

# AJSE

## American Journal of Science & Engineering

Volume 1 Issue 3

October 2020



**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**

Capitol Technology University, USA

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

Dr. Chuck Easttom is the author of 29 books, including several on computer security, forensics, and cryptography. He has also authored scientific papers on digital forensics, cyber warfare, cryptography, and applied mathematics. He is an inventor with 20 computer science patents. He holds a Doctor of Science (D.Sc.) in cyber security (dissertation topic: a study of lattice-based algorithms for post quantum cryptography) and a Doctor of Philosophy (Ph.D.) in Technology focused on bioengineering and nanotechnology. Dissertation topic is "The effects of nonlinear dynamics on nanotechnology and bioengineering" as well as three master's degrees (one in applied computer science and one in systems engineering). He also has 55 industry certifications (CISSP, CEH, etc.) He has been active in the IEEE Systems and Software Engineering Standards Committee in the 2675 DevOps group and the P2731 Brain

Computer Interface standards group, as well as a reviewer for IEEE Open Access. He is a Distinguished Speaker of the ACM and senior member of the ACM as well as a Distinguished Visitor of the IEEE and an IEEE Senior Member. Dr. Easttom is an adjunct lecturer for Georgetown University teaching graduate cyber security courses in their Master of Professional Studies in Applied Intelligence program and an adjunct professor for the University of Dallas teaching digital forensics.

### **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** (Capitol Technology 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
1-7	<p><b>A Two-User Variational Mode Decomposition Algorithm for Blind Source Separation of Arbitrary Signals</b></p> <p><i>Blind source separation (BSS) is a problem wherein two unknown source signals that have been combined in some fashion when collected at a single receiver must be separated. This problem is exacerbated by source signals that are time-varying, i.e. non-stationary, in nature. All commonly used techniques, such as Fourier analysis, wavelet analysis, and adaptive filtering algorithms do not apply. First, there is no training data, the data may not be digital in nature but have a random, time-varying amplitude, and there are not enough samples to estimate the signal due to the non-stationarity. A novel technique known as empirical mode decomposition (EMD) overcomes many of the issues related to non-stationarity, but suffers in noisy channels. More recently, a technique known as variational mode decomposition (VMD) was introduced that overcomes the limitations of the other techniques to reconstruct unknown, non-stationary signals; this is termed the single user (SU) VMD algorithm. In this paper, we describe a two-user VMD algorithm that improves performance over the SU algorithm by up to an order of magnitude for the problem of BSS. The two-user VMD only assumes that an estimate of the power levels of the two users may be obtained. Performance is most improved when the powers of the two signals are close to equal.</i></p> <p>DOI: doi.org/10.15864/ajse.1301 Seema Sud (The Aerospace Corporation, USA)</p>
8-13	<p><b>IoT-Based Telemetry System of Ultrasonic Measurement for Experimental Study under COVID-19 Situation</b></p> <p><i>As an epidemic situation of COVID-19, social gathering and working activity have limited. The research execution and engineering activities are significantly affected, especially experimental work. This paper proposes the telemetry system concept for experimenting remotely outside the laboratory. The power source of the testing equipment is controlled remotely. The camera observes the execution of the experiment. The measuring equipment can be set remotely. Consequently, the user can operate and monitor the real-time conducting the experiment from the outside. The experiment of velocity profile measurement on the bubbly flow, which is the vital task in fluid engineering, was demonstrated to confirm the concept's applicability. Finally, the experimental result could be obtained successfully under the remoted experiment.</i></p> <p>DOI: doi.org/10.15864/ajse.1302 Wongsakorn Wongsaroj, Naruki Shoji, Hideharu Takahashi and Hiroshige Kikura (Tokyo Institute of Technology Tokyo, Japan)</p>
14-29	<p><b>Consensus and Eigenvalue Estimation in Dynamic Consensus Networks with Strictly Positive Real Systems</b></p> <p><i>The use of consensus networks in networking has received great attention due to its wide array of applications in fields such as robotics, transportation, sensor networking, communication networking, biology, and physics. The focus of this paper is to study a generalization of consensus problems whereby the weights of network edges are no longer static gains, but instead are dynamic systems, leading to the notion of dynamic consensus networks.. Specifically, we consider networks whose nodes are transfer functions (typically integrators) and whose edges are strictly positive real transfer functions representing dynamical systems that couple the nodes. We transform each concept of static graph theory into dynamic terms, out of which a generalized dynamic graph theory naturally emerges. 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. We show that strictly positive realness of the edges is a sufficient condition for dynamic networks to be stable (i.e., to reach consensus). To study the spectral properties of dynamic networks, we introduce the Dynamic Grounded Laplacian matrix, which is used to estimate lower and upper bounds for the real parts of the smallest and largest non-zero eigenvalues of the dynamic Laplacian matrix. These bounds can be used to obtain stability conditions using the Nyquist graphical stability test for undirected dynamic networks controlled using distributed controllers. Numerical simulations are provided to verify the effectiveness of the results.</i></p> <p>DOI: doi.org/10.15864/ajse.1303 Fadel Lashhab (Howard University, Washington DC, USA)</p>
30-39	<p><b>Development of Breast Cancer Detection Model Using Transfer Learning of Residual Neural Network (ResNet-50)</b></p> <p><i>Breast cancer disease is one of the most common and dangerous as well as being considered as the second most common world cause of cancer death in women. However, the early diagnostics and detection can provide a significant chance for correct treatment and survival. One of the most powerful tools that have shown extraordinary and superior results is the deep convolutional neural network. In this work, we propose an accurate and inclusive computational breast cancer diagnosis framework using ResNet-50 convolutional neural network to classify histopathological microscopy images. The proposed model employs transfer learning technique of the powerful ResNet-50 CNN pretrained on ImageNet to train and classify <i>BreakHis</i> dataset into benign or malignant. The simulation results showed that our proposed model achieves exceptional classification accuracy of 99% outperforming other compared models trained on the same dataset. Based on our novel approach, earlier detection to breast cancer as whether it is being benign or malignant can be stimulated and classified and thus save life and efforts.</i></p> <p>DOI: doi.org/10.15864/ajse.1304 Qasem Abu Al-Haija (Tennessee State University, Nashville, TN, USA) and Ghandi F. Manasra (Palestine Polytechnic University, Hebron, Palestine)</p>
40-44	<p><b>Design and Implementation of A Wearable Monitoring System for Alzheimer's Patient and Elders</b></p> <p><i>Advancing digital networking and improving technology offers an opportunity for the health sector to play a crucial role. The technology for interactive and remote patient care is now much more available and affordable. A patient with a severe disease like Alzheimer's should be checked for wellbeing. In terms of a critical patient, the major problem is to go to the hospital for the routine checkup. In regards to vital care, the main challenge is regular screening in the hospital. Based on the capabilities of IoT based technology, a frequent medically impaired patient may resolve their difficulties in consulting a doctor in a regular basis. A wearable device that allows people to get their real-time pulse rate, temperature, ECG and orientation is what this project suggested. This project led to the creation of a prototype health monitoring device. Four health sensors are part of this prototype: MAX30100, LM35, MPU6050, AD8232. All these sensors were integrated into one system using Arduino Nano. In real-time, cloud storage is continuously updated. An android application was created, where the database was accessed and the health parameters were graphically represented. This wearable health monitoring device will resolve the need for hospital admission or routine checkup in case of minor health issues. In order to clarify the functionality of the sensors used, a thorough study of the signal to respond to variability in physical and environmental behaviours was carried out.</i></p> <p>DOI: doi.org/10.15864/ajse.1305 Mohammed Asaf-Ud-Doulah, Md. Tauhidur Rahman, Naymul Bari, Jobayer Ibne Azad, Shantanu Kumar Nath and Sumaiya Umme Hani ( American International University -Bangladesh Dhaka, Bangladesh)</p>



# A Two-User Variational Mode Decomposition Algorithm for Blind Source Separation of Arbitrary Signals

Seema Sud

The Aerospace Corporation, USA

[seema.sud@aero.org](mailto:seema.sud@aero.org)

**Abstract**—Blind source separation (BSS) is a problem wherein two unknown source signals that have been combined in some fashion when collected at a single receiver must be separated. This problem is exacerbated by source signals that are time-varying, i.e. non-stationary, in nature. All commonly used techniques, such as Fourier analysis, wavelet analysis, and adaptive filtering algorithms do not apply. First, there is no training data, the data may not be digital in nature but have a random, time-varying amplitude, and there are not enough samples to estimate the signal due to the non-stationarity. A novel technique known as empirical mode decomposition (EMD) overcomes many of the issues related to non-stationarity, but suffers in noisy channels. More recently, a technique known as variational mode decomposition (VMD) was introduced that overcomes the limitations of the other techniques to reconstruct unknown, non-stationary signals; this is termed the single user (SU) VMD algorithm. In this paper, we describe a two-user VMD algorithm that improves performance over the SU algorithm by up to an order of magnitude for the problem of BSS. The two-user VMD only assumes that an estimate of the power levels of the two users may be obtained. Performance is most improved when the powers of the two signals are close to equal.

**Keywords**—Blind source separation (BSS), empirical model decomposition (EMD), non-stationary, variational mode decomposition (VMD)

## I. INTRODUCTION

Signal estimation is a challenging problem, especially in an unknown environment, with non-stationary signals that are random and lack known training data. Two novel techniques have been developed to perform signal reconstruction under such conditions, far surpassing the performance of conventional methods, such as Fourier analysis, wavelet processing, principal components analysis (PCA), singular value decomposition (SVD), or other signal processing methods that require training or stationarity. The first method, called empirical mode decomposition (EMD), constructs a signal using a series of intrinsic mode decompositions, by sequentially extracting frequency components. This is done by an averaging process of the envelopes constructed by the local minima and maxima of the signals [6]. The second method, termed variational mode decomposition (VMD) improves upon the EMD, which suffers if errors occur due to noise or sampling rates [2]. VMD forms a signal from a set of modes, where each mode is selected to be narrowband about some center frequency, which is computed along with the mode [2].

In [2], the authors demonstrated superior performance of VMD compared to EMD in reconstructing a signal in the

presence of noise. In [1], VMD was applied to separation of a sinusoidal signal from speech, using PCA also, but the application was limited to this single use case. The issue of selection of the number of modes  $K$  in the reconstruction was not addressed in [2]. However, it was addressed more recently in [9], where selection of the number of modes,  $K$ , of the VMD algorithm is discussed for several test cases. Also, [9] applies VMD to the problem of blind source separation (BSS). It applies VMD directly to estimate a stronger signal, termed the interferer, first. Then, the interferer estimate is removed from the collected signal to estimate the other signal, termed the signal-of-interest (SOI). It further discusses a modification to the VMD algorithm, wherein certain modes of the decomposition are removed in estimating the interferer.

This process, termed culling, is based on an estimate of the SOI's amplitude or frequency, so that spurious modes that correspond to the SOI can be removed from the interferer modes to produce better estimates of both signals. This is shown to provide an improvement in MSE of the interferer and hence also of the SOI, which is obtained by subtracting the interferer estimate from the received signal [9]. However, the culling approach suffers when the SOI is much weaker in power than the interferer, because all modes estimate the interferer, making it difficult to cull the SOI. In this paper, we describe a two-user version of the VMD algorithm where the SOI is estimated along with the interferer to address the blind BSS problem for non-stationary signals; this two-user algorithm overcomes the limitations of the culling approach, and also performs well when the two signals are at or near equal power.

An outline of the paper is as follows: Section II describes the signal model and problem. Section III discusses conventional techniques that could be applied to BSS, including the original, single user (SU) VMD algorithm introduced in [2] for signal reconstruction and applied in [9] to BSS. We compare also to the culling algorithm in [9]. We also describe more traditional methods of singular value decomposition (SVD) and empirical mode decomposition (EMD). Section IV describes the proposed two-user method. Section V presents simulation results comparing performance of all the algorithms. Finally, conclusions and remarks on further work are given in Section VI.



## II. SIGNAL MODEL

We have a received signal, modeled as

$$y(t) = A_x x(t) + I(t) + \sigma_n n(t), \quad (1)$$

where  $x(t)$  is the signal-of-interest (SOI),  $I(t)$  is an interferer, and  $n(t)$  is noise, modeled as additive white Gaussian noise (AWGN). The goal is to jointly estimate both the SOI and the interferer in the presence of the noise, hence this is a BSS problem. Without loss of generality, we assume that the amplitude of  $I(t)$  is unity, and we vary the amplitude of the SOI,  $A_x$ , alone. We set the SOI amplitude to achieve a particular SOI to interferer power ratio, denoted as a carrier-to-interference (CIR), in dB.  $A_x$  is computed from CIR using

$$A_x = \sqrt{10^{CIR/10}}. \quad (2)$$

Without loss of generality, we assume that  $A_x < 1$ , meaning the SOI is weaker in power than the interferer; we limit our attention, therefore, to the case where  $-10 < CIR < 0$  dB, resulting in  $0.316 < A_x < 1$ . The noise standard deviation,  $\sigma_n$  is computed from the desired carrier-to-noise ratio (C/N), also in dB, written as

$$\sigma_n = \sqrt{10^{-(C/N)/10} \cdot 0.5}. \quad (3)$$

We restrict attention to typical scenarios where  $0 < C/N < 15$  dB, which results in  $0.707 < \sigma_n < 4$ . The goal is to extract both the interferer  $I(t)$  and the SOI  $x(t)$  from the received signal  $y(t)$ ; again, we note that these signals are time-varying, and we do not have training data. First, we present some conventional methods for separating the signals, including the single user (SU) VMD algorithm, SVD, and EMD; we then discuss the proposed two-user VMD.

## III. CONVENTIONAL METHODS

### A. Variational Mode Decomposition

The VMD was originally proposed in [2], as a way to reconstruct a non-stationary signal  $x(t)$  as a series of  $K$  modes, wherein the  $k^{\text{th}}$  mode  $u_k(\omega_k)$ , in the frequency domain, is constructed using a narrowband Wiener filter about a chosen center frequency,  $\omega_k$ . Modes are selected based on minimization of the bandwidth about  $\omega_k$ , resulting in a constrained problem written as

$$\begin{aligned} \arg \min_{u_k, \omega_k} & \|\partial_t [(\delta(t) + \frac{j}{\pi t}) * u_k(t)] e^{-j\omega_k t}\|^2 \\ \text{s.t.} & \sum_{k=1}^K u_k(t) = x(t), \end{aligned} \quad (4)$$

where  $k = 1, 2, \dots, K$ . The term in parenthesis is the Hilbert Transform of  $u_k(t)$ ,  $*$  denotes convolution, the exponential term translates the component  $\omega_k$  to baseband, and the partial derivative  $\partial_t$  is the gradient, used to estimate the bandwidth. Finally, the norm of the entire term is computed using  $\|\cdot\|^2$ . This problem is solved by applying two additional constraints.

The first constraint is a parameter,  $\alpha$ , used to adjust the weight of the first term in Eq. (4) depending on the strength of the noise; larger noise results in a smaller  $\alpha$ . The second constraint is a Lagrange multiplier ( $\lambda$ ) used to adhere strictly to the constraint dictated by the second term in Eq. (4). The result is the new problem formulation

$$\begin{aligned} \arg \min_{u_k, \omega_k} & \alpha \|\partial_t [(\delta(t) + \frac{j}{\pi t}) * u_k(t)] e^{-j\omega_k t}\|^2 \\ & + \|x(t) - \sum_{k=1}^K u_k(t) + \frac{\lambda(t)}{2}\|^2. \end{aligned} \quad (5)$$

After some mathematical manipulation (details provided in [2]), the iterative solution for computing the modes is as follows: initialize  $n = 0$ ,  $u_k^0 = \lambda^0 = \omega_k^0 = 0$ , and compute the modes and frequencies as

$$u_k^{n+1}(\omega) = \frac{\hat{X}(\omega) - \sum_{j < k} \hat{u}_j^{n+1}(\omega) - \sum_{j > k} \hat{u}_j^n(\omega) + \frac{\lambda^n(\omega)}{2}}{1 + 2\alpha(\omega - \omega_k^n)^2}, \quad (6)$$

$$\omega_k^{n+1} = \frac{\int_0^\infty \omega |\hat{u}_k^{n+1}(\omega)|^2 d\omega}{\int_0^\infty |\hat{u}_k^{n+1}(\omega)|^2 d\omega}, \quad (7)$$

where

$$\lambda^{n+1}(t) = \lambda^n(t) + \tau [x(t) - \sum_{k=1}^K u_k^{n+1}(t)], \quad (8)$$

and where  $\tau$  is a time step constant (may be zero). We iterate from  $n = 1$  to  $N-1$  until the modes and frequencies converge.

We input  $y(t)$  to the VMD algorithm, and the output modes are used to compute the estimate of  $I(t)$ , denoted  $\hat{I}(t)$  as

$$\hat{I}(t) = \sum_{k=1}^K u_k(t). \quad (9)$$

This equation is based on the assumption that the interferer power is higher, hence it will be estimated first by VMD. We then estimate the SOI as

$$\hat{x}(t) = y(t) - \hat{I}(t). \quad (10)$$

Finally, performance is determined by computing the mean-square error (MSE) between the true signals and their estimates, using

$$MSE_{I(t)} = \overline{((I(t) - \hat{I}(t))^2)}, \quad (11)$$

and

$$MSE_{x(t)} = \overline{((x(t) - \hat{x}(t))^2)}, \quad (12)$$



Note that an issue with the VMD algorithm is in selection of the number of modes,  $K$ , similar to selection of the number of signals when using PCA. If under selected, the interferer is not fully captured and errors in estimating the SOI occur. If overestimated, then partial SOI cancellation occurs, again resulting in errors in estimating the SOI. In [9], it is discussed how  $K$  can be selected based on the types of signals expected.

When the two signals are near or at equal power, this algorithm will suffer as it does not take into account the presence of the second signal, the SOI; hence, we call this the single user (SU) VMD algorithm. The reconstruction error that results due to the lack of knowledge of the SOI motivates the need to develop an improved two-user version of this algorithm.

### B. Variational Mode Decomposition with Mode Culling

An improved SU VMD algorithm using mode culling is presented in [9]. This involves an initial estimate of the frequency bands at which a strong SOI component is present. Then, the modes  $u_k$  at those frequencies,  $\omega_k$  are removed from Eq. (9) to improve the interferer estimate. This is written as

$$\hat{I}(t)_{culled} = \sum_{\substack{k=1 \\ k \neq j, \omega_j \in X(\omega)}}^K u_k(t), \tag{13}$$

where we do not include any mode  $j$ , where  $\omega_j$  is estimated to be a frequency of  $X(\omega)$ , and  $X(\omega) = \text{FFT}\{x(t)\}$ . Note that the  $\omega_j$ 's have to be estimated or guessed at, since we do not know  $X(\omega)$ . This can be estimated in practice, e.g. by knowing what bands a particular SOI is in. The SOI is still indirectly obtained from the new interferer estimate using Eq. (10), and MSEs are computed as before from Eqs. (11) and (12).

An alternate way to do the culling is to estimate the amplitude of the SOI, and then remove the  $u_k(t)$ 's whose amplitude is closest, i.e. the  $k^{\text{th}}$  component where  $A_x \approx \max\{u_k(t)\}$ . We can write

$$\hat{I}(t)_{culled} = \sum_{\substack{k=1 \\ k \neq j, u_j(t) \in A_x \approx \max(u_j(t))}}^K u_k(t). \tag{14}$$

Removal of the estimated SOI modes by culling produces a better estimate of the interferer, which in turn results in an improved estimate of the SOI.

### C. Singular Value Decomposition

A technique, which is similar to the well-known method of PCA, is known as singular value decomposition (SVD). It constructs a signal from a subset of its eigenvalues and eigenvectors ([3] and [4]). The idea is that noise is eliminated by choosing a subset that only spans the signal space and does not include the noise space. Suppose we form a received data

matrix  $\mathbf{Y}$  from our received signal  $y(t)$ , which may be formed from many observations of the received signal. Thus,  $\mathbf{Y}$  has dimension  $M \times N$ , where  $M$  is the number of observations of the sampled statistical process, and  $N$  is the number of samples collected in each observation. We can compute the SVD of the matrix  $\mathbf{Y}$  as [8]

$$\mathbf{Y} = \mathbf{U}\mathbf{\Lambda}\mathbf{V}^H, \tag{15}$$

where  $\mathbf{U}$ ,  $\mathbf{\Lambda}$ , and  $\mathbf{V}$  are  $M \times N$  left singular,  $N \times N$  diagonal singular, and  $N \times N$  right singular matrices of  $\mathbf{Y}$ , respectively. Matrix  $\mathbf{V}$  contains the eigenvectors of

$$\mathbf{R}_{\mathbf{Y}\mathbf{Y}} = \mathbf{Y}^H\mathbf{Y}, \tag{16}$$

where  $\mathbf{R}_{\mathbf{Y}\mathbf{Y}}$  has been defined as the covariance matrix of  $\mathbf{Y}$ . The column vectors of  $\mathbf{U}$  form an orthonormal basis of  $\mathbf{R}_{\mathbf{Y}\mathbf{Y}}$ . To estimate the strong interferer, we form a subset  $\mathbf{D}$ ,  $D < N$ , of these vectors, e.g.

$$\mathbf{Y}_D = \sum_{i=1}^D \lambda_i \mathbf{U}_i \mathbf{V}_i^H, \tag{17}$$

and then compute the estimate of the interferer as

$$\hat{I}(t) = \mathbf{Y}_{D,i}, \tag{18}$$

where  $\lambda = \text{diag}(\mathbf{\Lambda})$ , and  $\lambda_i$  is the  $i^{\text{th}}$  element of the vector  $\lambda$ ; similarly,  $\mathbf{Y}_{D,i}$ ,  $\mathbf{U}_i$ , and  $\mathbf{V}_i$  denote the  $i^{\text{th}}$  column of matrices  $\mathbf{Y}_D$ ,  $\mathbf{U}$  and  $\mathbf{V}$ , respectively. By choosing a set of  $D$  basis vectors that correspond to the  $D$  largest  $\lambda_i$ 's, a reduced rank subspace is formed. Since we typically do not know a priori what value to select for  $D$ , we simply choose it to be the number of non-zero eigenvalues. We then estimate the SOI using Eq. (10), and the MSEs using Eqs. (11) and (12), as with the VMD. However, while this algorithm generally performs slightly better than PC because it is not as sensitive to the selection of  $D$  being less than the rank of the subspace, it suffers from computational complexity issues [5]. More importantly, the SVD metric is also dependent on eigen-decomposition and estimation of the covariance matrix  $\mathbf{R}_{\mathbf{Y}\mathbf{Y}}$ . Furthermore, it does not lend itself easily to handling a two-user BSS problem.

### D. Empirical Mode Decomposition

The EMD decomposes a signal into what is called intrinsic mode functions (IMFs). These are functions that take the form [2]

$$u_k(t) = A_k(t) \cdot \cos(\phi_k(t)), \tag{19}$$

where  $A_k(t)$  and  $\phi_k(t)$  are time-varying amplitude and phase components. The IMFs are computed by a repetitive sifting



process, until only a residual component remains. This is described in [6] and [10] and summarized next:

First, we find local maxima and minima; we use these to compute the upper and lower envelopes of the signal, denoted  $u(t)$  and  $l(t)$ , respectively. Initializing index  $k = 1$ , we compute the mean as

$$m_k(t) = \frac{u(t) + l(t)}{2}. \quad (20)$$

We compute the sifted IMF,

$$h_k(t) = y(t) - m_k(t). \quad (21)$$

We use this to update the signal

$$y_k(t) = h_k(t) \quad (22)$$

Now we increment  $k$ , compute new envelopes of the updated signal, and repeat the calculations in Eqs. (20)-(22) until all IMFs are computed. This completes the first part of the algorithm. In the second part, we set

$$c_1(t) = h_k(t), \quad (23)$$

the last IMF, which represents the highest frequency component present in the signal. We compute a residue

$$r_1(t) = y(t) - c_1(t). \quad (24)$$

We update the signal using the computed residue

$$y_k(t) = r_1(t) \quad (25)$$

Next, we repeat and compute the residues for the remaining IMFs

$$r_j(t) = r_{j-1}(t) - c_j(t), \quad (26)$$

for  $j = 2, 3, \dots, n$  until  $r_n(t)$  is a monotonic function. Finally, the reconstructed signal, assumed once again to be the interferer, can be written as

$$\hat{I}(t) = \sum_{j=1}^n c_j(t) + r_n(t). \quad (27)$$

Note that  $r_n(t)$  represents the mean of  $y(t)$ . With the interferer estimate having been computed, the SOI and MSEs are computed as with the VMD and SVD algorithms, using Eqs. (10)-(12). As with the preceding algorithms, this algorithm is not suited to handle multiple users. So, we turn our attention back to the VMD and develop a two-user modification of the SU VMD.

#### IV. TWO-USER VARIATIONAL MODE DECOMPOSITION (VMD) ALGORITHM

Referring to Eqs. (1) and (4), the two-user constrained problem may be written as

$$\begin{aligned} \underset{u_{k,1}, u_{k,2}}{\operatorname{arg\,min}} \quad & \|\partial_t [(\delta(t) + \frac{j}{\pi t}) * (u_{k,1}(t) + \hat{A}_x u_{k,2}(t))] e^{-j\omega_k t}\|^2, \\ \text{s.t.} \quad & \sum_{k=1}^K [u_{k,1}(t) + \hat{A}_x u_{k,2}(t)] = x(t), \end{aligned} \quad (28)$$

where  $\hat{A}_x$  is an estimate of the amplitude of the SOI, and we introduce a second set of modes for the weaker SOI,  $u_{k,2}(t)$ . The modes  $u_{k,1}(t)$  are for the stronger interferer. Applying the same constraints as for the single user (SU) case, we obtain

$$\begin{aligned} \underset{u_{k,1}, u_{k,2}}{\operatorname{arg\,min}} \quad & \alpha \|\partial_t [(\delta(t) + \frac{j}{\pi t}) * (u_{k,1}(t) + \hat{A}_x u_{k,2}(t))] e^{-j\omega_k t}\|^2 \\ & + \|x(t) - \sum_{k=1}^K (u_{k,1}(t) + \hat{A}_x u_{k,2}(t)) + \frac{\lambda(t)}{2}\|^2. \end{aligned} \quad (29)$$

Performing similar mathematical manipulations as in [2] for the SU VMD (details shown in the Appendix), we obtain the modes for both users as

$$u_{k,1}^{n+1}(\omega) = \frac{\hat{X}(\omega) - \sum_{j<k} \hat{u}_{j,1}^{n+1}(\omega) - \sum_{j>k} \hat{u}_{j,1}^n(\omega) + \frac{\lambda^n(\omega)}{2}}{1 + 2\alpha(\omega - \omega_k^n)^2}, \quad (30)$$

$$u_{k,2}^{n+1}(\omega) = \frac{\hat{X}(\omega) - \sum_{j<k} \hat{A}_x \hat{u}_{j,2}^{n+1}(\omega) - \sum_{j>k} \hat{A}_x \hat{u}_{j,2}^n(\omega) + \frac{\lambda^n(\omega)}{2}}{\hat{A}_x + 2\alpha(\omega - \omega_k^n)^2}, \quad (31)$$

where  $\omega_k$  is calculated as in Eq. (7), and

$$\lambda^{n+1}(t) = \lambda^n(t) + \tau [x(t) - \sum_{k=1}^K (u_{k,1}^{n+1}(t) - \hat{A}_k u_{k,2}^{n+1}(t))], \quad (32)$$

Note the presence of the term  $\hat{A}_x$ , the estimate of the SOI amplitude, indicating that this term is required to compute the modes for the SOI. Based on the above equations, we obtain the estimates of the interferer and SOI with the two user algorithm as

$$\hat{I}_{Two-User}(t) = \sum_{k=1}^K u_{k,1}(t), \quad (33)$$

and

$$\hat{x}_{Two-User}(t) = \sum_{k=1}^K u_{k,2}(t). \quad (34)$$

We can therefore compute the MSEs for the two-user algorithm as



$$MSE_{I_{T_{wo-U_{ser}}}(t)} = \overline{((I(t) - \hat{I}_{T_{wo-U_{ser}}}(t))^2)}, \quad (35)$$

and

$$MSE_{x_{T_{wo-U_{ser}}}(t)} = \overline{((x(t) - \hat{x}_{T_{wo-U_{ser}}}(t))^2)}, \quad (36)$$

In the next section, we perform simulations to compare the performance of the SU VMD and two-user VMD algorithms, as well as the conventional SVD and EMD algorithms, comparing the MSEs and plotting the estimated signals.

### V. SIMULATIONS

We assume the SOI is a chirp signal that takes the form

$$x(t) = e^{j2\pi f_d t} e^{j\pi\beta t^2}, \quad (37)$$

where  $f_d$  is the initial frequency of the chirp, and  $\beta$  is the rate of change in the frequency in Hz/sec. We further assume the interferer is a speech signal, modeled as

$$I(t) = \sum_{i=1}^{N_s} s_i(t), \quad (38)$$

where

$$s_i(t) = a_i(t) \cos(2\pi[f_{c_i} + \int_0^t f_i(\tau) d\tau] + \theta); \quad (39)$$

and where  $a_i(t)$  is the amplitude, which will be time-varying, i.e. non-stationary,  $f_{c_i}$  is the center frequency of the signal,  $f_i(t)$  is the frequency modulation, and  $N_s$  is the number of signals combined to form the speech signal [7]. For simulation purposes, we let  $f_d = 40$  kHz,  $\beta = 10$  kHz/s, and  $1 \text{ Hz} < f_c < 1$  kHz. The interferer and SOI are shown in Fig. 1. We now study the performance of the two-user VMD algorithm and compare it to the others.

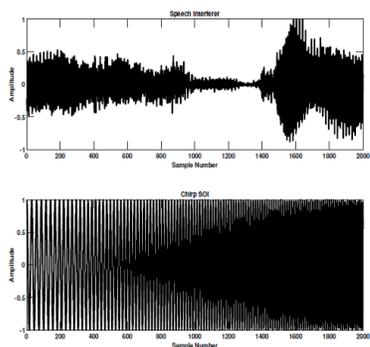


Fig. 1. Speech Interferer and Chirp SOI;  $f_d = 40$  kHz,  $\beta = 10$  kHz/s, and  $1 \text{ Hz} < f_c < 1$  kHz

Note that the two-user VMD requires estimation of the amplitude of the SOI,  $A_x$ . Hence, it is important to understand how its performance will degrade if there are errors in the estimate.

