begin{bmatrix} sI_n - A_p + B_p \widetilde{D}_e C_p & -B_p C_e & -B_p \\ \widetilde{B}_e C_p & sI_{\bar{n}} - A_e & 0 \end{bmatrix} \begin{bmatrix} x_p(t) \\ x_e(t) \\ u^{in}(t) \end{bmatrix} = 0, \tag{49}$$

Using the above kernel representation (49), the following theorem is a direct result of applying the Theorem 4.1 above:

Theorem 5.1: The dynamic consensus network (47) is controllable if and only if the rank of $\begin{bmatrix} sI_n - A_p + B_p \widetilde{D}_e C_p & -B_p C_e & -B_p \\ \widetilde{B}_e C_p & sI_{\bar{n}} - A_e & 0 \end{bmatrix}$ is the same for all $s \in \mathbb{C}$.

A proof of this theorem can be done directly. Alternately, we can use Theorem 9.4 in [20], which states that for two systems S_1 and S_2 connected in feedback configuration shown in Fig. 11, the closed-loop system is controllable if S_1, S_2 , and the series connection $S_1 S_2$ are each controllable. The latter condition requires that no pole of S_2 is a zero of S_1 . Using this result, we can state the following:

Theorem 5.2: The dynamic network described by (47) is controllable if each of the following is true:

- 1) The system of nodes defined by (40) is controllable.
- 2) The system of edges defined by (46) is controllable.
- 3) No pole of the edge system (46) is a transmission zero¹ of the node system (40).

Proof:

From Theorem

For $s \in \lambda(A_e)$: We can rewrite $\bar{Q}(s)$ in (??) as

$$\bar{Q}(s) = \bar{Q}_1(s)\bar{Q}_2(s), \tag{50}$$

where

$$\begin{aligned} \bar{Q}_1(s) &= \begin{bmatrix} I_n & 0 & 0 \\ 0 & \widetilde{B}_e & sI_{\bar{n}} - A_e \end{bmatrix}, \\ \bar{Q}_2(s) &= \begin{bmatrix} sI_n - A_p & -B_p & 0 \\ C_p & 0 & 0 \\ 0 & 0 & I_{\bar{n}} \end{bmatrix}. \end{aligned} \tag{51}$$

Form (51), condition 2. in this theorem implies that the $rank(sI_{\bar{n}} - A_e, \widetilde{B}_e) = \bar{n}$, for all s and then $rank(\bar{Q}_1(s)) = n + \bar{n}$, for all s . For $s \in \lambda(A_e)$, the third condition implies that the $rank(\bar{Q}_2(s)) = n + p + \bar{n}$, for all $s \in \lambda(A_e)$, where $p = \sum_i p_i$ is the dimension of the output vector $y_p(t)$ of the system.

Applying Sylvester's rank inequality (see [21] and [22]) in the equation (50), we get

$$n + \bar{n} \geq rank(\bar{Q}(s)) \geq rank(\bar{Q}_1(s)) + rank(\bar{Q}_2(s)) - d, \tag{52}$$

where $d = n + p + \bar{n}$ is the number of columns in the matrix $\bar{Q}_1(s)$.

From (52) and (51), we can rewrite the last inequality as

$$n + \bar{n} \geq rank(\bar{Q}(s)) \geq (n + \bar{n}) + (n + p + \bar{n}) - (n + p + \bar{n}). \tag{53}$$

¹A transmission zero for $\dot{x} = Ax + Bu, y = Cx$ is a value $\lambda \in \mathbb{C}$ such that $\begin{bmatrix} \lambda I_n - A & B \\ C & 0 \end{bmatrix}$ loses rank. Transmission zeros correspond to terms $e^{\lambda t}$ that will not appear in the output when they are in the input.



From the above equation we can conclude that the $\text{rank}(\tilde{Q}(s)) = n + \bar{n}$, for all $s \in \lambda(A_e)$ and then the rank of the matrix $Q(s)$ is the same for all s and hence the system is controllable. ■

Note that Theorem 5.1 and Theorem 5.2 are equivalent. The controllability conditions introduced in Theorem 5.2 are attractive because it concludes that in addition to the controllability of the nodes and edges, controllability requires the poles of the edges not to match the transmission zeros of the nodes. Furthermore, the third condition can be investigated using only the region of the eigenvalues of the system matrix A_e . This makes controllability analysis more straightforward because it can be analyzed using only the bounds of $\lambda_i(A_e)$ or by simply checking the poles and zeros of the nodes and edges dynamics.

VI. CONCLUSION

This paper studied a generalization of consensus network problems whereby the network edges' weights are no longer modeled as static gains. Instead, they are represented as dynamic systems coupling the nodes. We call such networks dynamic consensus networks because, under some conditions, all node variables converge to a common value called a consensus. We presented examples of how dynamic graphs can arise in applications. Detailed studies of modeling thermal processes in buildings as directed dynamic graphs were presented. Motivated by these examples, a framework was proposed for dynamic graphs and dynamic consensus networks. This framework introduced the idea of dynamic degree, adjacency, incident, and Laplacian matrices in a way that naturally extends these concepts from the static case. The dynamic consensus networks addressed herein considered various dynamics of nodes and interconnection topology, including (1) directed dynamic networks with integrator nodes and real-rational transfer function edges; (2) undirected dynamic networks with integrator nodes and strictly-positive-real transfer function edges; and (3) undirected dynamic networks with identical linear time-invariant nodes and dynamic edges. We used the established aspects and properties of the defined dynamic graph theory in conjunction with the behavioral approach to developing a controllability-analysis methodology for dynamic networks. Additionally, the dynamic interconnection topology application in analyzing the interconnections of nodes to establish controllability analysis was demonstrated. The behavioral approach was described, followed by a discussion of its use in the two-rooms example's controllability. This led to presenting the resulting general usage of dynamic graph theory in conjunction with the behavioral approach for controllability-analysis of dynamic consensus networks.

ACKNOWLEDGMENTS

The authors would like to thank Professor Kevin L. Moore and Professor Tyrone L. Vincent at the Colorado School of Mines for their help and advice.

REFERENCES

- [1] R. Olfati-Saber and R. M. Murray, "Consensus problems in networks of agents with switching topology and time-delays," *IEEE Transactions on automatic control*, vol. 49, no. 9, pp. 1520–1533, 2004.
- [2] K. L. Moore, T. L. Vincent, F. Lashhab, and C. Liu, "Dynamic consensus networks with application to the analysis of building thermal processes," *Proceedings of the 18th IFAC World Congress*, vol. 18, no. 1, pp. 3078–3083, 2011.
- [3] Fadel Lashhab, "Estimation of Dynamic Laplacian Eigenvalues in Dynamic Consensus Networks," *IEEE International IOT, Electronics and Mechatronics Conference (IEMTRONICS)*, pp. 1–10, 2020.
- [4] Fadel Lashhab, "Dynamic Consensus Networks: Dynamic Graph Definitions and Controllability Analysis Using the Behavioral Approach," *IEEE International IOT, Electronics and Mechatronics Conference (IEMTRONICS)*, pp. 1–9, 2020.
- [5] Fadel Lashhab, Kevin Moore, Tyrone Vincent, Deyuan Meng, and Khalid Kuwairi, "Robust H_∞ controller design for dynamic consensus networks", *International Journal of Control*, vol. 91, no. 8, pp. 1906–1919, 2018.

- [6] D.B. Crawley, L.K. Lawrie, F.C. Winkelmann, WF Buhl, Y.J. Huang, C.O. Pedersen, R.K. Strand, R.J. Liesen, D.E. Fisher, M.J. Witte, et al. EnergyPlus: creating a new-generation building energy simulation program. *Energy & Buildings*, 33(4):319–331, 2001.
- [7] J.E. Braun and N. Chaturvedi. An inverse gray-box model for transient building load prediction. *HVAR&R Research*, 8(1):73–100, 2002.
- [8] K. Lee and J.E. Braun. Model-based demand-limiting control of building thermal mass. *Building and Environment*, 43:1633–1646, 2008.
- [9] X. Xu and S. Wang. Optimal simplified thermal models of building envelope based on frequency domain regression using genetic algorithm. *Energy and Buildings*, 39:525–536, 2007.
- [10] K.-K. Oh, F. Lashhab, K. L. Moore, T. L. Vincent, and H.-S. Ahn, "Consensus of positive real systems cascaded with a single integrator," *International Journal of Robust and Nonlinear Control*, vol. 25, no. 3, pp. 418–429, 2015.
- [11] X. Liu, H. Lin, and B. Chen, "A graph-theoretic characterization of structural controllability for the multi-agent system with switching topology," in *Proceedings of the 48th IEEE Conference on Decision and Control, held jointly with the 28th Chinese Control Conference*, pp. 7012–7017, IEEE, 2009.
- [12] H. Tanner, "On the controllability of nearest neighbor interconnections," in *Proceedings of the 43rd IEEE Conference on Decision and Control*, vol. 3, pp. 2467–2472, IEEE, 2004.
- [13] M. Ji, A. Muhammad, and M. Egerstedt, "Leader-based multi-agent coordination: Controllability and optimal control," in *Proceedings of 2006 American Control Conference*, pp. 1358–1363, IEEE, 2006.
- [14] A. Rahmani, M. Ji, M. Mesbahi, and M. Egerstedt, "Controllability of multi-agent systems from a graph-theoretic perspective," *SIAM Journal on Control and Optimization*, vol. 48, no. 1, pp. 162–186, 2009.
- [15] J. Willems, "The behavioral approach to open and interconnected systems," *IEEE Control Systems Magazine*, vol. 27, no. 6, pp. 46–99, 2007.
- [16] J. Willems, "On interconnections, control, and feedback," *IEEE Transactions on Automatic Control*, vol. 42, no. 3, pp. 326–339, 1997.
- [17] J. Willems and Y. Yamamoto, "Linear differential behaviors described by rational symbols," in *Proceedings of the 17th IFAC World Congress, Seoul, Korea*, pp. 12266–12272, 2008.
- [18] J. Polderman and J. Willems, *Introduction to Mathematical Systems Theory: a Behavioral Approach*. Springer Verlag, 1998.
- [19] J. Willems, "The Behavioral Approach to Open and Interconnected Systems Modeling by tearing, zooming, and linking," *IEEE Control Systems Magazine*, vol. 27, no. 6, pp. 46–99, 2007.
- [20] C. Chen, *Linear System Theory and Design*. Oxford University Press, Inc., 1998.
- [21] G. Matsaglia and G. Styan, "Equalities and inequalities for ranks of matrices†," *Linear and Multilinear Algebra*, vol. 2, no. 3, pp. 269–292, 1974.
- [22] A. Ramakrishna and N. Viswanadham, "Decentralized control of interconnected dynamical systems," *IEEE Transactions on Automatic Control*, vol. 27, no. 1, pp. 159–164, 1982.
- [23] Vikas Kumar Mishra, Ivan Markovskiy, and Ben Grossmann, "Data-Driven Tests for Controllability," *IEEE Control Systems Letters*, vol. 5, no. 2, pp. 517–522, 2020.
- [24] M. Ji and M. Egerstedt, "A graph-theoretic characterization of controllability for multi-agent systems," *In IEE American Control Conference, ACC'07*, pp. 4588–4593., 2007.
- [25] M. Ji, A. Muhammad, and M. Egerstedt, "Leader-based multi-agent coordination: Controllability and optimal control," *In IEEE American Control Conference*, pp. 6, 2006.
- [26] A. Rahmani and M. Mesbahi, "On the controlled agreement problem," *In IEEE American Control Conference*, pp. 6, 2006.
- [27] Amirreza Rahmani, Meng Ji, Mehran Mesbahi, and Magnus Egerstedt, "Controllability of multi-agent systems from a graph-theoretic perspective," *SIAM Journal on Control and Optimization*, vol. 48, no. 1, pp. 162–186, 2009.
- [28] M. Zamani and H. Lin, "Structural controllability of multi-agent systems," *In IEEE American Control Conference, ACC'09*, pp. 5743–5748, 2009.
- [29] Tadeusz Kaczorek "Minimum energy control of positive continuous-time linear systems with bounded inputs," *International Journal of Applied Mathematics and Computer Science*, vol. 23, no. 4, pp. 725–730, 2013.
- [30] Jerzy Klamka "Controllability and minimum energy control problem of fractional discrete-time systems," *In New Trends in Nanotechnology and Fractional Calculus Applications*, pp. 503–509, Springer, 2010.
- [31] He, Wangli and Xu, Bin and Han, Qing-Long and Qian, Feng," *Adaptive consensus control of linear multiagent systems with dynamic event-*



- triggered strategies*, IEEE transactions on cybernetics, vol. 50, no. 7, pp. 2996–3008, 2019.
- [32] Ren, Wei, "Multi-vehicle consensus with a time-varying reference state, Systems & Control Letters, vol. 56, no. 7-8, pp. 474–483, 2007.



Distributed Constraint Optimization Problem Solving in Unstable Environments

Saeid SamadiDana

Department of Computer Science and IT

Austin Peay State University

Clarksville, TN, USA

samadidan@apsu.edu

Abstract—Distributed systems are widely used to share tasks in various systems which communicate to each other. one main reason to use these systems is their ability to keep each system's privacy and share only the required information. The success of these systems relies on robust communications among their nodes. Advances in network and communication technologies have led to a more robust quality solution for distributed problems as these systems heavily rely on network robustness and stability. Despite this progress, the communication problems such as delay, loss, and noise still exist in many environments that have dramatically affected the quality of distributed problem solutions. In some recent studies, these issues have been explored partially; however, the need to investigate these issues' impact specifically when combined seems necessary. This article studies the effect of message loss while there is a chance of distortion in the receiving messages. To have a better view of communication issues, both static and dynamic problems are tested. Three distributed algorithms, Distributed Stochastic Algorithm (DSA), Distributed Breakout Algorithm (DBA), and Max-Gain Message algorithm (MGM), are chosen to be tested in these environments, and their performance has been compared to each other. Test results show that all three algorithms are highly impacted by network instability, while DSA provides better results in general.

Keywords—Multi-Agent System, Distributed, Problem, Dynamic, Noise, Optimization, Communication, Privacy

I. INTRODUCTION

Recent advances in network technology have provided a relatively robust environment for many distributed systems. Especially the 5-G, despite its challenges, seems will provide a more promising network reliability (Andrews et al., 2014). This progress can connect many heterogeneous devices and systems possible through the Internet of Things (Kiran, 2019; Mira, 2019). Nonetheless, in many cases having a robust and reliable network is difficult. The existence of distributed systems relies on communication among various parts (nodes) of them. Without a trustworthy network, distributed systems cannot exchange information promptly. For various reasons, a problem might be distributed over a network of solvers that work together to solve a problem. The need to maintain the privacy of individual systems or limit the amount of information shared or the computing power is why one may use distributed systems rather than centralized ones. Some of the main problems that distributed systems have solved are constraint satisfaction problems (CSP) and constrain optimization problems (COP). Distributed supply change managements(Aleman, Estes, Ángel Ortiz, & del

Pino, 2021) and various versions of path-finding(Yao, Qi, Wan, & Liu, 2019; Samadidana, Paydar, & Jouzdani, 2017) are just two examples of thousands of such problems. Generally, distributed algorithms share the tasks among various individual systems known as nodes (or agents). Each node is responsible for solving the local problem while communicating with other system nodes to solve the whole problem. Hence, the key to having a robust solution is exchanging the most up-to-date information in a timely manner. However, even a reliable network might face noise that can adversely impact the distributed algorithms' performance. Delay in communication and loss of messages are very common issues. Several studies such as (Samadidana, 2019; Wahbi & Brown, 2014) have investigated issues such as communication delay and loss and explored the effect of various sources of information stagnancy.

In this study, the performance of three distributed algorithms used to solve distributed constraint optimization problems (DCOP) has been analyzed. It should be noted that this work extends the findings presented at IEEE CCWC 2021 conference (Samadidana, 2021). To have a better understanding of the communication issues, both static and dynamic DCOP problems are tested. Moreover, this paper investigates the effect of communication delay and loss on these problems while there are some chances that the delivered messages are noisy and contain invalid information. To the author's best knowledge, this work is the first study that considers the combination of various sources of information stagnancy in DCOP problems. This study aims to advance the research community's knowledge of information stagnancy in distributed problem-solving. The algorithms that are used in this paper are DSA (Zhang, Wang, & L.Wittenburg, 2002), DBA (Yokoo & K.Hirayama, 1996), and MGM (Maheswaran, Pearce, & D, 2004). These algorithms are chosen as each represents a different approach to solve the DCOP problems. However, in all of them, the nodes communicate the latest information with each other. The structure of this paper is as follows: In Section II, Distributed Constraint Optimization Problems (DCOP) and Dynamic DCOP (DynDCOP) problems are introduced. Section III describes the three distributed algorithms. Test results are explained in Section IV. This section is divided into two subsections to investigate both static(Subsection IV-A) and dynamic problems (Subsection IV-B). Finally, in Section V the study is concluded, and the results are summarized.



II. DEFINITIONS AND BACKGROUND

A. Constraint Satisfaction Problems (CSP)

A Constraint Satisfaction Problem(CSP) is described as following (Russell & Norvig, 2020):

- A set of n variables: $\mathbf{V} = \{v_1, \dots, v_n\}$.
- Discrete, finite domains for each variable: $\mathbf{D} = \{\mathbf{D}_1, \dots, \mathbf{D}_n\}$ where $\mathbf{D}_i = \{d_{i,1}, \dots, d_{i,j}\}$ and each $d_{i,j}$ is an element of domain \mathbf{D}_i
- a set of constraints $\mathbf{C} = \{c_1, \dots, c_m\}$ where each $c_h(d_{i,1}, \dots, d_{i,j} | 1 \leq i, j \leq n)$ is a function $c_h : D_{i,1} \times \dots \times D_{i,j} \rightarrow \{true, false\}$.

Based on the definition, $D_{i,j}$ is a pair of domain values that have been assigned to variables i and j . In addition, a constraint can be set either to true or false. The goal of the problem is to satisfy all the constraints or to conclude that no solution exists. CSP problems are known by some parameters such as density and tightness. Density of a CSP problem is defined as $p_1 = \frac{2m}{n*(n-1)}$ and its tightness is $p_2 = \frac{\#false\ assignments}{\#total\ assignments}$ in which n is the number of variables and m is the number of constraints.

B. Distributed Constraint Optimization Problems (DCOP)

As mentioned, the goal of solving a CSP problem is to find out if the constraints can be satisfied. Similarly, a Constraint Optimization Problem (COP) problem can be defined based on the cost or gain function among variables. Following definition describes a distributed COP(DCOP) problems $\mathbf{P} = \langle \mathbf{V}, \mathbf{A}, \mathbf{D}, \mathbf{C} \rangle$ as (Yokoo & Durfee, 1991):

- A set of n variables: $\mathbf{V} = \{v_1, \dots, v_n\}$.
- A set of g agents: $\mathbf{A} = \{a_1, \dots, a_g\}$.
- Discrete, finite domains for each variable: $\mathbf{D} = \{\mathbf{D}_1, \dots, \mathbf{D}_n\}$.
- A set of m cost functions $\mathbf{C} = \{c_1, \dots, c_m\}$, where each $c_h(d_{i,1}, \dots, d_{i,j} | 1 \leq i, j \leq n)$ is a function where $c_h : D_{i,1} \times \dots \times D_{i,j} \rightarrow \{q | q \in \mathbb{N} \wedge 0 \leq q \leq c_{max}\}$.

The set of g agents are responsible for managing the variables of the problem. Each agent can be assigned one or more variables. That means an agent will manage the communication among its assigned variables with other variables and try to solve the whole problem while communicating with other agents. Moreover, $c_{max} \in C$ is the maximum possible cost (or gain) functions value. In this definition, it is assumed that the problem does not change over time, which means the constraints table is always the same. In other words, the graph structure of the problem never changes.

C. Dynamic Distributed Constraint Optimization Problems (DynDCOP)

In the previous section static DCOP problems are introduced. In this section dynamic DCOP problems (DynDCOP) are explained. A DynDCOP problem can be defined as a sequence of static problems (Dechter & Dechter, 1988). Assume a DCOP problem p_t at time t that changes to the problem p_{t+1} at time $t + 1$. The sequence of these static

problems $\{P_0, P_1, \dots, P_t\}$ over a period of time (t) creates a dynamic DCOP problem. The difference between any two consecutive static problems p_{t-1} and p_t is the added (c_t^a), removed (c_t^r) or changed constraints. Therefore the problem P_t can be written as (Verfaillie & Jussien, 2005):

$$P_t = P_{t-1} + c_t^a - c_t^r \quad (1)$$

One can see that in the Eq. 1 the changed constraints are also included. A constraint change can be seen as removing it and adding it back with a new constraint table. As DynDCOP problems change over time, it is possible to measure their change rate (Mailler, 2005).

$$rate = \frac{dP}{dt} = \lim_{\Delta t \rightarrow 0} \frac{P_{t+\Delta t} - P_t}{\Delta t} \quad (2)$$

Eq. 2 can be expanded by including the definition of the problem P_t at time t . (Mailler, 2005):

$$rate = \frac{1}{\Delta t} \sum_{i=t}^{t+\Delta t} \frac{|c_i^a| + |c_i^r|}{2} \quad (3)$$

where Δt is the time that the problem P_t changes to the problem $P_{t+\Delta t}$.

Since formulating the DynDCOP problem, several researchers have studied these problems' behaviors and have tried to model their behaviors. Some outstanding work on this area can be found in the work of (Ridgway & Mailler, 2015; Mailler & Zheng, 2014). They showed that DynDCOP problems follow the thermodynamic laws; therefore, the change of solution quality (energy) can be modeled and predicted. Some recent work in this area can be found in (Ridgway & Mailler, 2015; Samadidana & Mailler, 2019).

III. DISTRIBUTED PROTOCOLS

Various distributed algorithms have been designed to solve DCSP problems. In this study, Distributed Stochastic Algorithm (DSA) (Zhang et al., 2002), Distributed Breakout Algorithm (DBA) (Yokoo & K.Hirayama, 1996), and (Maheswaran et al., 2004) are selected to test and analyze the instability of the network. In the following subsections, each algorithm is described briefly. It must be noted that all of these algorithms have been adjusted to be able to solve DynDCOP problems.

A. Distributed Stochastic Algorithm (DSA)

The first algorithm explained is Distributed Stochastic Algorithm (DSA) (Zhang et al., 2002). DSA is a simple algorithm that applies a hill-climbing approach to solve distributed problems. In this algorithm, agents who manage variables communicate and exchange information to get the latest information. The change is done by a fixed probability value p whenever they have an improvement in their local solution. The local problem in DSA is finding the best assignments to their variables such that the constraints are satisfied (DCSP problems) or the cost(/gain) functions(DCOP problem) is minimized(/maximized). Each agent communicates directly with their direct neighbors using a message called *OK?* message.



```

procedure main
  while (not terminated) do
    update agent_view with incoming ok? ( $x_j, d_j$ )
    messages;
    new_value  $\leftarrow$  choose_value;
    if new_value  $\neq d_i$  do
       $d_i \leftarrow$  new_value;
      send (ok?, ( $x_i, d_i$ )) to all  $x_j \in$  neighbors;
    end if;
  end do;
end main;
procedure choose_value
  if  $d_i$  has no conflicts do
    return  $d_i$ ;
   $v \leftarrow$  the value with the least conflict ( $v \neq d_i$ );
  if  $v$  has the same or fewer conflicts than  $d_i$ 
    and random  $< p$  do
      return  $v$ ;
    else
      return  $d_i$ ;
    end choose_value;

```

Fig. 1. The procedures of the DSA-B algorithm

In each time cycle, agents calculate their improvement values based on their local value and the received messages from their direct neighbors. Each *OK?* message contains the current value assigned to the sender's variable. Based on the probability p , if agents find an improvement, they change their values and communicates the new value to their neighbors. This process repeats until a fixed number of cycles are reached, or an optimal solution is found. It is worth mentioning that DSA does not guarantee to find the global optimal solution (Leite, Enembreck, & Barthes, 2014). Despite this issue, DSA is very fast and in many problems yields a good solution. From five variants of DSA, DSA-B is used in this paper that allows an agent to change their values despite not having an improvement (Fig. 1).

B. Distributed Breakout Algorithm (DBA)

The Distributed Breakout Algorithm (DBA) (Yokoo & K.Hirayama, 1996) is another distributed algorithm similar to DSA works based on communication. However, DBA uses two kinds of messages called *ok?* and *improve?*. The *ok?* message contains the variable's current value, and the *improve?* message includes the amount of improvement of the variable's local problem's solution. After an agent receives the *ok?* messages from its neighbors, it calculates its improvement and communicates it with its direct neighbors through *improve?* messages. Once an agent receives all the *improve?* messages, it compares its improvement with its neighbors, and if it has a better improvement, it will change its value. DBA uses a

```

when received (ok,  $x_j, d_j$ ) do
  if mode == wait_improve
    add message to queue;
    return;
  else
    add ( $x_j, d_j$ ) to agent_view;
    when received ok? messages from all neighbors do
      send_improve;
      mode  $\leftarrow$  wait_improve;
    end do;
  end if;
end do;
procedure send_improve
  current_eval  $\leftarrow$  evaluation value of current_value;
  improve_i  $\leftarrow$  possible maximum improvement;
  new_value  $\leftarrow$  the value which yields the best
  improvement;
  send (improve,  $x_i, improve_i, current_eval$ ) to
  neighbors;
end send_improve;

```

Fig. 2. The procedures of the *wait_ok?* mode in DBA

weighting mechanism to escape from the quasi-local-minimum by changing the weight of violated constraints.

Figures 2 and 3 show the various procedures of DBA.

C. Max-Gain Message Algorithm

Max-Gain Message Algorithm (MGM) is another distributed algorithm that, similar to DBA, is a two-phase protocol (Maheswaran et al., 2004). Each agent calculates its maximum gain value and propagates it through the network. If an agent has a higher gain value, it changes its value and notifies its neighbors. Nonetheless, MGM also can suffer from the local minimum issue. Brief pseudocode of MGM is presented in Fig. 4.

IV. TEST RESULTS

As mentioned in Section II, several studies have confirmed the negative effect of information stagnancy on distributed problem-solving. Nevertheless, more work is needed to gain a comprehensive view of the effect of information stagnancy. This work provides a detailed examination of various information stagnancy sources, including message loss, delay, and distortion. In order to investigate the effect of communication issues comprehensively, both static and dynamic DCOP problems are evaluated. Moreover, various test parameters are chosen to demonstrate the effects of information stagnancy better. Each test scenario is run 50 times, and the average function is used to summarize the result. A test scenario includes a DCOP problem with a determined tightness, density, and the number of variables. Furthermore, various amounts of noise and loss probability are used to have a combination of



```

when received (improve,  $x_j, improve_j, eval$ ) do
  if mode == wait_ok
    add message to queue;
    return;
  else
    record message;
    when received improve messages from all neighbors
  do
    send_ok;
    clear agent_view;
    mode  $\leftarrow$  wait_ok;
  end do;
end if;
end do;
procedure send_ok
  if  $improve_i$  is better than all of my neighbors
    current_value  $\leftarrow$  new_value;
  end if;
  when in a quasi-local-minimum do
    increase the weights on all violated constraints;
  end do;
  send (ok?,  $x_i, current\_value$ ) to neighbors;
end send_ok;

```

Fig. 3. The procedures of the wait_improve mode in DBA

```

procedure main
  initialize;
  while (not terminated) do
    send current value to neighbors;
    get all neighbors value;
    calculate and send maximum gain mg;
    collect all neighbors maximum gain_ok;
    if  $mg > all\ neighbors\ maximum\ gain\_ok$ ;
      set new value to value that gives mg;
    end if;
    send_improve;
  end do;
end main;

```

Fig. 4. The Max-Gain Message algorithm

communication degradation. The complete list of test scenario parameters is described in the following sections. Usually, the gain/cost value is used to evaluate the performance of DCOP problems.

As in DCOP problems, agents communicate the most up-to-date information on a cycle-based time; the solution at the end of the last cycle is used. In other words, this solution is the one that the solver converges to after a certain period. As each distributed algorithm completes the task of solving a single

instance of the problem at a different time, another parameter called cycle is used. This parameter allows a distributed algorithm to run for a determined number of times it takes to receive/send the messages from/or the agents. For example, in DSA, a cycle for an agent is the amount of time required to receive the *ok?* messages and decide if they have a better value and sending the new information. In DBA, an agent either processes *ok?* or *improve?* messages in each cycle. Therefore, DBA is a two-phase algorithm that needs cycles to complete the solving process. Similarly, MGM works in two phases to complete solving the problem. It is worth mentioning that this process repeats until either a certain number of cycles are reached, or the algorithms reach an expected solution. In this study, each algorithm runs 1000 cycles.