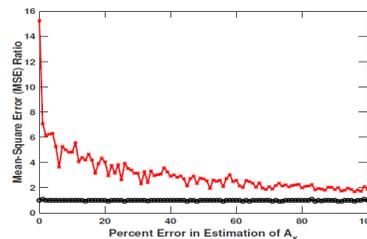


Fig. 2. MSE Ratio of Two-User VMD to Single User (SU) VMD Algorithms vs. Percent Error in Estimation of  $A_x$ ; Speech Interferer and Chirp SOI;  $C/N = 10$  dB,  $CIR = 0$  dB,  $K_1 = 6$ ,  $K_2 = 3$

In Fig. 2. we show the MSE ratio of the two-user VMD algorithm to the SU VMD algorithm as a function of percentage error in  $A_x$ . In this example,  $C/N = 10$  dB, and  $CIR = 0$  dB, so  $A_x = 1$ . From this plot, we see that up to about 20% error in the amplitude can occur before the two-user VMD algorithm degrades to the point where the SU VMD algorithm would be just as effective.

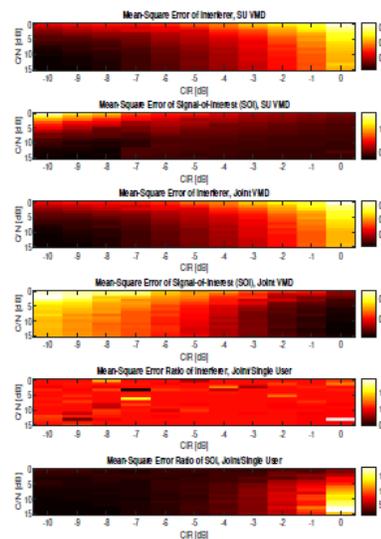


Fig. 3. MSEs of Single User (SU) VMD Algorithm, Two-User VMD Algorithm, and Ratio of Two-User to SU Algorithms; Speech Interferer and Chirp SOI;  $K_1 = 6$ ,  $K_2 = 3$

Fig. 3. shows the mean-square error (MSE) of the interferer and SOI for the SU VMD algorithm in the top two plots and the two-user VMD in the next two plots. The ratio of MSEs for the two-user and SU algorithms is shown in the last two plots. Notice that the two-user VMD algorithm does not improve the MSE of the interferer, which is the stronger signal. However, it significantly improves the MSEs of the lower powered SOI. Improvement is significant when  $-5 < CIR < 0$  dB, and most significant when CIR approaches 0 dB. This is due to the two-user VMD algorithm taking into account the SOI, and its amplitude, to estimate it, whereas the SU VMD does not. This is the range of CIR where most improvement is needed, because when CIR is near -10 dB, the SOI is weak, so the SU VMD is sufficient.

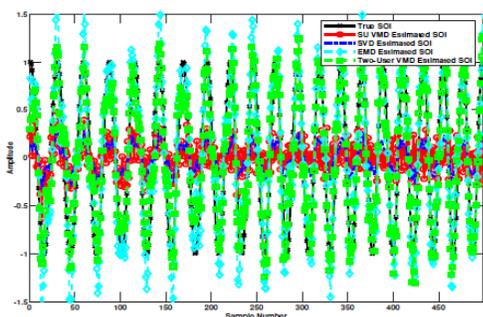


Fig. 4. Estimate of Chirp SOI; Speech Interferer and Chirp SOI; Comparison of SU VMD, SVD, EMD, and Two-User VMD; C/N = 10 dB, CIR = 0 dB,  $K_1 = 6$ ,  $K_2 = 3$

Fig. 4. plots the estimate of the SOI for the SU VMD, SVD, EMD, and the proposed two-user VMD algorithms, comparing them to the true chirp SOI, in the presence of the random speech interferer. We show only the first 500 samples to make it more visible. We have set C/N = 10 dB and CIR = 0 dB in this example. Because the SOI power is equal to the interferer power, we would expect the SU algorithm to perform poorly, since it is operating without knowledge of the presence of a second signal. The SVD algorithm also fails, because of a lack of knowledge of the presence of a second signal. The two-user VMD algorithm, however, is able to extract the SOI successfully. The MSEs are calculated from the samples as  $MSE_{SU\ VMD} = 0.44$ ,  $MSE_{SVD} = 0.60$ ,  $MSE_{EMD} = 0.08$ , and  $MSE_{two-user\ VMD} = 0.03$ , reflecting the improved two-user VMD algorithm performance. This is an improvement of 11.1, 12.5, and 3.7 dB over the SU VMD, SVD, and EMD, respectively. The EMD algorithm performs well too, but not as well as the two-user VMD. While EMD is able to estimate the frequency structure of the SOI nearly as well as the two-user VMD, it has errors in estimating the SOI amplitude, due to the interferer. This results in fluctuations that increase its MSE.

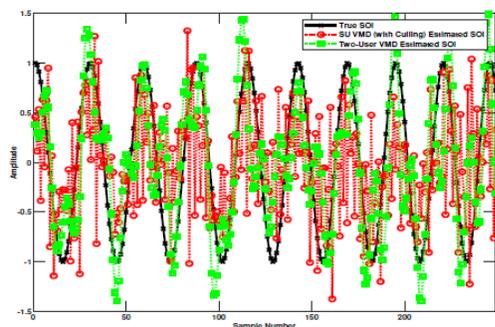


Fig. 5. Estimate of Chirp SOI; Speech Interferer and Chirp SOI; Comparison of Culled SU VMD and Two-User VMD; C/N = 10 dB, CIR = -10 dB,  $K_1 = 6$ ,  $K_2 = 3$

Fig. 5 plots the SOI estimate for C/N = 10 dB and CIR = -10 dB to compare the improvement of the two-user algorithm vs. the culling VMD algorithm presented in [9]. Here, the last two modes are culled, per [9]. As mentioned above and discussed in [9], the culling VMD approach suffers at low CIR due to the inability to estimate the weak SOI modes and remove them from the interferer estimate. Here, we see that

the two-user VMD approach works better, because it directly uses the SOI's amplitude estimate to compute the SOI from the received signal. MSE estimates are  $MSE_{SU\ VMD, Culling} = 0.59$  and  $MSE_{two-user\ VMD} = 0.21$ , an improvement of 4.5 dB. The SVD and EMD algorithms fail, producing MSEs > 0.8, so these are not shown.

## VI. CONCLUSION

This paper presents a two-user, blind source separation (BSS) algorithm to simultaneously estimate two unknown signals in a time-varying environment. This is done without any knowledge of the signals, i.e. no known sequences or training data, other than an estimate of the relative amplitudes of the two signals. The proposed method is a two-user version of the previously introduced variational mode decomposition (VMD) algorithm, where the modes that reconstruct the signals are computed using their amplitude estimates. Performance results showed an order of magnitude improvement of the two-user VMD algorithm over the single user (SU) case, both with and without mode culling, wherein the stronger signal is estimated with the VMD and the weaker one is obtained by subtracting this estimate from the received signal. We also demonstrated improvement in performance of the two-user VMD over conventional single user (SU) methods including singular value decomposition (SVD) and empirical mode decomposition (EMD). Improvement of the two-user VMD is most pronounced when the powers of the two signals are close, i.e. the CIR is near 0 dB; this is where all the other algorithms suffer because they do not take into account any knowledge of a second signal. We also analyze performance degradation as a function of errors in estimating the signal amplitudes, determining that up to a 20% error is tolerable before performance degrades to where the other algorithms perform just as well. Future work includes development of a truly joint algorithm where both signals may be estimated together, although this is expected to pose significant challenges.

## VII. APPENDIX: DETAILED DERIVATION OF SECOND USER MODES

To derive the modes, we start with the constrained problem from Eq. (29), repeated here for convenience as

$$\arg \min_{\substack{u_{k,1}, u_{k,2} \\ \omega_k}} \alpha \left\| \partial_t \left[ \left( \delta(t) + \frac{j}{\pi t} \right) * (u_{k,1}(t) + \hat{A}_x u_{k,2}(t)) \right] e^{-j\omega_k t} \right\|^2 + \|x(t) - \sum_{k=1}^K (u_{k,1}(t) + \hat{A}_x u_{k,2}(t)) + \frac{\lambda(t)}{2}\|^2. \quad (40)$$

We first exploit Parseval's theorem, which states that the magnitude squared of a function integrated over all time is equivalent to its magnitude squared integrated over all frequency, to translate the above to the frequency domain,



$$\begin{aligned} & \underset{\substack{u_{k,1}, u_{k,2} \\ \omega_k}}{\arg \min} \alpha \|j\omega[(1 + \text{sgn}(\omega + \omega_k)) \cdot \\ & (u_{k,1}(\omega + \omega_k) + \hat{A}_x u_{k,2}(\omega + \omega_k))] \|^2 \\ & + \|\hat{X}(\omega) - \sum_{k=1}^K (u_{k,1}(\omega) + \hat{A}_x u_{k,2}(\omega)) + \frac{\lambda(\omega)}{2}\|^2. \end{aligned} \quad (41)$$

Following [2], replace  $\omega$  with  $\omega - \omega_k$  in the first term of the above equation to get

$$\begin{aligned} & \underset{\substack{u_{k,1}, u_{k,2} \\ \omega_k}}{\arg \min} \alpha \|j(\omega - \omega_k)[(1 + \text{sgn}(\omega)) \cdot (u_{k,1}(\omega) + \hat{A}_x u_{k,2}(\omega))] \|^2 \\ & + \|\hat{X}(\omega) - \sum_{k=1}^K (u_{k,1}(\omega) + \hat{A}_x u_{k,2}(\omega)) + \frac{\lambda(\omega)}{2}\|^2. \end{aligned} \quad (42)$$

Note that we do not have to do this with the second term as all the terms use the variable  $\omega$ , so a change of terms does nothing. Next, we exploit the fact that real signals in the frequency domain are symmetric about the zero frequency, so we can write each term as twice the integral of the term over positive frequencies only. This further allows us to simplify, since  $\text{sgn}(\omega) = 1$  for positive  $\omega$ . We therefore obtain

$$\begin{aligned} & \underset{\substack{u_{k,1}, u_{k,2} \\ \omega_k}}{\arg \min} \int_0^\infty \{4\alpha(\omega - \omega_k)^2 |u_{k,1}(\omega) + \hat{A}_x u_{k,2}(\omega)|^2 \\ & + 2|\hat{X}(\omega) - \sum_{k=1}^K (u_{k,1}(\omega) + \hat{A}_x u_{k,2}(\omega)) + \frac{\lambda(\omega)}{2}|^2\} d\omega. \end{aligned} \quad (43)$$

Now, treating the two signals as independent from one another, we solve the minimization problem with respect to the second user first, taking the partial derivative  $\partial/\partial u_{k,2}$  and setting the result to zero. This gives

$$8\alpha(\omega - \omega_k)^2 \hat{A}_x u_{k,2}(\omega) - 4\hat{A}_x (\hat{X}(\omega) + \hat{A}_x \sum_{k=1}^K u_{k,2} + \frac{\lambda(\omega)}{2}) = 0. \quad (44)$$

Separating the  $k$ th term in the summation of the second term, combining it with that of the first term, i.e.  $u_{k,2}(\omega)$ , and including the iteration index  $n$  to account for the most recent updates, we obtain

$$\begin{aligned} & (2\alpha(\omega - \omega_k)^2 + \hat{A}_x) u_{k,2}^{n+1}(\omega) \\ & - (\hat{X}(\omega) + \sum_{j<k} \hat{A}_x u_{j,2}^{n+1}(\omega) + \sum_{j>k} \hat{A}_x u_{j,2}^n(\omega) + \frac{\lambda(\omega)}{2}) = 0. \end{aligned} \quad (45)$$

Solving for  $u_{k,2}^{n+1}(\omega)$ , we obtain the final result given in Eq. (31), i.e.

$$u_{k,2}^{n+1}(\omega) = \frac{\hat{X}(\omega) - \sum_{j<k} \hat{A}_x \hat{u}_{j,2}^{n+1}(\omega) - \sum_{j>k} \hat{A}_x \hat{u}_{j,2}^n(\omega) + \frac{\lambda(\omega)}{2}}{\hat{A}_x + 2\alpha(\omega - \omega_k)^2}, \quad (46)$$

The modes for the stronger interferer may be obtained similarly, as first described in [2]. Here, we can obtain the modes for the first user directly from the modes of the second user, given in Eq. (46). To do this, recall that the amplitude of the first user, the interferer, was set to unity, so we can simply set the variable  $\hat{A}_x$  to unity, giving

$$u_{k,1}^{n+1}(\omega) = \frac{\hat{X}(\omega) - \sum_{j<k} \hat{u}_{j,1}^{n+1}(\omega) - \sum_{j>k} \hat{u}_{j,1}^n(\omega) + \frac{\lambda(\omega)}{2}}{1 + 2\alpha(\omega - \omega_k)^2}, \quad (47)$$

which of course, are the same modes gives in the single user algorithm in [2]. Note that if the interferer had non-unity amplitude, we could instead replace  $\hat{A}_x$  with a new variable representing the amplitude estimate of the interferer,  $\hat{A}_I$  to obtain its modes.

### VIII. ACKNOWLEDGMENT

The author thanks The Aerospace Corporation for funding this work.

### REFERENCES

- [1] Dey, P., Satija, U., and Ramkumar, B., "Single Channel Blind Source Separation Based on Variational Mode Decomposition and PCA," *Proc. Annual IEEE India Conference (INDICON)*, New Delhi, India, pp. 1-5, 2015.
- [2] Dragomiretskiy, K., and Zosso, D., "Variational Mode Decomposition", *IEEE Trans. on Signal Proc.*, Vol. 62, No. 3, pp. 531-544, 2014.
- [3] Goldstein, J.S., *Optimal Reduced Rank Statistical Signal Processing, Detection, and Estimation Theory*, Ph.D. Thesis, Dept. of Electrical Engineering, University of Southern California, Los Angeles, CA, Dec. 1997.
- [4] Goldstein, J.S., and Reed, I.S., "Reduced Rank Adaptive Filtering", *IEEE Trans. on Signal Proc.*, Vol. 45, No. 2, pp. 492-496, Feb. 1997.
- [5] Honig, M.L., "A Comparison of Subspace Adaptive Filtering Techniques for DS-CDMA Interference Suppression", *Proc. of IEEE MILCOM*, Vol. 2, pp. 836-840, Monterey, CA, Nov. 1997.
- [6] Huang, N.E., Shen, Z., Long, S.R., Wu M.C., Shih, H. H., Zheng, Q. Yen, N.-C., Tung, C. C., and Liu, H. H., "The Empirical Mode Decomposition and the Hilbert Spectrum for Nonlinear and Non-stationary Time Series Analysis", *Proc. Royal Soc. A: Math., Phys. Eng. Sci.*, Vol. 454, No. 1971, pp. 903-995, Mar. 1998.
- [7] Lai, E., "Practical Digital Signal Processing", Elsevier Science and Technology, Oxford, UK, Jan. 2004.
- [8] Strang, G., "Linear Algebra and Its Applications", Harcourt Brace Jovanovich, San Diego, CA, 1988.
- [9] Sud, S., "Blind, Non-stationary Source Separation Using Variational Mode Decomposition with Mode Culling". *Signal Processing: An International Journal (SPIJ)*, Vol. 13, No. 2, pp. 11-20, Aug. 2020.
- [10] Wu, X., Fu, R., Hu, S., and Xu, C., "Single-Channel Blind Signal Separation Based on Empirical Mode Decomposition and Fast Independent Component Analysis", In: *Qian Z., Cao L., Su W., Wang T., and Yang H. (eds) Recent Advances in Computer Science and Information Engineering. Lecture Notes in Electrical Engineering*, Vol. 128, Springer, Berlin, Heidelberg, 2012.



# IoT-Based Telemetry System of Ultrasonic Measurement for Experimental Study under COVID-19 Situation

Wongsakorn Wongsaroj

*Department of Mechanical Engineering  
Tokyo Institute of Technology  
Tokyo, Japan  
wongsaroj.w.aa@m.titech.ac.jp*

NarukiShoji

*Department of Mechanical Engineering  
Tokyo Institute of Technology  
Tokyo, Japan  
shoji.n.aa@m.titech.ac.jp*

Hideharu Takahashi

*Laboratory for Advanced Nuclear Energy, Institute of Innovative Research  
Tokyo Institute of Technology  
Tokyo, Japan  
htakahashi@lane.iir.titech.ac.jp*

Hiroshige Kikura

*Laboratory for Advanced Nuclear Energy, Institute of Innovative Research  
Tokyo Institute of Technology  
Tokyo, Japan  
kikura@lane.iir.titech.ac.jp*

**Abstract**— As an epidemic situation of COVID-19, social gathering and working activity have limited. The research execution and engineering activities are significantly affected, especially experimental work. This paper proposes the telemetry system concept for experimenting remotely outside the laboratory. The power source of the testing equipment is controlled remotely. The camera observes the execution of the experiment. The measuring equipment can be set remotely. Consequently, the user can operate and monitor the real-time conducting the experiment from the outside. The experiment of velocity profile measurement on the bubbly flow, which is the vital task in fluid engineering, was demonstrated to confirm the concept's applicability. Finally, the experimental result could be obtained successfully under the remoted experiment.

**Keywords**—*Telemetry, COVID-19, Experiment, Ultrasonic,*

## I. INTRODUCTION

As the high demand for electrical energy consumption in the world, the power plant is imperative to be built to respond to the requirement. One of the electrical power generations base on the utilizing of steam-power. This concept is placed on coal, gas combine cycle, and nuclear power plant. However, these power plants must be operated safely without the accident. The operation is compulsory under the safety aspect. The crucial part of the plant operation according to safety standard and sustainability is the engineering design. The thermal-hydraulic and fluid mechanic behaviors of the water-steam cycle in the steam generation's boiling unit are the main sections on the accident's occurrence. It is necessary to be predicted and optimized accurately by the numerical model [1, 2]. These characteristics precisely determine the geometry, material, and other specifications, which affect the unit's safety criteria. To confirm the reliability of the model indispensably,

the model validation by the experimental data is an important task. Hence, the experimental investigation on this behavior is needed [3].

Presently, due to the COVID-19 situation [4, 5], the outbreak of COVID-19 had spread in more than 200 countries. Approximately 146,198 people had died. Several measures have been used to deal with this situation, such as complete lockdown, suspension of transportation, etc. The social gathering, education, tourism, and working activity in the workplace have been prohibited or minimized. Significantly, it influences the action on the experimental task in the facility placed on the laboratory or company.

Therefore, the completion of the experiment work is a delay. Then, the engineering design and project execution as well are belated. Hence, to solve this problem, the conducting of experimental work remotely from other places outside the workplace has been considered.

The internet of Things (IoT) [6] is a system of interrelated computing devices, mechanical and digital machines provided with unique identifiers (UIDs). It can transfer data over a network without requiring human-to-human or human-to-computer interaction. As this technology, it opens the opportunity for smart activity, automation, and so on in several sectors, as shown in figure 1 such as industrial[7, 8], medical system [9], Agrolological [10, 11], education [12], transportation [13] or even in for smart home [14]. Therefore, this platform has a potential significantly to apply as a telemetry system for the execution of experimental work remotely under pandemic disease circumstance as the COVID-19.

This study presents the telemetry concept to hand on the experimental activity remotely. The experiment can be conducted anywhere on the internet is available. The demonstration is focused on the experiment of the velocity distribution measurement in the bubbly flow. It is one of the critical parameters of fluid mechanics investigation. The investigation is conducted by employing the ultrasonic velocity profiler measurement system, a powerful tool for obtaining the fluid's velocity profile.

## II. MEASUREMENT DEVICE AND TELEMETRY SYSTEM

### A. Ultrasonic velocity profiler

The UVP measurement [15-17] is an ultrasound-reflection technique that can obtain the liquid's velocity profile. Figure 2 illustrates the UVP configuration, ultrasonic pulse-echo, and velocity profile reconstruction. An ultrasonic pulse is transmitted from the transducer along the measurement line to the liquid. The same transducer derives the echo signal

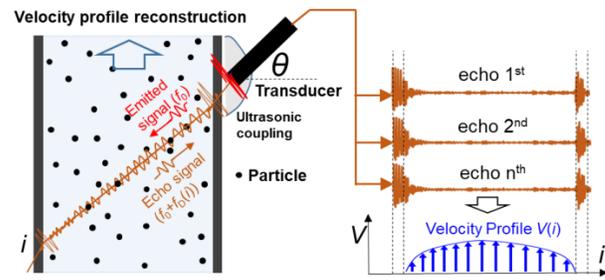


Fig. 2. The UVP measurement configuration.

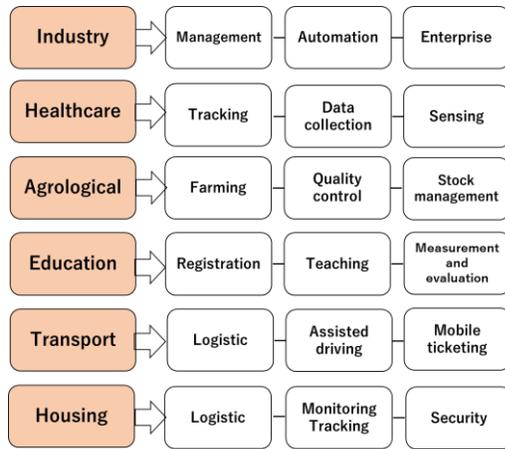


Fig. 1. Applications of IoT and relevant major scenarios.

reflected from moving reflectors such as small particles dispersed in the liquid. As the ultrasonic wave is emitted repeatedly, the echo signals are obtained sequentially. Doppler signal influenced by the velocity of a moving particle can be demodulated from the echo signals shown in figure 3. The Doppler frequency  $f_D(i)$  directly relates to the velocity of the particle ( $i$  is position or channel). Hence, the velocity of the particle at that position  $V(i)$  can be computed as

$$V(i) = \frac{cf_D(i)}{2f_0 \sin\theta} \quad (1)$$

where  $f_0$  is the basic frequency, and  $\theta$  is the incident angle. If the Stokes number on the relation between small particles and liquid  $< 0.1$ , the particle will closely follow the liquid streamline. Then, several particles dispersed in the liquid. Consequently, the velocity profile of the liquid can be obtained.

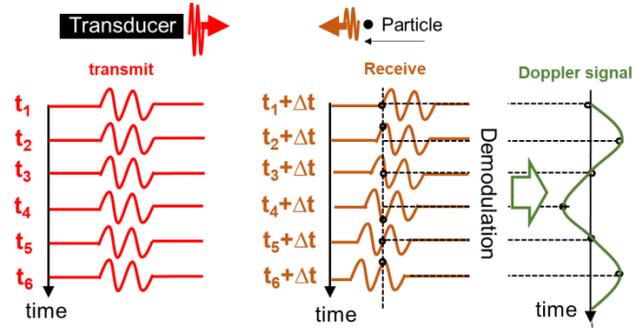


Fig. 3. Doppler signal demodulation.

In the bubbly flow, the phase separation technique for the UVP system's operation to obtain the velocity in the bubbly used algorithm. Firstly, the Doppler signal  $D(n)$  (discrete data) is extracted from the echo signals  $e(t)$ , as shown in equation (2) obtained from the transducer and pulser/receiver, respectively. The extraction is processed by quadrature demodulation [19].

$$D_i(n) = A_{n,i} \cos\left(\frac{2\pi n f_{D,i}}{f_{PRF}} - \varphi\right) - j A_{n,i} \sin\left(\frac{2\pi n f_{D,i}}{f_{PRF}} - \varphi\right) \quad (2)$$

where  $n$  represents a sampling of Doppler signal,  $A$  is the amplitude, and  $\varphi$  is the initial phase. Then, the Doppler signal is sent to STFT to derive a time-frequency spectrogram of the signal. The calculation is expressed in equation (3), and the energy density of spectra at time  $k$  is denoted by Equation (4). Time-frequency resolution depends on time step  $S_n$  and window length  $W_n$ . The spectrogram is sent to the peak detector for analyzing the energy peaks of the spectrogram. Peak value in each position informs the Doppler frequency data ( $\mathbf{f}_D = [f_{Da}, f_{Db}, \dots, f_{Dm}]$ ) and time location ( $\mathbf{t} = [t_a, t_b, \dots, t_m]$ ). Furthermore, the Doppler amplitude at each point ( $\mathbf{a} = [a_{n=0}, a_{n=1}, \dots, a_{n=N_{rep}-1}]$ ) is detected by making the envelope on the Doppler signal. These data are selected by time location index obtained from peak detector ( $\mathbf{a}_S = [a_a, a_b, \dots, a_m]$ ).

$$X(k, f_D) = \sum_{n=0}^{N_{REP}-1} D(n) W_n(n - kS_n) \exp(-jn2\pi f_D) \quad (3)$$

$$P(k, f_D) = |X(k, f_D)|^2 \quad (4)$$



The selected amplitude data is then compared with a threshold value. The value is defined as being higher than the maximum Doppler amplitude of the particle and lower than the Doppler amplitude obtained from the bubble. The amplitude index is classified into the index of bubble and liquid. When the amplitude value is higher than the threshold, the index is defined as a bubble index ( $in_b=[in_{b1},in_{b2}..in_{bn}]$ ). Furthermore, when the value is lower than the threshold, the index is expressed as a particle index ( $in_p=[in_{p1},in_{p2}..in_{pn}]$ ). Doppler frequency data analyzed by peak detector is classified by these amplitude indexes to be Doppler frequency of bubble group ( $f_{Dbubble}=[f_{Db_{a1}},f_{Db_{b1}}...f_{Db_{m1}}]$ ) and particle group ( $f_{Dparticle}=[f_{D_{p1}},f_{D_{p2}}...f_{D_{pn1}}]$ ). The Doppler frequency in each group is averaged. Hence, the Doppler frequency of bubble  $f_{Dbubble}$  and particle  $f_{Dparticle}$  in the same measurement channel is decomposed apparently. Consequently, the bubble and particle (liquid) velocity can be calculated simultaneously follow the equation (1).

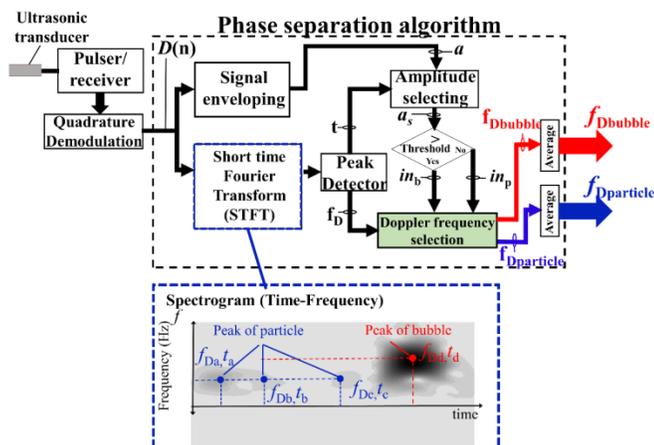


Fig. 4. The UVP with phase separation for measurement in the bubbly flow.

**B. Integrated telemetry system**

The internet of things (IoT) is a hot issue and plays an essential role in several sectors. For the telemetry system applied for the experimental work remotely, the client units that work and facilitate the remote work execution communicates with the outside users via Wi-Fi signal internally and with internet services provider (ISP), respectively, as shown in figure 5. The ISP provides internet service that connects data sent through router devices such as Dial, DSL, and wireless modems [20] where the latter is selected for this study. The computer, tablet, and smartphones are responsible for remote control over the internet anywhere via 3G/4G or routers that communicate with the ISP.

**III. EXPERIMENTAL SETUP**

Figure 6 illustrates a schematic diagram of the UVP, the experimental apparatus, and the telemetry system. The UVP system consisted of 4 MHz transducers, an ultrasonic pulser/receiver, a digitizer, and a computer with LabVIEW software. The pulser/receiver emitted ultrasonic pulses via a transducer. The echo signals received by the pulser/receiver were converted into a digital signal by the digitizer; with a

sampling rate of 100MS/s. Data from the digitizer were transferred to the computer via USB port. The UVP calculation and analysis were performed by using the LabVIEW software on the experimental computer. The UVP parameter setting shown in table 1 was set to be compatible with the measurement condition. For the demonstration, the UVP was applied to measure the velocity profile of the single-phase liquid and air bubbly flow on the vertical pipe flow apparatus, respectively. The pipe with a 50 mm internal diameter was used. The tap water dispersed with nylon particles 80 μm and bubbles (diameter ≈ 2-3 mm) were used as working fluid. Its temperature is 25± 4°C. The pump circulated the water. The electromagnetic flowmeter monitored the water flow rate. A bubble generator generated bubbles put on upstream 50D from the test section. The transducers were immersed into the water box—the transducer end located outside the pipe with the incident angle 45 degrees. In the telemetry section, the camera, smart electrical switch, and remote software were employed to control the experimental work remotely. Three cameras and two smart switches were utilized on the system.

TABLE I. PARAMETER CONFIGURATION OF UVP SYSTEMS

Parameter	Value
Basic frequency	4 MHz
Active diameter of transducer	5 mm
Emission voltage	150 Vp-p
Receiving gain	40 dB
Pulse repetition frequency	8 kHz
Number of repetition	128
Channel width	0.74 mm
Number of channel	60 channel
Sound velocity in water	1493 at 25 degree C
Incident angle	45 degree

These cameras were used for overview observation, monitor the flow rate value on the flow meter, and observe the test section. The smart switches were employed to control the power supply of the UVP system, water pump, and air compressor separately. The camera was connected to the experimental computer via USB port, and its data was transferred via Wi-Fi, respectively. The smart switch control was executed via a Wi-Fi network. The ordering, information, and status were communicated with the user interface by utilizing the ISP. The experimental computer was proposed to operate via the remote access tool (Team viewer software was employed).