It is worth mentioning that it is assumed that the algorithms do not use a checking system to verify the correctness of messages. This assumption is necessary because having a verification mechanism reduces the noise issue with messaging loss problem (Samadidana & Mailler, 2017) however, having a verification mechanism is not always possible due to various reasons such as cost, power, or bandwidth issues; therefore, the noise problem is not always avoidable. For example, adding a checksum to the messages might increase the processing time in addition to the size of the messages.

Two subsections continue this section. The first one provides the results and analysis for the static DCOP problems, and the second one investigates the dynamic ones.

A. Static DCOP Problems

The first part of the tests is dedicated to static problems. In the static problems, the problems do not change over time; hence the distributed algorithms will solve the problems over a fixed number of cycles. Then the final solution (convergence point), which is obtained at the end of the last cycle of solving the problem, is used to determine the algorithms' performance. The parameters used in this section are as following:

- number of variable = 100
- number of domain value = 3
- density = {0.02, 0.035, 0.055}
- tightness = 0.33
- min cost = 1
- max cost = 10
- noise probability = {0.1, 0.2, 0.3}

For each combination of parameters, 50 test runs were run, and the average function is used to analyze the result. Each problem contains 100 variables (agents), and each variable can be assigned three domain values. The minimum cost of each constraint is set to 1 and the maximum cost to 10. In addition, to evaluate the impact of the problem densities, three densities are chosen. In each test scenario, various noise probabilities are chosen to investigate the amount of impact.

1) *Noisy Environment (No Messages Loss)*: Let us begin with the scenario where all messages are delivered with a corruption chance. Tables I, II, and III present the total costs (convergence points) of the solutions for DSA, DBA, and MGM respectively. It is clear that the high chance of noise



has decreased the solutions' quality for all the algorithms. Especially, the noise chance of 0.3 shows a more negative impact on the result. The outputs suggest that DSA performs worse than DBA and MGM. This completely makes sense as DBA and MGM send the latest information constantly. Therefore, even if the noise issue corrupts some messages, in the next cycles, the agents have the chance to send the correct information. In DSA, not all the agents send the latest information in each cycle; therefore, message corruption impacts the solution more dramatically. Even if an agent gets a chance to communicate its value after some cycles, the noise chance might impact it. Besides, the results suggest that MGM works slightly better than DBA, although the difference becomes less apparent as the noise chance increases. Fig. 5 provides a summary of these three algorithms' results for various noise chances and the density $p = 0.035$.

The result also shows an interesting behavior of MGM and DBA. When both algorithms are tested in an environment with a chance of noise 0.1, they yielded better results than the problems where the communication is perfect. The reason for this peculiar phenomenon is that both DBA and MGM may get stuck in local minima; however, the small amount of noise helped them to be able to improve the solution.

It can be concluded that algorithms that communicate the information more often have a clear advantage in static problems when the noise exists. It must be noted that this advantage comes at the cost of more communication. The number of messages exchanged in DBA and MGM is much higher than the DSA's. In networks with limited bandwidth, this massive number of communication can be problematic. Therefore, a justification needs to be made to choose the proper algorithm.

TABLE I
CONVERGENCE POINTS OF DSA FOR DENSITIES $p = \{0.02, 0.035, 0.055\}$ (SAMADIDANA, 2021)

Noise	Density		
	0.02	0.035	0.055
No Noise (Samadidana, 2019)	229.91	474.63	848.61
0.1	236.23	489.31	864.43
0.2	249.10	504.32	889.27
0.3	255.97	513.19	910.51

TABLE II
CONVERGENCE POINTS OF DBA FOR DENSITIES $p = \{0.02, 0.035, 0.055\}$ (SAMADIDANA, 2021)

Noise	Density		
	0.02	0.035	0.055
No Noise (Samadidana, 2019)	223.59	472.71	846.16
0.1	215.06	455.21	819.17
0.2	235.42	485.11	845.40
0.3	257.57	507.17	872.94

2) *Noisy Environment (With Message Loss)*: In this part, message loss is included in the parameters. Loss probabilities that are tested in this study are 0.1, 0.2, and 0.3. Tables 6, 7, and 8 show the results of solving static problems which are

TABLE III
CONVERGENCE POINTS OF MGM FOR DENSITIES $p = \{0.02, 0.035, 0.055\}$ (SAMADIDANA, 2021)

Noise	Density		
	0.02	0.035	0.055
No Noise	223.47	474.47	850.28
0.1	217.18	449.6	806.49
0.2	236.28	482.57	843.49
0.3	253.78	506.64	865.9

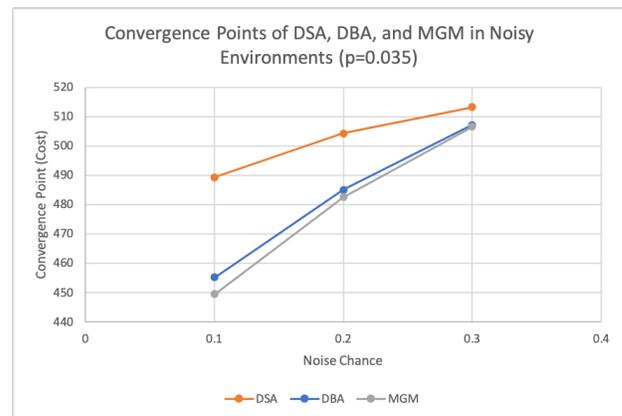


Fig. 5. Comparison of convergence points of DSA, DBA, and MGM in noisy static DCOP problems with density $p = 0.035$ (Samadidana, 2021)

impacted by both noise and message loss. When compared with the results of solving static problems in a perfect environment (no noise and loss issue), it can be observed that all of the algorithms have been impacted dramatically by the communication issues. With a lower chance of message loss and noise, the difference among DSA, MGM, and DBA results is small; however, with a higher chance of communication issues, the difference becomes more evident. For example, in Fig. 8, when the noise and message loss chances are both equal to 0.3, DBA and MGM work much worse than DSA. These results can be compared to the previous sections' results in which DBA and MGM were showing better results. To understand the reason for this difference, the structure of DSA, DBA, and MGM must be reviewed. As mentioned in DSA, agents send their information when they have an improvement based on a fixed probability. When the DCOP problem is impacted by noisy communication, the message loss can prevent agents from getting the wrong information. It is worth mentioning that in DSA-B, agents are allowed to change their value even if they do not have an improvement. However, in DBA and MGM, the issue of noise is doubled by the message loss problem. Both DBA and MGM need information from all of their neighbors in order to decide about their values. When some messages are lost, the decision-making process is delayed. This problem is worsened when the noise issue causes delivering invalid messages.

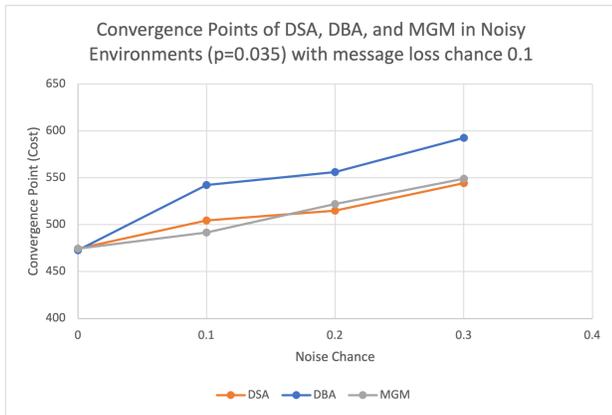


Fig. 6. Comparison of convergence points of DSA, DBA, and MGM in noisy static DCOP problems with density $p = 0.035$ and loss chance 0.1

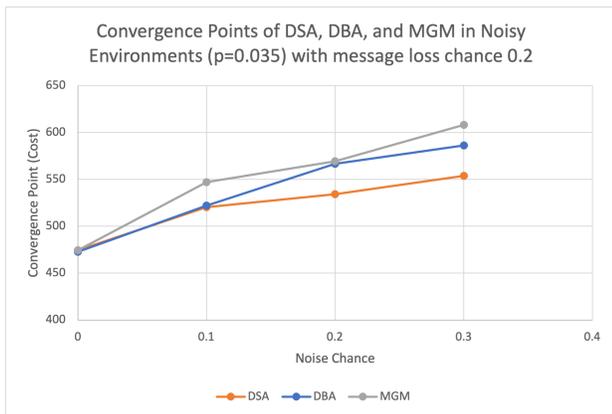


Fig. 7. Comparison of convergence points of DSA, DBA, and MGM in noisy static DCOP problems with density $p = 0.035$ and loss chance 0.2

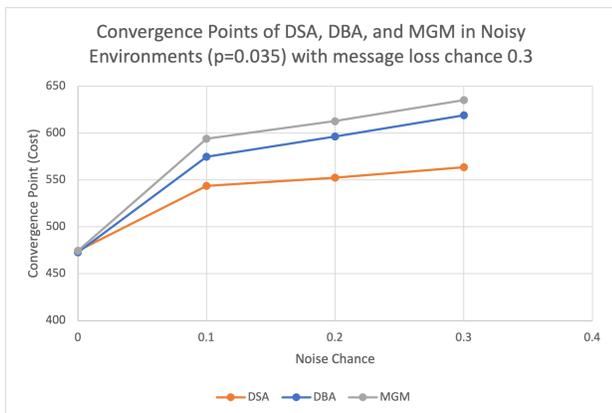


Fig. 8. Comparison of convergence points of DSA, DBA, and MGM in noisy static DCOP problems with density $p = 0.035$ and loss chance 0.3

B. Dynamic DCOP Problems

In order to have a comprehensive evaluation of message loss and noise, Dynamic DCOP problems (DynDCOP) are also evaluated. Similar to the previous sections, first, the noise issue in dynamic problems is tested. Then, the effect of having both loss and noise issues on a variety of problems has been tested. The parameters used in this section are the ones used in the static DCOP problems with the addition of the change in the problems.

- all static problems parameters
- change rate= $\{1, 5, 10, 20, 30, 40, 50, 60, 70, 80, 90\}$ of the messages

Here the change rate means the number of added or removed constraints. For example, a change rate of 1 means one constraint is removed, and a new constraint is added. It is evident that in this case, the density of the problems does not change. Each test scenario is tested 50 times, and similar to the static problems, the average function is used to explain the result. First, the DynDCOP problems are tested under noisy conditions and without the message loss issue, followed by the results of solving DynDCOP problems with both noise and message loss issues.

1) *Noisy Environment (No Messages Loss)*: As mentioned, dynamic DCOP problems change over time. In this study, it is assumed that the change rate is constant. Hence, in each cycle, only a determined number of constraints will be changed. Also, the change probability for each constraint is the same. Figures 9, 10, and 11 provide an overview of the results for density $p = 0.035$. The results confirm that by increasing the change rate of the problems, they converge to a worse result. Besides, in the absence of noise, algorithms work better (Samadidana, 2019). This is not surprising as distributed algorithms depend on communications; therefore, corruption in messages leads to lower quality solutions. Furthermore, as the noise chance increases, the results become worse; however, the difference among the results for different noise chances becomes smaller at the higher change rates. This phenomenon suggests that the high change rate overshadows the noise issue. Clearly, at the high change rates, the algorithms cannot keep up with the changes. Therefore, agents cannot make correct decisions. Even sending the correct result might not work as there is a chance that the constraints have been changed; therefore, the information will be out of date.

As shown in Fig. 9, DSA is resistant to noise in dynamic DCOP problems, especially for a lower chance of noise such as 0.1 the results are very close to the instances with perfect communication. One reason for the results can be the quickness of DSA to adapt itself to the changes in the environment. Therefore, even with some noise, it still can converge to a good solution. Nevertheless, at the very high change rates, DSA converges to the point (equilibrium point) where no progress can be made. More information can be found in the work of (Ridgway & Mailler, 2015)

Fig. 10 shows the results for DBA. Similar to DSA, with the increase in the noise probability, the results become worse.



However, in DBA, the difference among the results of solving the problems when the communication is perfect is higher than DSA's. As DBA works in a two-phase model, its decision-making process takes a longer time. Therefore, while the agents are processing the data for the current instance of the problem, the problems have changed to a new instance. The noise issue worsens the situation as the delivered messages for the unchanged constraints might be invalid.

MGM's result is very similar to the DBA's, although it shows a slightly better performance. The propagation of data in MGM by the agents can reduce the error of decision-making. However, it works in two phases that can create an unwanted delay in dynamic programs. That is why both DBA and MGM show a worse result compared to DSA.

a summary of solving DynDCOP problems with noisy communication is presented in the figures 12, 13, 14. As discussed, DSA outperforms both DBA and MGM while both DBA and MGM produce very similar results.

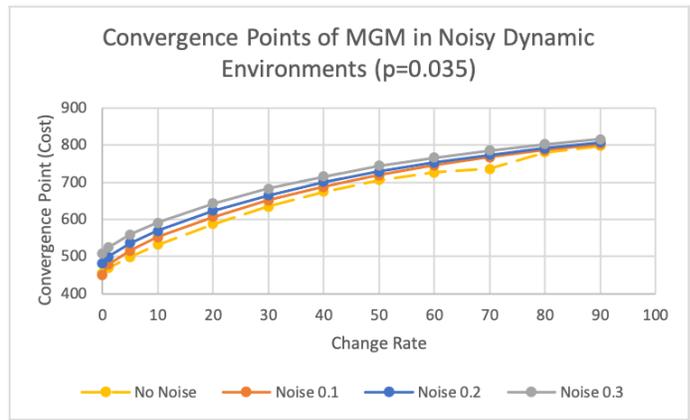


Fig. 11. Comparison of convergence points of MGM in a noisy dynamic environment with density $p = 0.035$ (Samadidana, 2021)

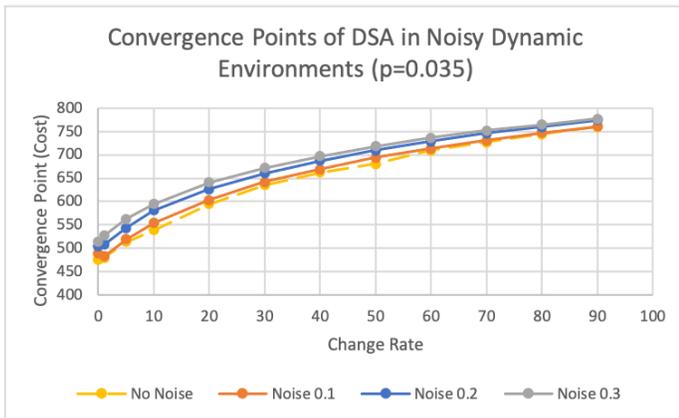


Fig. 9. Comparison of convergence points of DSA in a noisy dynamic environment with problem density $p = 0.035$ (Samadidana, 2021)

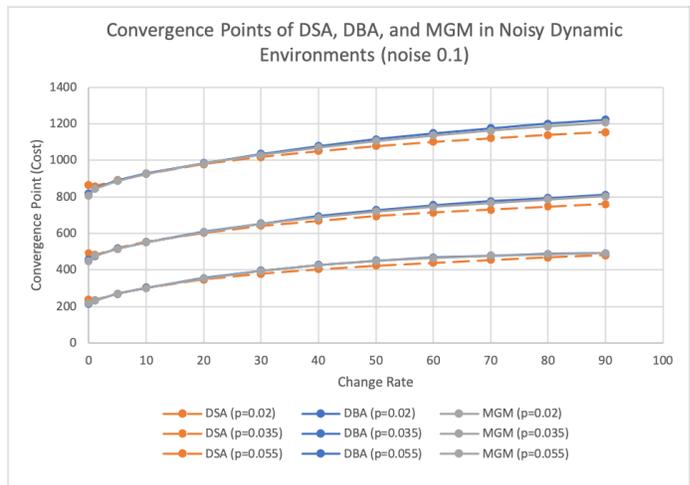


Fig. 12. Comparison of convergence points of DSA, DBA, and MGM in noisy dynamic environment with noise chance of 0.1 (Samadidana, 2021)

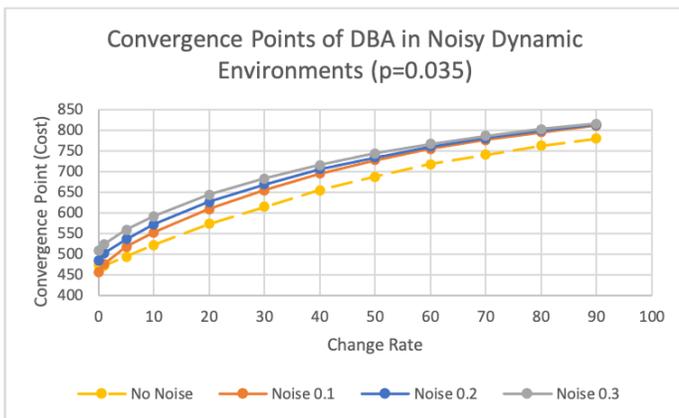


Fig. 10. Comparison of convergence points of DBA in a noisy dynamic environment with density $p = 0.035$ (Samadidana, 2021)

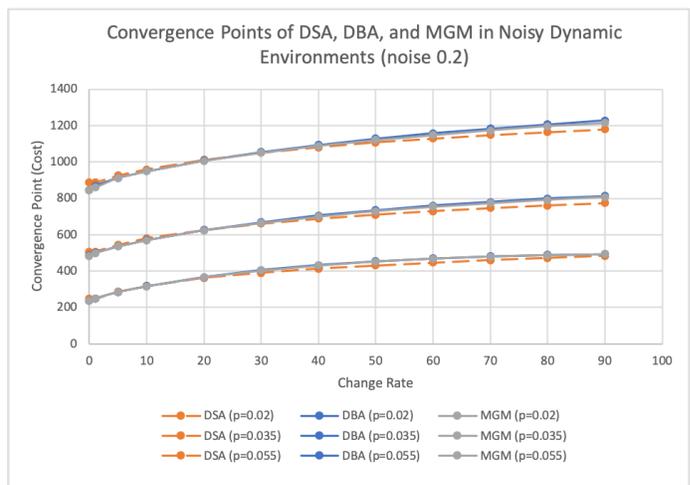


Fig. 13. Comparison of convergence points of DSA, DBA, and MGM in noisy dynamic environment with noise chance of 0.2 (Samadidana, 2021)

2) *Noisy Environment (With Message Loss)*: The last part of the tests is dedicated to solving DynDCOP problems while the problems change over time, and the communication is

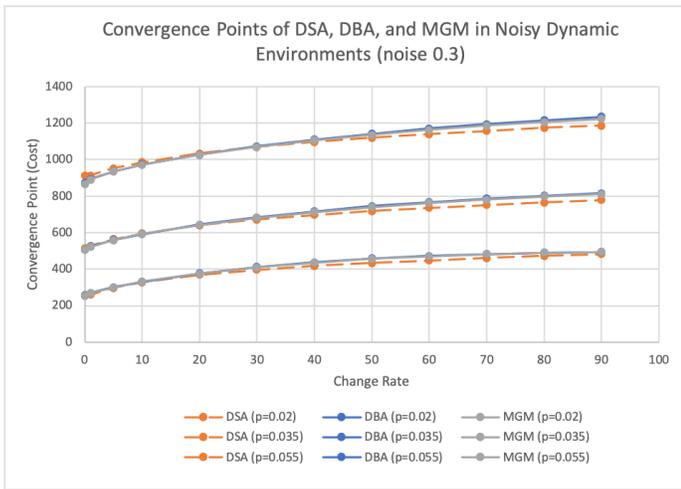


Fig. 14. Comparison of convergence points of DSA, DBA, and MGM in noisy dynamic environment with noise chance of 0.3 (Samadidana, 2021)

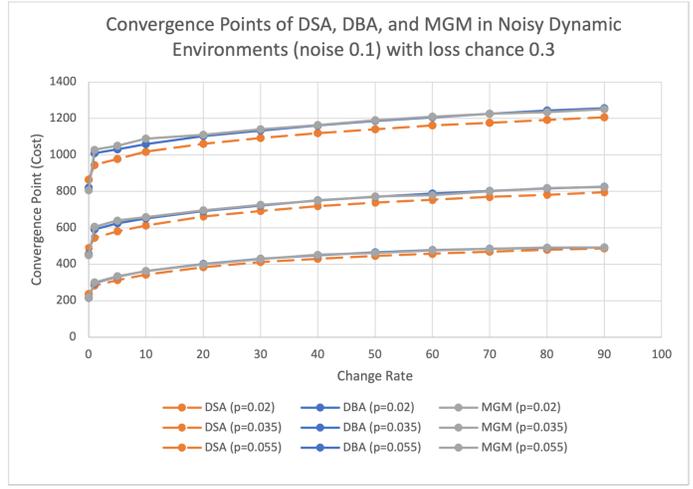


Fig. 17. Comparison of convergence points of DSA, DBA, and MGM in noisy dynamic environment with noise chance of 0.1 and loss chance of 0.3

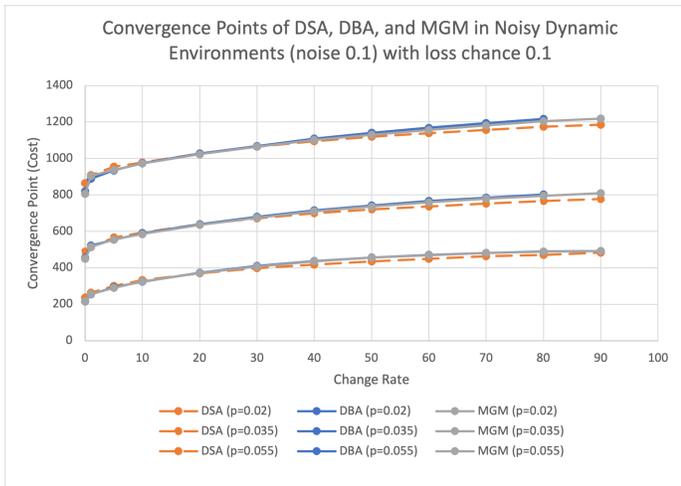


Fig. 15. Comparison of convergence points of DSA, DBA, and MGM in noisy dynamic environment with noise chance of 0.1 and loss chance of 0.1

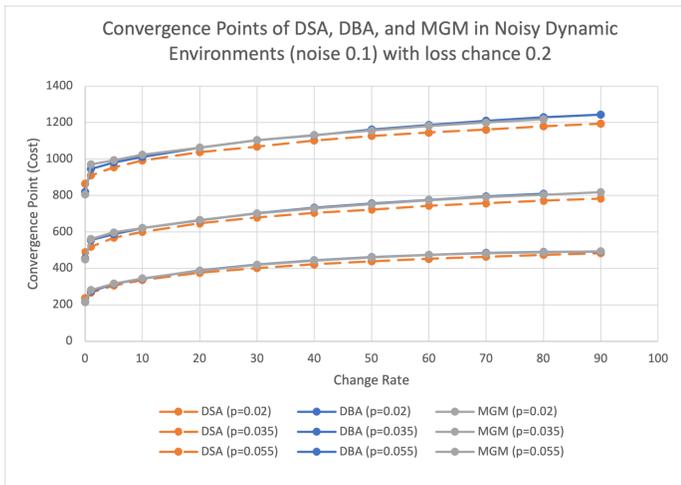


Fig. 16. Comparison of convergence points of DSA, DBA, and MGM in noisy dynamic environment with noise chance of 0.1 and loss chance of 0.2

impacted by both noise and message loss. The parameters used in this section are the ones used in the previous sections in addition to the message loss probability of 0.1, 0.2, and 0.3. All three distributed algorithms have been tested under the mentioned conditions to explore the effect of both noise and message loss.

Figures 15-23 show the result for DynDCOP problems with various densities $p = 0.02, 0.035, 0.055$ for DSA, DBA, and MGM respectively. Similar to the DynDCOP problems with noise issue and without the message loss problem, DSA had provided better results. Here, DSA outperforms two other algorithms. MGM and DBA had very similar results. Interestingly in the problems with higher density, the difference among the results becomes less apparent. This suggests that algorithms cannot keep up with the number of changes in the problems in the more dense problems. It must be noted that with a higher density amount, the number of variables that are impacted increases. Hence, the error amount produced by noise or message loss also increases.

All algorithms show a sharp increase in the convergence points than the static problems without noise or message loss. For example, for a problem with a density of 0.02 DSA, DBA, and MGM yield 236.23, 215.06, and 217.18, however, with a change rate of 1 and noise and message loss probabilities both equal to 0.1, the convergence points become 264.32, 254.94, and 256.65 respectively. With the same chance of noise and noise probability 0.2, the convergence points increase to 267.15, 273.92, and 280.81. Similarly, for all other densities, noise and message loss probabilities pattern repeats. By looking at another side of the results when the change rate is very high, it is visible that none of the algorithms can improve the solutions. Even DSA that works in one phase and is fast cannot produce better results compared to DBA and MGM.

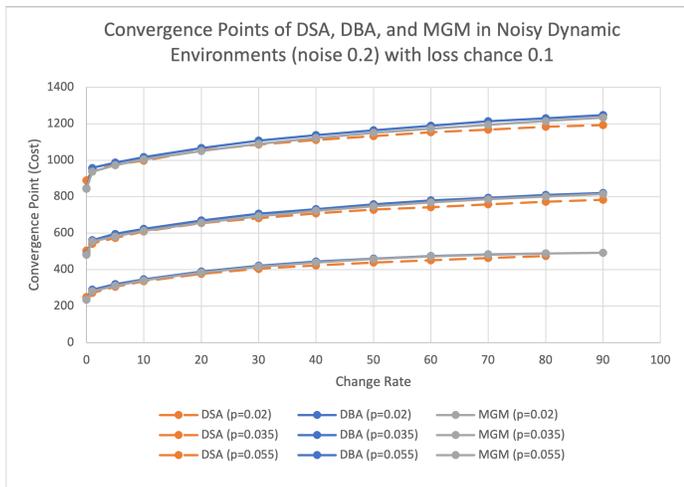


Fig. 18. Comparison of convergence points of DSA, DBA, and MGM in noisy dynamic environment with noise chance of 0.2 and loss chance of 0.1

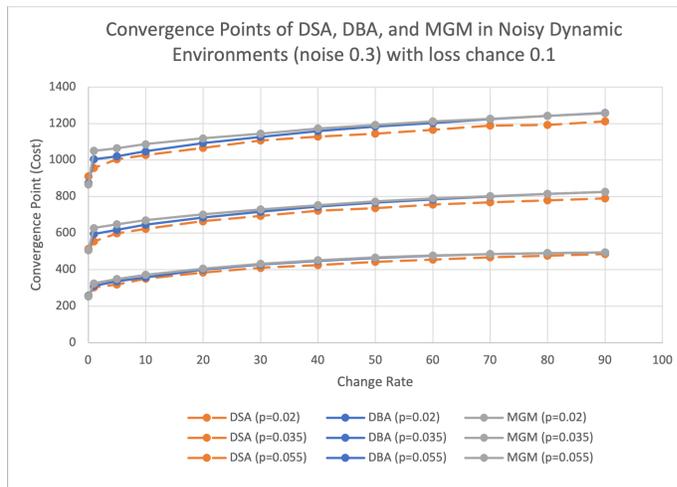


Fig. 21. Comparison of convergence points of DSA, DBA, and MGM in noisy dynamic environment with noise chance of 0.3 and loss chance of 0.1

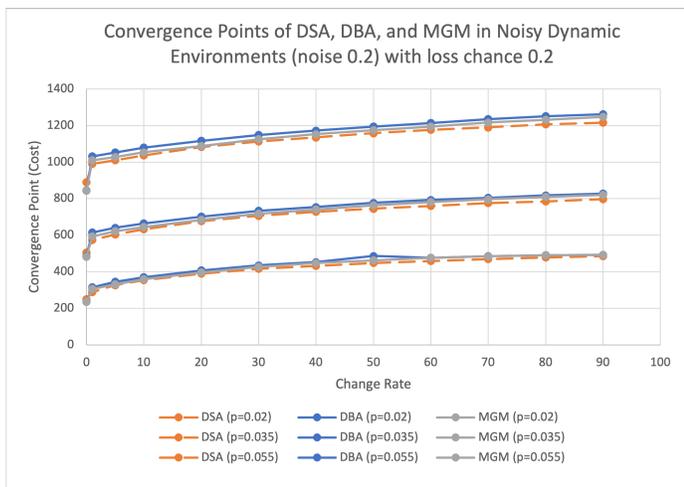


Fig. 19. Comparison of convergence points of DSA, DBA, and MGM in noisy dynamic environment with noise chance of 0.2 and loss chance of 0.2

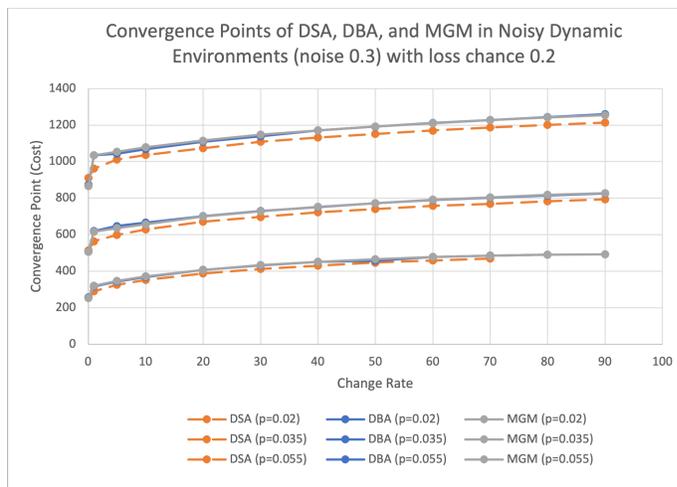


Fig. 22. Comparison of convergence points of DSA, DBA, and MGM in noisy dynamic environment with noise chance of 0.3 and loss chance of 0.2

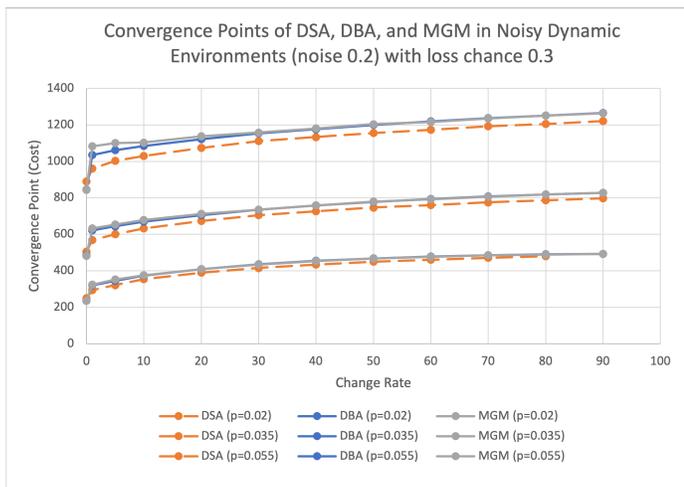


Fig. 20. Comparison of convergence points of DSA, DBA, and MGM in noisy dynamic environment with noise chance of 0.2 and loss chance of 0.3

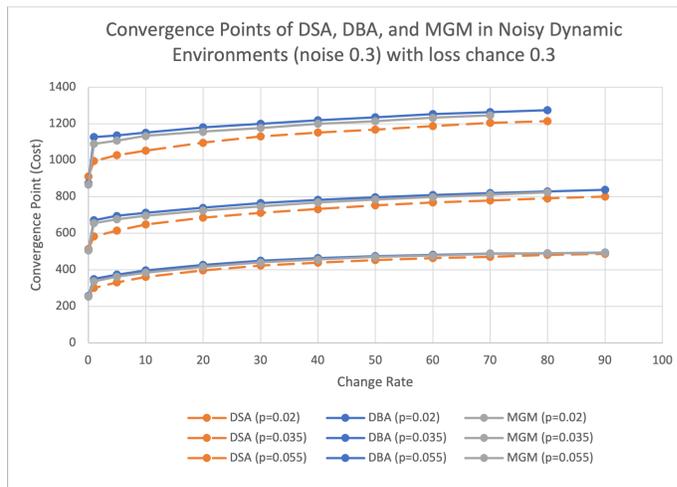


Fig. 23. Comparison of convergence points of DSA, DBA, and MGM in noisy dynamic environment with noise chance of 0.3 and loss chance of 0.3



V. CONCLUSION

The advances in network technologies have provided a robust communication environment for many distributed systems. Nevertheless, despite all the progress, not all distributed systems work in a perfect environment. Issues such as noise and message loss may dramatically impact the performance of distributed algorithms. In this study, three distributed algorithms DSA, DBA, and MGM, are tested to analyze the effect of both noise and message loss. The result confirms that all of the algorithms are highly impacted by communication instability. DSA shows better results in many cases; however, in dynamic DCOP problems, its performance is degraded. Results confirm that fast, responsive algorithms are better candidates for solving DynDCOP problems. In future works, the DynDCOP problems in which other problems' parameters such as density change over time will be studied.