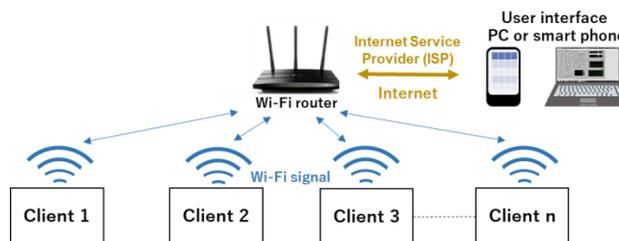


Fig. 5. Schematic of a communication network for the remoted work operation

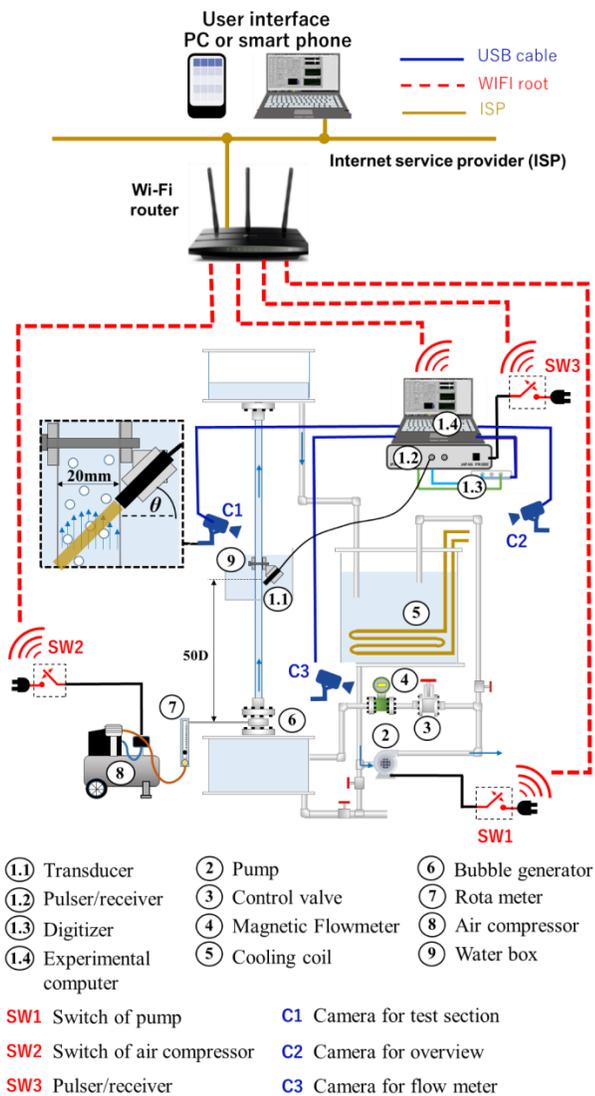


Fig. 6. Experimental setup.

IV. RESULT AND DISCUSSION

A. Telemetry system testing

Figure 7 shows the web browser of smart switch control. These switches were used to control the power source to energize the UVP system, fluid circulating pump, and bubble generator's air compressor. The power of equipment can be controlled remotely on the web browser. Besides, the remoted observation during the experiment was operated via cameras, as illustrated in figure 8. The flow meter display unit was monitored continuously for an accurate experiment. The test section and an overall view of the apparatus were observed due to the safety aspect. Figure 9 shows the operating panel of the UVP program. The program was developed in the LabVIEW version 2011.

Figure 10 illustrates the remote access screen of the telemetry system. Every browser was arranged to fit the monitor.

B. The experiment in single-phase liquid flow measurement

UVP parameters such as basic frequency, pulse repetition frequency, measurement channel, number of the profile, and others could be set via the remote access software. The UVP system and liquid pump were started remotely. Figure 11 represents the results of the instantaneous of the velocity profile of single-phase liquid flow. Figure 12 illustrates the averaging profile of 5,000 instantaneous data in single-phase liquid flow. The superficial liquid velocity  $U_L$  was set at 300 mm/s. The horizontal axis is the distance from the wall ( $r$ ) nominalized by the pipe radius ( $R$ ). However, the velocity value of UVP near the wall's vicinity showed small fluctuations due to some parts of the ultrasonic measurement volume located within the wall, which overlapped the ultrasonic wave on the fluid and pipe wall. The echo signal in

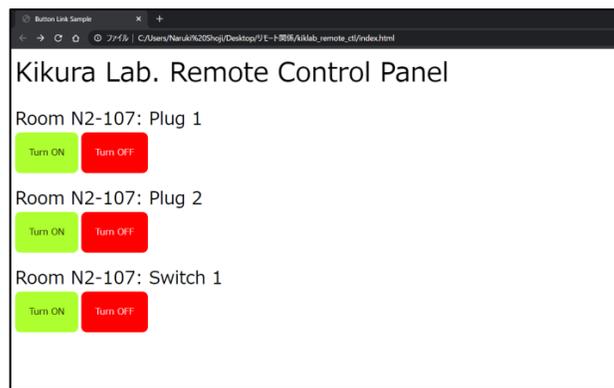


Fig. 7. Smart switch control browser.

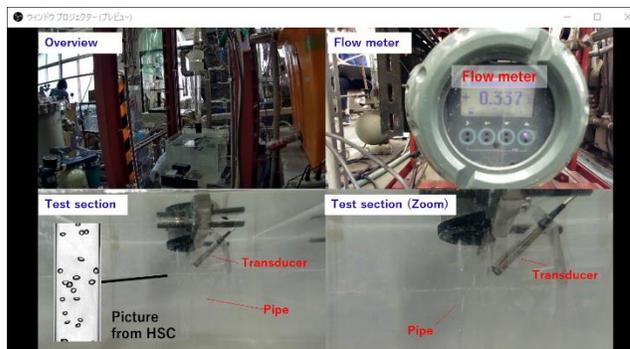


Fig. 8. Camera viewing browser.

that zone is not only affected by the particle. Moving particles and the pipe wall influence the measurement result. Hence, it can be concluded that the UVP efficiently measured the velocity profile on a single-phase flow even executed via the telemetry system. Moreover, the measurement result could be transferred to the user interface device promptly for post-processing.

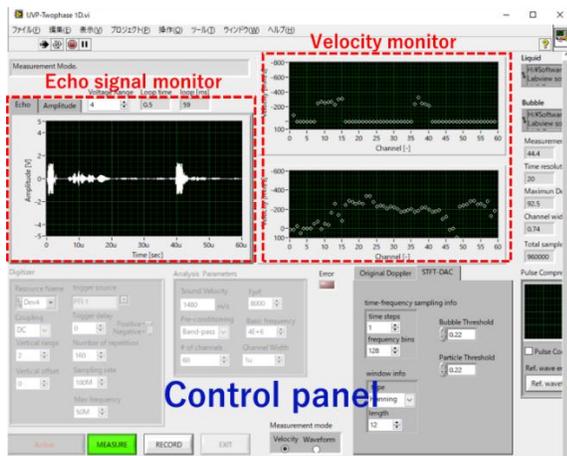


Fig. 9. The UVP program operation panel

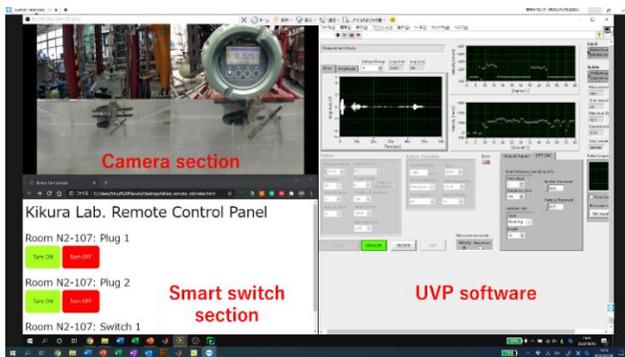


Fig. 10. The screen of the remote access to experimental computer.

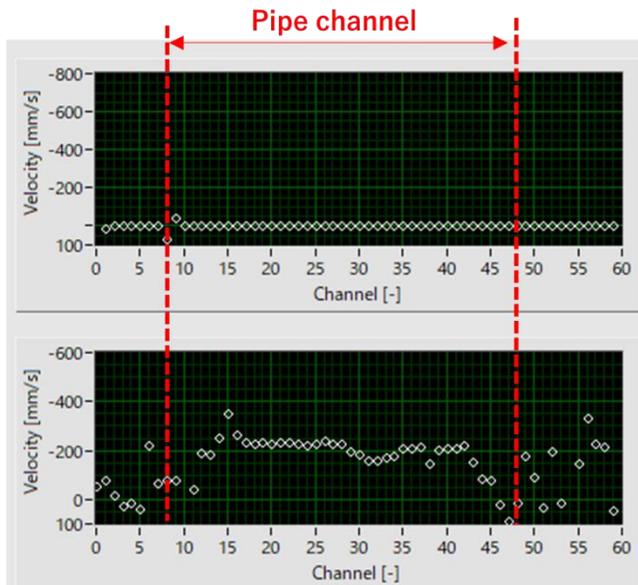


Fig. 11. The result of instantaneous velocity profile in single-phase liquid flow.

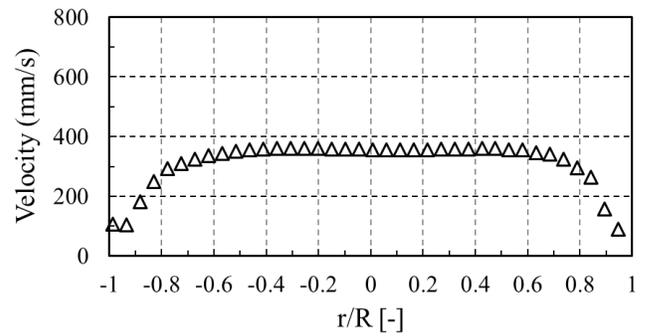


Fig. 12. The result of average velocity in single-phase liquid flow.

### C. The experiment in bubbly flow measurement

In the bubbly flow, the experiment was conducted at  $U_L = 300$  mm/s and  $U_G = 5.3$  mm/s. The bubble diameter in this experiment was about 2-3 mm. The UVP system and liquid pump were started. Also, the air compressor was powered up by the telemetry system to inject bubbles. Figure 13 represents that the instantaneous velocity profile of the bubble and liquid can be obtained remotely. Figure 14 shows the average measurement results of two-phase bubbly flows. Bubbles rise mainly near the wall region. The graph shows the mean velocity profile data. Liquid velocity distribution is the averaging of 5,000 profiles, and bubble velocity is averaged by the amount of data obtained. The measurement result of the bubble velocity profile after separation by the developed technique was derived. The velocity level was higher than the liquid velocity due to the buoyancy force effect. Besides, the liquid velocity profile was obtained and separated from the bubble phase. The velocity of both phases could be measured separately. Then, the slip ratio also was derived.

It can be summarized that the bubble velocity, liquid velocity, and slip ratio of the bubbly flow can be derived, although the experiment was conducted on the telemetry system.

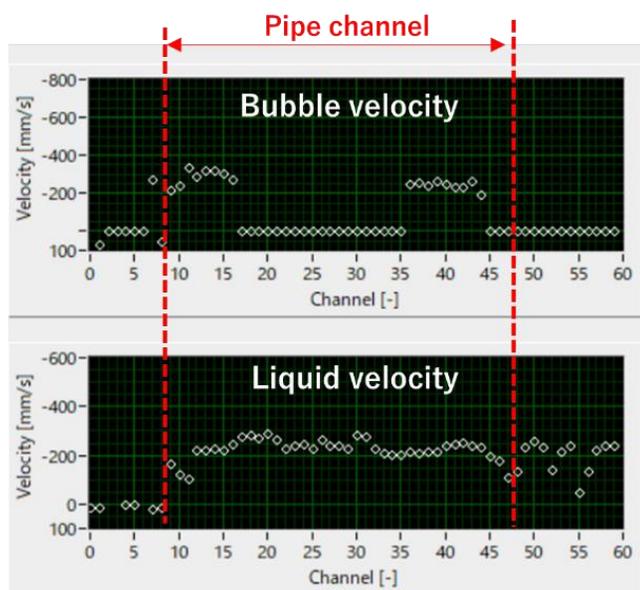


Fig. 13. The result of instantaneous velocity profile on bubbly flow.

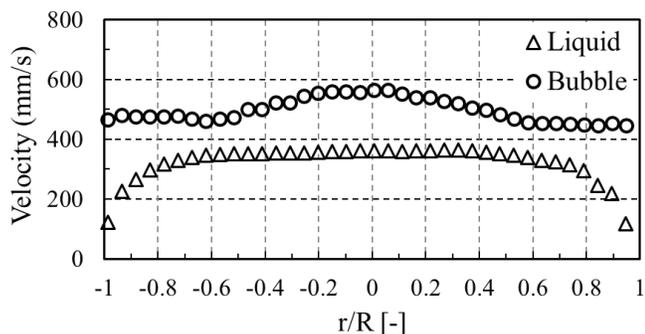


Fig. 14. The result of average velocity profile on bubbly flow.

## V. CONCLUSION

The IoT based telemetry concept to execute the experimental activity remotely was proposed. The experimental investigation on the facility can be conducted anywhere outside on the internet is provided. The limitation of the experimental task in crisis or pandemic disease situation is removed. The experimental study on the velocity distribution measurement in the bubbly flow using ultrasonic measurement, which is vital in fluid engineering, was demonstrated to confirm the telemetry system's ability. The power supply of the UVP system, liquid pump, and bubble generator was controlled via the web browser interface. The UVP program operation and cameras were executed with the remote access tool. The remoted experiment's velocity profile results, whether single-phase liquid or bubbly flow, were obtained apparently.

## REFERENCES

- [1] M. Ishii, and T. Hibiki, Thermo-fluid dynamics of two-phase flow, Springer, 2012.
- [2] T. Ozaki, R. Suzuki, H. Mashiko, and T. Hibiki, Development of drift-flux model based on 8×8 BWR rod bundle geometry experiments under prototypic temperature and pressure conditions, Journal of Nuclear Science and Technology, 2013, vol. 50, pp. 563-580.
- [3] S. Hosokawa, K. Hayashi, and A. Tomiyama, Void distribution and bubble motion in bubbly flows in a 4×4 rod bundle. Part I: Experiments, Journal of Nuclear Science and Technology, 2014, vol. 51, pp. 220-230.
- [4] I. Chakraborty, and P. Maity, COVID-19 outbreak: Migration, effects on society, global environment and prevention, Science of the Total Environment, 2020, vol. 728, pp. 1-7.
- [5] K. Prem, Y. Liu, K. Prem, T. Russell, A. J. Kucharski, R. M. Eggo, and N. Davies, The effect of control strategies to reduce social mixing on outcomes of the COVID-19 epidemic in Wuhan, China: a modelling study, Lancet Public Health, 2020, vol. 5, pp. 261-270.
- [6] L. Atzori, A. Iera, and G. Morabito, The Internet of Things: A survey, Computer Networks, 2010, vol. 54, pp. 2787-2805.
- [7] H. Boyes, B. Hallaq, J. Cunningham, and T. Watson, The industrial internet of things (IIoT): An analysis framework, Computers in Industry, 2018, vol. 101, pp. 1-12.
- [8] L. D. Xu, W. He, and S. Li, Internet of Things in Industries: A Survey, IEEE Transactions on Industrial Informatics, 2014, vol. 10, pp. 2233 - 2243.
- [9] S. Chaudhury, R. Mukherjee, D. Paul, and S. Haldar, Internet of Thing based healthcare monitoring system, 8th Annual Industrial Automation and Electromechanical Engineering Conference (IEMECON), 16-18 Aug 2017, Bangkok, Thailand.
- [10] J. Muangprathub, N. Boonnam, S. Kajornkasirat, N. Lekbangpong, A. Wanichsombat, and P. Nillaor, IoT and agriculture data analysis for smart farm, Computers and Electronics in Agriculture, 2019, vol. 156, pp. 467-474.
- [11] J. Pitakphongmetha, N. Boonnam, S. Wongkoon, T. Horanont, D. Somkiadcharoen, and J. Prapakornpilai, Internet of things for planting in smart hydroponics style, International Computer Science and Engineering Conference (ICSEC), 14-17 Dec 2016, Chiang Mai, Thailand.
- [12] H. Aldowah, S. U. Rehman, S. Ghazal, and I. N. Umar, Internet of Things in Higher Education: A Study on Future Learning The 6th International Conference on Computer Science and Computational Mathematics (ICCSM), 4-5 May 2017, Langkawi, Malaysia.
- [13] B. Jan, H. Farman, M. Khan, M. Talha, and I. U. Din, Designing a Smart Transportation System: An Internet of Things and Big Data Approach, IEEE Wireless Communications, 2019, vol. 26, pp. 73-79.
- [14] S. Dey, A. Roy, and S. Das, Home automation using internet of thing, IEEE 7th Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON), 20-22 Oct 2016, New York, NY, USA.
- [15] Y. Takeda, Velocity profile measurement by ultrasonic Doppler shift method, International Journal of Heat Fluid Flow, 1986, vol. 7, pp. 313-318.
- [16] H. Kikura, Y. Takeda, and F. Durst, Velocity profile measurement of the Taylor vortex flow of a magnetic fluid using the ultrasonic Doppler method, Experiments in Fluids, 1999, vol. 26, pp. 208-214.
- [17] S. Eckert, and G. Gerbeth, Velocity measurements in liquid sodium by means of ultrasound Doppler velocimetry, Experiments in Fluids, 2002, vol. 32, pp. 542-546.
- [18] W. Wongsaroj, A. Hamdani, N. Thong-un, H. Takahashi, and H. Kikura, Extended Short-Time Fourier Transform for Ultrasonic Velocity Profiler on Two-Phase Bubbly Flow Using a Single Resonant Frequency, Computers in Industry, 2019, vol. 9, 50.
- [19] Y. Takeda, and et al., Ultrasonic Doppler Velocity Profiler for Fluid Flow: Springer: Tokyo, Japan, 2012.
- [20] P. Park, S. C. Ergen, C. Fischione, and C. Lu, Wireless Network Design for Control Systems: A Survey, IEEE Communications Surveys & Tutorials, 2018, vol. 20, pp. 978-1013.



# Consensus and Eigenvalue Estimation in Dynamic Consensus Networks with Strictly Positive Real Systems

Fadel Lashhab, Member, IEEE

*Department of Electrical Engineering and Computer Science*  
*College of Engineering and Architecture*  
 Howard University, Washington DC, USA.  
 fadel.lashhab@howard.edu

**Abstract**—The use of consensus networks in networking has received great attention due to its wide array of applications in fields such as robotics, transportation, sensor networking, communication networking, biology, and physics. The focus of this paper is to study a generalization of consensus problems whereby the weights of network edges are no longer static gains, but instead are dynamic systems, leading to the notion of dynamic consensus networks.. Specifically, we consider networks whose nodes are transfer functions (typically integrators) and whose edges are strictly positive real transfer functions representing dynamical systems that couple the nodes. We transform each concept of static graph theory into dynamic terms, out of which a generalized dynamic graph theory naturally emerges. 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. We show that strictly positive realness of the edges is a sufficient condition for dynamic networks to be stable (i.e., to reach consensus). To study the spectral properties of dynamic networks, we introduce the Dynamic Grounded Laplacian matrix, which is used to estimate lower and upper bounds for the real parts of the smallest and largest non-zero eigenvalues of the dynamic Laplacian matrix. These bounds can be used to obtain stability conditions using the Nyquist graphical stability test for undirected dynamic networks controlled using distributed controllers. Numerical simulations are provided to verify the effectiveness of the results.

**Keywords**—Dynamic graphs, dynamic Laplacian, Dynamic Consensus, Grounded Laplacian, strictly-positive real systems, spectrum of a dynamic Laplacian.

## I. INTRODUCTION

In recent years, consensus in networks has received considerable attention due to a wide array of applications in fields such as robotics, cooperative unmanned air vehicles, transportation, sensor networks, communication networks, air traffic control, biology, and physics. The focus of this paper is to study a generalization of consensus problems in which the weights of network edges are no longer static gains, but

instead are dynamic systems, leading to the notion of *dynamic consensus networks*.

To understand the basic idea of *consensus*, consider the graph or network shown in Fig. 1. This graph consists of three nodes or agents with node variables denoted as  $v_i$ ,  $v_j$ , and  $v_k$  and edges  $e_{ij}$  connecting the  $i^{th}$  and  $j^{th}$  nodes. Such graphs can be described as a set of nodes (or vertices)  $\mathcal{N} = \{n_i\}$  connected by a set of edges  $\mathcal{E} = \{(n_i, n_j) : n_i, n_j \in \mathcal{N}\}$ . Each edge  $\mathcal{E}$  is associated with a flow  $f_{ij}(t)$ , which in this figure is the same as  $f_{ji}(t)$ , implied by the bi-directional arrows.

In Fig. 1, we assume the node does some form of processing of its inputs and that edges carry a flow of information or energy from one node to the other. Assuming an external input  $u_i$  to each node, then a typical form of the processing in each node is, for example,

$$\dot{v}_i(t) = u_i(t) - \sum_{j \in \mathcal{N}_i} f_{ij}(t) \quad (1)$$

where  $\mathcal{N}_i$  is the set of indices for nodes that communicate with node  $i$ . This is called the neighborhood of node  $i$  and the nodes in this case are considered to be integrators. Note that a popular form for  $f_{ij}(t)$  is

$$f_{ij}(t) = \lambda_{ij}(v_i(t) - v_j(t)), \quad (2)$$

where  $\lambda_{ij}$  is a static gain or interconnection weight between node  $i$  and node  $j$ . With  $u_i(t) = 0$ , (1) results in the so-called *consensus protocol*

$$\dot{v}_i(t) = - \sum_{j \in \mathcal{N}_i} \lambda_{ij}(v_i(t) - v_j(t)). \quad (3)$$

For the traditional consensus networks, the network topology is static, 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 [24]. Thus, static

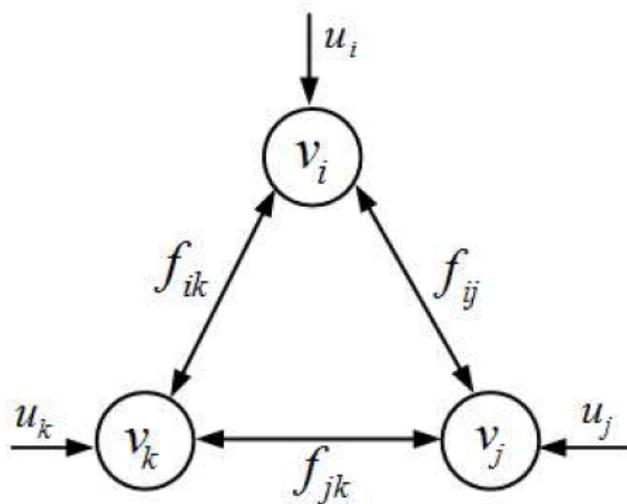


Fig. 1. Graph or network.

consensus problems can be written in the time domain for each node  $i = 1, 2, \dots, n$  as

$$\dot{x}_i = \sum_{j \in \mathcal{N}_i} \lambda_{ij} (x_j(t) - x_i(t)), \quad (4)$$

where  $\mathcal{N}_i$  is called the neighborhood of node  $i$ . The continuous-time linear consensus protocol (4) can be written in matrix form as:

$$\dot{v}(t) = -Lv(t), \quad (5)$$

where  $v(t) = [v_1(t), v_2(t), \dots, v_n(t)]^T$  and  $L = [l_{ij}]$ , is the Laplacian matrix, defined by

$$l_{ij} = \begin{cases} \sum_{j \in \mathcal{N}_i} \lambda_{ij} & i = j \\ -\lambda_{ij} & i \neq j \text{ and } (i, j) \in \mathcal{E} \\ 0 & \text{otherwise} \end{cases} \quad (6)$$

For the multi-agent consensus problem, the state  $v_i \in \mathbb{R}^1$  represents the belief of agent  $i$  about a so-called global consensus variable  $v$ . These beliefs are updated using communications with their neighbors according to the consensus protocol (4). A key result is that the solution of  $\dot{v}(t) = -Lv(t)$  gives  $v_i \rightarrow v^*$ , where  $v^*$  is a constant, if the static graph is connected [24]. This specific fact has been the basis of much of the literature related to consensus problems. The focus of this paper is to study a generalization of consensus problems whereby the weights of the network edges are no longer modeled as static gains, but instead are represented as dynamic systems coupling the nodes.

By dynamic systems, we mean that the relationships between the system variables are described by linear ordinary differential equations. We call such networks *dynamic consensus networks* because under some conditions, all the node variables converge to a common value called a consensus value.

Consensus problems using Fig. 1 and its variants have attracted much research interest in recent years, because many

applications admit a model in the form of (1), including formation control [8], [17], [20], distributed sensor fusion in sensor networks [9], [25], [34], flocking [4], [31], [32], rendezvous problems in space [1], [19], and synchronization in power networks [6], [21], [26], among others. The origin of formal studies of consensus problems had its roots in the 1960s (see [5] and references therein). A good survey of consensus problems is found in [23], [29], and the references therein.

Recently, several groups have begun to consider consensus over networks with dynamic edges. Such networks arise in applications in several ways. In [33], the authors consider consensus of networks with dynamic channels. In these types of consensus problems, it is assumed that there exists a channel (filter) from node  $j$  to node  $i$ , making the signal received by  $i$  from  $j$  to be  $\hat{v}_j(t) = h_{ij}(t) * v_j(t)$ , where  $h_{ij}(t)$  is the impulse response of a linear, time-invariant (LTI) filter of the edge  $e_{ij}$  and “ $*$ ” denotes the convolution operation. By taking the Laplace transform of  $\hat{v}_j(t)$ , we can write  $\hat{V}_j(s) = H_{ij}(s)V_j(s)$ , where  $H_{ij}(s)$  represents the transfer function of the LTI filter of the edge  $e_{ij}$ . Based on this, the dynamics of node  $i$  for  $i = 1, 2, \dots, n$  can be written in the time domain as:

$$\dot{v}_i(t) = \sum_{j \in \mathcal{N}_i} \lambda_{ij} [\hat{v}_j(t) - v_i(t)]. \quad (7)$$

By taking the Laplace transform, we have

$$sV_i(s) - v_i(0) = \sum_{j \in \mathcal{N}_i} \lambda_{ij} [H_{ij}(s)V_j(s) - V_i(s)], \quad (8)$$

which leads us to

$$V(s) = (sI_n + L(s))^{-1}V(0), \quad (9)$$

where  $V(s) = [V_1(s), V_2(s), \dots, V_n(s)]^T$ ,  $V(0) = [v_1(0), v_2(0), \dots, v_n(0)]^T$ , and  $L(s) = [l_{ij}(s)]$  is the Laplacian matrix defined by:

$$L_{ij}(s) = \begin{cases} \sum_{j \in \mathcal{N}_i} \lambda_{ij} & i = j \\ -\lambda_{ij} H_{ij}(s) & i \neq j \text{ and } (i, j) \in \mathcal{E} \\ 0 & \text{otherwise} \end{cases} \quad (10)$$

Consensus protocols such as (8) suggest further generalization: Filtering of the difference  $(v_i - v_j)$ : This type of consensus protocol extends the protocol (8) by filtering the difference between the node variables instead of filtering the difference between the received and transmitted signals. Thus, the dynamic consensus protocol can be written in the time domain for the node  $i$ ,  $i = 1, 2, \dots, N$ , as

$$\dot{v}_i(t) = \sum_{j \in \mathcal{N}_i} h_{ij}(t) * [v_j(t) - v_i(t)], \quad (11)$$

By taking the Laplace transform of (11), we get

$$sV_i(s) - v_i(0) = \sum_{j \in \mathcal{N}_i} H_{ij}(s)[V_j(s) - V_i(s)], \quad (12)$$

$$V(s) = (sI_n + L(s))^{-1}V(0), \quad (13)$$



where  $L(s) = [l_{ij}(s)]$  is defined by:

$$l_{ij}(s) = \begin{cases} \sum_{j \in \mathcal{N}_i} H_{ij}(s) & i = j \\ -H_{ij}(s) & i \neq j \text{ and } (i, j) \in \mathcal{E} \\ 0 & \text{otherwise} \end{cases}, \quad (14)$$

$L(s)$  here is different from the Laplacian matrix defined in (10) for the dynamic networks introduced by [33]. We call (14) a dynamic Laplacian matrix. To our knowledge, the model is unique to our research, though note that in a recent paper [30], the filter of the difference  $[v_i - v_j]$  also appears.

The consensus protocol given in (4) and its variants have been studied extensively in the literature and have been applied in a number of areas, notably to formation motion control when the agents are mobile. We have been motivated to consider this extension to transfer functions weights from several applications. In particular, the thermal model of heat flows in buildings can be shown to admit a generalized dynamic consensus network over a directed graph [22]. In another example, modeling the load frequency control (LFC) network of an electrical power grid can be viewed as a single-integrator consensus network with dynamic interconnection coefficients  $\lambda_{ij} = G_i(s)a_{ij}$  [16]. To further motivate the concept from an application, in the next section we show how an electrical circuit whose nodes are grounded capacitors can be modeled as an undirected dynamic consensus network.

This paper offers several distinct contributions. First, we show that the nodes variables in dynamic consensus networks whose nodes are integrators and whose edges are strictly positive real transfer functions will converge to a common consensus value. Second, we develop methods for estimating the bounds of the eigenvalues of a dynamic Laplacian matrix. This is accomplished by introducing the idea of a grounded dynamic Laplacian sub-matrix as a means of reducing the order of complexity of computation that would be necessary in the case of the full dynamic Laplacian matrix. Third, we use the estimated bounds of the eigenvalues of a dynamic Laplacian to analyze the consensus and stability of a dynamic consensus network under distributed feedback control.

The paper is organized as follows: Section II presents a motivating physical example of a dynamic consensus network. In Section III, 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 dynamic interconnection matrices and dynamic consensus networks. Section IV we investigate the consensus conditions of dynamic consensus networks with integrator nodes and strictly positive real (SPR) edges. For such graphs, we can introduce what we call the dynamic Laplacian and dynamic Grounded Laplacian of the dynamic graphs. In Section V, lower and upper bounds for the largest and smallest non-zero eigenvalues of the dynamic Laplacian matrix are derived. Finally, using these bounds, we characterize in Section VI the stability of a closed loop system consisting of an undirected dynamic consensus network controlled by distributed

controllers, and a numerical example is provided to illustrate the results.

## II. DYNAMIC CONSENSUS NETWORK

A generalization of static consensus networks can be motivated when the consensus network consists of an interconnection of physical systems (e.g., an electrical network that requires a common operating condition such as voltage or frequency) or in cases when the communication between agents in a consensus network includes dynamic behavior (e.g., delays). This paper studies a generalization of the static consensus problem whereby the weights are no longer positive gains, but instead represent strictly-positive real dynamic systems. The use of positive real (PR) transfer functions is motivated by the study of dynamic networks that represent linear electrical circuits composed of resistors, inductors and capacitors. A strictly positive real transfer (SPR) function corresponds to the driving input impedance in dissipative networks that contain resistors, inductors and capacitors [28].

*Definition 2.1:* [28] A  $n \times n$  proper rational transfer function matrix  $G(s)$  is called positive real if any of the following equivalent condition hold:

- 1) Poles of all elements of  $G(s)$  are in  $\text{Re}[s] \leq 0$ .
- 2) For all real  $\omega$  for which  $j\omega$  is not a pole of any element of  $G(s)$ , the matrix  $G(j\omega) + G^T(-j\omega)$  is positive semidefinite (PSD).
- 3) Any pure imaginary pole  $j\omega$  of any element of  $G(s)$  is a simple pole and the residue matrix  $\lim_{s \rightarrow j\omega} (s - j\omega)G(s)$  is positive semidefinite Hermitian.

When  $n = 1$ , the second condition of the definition reduces to  $[\text{Re}G(j\omega)] \geq 0, \forall \omega \in \mathbb{R}$ . In addition, a real rational transfer function  $h(s)$  is said to be strictly positive real (SPR) if  $h(s - \epsilon)$  is positive real for some  $\epsilon > 0$ .

Consider a dynamic network whose node equations can be written in the time domain for the node  $i, i = 1, 2, \dots, n$ , as

$$\dot{v}_i(t) = - \sum_{j \in \mathcal{N}_i} y_{ij}(t) * [v_i(t) - v_j(t)], \quad (15)$$

where  $v_i$  is a consensus variable of the node  $i$ ,  $y_{ij}(t)$  is the impulse response of the edge  $e_{ij}$ , and “\*” denotes the convolution operation. Taking the Laplace transform of (15) results in

$$V_i(s) = - \frac{1}{s} \sum_{j \in \mathcal{N}_i} Y_{ij}(s) [V_i(s) - V_j(s)] + \frac{1}{s} v_i(0), \quad (16)$$

where  $Y_{ij}(s)$  is the Laplace transform of  $y_{ij}(t)$ . We refer to (15) or (16) as a *dynamic consensus* protocol. Such a transfer function can be viewed as a weight on the edge connecting node  $j$  to node  $i$ . As a special case, we consider graphs where these weights  $Y_{ij}(s)$  are strictly-positive real transfer functions.

We now consider graphs formed by nodes with multiple terminals that are connected by weighted edges where the weights are represented as transfer functions taken from real rational functions analytic in the right half of the complex plane. These graphs are connected with the dynamic consensus protocol introduced earlier, and the dynamic Laplacian matrix

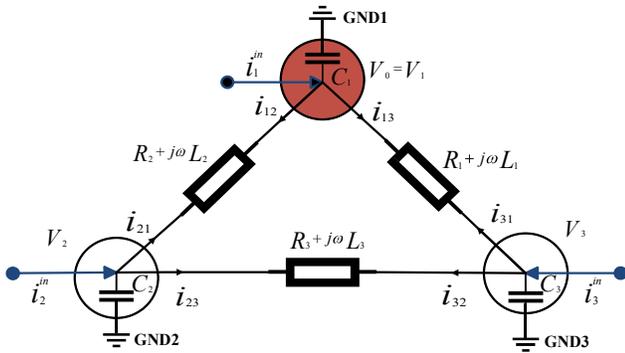


Fig. 2. Electrical network as dynamic graph.

is introduced. To illustrate how such a network can arise in applications, consider a simple electrical network shown in Fig. 2. This circuit can be represented as a dynamic graph by viewing the capacitors as nodes of the dynamic graph and the connections between capacitors as edges in which each edge is assigned a real rational transfer function  $Z_{ij}(s)$  as its impedance. The node variables ( $v_i, \forall i = 1, \dots, N$ ) are voltages across the capacitors, and current is the flow along the edges between nodes. The weight of an edge can be viewed as the admittance of that edge, which we will denote  $Y_{ij}(s) = \frac{1}{Z_{ij}(s)}$  for a given edge  $e_{ij}$ . If there is a potential difference ( $v_i - v_j$ ) between the nodes  $i$  and  $j$ , an electrical current  $i_{ij}$  will flow across the edge  $e_{ij}$  from  $i$  to  $j$  according to Ohm's law, which states that the potential drop across an edge  $e_{ij}$  must be equal to the product (in the complex frequency domain) of its impedance and the current flowing through it:

$$I_{ij}(s) = Y_{ij}(s)[V_i(s) - V_j(s)], \quad (17)$$

where  $I_{ij}(s)$  and  $V_i(s)$  are the Laplace transform of the current  $i_{ij}(t)$  and the node voltage  $v_i(t)$ , respectively. Using Kirchoff's law, it can be established that the algebraic sum of all of the currents entering a node  $i$  is equal to the algebraic sum of all currents leaving that node. Therefore, for Fig. 2, the dynamic model of each node is obtained as:

$$\begin{aligned} C\dot{v}_1(t) &= i_1^{in}(t) - (i_{12}(t) + i_{13}(t)); \\ C\dot{v}_2(t) &= i_2^{in}(t) - (i_{21}(t) + i_{23}(t)); \\ C\dot{v}_3(t) &= i_3^{in}(t) - (i_{31}(t) + i_{32}(t)). \end{aligned} \quad (18)$$

Noting that

$$i_{ij}(t) = -i_{ji}(t), \quad (19)$$

and, by combining (17) with the Laplace transformed version of (18) and setting  $C = 1F$ , we obtain:

$$\begin{aligned} V_i(s) &= \frac{1}{s}I_i^{in}(s) - \frac{1}{s} \sum_{j \in \mathcal{N}_i} Y_{ij}(s)[V_i(s) - V_j(s)] \\ &+ \frac{1}{s}v_i(0), \forall i = 1, 2, 3, \end{aligned} \quad (20)$$

where  $\mathcal{N}_i$  is the set of neighbors node  $i$ . For the autonomous case ( $I_i^{in}(s) = 0$ ), the dynamic model of each node with zero initial conditions ( $v_i(0) = 0$ ) can then be written as:

$$V_i(s) = -\frac{1}{s} \sum_{j \in \mathcal{N}_i} Y_{ij}(s)[V_i(s) - V_j(s)], \forall i = 1, 2, 3. \quad (21)$$

Defining the node-potential vector  $V(s) = [V_1(s), V_2(s), \dots, V_N(s)]^T$ , the overall system can be represented in the frequency domain by:

$$V(s) = -\frac{1}{s}L(s)V(s). \quad (22)$$

In the time domain, the above system can be represented by

$$\dot{v}(t) = -L(t) * v(t), \quad (23)$$

where  $v(t) = [v_1(t), v_2(t), \dots, v_N(t)]^T$  is the node potential vector and  $L(t)$  In the autonomous case we have  $V(s) = \frac{1}{s}L(s)V(s)$ .

$$V(s) = (sI_N + L(s))^{-1}v(0), \quad (24)$$

In this expression the matrix  $L(s) = [l_{ij}(s)]$  is given as:

$$l_{ij}(s) = \begin{cases} \sum_{j \in \mathcal{N}_i} Y_{ij}(s) & i = j \\ -Y_{ij}(s) & i \neq j \text{ and } (i, j) \in \mathcal{E} \\ 0 & \text{otherwise.} \end{cases} \quad (25)$$

This leads to:

$$L(s) = \begin{bmatrix} Y_{12}(s) + Y_{13}(s) & -Y_{12}(s) & -Y_{13}(s) \\ -Y_{12}(s) & Y_{12}(s) + Y_{23}(s) & -Y_{23}(s) \\ -Y_{13}(s) & -Y_{23}(s) & Y_{13}(s) + Y_{23}(s) \end{bmatrix}. \quad (26)$$

$L(s)$  is defined in this way as a dynamic Laplacian matrix of the dynamic graph  $G$ . In the sequel, we will also refer to the restriction  $L(j\omega) = L(s)$  where  $s = (0 + j\omega)$ .

We can generalize this example by considering the general-undirected, dynamic consensus network shown in Fig. 3. For this system, the dynamic Laplacian has the form shown in (27). For this case, the dynamic Laplacian is a symmetric matrix ( $\lambda_{ij}(s) = \lambda_{ji}(s)$  for all  $e_{ij}$ ).

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,

$$x_i(s) = -\frac{1}{s} \sum_{j \in \mathcal{N}_i} [\lambda_{ij}^S(s)x_i(s) - \lambda_{ij}^C(s)x_j(s)] \quad (28)$$

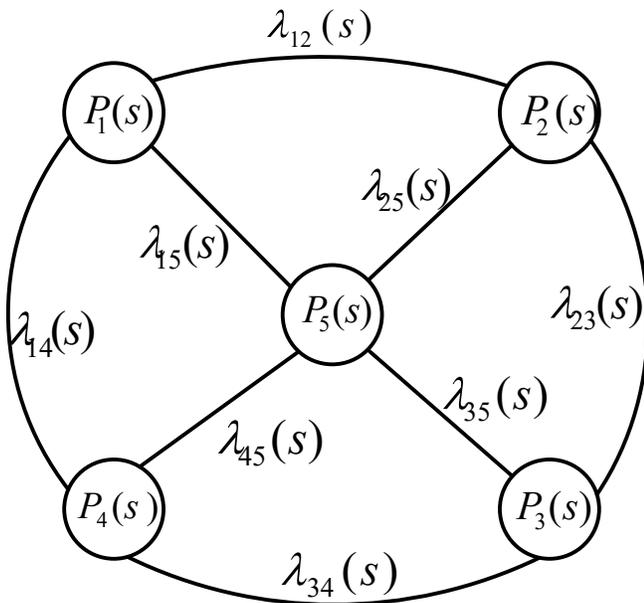


Fig. 3. Undirected, dynamic consensus network as undirected dynamic graph.

- **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,

$$x_i(s) = -\frac{1}{s} \sum_{j \in \mathcal{N}_i} [\lambda_{ij}(s)(x_i(s) - x_j(s))] \quad (29)$$

- **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,

$$x_i(s) = -p(s) \sum_{j \in \mathcal{N}_i} [\lambda_{ij}(s)(x_i(s) - x_j(s))] \quad (30)$$

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

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

$$x_i(s) = -p_i(s) \sum_{j \in \mathcal{N}_i} [\lambda_{ij}(s)(x_i(s) - x_j(s))] \quad (31)$$

**Assumptions:** For the dynamic networks (28 - 31), we make the following assumptions:

- 1) The node and edge processing in the proposed dynamic networks (28 - 31) 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 (29), we assume the dynamics of the edges are strictly positive real (SPR) transfer functions.

- 3) The topology of a network can be directed or undirected. The first dynamic network (28) uses a directed topology, whereas the second dynamic network (29) 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 work [22]. The difference between these two cases is illustrated in (28, 29). These flow models are rooted in the types of dynamic networks to be modeled per the motivation for each network. The first dynamic network (28) is based upon the motivation of modeling the thermal processes of buildings as directed dynamic graphs, while the second dynamic network (29) is based upon the motivation of modeling micro-grids of power systems as undirected dynamic graphs. This model is described in more detail in Section II.
- 5) The dynamics of the nodes can be integrators (28, 29) or more general dynamics (30, 31).
- 6) The dynamics of the nodes and of the edges can be identical (30) or heterogeneous (31).
- 7) These models are often autonomous, meaning that there are no input flows into the dynamic consensus networks. However, in some problems, we add inputs and disturbances into the general forms of the proposed dynamic consensus networks.

### III. DYNAMIC GRAPHS DEFINITIONS

The previous work [22] showed how two different phenomena can be modeled in the form of a graph whose edges are transfer functions (i.e., dynamic systems). In this section, from the motivation of the models 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 static graph.

Consider the example of a directed, dynamic graph shown in Fig. 4. Such graphs can be described as a set of nodes (or vertices)  $\mathcal{N} = \{n_i\}$  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 sequence of arcs from one of the nodes to the other. If there is at least one node that has at least one path to every other node, the graph is said to be 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 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



$$L(s) = \begin{bmatrix} \sum_{j=2,4,5} \lambda_{1j}(s) & -\lambda_{12}(s) & 0 & -\lambda_{14}(s) & -\lambda_{15}(s) \\ -\lambda_{12}(s) & \sum_{j=1,3,5} \lambda_{2j}(s) & -\lambda_{23}(s) & 0 & -\lambda_{25}(s) \\ 0 & -\lambda_{32}(s) & \sum_{j=2,4,5} \lambda_{3j}(s) & -\lambda_{34}(s) & -\lambda_{35}(s) \\ -\lambda_{41}(s) & 0 & -\lambda_{43}(s) & \sum_{j=1,3,5} \lambda_{4j}(s) & -\lambda_{45}(s) \\ -\lambda_{51}(s) & -\lambda_{52}(s) & -\lambda_{53}(s) & -\lambda_{54}(s) & \sum_{j=1,2,3,4} \lambda_{5j}(s) \end{bmatrix}. \quad (27)$$

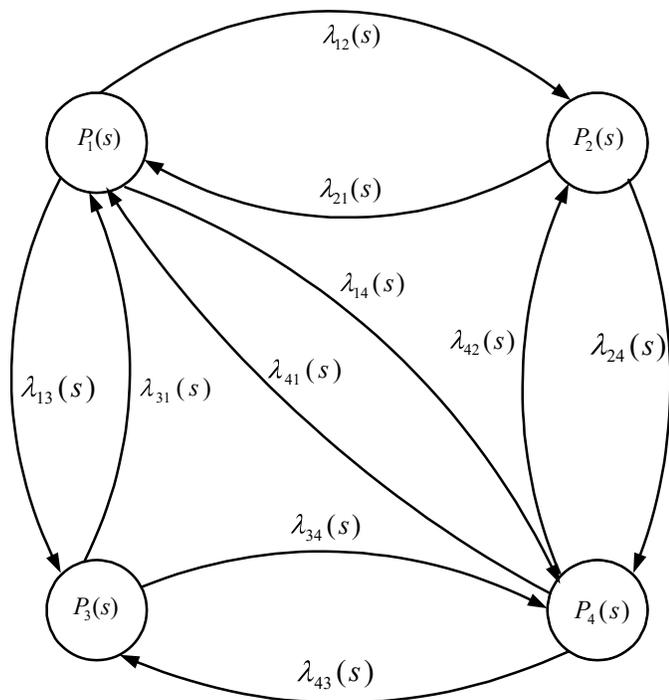


Fig. 4. Directed-Dynamic Graph.

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 special case is when we have integrator nodes with dynamic edges.

Each node  $P_i(s)$  in a directed dynamic graph, such as Fig. 4, 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  $\mathcal{N}_i = \{j : (n_i, n_j) \in \mathcal{E}(s)\}$ . 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

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

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

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

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

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

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  $B_S = B_S^{in} + B_S^{out}$ .

Having derived dynamic degree, adjacency, and incident matrices, we can now give the dynamic equivalent of the static Laplacian matrix. 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

$$l_{ij}(s) = \begin{cases} \sum_{j \in \mathcal{N}_i} \lambda_{ij}(s) & i = j \\ -\lambda_{ij}(s) & i \neq j \text{ and } (i, j) \in \mathcal{E} \\ 0 & \text{otherwise} \end{cases} \quad (32)$$

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

$$L(s) = BD(s)B^T = D(s) - A(s), \quad (33)$$

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:

$$L^{out}(s) = D^{out}(s) - A^{out}(s), \quad (34)$$

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  $D^{out} = B^{out}B^{out^T}$  and  $A^{out} = -B^{out}B^{in^T}$ . 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:

$$\begin{aligned} D^{out}(s) &= B_D^{out}(s)B_S^{out^T}, \\ A^{out}(s) &= -B_D^{out}(s)B_S^{in^T}. \end{aligned} \quad (35)$$

By combining (34) and (35), the outgoing, dynamic Laplacian matrix  $L^{out}(s)$  can be defined as

$$\begin{aligned} L^{out}(s) &= B_D^{out}(s)B_S^{out^T} + B_D^{out}(s)B_S^{in^T} \\ &= B_D^{out}(s)(B_S^{out^T} + B_S^{in^T}) = B_D^{out}(s)B_S^T. \end{aligned} \quad (36)$$

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

To illustrate, for the example shown in Fig. 4, the associated dynamic degree, adjacency, and Laplacian matrices are given by (37). Note that all  $\lambda_{ij}(s)$  in (37) are transfer functions that describe the interconnections (edges) between the nodes. To understand the steady-state (consensus) properties of such a network, as well as its control, we are led to study the spectral properties of  $L(s)$ .

#### IV. CONSENSUS CONDITIONS OF NETWORKS WITH INTEGRATOR NODES AND STRICTLY POSITIVE REAL (SPR) EDGES

In this section we show that the strictly positive realness of the edges is a sufficient condition for dynamic networks of the form (21) to reach a consensus (i.e., all node variables converge to the same constant value in the steady-state).

Analogous to the static consensus networks, we can define a dynamic matrix  $L(s) = [L_{ij}(s)]$  to be a dynamic interconnection matrix as follows:

*Definition 4.1:* Define the set  $\mathbb{S} = \{s : \text{Re}(s) \geq 0, s \neq 0\}$ .  $L(s)$  is a *dynamic interconnection matrix* if it satisfies the following properties:

- 1)  $L(0)$  is an interconnection matrix.
- 2) The elements of  $L(s)$  have no poles in the closed right half of the complex plane.
- 3) The diagonal elements of  $L(s)$  satisfy the following positivity condition: for all  $s \in \mathbb{S}$ ,  $\text{Re}\{L_{ii}(s)\} > 0$ .



$$\begin{aligned}
 D^{out}(s) &= \begin{bmatrix} \lambda_{12}(s) + \lambda_{13}(s) + \lambda_{14}(s) & 0 & 0 & 0 \\ 0 & \lambda_{21}(s) + \lambda_{24}(s) & 0 & 0 \\ 0 & 0 & \lambda_{31}(s) + \lambda_{34}(s) & 0 \\ 0 & 0 & 0 & \lambda_{41}(s) + \lambda_{42}(s) + \lambda_{43}(s) \end{bmatrix}, \\
 A^{out}(s) &= \begin{bmatrix} 0 & \lambda_{12}(s) & \lambda_{13}(s) & \lambda_{14}(s) \\ \lambda_{21}(s) & 0 & 0 & \lambda_{24}(s) \\ \lambda_{31}(s) & 0 & 0 & \lambda_{34}(s) \\ \lambda_{41}(s) & \lambda_{42}(s) & \lambda_{43}(s) & 0 \end{bmatrix}, \\
 L^{out}(s) &= \begin{bmatrix} \lambda_{12}(s) + \lambda_{13} + \lambda_{14}(s) & -\lambda_{12}(s) & -\lambda_{13}(s) & -\lambda_{14}(s) \\ -\lambda_{21}(s) & \lambda_{21}(s) + \lambda_{24}(s) & 0 & -\lambda_{24}(s) \\ -\lambda_{31}(s) & 0 & \lambda_{31}(s) + \lambda_{34}(s) & -\lambda_{34}(s) \\ -\lambda_{41}(s) & -\lambda_{42}(s) & -\lambda_{43}(s) & \lambda_{41}(s) + \lambda_{42}(s) + \lambda_{43}(s) \end{bmatrix}. \tag{37}
 \end{aligned}$$

4)  $L(s) = [L_{ij}(s)]$  satisfies the following diagonal dominance condition: for all  $s \in \mathbb{S}$ ,

$$\operatorname{Re}\{L_{ii}(s)\} > \sum_{j \neq i} |L_{ij}(s)|.$$

Although condition 3 implies that  $L_{ii}(s)$  is passive, the diagonal dominance condition does not require  $L_{ij}, j \neq i$  to be passive.

**Definition 4.2:** A dynamic interconnection matrix is *connected* if  $L(0)$  is connected.

In prior work [22], the diagonal elements of the dynamic matrix  $L(s)$  of the proposed model are assumed to satisfy the following positivity condition (see Definition 4.1): for all  $s \in \mathbb{S}$ ,  $\operatorname{Re}\{L_{ii}(s)\} > 0$ . Indeed,  $L(s) = [L_{ij}(s)]$  in [22] satisfies the following diagonal dominance condition: for all  $s \in \mathbb{S}$ ,

$$\operatorname{Re}\{L_{ii}(s)\} > \sum_{j \neq i} |L_{ij}(s)|.$$

Note that the dynamic Laplacian matrix  $L(s)$  defined in this paper does not satisfy the properties of a dynamic interconnection matrix because the condition of diagonal dominance is not satisfied. Indeed,  $L(s) = [L_{ij}(s)]$  in (25) satisfies the following equality:

$$L_{ii}(s) = \sum_{j \neq i, j=1}^N L_{ij}(s).$$

Furthermore, an approach to prove these last statements is based-on the Gershgorin circle theorem [11]). Let  $A \in \mathbb{C}^{n \times n}$  with elements  $a_{ij}$ . Let  $R_i = \sum_{j \neq i} |a_{ij}|$  and define  $D(a_{ii}, R_i)$  to be the closed disk centered at  $a_{ii}$  with radius  $R_i$ . Then every eigenvalue of  $A$  lies within at least one of the disks  $D(a_{ii}, R_i)$ .

The Gershgorin circle theorem Lemma A.1 is useful for analyzing a network satisfying the diagonal dominance condition as in [22], whereas it is not useful in application to the proposed dynamic network (21, 29).

In this paper, the edges are assumed to be strictly positive real. Thus, for the dynamic network (21) with strictly-positive real edges, we must look to the strictly positive realness and positive definiteness of the complex matrix. To proceed, we first define the concepts of strictly-positive real edges transfer

functions and positive definiteness for complex matrices, and we give two definitions and two lemmas needed for our result:

**Definition 4.3:** [13] An  $n \times n$  complex matrix  $A$  is called positive definite (PD) matrix if  $\operatorname{Re}[x^H Ax] > 0$  and positive semidefinite (PSD) if  $\operatorname{Re}[x^H Ax] \geq 0$ , for all complex vector  $x \in \mathbb{C}^n$ , where  $x^H$  denotes the conjugate transpose of the vector  $x$ .

**Definition 4.4:** [28] An  $n \times n$  proper rational transfer function matrix  $G(s)$  is positive real if

- 1) Poles of all elements of  $G(s)$  are in  $\operatorname{Re}[s] \leq 0$ .
- 2) For all real  $\omega$  for which  $j\omega$  is not a pole of any element of  $G(s)$ , the matrix  $G(j\omega) + G^T(-j\omega)$  is positive semidefinite (PSD).
- 3) Any purely imaginary pole  $j\omega$  of any element of  $G(s)$  is a simple pole and the residue matrix  $\lim_{s \rightarrow j\omega} (s - j\omega)G(s)$  is positive semidefinite Hermitian.

In addition, a real rational transfer function  $h(s)$  is said to be strictly positive real (SPR) if there exists  $\epsilon_0$  such that  $h(s - \epsilon)$  is positive real for  $0 < \epsilon < \epsilon_0$ .

**Lemma 4.5:** [13] A necessary and sufficient condition for a complex matrix  $A$  to be PD or PSD is that the Hermitian part  $H(A) = \frac{1}{2}(A + A^H)$  must be PD or PSD, respectively.

**Lemma 4.6:** [13] Let  $A \in \mathbb{C}^{n \times n}$ . If  $A + A^H$  is PD or PSD, then all the eigenvalues of matrix  $A$  have positive or non-negative real parts, respectively.

Using these definitions and lemmas, for the proposed dynamic network (21) we can give the following result:

**Theorem 4.7:** [14] Given system (22) for the dynamic consensus protocol (21) with edges that are SPR transfer functions, the following statements are true:

- 1) The dynamic Laplacian  $L(s)$  is a complex, symmetric, positive semidefinite (CSPSD) matrix.
- 2) The real parts of the eigenvalues of  $L(j\omega)$  are non-negative ( $\operatorname{Re}[\lambda_i(L(j\omega))] \geq 0, \forall i \in 1, \dots, n$ ), with  $0 = \lambda_1(L(j\omega)) \leq \operatorname{Re}[\lambda_2(L(j\omega))] \leq \dots \leq \operatorname{Re}[\lambda_n(L(j\omega))]$ .
- 3) If  $L(0)$  (the dynamic Laplacian  $L(s)$  evaluated at  $s = 0$ ) is the Laplacian matrix of a connected graph, then  $s = 0$  is a distinct solution of  $\det[sI_N + L(s)] = 0$  and all node variables will reach consensus under the protocol (21).

**Proof:**

- 1) The dynamic Laplacian is a CSPSD matrix:



From the definition of the dynamic Laplacian matrix (25),  $Y_{ij}(j\omega) = Y_{ji}(j\omega)$  and then  $L(j\omega)$  is a complex symmetric (CS) matrix, that is,

$$L(j\omega)^H = \bar{L}(j\omega)^T = \bar{L}(j\omega), \quad (38)$$

where  $\bar{L}(j\omega)^T$  is the conjugate transpose of the dynamic Laplacian matrix. Using (38), the Hermitian part of the dynamic Laplacian matrix is

$$H(L(j\omega)) = \frac{1}{2}(L(j\omega) + L(j\omega)^H) = \text{Re}[L(j\omega)]. \quad (39)$$

From the above equation, we can conclude that if the dynamic Laplacian is a symmetric matrix, then the Hermitian part of  $L(j\omega)$  is equal to the real part of the dynamic Laplacian matrix.

Based on Lemma 4.5, the necessary and sufficient condition for the dynamic Laplacian  $L(j\omega)$  to be a PSD matrix is that the real part of  $L(j\omega)$  be a PSD matrix. Based on (25), we can write  $x^H \text{Re}[L(j\omega)]x = \sum_{j \in \mathcal{N}_i} \text{Re}[Y_{ij}(j\omega)]|x_i - x_j|^2 \geq 0$ . Since  $Y_{ij}(j\omega)$  is assumed to be strictly positive real, then  $\text{Re}[Y_{ij}(j\omega)]$  is non-negative  $\forall \omega \in \mathbb{R}$ . Thus, for all  $x \in \mathbb{C}^n$ ,  $x^H \text{Re}[L(j\omega)]x \geq 0$  and by Definition 4.3,  $\text{Re}[L(j\omega)]$  is a PSD matrix. Therefore, the dynamic Laplacian matrix  $L(j\omega)$  is also PSD matrix.

2) The real parts of the eigenvalues of  $L(j\omega)$  are non-negative with  $\lambda_1(L(j\omega)) = 0$ :

We can show  $\lambda_1(L(j\omega)) = 0$  with eigenvector  $\mathbf{1}_N$  as follows: Based on the definition of  $L(j\omega)$ , we can observe that the rows of  $L(j\omega)$  sum to zero, which implies that  $L(j\omega)x = 0$ . If all the entries of  $x$  are the same, then  $\mathbf{1}_N$  is the eigenvector of eigenvalue 0.

Since  $L(j\omega)$  is a PSD, then  $\text{Re}[x^H L(j\omega)x] \geq 0$  for all  $x \in \mathbb{C}^n$ . In particular, it is true for  $x = v_i$ , where  $v_i$  is the  $i$ th eigenvector of  $L(j\omega)$ ; thus, we have

$$\text{Re}[v_i^H L(j\omega)v_i] \geq 0. \quad (40)$$

Based on the definition of eigenvalues and eigenvectors ( $L(j\omega)v_i = \lambda_i v_i$ ), the above inequality can be written as:

$$\begin{aligned} \text{Re}[v_i^H \lambda_i v_i] &\geq 0 \\ \Rightarrow \text{Re}[\lambda_i v_i^H v_i] &\geq 0 \Rightarrow \text{Re}[\lambda_i \|v_i\|_2^2] \geq 0. \end{aligned} \quad (41)$$

Since  $\|v_i\|_2^2 > 0$ , then  $\text{Re}[\lambda_i] \geq 0, \forall i \in 1, \dots, n$ .

Now, we can show that there is no purely imaginary eigenvalue of  $L(j\omega)$ : Suppose that there exists a non-zero pure imaginary eigenvalue  $s = j\omega^*$  of  $L(j\omega)$  with associated eigenvector  $v^*$ . Then we can write the following quadratic form:

$$v^{*H} L(j\omega)v^* = \sum_{j \in \mathcal{N}_i} Y_{ij}(j\omega)|v_i^* - v_j^*|^2. \quad (42)$$

It follows from  $v$  being an eigenvector,

$$v^{*H} L(j\omega)v^* = v^{*H} (j\omega^* v^*) = j\omega^* \|v^*\|_2^2. \quad (43)$$

Combining (42) and (43) we get

$$j\omega^* \|v^*\|_2^2 = \sum_{j \in \mathcal{N}_i} Y_{ij}(j\omega^*)|v_i^* - v_j^*|^2, \quad (44)$$

By equating the real parts of the above equation, we can conclude that  $v_i^* = v_j^*, \forall i, j$  because  $Y_{ij}(j\omega^*)$  is strictly positive real. However, only eigenvalue with eigenvector  $\mathbf{1}_N$  is 0, which is a contradiction.

3) To prove this part, it will be useful to define the set  $\mathbb{S} = \{s : \text{Re}(s) \geq 0, s \neq 0\}$ .

Since  $L(0)$  is the Laplacian of a connected graph (see Definition 4.2), then it is obvious that  $s = 0$  is a distinct solution of  $\det[sI_n + L(s)] = 0$ . We can show that all node variables will reach consensus if we can show that no solution to (22) exists with modes in  $\mathbb{S}$ . Let  $\beta_i(s)$  be the least common multiple of the denominators of the  $i$ th row of  $L(s)$ , and let  $\beta(s) = \text{diag}[\beta_1(s) \ \beta_2(s) \ \dots \ \beta_m(s)]$ . Let  $\alpha(s) = \beta(s)L(s)$ . Note that both  $\beta(s)$  and  $\alpha(s)$  are polynomial matrices. Then it can be shown (see e.g. [27]) that the solutions to (22) are given by  $x(t)$  that satisfy

$$(s\beta(s) + \alpha(s))|_{s=\frac{d}{dt}} x(t) = 0.$$

By Lemma A.2, the allowable time domain modes of  $x(t)$  are given by the roots of  $\det(s\beta(s) + \alpha(s))$ . Note that since  $\beta(s)$  has no roots in the closed right half plane, the solutions to  $\det(s\beta(s) + \alpha(s)) = 0$  on  $\mathbb{S}$  are identical to the solutions of  $\det(sI_n + \beta^{-1}(s)\alpha(s)) = \det(sI_n + L(s)) = 0$ . By part 2, all the eigenvalues of  $L(s)$  have non-negative real parts, which implies that all the roots of  $\det[sI_n + L(s)] = 0$  have non-positive real-parts, so there are no solutions in  $s \in \mathbb{S}$ , which proves the claim. ■

As an example, consider the dynamic network of Fig. 2 with SPR transfer functions  $Y_{ij}(s) = \frac{1}{Z_{ij}(s)}$  assigned for the edges, and for simplicity consider  $Z_{ij} = R + j\omega L, R = 1\Omega$ , and  $L = 1H$ . The dynamic Laplacian  $L(s)$  can be written as

$$L(s) = \begin{bmatrix} \frac{2}{1+s} & -\frac{1}{\frac{1}{2}+s} & -\frac{1}{1+s} \\ -\frac{1}{1+s} & \frac{1}{1+s} & -\frac{1}{\frac{1}{2}+s} \\ -\frac{1}{1+s} & -\frac{1}{1+s} & \frac{1}{1+s} \end{bmatrix} \Rightarrow L(0) = \begin{bmatrix} 2 & -1 & -1 \\ -1 & 2 & -1 \\ -1 & -1 & 2 \end{bmatrix}.$$

From the above, it is readily verified that  $-L(0)$  is the Laplacian matrix of a connected graph. In fact,  $L(0)$  is a connected interconnection matrix (see Definition 4.1). Applying Theorem 4.7, we expect that all node variables converge to consensus. This is verified in Fig. 5, which shows a simple simulation of the dynamic consensus protocol (21), with initial conditions for the node potentials taken to be  $V(0) = [20, 60, 70]^T$ .