ACKNOWLEDGMENT

The author would like to thank the anonymous reviewers for their time.

REFERENCES

- Alemany, M., Estes, A., Ángel Ortiz, & del Pino, M. (2021). Centralized and distributed optimization models for the multi-farmer crop planning problem under uncertainty: Application to a fresh tomato argentinean supply chain case study. *Computers & Industrial Engineering*, 153, 107048.
- Andrews, J. G., Buzzi, S., Choi, W., Hanly, S. V., Lozano, A., Soong, A. C. K., & Zhang, J. C. (2014). What will 5g be? *IEEE Journal on Selected Areas in Communications*, 32(6), 1065-1082. doi: 10.1109/JSAC.2014.2328098
- Dechter, R., & Dechter, A. (1988). Belief maintenance in dynamic constraint networks. In *Proceedings of the 6th national conference on artificial intelligence (aaai-88)* (p. 37-42).
- Kiran, D. (2019). *Chapter 35 - internet of things, production planning and control*. Butterworth-Heinemann.
- Leite, A. R., Enembreck, F., & Barthes, J.-P. A. (2014). Distributed constraint optimization problems: Review and perspectives. *Expert Systems with Applications*, 5139–5157.
- Maheswaran, R. T., Pearce, J. P., & D, M. T. (2004, September). Distributed algorithms for dcop: A graphical-game-based approach. In *Proc. Parallel and Distributed Computing Systems PDCS*, 432–439.
- Mailler, R. (2005). Comparing two approaches to dynamic, distributed constraint satisfaction. In *Proceedings of the fourth international joint conference on autonomous agents and multiagent systems (aamas)* (p. 1049-1056).
- Mailler, R., & Zheng, H. (2014). A new analysis method for dynamic, distributed constraint satisfaction. In *Proceedings of the 2014 international joint conference on autonomous agents and multiagent systems (aamas)* (Vol. 3, pp. 901–908).
- Mira, I. A. . F. (2019). Iot security threats analysis based on components, layers and devices. *American Journal of Science and Engineering*, 1–10.
- Ridgway, A., & Mailler, R. (2015). Dynamic theoretical analysis of the distributed stochastic and distributed breakout algorithms. In *Proceedings of the 14th international conference on autonomous agents and multiagent system*.
- Ridgway, A., & Mailler, R. (2015). On predictions for dynamic, self-adaptive techniques in dyndcsp's. In *2015 IEEE/WIC/ACM international conference on web intelligence and intelligent agent technology (wi-iat)* (Vol. 2, p. 265-272). doi: 10.1109/WI-IAT.2015.156
- Russell, S. J., & Norvig, P. (2020). *Artificial intelligence: A modern approach, 4th edition*. Prentice Hall.
- Samadidana, S. (2019). The effects of information stagnancy on distributed problem solving. *ProQuest Dissertations and Theses*, 166.
- Samadidana, S. (2021). Exploring the effect of noisy environment on distributed problem solving. In *2021 IEEE 11th annual computing and communication workshop and conference (ccwc)* (p. 0342-0348).
- Samadidana, S., & Mailler, R. (2017). Solving dcsp problems in highly degraded communication environments. In *Proceedings of international conference on web intelligence (wi17)* (p. 348—355).
- Samadidana, S., & Mailler, R. (2019). Using temporal awareness to improve distributed problem solving. In *IEEE 9th annual computing and communication workshop and conference (ccwc)* (pp. 342–348). Retrieved from doi : 10.1109/CCWC.2019.8666568
- Samadidana, S., Paydar, M., & Jouzdani, J. (2017). A simulated annealing solution method for robust school bus routing. *International Journal of Operational Research*, 28, 307–326. doi: https://doi.org/10.1504/IJOR.2017.081908
- Verfaille, G., & Jussien, N. (2005). Constraint solving in uncertain and dynamic environments: A survey. *Constraints*, 10(3), 253-281.
- Wahbi, M., & Brown, K. N. (2014). The impact of wireless communication on distributed constraint satisfaction. In *Proceedings of the 20th international conference on principles and practice of constraint programming* (pp. 738–754). Lyon, France.
- Yao, W., Qi, N., Wan, N., & Liu, Y. (2019). An iterative strategy for task assignment and path planning of distributed multiple unmanned aerial vehicles. *Aerospace Science and Technology*, 86, 455-464.
- Yokoo, M., & Durfee, E. H. (1991). *Distributed constraint optimization as a formal model of partially adversarial cooperation* (Tech. Rep. No. CSE-TR-101-91). Ann Arbor, MI 48109: University of Michigan.
- Yokoo, M., & K.Hirayama. (1996). Distributed breakout algorithm for solving distributed constraint satisfaction problems. *International Conference on Multi-Agent Systems(ICMAS)*, 401–408.



Zhang, W., Wang, G., & L.Wittenburg. (2002). Distributed stochastic search for constraint satisfaction and optimization:parallelism, phase transitions and performance. *In proceedings of the AAAI workshop on probabilistic Approaches in Search*, 53–59.



Study of Behaviors of Multi-Source Rauch Filters

MinhTri Tran*, Anna Kuwana, Haruo Kobayashi

Division of Electronics and Informatics, Gunma University, Kiryu 376-8515, Japan

* trantri.ks@gmail.com; kuwana.anna@gunma-u.ac.jp; koba@gunma-u.ac.jp

Abstract— This paper presents a study of behaviors of high-order multi-source Rauch filters based on the characteristics of complex functions. The advantages of Rauch's connections are an easier selection of circuit components and a simpler design in fully differential forms and complex topologies. A general superposition formula is also introduced for deriving the transfer functions of multi-source transmission networks such as fully differential amplifiers, fully differential low-pass filters, polyphase filters, complex filters, and quadrature signal generation circuits because the shapes of any motions and the characteristics of any transmission networks are analyzed based on superposition principle. The proposed Nichols chart of the self-loop function is used to do the ringing test for high-order Rauch low-pass filters because it can be easily calculated using MATLAB, simulated using SPICE, and measured in Network Analyzers. The innovation of Nichols chart of self-loop function is a useful tool for the ringing test for high-order networks. The simulation results of the phase margin at unity gain of the self-loop function in a proposed design of the 4th-order Rauch low-pass filter are 74 degrees (over-damping region), 68 degrees (critical damping region), and 59 degrees (under-damping region). The pass-band gain and the image rejection ratio of the proposed design of a 4th-order Rauch complex filter are 12 dB and 32 dB, respectively. It's shown that complex functions play an important role in mathematical models of motion in the multi-source networks.

Keywords— Ringing Test, Self-Loop Function, Nichols Chart, Rauch Filter, Multi-Source Network, Superposition Principle.

I. INTRODUCTION

Modeling of motion is one of the most important topics in physical science because everything in the universe moves (such as electrons, protons, photons, people, stars, galaxies, and so on...). Complex functions which imaginary unit j represents a 90-degree phase shift are used to describe the shapes of the natural motion in both two-dimensional and three-dimensional domains which are investigated in both time domain and frequency domain. In theoretical analysis and practical measurement, the characteristics of the natural waveforms and the real behaviors of their transmission networks are usually projected in a specific referent complex plain which is investigated at time variation or at frequency variation. In addition, sinusoidal signals are the projections of helix waves on a normal plain. Therefore, helix waves are proposed to analyze the behaviors of analogue signals such as sinusoidal signals, multi-harmonic signals, multi-phase signals in both time and frequency domains [1].

Rauch filters as a class of multi-feedback filters are popular and applied in modern electronics. These filters are often compared with Sallen-Key filters [2]. The motivation of this study on high-order multi-source Rauch filters based on complex functions comes from the high quality of the performance requirements of high-order systems. The conventional concept of loop gain is used to do the stability test in many research papers, textbooks, and commercial

products because ringing is a wasteful power. However, the theoretical calculation, the laboratory simulation and the practical measurement of loop gain are not unique, so it is not used to investigate the operating regions of high-order systems. The observation of the phase margin at unity gain of the self-loop function helps us determine the operating regions of high-order systems quickly. With the proposed superposition formula, the transfer functions of high-order complex filters and polyphase filters are easily derived and investigated in both positive and negative frequency domains. The behaviors of multi-source networks are clarified by the theoretical analysis, the laboratory simulations, and the practical measurements.

The main contributions of this work are the analysis of helix waves, the ringing test for a 4th-order fully differential Rauch low-pass filter and the derivation of the transfer function and the image rejection ratio of a 4th-order Rauch complex filter. The proposed Rauch complex filter overcomes some limitations of the conventional filters such as high power-loss in case of high-order RC polyphase filters, and large power-consumptions in case of high-order complex filters. Helix waves are considered as fundamental waveforms of the natural motion. In addition, the use of the superposition formula is a fast network analysis method for high-order multi-source systems.

This paper contains 7 sections. Some basic concepts for modelling of motion are presented in Section II. The classification of the analogue signals is presented in Section III. Section IV shows the proposed superposition formula for deriving the transfer functions of the multi-source systems. The ringing test for high-order Rauch low-pass filters is performed in Section V. The behaviors of a 4th-order Rauch complex filter are investigated in Section VI. The main points of this work are summarized in Section VII.

II. BASIC CONCEPTS FOR MODELING OF MOTION

A. Roles of motion models

This section will recall the roles of mathematical models of motion in physical sciences. The state of the object is usually investigated using Cartesian coordinate systems in both two-dimension and three-dimension domains. The shapes of objects are often projected from a referent three-dimension to a referent two-dimension domain. However, the Cartesian coordinate systems are not widely used in the mathematical models of motion because it is very difficult to express the shapes of motion of a natural signal using a pair of numerical coordinates, or a triplet of numerical coordinates at time variation. Therefore, the study of complex numbers in a referent plane continues and has been enhanced in many centuries [3].

A set of complex numbers is also called a complex function which describes the shape of a motion at time variation or the behaviour of a transmission network at frequency variation [4]. It is also known that complex

functions arise for a wide variety of reasons and are studied because they may be physically significant and hence cannot be neglected in mathematical models of motion in both time and frequency domains and in both two-dimensional and three-dimensional domains [5]. Moreover, the waveform of a natural motion is often expressed in three-dimension domain. For instance, the shape of a water drop from a faucet in three-dimension domain is given in (1). The waveform of the water drop is shown in Fig. 1.

$$V(t) = \left\{ \text{Re}(t) = t \cos(t); \text{Im}(t) = t \sin(t); \text{Al}(t) = -\frac{10}{2\sqrt{t}} \sin(t) \right\} \quad (1)$$

The quantitative tools and concepts needed to understand the physical nature of motion in multi-source networks are challenging. It is difficult to investigate the characteristics of a real motion of multi-source networks in both theoretical analysis and practical measurement because the basic theories for large-scale systems are not widely studied [6]. Therefore, the behaviours of a real motion are usually investigated in two-dimensional domain where the shape of waveforms and characteristics of transmission networks are projected in the specific referent complex planes and observed at time variation or at frequency variation [7].

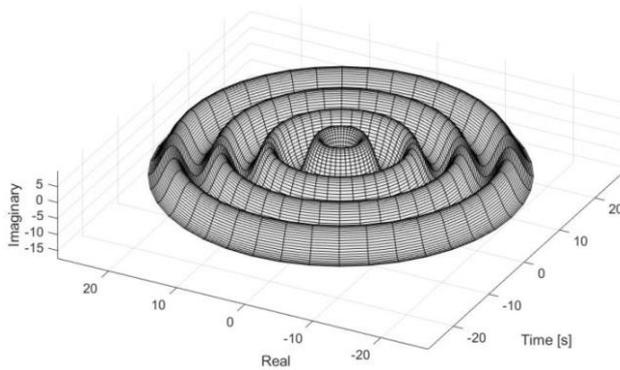


Fig. 1. Waveform of a water drop in a three-dimensional domain.

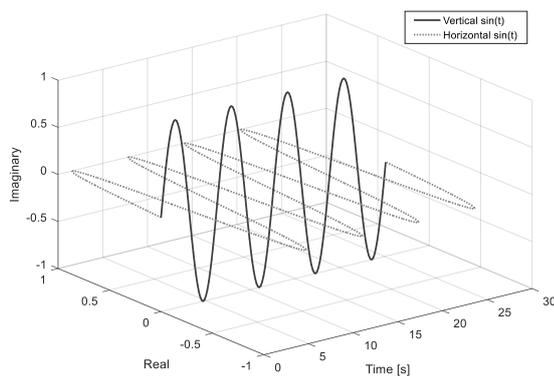


Fig. 2. Shapes of $\sin(t)$ and $j\sin(t)$ on the vertical and horizontal planes.

B. Meaning imaginary unit in complex numbers

The meaning of imaginary unit in two-dimensional and three-dimensional domains is explained in this section. Complex numbers can be efficiently used for the solutions of

polynomial equations because it holds all of the other numbers such as real numbers, rational numbers, and irrational numbers. Although the basic concepts of complex numbers are well known in a simplest equation $x^2 + 1 = 0$, the meaning of imaginary unit j and its applications might not be understood [8].

It's well known that Euler's formula is a mathematical formula in complex analysis that establishes the fundamental relationship between the trigonometric functions and the complex exponential function. Euler's formula is ubiquitous in mathematics, physics, and engineering because Euler's formula evaluates to $e^{j\pi} + 1 = 0$. It is very easy to proof this formula using a power series for sine function ($\text{Asin}(\omega_0 t + \Theta_0)$) and cosine function ($\text{Acos}(\omega_0 t + \Theta_0)$) at a specific time in (2). However, it is very difficult to clarify the differences of signal spectrum and system spectrum using Euler's formula or a power series of a complex number because the time and frequency domains of a system are analysed separately [9]. In other words, the waveform and spectrum in time and frequency domains are not well defined in the Euler's formula. Moreover, sinusoidal waves are the shapes of the natural periodic motions which are projected on a referent plane [10]. Therefore, the Euler's formula is not used in the mathematical model of motion because sine and cosine waves on two different planes cannot be added.

$$\begin{aligned} e^\theta &= 1 + \frac{\theta}{1!} + \frac{\theta^2}{2!} + \frac{\theta^3}{3!} + \frac{\theta^4}{4!} + \dots = \sum_{n=0}^{\infty} \frac{\theta^n}{n!} \\ \cos(\theta) &= 1 - \frac{\theta^2}{2!} + \frac{\theta^4}{4!} - \frac{\theta^6}{6!} + \dots = \sum_{n=0}^{\infty} \frac{(-1)^n \theta^{2n}}{(2n)!} \\ \sin(\theta) &= \theta - \frac{\theta^3}{3!} + \frac{\theta^5}{5!} - \frac{\theta^7}{7!} + \dots = \sum_{n=0}^{\infty} \frac{(-1)^n \theta^{2n+1}}{(2n+1)!} \\ e^{j\theta} &= \sum_{n=0}^{\infty} \frac{(-1)^n \theta^{2n}}{(2n)!} + j \sum_{n=0}^{\infty} \frac{(-1)^n \theta^{2n+1}}{(2n+1)!} = \cos(\theta) + j \sin(\theta) \end{aligned} \quad (2)$$

For example, the sine functions $V_1(t) = \text{Asin}(\omega_0 t + \Theta_0)$ and $V_2(t) = j\text{Asin}(\omega_0 t + \Theta_0)$ where the amplitude is one ($A=1$), the angular frequency ω_0 is one ($\omega_0=2\pi f_0=1$), and the original phase is zero ($\Theta_0 = 0$) on the horizontal and vertical planes (or real and imaginary planes) are given in (3). Fig. 2 shows the waveforms of two sinusoidal functions $V_1(t)=\sin(t)$ and $V_2(t)=j\sin(t)$ on the vertical and horizontal planes at time variation.

$$\begin{aligned} V_1(t) &= \{ \text{Re}(t) = \sin(t); \text{Im}(t) = 0; \text{Al}(t) = t \} \\ V_2(t) &= \{ \text{Re}(t) = 0; \text{Im}(t) = \sin(t); \text{Al}(t) = t \} \end{aligned} \quad (3)$$

There are three main factors that set up a model of motion including signal source, transmission network, and time [11]. In time domain, signal sources can be periodic signals (sinusoidal waves, square waves, and so on...), or non-periodic signals. The spectra of signal sources are analysed in frequency domain. Moreover, signal sources can be divided into the single-phase source and the multi-phase source according to the single-ended network and the multi-source network. The transmission network will affect the phase and the amplitude of the signal source in different ways. Transmission networks are usually analyzed in frequency domain. Complex functions are used to express both signal sources and transmission networks. Imaginary unit j represents the 90-degree phase shift from a referent

axis in two-dimension domain, or a referent plane in three-dimension domain [12].

In mathematical models of signal sources, complex functions can be expressed in both time and frequency domains. Sine and cosine waves are the shapes of helix waves in different planes. The amplitude in time domain and the magnitude in frequency domain of a sinusoidal signal are not the same. The phase-order of multi-phase signals determines the polarity of complex signals which are called positive and negative polyphase signals [13]. Up to now, the fundamental concepts of helix waves, single-ended signals, complex signals, and polyphase signals are not well defined in both time and frequency domains. In addition, multi-source transmission networks are widely used in wireless communication systems such as fully differential amplifiers, fully differential low-pass filters, polyphase filters, complex filters, and quadrature signal generation circuits while a general stability test for high-order multi-source systems, and a superposition formula for deriving the transfer functions and the image rejection ratios of multi-phase systems are still not introduced [14].

C. Limitations of conventional complex functions analysis

There are some limitations of the conventional methods such as ordinary differential equations, superposition theorem, Nyquist stability condition. Ordinary differential equations are used in many mathematical models of motions [15]. Based on Euler's formula, a second-order ordinary differential equation can be solved easily. In addition, Fourier's transforms, Laplace's transforms, and Hilbert's transforms can be also used to analyse some differential equations of some basic multi-source mechatronic networks in the frequency domain. However, the inability to explicitly solve high-order differential equations arises in elementary courses. One frequently gets the impression that these equations can usually be solved that explicit formulas for the solutions can usually be found in terms of powers, exponentials, and trigonometric functions which are complex functions. Therefore, mathematical models for energy propagations of objects based on complex functions are studied many recent decades [16].

In mathematical models of transmission networks, complex functions can be plotted in Nyquist diagram, Bode plot, and Nichols chart. Kirchhoff's laws lead to the conventional superposition principle for mathematical modeling that describe the relationship between the input and output of electronic systems [17]. However, it takes several times to analyze the behaviors of multi-source networks because the effects of each source should be considered independently. It requires that some signal sources should be removed and replaced without affecting the final result. The application of the conventional superposition principle becomes a painful experience when a high-order multi-phase network is analyzed because it requires a huge amount of hand calculation [18]. Therefore, a superposition formula is introduced for deriving the transfer functions and the image rejection ratios in these multi-phase networks.

In high-order electronic feedback and non-feedback systems, ringing or overshoot voltage causes extra current to flow, which leads to thereby wasting energy, causing extra heating of the components, and making the system unstable [19]. Ringing occurs in both passive and active circuits, and it happens in both feedback and non-feedback systems. Ringing makes the feedback systems unstable. Therefore, the

stability test should be done. The stability test is a valuable tool for analyzing high-order systems. Learners, however, often find this topic confusing and difficult to understand [20]. Some reasons for this difficulty and confusion are the derivation of the loop gain in feedback systems and the assumptions of the feedback analysis [21]. Furthermore, it is difficult to analyze the response of overshoot and undershoot of a high-order network, because the corresponding inverse Laplace transform of the response is difficult to compute analytically [22].

The Nyquist diagram of loop gain cannot be applied for high-order linear electronic systems because Nyquist's stability condition does not show the operating regions of a linear network. Moreover, the Nichols chart of loop gain is only used in MATLAB simulation. And, it is not widely used in practical measurements. As a result, Nichols chart of the self-loop function is used to study the operating regions and compute the phase margins of high-order multi-source systems [23]. The inspection of the Nichols chart allows the determination of the exact values of the phase margin at unity gain of the self-loop function in a transfer function. Hence, there is no need to use the Nyquist criterion which is an important stability test with applications to adaptive feedback systems. Furthermore, the phase margin allows the prediction of ringing that drives the system unstable. Therefore, the investigation of phase margin at unity gain of the self-loop function has been the focal point of many extensive research works [24].

III. CLASSIFICATION OF ANALOGUE SIGNALS

A. Behaviors of helix waves

In this section, the properties of helix waves are presented. The sine and cosine waves are the projections of the helix waves on different reference planes. Helix waves are the fundamental waveforms in mathematical models of motion [25]. The patterns of motion in the coil (or inductor), spring pendulum are helix waves. In general, helix waves can be used to express the behaviours of a periodic motion at time variation. The polarity of a helix wave is conventionally determined based on the direction of motion in the transmission network. In the anti-clockwise direction, a positive helix function $V_{he+}(t)$ is written in (4). A positive helix function $V_{he+}(t) = Ahe(\omega_0 t + \theta_0)$ can be expressed separately in three axis functions: real function $Re(t) = V_{cos+}(t) = A \cos(\omega_0 t + \theta_0)$, imaginary function $Im(t) = V_{sin+}(t) = A \sin(\omega_0 t + \theta_0)$, and altitude function $Al(t) = t$. Here, the amplitude is A and the positive rotation angle function is $\Theta_p(t) = \omega_0 t + \theta_0$. Fig. 3 shows the positive helix function $V_{he+}(t) = he(t)$ at a specific angular frequency ($\omega_0 = 2\pi f_0 = 1$) on the complex plane at time variation. The spectrum of the positive helix wave is given in (5).

$$V_{he+}(t) = Ahe(\omega_0 t + \theta_0) = \begin{cases} Re(t) = A \cos(\omega_0 t + \theta_0) \\ Im(t) = A \sin(\omega_0 t + \theta_0) \\ Al(t) = t \end{cases} \quad (4)$$

$$|V_{he+}(T_0)| = |Ahe(\omega_0 T_0 + \theta_0)| = A\sqrt{2}e^{j(\omega_0 T_0 + \theta_0)} \quad (5)$$

In the clockwise direction, a general negative helix function $V_{he-}(t)$ is written in (6). The negative helix function $V_{he-}(t)$ in three separated axis functions are real function

$\text{Re}(t)=V_{\cos}(t) = A\cos(-\omega_0t-\Theta_0)$, imaginary function $\text{Im}(t) = V_{\sin}(t) = A\sin(-\omega_0t-\Theta_0)$, and altitude function is $\Theta_n(t)=-\omega_0t-\Theta_0$. Fig. 4 shows the negative helix function $V_{he-}(t) = he(t)$ at the specific angular frequency ($\omega_0=2\pi f_0=1$) in the complex plane. The spectrum of the negative helix wave is given in (7). In the time domain, the amplitudes and the phases of helix, cosine and sine waves are the same. However, the amplitude in time domain and the magnitude in frequency domain of these waves are not the same.

$$V_{he-}(t) = Ahe(-\omega_0t - \theta_0) = \begin{cases} \text{Re}(t) = A \cos(-\omega_0t - \theta_0) \\ \text{Im}(t) = A \sin(-\omega_0t - \theta_0) \\ \text{Al}(t) = t \end{cases} \quad (6)$$

$$|V_{he-}(T_0)| = |Ahe(-\omega_0T_0 - \theta_0)| = A\sqrt{2}e^{-j(\omega_0T_0 + \theta_0)} \quad (7)$$

As the amplitude of a helix wave is one, its spectrum is a square root of two. On the contrary, the spectrum of a sine wave is a half of a square root of two when its amplitude is one. The spectra of helix and sine waves are shown in Fig. 5. In addition, the impedances of transmission networks may change the amplitude and phase of signal sources at different frequencies. Therefore, the behaviours of transmission networks are usually analysed in frequency domain. Understanding of helix functions is being able to think flexibly the properties of natural waves in both two-dimensional and three-dimensional domains. Figs. 3, 4, and 5 clearly demonstrate the differences between the amplitude in time domain and the magnitude in frequency domain of helix waves and sinusoidal waves.

In a referent normal plane, waveforms can be divided into single-ended signals, multi-harmonic signals, and multi-phase signals. In a single-ended system, the sinusoidal functions and their spectra are easily derived based on the properties of helix waves. The wave functions and the spectra of sine and cosine waves are given in (8)-(11).

$$V_{\cos}(t) = A\cos(\omega_0t + \theta_0) = \text{Re}\{V_{he+}(t)\} = \text{Re}\{V_{he-}(t)\} \quad (8)$$

$$= \frac{1}{2}(\text{Re}\{V_{he+}(t)\} + \text{Re}\{V_{he-}(t)\})$$

$$|V_{\cos}(T_0)| = |A\cos(\omega_0T_0 + \theta_0)| \quad (9)$$

$$= \frac{A\sqrt{2}}{2}(e^{j(\omega_0T_0 + \theta_0)} + e^{-j(\omega_0T_0 + \theta_0)})$$

$$V_{\sin}(t) = A\sin(\omega_0t + \theta_0) = \text{Im}\{V_{he+}(t)\} = -\text{Im}\{V_{he-}(t)\} \quad (10)$$

$$= \frac{1}{2}(\text{Im}\{V_{he+}(t)\} - \text{Im}\{V_{he-}(t)\})$$

$$|V_{\sin}(T_0)| = |A\sin(\omega_0T_0 + \theta_0)| \quad (11)$$

$$= \frac{A\sqrt{2}}{2}(e^{j(\omega_0T_0 + \theta_0 + \frac{\pi}{2})} + e^{-j(\omega_0T_0 + \theta_0 + \frac{\pi}{2})})$$

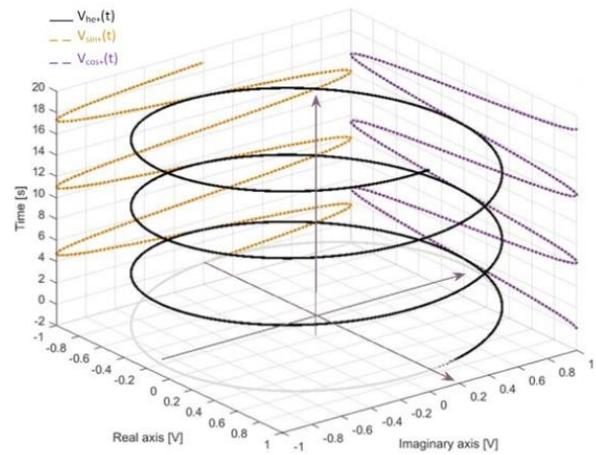


Fig. 3. Waveforms of $V_{he-}(t)$, $V_{sin+}(t)$, and $V_{cos+}(t)$ functions.