## V. BOUNDING THE EIGENVALUES OF THE DYNAMIC LAPLACIAN MATRIX

In this section, we develop a method to estimate the bounds of the eigenvalues of the dynamic Laplacian matrix  $L(j\omega)$  for all  $\omega$ . Our approach uses the idea of the *Dynamic Grounded Laplacian*. Motivated by the static case, we introduce the *Dynamic Grounded Laplacian* matrix to estimate lower and upper bounds for the real parts of the smallest and largest non-zero eigenvalues of the dynamic Laplacian matrix. We begin by defining the Dynamic Grounded Laplacian matrix. We then show how to obtain the lower and upper bounds for the eigenvalues of the dynamic Laplacian matrix.

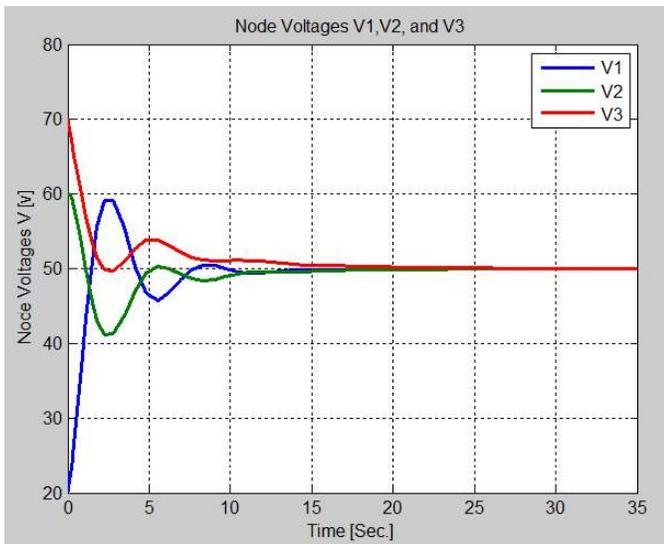


Fig. 5. Simulation of the node voltages of the dynamic network.

### A. Dynamic Grounded Laplacian Matrix

The Dynamic Grounded Laplacian matrix can be defined from the dynamic Laplacian matrix by extending the definition of the Grounded Laplacian matrix in the case of static networks (e.g., see [2] and [3]) to the case of dynamic networks.

To illustrate the idea, we will define the Dynamic Grounded Laplacian matrix for the undirected dynamic graph that represents a simple electrical network as shown in Fig. 2. As we have mentioned earlier, this network can be modeled as a dynamic graph by representing the capacitors on the vertices as integrator nodes and the transmission lines or the branches between the nodes as SPR transfer functions edges.

Using Kirchhoff’s law, it can be established that the dynamic Laplacian matrix  $L(j\omega)$  expresses a mapping from input currents to node potentials as

$$\begin{bmatrix} I_1^{in}(j\omega) \\ I_2^{in}(j\omega) \\ I_3^{in}(j\omega) \end{bmatrix} = L(j\omega) \begin{bmatrix} V_1(j\omega) \\ V_2(j\omega) \\ V_3(j\omega) \end{bmatrix}, \quad (45)$$

where  $L(j\omega)$  is defined by (26), where  $Y_{ij}(j\omega)$  is the admittance between nodes  $i$  and  $j$ .

For a given dynamic network, assume that the node  $V_1$  is connected to the ground (this node is colored red in the electrical network shown in Fig. 2). Grounding node  $V_1$  means that the potential and the current of this node are zeros ( $V_1(j\omega) = 0$  and  $I_1^{in}(j\omega) = 0$ ).

Substituting  $V_1(j\omega) = 0$  and  $I_1^{in}(j\omega) = 0$  in the equation (45), we get

$$\begin{bmatrix} I_2^{in}(j\omega) \\ I_3^{in}(j\omega) \end{bmatrix} = \underbrace{\begin{bmatrix} Y_{12}(j\omega) + Y_{23}(j\omega) & -Y_{23}(j\omega) \\ -Y_{23}(j\omega) & Y_{13}(j\omega) + Y_{23}(j\omega) \end{bmatrix}}_{L_0(j\omega)} \begin{bmatrix} V_2(j\omega) \\ V_3(j\omega) \end{bmatrix}. \quad (46)$$

We call  $L_0(j\omega)$ , defined in this way, a Dynamic Grounded Laplacian matrix. We give this matrix this name because it is

defined from the dynamic Laplacian by grounding the subset nodes  $\mathbf{V}_0 = V_1$  in this case.

These ideas can be generalized: Given a dynamic graph  $\mathbf{G} = (\mathbf{V}, \mathbf{E})$ , where  $\mathbf{V}$  is a set of  $n$  nodes;  $\mathbf{E} \subset \mathbf{V} \times \mathbf{V}$  is a set of  $m$  edges, and given a subset of grounded nodes  $\mathbf{V}_0 \subset \mathbf{V}$  consisting of  $n_g < n$  nodes,  $n_0 = n - n_g$ , then the *Dynamic Grounded Laplacian* matrix  $L_0(j\omega) \in \mathbb{C}^{n_0 \times n_0}$  is defined from the dynamic Laplacian matrix  $L(j\omega)$  by removing all rows and columns corresponding to the nodes in  $\mathbf{V}_0$ . Note that this must be done in such a way that  $L_0(j\omega)$  corresponds to a connected graph. The following lemma describes the properties of the dynamic Grounded Laplacian  $L_0(j\omega)$ .

**Lemma 5.1:** For an undirected, connected dynamic graph with SPR edges, the Dynamic Grounded Laplacian matrix  $L_0(j\omega) \in \mathbb{C}^{n_0 \times n_0}$  is a complex, symmetric, positive definite (CSPD) matrix that is always invertible for all  $\omega$ .

**Proof:**

Since the dynamic graph is undirected, the dynamic Laplacian matrix is symmetric  $Y_{ij}(j\omega) = Y_{ji}(j\omega)$  and then  $L_0(j\omega)$  is a complex symmetric (CS) matrix.

Now, we can show  $L_0(j\omega)$  is PD by examining the positive definiteness of the Hermitian part of  $L_0(j\omega)$  (see Definition 4.3),

$$\begin{aligned} H(L_0(j\omega)) &= \frac{1}{2}(L_0(j\omega) + L_0(j\omega)^H) \\ &= \frac{1}{2}(L_0(j\omega) + \bar{L}_0(j\omega)) = \text{Re}[L_0(j\omega)]. \end{aligned}$$

Based on Lemma 4.5, the positive definiteness of  $L_0(j\omega)$  is due to  $x^T \text{Re}[L_0(j\omega)]x = \sum_{j \in \mathcal{N}_i^0} \text{Re}[Y_{ij}(j\omega)]|x_i - x_j|^2 > 0$  for all  $x \in \mathbb{R}^{n_0}$ , where  $\mathcal{N}_i^0$  is the set of neighborhood nodes to the node  $i$  in the remaining graph after grounding. Thus, the Dynamic Grounded Laplacian matrix  $L_0(j\omega)$  is always invertible for all  $\omega$ . ■

### B. Finding Bounds for the Eigenvalues of the Dynamic Laplacian Matrix

In this subsection, the bounds for the eigenvalues of the dynamic Laplacian are estimated using the properties of the dynamic Laplacian and Dynamic Grounded Laplacian matrices.

1) *Lower Bound for the Smallest Non-Zero Eigenvalue of the Dynamic Laplacian Matrix:* The following theorem provides a lower bound of the dynamic Laplacian matrix. Later this bound will be used to characterize the stability of the closed loop system in the case of distributed controllers.

**Theorem 5.2:** Let  $L(j\omega) \in \mathbb{C}^{n \times n}$  be the dynamic Laplacian matrix for an undirected dynamic graph with SPR edges. Let  $\lambda_i(L(j\omega))$  and  $\lambda_i(L_0(j\omega))$  denote the eigenvalues of  $L(j\omega)$  and  $L_0(j\omega)$ , respectively, for any choice of a boundary set of grounded nodes  $\mathbf{V}_0$ , for which the graph is connected. Assume that  $\text{Re}[\lambda_1(L(j\omega))] \leq \text{Re}[\lambda_2(L(j\omega))] \leq \dots \leq \text{Re}[\lambda_n(L(j\omega))]$  and  $\text{Re}[\lambda_1(L_0(j\omega))] \leq \text{Re}[\lambda_2(L_0(j\omega))] \leq \dots \leq \text{Re}[\lambda_{n_0}(L_0(j\omega))]$ ,  $n_0 < n$ . Then a lower bound for the real part of the smallest non-zero eigenvalue of the dynamic Laplacian is given by

$$\min_{\omega} \text{Re}[\lambda_2(L(j\omega))] \geq \frac{1}{\|\text{trace}((\text{Re}[L_0(j\omega)])^{-1})\|_{\infty}}. \quad (47)$$



**Proof:**

The proof of this Theorem is given in the following steps:

- 1) We begin the proof by finding lower bounds for the eigenvalues of the Hermitian part of a Dynamic Grounded Laplacian matrix  $\lambda_i(H(L_0(j\omega)))$ :

Based on Lemma A.3, these bounds can be obtained as

$$\lambda_i(H(L_0(j\omega))) \geq \frac{1}{\text{trace}((\text{Re}[L_0(j\omega)])^{-1})}, \quad (48)$$

for all  $\omega, i \in 1, 2, \dots, n$ .

- 2) In this step, we derive the inequality that relates the eigenvalues of the matrices  $H(L_0(j\omega))$  and  $H(L(j\omega))$ : Applying the interlacing eigenvalue Theorem A.4, we can conclude that the eigenvalues of  $H(L(j\omega)) \in \mathbb{R}^{n \times n}$  and  $H(L_0(j\omega)) \in \mathbb{R}^{n_0 \times n_0}$  are interlaced for  $i = 1, 2, \dots, n_0$ , as

$$\begin{aligned} \lambda_i(H(L(j\omega))) &\leq \lambda_i(H(L_0(j\omega))) \\ &\leq \lambda_{i+(n-n_0)}(H(L(j\omega))). \end{aligned} \quad (49)$$

Combining (48) and (49), we get

$$\lambda_i(H(L(j\omega))) \geq \frac{1}{\text{trace}((\text{Re}[L_0(j\omega)])^{-1})}, \quad (50)$$

for all  $\omega$  and  $i = (n - n_0) + 1, \dots, n$ .

- 3) Here, we derive the relationship between the real part of the smallest and largest nonzero eigenvalues of  $L(j\omega)$  and  $H(L(j\omega))$ . This relationship can be achieved using Theorem A.5 as

$$\sum_{i=n-k}^n \text{Re}[\lambda_i(L(j\omega))] \leq \sum_{i=n-k}^n \lambda_i(H(L(j\omega))), \quad (51)$$

$k = 0, 1, \dots, n - 1$ , with equality for  $k = n - 1$ .

Based on Mirsky Theorem A.5, the relationship between the eigenvalues of the matrices  $L(j\omega)$  and  $H(L(j\omega))$  can be obtained by substituting  $k = n - 3, k = n - 1$ , and  $k = 0$  in (51) and then combining the result. Thus, the inequalities that characterize the relationship between the eigenvalues of these matrices are given by Lemma A.6 as

$$\text{Re}[\lambda_2(L(j\omega))] \geq \lambda_2(H(L(j\omega))) \quad (52)$$

and

$$\text{Re}[\lambda_n(L(j\omega))] \leq \lambda_n(H(L(j\omega))), \quad (53)$$

where  $\text{Re}[\lambda_2(L(j\omega))]$  and  $\text{Re}[\lambda_n(L(j\omega))]$  are the real part of the smallest and largest nonzero eigenvalues of  $L(j\omega)$  and  $H(L(j\omega))$ , respectively. Note that these result are obtained for the case of a connected dynamic graph with SPR edges, which is mean  $\text{Re}[\lambda_1(L(j\omega))] = \lambda_1(H(L(j\omega))) = 0$ .

- 4) Finally, a lower bound for the real part of the smallest non-zero eigenvalue of  $L(j\omega)$  can be obtained by combining (50) and (52) as

$$\text{Re}[\lambda_2(L(j\omega))] \geq \frac{1}{\text{trace}((\text{Re}[L_0(j\omega)])^{-1})}, \forall \omega. \quad (54)$$

Taking the minimum values of both sides in the above inequality over all  $\omega$ , yields

$$\min_{\omega} \text{Re}[\lambda_2(L(j\omega))] \geq \min_{\omega} \left[ \frac{1}{\text{trace}((\text{Re}[L_0(j\omega)])^{-1})} \right];$$

$$\min_{\omega} \text{Re}[\lambda_2(L(j\omega))] \geq \frac{1}{\max_{\omega} [\text{trace}((\text{Re}[L_0(j\omega)])^{-1})]}. \quad (55)$$

Therefore, a lower bound for the real part of the smallest non-zero eigenvalue of the dynamic Laplacian matrix is given by

$$\min_{\omega} \text{Re}[\lambda_2(L(j\omega))] \geq \frac{1}{\|\text{trace}((\text{Re}[L_0(j\omega)])^{-1})\|_{\infty}}. \quad (56)$$

■

2) *An Upper Bound for the Largest Eigenvalue of the Dynamic Laplacian Matrix:* The following theorem provides an upper bound for the real part of the largest eigenvalue of the dynamic Laplacian matrix:

*Theorem 5.3:* Let  $L(j\omega) \in \mathbb{C}^{n \times n}$  be the dynamic Laplacian matrix for an undirected dynamic graph with SPR edges. Let  $\lambda_i(L(j\omega))$  denote the eigenvalues of  $L(j\omega)$ , ordered so that  $\text{Re}[\lambda_1(L(j\omega))] \leq \text{Re}[\lambda_2(L(j\omega))] \leq \dots \leq \text{Re}[\lambda_n(L(j\omega))]$ . Then an upper bound for the real part of the largest eigenvalue of the dynamic Laplacian matrix is given by

$$\max_{\omega} \text{Re}[\lambda_n(L(j\omega))] \leq 2 \max_i \|\text{Re}[D(j\omega)(i, i)]\|_{\infty}, \quad (57)$$

where  $D(j\omega)(i, i)$  is the  $i$ th diagonal element of the dynamic Degree matrix of the dynamic graph.

**Proof:**

First, we show that the  $H[2D(j\omega) - L(j\omega)]$  is a PSD matrix for all  $\omega$ . Because of the positivity realness of the edges and the symmetry property (because the assumed dynamic graph uses undirected topology), we can write

$$\begin{aligned} H[2D(j\omega) - L(j\omega)] &= H[2D(j\omega) - (D(j\omega) - A(j\omega))] \\ &= H[D(j\omega) + A(j\omega)] = \text{Re}[D(j\omega) + A(j\omega)], \end{aligned} \quad (58)$$

where  $D(j\omega)$  and  $A(j\omega)$  are the dynamic degree and adjacency matrix of the dynamic graph. In addition, the symmetry property of the matrix  $[D(j\omega) + A(j\omega)]$  can be easily verified from the following definition:  $[D(j\omega) + A(j\omega)] = [l_{ij}(j\omega)]$ , where

$$l_{ij}(j\omega) = \begin{cases} \sum_{j \in \mathcal{N}_i} Y_{ij}(j\omega) & i = j \\ +Y_{ij}(j\omega) & i \neq j \text{ and } (i, j) \in \mathcal{E} \\ 0 & \text{otherwise.} \end{cases} \quad (59)$$

Note that the matrix defined above is different from the dynamic Laplacian matrix  $L(s)$ . The difference is the off-diagonal terms are positive.

By Lemma 4.5, the necessary and sufficient condition for the matrix  $H(2D(j\omega) - L(j\omega)) = \text{Re}[D(j\omega) + A(j\omega)]$  to be a PSD matrix is that the real part of  $[D(j\omega) + A(j\omega)]$  be a PSD matrix.



Based on Definition 4.3 and (59), we can write

$$x^T \text{Re}[D(j\omega) + A(j\omega)]x = \sum_{j \in \mathcal{N}_i} \text{Re}[Y_{ji}(j\omega)](x_i + x_j)^2 \geq 0,$$

for all vectors  $x \in \mathbb{R}^n$ . Because of the strictly positive realness of the edges, then  $\text{Re}[Y_{ji}(j\omega)]$  is non-negative  $\forall \omega \in \mathbb{R}$ . Thus, by Definition 4.3,  $\text{Re}[D(j\omega) + A(j\omega)]$  is a PSD matrix. Therefore, the matrix  $H[2D(j\omega) - L(j\omega)]$  is also a PSD matrix. Hence,

$$H[2D(j\omega) - L(j\omega)] \geq 0 \Leftrightarrow H(L(j\omega)) \leq 2H(D(j\omega)). \quad (60)$$

From the above inequality, we can write

$$\lambda_i(H(L(j\omega))) \leq 2\lambda_i(H(D(j\omega))) = 2\text{Re}[D(j\omega)(i, i)]. \quad (61)$$

Thus, the largest eigenvalue of the Hermitian part of the dynamic Laplacian can be upper-bounded by

$$\begin{aligned} \lambda_n(H(L(j\omega))) &\leq 2 \max_i (\max_{\omega} \text{Re}[D(j\omega)(i, i)]) \\ \Leftrightarrow \lambda_n(H(L(j\omega))) &\leq 2 \max_i \|\text{Re}[D(j\omega)(i, i)]\|_{\infty}. \end{aligned} \quad (62)$$

Combining (53) and (62), we get

$$\text{Re}[\lambda_n(L(j\omega))] \leq 2 \max_i \|\text{Re}[D(j\omega)(i, i)]\|_{\infty}. \quad (63)$$

Thus,

$$\max_{\omega} \text{Re}[\lambda_n(L(j\omega))] \leq 2 \max_i \|\text{Re}[D(j\omega)(i, i)]\|_{\infty}.$$

The lower and upper bounds for the eigenvalues of the dynamic Laplacian matrix are useful for stability analysis of dynamic consensus networks under feedback control. In the next section we show how these bound can be used to establish the stability of an undirected dynamic consensus network under distributed control.

## VI. STABILITY ANALYSIS OF DYNAMIC CONSENSUS NETWORKS WITH DISTRIBUTED CONTROLLERS

In this section, we provide conditions for the stability of undirected dynamic networks controlled using distributed controllers. The bounds of the eigenvalues of the dynamic Laplacian matrix  $\lambda_i(L(j\omega))$ , which can be found using the methodology from the previous section, can be used to analyze the stability of such dynamic networks using the Nyquist graphical-stability test [12]. The authors of [7] and [12] applied tools from graph theory to relate the topology of a static network to the stability of the overall system. They also provided a Nyquist criterion, which uses the eigenvalues of the static Laplacian matrix to determine the effect of the static graph on stability. Here, we extend this stability criterion from static networks to dynamic networks. Before analyzing the stability of the dynamic consensus networks, we will formulate our problem for the proposed dynamic network (undirected dynamic consensus network with identical LTI nodes and dynamic edges).

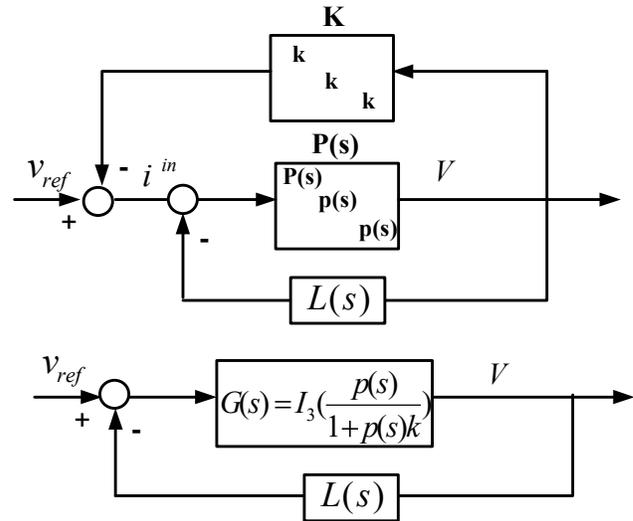


Fig. 6. Upper figure: Block diagram of the controlled system. Lower figure: Equivalent standard feedback diagram.

### A. Problem Formulation

Consider a dynamic network with  $N$  identical nodes and with dynamic edges  $e_{ij}$  (here, we assume the dynamics of each edge is a SPR transfer function), as in Fig. 2, where the voltage of the  $i$ th node is represented as  $v_i, i = 1, \dots, N$ . Furthermore, we assume that the nodes have identical dynamics  $p(s)$  and controllers  $k(s)$ .

Consider the dynamic network Fig. 2 as defined in (20) and let the integrator nodes be replaced by a common transfer function such as  $p_i(s) = p(s)$ . For example, assume that the integrator nodes can be replaced by filters with a common transfer function  $p_i(s) = p(s) = \frac{\beta}{s+\alpha}$ . Thus, the dynamics of each node can be rewritten in the time domain as

$$\frac{1}{\beta}(\dot{v}_i(t) + \alpha v_i(t)) = i_i^{in}(t) - \sum_{j \in \mathcal{N}_i} y_{ij}(t) * (v_i(t) - v_j(t)), \quad (64)$$

$\forall i = 1, \dots, N$ . Now suppose the node voltage  $v_i$  of each node is controlled by a static controller  $k$ , that determines the input  $i_i^{in}(t)$ . Then the overall system can be expressed in the frequency domain as

$$\begin{aligned} P(s)^{-1}V(s) &= I^{in}(s) - L(s)V(s) \\ I^{in}(s) &= v^{ref} - KV(s), \end{aligned} \quad (65)$$

where  $P(s) = \text{diag}(p(s), p(s), \dots, p(s))$ ,  $K = \text{diag}(k(s), k(s), \dots, k(s))$ , where  $k(s) = k$  and  $V(s), I^{in}(s)$  are the Laplace transform of the vectors  $v(t)$  and  $i^{in}(t)$ , respectively. The control scheme of this problem can be depicted as shown in the upper part of Fig. 6 (for clarity, shown for  $N = 3$ ). We will apply the Nyquist criterion for this block diagram for our controller design, using the estimated bounds of the eigenvalues of the dynamic Laplacian matrix. Thus, the upper block diagram is first transformed into an equivalent standard feedback diagram as shown in the part of Fig. 6. Note that the identical controller  $k(s) = k$  in the equivalent standard feedback diagram is located in the forward path ( $G(s)$ ).



The closed-loop transfer function  $H(s)$  from  $v^{ref}(s)$  to  $V(s)$  can be expressed as:

$$H(s) = G(s)(I_3 + G(s)L(s))^{-1}, \quad (66)$$

where  $G(s) = I_N \frac{p(s)}{1+p(s)k}$ .

The feedback matrix  $(L(s))$  in the equivalent standard feedback diagram of the controlled dynamic network in the lower part of Fig. 6 is a dynamic matrix, whereas this matrix is a constant in the static case. This leads us to proceed with bounding the eigenvalues of the dynamic Laplacian matrix in order to locate them for analyzing the stability of the closed loop system using the Nyquist criterion. These bounds can be predicted for static networks based on properties of the graph [7]. The authors of [7] considered locations of the eigenvalues of the Laplacian matrix to investigate the stability of a consensus network. The region of reciprocal eigenvalues of the Laplacian matrix can be defined using the estimated regions of the smallest and largest real parts of the non-zero eigenvalues of the dynamic Laplacian matrix as introduced in the previous section. Using these estimated regions, the stability conditions of the closed loop system  $H(j\omega)$  can be analyzed with respect to the eigenvalues of  $L(j\omega)$  ( $\lambda_i(L(j\omega))$ ) as we will see in the next section.

### B. Stability Analysis

Here we analyze the stability of the closed-loop system  $H(s)$  with respect to the eigenvalues of  $L(j\omega)$  ( $\lambda_i(L(j\omega))$ ). Using Nyquist's graphical-stability test, the stability of the closed loop system (66) with identical node dynamics  $p(s)$  and controller  $k(s) = k$  can be analyzed as follows:

By Theorem 2 in [10], it can be seen that closed loop system  $H(j\omega)$  in (66) is stable if and only if the Nyquist plot of

$$\prod_{i=1}^N [1 + \lambda_i(L(j\omega))G(j\omega)] \quad (67)$$

makes  $m$  c.c.w encirclements of the origin, where  $m$  is the number of unstable poles of  $G(j\omega)L(j\omega)$ , and

$$G(j\omega) = I_N \frac{p(j\omega)}{1 + p(j\omega)k}.$$

Now consider the case when  $G(j\omega)$  and  $L(j\omega)$  are stable. Then the stability analysis condition stated above is equivalent to the closed loop system (66) is stable if the Nyquist plot of

$$\prod_{i=1}^N [1 + \lambda_i(L(j\omega))G(j\omega)]$$

makes zero encirclements around the origin, which is equivalent to the Nyquist plot of

$$\lambda_i(L(j\omega))G(j\omega)$$

makes zero encirclements around the  $-1 + 0j$ , for all  $i$ .

Furthermore, the above stability condition is equivalent to the closed loop system  $H(j\omega)$  is stable if the Nyquist plot of a stable transfer function  $G(j\omega)$  does not encircle the points  $\frac{-1}{\lambda_i(L(j\omega))}$  for all  $\omega$ , and  $i = 2, \dots, n$  (see [12] in the case of static Laplacian matrix). Thus, satisfying this statement

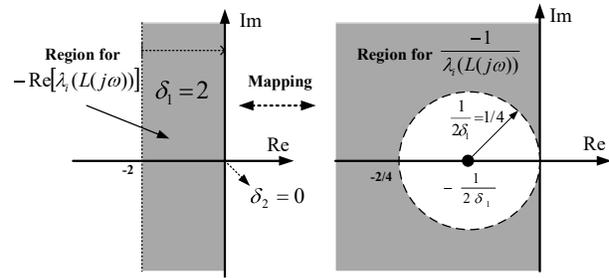


Fig. 7. Left figure: Inclusion region of  $-\text{Re}[\lambda_i(L(j\omega))]$ . Right figure: Inclusion region of  $-\frac{1}{\lambda_i(L(j\omega))}$ .

means that the closed loop system  $H(j\omega)$  is stable and then all the eigenvalues of  $H(j\omega)$  are in the left half of the complex plane. Note that the stability of the system does not mean the system will achieve consensus. Thus, we require an additional condition for guaranteeing consensus, which is the connectivity of the graph that is associated with the static Laplacian matrix  $L(0)$ , because under the connectivity condition, the zero eigenvalue of the dynamic Laplacian is distinct and then we can guarantee consensus.

In order to apply the above conditions to designing a controller  $k(s)$  such that the closed loop system  $H(s)$  is stable, the bounds of the eigenvalues of the dynamic Laplacian must be estimated. These bounds are estimated in Section V-B and given by (47) and (57).

### C. Illustrative Simulations

To illustrate how our approach can be used for the design of controllers, consider the dynamic network Fig. 2 with the dynamic Laplacian  $L(s)$  defined by (26) with parameters  $Y_{12}(s) = \frac{1}{4s+4}$  and  $Y_{13} = Y_{23}(s) = \frac{1}{2s+2}$ . Using these parameters, the bounds for the real parts of all non-zero eigenvalues of  $L(j\omega)$ ,  $-\text{Re}[\lambda_i(L(j\omega))]$  can be estimated by (47) and (57) as

$$\begin{aligned} & [-\max_{\omega} \text{Re}[\lambda_n(L(j\omega))], -\min_{\omega} \text{Re}[\lambda_2(L(j\omega))]] \subset \\ & [-2 \max_i \| \text{Re}[D(j\omega)(i, i)] \|_{\infty}, \frac{-1}{\| \text{trace}((\text{Re}[L_0(j\omega)])^{-1}) \|_{\infty}}] \\ & = [-2, 0]. \end{aligned} \quad (68)$$

Using the above estimated bounds, the region containing negative of the eigenvalues of the dynamic Laplacian matrix  $L(j\omega)$  indicated as the shaded region on the left side of Fig. 7. For the stability analysis described above, we need to define the region of  $\frac{-1}{\lambda_i(L(j\omega))}$  for all  $\omega$ ,  $i = 1 \dots N$ . Therefore, the bound  $-\max_{\omega} \text{Re}[\lambda_3(L(j\omega))] \geq -4$  is mapped to the exterior of a circle centered at  $(-1/4, 0)$  with radius  $1/4$ . Furthermore, the bound  $-\min_{\omega} \text{Re}[\lambda_2(L(j\omega))] \leq 0$  is mapped to the left half of the complex plane. Thus, the region of  $\frac{-1}{\lambda_i(L(j\omega))}$  is the intersection between the two mapped regions, which is indicated as the shaded region on the right side of Fig. 7. That is a conservative estimate; all the points  $-\frac{1}{\lambda_i(L(j\omega))}$  lie in this shaded region. Thus if the Nyquist plot of  $G(j\omega)$  does not enter this region, we can be sure consensus occurs.

For this example, assume the nodes of the given dynamic network have a common stable transfer function  $p(s) = \frac{\beta}{s+\alpha}$ ,

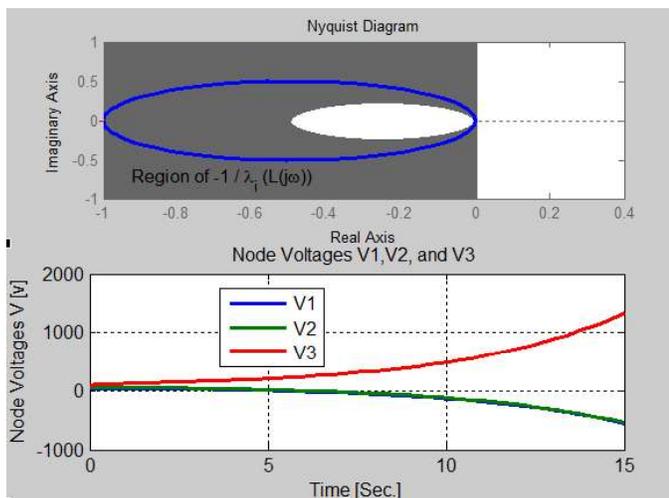


Fig. 8. Upper figure: Nyquist plot of  $G(j\omega)$  (blue curve) with the region of  $\frac{-1}{\lambda_i(L(j\omega))}$  (shaded region). Lower figure: Simulation of the node voltages in the dynamic network.

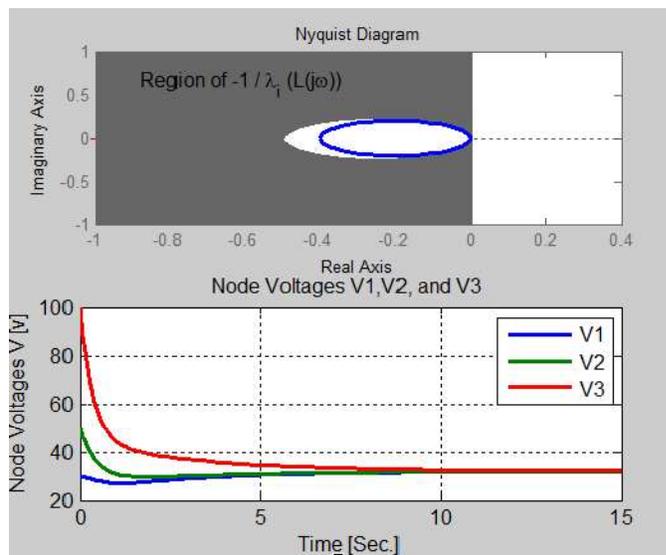


Fig. 9. Upper figure: Nyquist plot of  $G(j\omega)$  (blue curve) with the region of  $\frac{-1}{\lambda_i(L(j\omega))}$  (shaded region). Lower figure: Simulation of the node voltages in the dynamic network.

$\beta = -1, \alpha = 2$ . Further, assume a simple static controller  $k(s) = k$ . Then the stability of the closed loop system discussed above and the estimated bounds can be verified using the Nyquist plot of  $G(j\omega)$  with respect to the defined region of  $\frac{-1}{\lambda_i(L(j\omega))}$  by considering the following two cases:

**Case 1:** With a common static controller  $k = 1$ , the Nyquist plot of  $G(j\omega) = I_3 \frac{p(j\omega)}{1+p(j\omega)k}$  (blue curve as shown in the upper side of Fig. 8) does encircle (i.e., intersects) one of the points  $\frac{-1}{\lambda_i(L(j\omega))}$ . Thus, the closed loop system  $H(j\omega)$  in (66) is unstable as shown in the lower side of Fig. 8.

**Case 2:** If we change the static controller to  $k = -0.5$ , then the Nyquist plot of  $G(j\omega) = I_3 \frac{p(j\omega)}{1+p(j\omega)k}$  (blue curve as shown in the upper side of Fig. 9) does not encircle any point  $\frac{-1}{\lambda_i(L(j\omega))}$  because it does not intersect the shaded region. Thus, the closed loop system is stable as shown in the lower side of Fig. 9.

Thus, we can conclude that  $k(s)$  stabilizes the closed loop system  $H(j\omega)$  if the number of net encirclements of the points  $\frac{-1}{\lambda_i(L(j\omega))}$  by the Nyquist plot of the stable transfer function  $G(j\omega)$  is zero for all nonzero  $\lambda_i(H(j\omega))$ . In other words, the stability of the closed-loop system  $H(j\omega)$  can be investigated from the locations of the eigenvalues of the dynamic Laplacian matrix  $L(j\omega)$ , which was estimated in the previous Section, in relation to a Nyquist plot of stable  $G(j\omega)$ .

In order to design a controller for the proposed dynamic network (identical LTI systems with dynamic edges) that causes all states of nodes to achieve consensus, the controller  $k(s)$  must be chosen such that the closed-loop system  $H(j\omega)$  is stable or, ideally, reaches consensus. In other words, we must avoid the net encirclement of  $-\frac{1}{\lambda_i(L(s))}$  by the Nyquist plot of the stable transfer function  $G(j\omega)$  for all nonzero, as was in the second case. For consensus investigation, we require an additional condition, which is the connectedness of  $L(0)$  as was previously mentioned, meaning that the zero eigenvalue is distinct.

## VII. CONCLUSION

Motivated by several physical examples, including modeling micro-grids of power systems as undirected dynamic consensus network that we presented, we considered dynamic networks whose nodes are integrators and whose edges are SPR transfer functions representing dynamical systems that couple the nodes. We presented a framework 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. From this, equivalent concepts for dynamic interconnection matrices and dynamic consensus networks were defined. We showed that the strictly positive realness of the edges is a sufficient condition for these kinds of dynamic networks to reach a consensus. We developed a method to estimate the bounds of the eigenvalues of the dynamic Laplacian matrix. Our approach used the idea of the *Dynamic Grounded Laplacian*. These bounds were used for a Nyquist stability analysis of a closed loop system comprising an undirected dynamic network subject to distributed control. The region of reciprocal eigenvalues of the Laplacian matrix was defined using the prediction regions of the smallest and largest real parts of the non-zero eigenvalues of the dynamic Laplacian matrix. Using these prediction-regions, the stability conditions of the closed loop system was analyzed with respect to the eigenvalues of the dynamic Laplacian matrix. In future work, we extend these notions to consider the stability and consensus for a dynamic consensus networks with non-identical nodes and dynamic edges.

## ACKNOWLEDGMENTS

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



## REFERENCES

- [1] H. Ando, Y. Oasa, I. Suzuki, and M. Yamashita. Distributed memoryless point convergence algorithm for mobile robots with limited visibility. *IEEE Transactions on Robotics and Automation*, 15(5):818–828, 1999.
- [2] P. Barooah and J.P. Hespanha. Graph effective resistance and distributed control: Spectral properties and applications. In *Proceedings of 45th IEEE Conference on Decision and Control*, pages 3479–3485, dec. 2006.
- [3] P. Barooah and J.P. Hespanha. Estimation on graphs from relative measurements. *IEEE Control Systems*, 27(4):57–74, aug. 2007.
- [4] J. Cortes, S. Martinez, T. Karatas, and F. Bullo. Coverage control for mobile sensing networks. *IEEE Transactions on Robotics and Automation*, 20(2):243–255, 2004.
- [5] M.H. DeGroot. Reaching a consensus. *Journal of the American Statistical Association*, pages 118–121, 1974.
- [6] F. Dorfler and F. Bullo. Synchronization and transient stability in power networks and non-uniform kuramoto oscillators. In *Proceedings of 2010 American Control Conference*, pages 930–937. IEEE, 2010.
- [7] J. Alexander Fax and Richard M. Murray. Information flow and cooperative control of vehicle formations. *IEEE transactions on automatic control*, 49(9):1465–1476, 2004.
- [8] J.A. Fax and R.M. Murray. Information flow and cooperative control of vehicle formations. *IEEE Transactions on Automatic Control*, 49(9):1465–1476, 2004.
- [9] R.A. Freeman, P. Yang, and K.M. Lynch. Distributed estimation and control of swarm formation statistics. In *Proceedings of 2006 American Control Conference*, pages 7–13. IEEE, 2006.
- [10] A. Gattami and R. Murray. A frequency domain condition for stability of interconnected mimo systems. In *Proceedings of the American Control Conference*, volume 4, pages 3723–3728 vol.4, 30 2004–july 2 2004.
- [11] G.H. Golub and C.F. Van Loan. *Matrix Computations*. Johns Hopkins Univ Pr, 1996.
- [12] S. Hara, T. Hayakawa, and H. Sugata. Stability analysis of linear systems with generalized frequency variables and its applications to formation control. In *Proceedings of 46th IEEE Conference on Decision and Control*, pages 1459–1466, Dec. 2007.
- [13] R. A. Horn and C. R. Johnson. *Topics in Matrix Analysis*. Cambridge University Press, 1991.
- [14] F. Lashhab. Dynamic consensus networks: spectral properties, consensus, and control. *Colorado School of Mines. Arthur Lakes Library*, 2012.
- [15] R. A. Horn and C. R. Johnson. *Matrix Analysis*. University Press., Cambridge, 1993.
- [16] K. Kyo. Decentralized control of multi-agent systems: Theory and applications. *PhD Dissertation, Gwangju Institute of Technology, Gwangju, South Korea*, 2012.
- [17] G. Lafferriere, A. Williams, J. Caughman, and JJP Veerman. Decentralized control of vehicle formations. *Systems and Control letters*, 54(9):899–910, 2005.
- [18] Fadel Lashhab, Kevin Moore, Tyrone Vincent, Deyuan Meng, and Khalid Kuwairi. Robust  $H_\infty$  controller design for dynamic consensus networks. *International Journal of Control*, (just-accepted):1–20, 2017.
- [19] J. Lin, AS Morse, and BDO Anderson. The multi-agent rendezvous problem. In *Proceedings of the 42nd IEEE Conference on Decision and Control*, volume 2, pages 1508–1513. IEEE, 2003.
- [20] Z. Lin, B. Francis, and M. Maggiore. Necessary and sufficient graphical conditions for formation control of unicycles. *IEEE Transactions on Automatic Control*, 50(1):121–127, 2005.
- [21] E. Mallada and Ao Tang. Improving damping of power networks: Power scheduling and impedance adaptation. In *Proceedings of 50th IEEE Conference on Decision and Control and European Control Conference*, pages 7729–7734, dec. 2011.
- [22] 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*, 18(1), 2011.
- [23] R. Olfati-Saber, J.A. Fax, and R.M. Murray. Consensus and cooperation in networked multi-agent systems. *Proceedings of the IEEE*, 95(1):215–233, 2007.
- [24] R. Olfati-Saber and R.M. Murray. Consensus problems in networks of agents with switching topology and time-delays. *IEEE Transactions on Automatic Control*, 49(9):1520–1533, 2004.
- [25] R. Olfati-Saber and J.S. Shamma. Consensus filters for sensor networks and distributed sensor fusion. In *Proceedings of 44th IEEE Conference on Decision and Control and European Control Conference*, pages 6698–6703. IEEE, 2005.
- [26] A. Papachristodoulou and A. Jadbabaie. Synchronization in oscillator networks: Switching topologies and non-homogeneous delays. In *Proceedings of 44th IEEE Conference on Decision and Control European Control Conference*, pages 5692–5697. IEEE, 2005.
- [27] J.W. Polderman and J.C. Willems. *Introduction to Mathematical Systems Theory: a Behavioral Approach*. Springer Verlag, 1998.
- [28] O. Egeland R. Lozano, B. Brogliato and B. Maschke. *Dissipative Systems Analysis and Control Theory and Application*. Springer, 2000.
- [29] W. Ren, R.W. Beard, and E.M. Atkins. Information consensus in multi-vehicle cooperative control. *IEEE Control Systems*, 27(2):71–82, 2007.
- [30] B.M. Sanandaji, T.L. Vincent, and M.B. Wakin. Exact topology identification of large-scale interconnected dynamical systems from compressive observations. In *Proceedings of American Control Conference*, pages 649–656. IEEE, 2011.
- [31] H.G. Tanner, A. Jadbabaie, and G.J. Pappas. Stable flocking of mobile agents part i: Dynamic topology. In *Proceedings of the 42nd IEEE Conference on Decision and Control*, volume 2, pages 2016–2021. IEEE, 2003.
- [32] H.G. Tanner, A. Jadbabaie, and G.J. Pappas. Stable flocking of mobile agents, part i: Fixed topology. In *Proceedings of the 42nd IEEE Conference on Decision and Control*, volume 2, pages 2010–2015. IEEE, 2003.
- [33] J. Wang and N. Elia. Consensus over networks with dynamic channels. In *Proceedings of 2008 American Control Conference*, pages 2637–2642. IEEE, 2008.
- [34] L. Xiao, S. Boyd, and S. Lall. A scheme for robust distributed sensor fusion based on average consensus. In *Proceedings of Fourth International Symposium on Information Processing in Sensor Networks*, pages 63–70. IEEE, 2005.

## APPENDIX

The proof of Theorem 4.7 depends on a characterization of solutions for systems defined using a behavioral approach.

**Lemma A.1 (Gershgorin circle theorem):** (see. e.g. [11]) Let  $A \in \mathbb{C}^{n \times n}$  with elements  $a_{ij}$ . Let  $R_i = \sum_{j \neq i} |a_{ij}|$  and define  $D(a_{ii}, R_i)$  to be the closed disk centered at  $a_{ii}$  with radius  $R_i$ . Then every eigenvalue of  $A$  lies within at least one of the disks  $D(a_{ii}, R_i)$ .

**Lemma A.2 ([27]):** Given  $m \times m$  polynomial matrix  $P(s)$ , let  $\lambda_i, i = 1, \dots, N$  be the distinct roots of  $\det P(s)$  with multiplicity  $n_i$ . Then the solutions to

$$P\left(\frac{d}{dt}\right)x(t) = 0$$

are of the form

$$x(t) = \sum_{i=1}^N \sum_{j=0}^{n_i-1} \alpha_{ij} t^j e^{\lambda_i t}$$

where  $\alpha_{ij}$  are constants.

**Lemma A.3:** Let the dynamic Laplacian matrix  $L(j\omega) \in \mathbb{C}^{n \times n}$  and a Dynamic Grounded Laplacian  $L_0(j\omega) \in \mathbb{C}^{n_0 \times n_0}$  be given, with  $n_0 = n - n_g < n$ . For any choice of a boundary set of grounded nodes  $\mathbf{V}_0$ , for which  $L_0(j\omega)$  is a Laplacian that corresponds to a connected graph, a lower bound for the eigenvalues of the Hermitian part of the Dynamic Grounded Laplacian matrix  $\lambda_i(H(L_0(j\omega)))$  is given by

$$\lambda_i(H(L_0(j\omega))) \geq \frac{1}{\text{trace}((\text{Re}[L_0(j\omega)])^{-1})}, \quad (69)$$

for all  $\omega, i \in 1, 2, \dots, n$ .

**Proof:**



It is well known that the sum of the eigenvalues of any matrix is equal to its trace,

$$\sum_{i=1}^n \lambda_i(H(L_0(j\omega))^{-1}) = \text{trace}(H(L_0(j\omega))^{-1}). \quad (70)$$

Since  $H(L_0(j\omega))^{-1}$  is a real and positive definite matrix (see Lemma 5.1, and Definition 4.3), then

$$\lambda_i(H(L_0(j\omega))^{-1}) > 0, \forall \omega, i = 1, 2, \dots, n.$$

Thus,

$$\lambda_i(H(L_0(j\omega))^{-1}) \leq \text{trace}(H(L_0(j\omega))^{-1}),$$

$$\lambda_i(H(L_0(j\omega))^{-1}) \leq \text{trace}(\text{Re}[L_0(j\omega)]^{-1}). \quad (71)$$

Since the eigenvalues of  $H(L_0(j\omega))$  and  $[H(L_0(j\omega))]^{-1}$  are reciprocals of each other, then

$$\lambda_i([H(L_0(j\omega))]^{-1}) = \frac{1}{\lambda_i(H(L_0(j\omega)))}. \quad (72)$$

Combining (71) and (72), we get

$$\lambda_i(H(L_0(j\omega))) \geq \frac{1}{\text{trace}(\text{Re}[L_0(j\omega)]^{-1})}, \quad (73)$$

for all  $\omega, i \in 1, 2, \dots, n$ . ■

**Theorem A.4:** Interlacing Eigenvalue Theorem [15]: Let  $A$  be a Hermitian matrix of order  $n$ , and let  $B$  be a principal sub matrix of  $A$  of order  $n - 1$ . if  $\lambda_{min} = \lambda_n \leq \lambda_{n-1} \leq \dots \leq \lambda_2 \leq \lambda_1 = \lambda_{max}$  lists the eigenvalues of  $A$  and  $\mu_n \leq \mu_{n-1} \leq \dots \leq \mu_3 \leq \mu_2$  the eigenvalues of  $B$ . then

$$\lambda_n \leq \mu_n \leq \lambda_{n-1} \leq \mu_{n-1} \leq \dots \leq \lambda_2 \leq \mu_2 \leq \lambda_1. \quad (74)$$

**Theorem A.5:** Mirsky Theorem [13]: Let  $A \in \mathbb{C}^{n \times n}$  be given and set  $H(A) = \frac{1}{2}(A + A^H)$ . Let  $\lambda_i(A)$  and  $\lambda_i(H(A))$  denote the eigenvalues of  $A$  and  $H(A)$ , respectively, ordered so that  $\text{Re}[\lambda_1(A)] \leq \text{Re}[\lambda_2(A)] \leq \dots \leq \text{Re}[\lambda_n(A)]$  and  $\lambda_1(H(A)) \leq \lambda_2(H(A)) \leq \dots \leq \lambda_n(H(A))$ . Then

$$\sum_{i=n-k}^n \text{Re}[\lambda_i(A)] \leq \sum_{i=n-k}^n \lambda_i(H(A)), \quad (75)$$

$k = 0, 1, \dots, n - 1$ , with equality for  $k = n - 1$ .

**Lemma A.6:** Let the dynamic Laplacian  $L(j\omega) \in \mathbb{C}^{n \times n}$  be given. The Hermitian matrix associated with  $L(j\omega)$  is  $H(L(j\omega)) = \frac{1}{2}(L(j\omega) + L(j\omega)^H) = \text{Re}[L(j\omega)]$ . Let  $\lambda_i(L(j\omega))$  and  $\lambda_i(H(L(j\omega)))$  denote the eigenvalues of  $L(j\omega)$  and  $H(L(j\omega))$ , respectively, ordered so that  $0 = \text{Re}[\lambda_1(L(j\omega))] \leq \text{Re}[\lambda_2(L(j\omega))] \leq \dots \leq \text{Re}[\lambda_n(L(j\omega))]$  and  $0 = \lambda_1(H(L(j\omega))) \leq \lambda_2(H(L(j\omega))) \dots \leq \lambda_n(H(L(j\omega)))$ . Then the relationship between the real part of the smallest and largest nonzero eigenvalues of  $L(j\omega)$  and  $H(L(j\omega))$  can be given by

$$\text{Re}[\lambda_2(L(j\omega))] \geq \lambda_2(H(L(j\omega))), \quad (76)$$

and

$$\text{Re}[\lambda_n(L(j\omega))] \leq \lambda_n(H(L(j\omega))). \quad (77)$$

**Proof:**

Theorem A.5 can be applied for the matrices  $L(j\omega)$  and  $H(L(j\omega))$  by substituting  $k = n - 3$  in (75). Thus, we have

$$\sum_{i=3}^n \text{Re}[\lambda_i(L(j\omega))] \leq \sum_{i=3}^n \lambda_i(H(L(j\omega))), \quad (78)$$

with equality for  $k = n - 1$

$$\sum_{i=1}^n \text{Re}[\lambda_i(L(j\omega))] = \sum_{i=1}^n \lambda_i(H(L(j\omega))). \quad (79)$$

For a connected graph,  $\text{Re}[\lambda_1(L(j\omega))] = \lambda_1(H(L(j\omega))) = 0$ , thus we can rewrite (79) as

$$\begin{aligned} 0 + \text{Re}[\lambda_2(L(j\omega))] + \sum_{i=3}^n \text{Re}[\lambda_i(L(j\omega))] \\ = 0 + \lambda_2(H(L(j\omega))) + \sum_{i=3}^n \lambda_i(H(L(j\omega))). \end{aligned}$$

From the above quality and (78), we can conclude

$$\text{Re}[\lambda_2(L(j\omega))] \geq \lambda_2(H(L(j\omega))). \quad (80)$$

Now, substitute  $k = 0$  in (75), we get

$$\text{Re}[\lambda_n(L(j\omega))] \leq \lambda_n(H(L(j\omega))). \quad (81)$$



# Development of Breast Cancer Detection Model Using Transfer Learning of Residual Neural Network (ResNet-50)

Qasem Abu Al-Haija<sup>1</sup> and Ghandi F. Manasra<sup>2</sup>

<sup>1</sup>*Department of Electrical and Computer Engineering, Tennessee State University, Nashville, TN, USA*

<sup>2</sup>*Department of Electrical and Computer Engineering, Palestine Polytechnic University, Hebron, Palestine*

<sup>1</sup>*Qabualha@Tnstate.edu,* <sup>2</sup>*Ghandi@ppu.edu*

**Abstract**—Breast cancer disease is one of the most common and dangerous as well as being considered as the second most common world cause of cancer death in women. However, the early diagnostics and detection can provide a significant chance for correct treatment and survival. One of the most powerful tools that have shown extraordinary and superior results is the deep convolutional neural network. In this work, we propose an accurate and inclusive computational breast cancer diagnosis framework using ResNet-50 convolutional neural network to classify histopathological microscopy images. The proposed model employs transfer learning technique of the powerful ResNet-50 CNN pretrained on ImageNet to train and classify BreakHis dataset into benign or malignant. The simulation results showed that our proposed model achieves exceptional classification accuracy of 99% outperforming other compared models trained on the same dataset. Based on our novel approach, earlier detection to breast cancer as whether it is being benign or malignant can be stimulated and classified and thus save life and efforts.

**Keywords** — Breast Cancer, Histopathological Images, Medical image processing, ResNet-50, Convolutional Neural Network (CNN), Deep learning, Transfer Learning.

## I. INTRODUCTION

Cancer disease causes cells to divide uncontrollably in which abnormal cells grow and invade healthy cells in the body [1]. This can result in tumors, damage to the immune system, and other impairment that can be fatal. Indeed, the majority of cancer cases in the united states (US) is breast cancer as reported by the National Cancer Institute (NCI), in 2020 [2]. Breast Cancer Disease (BCD) is basically described as excessive or uncontrolled growth of breast tissues occurs. According to WHO (World Health Organization) [3], BCD is the second most common cause of death from cancer in women. However, BCD can be treated if it is detected at the earlier stages as reported by National Breast Cancer Foundation (NBCF). Therefore, BCD detection system is significantly on-demand to be addressed using autorecognition based artificial neural networks is needed [4].

To cope with such medical image detection and classification task, deep learning (DL) has evolved as a subset of artificial intelligence that does its inferencing using deep neural networks by employing the artificial

neural networks with several layers among the input layer and output layer. Indeed, image classification task using deep learning techniques has generated a lot of interest in recent years. ImageNet Large Scale Visual Recognition Challenge (ILSVRC) has been a genuine platform for researchers to showcase new ideas for classification. For example, in 2010, the use of GPUs, and a highly optimized implementation of 2D convolution was demonstrated as powerful enough to facilitate the training of large Convolutional Neural Networks (CNNs) [5].

CNNs have at least one Convolution layer, wherein instead of matrix multiplication, a convolution operation is performed on the input matrix in order to learn distinct low-level and high-level features of the image [6]. Deep CNNs are able to learn more features by increasing the depth of the network. However, increasing the depth of the network results in problems of vanishing gradients and degradation [7]. Thereafter, the continuous development in deep neural networks has enrich the AI field with the residual learning framework was presented in 2015 to ease the training of deep CNN networks [8]. This framework resulted in easier optimization of the network, and a higher accuracy. The network, later known as *ResNet*, was the basis of submissions to ILSVRC competition, where it won the first place on the task of ImageNet detection and ImageNet localization [9]. Residual neural networks (*ResNet*) address these challenges by introducing a “Residual block”, which features a “skip connection”, that adds the output from the previous layer to the layer. Empirical results from [9] demonstrate that the network is able to maintain stability even with far more layers than typical Convolutional Neural Networks.

In this paper, we propose to use a ResNet-50(residual CNN with 50 layers deep) to produce classifications of histopathologic images to help providing an early detecting of BCD. The BCD dataset stores different kinds of breast cancer tissues and classifies them as either benign or malignant. While the benign tissues are non-cancerous tumors, malignant tumors are cancerous, and can invade nearby tissue or spread to other parts of the body. Based on



the collected images, a ResNet-50 network is trained using transfer learning. We show that the testing accuracy of our model is superior. In particular, the core contributions of the proposed work can be listed as follows:

- We provide a comprehensive efficient classification model that can classify the breast cancer imaging of *BreakHis* dataset into benign and malignant. Besides, we present detailed preprocessing operations for the collected medical images prior to the use with deep learning algorithms.
- We employ the transfer learning technique for *ResNet50 CNN* that is pre-trained with ImageNet dataset to learn the new features for *BreakHis* dataset leveraging the power of free access *GPU* runtime provided by Google Co-laboratory.
- Extensive experimental findings are given to provide more insight into the proposed architecture and methodology. This includes simulation results related to the classification error and accuracy for training and testing as well as benchmarking of our results with existing related work.

The remainder of this paper is structured as follows: the next section, section II review some of the related research in the field, section III describes and discusses the system design modeling and architecture. Section IV provides details about experimental environment, evaluation, and discussion. Finally, Section V concludes the paper.

## II. RELATED WORK

The goal of being able to non-invasively discriminate, understand and characterize the normal and abnormal state of the breast cancer has been of great importance to both physicians and researchers due to the vital role that can be of high value in saving life to patients with breast cancer. For this, earlier detection is very essential for future treatment as it can contribute to major conditions of being able to classify cancer type as being benign or malignant breast cancer. As mentioned earlier, for women, breast cancer is a common disease worldwide. Large percentage of cancer patients are breast cancer patients. The main available methodology to have proper data that can be used to examine breast cancer is the Mammography. This can be used for experts as a way to detect signs of breast cancer if some signs are present. For this, physicians have developed standard diagnostic techniques to monitor such signs. Though the use of mammogram images has led to many clinical diagnoses that have helped to characterize underlying breast activity, the images from mammogram is subject to individual interpretation and can approximately characterize the actual state of the breast. In addition, interpretation of such images is not based on a well-determined physical state of the breast, but rather depends on heuristic pattern matching. As a result of relying on such

techniques, the mammography is based on human perception and observations so as many wrong decisions can be made. Although of the importance and that there is no doubt of the evaluation of breast cancer by the medical experts and their judgment, there are many other factors that can affect the diagnosis type. Those factors can include the presence of noise in images, the radiologist visual perception and experience, and poor image contrast or quality that makes it hard to be recognized with the eye. For this and from engineering point of view, it is essential to develop standard scientific techniques to coop with such physical situations. For this and as related to our work building an AI NN computerized system can be used to take major role in early signs of detection.

Due to the importance of detection of breast cancer as it could be very successfully treated if been detected early, several studies have been conducted in order to develop automated techniques using various medical images modules [10-13]. In order to incorporate scientific methodologies to advance procedural decision analysis, many types of images, data sets, pre-processing techniques, and types of deep learning networks along with their error metric techniques have been developed to provide quantitative and qualitative measures for the analysis of breast cancer as presented in [14-17]. Convolutional neural network is often utilized frequently to construct an effective breast cancer classification model. Such studies employed a pre-trained network or developed new deep neural networks to classify breast cancer [12, 18-20]. In related work that been done in this paper and in order to be able to differentiate between benign and malignant breast cancer, many researchers have developed many machine learning techniques to classify such breast cancer images using various pre-trained networks such as visual geometry group, *GoogleNet*, and *AlexNet* which can be applied and used on breast cancer data sets [21-24].

For this and unlike traditional machine learning methodologies that required human perception of judgments, deep machine learning has a higher probability of assurance of underlying features from studies images automatically [25]. As a result of this. many recent studies have employed deep learning approaches. For example, Spanhol et al. has employed CNN for the classification of breast cancer histopathology images [26]. Other researches like Aurojo et al. and Hand et al. have used multi classification approaches of breast cancer from histopathological images using specific deep learning models as in [27,28]. Furthermore, other research work using recognition techniques has been conducted on breast cancer using deep learning approaches, in which CNN changes and discriminates are used as certain classifiers [29].

Unlike aforementioned researches, we are leveraging the power of ResNet-50 neural network along with its associated machine learning techniques and high-performance GPUs freely offered by the google labs.

### III. SYSTEM MODELING ARCHITECTURE

In this section, we describe our proposed system model which comprises four modules including: data collection module, data preprocessing module, feature learning module and data classification module. The complete system architecture showing all components is illustrated in Fig.1. All subsystems (modules) are explained in the upcoming subsections.