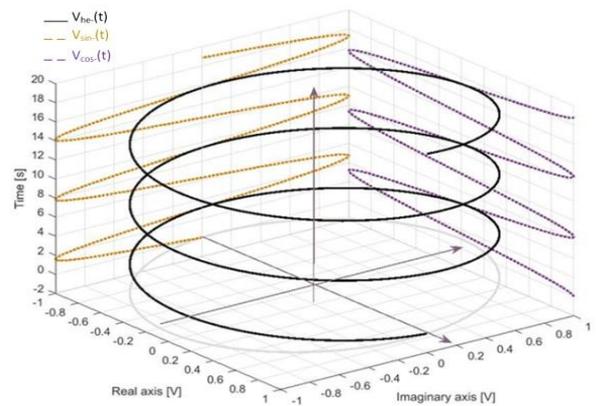


Fig. 4. Waveforms of $V_{he+}(t)$, $V_{sin-}(t)$, and $V_{cos-}(t)$ functions.

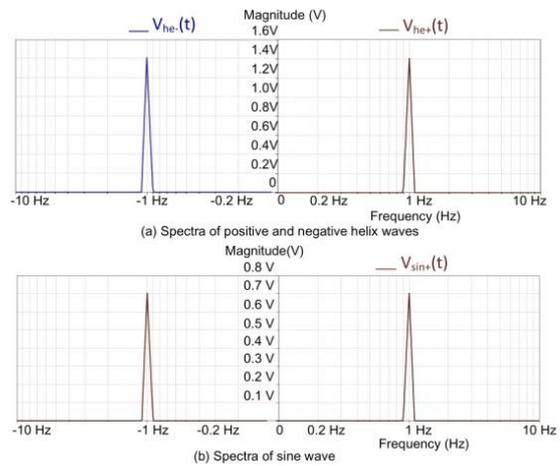


Fig. 5. Spectra of helix and sine waves.

B. Characteristics of multi-harmonic signals

In this section, the characteristics of multi-harmonic signals are analysed. It's well known that a square wave is a multi-harmonic signal. Fourier series expansion shows that any periodic signal can be written as a sum of harmonic signals having exactly integral number of cycles within the period time [26]. In other words, power series and Fourier's series are the simplified forms of the superposition principle

for the multi-harmonic signals. In time domain, a square signal $V_{sq}(t) = A \text{sq}(\omega_0 t + \theta_0)$ can be defined in (12). Fig. 6 shows the waveform of a square signal and a ninth-order multi-harmonic sinusoidal wave. In laboratory simulation and practical measurement, the signal spectrum and system spectrum are usually investigated in the positive frequency domain or a half-side spectrum [27]. In addition, the concept of negative frequency for signal and system is still not well defined. Hence, the signal spectra of the common waves are written in the forms of half-side spectrum. Based on the superposition principle for analogue signals, the signal spectrum for the square wave is given in (13).

$$V_{sq}(t) = A \text{sq}(\omega_0 t + \theta_0) = A \left(\frac{1}{2} + \sum_{n=1}^{\infty} \frac{\sin[(2n+1)\omega_0 t + \theta_0]}{(2n+1)\pi} \right) \quad (12)$$

$$|V_{sq}(T_0)| = |A \text{sq}(\omega_0 T_0 + \theta_0)| = A \left(\frac{1}{2} + \sum_{n=1}^{\infty} \frac{\sqrt{2} e^{j((2n+1)\omega_0 T_0 + \theta_0 + \frac{\pi}{2})}}{2(2n+1)\pi} \right) \quad (13)$$

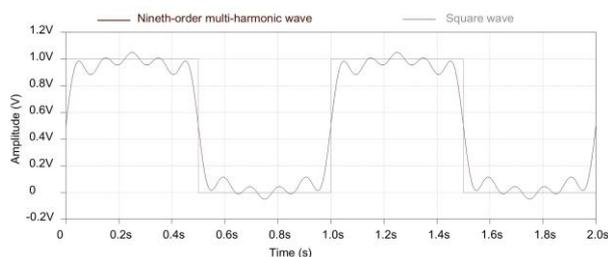


Fig. 6. Square wave and a ninth-order Fourier series.

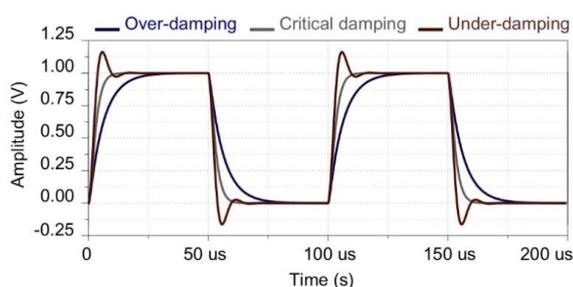


Fig. 7. Waveforms of output square signals.

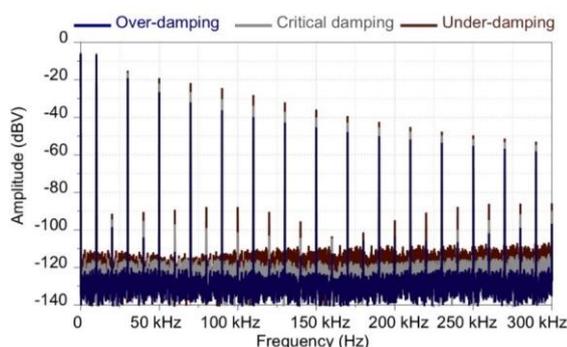


Fig. 8. Spectra of output square signals.

To do the ringing test for high-order systems, a square wave is usually used to put in the input port. Depending on the operating region of the system, the output waveforms can be over-damping, or critical damping, or under-damping as shown in Fig. 7. The spectra of the output square signals are plotted in Fig. 8. The ringing test will be analysed in detail in section V.

C. Characteristics of multi-phase signals

This section deals with the complex functions for the multi-phase signals in wireless communication systems. Multi-phase signals are used in wireless communication systems such as filters, amplifiers, frequency converters, and signal generators. The characteristics of the input sources are determined based on the transmission networks which are called single-ended systems, differential systems, complex systems, and polyphase systems [28]. So, single-ended signals such as sinusoidal signals and multi-harmonic signals are used in single-ended systems. The multi-phase signals can be divided into differential signals, complex signals, and polyphase signals. These signals are commonly used in fully differential amplifiers, polyphase filters, complex filters, and quadrature signal generation circuits. The normal sinusoidal signal on the real plane is the cosine wave. In the differential networks, there are two opposite-phase signal sources which are called differential signals. Let $V(t) = A \cos(\omega_0 t + \theta_0)$ be a normal voltage, a differential signal $S_{dif}\{V_1, V_2\}$ can be defined as in (14). Hence, the half-side spectra in a referent plane of the differential signals are given in (15).

$$S_{dif}\{V_1; V_2\} = \{1; (\pm j)^2\} A \cos(\omega_0 t + \theta_0) \quad (14)$$

$$|S_{dif}\{V_1; V_2\}| = \{1; e^{\pm j\pi}\} |A \cos(\omega_0 T_0 + \theta_0)| = \{1; e^{\pm j\pi}\} \frac{A\sqrt{2}}{2} e^{j(\omega_0 T_0 + \theta_0)} \quad (15)$$

In the complex systems, there are two signal sources which are called complex signals. On the phase-order plane, the phase-orders of these signals are clockwise and anti-clockwise [29]. Let $V_p(t) = A \cos(\omega_0 t + \theta_0)$ be a normal voltage, a positive complex signal $S_{cp}\{V_{p1}, V_{p2}\}$ can be defined as in (16). Fig. 9 shows the waveforms of the positive complex signals on a real referent plane. The concepts of positive and negative angular frequencies are used to define the phase directions of complex signals in the frequency domain. The half-side spectra in a referent plane of the positive complex signals are given in (17). The anti-clockwise phase-order is a positive complex signal as shown in Fig. 10(c).

$$S_{cp}\{V_{p1}; V_{p2}\} = \{1; (+j)\} A \cos(\omega_0 t + \theta_0) \quad (16)$$

$$|S_{cp}\{V_{p1}; V_{p2}\}| = \left\{1; e^{+j\frac{\pi}{2}}\right\} |A \cos(\omega_0 T_0 + \theta_0)| = \left\{1; e^{+j\frac{\pi}{2}}\right\} \frac{A\sqrt{2}}{2} e^{j(\omega_0 T_0 + \theta_0)} \quad (17)$$

And, let $V_n(t) = A\cos(\omega_0 t + \Theta_0)$ be a normal voltage, a negative complex signal $S_{cn}\{V_{n1}, V_{n2}\}$ can be defined as in (18). The half-side spectra in a referent plane of the negative complex signals are given in (19). The clockwise phase-order is a negative complex signal as shown in Fig. 10(d).

$$S_{cn}\{V_{n1}; V_{n2}\} = \{1; (-j)\} A\cos(-\omega_0 t - \theta_0) \quad (18)$$

$$\begin{aligned} |S_{cn}\{V_{n1}; V_{n2}\}| &= \left\{1; e^{-j\frac{\pi}{2}}\right\} \left|A\cos(-\omega_0 T_0 - \theta_0)\right| \\ &= \left\{1; e^{-j\frac{\pi}{2}}\right\} \frac{A\sqrt{2}}{2} e^{j(-\omega_0 T_0 - \theta_0)} \end{aligned} \quad (19)$$

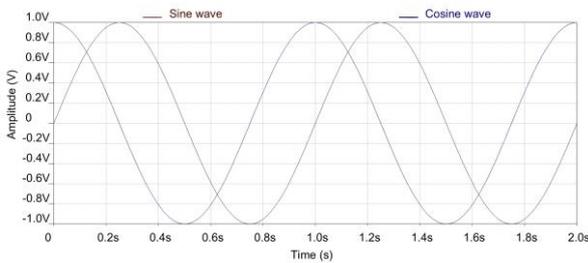


Fig. 9. Waveforms of sine and cosine functions in time domain.

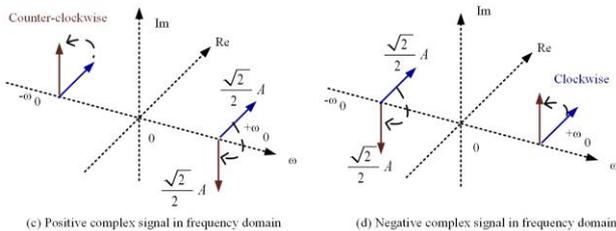
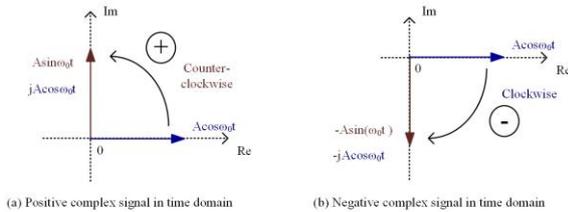


Fig. 10. Vectors of complex signals in time and frequency domains.

In polyphase networks, there are four signal sources which are called polyphase signals. In order words, polyphase signals are the differential forms of complex signals. Let $V_p(t) = A\cos(\omega_0 t + \Theta_0)$ be a normal voltage, a positive polyphase signal $S_p\{V_{p1}, V_{p2}, V_{p3}, V_{p4}\}$ can be defined as in (20). The phase-order of polyphase signals is also used to define the polarity of the multi-phase signals. The half-side spectra in a referent plane of the positive polyphase signals are given in (21).

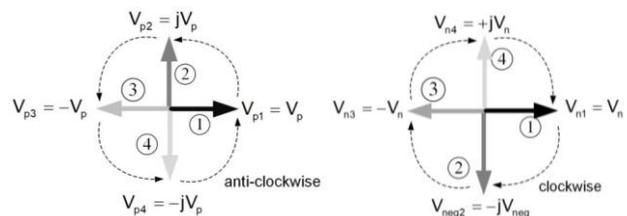
$$S_p\{V_{p1}; V_{p2}; V_{p3}; V_{p4}\} = \{1; (+j); (+j)^2; (+j)^3\} A\cos(\omega_0 t + \theta_0) \quad (20)$$

$$\begin{aligned} |S_p\{V_{p1}; V_{p2}; V_{p3}; V_{p4}\}| &= \left\{1; e^{+j\frac{\pi}{2}}; e^{+j\pi}; e^{+j\frac{3\pi}{2}}\right\} \left|A\cos(\omega_0 T_0 + \theta_0)\right| \\ &= \left\{1; e^{+j\frac{\pi}{2}}; e^{+j\pi}; e^{+j\frac{3\pi}{2}}\right\} \frac{A\sqrt{2}}{2} e^{j(\omega_0 T_0 + \theta_0)} \end{aligned} \quad (21)$$

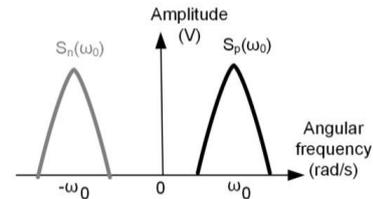
And, let $V_n(t) = A\cos(\omega_0 t + \Theta_0)$ be a normal voltage, a negative polyphase signal $S_n\{V_{n1}, V_{n2}, V_{n3}, V_{n4}\}$ can be defined as in (22). The half-side spectra in a referent plane of the negative complex signals are given in (23). The clockwise phase-order is a negative polyphase signal and vice versa as shown in Fig. 11(a). The angular frequency variables ω and $-\omega$ are proposed to represent these signals on the angular frequency plane as described in Fig. 11(b).

$$S_n\{V_{n1}; V_{n2}; V_{n3}; V_{n4}\} = \{1; (-j); (-j)^2; (-j)^3\} A\cos(-\omega_0 t - \theta_0) \quad (22)$$

$$\begin{aligned} |S_n\{V_{n1}; V_{n2}; V_{n3}; V_{n4}\}| &= \left\{1; e^{-j\frac{\pi}{2}}; e^{-j\pi}; e^{-j\frac{3\pi}{2}}\right\} \left|A\cos(-\omega_0 T_0 - \theta_0)\right| \\ &= \left\{1; e^{-j\frac{\pi}{2}}; e^{-j\pi}; e^{-j\frac{3\pi}{2}}\right\} \frac{A\sqrt{2}}{2} e^{j(-\omega_0 T_0 - \theta_0)} \end{aligned} \quad (23)$$



(a) Phase-order of positive and negative polyphase signals.



(b) Polyphase signals on the angular frequency plane.

Fig. 11. Characteristics of positive and negative polyphase signals.

D. Summary of characteristics of common analogue signals

In this section, the characteristics of waveforms and spectra of some common single-ended, multi-harmonic, and multi-phase signals are summarized. TABLE I shows the properties of helix signals, sinusoidal signals, square waves, differential signals, complex signals, polyphase signals in both time and frequency domains.

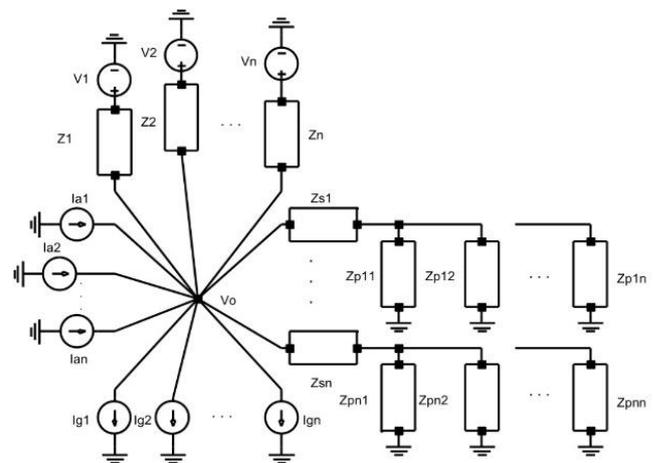


Fig. 12. Voltage at one node caused by multiple sources.

TABLE I. SIGNAL SPECTRUM OF COMMON ANALOG SIGNALS.

Signal type	Time domain	Half-side spectrum
Positive helix	$Ahe^{j(\omega_0 t + \theta_0)}$	$A\sqrt{2}e^{j(\omega_0 T_0 + \theta_0)}$
Negative helix	$Ahe^{-j(\omega_0 t + \theta_0)}$	$A\sqrt{2}e^{j(-\omega_0 T_0 - \theta_0)}$
Cosine	$A \cos(\omega_0 t + \theta_0)$	$\frac{A\sqrt{2}}{2} e^{j(\omega_0 T_0 + \theta_0)}$
Sine	$A \sin(\omega_0 t + \theta_0)$	$\frac{A\sqrt{2}}{2} e^{j(\omega_0 T_0 + \theta_0 + \frac{\pi}{2})}$
Square	$Asq(\omega_0 t + \theta_0)$ $= A \left(\frac{1}{2} + \sum_{n=1}^{\infty} \frac{\sin[(2n+1)\omega_0 t + \theta_0]}{(2n+1)\pi} \right)$	$ Asq(\omega_0 T_0 + \theta_0) $ $= A \left(\frac{1}{2} + \sum_{n=1}^{\infty} \frac{\sqrt{2} e^{j(2n+1)\omega_0 T_0 + \theta_0 + \frac{\pi}{2}}}{2(2n+1)\pi} \right)$
Differential cosine	$\{1; (\pm j)^2\} A \cos(\omega_0 t + \theta_0)$	$\{1; e^{\pm j\pi}\} \frac{A\sqrt{2}}{2} e^{j(\omega_0 T_0 + \theta_0)}$
Differential sine	$\{1; (\pm j)^2\} A \sin(\omega_0 t + \theta_0)$	$\{1; e^{\pm j\pi}\} \frac{A\sqrt{2}}{2} e^{j(\omega_0 T_0 + \theta_0 + \frac{\pi}{2})}$
Positive complex	$\{1; +j\} A \cos(\omega_0 t + \theta_0)$	$\left\{1; e^{+j\frac{\pi}{2}}\right\} \frac{A\sqrt{2}}{2} e^{j(\omega_0 T_0 + \theta_0)}$
Negative complex	$\{1; -j\} A \cos(-\omega_0 t - \theta_0)$	$\left\{1; e^{-j\frac{\pi}{2}}\right\} \frac{A\sqrt{2}}{2} e^{j(-\omega_0 T_0 - \theta_0)}$
Positive polyphase	$\{1; +j; (+j)^2; (+j)^3\} A \cos(\omega_0 t + \theta_0)$	$\left\{1; e^{+j\frac{\pi}{2}}; e^{+j\pi}; e^{+j\frac{3\pi}{2}}\right\}$ $\frac{A\sqrt{2} e^{j(\omega_0 T_0 + \theta_0)}}{2}$
Negative polyphase	$\{1; -j; (-j)^2; (-j)^3\} A \cos(-\omega_0 t - \theta_0)$	$\left\{1; e^{-j\frac{\pi}{2}}; e^{-j\pi}; e^{-j\frac{3\pi}{2}}\right\}$ $\frac{A\sqrt{2} e^{j(-\omega_0 T_0 - \theta_0)}}{2}$

IV. PROPOSAL OF SUPERPOSITION FORMULA

A. Definition of superposition formula

In this section, a superposition formula is proposed to derive the transfer function in multi-source networks. The main purpose of this formula is that the actuating voltage and current sources can be expressed separately with the node voltage, and the effects of these sources can be computed at one time. Let $V_o(t)$ be a general voltage at one node. Assume that current $I_i(t)$ and voltage $V_i(t)$ are the actuating current and voltage sources, the total currents caused by a node voltage $V_o(t)$ are equal to the sum of currents distributed by each actuating source. The direction of the current source is positive as its arrow goes ahead into a node, and negative as its arrow goes toward the ground as shown in Fig. 12. The ahead-arrow and ground-arrow current sources are denoted by $I_a(t)$ and $I_g(t)$. The voltage node equation for multiple currents and multiple voltages is expressed in (24). If the impedances are divided into small branches, voltage at one node will be calculated by the impedances of every branch [30]. Here, Z_{si} are sub-series impedances, and Z_{pii} are sub-parallel impedances. The simplified superposition theorem formula is a standard nodal analysis equation for all types of actuating sources

(direct current/voltage sources, alternating current/voltage sources, dependent and independent current/voltage sources, multiphase current/voltage sources), and all types of loads (resistor, capacitor, inductor, transistor) [31].

$$V_o(t) \left(\sum_{i=1}^n \frac{I_i}{Z_i} + \sum_{i=1}^n \frac{1}{Z_{si}} + \frac{1}{\sum_{k=1}^n \frac{1}{Z_{pik}}} \right) = \sum_{i=1}^n \left(\frac{V_i(t)}{Z_i} + I_{ai}(t) - I_{gi}(t) \right) \quad (24)$$

B. Behaviour of 4th-order RC polyphase filter

In this section, the behaviors of a 4th-order polyphase filter shall be analyzed. Fig. 13 shows the schematic diagram of a 4th-order RC polyphase filter. As the input signal sources are the positive polyphase signals $S_p\{V_1; V_2; V_3; V_4\} = \{1; (+j); (+j)^2; (+j)^3\} V(t)$, after the proposed superposition theorem is applied at each node, the current node equations at nodes V_a, V_b, V_c , and V_{out} are derived in (25). The impedances in the sub-branches are calculated by the impedances of every branch [32].

$$V_{out} \left(\frac{1}{Z_{C1}} + \frac{1}{R_1} \right) = \frac{V_a}{R_1} + \frac{(+j)^3 V_a}{Z_{C1}}; V_a \left(\frac{1}{Z_{C2}} + \frac{1}{R_2} + \frac{2}{M_A} \right) = \frac{V_a}{R_2} + \frac{(+j)^3 V_b}{Z_{C2}}; \quad (25)$$

$$V_b \left(\frac{1}{Z_{C3}} + \frac{1}{R_3} + \frac{1}{M_{B1}} + \frac{1}{M_{B2}} \right) = \frac{V_c}{R_3} + \frac{(+j)^3 V_c}{Z_{C3}};$$

$$V_c \left(\frac{1}{Z_{C4}} + \frac{1}{R_4} + \frac{1}{M_{C1}} + \frac{1}{M_{C2}} \right) = \frac{V_{in}}{R_4} + \frac{(+j)^3 V_{in}}{Z_{C4}};$$

$$M_A = R_1 + Z_{C1}; M_{B1} = R_2 + \frac{1}{\frac{1}{Z_{C2}} + \frac{1}{M_A}}; M_{B2} = Z_{C2} + \frac{1}{\frac{1}{R_2} + \frac{1}{M_A}}$$

$$M_{C1} = R_3 + \frac{1}{\frac{1}{Z_{C3}} + \frac{1}{M_{B1}} + \frac{1}{M_{B2}}}; M_{C2} = Z_{C3} + \frac{1}{\frac{1}{R_3} + \frac{1}{M_{B1}} + \frac{1}{M_{B2}}}$$

Then, the transfer function of this polyphase filter in the positive frequency domain is given in (26).

$$H_P(\omega) = \frac{b_{0P}(j\omega)^4 + b_{1P}(j\omega)^3 + b_{2P}(j\omega)^2 + b_{3P}j\omega + 1}{a_0(j\omega)^4 + a_1(j\omega)^3 + a_2(j\omega)^2 + a_3j\omega + 1}; \quad (26)$$

As the input sources are negative polyphase signals $S_n\{V_1; V_2; V_3; V_4\} = \{1; (-j); (-j)^2; (-j)^3\} V(t)$, the transfer function of the 4th-order polyphase filter in the negative frequency domain is given in (27).

$$H_N(\omega) = \frac{b_{0N}(j\omega)^4 + b_{1N}(j\omega)^3 + b_{2N}(j\omega)^2 + b_{3N}j\omega + 1}{a_0(j\omega)^4 + a_1(j\omega)^3 + a_2(j\omega)^2 + a_3j\omega + 1}; \quad (27)$$

Hence, the image rejection ratio of this polyphase filter is derived in (28). The values of the constant variables of the 4th-order RC polyphase filter are given in TABLE II. The entire device parameters of the 4th-order RC polyphase filter are summarized in TABLE III. Fig. 14 shows the simulation result of the proposed design of 4th-order polyphase filter. The image signal rejection ratio of this model is 36 dB. It is properly considered for a narrow band low-IF receiver.



$$IRR(\omega) = \frac{b_{0P}(j\omega)^4 + b_{1P}(j\omega)^3 + b_{2P}(j\omega)^2 + b_{3P}j\omega + 1}{b_{0N}(j\omega)^4 + b_{1N}(j\omega)^3 + b_{2N}(j\omega)^2 + b_{3N}j\omega + 1}; \quad (28)$$

TABLE II. VALUES OF GIVEN VARIABLES OF RC POLYPHASE FILTER.

Variable	Value
b_{0P}	$(+j)^{12} R_1 R_2 R_3 R_4 C_1 C_2 C_3 C_4$
b_{1P}	$(+j)^9 [R_1 R_2 C_1 C_2 (R_3 C_3 + R_4 C_4) + R_3 R_4 C_3 C_4 (R_1 C_1 + R_2 C_2)]$
b_{2P}	$(+j)^6 [R_1 C_1 (R_2 C_2 + R_3 C_3 + R_4 C_4) + R_2 C_2 (R_3 C_3 + R_4 C_4) + R_3 C_3 R_4 C_4]$
b_{3P}	$(+j)^3 (R_1 C_1 + R_2 C_2 + R_3 C_3 + R_4 C_4)$
b_{0N}	$(-j)^{12} R_1 R_2 R_3 R_4 C_1 C_2 C_3 C_4$
b_{1N}	$(-j)^9 [R_1 R_2 C_1 C_2 (R_3 C_3 + R_4 C_4) + R_3 R_4 C_3 C_4 (R_1 C_1 + R_2 C_2)]$
b_{2N}	$(-j)^6 [R_1 C_1 (R_2 C_2 + R_3 C_3 + R_4 C_4) + R_2 C_2 (R_3 C_3 + R_4 C_4) + R_3 C_3 R_4 C_4]$
b_{3N}	$(-j)^3 (R_1 C_1 + R_2 C_2 + R_3 C_3 + R_4 C_4)$
a_0	$R_1 R_2 R_3 R_4 C_1 C_2 C_3 C_4$
a_1	$R_1 R_2 R_3 C_1 C_2 C_3 + R_1 R_2 R_4 (C_2 C_3 C_4 + 2C_1 C_2 C_3 + 2C_1 C_2 C_4 + 2C_1 C_3 C_4) + R_1 R_3 R_4 C_1 C_2 (2C_3 + C_4) + R_1 R_4 R_3 (C_1 C_3 C_4 + 2C_1 C_2 C_3 + 2C_1 C_2 C_4)$
a_2	$R_1 R_2 C_1 C_2 + R_3 R_4 (C_3 C_4 + 2C_1 C_4 + 2C_1 C_2 + 2C_1 C_3 + 2C_2 C_3) + R_1 R_3 (C_1 C_2 + 2C_1 C_3) + R_1 R_4 (C_1 C_2 + 2C_1 C_3 + 2C_2 C_3) + R_2 R_3 (C_2 C_3 + 2C_1 C_2 + 2C_1 C_3)$
a_3	$R_1 C_1 + R_2 (2C_1 + C_2) + R_3 (2C_1 + 2C_2 + C_3) + R_4 (2C_1 + 2C_2 + 2C_3 + C_4)$

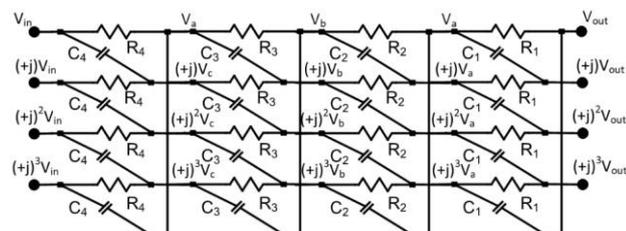


Fig. 13. Schematic of the 4th-order RC polyphase filter.

TABLE III. PARAMETERS OF RC POLYPHASE FILTER.

	Stage 1	Stage 2	Stage 3	Stage 4
Resistor	R₁	R₂	R₃	R₄
Value	1 kΩ	1 kΩ	1 kΩ	1 kΩ
Capacitor	C₁	C₂	C₃	C₄
Value	227 pF	106 pF	39.8 pF	19.9 pF
Cut-off frequency	f₁	f₂	f₃	f₄
Value	0.7 MHz	1.5 MHz	4 MHz	8 MHz

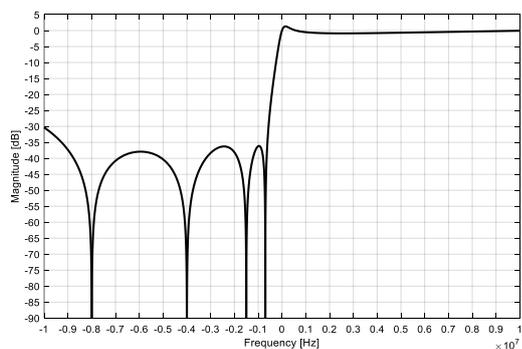


Fig. 14. Simulation result of the 4th-order RC polyphase filter.

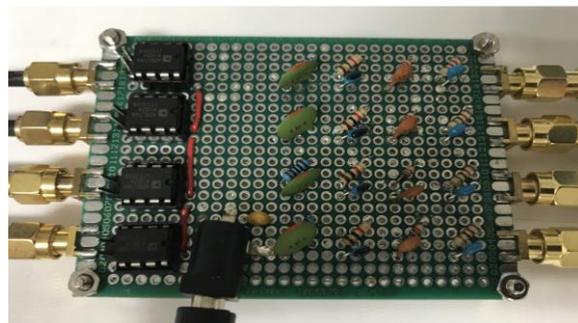


Fig. 15. Implemented circuit for the 4th-order RC polyphase filter.

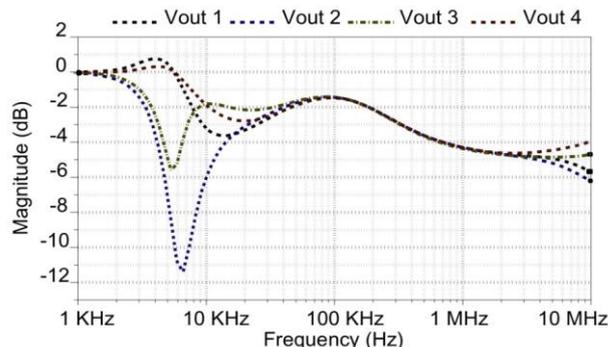


Fig. 16. Behaviors of pass-band gain of the 4th-order RC polyphase filter.

In practical design, the parasitic elements and the value variation of chosen components will affect the behavior of the circuit [33]. Fig. 16 shows the frequency responses of the pass-band gain of the implemented 4th-order RC polyphase filter. The IQ mismatches occurred at below 40 kHz and above 2 MHz in the range from 1 kHz to 10 MHz. The IQ mismatches were caused by the parasitic elements and the value variation of the R, C components. In addition, the limitations of the measurement equipment and the characteristics of the active unity-gain circuits also affected the measurement results [34].

C. Merits of proposed superposition formula

In this section, the merits of the proposed superposition formula are presented. There are some new concepts of multi-source signals such as negative and positive frequency, negative and positive complex signals, negative and positive polyphase signals, and spectrum of multi-source signals which are not well defined in the conventional methods [35]. In addition, there is not a general formula for analyzing the behaviors of multi-source networks. In a circuit having more than one independent source, the effects of all the sources can be considered at one time [36]. From the concept of the proposed formula, the transfer function of a multi-source network can be easily derived. The superposition formula was used to compare with the conventional superposition principle and the Millan's theorem as shown in TABLE IV. However, Millan's theorem is not widely applied in multi-source network analysis. Furthermore, the behaviours of feedback amplifiers with Miller capacitance due to parasitic capacitance between the output and input of active devices are simply analysed based on the proposed formula [37].



TABLE IV. COMPARISON OF PROPOSED METHOD WITH OTHER METHODS.

Features	Superposition formula	Conventional Superposition	Millan's theorem
Effects of all actuating sources	one time	Several times	one time
Transfer function accuracy	Yes	No	No
Single-input network analysis	Yes	Yes	Yes
Polyphase network analysis	Yes	No	No
Complex network analysis	Yes	No	No
Image rejection ratio accuracy	Yes	No	No

V. BEHAVIOURS OF RAUCH LOW-PASS FILTERS

A. Operating regions of high-order systems

In this section, the operating regions of high-order systems shall be reviewed. Ringing represents the voltage overshoot or current overshoot. It can be seen in many research papers, in datasheets of the commercial operational amplifiers, in the negative feedback amplifiers, and in the feedback control systems. Complex functions for transmission networks are usually expressed in three forms: magnitude-angular plots (Bode plots), polar charts (Nyquist charts), and magnitude-argument diagrams (Nichols diagrams) [38]. When the input source, output signal, and some other sources in the internal dynamics of a network are known, the transfer function is used to examine the behavior of the network. The transfer function $H(\omega)$ of a filter is the ratio of the output signal $V_{out}(\omega)$ to the input signal $V_{in}(\omega)$ as a function of the frequency. A simplified transfer function is rewritten as in (29). The term “self-loop function” is proposed to define $L(\omega)$ for linear networks in both cases with and without feedback systems.

$$H(\omega) = \frac{V_{out}(\omega)}{V_{in}(\omega)} = \frac{A(\omega)}{1 + L(\omega)} \tag{29}$$

where, $A(\omega)$ is the numerator function of the transfer function. Fig. The phase margin of the self-loop function is directly related to the transient response of a system. It shows how much the phase margin can be increased before the system becomes unstable.

In a 2nd-order system, the operating regions are over-damping, critical damping, and under-damping. In the over-damping region, the amplitudes of high-order harmonics of the step signal are significantly reduced from the first cut-off angular frequency; therefore, the rising time is short [39]. The transfer function and the self-loop function of a 2nd-order system are given in (30). The properties of a second-order system are summarized in TABLE V. Three typical cases of a second-order system are given to determine the operating region of the circuit as shown in Eq. (31). The coefficients of the transfer functions are

- Under-damped (binomial coefficients ratio is 1:1:1),
- Critically damped (binomial coefficients ratio is 1:2:1),
- Over-damped (binomial coefficients ratio is 1:3:1).

$$H(\omega) = \frac{b_0}{a_0(j\omega)^2 + a_1j\omega + 1}; L(\omega) = a_0(j\omega)^2 + a_1j\omega; \tag{30}$$

TABLE V. OPERATING REGIONS OF 2ND-ORDER SELF-LOOP FUNCTION

Case	Over-damping	
Delta (Δ)	$\Delta = a_1^2 - 4a_0 > 0$	
$ L(\omega) $	$\omega\sqrt{(a_0\omega)^2 + a_1^2}$	
$\theta(\omega)$	$\frac{\pi}{2} + \arctan \frac{a_0\omega}{a_1}$	
$\omega_1 = \frac{b}{2a}\sqrt{\sqrt{5}-2}$	$ L(\omega_1) > 1$	$\pi - \theta(\omega_1) > 76.3^\circ$
Case	Critical damping	
Delta (Δ)	$\Delta = a_1^2 - 4a_0 = 0$	
$ L(\omega) $	$\omega\sqrt{(a_0\omega)^2 + a_1^2}$	
$\theta(\omega)$	$\frac{\pi}{2} + \arctan \frac{a_0\omega}{a_1}$	
$\omega_1 = \frac{b}{2a}\sqrt{\sqrt{5}-2}$	$ L(\omega_1) = 1$	$\pi - \theta(\omega_1) = 76.3^\circ$
Case	Under-damping	
Delta (Δ)	$\Delta = a_1^2 - 4a_0 < 0$	
$ L(\omega) $	$\omega\sqrt{(a_0\omega)^2 + a_1^2}$	
$\theta(\omega)$	$\frac{\pi}{2} + \arctan \frac{a_0\omega}{a_1}$	
$\omega_1 = \frac{b}{2a}\sqrt{\sqrt{5}-2}$	$ L(\omega_1) < 1$	$\pi - \theta(\omega_1) < 76.3^\circ$

$$\begin{aligned}
 H_1(\omega) &= \frac{1}{(j\omega)^2 + j\omega + 1}; L_1(\omega) = (j\omega)^2 + j\omega; \\
 H_2(\omega) &= \frac{1}{(j\omega)^2 + 2j\omega + 1}; L_2(\omega) = (j\omega)^2 + 2j\omega; \\
 H_3(\omega) &= \frac{1}{(j\omega)^2 + 3j\omega + 1}; L_3(\omega) = (j\omega)^2 + 3j\omega;
 \end{aligned} \tag{31}$$

The Bode plot of these transfer functions, the Nichols plot of these self-loop functions, transient responses are shown in Fig. 17, Fig. 18, and Fig. 19, respectively. The phase margin is observed at unity gain of the self-loop function. The simulation results of the self-loop functions show that

- In the over-damping region, the phase margin is 82 degrees (observed at 98 degrees),
- In the critical damping region, the phase margin is 76.3 degrees (observed at 103.7 degrees), and
- In the under-damping region, the phase margin is 35 degrees (observed at 145 degrees).

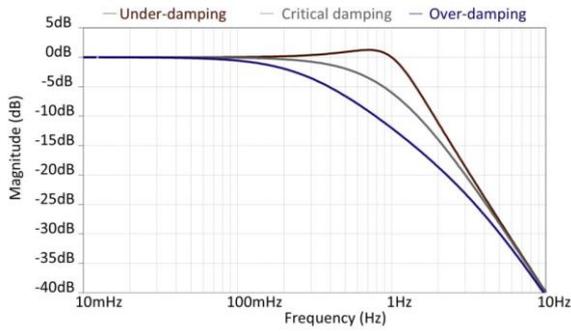


Fig. 17. Bode plot of the 2nd-order transfer function.

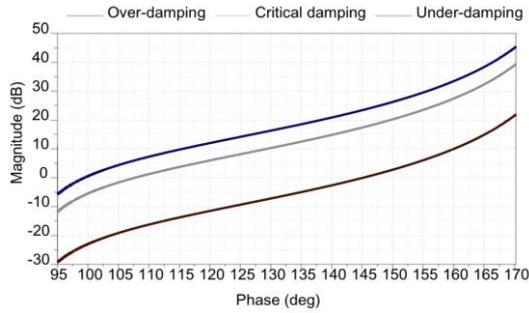


Fig. 18. Nichols chart of the 2nd-order self-loop function.

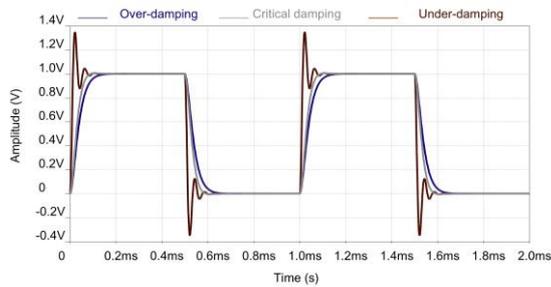


Fig. 19. Transient responses of the 2nd-order self-loop function.

The use of Pascal's triangle is a simple way to calculate the binomial coefficients of a high-order polynomial. In addition, the properties of the binomial coefficients can be also used to determine the operating regions of high-order systems [40]. In general, the coefficients of the expanded polynomial $(j\omega+1)^n$ are defined as shown in Eq. (32). For instance, based on the 4th row of Pascal's triangle as shown in Table VI, the binomial coefficients of the expanded polynomial are simplified as 1 : 4 : 6 : 4 : 1.

$$(j\omega + 1)^n = a_0(j\omega)^n + a_1(j\omega)^{n-1} + \dots + a_{n-1}j\omega + 1 \quad (32)$$

TABLE VI. PASCAL'S TABLE OF THE BINOMIAL COEFFICIENTS

	Binomial coefficients							
Row2		1	2	1				
Row3		1	3	3	1			
Row4		1	4	6	4	1		
Row5		1	5	10	10	5	1	
Row6		1	6	15	20	15	6	1

B. Behaviours of op amp without frequency compensation

In this section, the behaviors of a single-ended two-stage CMOS operational amplifier (op amp) without frequency compensation will be presented. Operational amplifiers are used in many applications such as unity-gain amplifiers, inverting amplifiers, and active analog filters [41]. The overall performance of these circuits may be stable or not. Thus, the stability test and the frequency compensation of this op amp are of interest [42]. Fig. 20 shows the schematic of a single-ended two-stage op amp without frequency compensation. A half small signal model of this op amp is drawn in Fig. 21. In the conventional Miller approximation, the interpretation of the Miller capacitance is very difficult because there are two loading capacitors in this two-stage op amp. After applying the proposed superposition theorem at each node, the current node equations at nodes V_a , V_b , and V_{out} are derived in (33).

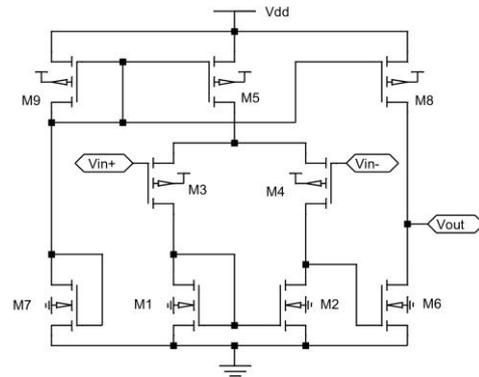


Fig. 20. Schematic of an op amp without frequency compensation

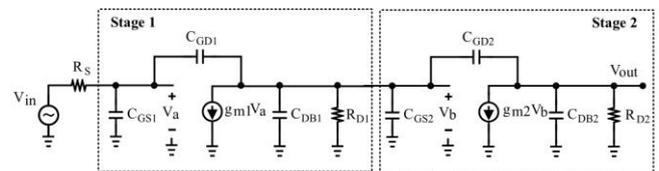


Fig. 21. Small signal model of the op amp without frequency compensation.

$$V_a \left[\frac{1}{R_s} + j\omega(C_{GS1} + C_{GD1}) \right] = \frac{V_m}{R_s} + V_b j\omega C_{GD1}; \quad (33)$$

$$V_b \left[j\omega(C_{GD1} + C_{DB1} + C_{GS2} + C_{GD2}) + \frac{1}{R_{D1}} \right] = V_a(j\omega C_{GD1} - g_{m1}) + V_{out} j\omega C_{GD2};$$

$$V_{out} \left[j\omega(C_{GD2} + C_{DB2}) + \frac{1}{R_{D2}} \right] = V_b(j\omega C_{GD2} - g_{m2});$$

Hence, the open-loop function $A(\omega)$ and the self-loop function $L(\omega)$ of this op amp are 3th-order complex functions as in (34). The values of the constant variables are given in TABLE VII. The gain-bandwidth (GBW) of this op amp is defined by the multiplication of the direct current gain A_0 and the cut-off frequency f_{BW} . The gain-bandwidth is also measured at unity gain ($GBW=A_0f_{BW}$). As the two-stage op amp is used to design a unity-gain amplifier as shown in Fig. 22, the overall performance of the circuit may be stable or not.

$$A(\omega) = \frac{V_{out}(\omega)}{V_{in}(\omega)} = \frac{b_0(j\omega)^2 + b_1j\omega + b_2}{a_0(j\omega)^3 + a_1(j\omega)^2 + a_2j\omega + 1}; \quad (34)$$

$$L(\omega) = a_0(j\omega)^3 + a_1(j\omega)^2 + a_2j\omega;$$

TABLE VII. VALUES OF GIVEN VARIABLES OF WITHOUT FREQUENCY COMPENSATION.

Variable	Value
b_0	$R_{D1}R_{D2}C_{GD1}C_{GD2}$
b_1	$-R_{D1}R_{D2}(C_{GD1}g_{m2} + C_{GD2}g_{m1})$
b_2	$R_{D1}R_{D2}g_{m1}g_{m2}$
a_0	$R_S R_{D1} R_{D2} (C_{GS1} + C_{GD1}) [C_{DB2} C_{GD2} + (C_{DB2} + C_{GD2})(C_{DB1} + C_{GS2})]$ $+ R_S R_{D1} R_{D2} C_{GS1} C_{GD1} (C_{DB2} + C_{GD2})$
a_1	$R_{D1} R_{D2} [C_{GD2} C_{DB2} + (C_{GD2} + C_{DB2})(C_{GD1} + C_{DB1} + C_{GS2})]$ $+ R_{D1} R_{D2} R_S [g_{m1} C_{GD1} (C_{GD2} + C_{DB2}) + g_{m2} C_{GD2} (C_{GS1} + C_{GD1})]$ $+ R_{D1} R_S [C_{GS1} C_{GD1} + (C_{GS1} + C_{GD1})(C_{DB1} + C_{GS2} + C_{GD2})]$ $+ R_{D2} R_S (C_{GS1} + C_{GD1})(C_{DB2} + C_{GD2})$
a_2	$R_{D1} (C_{GD1} + C_{DB1} + C_{GS2} + C_{GD2}) + R_{D2} [C_{DB2} + C_{GD2} (1 + R_{D1} g_{m2})]$ $+ R_S [C_{GS1} + C_{GD1} (1 + g_{m1} R_{D1})]$

After applying the superposition principle at node V_{out} , the output voltage is given in (35).

$$V_{out} = A(\omega)(V_{in} - V_{out}); \quad (35)$$

Hence, the transfer function $H(\omega)$ and the self-loop function $L(\omega)$ of the unity gain amplifier are derived in (36). The entire device parameters of the two-stage CMOS op amp are summarized in TABLE V. In this simulation, the transistor models of TSMC 0.18 μm CMOS technology are used to perform the ringing test for this amplifier. The supply voltage is 1.8 V, and the bias circuit is not shown.

$$H(\omega) = \frac{V_{out}}{V_{in}} = \frac{1}{1 + \frac{1}{A(\omega)}} \approx 1; L(\omega) = \frac{1}{A(\omega)}; \quad (36)$$

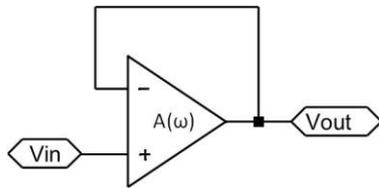


Fig. 22. Simplified model of the unity-gain amplifier circuit

TABLE VIII. PARAMETERS OF SINGLE-ENDED OP AMP.

Elements	Value (W/L)	Elements	Value (W/L)
M1	10 μm / 300 nm	M5	10 μm / 500 nm
M2	18 μm / 300 nm	M6	1.7 μm / 300 nm
M3	1.6 μm / 800 nm	M7	4 μm / 300 nm
M4	1.6 μm / 800 nm	M8, M9	1 μm / 300 nm

Fig. 23 and Fig. 24 show the Bode plot of the transfer function and the Nichols plot of the self-loop function of the unity-gain amplifier. The magnitude of the transfer function is very large (15 dB) at high frequency (2 GHz), and the phase margin at the unity gain of the self-loop function is very small (13 degrees). Therefore, overshoot and undershoot occur as shown in Fig. 25.

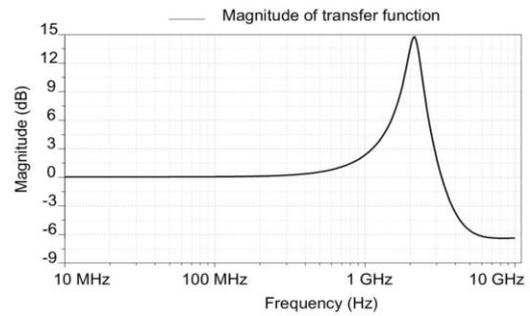


Fig. 23. Bode plot of the transfer function of the unity-gain amplifier.

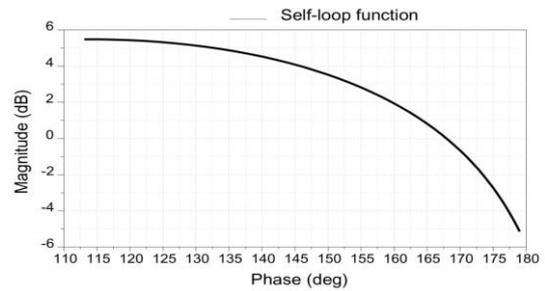


Fig. 24. Nichols plot of the self-loop function of the unity-gain amplifier.

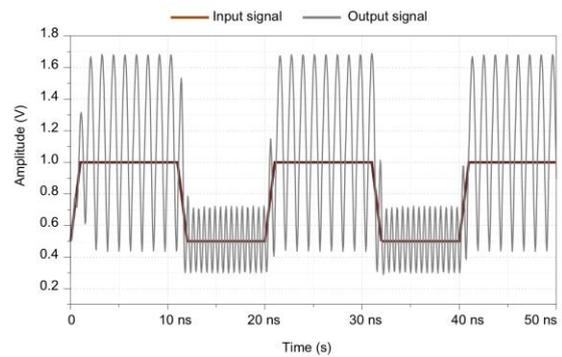


Fig. 25. Transient responses of the unity-gain amplifier.

C. Behaviours of a fully differential operational amplifier

In this section, the behaviors of a fully differential two-stage CMOS op amp with frequency compensation will be presented. Fig. 26 shows the schematic of a fully differential two-stage op amp with frequency compensation. A half small signal model of the fully differential two-stage op amp is drawn in Fig. 27. After applying the proposed superposition theorem at each node, the current node equations at nodes V_a , V_b , and V_{out} are derived in (37).

$$V_a \left[\frac{1}{R_S} + j\omega(C_{GS1} + C_{GD1}) \right] = \frac{V_{in}}{R_S} + V_b j\omega C_{GD1}; \quad (37)$$

$$V_b \left[j\omega(C_{GD1} + C_{DB1} + C_{GS2} + C_{GD2}) + \frac{1}{R_{D1}} + \frac{j\omega C_C}{1 + j\omega R_C C_C} \right]$$

$$= V_a (j\omega C_{GD1} - g_{m1}) + V_{out} \left(j\omega C_{GD2} + \frac{j\omega C_C}{1 + j\omega R_C C_C} \right);$$

$$V_{out} \left[j\omega(C_{GD2} + C_{DB2}) + \frac{j\omega C_C}{1 + j\omega R_C C_C} + \frac{1}{R_{D2}} \right] = V_b \left(j\omega C_{GD2} + \frac{j\omega C_C}{1 + j\omega R_C C_C} - g_{m2} \right);$$

Then, the open-loop function $A(\omega)$ and the self-loop function $L(\omega)$ of the fully differential two-stage op amp are 5th-order complex functions as in (38). The values of the constant variables are given in TABLE IX.

$$A(\omega) = \frac{V_{out}(\omega)}{V_{in}(\omega)} = \frac{b_0(j\omega)^4 + b_1(j\omega)^3 + b_2(j\omega)^2 + b_3j\omega + b_4}{a_0(j\omega)^5 + a_1(j\omega)^4 + a_2(j\omega)^3 + a_3(j\omega)^2 + a_4j\omega + 1}; \quad (38)$$

$$L(\omega) = a_0(j\omega)^5 + a_1(j\omega)^4 + a_2(j\omega)^3 + a_3(j\omega)^2 + a_4j\omega;$$

TABLE IX. VALUES OF GIVEN VARIABLES OF TWO-STAGE OP AMP WITH FREQUENCY COMPENSATION.

Variable	Value
b_0	$R_{D1}R_{D2}R_C^2C_{GD1}C_{GD2}C_C^2$
b_1	$R_C R_{D1}R_{D2}C_C [2C_{GD1}C_{GD2} + C_C C_{GD1} - R_C C_C (C_{GD1}g_{m2} + C_{GD2}g_{m1})]$
b_2	$R_{D1}R_{D2}C_{GD1}(C_{GD2} + C_C) + R_{D1}R_{D2}R_C C_C (R_C C_C g_{m1}g_{m2} - g_{m1}C_C - 2g_{m1}C_{GD2} - 2g_{m2}C_{GD1})$
b_3	$R_{D1}R_{D2} [2g_{m1}g_{m2}R_C C_C - g_{m1}(C_{GD2} + C_C) - g_{m2}C_{GD1}]$
b_4	$R_{D1}R_{D2}g_{m1}g_{m2}$
a_0	$R_S R_{D1}R_{D2}R_C^2C_C^2 [C_{DB2}C_{GD2}(C_{GS1} + C_{GD1}) + C_{GS1}C_{GD1}(C_{DB2} + C_{GD2})] + R_S R_{D1}R_{D2}R_C^2C_C^2 (C_{DB2} + C_{GD2})(C_{GS1} + C_{GD1})(C_{DB1} + C_{GS2})$
a_1	$R_{D1}R_{D2}R_C^2C_C^2 [C_{GD2}C_{DB2} + (C_{GD2} + C_{DB2})(C_{GD1} + C_{DB1} + C_{GS2})] + R_{D1}R_{D2}R_C^2C_C^2 [g_{m1}C_{GD1}(C_{GD2} + C_{DB2}) + g_{m2}C_{GD2}(C_{GS1} + C_{GD1})] + R_{D1}R_{D2}R_C^2C_C^2 [C_{GS1}C_{GD1} + (C_{GS1} + C_{GD1})(C_{DB1} + C_{GS2} + C_{GD2})] + R_{D2}R_S R_C^2C_C^2 (C_{GS1} + C_{GD1})(C_{DB2} + C_{GD2}) + R_{D1}R_{D2}R_S R_C C_C [2C_{DB2}C_{GD2}(C_{GS1} + C_{GD1}) + 2C_{GS1}C_{GD1}(C_{GD2} + C_{DB2})] + R_{D1}R_{D2}R_S R_C C_C (C_{GS1} + C_{GD1})(C_{DB2} + C_{GD2}) [2(C_{DB1} + C_{GS2}) + C_{GD2}] + R_{D1}R_{D2}R_S R_C C_C^2 [C_{DB2}(C_{GS1} + C_{GD1}) + C_{GS1}C_{GD1} + (C_{GS1} + C_{GD1})(C_{DB1} + C_{GS2})]$
a_2	$2R_{D1}R_S R_C C_C [C_{GS1}C_{GD1} + (C_{GS1} + C_{GD1})(C_{DB1} + C_{GS2} + C_{GD2})] + R_{D1}R_S R_C C_C [C_C (C_{GS1} + C_{GD1}) + R_{D2}g_{m2}C_{GD2}(C_{GS1} + C_{GD1})] + R_{D2}R_S R_C C_C (2C_{GD2} + 2C_{DB2} + C_C) [C_{GS1} + C_{GD1}(1 + g_{m1}R_{D1})] + 2R_{D1}R_{D2}R_C C_C [C_{DB2}C_{GD2} + (C_{DB2} + C_{GD2})(C_{GD1} + C_{DB1} + C_{GS2})] + R_{D1}R_{D2}R_C C_C^2 (C_{GD1} + C_{DB1} + C_{GS2} + C_{DB2} + C_{GD2}g_{m2}R_C) + R_{D1}R_{D2}R_S R_C C_C (C_{GD2} + C_{DB2} + C_C) + R_{D1}R_{D2}R_S C_{DB2} (C_{GS1} + C_{GD1})(C_{DB1} + C_{GS2} + C_{GD2} + C_C) + R_{D1}R_{D2}R_S (C_{GS1} + C_{GD1})(C_{DB1} + C_{GS2} + g_{m2}R_C C_C)(C_{GD2} + C_C) + R_C^2 C_C^2 [R_{D1}(C_{GD1} + C_{DB1} + C_{GS2} + C_{GD2}) + R_{D2}(C_{GD2} + C_{DB2})] + R_S R_C^2 C_C^2 [C_{GS1} + C_{GD1}(1 + g_{m1}R_{D1})]$
a_3	$2R_C C_C [R_{D1}(C_{GD1} + C_{DB1} + C_{GS2} + C_{GD2}) + R_S (C_{GS1} + C_{GD1})] + R_C C_C [C_C (R_C + R_{D1} + R_{D2}) + 2R_{D2}(C_{GD2} + C_{DB2})] + R_{D1}R_S [C_{GS1}C_{GD1} + (C_{GS1} + C_{GD1})(C_{DB1} + C_{GS2} + C_{GD2} + C_C)] + R_{D2}R_S (C_{GS1} + C_{GD1})(C_{GD2} + C_{DB2} + C_C) + R_{D1}R_{D2} [C_{DB2}(C_{GD2} + C_C) + (C_{GD1} + C_{DB1} + C_{GS2})(C_{GD2} + C_{DB2} + C_C)] + R_{D1}R_S g_{m1}C_{GD1} [2R_C C_C + R_{D2}(C_{GD2} + C_{DB2} + C_C)] + R_{D1}R_{D2}g_{m2} [R_S (C_{GD2} + C_C)(C_{GS1} + C_{GD1}) + R_C C_C (2C_{GD2} + C_C)]$
a_4	$R_{D1}(C_{GD1} + C_{DB1} + C_{GS2} + C_{GD2} + C_C) + R_{D2}(C_{GD2} + C_{DB2} + C_C) + 2R_C C_C + R_{D1}R_{D2}g_{m2}(C_{GD2} + C_C) + R_S (C_{GS1} + C_{GD1} + R_{D1}g_{m1}C_{GD1})$

In this simulation, the transistor models of TSMC 0.18 um CMOS technology are also used to design the fully differential op amp. The entire device parameters of this op amp are summarized in TABLE X. Fig. 28 shows the Bode plot of the open-loop function and the Nichols plot of the self-loop function in the proposed design of the two-stage op amp. The magnitude of the open-loop function is 55 dB at around cutoff frequency 1 MHz, and the phase margin at the unity gain of the self-loop function is 90 degrees.