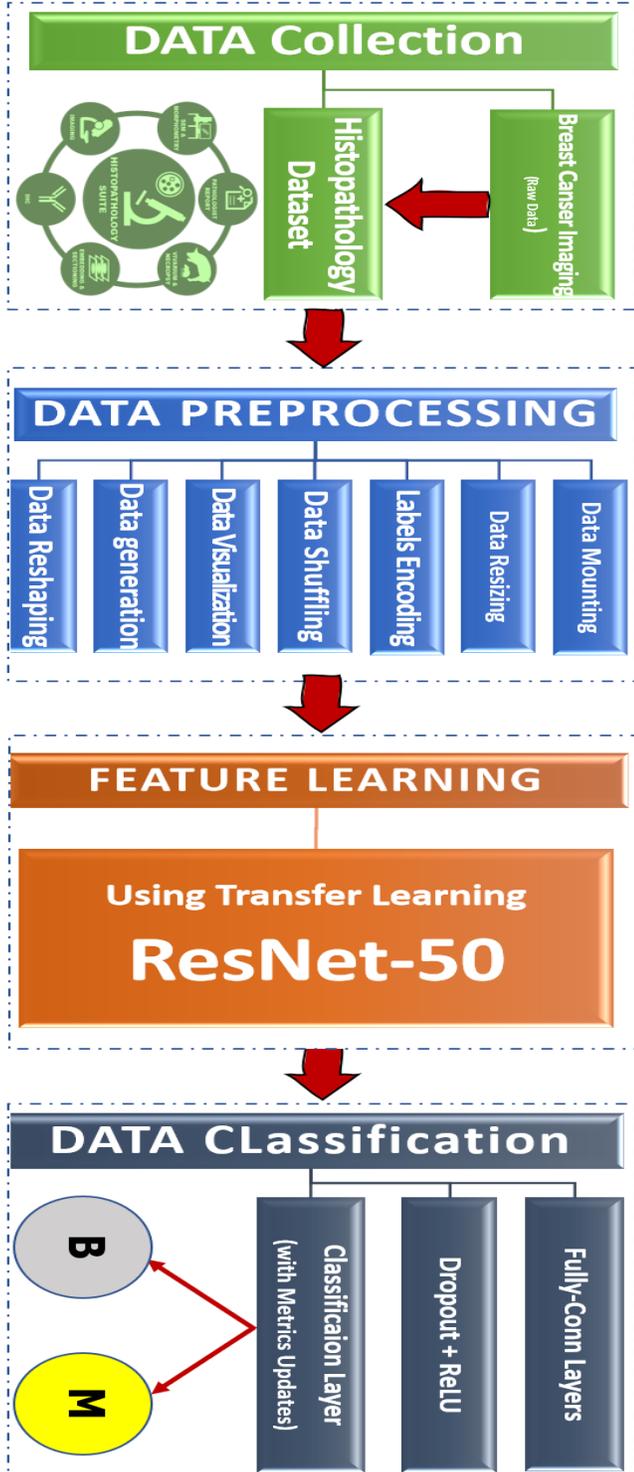


Fig. 1. Proposed System Architecture (dataflow and system modules)

### A. Data Collection Module

BCD data are available in the form of histopathological stained images [30] datasets such as WDBC breast cancer dataset [31], MITOS&ATYPIA-14breast cancer dataset [32] and BreakHis-16 breast cancer dataset [33]. In this research, we are employing the *BreakHis* histopathology images breast cancer dataset to implement a deep detection system that can be used to provide binary classification of the breast cancer image-data into *benign* (B) or *malignant* (M). Illustration samples for the benign and malignant images are provided in Fig.2. *BreakHis* composed of RGB images (700 x 460 resolution) are taken by an accurate system composed of high-resolution camera (Samsung SCC-131AN) coupled with a 3.3x microscopic unit (Olympus BX-50). Images are captured in four different magnification levels that are equivalently distributed (~25% for each level) as provided in Table 2.

TABLE I. IMAGES' DISTRIBUTION IN THE BREAKHIS DATASET

Category	Magnification level				Total
	40X	100X	200X	400X	
Benign	652	644	623	588	2480
Malignant	1370	1437	1390	1232	5429
Total	1995	2081	2013	1820	7909

Besides, there are many reasons to for this selection including:

- A comprehensive dataset with 7909 biopsy histopathology images breast cancer including both benign and malignant images acquired on 82 patients, publicly available.
- A Recent dataset associated with automated classification tasks, published by IEEE Transactions on Biomedical Engineering, 2016.
- A common histopathology dataset, in which its automated classification system is would be very valuable computer-aided diagnosis tool for clinician, if developed with high accuracy classification by employing CNN techniques.

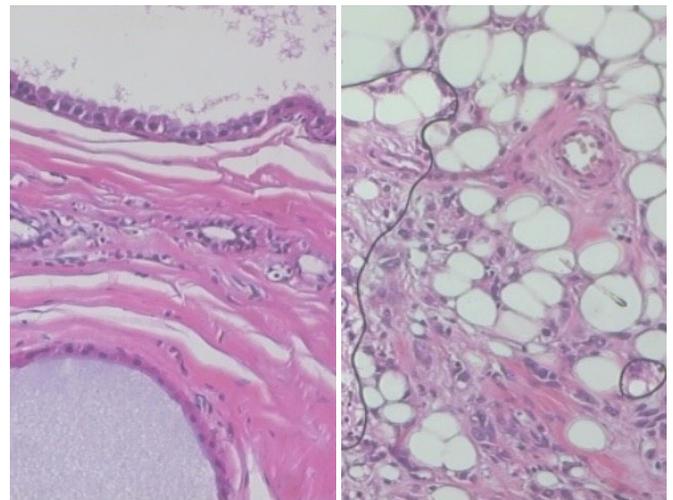


Fig. 2. Sample Images (a) Benign (b) Malignant

## B. Data Preprocessing Module

Generally, data preprocessing term belongs to all the conversions performed on the raw data prior to be processed by the deep learning module. For instance, training a convolutional neural network on raw images will probably lead to bad classification performances [34]. In this work, the collected dataset has passed through seven preprocessing operations (as shown in the second stage of Fig.2) before it is fed to the next *ResNet* module, these operations are:

**Data Mounting:** This stage is used to mount Google Drive Account (DGA) as a virtual drive, just like a USB Drive on windows OS so you can browse and access your Drive from Google Co-laboratory. Therefore, we have uploaded our *BreakHis* dataset into a folder (named as BC Dataset) on Google drive and then accessed the data into Co-Lab using *Python/gloplib* that enable the reading of dataset from external folders and *Python/pandas* library that provides data manipulation including data framing, data reading and writing between in-memory data structures and different formats[35].

**Data Resizing:** This stage is necessary to remove redundancy from the input data which only contributes to the computational complexity of the network without providing any significant improvements in the result. This is achieved with the help of *Python/keras.preprocessing* library. As a result of traying several image dimensions, we end up with image dimension of  $144 \times 96$  which minimizes the size of image dimensions by a factor of 5 while maintaining image readability with efficient computational complexity.

**Data Encoding:** This stage is used to convert categorical data (textual data) into numerical values in which our deep learning predictive models can understand. In this step, we have employed the label encoding technique to converting each value in a the category column into a number, that is it, the category 'Benign' has given the value '0' while the category 'Malignant' has given the value '1'. This is achieved with the help of *Python/sklearn.preprocessing* library.

**Data Shuffling:** This stage is used to redistribute the data samples of the training dataset to ensure that each data sample creates an "independent" change on the model, without being biased by the same points before them. This is achieved with the help of *Python/keras.preprocessing* library.

**Data Visualization:** This stage is necessary to sample and examine the input data to ensure the readability of the input images by plotting few random samples of the training dataset via 2D representation with the new image dimensions. This is achieved with the help of both *Python/tensorflow* and *Python/numpy* libraries.

**Data Generation:** This stage is used to generate batches of tensor image data with real-time data augmentation. The data will be looped over (in batches) for both training and testing.

Also, batch normalization is performed at this stage with image plotting for sample normalized images along with encoded labels. This is achieved with the help of both *Python/tensorflow*, *Python/keras.preprocessing* and *Python/matplotlib*.

**Data Reshaping:** This stage is used to customize the input layer of ResNet-50 to accommodate the input shape for our preprocessed dataset ( $Img_{width} = 144, Img_{Height} = 96, No_{Channels} = 3$ ). This is achieved with the help of *Python/keras.preprocessing* library.

## C. Feature Learning Module

In machine learning, feature learning (FL) is a set of techniques that allows a system to automatically discover the representations needed for feature detection, prediction, or classification from the preprocessed dataset [36]. This allows a machine to learn the features and use them to perform a specific task such as classification or prediction. In deep learning, the feature learning can be accomplished by developing a complete convolutional neural network (CNN) to train and test the set of images or by customizing a pretrained CNN in the classification/prediction for the new images-set. The later technique is called Transfer learning. The idea of both techniques is illustrated in Fig.3. According to the figure, with transfer learning, you use the convolutional base (green module in the figure) and only re-train the classifier to your dataset (pink module).

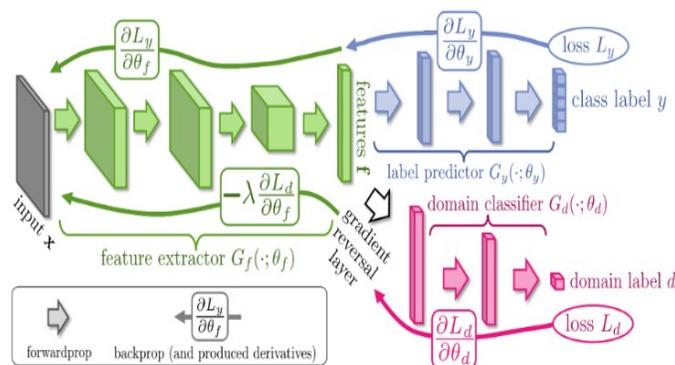


Fig. 3. Illustration of no-transfer learning vs transfer learning CNN [37].

*Transfer learning (TL)* is usually employed in the applications of DL which enable you to utilize a pretrained network to perform new prediction/classification tasks. This, indeed, require fine-tuning the learning parameters of the utilized pretrained network with randomly initialized weights to accommodate the new learning tasks. TL usually provides much faster and easier learning/training than training the network from scratch. As reported in [38], transfer learning is an optimization, a shortcut to saving time or getting better performance. This is illustrated in Fig.4 that analyze the training performance of CNN employing transfer learning vs CNN with no transfer learning.

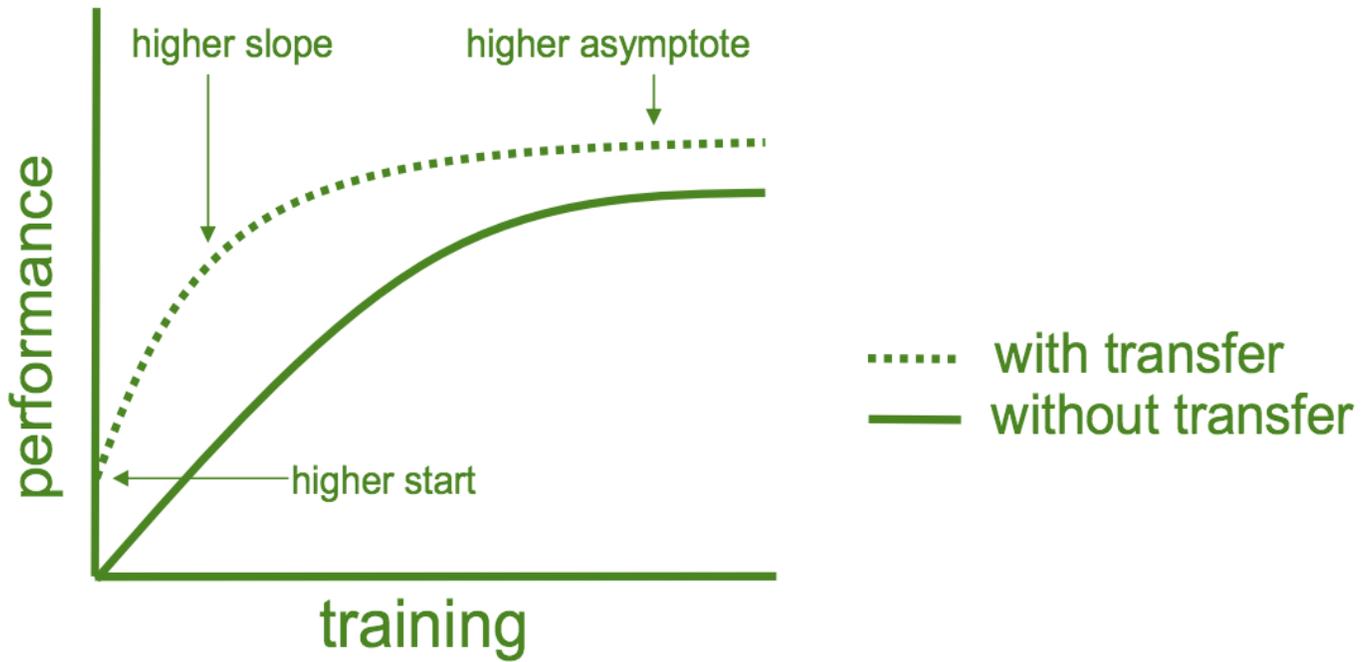


Fig. 4. Training performance of no-transfer learning CNN vs transfer learning CNN [38].

In this module, we are utilizing the transfer learning technique to retrain the powerful ResNet-50 CNN with Fine-tuning for the network parameters and hyperparameters. This is accomplished by creating a model of ResNet-50 with pretrained parameters (weights) from

ImageNet dataset [39] after the preprocessing of collected dataset (histopathologic images). This is achieved with the help of both Python libraries including: *Python/keras.applications.resnet50* and *Python/keras.models*. Illustration of this module can be depicted from Fig.5.

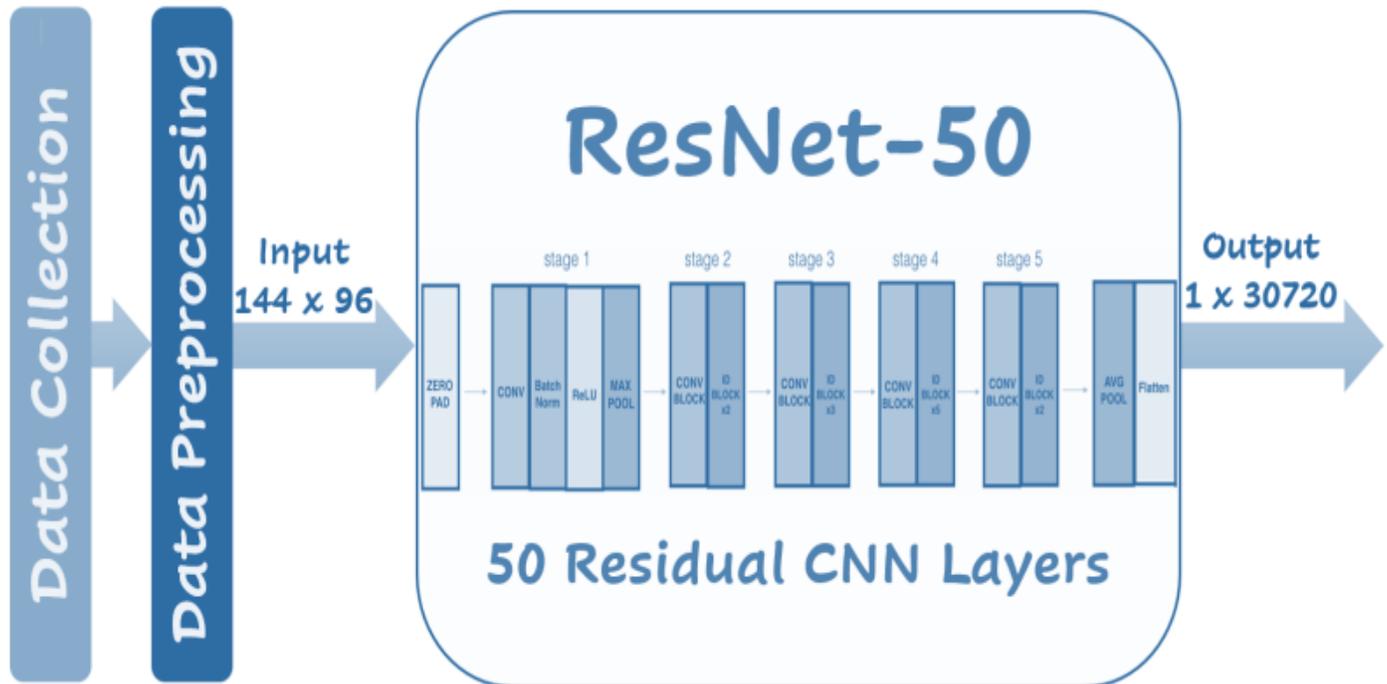


Fig. 5. Feature Learning Module Using ResNet-50 Transfer Learning for the proposed breast cancer detection system

### D. Data Classification Module

Data classification is an essential feature to separate large datasets into classes for the purpose of decision making, pattern recognition and others [40]. A classification layer makes use of the fully connected layer and computes the cross-

entropy loss for the multi-class classification problems with mutually exclusive classes. This is achieved using *Paython/keras.layers*, *Paython/keras.models* and *Paython/keras/optimizers*. Illustration of this module is provided in Fig.6.

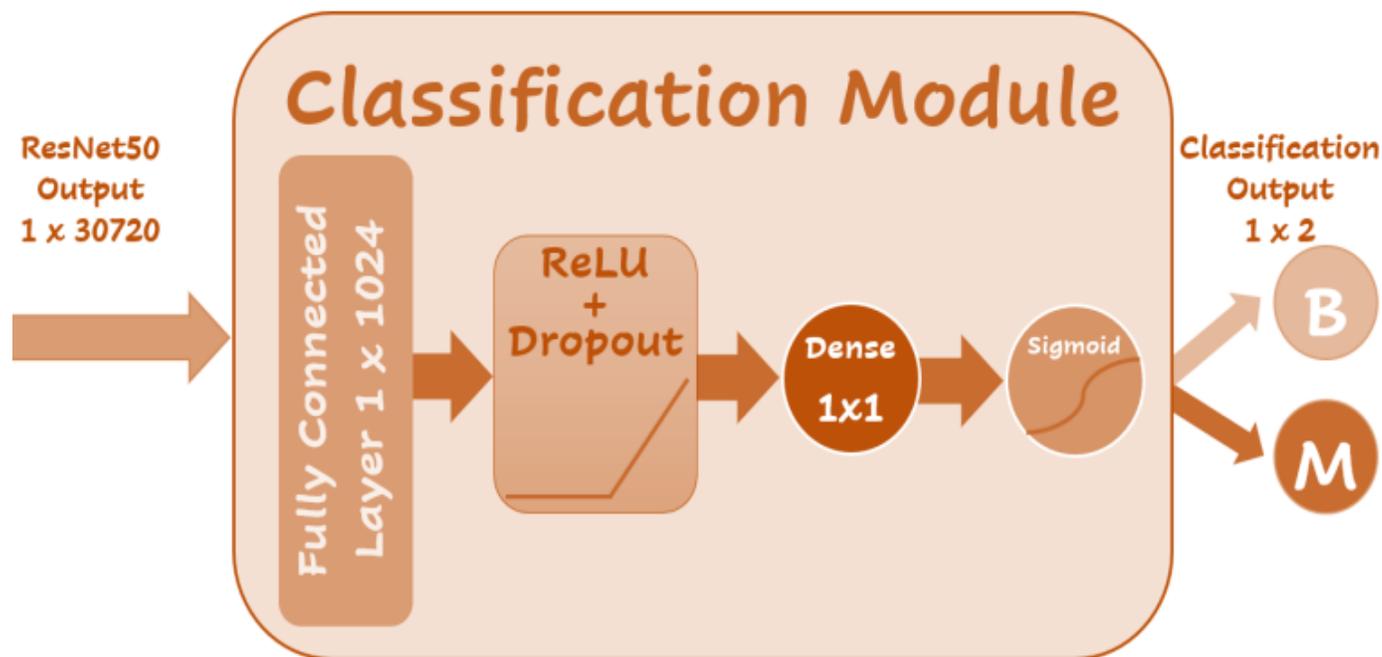


Fig. 6. Data Classification Module of proposed breast cancer detection system

According to the figure, the module receives the features from ResNet-50 and pass through a fully connected (FC) layer composed of 1024 neurons configured a 40% of dropout to prevent over-fitting [41]. After that, the units have been activated with rectification function namely known as *ReLU*. *ReLU* function is  $MAX(X, 0)$  that sets all negative values in the matrix  $X$  to zero while all other values are kept constant. The reason of using *ReLU* is that training a deep network with *ReLU* tended to converge much more quickly and reliably than training a deep network with sigmoid activation [42]. Finally, to provide the probabilities for the classes, the output layer was composed of one neuron unit configured with Sigmoid function (Binary classifier). *Sigmoid* is mathematical function that takes as input a vector of  $K$  real numbers and normalizes it into a probability distribution consisting of two probabilities (e.g. Benign vs Malignant) [43]. *Sigmoid* function is defined as follows:

$$S(x)_i = \frac{1}{1 + e^{-x}} = \frac{e^x}{e^x + 1}, \quad S: \mathbb{R}^k \mapsto \mathbb{R}^k$$

for  $i = 1, 2, \dots, K$  and  $x = (x_1, x_1, \dots, x_K) \in \mathbb{R}^k$

Besides, to calculate the loss for training and testing we have employed *Mean Squared Error (MSE) loss*. MSE Loss [44] is calculated as the average of the squared differences between the predicted and actual values. The result is always positive regardless of the sign of the predicted and actual values and a perfect value is 0.05 MSE function is defined as follows:

$$MSE = \frac{1}{n} \sum_{i=1}^n (Y_i - \hat{Y}_i)^2$$

Where:  $n$  is the number of samples,  $Y_i$  is the original data sample, and  $\hat{Y}_i$  is the predicted data sample

Finally, to finalize the compilation of *Kerasmodel*, we have utilized Root Mean Square Propagation optimizer (RMSprop) [45]. *RMSprop* utilizes the magnitude of recent gradients to normalize the gradients. In *RMSprop*, we divide the learning rate for a weight by a running average of the magnitudes of recent gradients for that weight. That is, it, keep a moving average of the squared gradient for each weight. The update is done separately for each parameter as follows:



For each Parameter  $w^j$

( $j$  subscript dropped for clarity)

$$\nu_t = \rho\nu_{t-1} + (1 - \rho) * g_t^2$$

$$\Delta\omega_t = -\frac{\eta}{\sqrt{\nu_t + \epsilon}} * g_t$$

$$\omega_{t+1} = \omega_t + \Delta\omega_t$$

$\eta$  : Initial Learning rate

$\nu_t$  : Exponential Average of squares of gradients

$g_t$  : Gradient at time  $t$  along  $w^j$

#### IV. EXPERIMENTAL ENVIRONMENT AND EVALUATION

To accomplish this proposed classification task, we have utilized different development tools and packages including Python 3.7 language along with TensorFlow 2.0 package and other aforementioned libraries, Google Colaboratory development environment leveraging the power of free access GPU runtime and Google Drive for storing and accessing

the dataset. Besides, the experimental setup for training/testing model has been configured as follows: 75% of the dataset used for training (i.e., ~6000 images, here we used all images with 100X, 200X, 400X magnifications from both classes for training), 25% of the dataset used for testing (i.e., ~2000 images, here we used all images with 40X magnification from both classes for testing), the number of epochs=200, the number of steps per epoch (iterations)=200, the number of verbose=1(i.e., one progress bar line per epoch), and the `batch_size = 32`. This configurations have been achieved using Python/model.fit & Python/matplotlib libraries.

Also, the plot for loss function comparing the behavior of training loss and testing loss obtained during the training process is presented in Fig.7. It can be clearly seen, both losses are systematically decreasing while training proceeds with faster threshold obtained for the training loss after only 25 epochs. However, the testing loss has saturated after almost 125 epochs of training process with less than 0.05 of MSE. This difference in saturation levels and threshold of training loss and testing loss is permitted to avoid underfitting or overfitting.

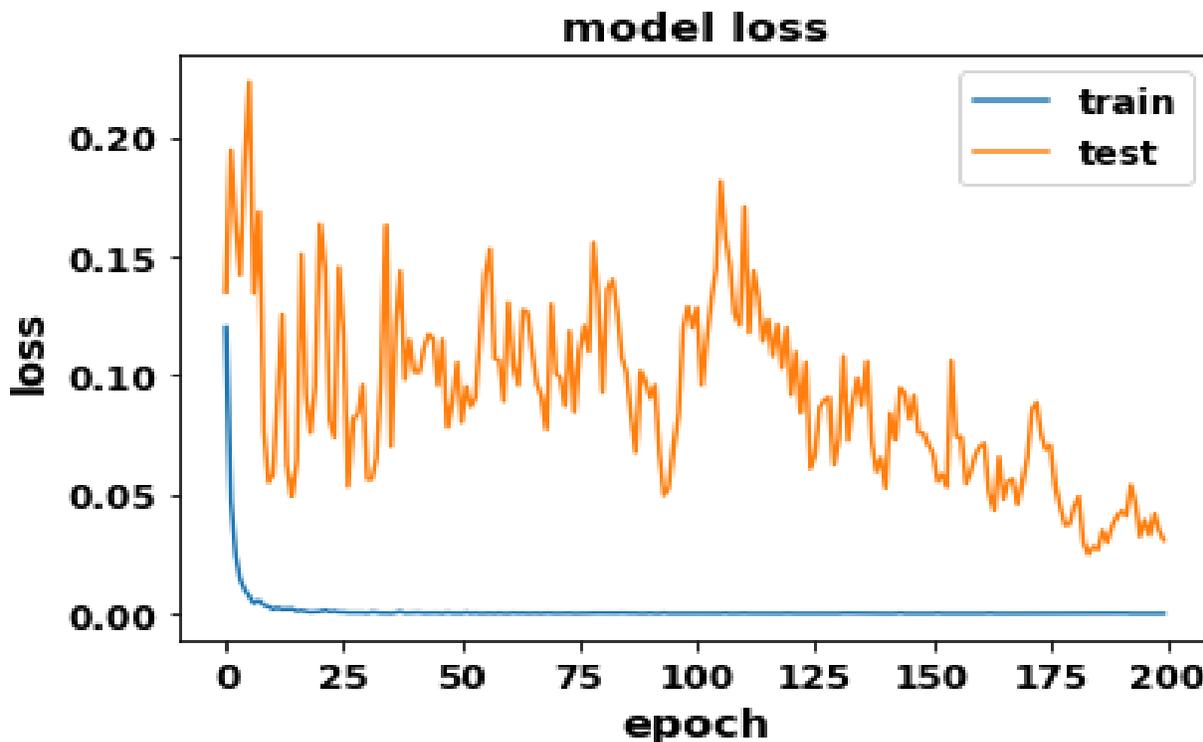


Fig. 7. Training/Testing Losses vs. number of epochs for the proposed detection system

Moreover, the plot for accuracy metric comparing the performance of training accuracy and testing accuracy obtained during the training process is given in Fig. 8. According to the figure, both accuracy curves are steadily increasing while training proceeds with faster ceiling level

obtained for training accuracy after which recorded almost 100% only after 25 epochs. While the testing accuracy level was fluctuating between 98% and 99.8% after 100 epochs recoding an average testing accuracy of 99% of overall testing accuracy.

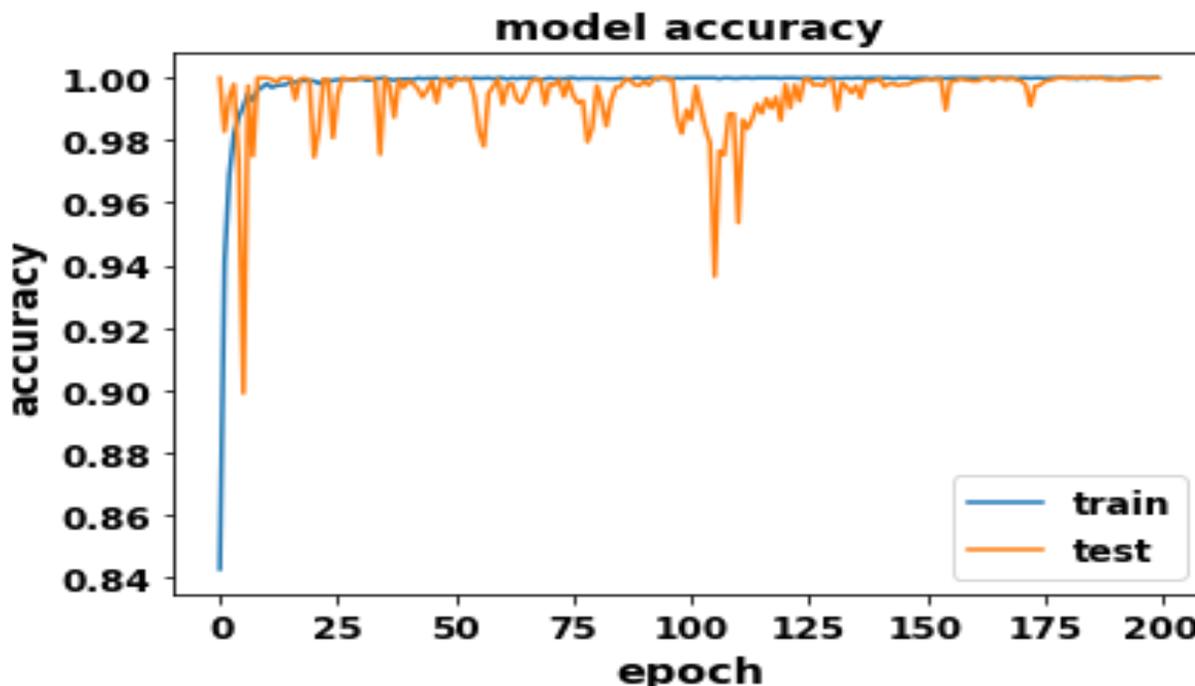


Fig. 8. Training/Testing Accuracies vs. number of epochs

Finally, even though the exiting state-of-art researches for classifying the histopathological images dataset use different network configurations, learning policies, programming techniques, and computing platforms, however, we still can compare the classification system performance in terms of training and testing accuracy metrics. Therefore, for better readability, we summarize the time accuracy metrics for related state-of-art research’s in the following table, Table I, in chronological order. According to the comparison of the table, it can be seen that our proposed model has recorded an attractive result in terms of both training and testing accuracy showing superiority over all other compared methods.

V. CONCLUSIONS

An efficient model for classifying the stained histological breast cancer images with high level of classification accuracy has been obtained and verified. To increase the robustness of the classifier, we employed the transfer learning of the powerful ResNet-50 CNN pretrained on ImageNet. The developed model makes use of *BreakHis* dataset with 75% of the images used for training and 25% used for testing. Indeed, the proposed work provides a comprehensive model for medical image processing and classification from input layer to the output layer. Eventually, to our knowledge, the reported results are superior to the automated analysis of breast cancer images reported in literature [26-35]. Table II summarizes such an achievement and the superiority of our proposed model that is been adopted with 99.1% accuracy as compared to other models and work in the same field.