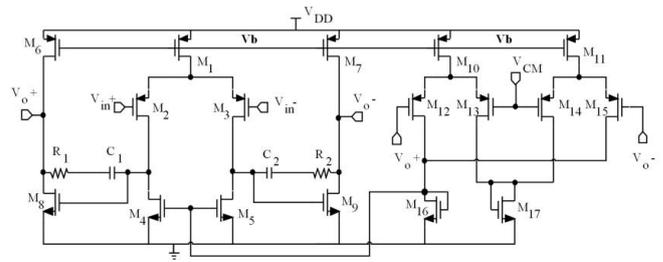


Fig. 26. Schematic of a fully differential two-stage op amp.

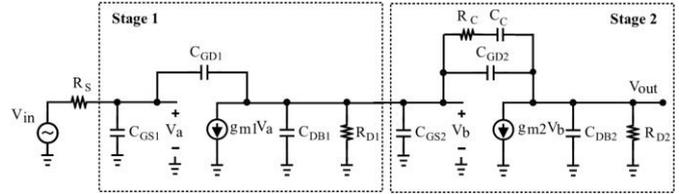


Fig. 27. Small signal model of the op amp with frequency compensation.

TABLE X. PARAMETERS OF OP AMP WITH FREQUENCY COMPENSATION.

Elements	Value	Elements	Value
M1	10 um / 300 nm	M11, M12	4 um / 300 nm
M2, M3	10 um / 180 nm	M13, M14	4 um / 300 nm
M4, M5	2 um / 300 nm	M15, M16	4 um / 300 nm
M6, M7	2 um / 180 nm	M17	4 um / 300 nm
M8, M9	10 um / 180 nm	M18	4 um / 300 nm
R1, R2	500 Ω	Vdd	1.8 V
C1, C2	500 fF	Vcm	0.9 V

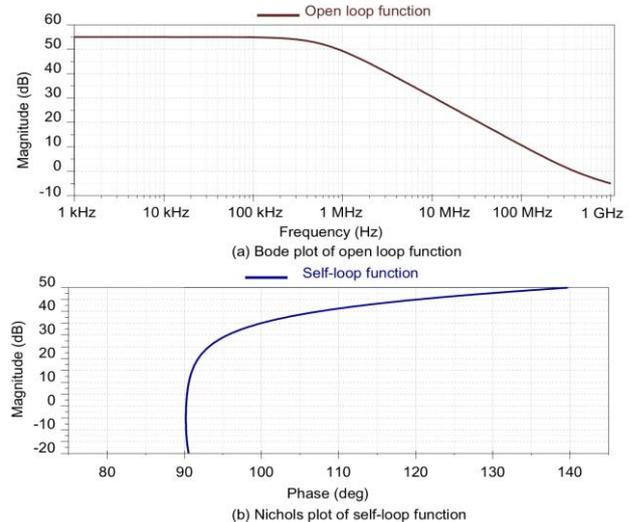


Fig. 28. Simulation results of the two-stage op amp.

D. Ringing test for a second-order Rauch LPF

In this section, the operating regions of a 2nd-order Rauch low-pass filter shall be analyzed. The single-ended form of this filter is shown in Fig. 29. After applying the proposed superposition theorem at each node, the current node equations at nodes V_A , V_B , and V_C are derived in (39).



$$V_A \left(\frac{1}{R_1} + \frac{1}{R_2} + j\omega C_1 + \frac{1}{R_3} \right) = \frac{V_{in}}{R_1} + \frac{V_B}{R_2} + \frac{V_{out}}{R_3}; \tag{39}$$

$$V_B \left(\frac{1}{R_2} + j\omega C_2 \right) = \frac{V_A}{R_2} + V_{out} j\omega C_2; V_{out} = -V_B A(\omega);$$

Then, the transfer function and the self-loop function of the low-pass filter are given in (40).

$$H(\omega) = -\frac{b_0}{a_0(j\omega)^2 + a_1j\omega + 1}; L(\omega) = a_0(j\omega)^2 + a_1j\omega; \tag{40}$$

$$b_0 = \frac{R_3}{R_1}; a_0 = R_2R_3C_1C_2; a_1 = \left(R_2 + R_3 + \frac{R_2R_3}{R_1} \right) C_2;$$

The operating regions of the Rauch low-pass filter can be determined through the passive components as in (41).

$$\frac{1}{R_2R_3C_1C_2} < \left(\frac{R_1R_2 + R_1R_3 + R_2R_3}{2R_1R_2R_3C_1} \right)^2; \text{Over-damping}$$

$$\frac{1}{R_2R_3C_1C_2} = \left(\frac{R_1R_2 + R_1R_3 + R_2R_3}{2R_1R_2R_3C_1} \right)^2; \text{Critical damping}$$

$$\frac{1}{R_2R_3C_1C_2} > \left(\frac{R_1R_2 + R_1R_3 + R_2R_3}{2R_1R_2R_3C_1} \right)^2; \text{Under-damping}$$

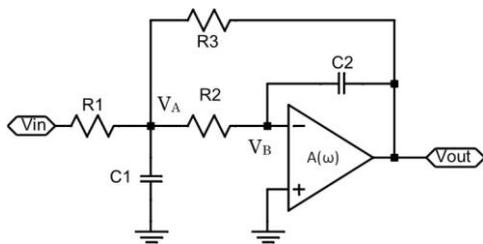


Fig. 29. Schematic diagram of a single-ended 2nd-order Rauch LPF.

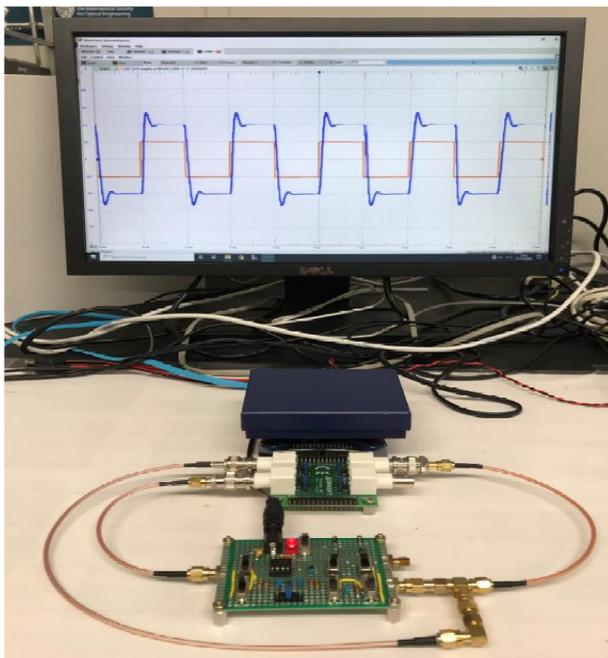


Fig. 30. Implemented circuit for the single-ended 2nd-order Rauch LPF.

A commercial op amp LM358 was used in this implementation. Fig. 30 shows the measurement set up of the implemented 2nd-order Rauch low-pass filter. The Bode plot of the transfer function, the Nichols plot of the self-loop function, and the transient response of the implemented circuit are shown in Figs. 31, 32, and 33, respectively. The measurement results of the self-loop functions in the implemented circuit show that

- In the over-damping region, the phase margin is 87 degrees (observed at 93 degrees),
- In the critical damping region, the phase margin is 80 degrees (observed at 100 degrees), and
- In the under-damping region, the phase margin is 50 degrees (observed at 130 degrees); therefore, the ringing occurs in the transient response of the under-damping region.

To reduce the white noise and random noise, the differential forms are used in many designs of high-order amplifiers and filters. Therefore, the differential forms of high-order systems have been studied recently [43].

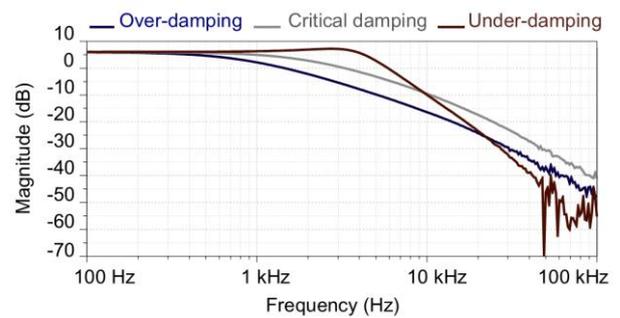


Fig. 31. Bode plot of the transfer function in the implemented circuit.

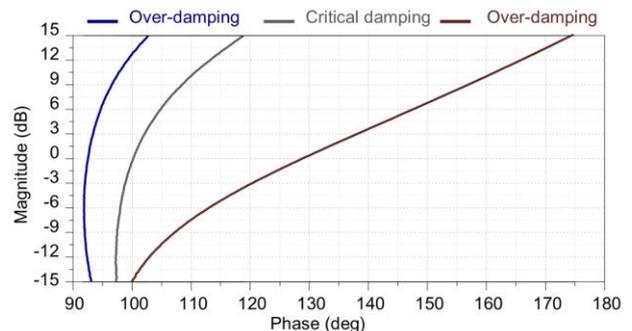


Fig. 32. Nichol plot of the self-loop function in the implemented circuit.

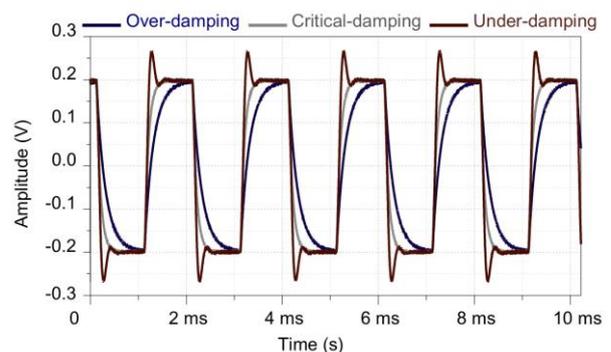


Fig. 33. Transient responses of the implemented circuit.



E. Ringing test for a fourth-order Rauch LPF

In this section, the operating regions of a fully differential 4th-order Rauch low-pass filter shall be analyzed. A differential form of the 4th-order Rauch low-pass filter is shown in Fig. 34. After applying the superposition principle at each node, the current node equations at nodes V_a, V_b, V_c, V_d, and V_e are derived in (42).

$$\begin{aligned}
 V_a \left(\frac{1}{R_1} + j\omega C_1 + \frac{1}{R_2} + \frac{1}{R_3} \right) &= \frac{V_{in}}{R_1} + \frac{V_b}{R_2} + \frac{(+j)^2 V_c}{R_3} + (+j)^2 V_a j\omega C_1; \\
 V_b \left(\frac{1}{R_2} + j\omega C_2 \right) &= \frac{V_a}{R_2} + (+j)^2 V_c j\omega C_2; V_c = (V_b - (+j)^2 V_b) A(\omega); \\
 V_d \left(\frac{1}{R_4} + j\omega C_3 + \frac{1}{R_5} + \frac{1}{R_6} \right) &= \frac{V_c}{R_4} + \frac{V_e}{R_5} + \frac{(+j)^2 V_{out}}{R_6} + (+j)^2 V_d j\omega C_3; \\
 V_e \left(\frac{1}{R_5} + j\omega C_4 \right) &= \frac{V_d}{R_5} + (+j)^2 V_{out} j\omega C_4; V_{out} = (V_e - (+j)^2 V_e) A(\omega);
 \end{aligned}
 \tag{42}$$

Then, the open-loop function and the self-loop function of this op amp are given in (43). The values of the constant variables of the 4th-order fully differential Rauch low-pass filter are given in TABLE XI.

$$\begin{aligned}
 H(\omega) &= \frac{b_0}{1 + a_0(j\omega)^4 + a_1(j\omega)^3 + a_2(j\omega)^2 + a_3j\omega}; \\
 L(\omega) &= a_0(j\omega)^4 + a_1(j\omega)^3 + a_2(j\omega)^2 + a_3j\omega;
 \end{aligned}
 \tag{43}$$

TABLE XI. VALUES OF GIVEN VARIABLES OF 4TH-ORDER RAUCH LPF.

Variable	Value
b₀	$\frac{R_5 R_6}{R_1 R_4}$
a₀	$R_2 R_3 R_5 R_6 C_1 C_2 C_3 C_4$
a₁	$R_2 R_3 \left(R_5 + R_6 + \frac{R_5 R_6}{R_4} \right) C_1 C_2 C_4 + R_5 R_6 \left(R_2 + R_3 + \frac{R_2 R_3}{R_1} \right) C_2 C_3 C_4$
a₂	$R_2 R_3 C_1 C_2 + R_5 R_6 C_3 C_4 + \left(R_2 + R_3 + \frac{R_2 R_3}{R_1} \right) \left(R_5 + R_6 + \frac{R_5 R_6}{R_4} \right) C_2 C_4$
a₃	$\left(R_2 + R_3 + \frac{R_2 R_3}{R_1} \right) C_2 + \left(R_5 + R_6 + \frac{R_5 R_6}{R_4} \right) C_4$

The behaviors of the open loop function are shown in the previous section. The fully differential Rauch low-pass filter is designed for cut-off frequency $f_0 = 100$ kHz. Various values of capacitor C₁ were used to change the operating region of this filter. The entire device parameters of this filter are summarized in TABLE XII. To derive the self-loop function in high-order systems, the proposed comparison measurement and the alternating current conservation methods were introduced [44]. The Bode plot of the transfer function, the Nichols plot of the self-loop function, and transient response of the fully differential 4th-order Rauch low-pass filter are shown in Fig. 35. The simulation results of the self-loop function show that

- In the over-damping region, the phase margin is 74 degrees (observed at 106 degrees),
- In the critical damping region, the phase margin is 68 degrees (observed at 112 degrees), and
- In the under-damping region, the phase margin is 59 degrees (observed at 121 degrees); and the ringing occurs in the transient response of the under-damping region.

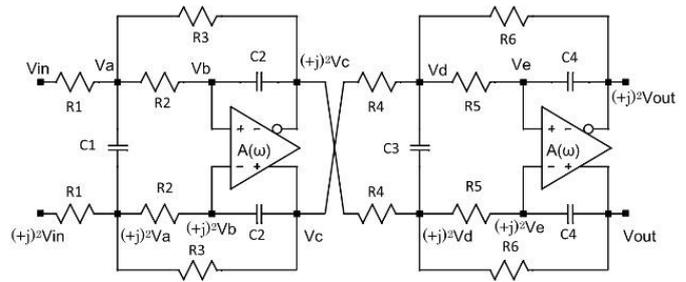
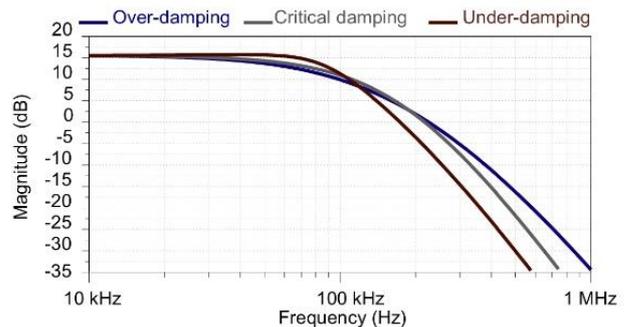


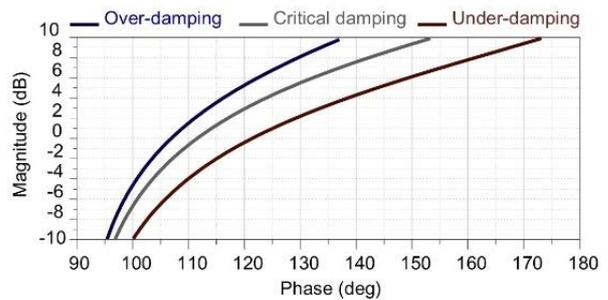
Fig. 34. Schematic diagram of a fully differential 4th-order Rauch LPF.

TABLE XII. DEVICE PARAMETERS OF 4TH-ORDER RAUCH LPF.

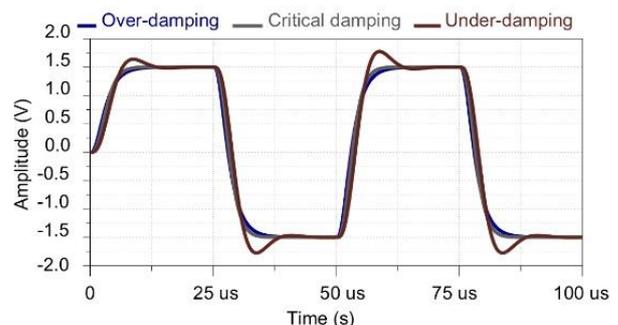
Element	Value	Element	Value	Element	Value
R₁	1 kΩ	R₃	3 kΩ	R₅	1 kΩ
R₂	1 kΩ	R₄	2 kΩ	R₆	4 kΩ
C₁	0.4 nF	C₂	0.2 nF	C₄	0.3 nF
Over-damping		Critical damping		Under-damping	
Element	Value	Element	Value	Element	Value
C₃	0.1 nF	C₃	0.5 nF	C₃	1.4 nF



(a) Bode plot of transfer function



(b) Nichols plot of self-loop function



(c) Transient response of output signal

Fig. 35. Simulation results of the 4th-order Rauch low-pass filter.

It is emphasized that the Nyquist criterion for adaptive feedback systems can be only used to verify if the transfer function of a closed-loop system and the loop gain are well defined. Numerous research papers have demonstrated that the design procedure used in feedback control theory does indeed lead to a stable design and even to robust stability in the case of uncertain plants. However, a rigorous proof for the stability test based on Nichols chart of loop gain has not been formulated. The Nichols plot of a loop gain is very complicated. In this work, the Nichols plot of the self-loop function is easier than the conventional one as shown in Fig. 35(b). The phase margin of the self-loop function indicates the operating region of a linear network. In addition, the operating region of a general transfer function can be theoretically defined based on Pascal's triangle [45]. In the next section, the behaviors of a 4th-order Rauch complex filter will be analyzed.

VI. MODELS OF HIGH-ORDER COMPLEX FILTERS

A. Design principle for active complex filter networks

The design principle for active complex filters is reviewed in this section. The transfer function and the image rejection ratio in the complex filters are not well defined in details. In recent years, the demand for fully integrated radio frequency transceivers in wireless communication gives rise to great attention to single chip low power transceivers [46]. Therefore, polyphase filters and complex filters are widely used in these transceivers. A complex filter is designed based on the frequency shifting of a real active low-pass filter as shown in Fig. 36. The transfer function of this complex filter is defined in (44). Here, ω_{cr} and ω_{cut} are cross and cut-off angular frequency [47].

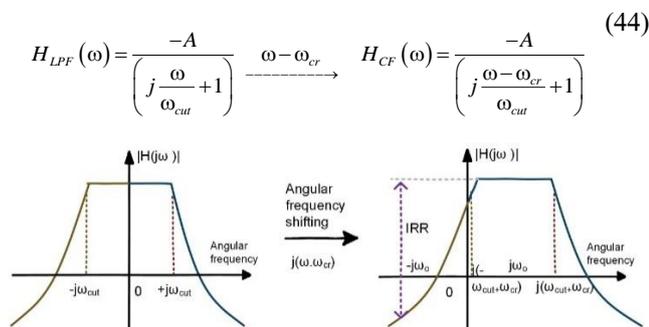


Fig. 36. Frequency shifting principle from a LPF into a complex filter.

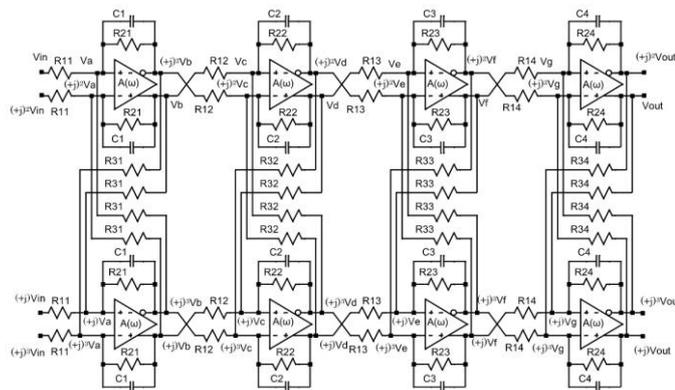


Fig. 37. Schematic of the 4th-order complex filter.

TABLE XIII. VALUES OF GIVEN VARIABLES OF RAUCH COMPLEX FILTER.

Variable	Value
b_0	$\frac{R_{21} R_{22} R_{23} R_{24}}{R_{11} R_{12} R_{13} R_{14}}$
a_0	$R_{21} R_{22} R_{23} R_{24} C_1 C_2 C_3 C_4$
a_{1P}	$R_{21} R_{22} C_1 C_2 (R_{23} C_3 + R_{24} C_4) + R_{22} R_{23} C_1 C_2 (R_{21} C_1 + R_{22} C_2) + j \left[R_{21} R_{22} C_1 C_2 \left(\frac{R_{23}}{R_{34}} R_{23} C_3 + \frac{R_{24}}{R_{33}} R_{24} C_4 \right) + R_{22} R_{23} C_1 C_2 \left(\frac{R_{21}}{R_{12}} R_{21} C_1 + \frac{R_{22}}{R_{11}} R_{22} C_2 \right) \right]$
a_{2P}	$R_{21} C_1 (R_{22} C_2 + R_{23} C_3 + R_{24} C_4) + R_{22} C_2 (R_{23} C_3 + R_{24} C_4) + R_{23} R_{24} C_1 C_2 + j R_{21} C_1 \left[R_{22} C_2 \left(\frac{R_{23}}{R_{33}} + \frac{R_{24}}{R_{34}} \right) + R_{23} C_3 \left(\frac{R_{22}}{R_{32}} + \frac{R_{24}}{R_{34}} \right) + R_{24} C_4 \left(\frac{R_{22}}{R_{32}} + \frac{R_{23}}{R_{33}} \right) \right] + j R_{22} C_2 \left[R_{23} C_3 \left(\frac{R_{21}}{R_{31}} + \frac{R_{24}}{R_{34}} \right) + R_{24} C_4 \left(\frac{R_{21}}{R_{31}} + \frac{R_{23}}{R_{33}} \right) \right] + (+j)^2 R_{21} C_1 \left[R_{22} C_2 \left(\frac{R_{23}}{R_{33}} R_{24} + R_{23} C_3 \frac{R_{24}}{R_{32} R_{34}} + R_{24} C_4 \frac{R_{23}}{R_{32} R_{33}} \right) \right] + (+j)^2 \left[R_{22} C_2 \left(R_{23} C_3 \frac{R_{24}}{R_{31} R_{34}} + R_{23} C_4 \frac{R_{24}}{R_{33} R_{31}} \right) + R_{23} R_{24} C_1 C_2 \frac{R_{21}}{R_{31} R_{32}} \right]$
a_{3P}	$(+j)^3 \left[R_{21} C_1 \frac{R_{22} R_{23} R_{24}}{R_{33} R_{34}} + R_{22} C_2 \frac{R_{21} R_{23} R_{24}}{R_{31} R_{33} R_{34}} + R_{23} C_3 \frac{R_{21} R_{22} R_{24}}{R_{34} R_{31} R_{32}} + R_{24} C_4 \frac{R_{21} R_{22} R_{23}}{R_{33} R_{31} R_{32}} \right] + R_{21} C_1 + R_{22} C_2 + (+j)^2 \left[R_{21} C_1 \left(\frac{R_{23}}{R_{33}} R_{24} + \frac{R_{24}}{R_{34}} R_{23} \right) + R_{22} C_2 \left(\frac{R_{21}}{R_{31}} R_{24} + \frac{R_{21}}{R_{31}} R_{23} \right) \right] + R_{23} C_3 + R_{24} C_4 + (+j) \left[R_{23} C_3 \left(\frac{R_{21}}{R_{31}} R_{24} + \frac{R_{24}}{R_{34}} R_{23} \right) + R_{24} C_4 \left(\frac{R_{21}}{R_{31}} R_{24} + \frac{R_{21}}{R_{31}} R_{23} \right) \right] + j \left[R_{21} C_1 \left(\frac{R_{23}}{R_{33}} + \frac{R_{24}}{R_{34}} + \frac{R_{22}}{R_{32}} \right) + R_{22} C_2 \left(\frac{R_{23}}{R_{33}} + \frac{R_{24}}{R_{34}} + \frac{R_{21}}{R_{31}} \right) + R_{23} C_3 \left(\frac{R_{21}}{R_{31}} + \frac{R_{22}}{R_{32}} + \frac{R_{24}}{R_{34}} \right) + R_{24} C_4 \left(\frac{R_{21}}{R_{31}} + \frac{R_{22}}{R_{32}} + \frac{R_{23}}{R_{33}} \right) \right]$
a_{4P}	$(+j)^4 \frac{R_{21} R_{22} R_{23} R_{24}}{R_{31} R_{32} R_{33} R_{34}} + (+j)^2 \left[\frac{R_{21}}{R_{31}} \left(\frac{R_{22}}{R_{32}} + \frac{R_{23}}{R_{33}} + \frac{R_{24}}{R_{34}} \right) + \frac{R_{22}}{R_{32}} \left(\frac{R_{23}}{R_{33}} + \frac{R_{24}}{R_{34}} \right) + \frac{R_{23}}{R_{33}} R_{24} \right] + (+j)^3 \left[\frac{R_{21} R_{22}}{R_{31} R_{32}} \left(\frac{R_{23}}{R_{33}} + \frac{R_{24}}{R_{34}} \right) + \frac{R_{23} R_{24}}{R_{33} R_{34}} \left(\frac{R_{21}}{R_{31}} + \frac{R_{22}}{R_{32}} \right) \right] + j \left[\frac{R_{21}}{R_{31}} + \frac{R_{22}}{R_{32}} + \frac{R_{23}}{R_{33}} + \frac{R_{24}}{R_{34}} \right]$
a_{1N}	$R_{21} R_{22} C_1 C_2 (R_{23} C_3 + R_{24} C_4) + R_{22} R_{23} C_1 C_2 (R_{21} C_1 + R_{22} C_2) - j \left[R_{21} R_{22} C_1 C_2 \left(\frac{R_{23}}{R_{34}} R_{23} C_3 + \frac{R_{24}}{R_{33}} R_{24} C_4 \right) + R_{22} R_{23} C_1 C_2 \left(\frac{R_{21}}{R_{12}} R_{21} C_1 + \frac{R_{22}}{R_{11}} R_{22} C_2 \right) \right]$
a_{2N}	$R_{21} C_1 (R_{22} C_2 + R_{23} C_3 + R_{24} C_4) + R_{22} C_2 (R_{23} C_3 + R_{24} C_4) + R_{23} R_{24} C_1 C_2 - j R_{21} C_1 \left[R_{22} C_2 \left(\frac{R_{23}}{R_{33}} + \frac{R_{24}}{R_{34}} \right) + R_{23} C_3 \left(\frac{R_{22}}{R_{32}} + \frac{R_{24}}{R_{34}} \right) + R_{24} C_4 \left(\frac{R_{22}}{R_{32}} + \frac{R_{23}}{R_{33}} \right) \right] - j R_{22} C_2 \left[R_{23} C_3 \left(\frac{R_{21}}{R_{31}} + \frac{R_{24}}{R_{34}} \right) + R_{24} C_4 \left(\frac{R_{21}}{R_{31}} + \frac{R_{23}}{R_{33}} \right) \right] + j R_{23} R_{24} C_1 C_2 \left(\frac{R_{21}}{R_{31}} + \frac{R_{22}}{R_{32}} \right) + (-j)^2 R_{21} C_1 \left[R_{22} C_2 \frac{R_{23}}{R_{33}} R_{24} + R_{23} C_3 \frac{R_{24}}{R_{32} R_{34}} + R_{24} C_4 \frac{R_{23}}{R_{32} R_{33}} \right] + (-j)^2 \left[R_{22} C_2 \left(R_{23} C_3 \frac{R_{24}}{R_{31} R_{34}} + R_{24} C_4 \frac{R_{23}}{R_{33} R_{31}} \right) + R_{23} R_{24} C_1 C_2 \frac{R_{21}}{R_{31} R_{32}} \right]$
a_{3N}	$(-j)^3 \left[R_{21} C_1 \frac{R_{22} R_{23} R_{24}}{R_{33} R_{34}} + R_{22} C_2 \frac{R_{21} R_{23} R_{24}}{R_{31} R_{33} R_{34}} + R_{23} C_3 \frac{R_{21} R_{22} R_{24}}{R_{34} R_{31} R_{32}} + R_{24} C_4 \frac{R_{21} R_{22} R_{23}}{R_{33} R_{31} R_{32}} \right] + R_{21} C_1 + R_{22} C_2 + (-j)^2 \left[R_{21} C_1 \left(\frac{R_{23}}{R_{33}} R_{24} + \frac{R_{24}}{R_{34}} R_{23} \right) + R_{22} C_2 \left(\frac{R_{21}}{R_{31}} R_{24} + \frac{R_{21}}{R_{31}} R_{23} \right) \right] + R_{23} C_3 + R_{24} C_4 + (-j) \left[R_{23} C_3 \left(\frac{R_{21}}{R_{31}} R_{24} + \frac{R_{24}}{R_{34}} R_{23} \right) + R_{24} C_4 \left(\frac{R_{21}}{R_{31}} R_{24} + \frac{R_{21}}{R_{31}} R_{23} \right) \right] - j \left[R_{21} C_1 \left(\frac{R_{23}}{R_{33}} + \frac{R_{24}}{R_{34}} + \frac{R_{22}}{R_{32}} \right) + R_{22} C_2 \left(\frac{R_{23}}{R_{33}} + \frac{R_{24}}{R_{34}} + \frac{R_{21}}{R_{31}} \right) + R_{23} C_3 \left(\frac{R_{21}}{R_{31}} + \frac{R_{22}}{R_{32}} + \frac{R_{24}}{R_{34}} \right) + R_{24} C_4 \left(\frac{R_{21}}{R_{31}} + \frac{R_{22}}{R_{32}} + \frac{R_{23}}{R_{33}} \right) \right]$
a_{4N}	$(-j)^4 \frac{R_{21} R_{22} R_{23} R_{24}}{R_{31} R_{32} R_{33} R_{34}} + (-j)^2 \left[\frac{R_{21}}{R_{31}} \left(\frac{R_{22}}{R_{32}} + \frac{R_{23}}{R_{33}} + \frac{R_{24}}{R_{34}} \right) + \frac{R_{22}}{R_{32}} \left(\frac{R_{23}}{R_{33}} + \frac{R_{24}}{R_{34}} \right) + \frac{R_{23}}{R_{33}} R_{24} \right] + (-j)^3 \left[\frac{R_{21} R_{22}}{R_{31} R_{32}} \left(\frac{R_{23}}{R_{33}} + \frac{R_{24}}{R_{34}} \right) + \frac{R_{23} R_{24}}{R_{33} R_{34}} \left(\frac{R_{21}}{R_{31}} + \frac{R_{22}}{R_{32}} \right) \right] - j \left[\frac{R_{21}}{R_{31}} + \frac{R_{22}}{R_{32}} + \frac{R_{23}}{R_{33}} + \frac{R_{24}}{R_{34}} \right]$

B. Behaviour of the conventional 4th-order complex filter

In this section, the behaviour of a 4th-order complex filter is reviewed. Fig. 37 shows the schematic diagram of the conventional 4th-order complex filter [48]. As the input signal sources are the positive polyphase signals $S_p\{V_1; V_2; V_3; V_4\} = \{1; (+j); (+j)^2; (+j)^3\} V(t)$, after the proposed superposition theorem is applied at each node, the current node equations at nodes $V_a, V_b, V_c, V_d, V_e, V_f, V_g$, and V_{out} are derived in (45).



$$\begin{aligned}
 V_a \left(\frac{1}{R_{11}} + \frac{1}{Z_{C1}} + \frac{1}{R_{21}} + \frac{1}{R_{31}} \right) &= \frac{V_{in}}{R_{11}} + \frac{(+j)^3 V_b}{R_{31}} + \left(\frac{1}{Z_{C1}} + \frac{1}{R_{21}} \right) (+j)^2 V_b; \\
 V_c \left(\frac{1}{R_{12}} + \frac{1}{Z_{C2}} + \frac{1}{R_{22}} + \frac{1}{R_{32}} \right) &= \frac{V_b}{R_{12}} + \frac{(+j)^3 V_d}{R_{31}} + \left(\frac{1}{Z_{C2}} + \frac{1}{R_{22}} \right) (+j)^2 V_d; \\
 V_b &= [V_a - (+j)^2 V_a] A(\omega); V_d = [V_c - (+j)^2 V_c] A(\omega); \\
 V_e \left(\frac{1}{R_{13}} + \frac{1}{Z_{C3}} + \frac{1}{R_{23}} + \frac{1}{R_{33}} \right) &= \frac{V_d}{R_{13}} + \frac{(+j)^3 V_f}{R_{33}} + \left(\frac{1}{Z_{C3}} + \frac{1}{R_{23}} \right) (+j)^2 V_f; \\
 V_g \left(\frac{1}{R_{14}} + \frac{1}{Z_{C4}} + \frac{1}{R_{24}} + \frac{1}{R_{34}} \right) &= \frac{V_{in}}{R_{14}} + \frac{(+j)^3 V_{out}}{R_{33}} + \left(\frac{1}{Z_{C4}} + \frac{1}{R_{24}} \right) (+j)^2 V_{out}; \\
 V_f &= [V_e - (+j)^2 V_e] A(\omega); V_{out} = [V_g - (+j)^2 V_g] A(\omega);
 \end{aligned} \tag{45}$$

Then, the transfer function of this complex filter in the positive frequency domain is given in (46).

$$H_P(\omega) = \frac{b_0}{a_0(j\omega)^4 + a_{1P}(j\omega)^3 + a_{2P}(j\omega)^2 + a_{3P}j\omega + a_{4P} + 1}; \tag{46}$$

As the input sources are negative polyphase signals $S_n\{V_1; V_2; V_3; V_4\} = \{1; (-j); (-j)^2; (-j)^3\} V(t)$, the transfer function of the 4th-order complex filter in the negative frequency domain is given in (47).

$$H_N(\omega) = \frac{b_0}{a_0(j\omega)^4 + a_{1N}(j\omega)^3 + a_{2N}(j\omega)^2 + a_{3N}j\omega + a_{4N} + 1}; \tag{47}$$

Hence, the image rejection ratio of this complex filter is defined in (48). The values of the constant variables of the 4th-order complex filter are given in TABLE XIV.

$$IRR(\omega) = \frac{a_0(j\omega)^4 + a_{1N}(j\omega)^3 + a_{2N}(j\omega)^2 + a_{3N}j\omega + a_{4N} + 1}{a_0(j\omega)^4 + a_{1P}(j\omega)^3 + a_{2P}(j\omega)^2 + a_{3P}j\omega + a_{4P} + 1}; \tag{48}$$

TABLE XIV. DEVICE PARAMETERS OF COMPLEX FILTER.

Element	Value	Element	Value
R₁₁	2 kΩ	R₁₂	1 kΩ
R₂₁	7 kΩ	R₂₂	3.5 kΩ
R₃₁	2 kΩ	R₃₂	1 kΩ
C₁	86 pF	C₂	52 pF
R₁₃	2 kΩ	R₁₄	1 kΩ
R₂₃	7 kΩ	R₂₄	3.5 kΩ
R₃₃	2 kΩ	R₃₄	1 kΩ
C₃	86 pF	C₄	52 pF

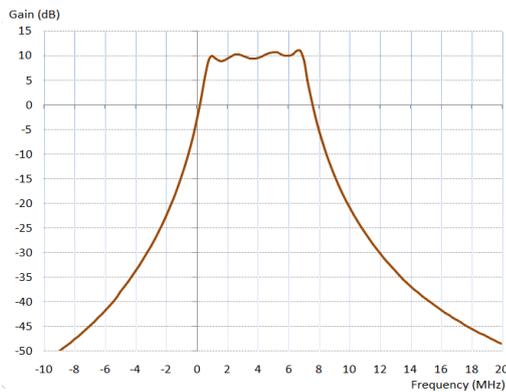


Fig. 38. Bode plot of the 4th-order complex filter.

Fig. 38 shows the behaviours of the 4th-order complex filter. In the positive frequency domain, the pass-band gain is 10 dB and the image signal rejection ratio is 33 dB.

C. Design of a 4th-order Rauch complex filter

In this section, a proposed design of a 4th-order Rauch complex filter is presented. The advantages of the Rauch architecture are an easier selection of circuit components and a simpler design in complex filters [49]. Fig. 39 shows the schematic diagram of a 4th-order Rauch complex filter. As the input signal sources are the positive polyphase signals $S_p\{V_1; V_2; V_3; V_4\}$, after the proposed superposition theorem is applied at each node, the current node equations at nodes V_a, V_b, V_c, V_d, V_e , and V_{out} are derived in (49).

$$\begin{aligned}
 V_a \left(\frac{1}{R_1} + j\omega C_1 + \frac{1}{R_2} + \frac{1}{R_3} \right) &= \frac{V_{in}}{R_1} + (+j)^2 V_a j\omega C_1 + \frac{V_b}{R_2} + \frac{(+j)^2 V_c}{R_3}; \\
 V_b \left(\frac{1}{R_2} + j\omega C_2 + \frac{1}{R_4} \right) &= \frac{V_a}{R_2} + (+j)^2 V_c j\omega C_2 + \frac{(+j)^3 V_c}{R_4}; \\
 V_c &= [V_b - (+j)^2 V_b] A(\omega); V_{out} = [V_e - (+j)^2 V_e] A(\omega); \\
 V_d \left(\frac{1}{R_5} + j\omega C_3 + \frac{1}{R_6} + \frac{1}{R_7} \right) &= \frac{V_c}{R_5} + (+j)^2 V_d j\omega C_3 + \frac{V_e}{R_6} + \frac{(+j)^2 V_{out}}{R_7}; \\
 V_e \left(\frac{1}{R_6} + j\omega C_4 + \frac{1}{R_8} \right) &= \frac{V_d}{R_6} + (+j)^2 V_{out} j\omega C_4 + \frac{(+j)^3 V_{out}}{R_8};
 \end{aligned} \tag{49}$$

Then, the transfer function of this complex filter in the positive frequency domain is given in (50).

$$H_P(\omega) = \frac{b_0}{a_0(j\omega)^4 + a_{p1}(j\omega)^3 + a_{p2}(j\omega)^2 + a_{p3}j\omega + a_{p4} + 1}; \tag{50}$$

As the input sources are negative polyphase signals $S_n\{V_1; V_2; V_3; V_4\}$, the transfer function of this complex filter in the negative frequency domain is given in (51).

$$H_N(\omega) = \frac{b_0}{a_0(j\omega)^4 + a_{N1}(j\omega)^3 + a_{N2}(j\omega)^2 + a_{N3}j\omega + a_{N4} + 1}; \tag{51}$$

Hence, the image rejection ratio of this complex filter is defined in (52). The values of the constant variables of the 4th-order Rauch complex filter are given in TABLE XV.

$$IRR(\omega) = \frac{a_0(j\omega)^4 + a_{N1}(j\omega)^3 + a_{N2}(j\omega)^2 + a_{N3}j\omega + a_{N4} + 1}{a_0(j\omega)^4 + a_{p1}(j\omega)^3 + a_{p2}(j\omega)^2 + a_{p3}j\omega + a_{p4} + 1}; \tag{52}$$

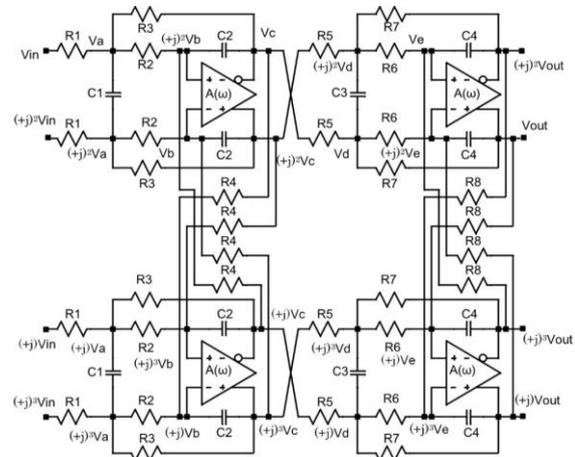


Fig. 39. Schematic of a 4th-order Rauch complex filter.



TABLE XV. VALUES OF GIVEN VARIABLES OF RAUCH COMPLEX FILTER.

Variable	Value
b_0	$\frac{R_3 R_7}{R_1 R_5}$
a_0	$4R_2 R_3 R_6 R_7 C_2 C_3 C_4$
a_{P1}	$2R_2 R_3 C_2 C_4 \left(R_6 + R_7 + \frac{R_6 R_7}{R_5} \right) + 2R_6 R_3 C_2 C_4 \left(R_2 + R_3 + \frac{R_2 R_3}{R_1} \right) + j 2R_2 R_3 R_6 R_7 C_2 C_4 \left(\frac{C_2 + C_4}{R_6} + \frac{C_2}{R_3} \right)$
a_{P2}	$\frac{(+j)^2 R_2 R_3 R_6 R_7 C_2 C_3}{R_4 R_8} + \left(R_2 + R_3 + \frac{R_2 R_3}{R_1} \right) \left(R_6 + R_7 + \frac{R_6 R_7}{R_5} \right) C_2 C_4 + 2R_2 R_3 C_2 C_4 + 2R_6 R_7 C_2 C_4$ $+ j \left[R_2 R_3 C_1 \left(R_6 + R_7 + \frac{R_6 R_7}{R_5} \right) \left(\frac{C_4 + 2C_2}{R_4} + \frac{R_6 R_7}{R_5} \right) + R_6 R_7 C_3 \left(R_2 + R_3 + \frac{R_2 R_3}{R_1} \right) \left(\frac{C_2 + 2C_4}{R_8} + \frac{C_2}{R_4} \right) \right]$
a_{P3}	$\frac{(+j)^2}{R_4 R_8} \left[\left(R_2 + R_3 + \frac{R_2 R_3}{R_1} \right) R_6 R_7 C_3 + \left(R_6 + R_7 + \frac{R_6 R_7}{R_5} \right) R_2 R_3 C_1 \right] + \left(R_2 + R_3 + \frac{R_2 R_3}{R_1} \right) C_2 + \left(R_6 + R_7 + \frac{R_6 R_7}{R_5} \right) C_4$ $+ j \left[\left(R_2 + R_3 + \frac{R_2 R_3}{R_1} \right) \left(R_6 + R_7 + \frac{R_6 R_7}{R_5} \right) \left(\frac{C_2 + C_4}{R_8} + \frac{R_6 R_7}{R_5} \right) + \frac{R_2 R_3 C_1 + R_6 R_7 C_3}{R_4} \right]$
a_{P4}	$\frac{(+j)^2}{R_4 R_8} \left(R_2 + R_3 + \frac{R_2 R_3}{R_1} \right) \left(R_6 + R_7 + \frac{R_6 R_7}{R_5} \right) + j \left[\left(R_2 + R_3 + \frac{R_2 R_3}{R_1} \right) \frac{1}{R_4} + \left(R_6 + R_7 + \frac{R_6 R_7}{R_5} \right) \frac{1}{R_8} \right]$
a_{N1}	$2R_2 R_3 C_2 C_4 \left(R_6 + R_7 + \frac{R_6 R_7}{R_5} \right) + 2R_6 R_3 C_2 C_4 \left(R_2 + R_3 + \frac{R_2 R_3}{R_1} \right) - j 2R_2 R_3 R_6 R_7 C_2 C_4 \left(\frac{C_2 + C_4}{R_6} + \frac{C_2}{R_3} \right)$
a_{N2}	$\frac{(-j)^2 R_2 R_3 R_6 R_7 C_2 C_3}{R_4 R_8} + \left(R_2 + R_3 + \frac{R_2 R_3}{R_1} \right) \left(R_6 + R_7 + \frac{R_6 R_7}{R_5} \right) C_2 C_4 + 2R_2 R_3 C_2 C_4 + 2R_6 R_7 C_2 C_4$ $- j \left[R_2 R_3 C_1 \left(R_6 + R_7 + \frac{R_6 R_7}{R_5} \right) \left(\frac{C_4 + 2C_2}{R_4} + \frac{R_6 R_7}{R_5} \right) + R_6 R_7 C_3 \left(R_2 + R_3 + \frac{R_2 R_3}{R_1} \right) \left(\frac{C_2 + 2C_4}{R_8} + \frac{C_2}{R_4} \right) \right]$
a_{N3}	$\frac{(-j)^2}{R_4 R_8} \left[\left(R_2 + R_3 + \frac{R_2 R_3}{R_1} \right) R_6 R_7 C_3 + \left(R_6 + R_7 + \frac{R_6 R_7}{R_5} \right) R_2 R_3 C_1 \right] + \left(R_2 + R_3 + \frac{R_2 R_3}{R_1} \right) C_2 + \left(R_6 + R_7 + \frac{R_6 R_7}{R_5} \right) C_4$ $- j \left[\left(R_2 + R_3 + \frac{R_2 R_3}{R_1} \right) \left(R_6 + R_7 + \frac{R_6 R_7}{R_5} \right) \left(\frac{C_2 + C_4}{R_8} + \frac{R_6 R_7}{R_5} \right) + \frac{R_2 R_3 C_1 + R_6 R_7 C_3}{R_4} \right]$
a_{N4}	$\frac{(-j)^2}{R_4 R_8} \left(R_2 + R_3 + \frac{R_2 R_3}{R_1} \right) \left(R_6 + R_7 + \frac{R_6 R_7}{R_5} \right) - j \left[\left(R_2 + R_3 + \frac{R_2 R_3}{R_1} \right) \frac{1}{R_4} + \left(R_6 + R_7 + \frac{R_6 R_7}{R_5} \right) \frac{1}{R_8} \right]$

TABLE XVI. DEVICE PARAMETERS OF RAUCH COMPLEX FILTER.

Element	Value	Element	Value
R_1	1 kΩ	R_4	3 kΩ
R_2	1 kΩ	C_1	0.55 nF
R_3	2.8 kΩ	C_2	0.35 nF
R_5	2 kΩ	R_8	3 kΩ
R_6	1 kΩ	C_3	0.6 nF
R_7	4 kΩ	C_4	0.22 nF

Values of the passive components of the 4th-order Rauch complex filter are chosen as shown in TABLE XVI. Fig. 40 shows the behaviors of the Rauch complex filter in both positive and negative frequency domains. In the positive frequency domain, the pass-band gain of this complex filter is 13 dB. The image signal rejection ratio of this complex filter is 32 dB. Compare the simulation results with mathematical analysis and other research papers, the properties of transfer functions are the same [50].

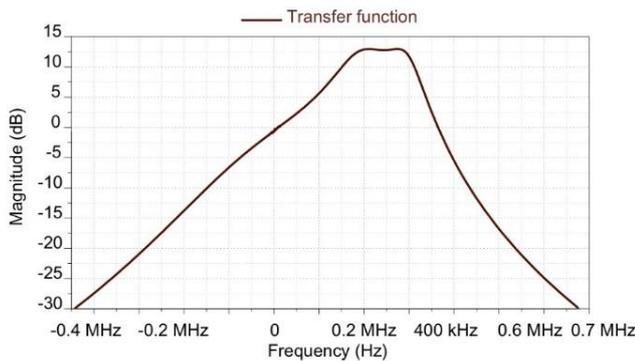


Fig. 40. Bode plot of the 4th-order Rauch complex filter.

VII. CONCLUSIONS

This paper has shown the analysis of complex functions in multi-source networks such as helix waves, polyphase signals, and high-order Rauch filters. The merits of Rauch filters are an easier selection of circuit components and a simpler design in fully differential forms. In complex functions, the imaginary unit j represents a 90-degree phase shift. A general superposition formula is proposed to analyse the behaviours of multi-source networks. This formula is considered as a fundamental concept for the theoretical multi-source network analysis because the effects of all the sources can be investigated at one time. The import of the superposition formula into the multi-source network analysis theory is relatively new with much recent progress regarding the circuit theory.

The ringing test for a fully differential 4th-order Rauch low-pass filter was performed based on the observation of the phase margin at unity gain of the self-loop function. The comparison measurement technique was used to measure the self-loop function in the transfer function of this low-pass filter. In case of under-damping, the magnitude of the transfer function is 15 dB and the phase margin is 59 degrees. Nichols chart of the self-loop function is considered as a practical tool for the stability test of electronic feedback systems. In future work, the ringing test for the parasitic components in the transmission lines, the printed circuit boards, and the physical layout layers will be done. A tool of the Nichols chart of self-loop function will be introduced.

This work also introduced a proposed design of a 4th-order Rauch complex filter. The image rejection ratio of this complex filter was also defined. The Bode plot of the Rauch complex filter was expressed in all frequency domains. The pass-band gain of this complex filter is 13 dB and the image signal rejection ratio is 32 dB. In future work, the effects of DC offsets and IQ mismatches in multi-source networks will be analyzed.

ACKNOWLEDGEMENTS

The authors would like to express their sincere gratitude to Prof. Hiroshi Tanimoto at Kitami Institute of Technology in Japan for his valuable comments.

REFERENCES

- [1] M. Tran, A. Kuwana, H. Kobayashi, "Study of Rauch Low-Pass Filters using Pascal's Triangle", Int. Conf. on Elect., Info., & Com. (ICEIC 2021), Korea, Feb. 2021.
- [2] I. Campanu, T. Salajan, R. Onet, M. Neag, "Comparative Analysis of Four Second-Order OA-RC Polyphase Filters for An ISM Low-IF Receiver," CAS 2012 Int. Semi. Conf., Romania, Oct. 2012.
- [3] H. Kobayashi, M. Tran, K. Asami, A. Kuwana, H. San, "Complex Signal Processing in Analog / Mixed-Signal Circuits", 3rd Int. Conf. on Tech. & Soc. Sci. (ICTSS2019), Japan, May, 2019.
- [4] M. Tran, Y. Sun, Y. Kobori, A. Kuwana, H. Kobayashi, "Design Proposal for Inductor Type Buck Converter in Bluetooth Receiver Chip with Overshoot Cancellation and Ripple Reduction Technique" Adv. Eng. For., vol. 38, pp.179-191, Nov. 2020.
- [5] M. Tran, A. Hatta, A. Kuwana, H. Kobayashi, "Derivation of Image Rejection Ratio for High-Order Complex Filters," 6th Conf. on TJCAS, Nov. 2020.
- [6] Y. Kobori, Y. Sun, M. Tran, A. Kuwana, H. Kobayashi, "EMI Reduction in Switching Converters with Pseudo Random Analog Noise", J. of Tech. & Soc. Sci., vol.4, no.2, pp.33-48, Apr. 2020.
- [7] M. Tran, A. Kuwana, H. Kobayashi, "Study of Complex Functions in Multi-Source Networks", The 11th Annual Com. & Com. Work. & Conf. (IEEE CCWC 2021), USA, Jan. 2021.



- [8] Y. Sun, Y. Kobori, M. Tran, A. Kuwana, H. Kobayashi, "Single-Inductor Dual-Output Converters With PWM, Hysteretic, Soft Switching and Current Controls", *J. of Tech. & Soc. Sci.*, vol.4, no.2, pp.7-32, Apr. 2020.
- [9] M. Tran, Y. Sun, N. Oiwa, Y. Kobori, A. Kuwana, H. Kobayashi, "Mathematical Analysis and Design of Parallel RLC Network in Step-down Switching Power Conversion System", *J. of Mech. & Elect. Int. Sys.*, vol. 3, no. 2, May 2020.
- [10] C. Zhang, Y. Sun, M. Tran, Y. Kobori, A. Kuwana, H. Kobayashi, "Multi-Phase Full/Half Wave Type Resonant Converters with Automatic Current Balance against Element Variation", 13th IEEE Int. Conf. on ASIC (ASICON 2019), Oct. 2019.
- [11] M. Tran, A. Kuwana, H. Kobayashi, "Derivation of Loop Gain and Stability Test for Multiple Feedback Low Pass Filter Using Deboo Integrator", 8th IIAE Int. Conf. on Ind. App. Eng., Japan, March 2020.
- [12] S. Delshadpour, "A 2.6 MHz Bandwidth, 3rd/5th Order Active-RC Polyphase Filter with Quadrature Offset Cancellation for Low-IF GPS Radio", 62nd Int. Mid. Symp. CAS, USA, Oct. 2019.
- [13] M. Tran, A. Hatta, A. Kuwana, H. Kobayashi, "Design of 6th-Order Passive Quadrature Signal Generation Network Based on Polyphase Filter", IEEE 15th Int. Conf. on Solid-State & Integ. Cir. Tech. (ICSICT2020), China, Nov. 2020.
- [14] A. Hatta, N. Kushita, M. Tran, K. Asami, A. Kuwana, H. Kobayashi, "Relationship Between Active Complex Bandpass Filter and Hilbert Filter" 5th Taiwan & Japan Conf. on CAS (TJCAS 2019), Aug. 2019.
- [15] M. Tran, A. Kuwana, H. Kobayashi, "Design of Active Inductor and Stability Test for Ladder RLC Low Pass Filter Based on Widened Superposition and Voltage Injection", 8th IIAE Int. Conf. on Ind. App. Eng., Japan, March, 2020.
- [16] M. Tran, A. Kuwana, H. Kobayashi "Ringing Test for Negative Feedback Amplifiers" 11th IEEE Annu. Inf. Tech., Elect. & Mob. Com. Conf. (IEMCON 2020), Canada, Nov. 2020.
- [17] L. Severo, W. Noije, "A 0.4-V 10.9- μ W/Pole 3rd-Order Complex BPF For Low Energy RF Receivers," IEEE Trans. CAS I, vol. 66, no. 6, pp. 2017–2026, 2019.
- [18] M. Tran, A. Kuwana, H. Kobayashi, "Design of Active Inductor and Stability Test for Passive RLC Low Pass Filter", 10th Int. Conf. on Comp. Sci., Eng. & App. (CCSEA 2020), UK, Jul. 2020.
- [19] Y. Kobori, J. Li, Y. Sun, M. Tran, A. Kuwana, H. Kobayashi, "Automatic Current Balance for Multi-Phase Switching Converters with Ripple Control or Soft Switch", *Adv. Eng. For.*, vol. 38, pp.143-156, Nov. 2020.
- [20] M. Tran, A. Kuwana, H. Kobayashi, "Derivation of Loop Gain and Stability Test for Low Pass Tow-Thomas Biquad Filter", 10th Int. Conf. on Comp. Sci., Eng. & App. (CCSEA 2020), UK, Jul. 2020.
- [21] M. Tran, A. Kuwana, H. Kobayashi, "Ringing Test for Tow-Thomas Low-Pass Filters", Int. Conf. on Prom. Elect. Tech. (ICPET 2020), Palestine, 16-17, Dec. 2020.
- [22] Y. Jou, H. Li, F. Chen, "Polyphase Design of 2-Channel Quadrature Mirror Filter Using Minor Component Analysis", IEEE 8th Global Conf. Consu. Elect., Japan, Oct. 2019.
- [23] M. Tran, A. Kuwana, H. Kobayashi, "Study of Behaviors of Electronic Amplifiers using Nichols Chart", IEEE the 3rd Int. Conf. on Elect. & Com. Eng. (ICECE 2020), Xi'an, China, 14-16, Dec. 2020.
- [24] Y. Kobori, Y. Sun, M. Tran, A. Kuwana, H. Kobayashi, "EMI Noise Reduction of DC-DC Switching Converters with Automatic Output Ripple Cancellation" *Adv. Eng. For.*, vol. 38, pp.157-170, Nov. 2020.
- [25] M. Tran, "Study of Multiphase Networks, Noise Reduction for DC-DC Converters, and Stability Test for Electronic Systems", ATS Doct. Dissert. Award Contest, 29th IEEE ATS, Malaysia, Nov. 2020.
- [26] Y. Kobori, Y. Sun, M. Tran, A. Kuwana, H. Kobayashi, "Pulse Coding Controlled Switching Converter with Notch Frequency Generation in Noise Spectrum", *Adv. Eng. For.*, vol. 38, pp.118-129, Nov. 2020.
- [27] M. Tran, A. Kuwana, H. Kobayashi, "Ringing Test for 3rd-Order Ladder Low-Pass Filters", 11th IEEE Ann. Ubiqu. Com., Elect. & Mob. Com. Conf. (UEMCON 2020), USA, Oct. 2020.
- [28] Z. Xie, J. Wu, C. Chen, "A Compact Low-Power Biquad for Active-RC Complex Filter" IEEE Trans. CAS II, vol. 65, no. 6, pp. 709-713, 2018.
- [29] M. Tran, A. Kuwana, H. Kobayashi, "Ringing Test for 2nd-Order Sallen-Key Low-Pass Filters", IEEE 2nd Int. Conf. on CAS (ICCS 2020), China, 10-13, Dec. 2020.
- [30] S. Delshadpour, "A Low Power CMOS gm-C Polyphase Filter for Low-IF GPS Receiver", 25th IEEE Int. Conf. Elect., CAS, France, Jan. 2019.
- [31] M. Tran, Y. Kobori, A. Kuwana, H. Kobayashi, "Phase Margin Test for Power-Stage of DC-DC Buck Converter", 4th Int. Conf. on Tech. & Soc. Sci. (ICTSS 2020), Japan, Dec. 2020.
- [32] D. Maier, H. Brachtendorf, "Design of an active 2nd-order polyphase filter for image rejection in subsampling receivers", IEEE 28th Int. Conf. Radioelektronika, Czech Republic, Apr. 2018.
- [33] M. Tran, A. Kuwana, H. Kobayashi, "Stability Test for High-Order Ladder Low-Pass Filters", 6th Conf. on TJCAS, Nov. 2020.
- [34] M. Tran, Y. Kobori, A. Kuwana, H. Kobayashi, "Design of LC Harmonic Notch Filter for Ripple Reduction in Step-Down DC-DC Buck Converter", IEEE 15th Int. Conf. on Solid-State & Integ. Cir. Tech. (ICSICT2020), China, Nov. 2020.
- [35] M. Tran, N. Kushita, A. Kuwana, H. Kobayashi, "Mathematical Analysis and Design of 4-Stage Passive RC Network in RF Front-End System", 3rd Int. Conf. on Tech. & Soc. Sci. (ICTSS 2019), Japan, May, 2019.
- [36] Y. Sun, M. Tran, Y. Kobori, A. Kuwana, H. Kobayashi, "EMI Noise Reduction and Output Ripple Cancellation for Full-Wave Type Soft-Switching Converter", 13th IEEE Int. Conf. on ASICON, Oct. 2019.
- [37] M. Tran, A. Kuwana, H. Kobayashi, "Measurements of Self-loop Functions in High-order Passive and Active Low-Pass Filters," IEEE 15th Int. Conf. on Solid-State & Integ. Cir. Tech. (ICSICT2020), China, Nov. 2020.
- [38] S. Katayama, J. Li, Y. Sun, M. Tran, Y. Kobori, A. Kuwana, H. Kobayashi, "Automatic Correction of Current Imbalance for Multi-Phase COT Ripple-Based Control DC-DC Converter", 13th IEEE Int. Conf. on ASICON, Oct. 2019.
- [39] M. Tran, N. Kushita, A. Kuwana, H. Kobayashi, "Flat Pass-Band Method with Two RC Band-Stop Filters for 4-Stage Passive RC Polyphase Filter in Low-IF Receiver Systems", 3th IEEE Int. Conf. on ASICON, Oct. 2019.
- [40] Y. Sun, M. Tran, Y. Kobori, A. Kuwana, H. Kobayashi, "Pulse Coding Control Switching Converter with Adjustable Conversion Voltage Ratio Notch Frequency Generation in Noise Spectrum", 13th IEEE Int. Conf. on ASICON, Oct. 2019.
- [41] M. Tran, N. Kushita, A. Kuwana, H. Kobayashi, "Pass-band Gain Improvement Technique for Passive RC Polyphase Filter in Bluetooth Low-IF Receiver using Two RC Band-stop Filters", *Advanced Engineering Forum*, vol. 38, pp.192-205, Nov. 2020.
- [42] Y. Kobori, N. Oiwa, S. Katayama, A. Bustoni, Y. Sun, M. Tran, A. Kuwana, H. Kobayashi, "EMI Reduction of PFC Rectifier and LLC Converter with Automatic Output Ripple Improvement", *Adv. Eng. For.*, vol. 38, pp.103-117, Nov. 2020.
- [43] M. Tran, Y. Sun, N. Oiwa, Y. Kobori, A. Kuwana, H. Kobayashi "Fast Response, Small Ripple, Low Noise Switching Converter with Digital Charge Time Control and EMI Harmonic Filter", *J. of Mech. & Elect. Int. Sys.*, vol. 2, no. 1, pp.14-23, Dec. 2019.
- [44] L. Severo, W. Noije, "A 10.9- μ W/Pole 0.4-V Active-RC Complex BPF for Bluetooth Low Energy RF Receivers", IEEE 9th Latin Ame. Symp. CAS, Mexico, Feb. 2018.
- [45] M. Tran, N. Kushita, A. Kuwana, H. Kobayashi, "Mathematical Model and Analysis of 4-Stage Passive RC Polyphase Filter for Low-IF Receiver", *J. of Mech. & Elect. Int. Sys.*, vol. 3, no. 2, May 2020.
- [46] K. Asami, N. Kushita, A. Hatta, M. Tran, Y. Tamura, A. Kuwana, H. Kobayashi, "I, Q Mismatch Analysis and Evaluation Method of Complex Analog Filter", 5th Conf. on TJCAS, Aug. 2019.
- [47] H. Kobayashi, N. Kushita, M. Tran, K. Asami, H. San, A. Kuwana, "Analog / Mixed-Signal / RF Circuits for Complex Signal Processing", 13th IEEE Int. Conf. on ASICON, Oct. 2019.
- [48] M. Tran, N. Kushita, A. Kuwana, H. Kobayashi, "Minimum Gain Ripple Technique for Pass-Band of 4-Stage Passive RC Polyphase Filter in Low IF Receivers", 5th Conf. on TJCAS, Aug. 2019.
- [49] K. Asami, N. Kushita, A. Hatta, M. Tran, Y. Tamura, A. Kuwana, H. Kobayashi, "Analysis and Evaluation Method of RC Polyphase Filter", 13th IEEE Int. Conf. on ASICON, Oct. 2019.
- [50] Y. Yin, S. Wang, Y. Ma, S. Kang, H. Guan, G. Jin, X. Qi, "The Design of Large Image Rejection and Wideband CMOS Active Polyphase Filter for BeiDou RF Receiver", *IEICE Elec. Express*, vol. 17, no. 12, 2020.



Filter-based Denoising Methods for AWGN corrupted images

Sumit Singh Parihar¹, Shailesh Khaparkar²

Student¹, Assistant Professor²

Department of Electronics & Communication Engineering^{1,2},
Gyan Ganga Institute of Technology & Sciences, Jabalpur

ABSTRACT: Visual information transfer in the form of digital images becomes a vast method of communication in the modern scenario, but the image obtained after transmission is many a times corrupted with noise. The received image requires some processing before it can be used. Image denoising includes the manipulation of the image data to produce a visually high-quality image. In this paper a review of some existing denoising algorithms, such as filtering approach; wavelet-based approach and their comparative study has been done. Different noise models including additive and multiplicative types are discussed. It includes Gaussian noise, salt and pepper noise, speckle noise and Brownian noise. Selection of the denoising algorithm is application dependent. Hence, it is necessary to have knowledge about the noise present in the image so that one can opt the appropriate denoising algorithm. The filtering approach seems to be a better choice when the image is corrupted with salt and pepper noise. Whereas, wavelet-based techniques are suited for more detailing. In this paper denoising techniques for AWGN corrupted image has been mainly focused.

KEYWORDS: AWGN, Image Denoising, Noise, Filtering, DWT, threshold.

I. INTRODUCTION

A very large portion of digital image processing is dedicated to image restoration. It includes research in algorithm development and routine goal-oriented image processing. Image restoration is the removal or reduction of degradations that are incurred while the image is being obtained. Degradation comes from blurring as well as noise due to electronic and photometric sources.[12] Blurring is a form of bandwidth reduction of the image caused by the imperfect image formation process such as relative motion between the camera and the original scene or by an optical system that is out of focus. When aerial photographs are produced for remote sensing purposes, blurs are introduced by atmospheric turbulence, aberrations in the optical system and relative motion between camera and ground. In addition to these blurring effects, the recorded image is corrupted by noises too. A noise is introduced in the transmission medium due to a noisy channel, errors during the measurement process and during quantization of the data for digital storage. Each element in the imaging chain such as lenses, film, digitizer, etc. contributes to the degradation [13].

Let us now consider the representation of a digital image. A 2-dimensional digital image can be represented as a 2-

dimensional array of data $s(x,y)$, where (x,y) represent the pixel location. The pixel value corresponds to the brightness of the image at location (x,y) . Some of the most frequently used image types are binary, gray-scale and color. Binary images are the simplest type of images and can take only two discrete values, black and white. Black is represented with the value „0“ while white with „1“. Note that a binary image is generally created from a gray-scale image. A binary image finds applications in computer vision areas where the general shape or outline information of the image is needed. They are also referred to as 1 bit/pixel images [5]. Gray-scale images are known as monochrome or one-color images. The images used for experimentation purposes in this thesis are all gray-scale images. They contain no color information. They represent the brightness of the image. This image contains 8 bits/pixel data, which means it can have up to 256 (0-255) different brightness levels.

For representation of pixels in brightness format “0” represents black and “255” denotes white. In between values from 1 to 254 represent the different gray levels. As they contain the intensity information, they are also referred to as intensity images. Color images are considered as three band monochrome images, where each band is of a different color. Each band provides the brightness information of the corresponding spectral band [6]. Typical color images are red, green and blue images and are also referred to as RGB images. This is a 24 bits/pixel image.

The main aim of this paper is to review all the existing methodology which are used for estimation of the uncorrupted image from the distorted or noisy image, and is also referred to as image “denoising”. There are various methods to help restore an image from noisy distortions. Selecting the appropriate method plays a major role in getting the desired image. The denoising methods tend to be problem specific. For example, a method that is used to denoise satellite images may not be suitable for denoising medical images. In this paper it is proposed that a study would be made on the various denoising algorithms. In case of image denoising methods, the characteristics of the degrading system and the noises are assumed to be known.

II. BASIC NOISE THEORY

Noise is defined as an unwanted signal that interferes with the communication or measurement of another signal. A noise itself is an information-bearing signal that conveys information regarding the sources of the noise and the environment in which it propagates.



There are many types and sources of noise or distortions and they include [7], thermal noise and shot noise, Acoustic noise, Electromagnetic noise Electrostatic noise, Quantization noise. Signal distortion is the term often used to describe a systematic undesirable change in a signal and refers to changes in a signal from the non-ideal characteristics of the communication channel, signal fading reverberations, echo, and multipath reflections and missing samples. Depending on its frequency, spectrum or time characteristics, a noise process is further classified into several categories:

II.I Additive and Multiplicative Noises

Noise is present in an image either in an additive or multiplicative form. An additive noise follows the rule [9]:

$$w(x,y) = s(x,y) + n(x,y)$$

While the multiplicative noise satisfies

$$w(x,y) = s(x,y) \times n(x,y)$$

Where $s(x,y)$ is the original signal, $n(x,y)$ denotes the noise introduced into the signal to produce the corrupted image $w(x,y)$, and (x,y) represents the pixel location.

II.II Gaussian Noise

Gaussian noise is evenly distributed over the signal this means that each pixel in the noisy image is the sum of the true pixel value and a random Gaussian distributed noise value. As the name indicates, this type of noise has a Gaussian distribution, which has a bell shaped probability distribution function given by,

$$F(g) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(g-m)^2}{2\sigma^2}}$$

Where g represents the gray level, m is the mean or average of the function and σ is the standard deviation of the noise.

II.III Salt and Pepper Noise

Salt and pepper noise is an impulse type of noise, which is also referred to as intensity spikes. This is caused generally due to errors in data transmission. It has only two possible values, a and b . The probability of each is typically less than 0.1. The corrupted pixels are set alternatively to the minimum or to the maximum value, giving the image a “salt and pepper” like appearance. Unaffected pixels remain unchanged. For an 8-bit image, the typical value for pepper noise is 0 and for salt noise 255 [16]. The salt and pepper noise is generally caused by malfunctioning of pixel elements in the camera sensors, faulty memory locations, or timing errors in the digitization process. The probability density function for this type of noise is shown in Figure 2.

II.IV Speckle Noise

Speckle noise is a multiplicative noise. This type of noise occurs in almost all coherent imaging systems such as laser, acoustics and SAR (Synthetic Aperture Radar) imagery. The source of this noise is attributed to random interference between the coherent returns. Fully developed speckle noise

has the characteristic of multiplicative noise. Speckle noise follows a gamma distribution and is given as:

$$F(g) = \frac{g^{\alpha-1}}{(\alpha-1)! \alpha^\alpha} e^{-\frac{g}{\alpha}}$$

Where variance is α^2 and g is the gray level.

On an image, speckle noise (with variance 0.05) looks as shown in Figure 3

II.V Brownian Noise

Brownian noise comes under the category of fractal or $1/f$ noises. The mathematical model for $1/f$ noise is fractional Brownian motion. Fractal Brownian motion is a non-stationary stochastic process that follows a normal distribution. Brownian noise is a special case of $1/f$ noise. It is obtained by integrating white noise. It can be graphically represented as shown in Figure 5. On an image, Brownian noise would look like Image 6.

III. FILTER-BASED IMAGE DENOISING

A. Linear and Nonlinear Filtering Approach

Filters play a major role in the image restoration process. The basic concept behind image restoration using linear filters is digital convolution and moving window principle. Let $w(x)$ be the input signal subjected to filtering, and $z(x)$ be the filtered output. If the filter satisfies certain conditions such as linearity and shift invariance, then the output filter can be expressed mathematically in simple form as

$$z(X) = \int W(t)h(x-t)dt$$

Where $h(t)$ is called the point spread function or impulse response and is a function that completely characterizes the filter. The integral represents a convolution integral and, in short, can be expressed as:

$$z = w * h.$$

For a discrete case, the integral turns into a summation as

$$z(i) = \sum_{t=-\infty}^{+\infty} w(t)h(i-t)$$

B. Mean Filter

A mean filter acts on an image by smoothing it; that is, it reduces the intensity variation between adjacent pixels [3]. The mean filter is nothing but a simple sliding window spatial filter that replaces the center value in the window with the average of all the neighboring pixel values including itself. By doing this, it replaces pixels that are unrepresentative of their surroundings. It is implemented with a convolution mask, which provides a result that is a weighted sum of the values of a pixel and its neighbors. It is also called a linear filter. The mask or kernel is a square. Often a 3×3 square kernel is used.:

C. LMS Adaptive Filter

An adaptive filter does a better job of denoising images compared to the averaging filter. The fundamental difference



between the mean filter and the adaptive filter lies in the fact that the weight matrix varies after each iteration in the adaptive filter while it remains constant throughout the iterations in the mean filter [15]. Compared to other adaptive filters, the Least Mean Square (LMS) adaptive filter is known for its simplicity in computation and implementation. The basic model is a linear combination of a stationary low-pass image and a non-stationary high-pass component through a weighting function. Thus, the function provides compromise between resolution of genuine features and suppression of noise [9].

D. Median Filter

A median filter belongs to the class of nonlinear filters unlike the mean filter [7]. The median filter also follows the moving window principle similar to the mean filter. A 3×3, 5×5, or 7×7 kernel of pixels is scanned over pixel matrix of the entire image. The median of the pixel values in the window is computed, and the center pixel of the window is replaced with the computed median. Median filtering is done by, first sorting all the pixel values from the surrounding neighborhood into numerical order and then replacing the pixel being considered with the middle pixel value [14]. Note that the median value must be written to a separate array or buffer so that the results are not corrupted as the process is performed.

E. External Bilateral Filter

Filters based on Gaussian functions are of particular importance because their shapes are easily specified and both the forward and inverse Fourier transforms of a Gaussian function are real Gaussian functions. Further if the frequency domain filter is narrower, the spatial domain filter will be wider which attenuates the low frequencies resulting in increased smoothing/blurring. These Gaussian filters are typical linear filters that have been widely used for image denoising. Gaussian filters assume that images have smooth spatial variations and pixels in a neighborhood have close values, by averaging the pixel values over a local neighborhood suppresses noise while preserving image features. However, this assumption fails at edges where the spatial variations are not smooth and the application of Gaussian filter blurs the edges. Bilateral filter overcomes this by filtering the image in both range and domain (space). Bilateral filtering is a local, nonlinear and non-iterative technique which considers both gray level (color) similarities and geometric closeness of the neighboring pixels.

F. Discrete Wavelet Transform Filter-banks

Wavelets are mathematical functions that analyze data according to scale or Resolution. They aid in studying a signal in different windows or at different resolutions. For instance, if the signal is viewed in a large window, gross features can be noticed, but if viewed in a small window, only small features can be noticed. Wavelets provide some advantages over Fourier transforms. For example, they do a good job in approximating signals with sharp spikes or signals having discontinuities. Wavelets can also model speech, music, video and non-stationary stochastic signals. Wavelets can be used in

applications such as image compression, turbulence, human vision, radar, earthquake prediction, etc.

The term “wavelets” is used to refer to a set of Ortho-normal basis functions generated by dilation and translation of scaling function ϕ and a mother wavelet ψ . The finite scale multi resolution representation of a discrete function can be called as a discrete wavelet transform. DWT is a fast linear operation on a data vector, whose length is an integer power of 2. This transform is invertible and orthogonal, where the inverse transform expressed as a matrix is the transpose of the transform matrix. The wavelet basis or function, unlike sine and cosines as in Fourier transform, is quite localized in space. But similar to sine and cosines, individual wavelet functions are localized in frequency.

DWT is the multi resolution description of an image. The decoding can be processed sequentially from a low resolution to the higher resolution. DWT splits the signal into high and low frequency parts [8]. The high frequency part contains information about the edge components, while the low frequency part is split again into high and low frequency parts. The high frequency components are usually used for watermarking since the human eye is less sensitive to changes in edges. In two dimensional applications, for each level of decomposition, we first perform the DWT in the vertical direction, followed by the DWT in the horizontal direction. After the first level of decomposition, there are 4 sub-bands: LL1, LH1, HL1, and HH1. For each successive level of decomposition, the LL Sub-band of the previous level is used as the input. To perform second level decomposition, the DWT is applied to LL1 band which decomposes the LL1 band into the four sub-bands LL2, LH2, HL2, and HH2. To perform third level decomposition, the DWT is applied to LL2 band which decompose this band into the four sub-bands – LL3, LH3, HL3, HH3. This results in 10 sub-bands per component. LH1, HL1, and HH1 contain the highest frequency bands present in the image tile, while LL3 contains the lowest frequency band.

DWT is currently used in a wide variety of signal processing applications, such as in audio and video compression, removal of noise in audio, and the simulation of wireless antenna distribution [5].

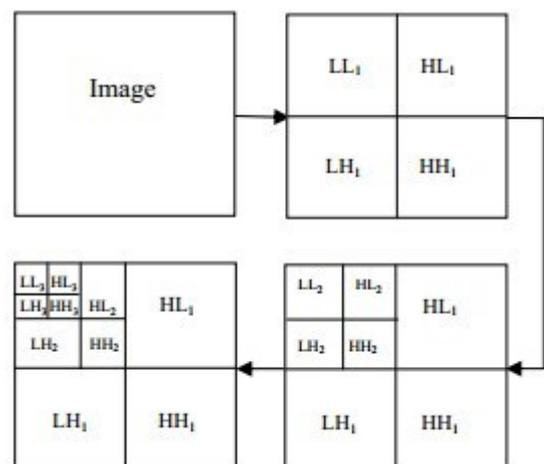


Fig. 1: 3 level discrete wavelet decomposition



Wavelets have their energy concentrated in time and are well suited for the analysis of transient, time-varying signals. Since most of the real life signals encountered are time varying in nature, the Wavelet Transform suits many applications very well. As mentioned earlier, the wavelet equation produces different wavelet families like Daubechies, Haar, Coiflets, etc [10].

IV. WAVELET THRESHOLDING

The term wavelet thresholding is explained as decomposition of the data or the image into wavelet coefficients, comparing the detail coefficients with a given threshold value, and shrinking these coefficients close to zero to take away the effect of noise in the data. The image is reconstructed from the modified coefficients. This process is also known as the inverse discrete wavelet transform. During thresholding, a wavelet coefficient is compared with a given threshold and is set to zero if its magnitude is less than the threshold; otherwise, it is retained or modified depending on the threshold rule. Thresholding distinguishes between the coefficients due to noise and the ones consisting of important signal information [4].

A noisy image is decomposed into a low frequency approximation sub-image and a series of high frequency detail sub-images at different scales and directions via transform. To estimate the noise-free coefficients in detail sub-bands, a Bayesian estimator is developed [2].

The choice of a threshold is an important point of interest. It plays a major role in the removal of noise in images because denoising most frequently produces smoothed images, reducing the sharpness of the image [11]. Care should be taken so as to preserve the edges of the denoised image. There exist various methods for wavelet thresholding, which rely on the choice of a threshold value. Some typically used methods for image noise removal include VisuShrink, SureShrink and BayesShrink. Prior to the discussion of these methods, it is necessary to know about the two general categories of thresholding. They are hard- thresholding and soft-thresholding types.

Wavelet theory has been developed rapidly in recent years, which has increasingly wide application in image denoising. It is very important to select threshold function and threshold in wavelet threshold denoising algorithm. Different selections will affect the denoising effect directly. In [1], traditional soft and hard threshold functions were further analyzed and studied, advantages of denoising performances in both soft and hard threshold functions were combined for an improved threshold function. Threshold proposed by Donoho was improved according to characteristics of wavelet decomposition layers and noise wavelet coefficient in the paper [8].

V. SIMULATION RESULTS & DISCUSSIONS

To check the performance of image denoising with filtering techniques alone, simulation has been performed for Lena test image using bilateral filters only. The performance of bilateral filtered denoised image has been compared for MSE, MAE, PSNR and SSIM image quality parameters with recent similar

kind of image denoising methods. Simulation results values of PSNR & MSE for different wavelets are tabulated in Table-I.

Table-I: Simulation Results for Test Image Lena

Noise Variance	Noisy Image				Denoised using external bilateral Filtering			
	PSNR (dB)	MSE	MAE	SSIM	PSNR (dB)	MS E	MAE	SSIM
5	34.1	25.08	22	0.842	34.58	22.6	38	0.902
10	28.1	100.0	45	0.607	32.04	40.6	53	0.868
15	24.6	224.2	67	0.446	30.64	56.1	71	0.841
20	22.1	398.1	89	0.340	29.73	69.1	82	0.817
25	20.2	616.6	111	0.270	29.05	80.8	88	0.792
30	18.7	877.3	133	0.220	28.47	92.3	92	0.766



Figure 2: Original test image ‘Lena’ used for simulation.

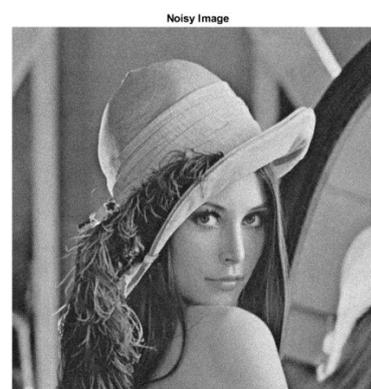


Figure 3: Gaussian noise corrupted test image ‘Lena’ for noise variance $\sigma=10$.

For simulation first image is taken and it is corrupted by gaussian noise from range of 5 to 30 dB, to check the effectiveness of bilateral filter approach for a wide range of noise variance, then the image is denoised using external bilateral filter.



Figure 4: Denoised noisy image ‘Lena’ using external bilateral filtering only.

To check the effectiveness simulations results comparison has been done, which is shown in Table-II. It can be seen that the bilateral filtering approach seems to outperforms many of the existing denoising methods, in terms of denoised image PSNR.

Algorithm	Noise →	$\sigma = 10$	$\sigma = 20$	$\sigma = 30$
	Variance σ			
This Work		32.04	29.73	28.47
NIG-WT [5]		31.97	28.42	26.27
Visu-shrink [12]		30.65	27.76	26.33

VI. CONCLUSION

In this paper image denoising techniques for the AWGN corrupted has been given. This paper reviews the existing denoising algorithms, such as filtering approach; wavelet-based approach. Different noise models including additive and multiplicative types are used. They include Gaussian noise, salt and pepper noise, speckle noise and Brownian noise. The filtering approach seems to be a better choice when the image is corrupted with salt and pepper noise and to check the performance of bilateral filtering approach a simulation exercise has been performed. The wavelet-based approach finds applications in denoising images corrupted with Gaussian noise. Selection of the denoising algorithm is application dependent. Hence, it is necessary to have knowledge about the noise present in the image so as to select the appropriate denoising algorithm. In the sequel paper performance of external bilateral filtering along with wavelet domain thresholding will be checked.

REFERENCES

[1] Tom Tirer et al., “Image Restoration by Iterative Denoising and Backward Projections”, IEEE Transactions on Image Processing, Volume: 28, Issue: 3, March 2019.

[2] H. Sadreazami et al., “Contourlet Domain Image Denoising based on the Bessel k-form Distribution”, IEEE 28th Canadian Conference on Electrical and Computer Engineering Halifax, Canada, May 3-6, 2015.

[3] Liqiang Shi, “An Improved Image Denoising Algorithm”, Seventh IEEE International Conference on Measuring Technology and Mechatronics Automation, 2015.

[4] Cuong Cao Pham et al., “Efficient image sharpening and denoising using adaptive guided image filtering”, IEEE, IET Image Processing Magazine, Pp. 71 – 79, 2015.

[5] Wangmeng Zuo et al., “Gradient Histogram Estimation and Preservation for Texture Enhanced Image Denoising”, IEEE Transactions on Image Processing, Volume 23, Nn. 6, JUNE 2014.

[6] Vikas Gupta et al., “Image Denoising using Wavelet Transform method”, Tenth IEEE International Conference on Wireless and Optical Communications Networks (WOCN), Pp 1-4, 2013.

[7] Fuqing Jia et al., “Image Denoising Using Hyper-Laplacian Priors and Gradient Histogram Preservation Model”, 12th IEEE International Conference on Signal Processing (ICSP), 2014.

[8] Ajay Boyat et al., “Image Denoising using Wavelet Transform and Median Filtering”, Nirma University IEEE International Conference on Engineering (NUICONE), 2013.

[9] Paras Jain & Vipin Tyagi, “Spatial and frequency domain filters for restoration of noisy images”, IETE Journal of Education, 54(2), 108-116, 2013.

[10] Maggioni, M., Katkovnik, V., Egiazarian, K., Foi, “A.: Nonlocal transform-domain filter for volumetric data denoising and reconstruction”, IEEE Transaction on Image Processing, 22(1), 119–133, 2013.

[11] Silva, R.D., Minetto, R., Schwartz, W.R., Pedrini, H.: Adaptive edge-preserving image denoising using wavelet transforms. Pattern Analysis and Applications. Springer, Berlin doi:10.1007/s10044-012-0266-x, 2012.

[12] Zhang, Y., Li, C., Jia, J., “Image denoising using an improved bivariate threshold function in tetrolet domain”, 2013.

[13] Dai, L., Zhang, Y., Li, Y.: Image denoising using BM3D combining tetrolet prefiltering. Inf. Technol. J. 12(10), 1995–2001, 2013.

[14] He, K., Sun, J., Tang, X.: Guided image filtering. In: Proceedings European Conference on Computer Vision, pp. 1–14, 2010.

[15] Porikli, F., “Constant time O(1) bilateral filtering”, In Proceeding IEEE Conference on Computer Vision and Pattern Recognition, Anchorage, pp. 1–8, 2008.

[16] Yang, Q., Tan, K.H., Ahuja, N., “Real-time O(1) bilateral filtering”, In Proceedings IEEE Conference on Computer Vision and Pattern Recognition, Miami, pp. 557–564, 2009.

[17] Farbman, Z., Fattal, R., Lischinski, D., Szeliski, “R.: Edge-preserving decompositions for multi-scale tone and detail manipulation”, ACM Transactions on Graphics 27(3), 1–10, 2008.

[18] Paris, S., Durand, F., “A fast approximation of the bilateral filter using signal processing approach”, In the Proceeding of European Conference on Computer Vision, pp. 568–580, 2006.

[19] Gonzalez, R.C., Woods, R.E.: Digital image processing, 3rd edn. Prentice-Hall, Upper Saddle River, 2008.

[20] Blu, T., Luisier, F., “The SURE-LET approach to image denoising”, IEEE Transaction Image Processing, 16(11), 2778–2786, 2007.

[21] Paris, S., Durand, F.: A fast approximation of the bilateral filter using signal processing approach. In: Proceeding European Conference on Computer Vision, pp. 568–580, 2006.

[22] Dabov, K., Foi, A., Katkovnik, V., Egiazarian, K.: Image denoising with block-matching and 3D filtering. In: SPIE electronic imaging: algorithms and systems, vol. 6064, pp. 606414-1–606414-12, 2006.

[23] Yuan, X., Buckles, B.: Subband noise estimation for adaptive wavelet shrinkage. In: Proceeding 17th International Conference on Pattern Recognition, vol. 4, pp 885–888, 2004.

[24] Elad, M.: On the origin of the bilateral filter and ways to improve it. IEEE Transaction Image Processing, 11(10), 1141–1151, 2002.

[25] Sendur, L., Selesnick, I.W.: Bivariate shrinkage functions for wavelet-based denoising exploiting interscale dependency. IEEE Trans. Signal Process. 50(11), 2744–2756, 2002.

[26] Chang, S., Yu, B., Vetterli, M.: Spatially adaptive wavelet thresholding based on context modeling for image denoising. IEEE Trans. Image Process. 9(9), 1522–1531, 2000.