TABLE. I. COMPARING ACCURACY METRIC WITH EXISTING MODELS

Research	Method	Accuracy
Gour et. al., 2020 [46]	Customized ResHist [152- Residual Learning-CNN]	84.34%
Gupta et. al., 2020 [47]	Hybrid CNN [Employed Several Pre-Trained CNNs]	93.27%
Dabeer et. al., 2019 [48]	Customized LeNet-5 CNN	93.45%
Sagar et. al., 2019 [49]	Pre-Trained DenseNet201CNN	98.30%
Kassani et. al., 2019 [50]	Pre-Trained Combined CNNs [DenseNet201+ VGG19 + MobileNetV2]	98.13%
Gandomkar et al., 2018 [51]	Pre-Trained ResNet-152 CNN	98.77%
Adeshina et. al., 2018, [52]	New Deep CNN/14 Layers (DCNN-14)	91.5%
Han et al., 2017 [53]	New Class Structure-Based Deep CNN (CSDCNN)	93.20%
Sun, et. al, 2017 [54]	Pre-Trained GoogLeNet CNN	95.00%
Spanhol et al. 2016 [55]	Pre-Trained AlexNet CNN	84.60%
Proposed Method [56]	Pre-Trained ResNet-50 CNN	<b>99.10 %</b>

## REFERENCES

- [1] R. Nall, Y. Ranchod, What to know about cancer, Medically reviewed by Yamini Ranchod, Medical News Today (MNT), Jan, 6, 2020.
- [2] NCI, Breast Cancer—Patient Version, National Cancer Institute (NCI), 2020.
- [3] WHO. (2018). Data of breast cancer sufferers in the world. World Health Organization (WHO), Jan. 3. 2019.
- [4] Q. Abu Al-Haija and N. A. Jebri, "A Systemic Study of Pattern Recognition System Using Feedback Neural Networks", WSEAS Transactions on Computers, Vol.19, 2020, Art. #16, pp. 115-121
- [5] Q. Abu Al-Haija, C.D. McCurry, S. Zein-Sabatto, "Intelligent Self-Reliant Cyber-Attacks Detection and Classification System for IoT Communication Using Deep Convolutional Neural Network", 12<sup>th</sup> International Network Conference 2020 (INC2020), Springer Lecture Notes in Networks and Systems, Sep. 2020.
- [6] S. Saha. A Comprehensive Guide to Convolutional Neural Networks-ELI5 way. Towards Data Science, A Medium publication sharing concepts, ideas, and codes, 2018.
- [7] S. Jia. Vanishing Gradient vs Degradation. Towards Data Science, A Medium publication sharing concepts, ideas, and codes, Sep, 2018.
- [8] C. Shorten. Introduction to ResNets. Towards Data Science, A Medium publication sharing concepts, ideas, and codes, Jan, 2019.
- [9] K. He, X. Zhang, S. Ren, J. Sun. Deep Residual Learning for Image Recognition. Microsoft Research. 2015.
- [10] Bejnordi BE, Lin J, Glass B, Mullooly M, Gierach GL, Sherman ME et al (2017a) Deep learning-based assessment of tumor-associated stroma for diagnosing breast cancer in histopathology images. Paper presented at the 2017 IEEE 14th international symposium on biomedical imaging (ISBI 2017).
- [11] Bekker AJ, Greenspan H, Goldberger J (2016) A multi-view deep learning architecture for classification of breast microcalcifications. Paper presented at the Proceedings—international symposium on biomedical imaging.
- [12] Byra M, Piotrkowska-Wroblewska H, Dobruch-Sobczak K, Nowicki A (2017) Combining Nakagami imaging and convolutional neural network for breast lesion classification. Paper presented at the IEEE international ultrasonics symposium, IUS
- [13] Cao J, Qin Z, Jing J, Chen J, Wan T (2016) An automatic breast cancer grading method in histopathological images based on pixel-, object-, and semantic-level features. Paper presented at the 2016 IEEE 13th international symposium on biomedical imaging (ISBI).
- [14] Carneiro G, Nascimento J, Bradley AP (2017) Automated analysis of unregistered multi-view mammograms with deep learning. *IEEE Trans Med Imaging* 36(11):2355–2365.
- [15] Cheng J-Z, Ni D, Chou Y-H, Qin J, Tiu C-M, Chang Y-C et al (2016) Computer-aided diagnosis with deep learning architecture: applications to breast lesions in US images and pulmonary nodules in CT scans. *Sci Rep* 6:24454.
- [16] Dhungel N, Carneiro G, Bradley AP (2017) A deep learning approach for the analysis of masses in mammograms with minimal user intervention. *Med Image Anal* 37:114–128.
- [17] Feng Y, Zhang L, Yi Z (2018) Breast cancer cell nuclei classification in histopathology images using deep neural networks. *Int J Comput Assist RadiolSurg* 13(2):179–191.
- [18] Kim DH, Kim ST, Ro YM (2016) Latent feature representation with 3-D multi-view deep convolutional neural network for bilateral analysis in digital breast tomosynthesis. Paper presented at the 2016 IEEE international conference on acoustics, speech and signal processing (ICASSP).
- [19] Abdullah-Al N, Bin Ali F, Kong YN, IEEE (2017) Histopathological breast-image classification with image enhancement by convolutional neural network. Paper presented at the 2017 20th International conference of computer and information technology, New York.
- [20] Araujo T, Aresta G, Castro E, Rouco J, Aguiar P, Eloy C et al (2017) Classification of breast cancer histology images using convolutional neural networks. *PLoS ONE* 12(6):14. <https://doi.org/10.1371/journal.pone.0177544>.
- [21] Alkhaleefah, M.; Ma, S.-C.; Chang, Y.-L.; Huang, B.; Chittem, P.K.; Achhannagari, V.P. Double-Shot Transfer Learning for Breast Cancer Classification from X-Ray Images. *Appl. Sci.* 2020, 10, 3999.
- [22] T. Araújo, G. Aresta, E. Castro et al., "Classification of breast cancer histology images using convolutional neural networks," *PLoS One*, vol. 12, no. 6, Article ID e0177544, 2017.
- [23] A. Cruz-Roa, H. Gilmore, A. Basavanahally et al., "Accurate and reproducible invasive breast cancer detection in whole-slide images: a Deep Learning approach for quantifying tumor extent," *Scientific Reports*, vol. 7, no. 1, Article ID 46450, 2017.
- [24] N. Weiss, H. Kost, and A. Homeyer, "Towards interactive breast tumor classification using transfer learning," in *Proceedings of the International Conference Image Analysis and Recognition*, pp. 727–736, Springer, Póvoa de Varzim, Portugal, June 2018.
- [25] Bengio, Y.; Courville, A.; Vincent, P. Representation Learning: A Review and New Perspectives. *IEEE Trans. Pattern Anal. Mach. Intell.* 2013, 35, 1798–1828.
- [26] Spanhol, F.A.; Oliveira, L.S.; Petitjean, C.; Heutte, L. Breast cancer histopathological image classification using Convolutional Neural Networks. In *Proceedings of the 2016 International Joint Conference on Neural Networks (IJCNN)*, Vancouver, BC, Canada, 24–29 July 2016.
- [27] Araújo, Teresa, Guilherme Aresta, Eduardo Castro, José Rouco, Paulo Aguiar, Catarina Eloy, António Polónia, and Aurélio Campilho. "Classification of breast cancer histology images using Convolutional Neural Networks." *PloS one* 12, no. 6 (2017): e0177544.
- [28] Han, Zhongyi, Benzhenq Wei, Yuanjie Zheng, Yilong Yin, Kejian Li, and Shuo Li. "Breast cancer multi-classification from histopathological images with structured deep learning model." *Scientific reports* 7, no. 1 (2017): 4172.
- [29] Spanhol, Fabio A., Luiz S. Oliveira, Paulo R. Cavalin, Caroline Petitjean, and Laurent Heutte. "Deep features for breast cancer histopathological image classification." In *Systems, Man, and Cybernetics (SMC), 2017 IEEE International Conference on*, pp. 1868-1873. IEEE, 2017.



- [30] J. D. Toro, et. al., Analysis of Histopathology Images. Chapter 10, Biomedical Texture Analysis Fundamentals, Tools and Challenges, Elsevier, 2017, Pages 281-314, 10.1016/B978-0-12-812133-7.00010-7.
- [31] W. H. Wolberg, W. N. Street, O. L. Mangasarian, Breast cancer Wisconsin (diagnostic) data set, UCI Machine Learning Repository
- [32] Lowe A KMe a, Grunkin M. Mitos atypia grand challenge 2014. 2014. [https://mitos-atypia-14.grand\\_challenge.org/Dataset/](https://mitos-atypia-14.grand_challenge.org/Dataset/)
- [33] Spanhol, F., Oliveira, L. S., Petitjean, C., Heutte, L. A Dataset for Breast Cancer Histopathological Image Classification, IEEE Transactions on Biomedical Engineering (TBME), 63(7):1455-1462, 2016. DOI: 10.1109/TBME.2015.2496264
- [34] K. K. Pal and K. S. Sudeep. Preprocessing for image classification by convolutional neural networks. IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT), Bangalore, 2016, pp. 1778-1781.
- [35] W. McKinney. pandas: powerful Python data analysis toolkit Release 1.0.3. By Pandas Development Team.
- [36] Y. Bengio; A. Courville; P. Vincent. Representation Learning: A Review and New Perspectives. IEEE Transactions on Pattern Analysis and Machine Intelligence. 35 (8): 1798–1828, 2013. arXiv:1206.5538. doi:10.1109/tpami.2013.50. PMID 23787338.
- [37] Ganin, Y., & Lempitsky, V. Unsupervised Domain Adaptation by Backpropagation. ACM 32nd International Conference on Machine Learning, vol. 37, 2015.
- [38] R.F. Lopes. Wild Data Part 3: Transfer Learning. Stratio Big Data Inc, 2018.
- [39] K. He, X. Zhang, S. Ren and J. Sun . Deep Residual Learning for Image Recognition. arXiv:1512.03385v1, 2015.
- [40] O. Knocklein. Classification Using Neural Networks. Medium: Toward Data Science. Jun. 2019.
- [41] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. J. Mach. Learn. Res., 15(1):1929–1958, Jan. 2014.
- [42] J. Brownlee. A Gentle Introduction to the Rectified Linear Unit (ReLU). Machine Learning Mastery Inc., Deep Learning Performance, Jan, 2019.
- [43] Fei-Fei Li, “CS231n: Convolutional Neural Networks for Visual Recognition”, Computer Science, Stanford University, 2019.
- [44] M. Binieli .Machine learning: an introduction to mean squared error and regression lines. #MATHEMATICS, Free Code Camp, 16 OCT 2018
- [45] A. C. Wilson, R. Roelofs, M. Stern, N. Srebro, B. Recht . The Marginal Value of Adaptive Gradient Methods in Machine Learning. Advances in Neural Information Processing Systems 30 (NIPS 2017), Pp. 4148—4158, 2017.
- [46] M. Gour, S. Jain, T. S. Kumar, Residual learning based CNN for breast cancer histopathological image classification. International journal of Imaging Systems Technology, 30: 621– 635. (2020).
- [47] K. Gupta, N. Chawla, Analysis of Histopathological Images for Prediction of Breast Cancer Using Traditional Classifiers with Pre-Trained CNN, Procedia Computer Science, Volume 167, 2020, Pages 878-889, ISSN 1877-0509, <https://doi.org/10.1016/j.procs.2020.03.427>.
- [48] S. Dabeer, M. M. Khan, S. Islam. Cancer diagnosis in histopathological image: CNN based approach. Informatics in Medicine Unlocked, Elsevier, Vol.16, 2019, 100231.
- [49] A. Sagar. Convolutional Neural Network for Breast Cancer Classification: Deep Learning for solving the most commonly diagnosed cancer in women. Towards Data Science, A Medium publication sharing concepts, ideas, and codes, 2019.
- [50] S. H. Kassani, P. H. Kassani, M. J. Wesolowski, K. A. Schneider, and R.,Deters. Classification of histopathological biopsy images using ensemble of deep learning networks. ACM 29th Annual International Conference on Computer Science and Software Engineering (CASCON '19). IBM Corp., USA, 92–99, 2019.
- [51] Z. Gandomkar, P. C. Brennan, and C. Mello-Thoms. MuDeRN: Multi-category classification of breast histopathological image using deep residual networks. Artificial Intelligence in Medicine 88, pp.14 – 24, 2018.
- [52] Adeshina SA, Adedigba AP, Adeniyi AA, Aibinu AM. Breast cancer histopathology image classification with deep convolutional neural networks. In: 2018 14th international conference on electronics computer and computation (ICECCO). IEEE; 2018. p. 206–12.
- [53] Z. Han, B. Wei, Y. Zheng, Y. Yin, K. Li, and S. Li. Breast cancer multi-classification from histopathological images with structured deep learning model. Scientific reports 7, 1 (4172), 2017.
- [54] J. Sun and A. Binder, "Comparison of deep learning architectures for H&E histopathology images," 2017 IEEE Conference on Big Data and Analytics (ICBDA), Kuching, 2017, pp. 43-48.
- [55] F. A. Spanhol, L. S. Oliveira, C. Petitjean, and L. Heutte. Breast cancer histopathological image classification using convolutional neural networks. International joint conference on neural networks (IJCNN). IEEE, 2560—2567, 2016.
- [56] F. A. Spanhol, L. S. Oliveira, C. Petitjean, and L. Heutte. Breast cancer histopathological image classification using convolutional neural networks. International joint conference on neural networks (IJCNN). IEEE, 2560—2567, 2016.
- [57] Q. Abu Al-Haija, A. Adebajo, “Breast Cancer Diagnosis in Histopathological Images Using ResNet-50 Convolutional Neural Network”, To Appear 2020 IEEE International IOT, Electronics and Mechatronics Conference (IEMTRONICS), Canada, 9th -12th Sep, 2020.



# Design and Implementation of A Wearable Monitoring System for Alzheimer's Patient and Elders

Mohammed Asaf-Ud-Doulah  
*Department of Electrical &  
 Electronic Engineering  
 American International  
 University -Bangladesh  
 Dhaka, Bangladesh  
 asafuddoulah@ieee.org*

Md. Tauhidur Rahman  
*Department of Electrical &  
 Electronic Engineering  
 American International  
 University -Bangladesh  
 Dhaka, Bangladesh  
 tauhidur.rahman@ieee.org*

Naymul Bari  
*Department of Electrical &  
 Electronic Engineering  
 American International  
 University -Bangladesh  
 Dhaka, Bangladesh  
 naymulbari@gmail.com*

Jobayer Ibne Azad  
*Department of Electrical &  
 Electronic Engineering  
 American International  
 University -Bangladesh  
 Dhaka, Bangladesh  
 jobayer.ibne.azad@ieee.org*

Shantanu Kumar Nath  
*Department of Electrical &  
 Electronic Engineering  
 American International  
 University -Bangladesh  
 Dhaka, Bangladesh  
 sknath@aiub.edu*

Sumaiya Umme Hani  
*Department of Electrical &  
 Electronic Engineering  
 American International  
 University -Bangladesh  
 Dhaka, Bangladesh  
 sumaiya@aiub.edu*

**Abstract**—Advancing digital networking and improving technology offers an opportunity for the health sector to play a crucial role. The technology for interactive and remote patient care is now much more available and affordable. A patient with a severe disease like Alzheimer's should be checked for wellbeing. In terms of a critical patient, the major problem is to go to the hospital for the routine checkup. In regards to vital care, the main challenge is regular screening in the hospital. Based on the capabilities of IoT based technology, a frequent medically impaired patient may resolve their difficulties in consulting a doctor in a regular basis. A wearable device that allows people to get their real-time pulse rate, temperature, ECG and orientation is what this project suggested. This project led to the creation of a prototype health monitoring device. Four health sensors are part of this prototype: MAX30100, LM35, MPU6050, AD8232. All these sensors were integrated into one system using Arduino Nano. In real-time, cloud storage is continuously updated. An android application was created, where the database was accessed and the health parameters were graphically represented. This wearable health monitoring device will resolve the need for hospital admission or routine checkup in case of minor health issues. In order to clarify the functionality of the sensors used, a thorough study of the signal to respond to variability in physical and environmental behaviours was carried out.

**Keywords**—Alzheimer, Arduino Nano, Health Monitoring Device, Android application

## I. INTRODUCTION

The Wearable Health Monitoring System is a focal point for both academics and industry due to the fast-growing ageing population and resulting health and difficulties. Government priorities include such a need to minimize or reduce healthcare costs and also to improve service quality. While technology plays a major role in achieving these goals, any approach must use sufficient domain knowledge, Planning, application and

validation. Therefore, more real-time health monitoring can be used to detect relapses under circumstances that require early intervention to address these challenges. Therefore, the creation of a Wearable Health Monitoring System capable of remotely monitoring elderly patients was achieved. [1]

In the field of healthcare, cloud-based technology enabling better collaboration has grown. When the wearable equipment is paired with cloud-based technology, it is much more efficient, and it will significantly attract more customers.

Wearing tools include a range of benefits, including economic, organizational and practical benefits contrasted with housekeeping. The following list of the advantages of using Wearable Health Monitoring System in health care:

- **Monitors Vulnerable Patients:** An exclusive use of wearable technology is that it can be used from a distance to track vulnerable patients. While many people use wearable devices for their health, And people continue to use them to protect the health of their family's elderly members.
- **Encourages Proactive Healthcare:** Many citizens are responsive when dealing with any potential health risks. When they begin to feel ill, uncomfortable or anything else, they go to see a doctor. They have a health problem and respond by testing it out. The prospect of a more proactive approach to wellness is offered by wearable technology. Wearable devices can be used to take action at an early stage rather than to respond to health issues when they start causing problems.
- **Communication with caregivers:** In case the patient faces any difficulties, the wearable Device will let the caregiver know that his/her patient is facing problems.

- Communication with physicians: In the situation when patients experience trouble or discomfort, the wearable Device can allow the doctor to track the vital parameters of the body, and can, therefore, enable automated monitoring.

#### A. Principle of photoplethysmography (PPG)

Photoplethysmography (PPG) is a simple optical technique used to detect volumetric changes in blood in the peripheral circulation. The surface calculation process is cost-efficient and non-invasive. The system provides valuable information about our cardiovascular system. This method, which is commonly used for clinical physiological measurements and monitoring, has revived recent technological developments. [2]

PPG makes uses of low-intensity infrared (IR) light. As passing through biological tissues, the muscles, skin pigments absorb light and heat, both the veins and arteries. Because light is absorbed more intensely by blood than surrounding tissues, PPG sensors can identify changes in blood flow as differences in light intensity. The PPG voltage signal is proportional to the amount of blood passing through the vessels. This approach may even detect minor changes in blood volume, but blood quantification cannot be achieved. A PPG signal has several components, including arterial blood volumetric changes associated with a heart attack.

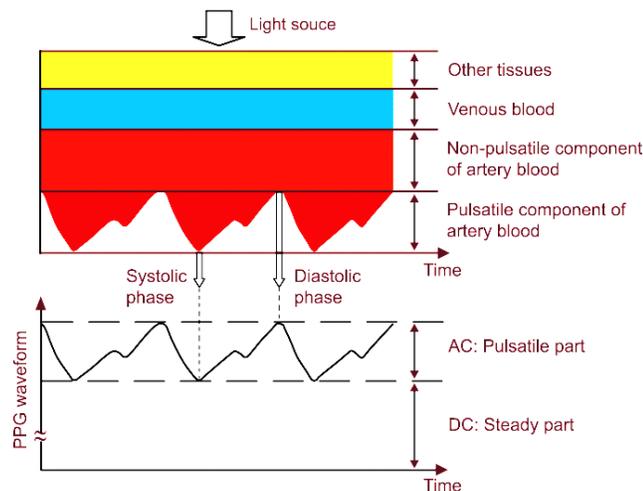


Fig. 1. Variation of light attenuation by tissue [3].

With the help of a bar or a graph, PPG displays the changes in blood supply as a waveform. The waveform has a portion of alternating current (AC) and a component of direct current (DC). In a rhythm of the pulse, the AC elements reflect changes in the blood volume. The DC portion is extracted from the absorbed or transmitted optical signals of the tissue and is calculated by both the tissue composition and the blood volume of the vein and arteries. The DC component shows minor changes with respiration. [2] The AC component's specific frequency relies on the heart rate and is superposed on the DC standard.

#### B. Principle of Galvanic Skin Response (GSR)

The Galvanic Skin Response (GSR), also named Electrodermal Activity (EDA) and Skin Conductance (SC), is the measure of the continuous variations in the electrical characteristics of the skin, i.e., for instance, the conductance, caused by the variation of the human body sweating. [8] The standard hypothesis of GSR analysis is that the resistance of skin differs from the condition of the sweat glands in the body. The Autonomous Nervous System (ANS) regulates the sweating of human bodies. If the autonomous nervous system's sympathetic branch (SNS) becomes extremely aroused, sweat gland activation often increases, growing skin actions and vice versa. Skin activity can thus test the reactions of the human Sympathetic Nervous System. Such a device is directly associated with human emotional actions.

Further research has demonstrated the correlation between GSR and certain mental conditions, such as distress, sleepiness, and engagement. [4]

The GSR signal is easy to capture: usually, it only requires two electrodes on one hand's second and third fingers. The difference of the low-voltage applied current is used as the EDA indicator between the two electrodes. New commercial medical devices like bracelets, watches have lately been developed, and are gradually portable and advanced. This test is, therefore, also can be done in non-laboratory conditions.

## II. HARDWARE AND SOFTWARE ARCHITECTURE

#### A. Hardware Architecture

The hardware architecture of this project is straightforward with Arduino Nano as a microcontroller along with the other sensors. The block diagram for the execution of the project is seen in the following diagram.

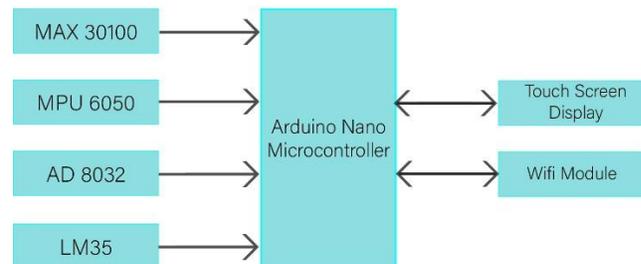


Fig. 2. Block diagram of the system architecture.

To execute the project without a functional error, each part was previously measured individually from each other. Eventually, the individual sensors were merged to work together to accomplish the main objective of the project. Sensor data is transferred through wireless communication to the Arduino Nano. Physiological information about blood pressure pulse rates and the oxygen flow can be given by the MAX30100. In addition to the MPU, data from sensors are obtained and important signals including heartbeat, temperature and more are measured.

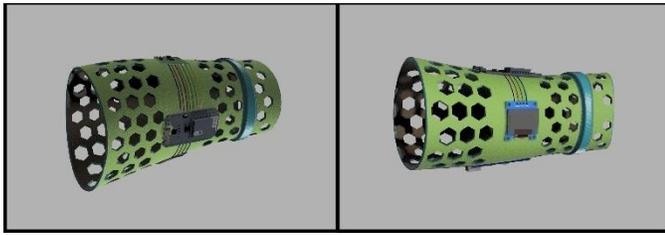


Fig. 3. 3-Dimensional modeling of electrical exposed parts.

The hardware implementation was simulated first in order to keep the measurement more precise for 3-Dimensional printing. A wearable device designed to keep in mind about elderly patients. The Device has a diameter of 181 mm to 252 mm and weight of 917 grams, and the 18650 rechargeable lithium ion battery is 3.7V. The standby time for the watch is seven days.

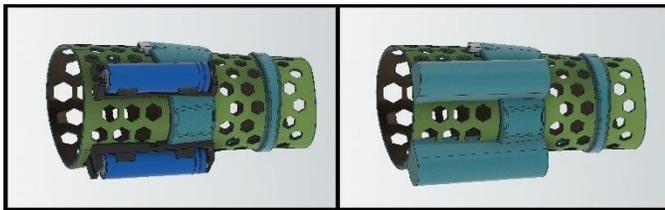


Fig. 4. 3-Dimensional modeling of battery holder with and without cover.

Flexible material for 3D printing was used to keeping in mind that it mainly will be used by the elderly peoples. Furthermore, the size was based on average hand size. Some dynamic style implemented to make in more user friendly by creating hexagonal holes which will facilitate the air and blood circulation standard once sporting this Device. A canopy was used to safeguard the electrical parts that increase the sturdiness of the wearable Device. By the essence of the raw material, the Device is slightly water-resistant.

**B. Software Architecture**

One of the most critical elements of a remote health control program is device convergence. This is the brain that leans to work flexibly according to the individual.

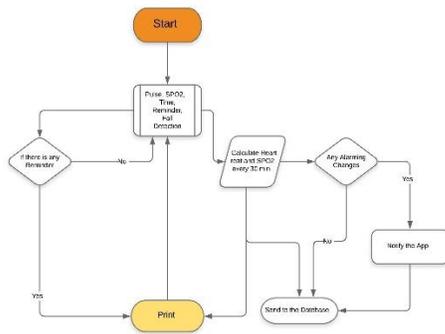


Fig. 5. Flowchart of the working principle of the software role.

The software architecture of the remote health monitoring system is based on the use of microcontroller Nano to use the Android Engineering tool. Arduino Nano IDE, an open-source program that helps to import the code into the development

board conveniently, extracts the details. The Raspberry Pi consequently transfers data to the cloud. The cloud service is Google Firebase, a software framework built by Firebase for the development of a web application. The project software was created for the Android Studio, and can access the Google Firebase database directly, but allows the user to encrypt it. It guarantees no unauthorized staff to access the patient's records. The accompanying example illustrates measures taken to build the remote health management system's software architecture. [5]

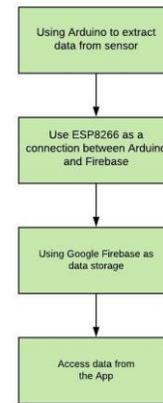


Fig. 6. Flowchart of the user interface.

**III. IMPLEMENTATION AND REVIEW OF THE RESULT**

**A. Measuring the heartbeat and SpO2**

This section collects important data from the patient. The theory is quite straightforward as the signal starts to drop, save a timestamp; since there are two timestamps, the distinction between them is which we can measure the BPM easily. We will measure the BPM after getting two timestamps. The following is the equation:

$$BPM = \frac{6000}{Currentbeattimestamp - Previousbeattimestamp}$$

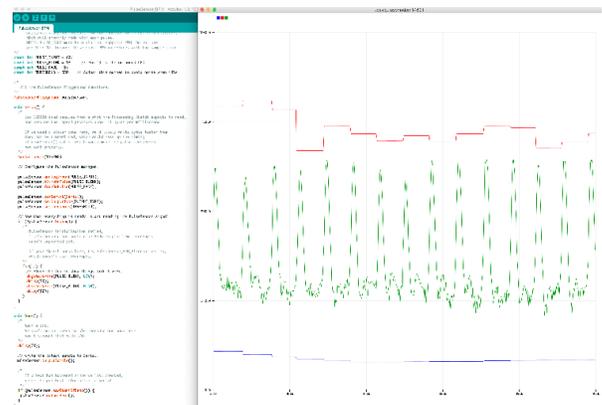


Fig. 7. PPG Signal displayed in Arduino Serial Plotter.



The oxygen concentration (SpO2) can be determined by the measurement of the ratio of absorbed light on the IR LED to the Red LED. Theoretically, the ratio of hemoglobin oxygenated to the overall hemoglobin amount may be measured [6]

Our Device is a module that tracks the BPM, SpO2, and temperature of the patient continuously. It needs an exact value to ensure that patients can be well treated so that data from 4 people are taken with our Device using different sensors and similarly with a certified device from the government. Therefore, we found that the data was close. The table I indicates the difference.

TABLE I. SHOWING BPM AND SPO2 WITH OUR DEVICE AND CERTIFIED DEVICE

Our Device	Result		Certified Device	Result	
	BPM	SpO2%		BPM	SpO2%
Person (1)	114.32	89	Person (1)	93	95
Person (2)	103.65	91	Person (2)	90	94
Person (3)	88.33	98	Person (3)	82	96
Person (4)	93.11	93	Person (4)	72	97



Fig. 8. Data taken from a commercial device.

**B. Detecting Orientation and Heart Rate**

First and foremost, the wire library was initialized and the sensor was reset via the power control register. Through the setup data, we have chosen a full set for the accelerometer and gyroscope. We had used the default scale +/- 2 for the gyroscope and 250 degrees for the accelerometer. The rest we commented on the section of the code.

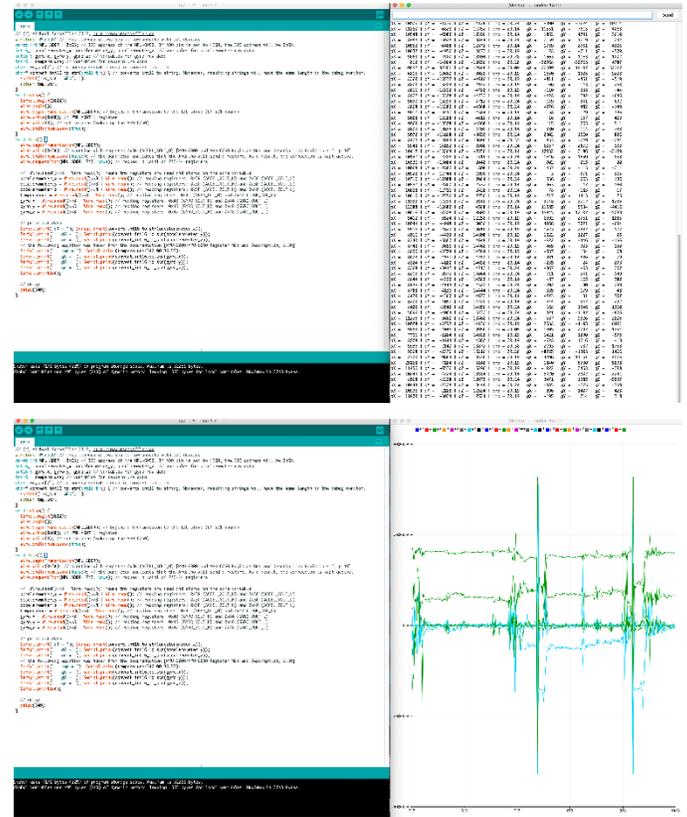


Fig. 9. Reading of Gyro Sensor.

Nine connections from the IC breaks by the AD8232 Heart Rate Monitor. We used jumper wires to connect that with the Arduino. After all the connections were done, we snapped the sensor pads on the leads before applying that in the demo patients' body. We got better measurement after placing the pad as near as possible to the heart.

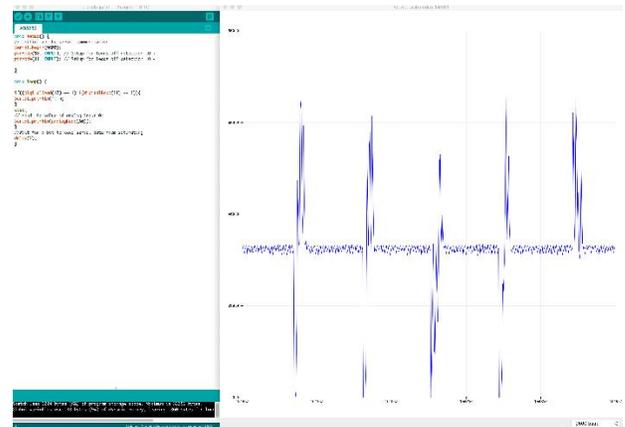


Fig. 10. ECG signal acquired from patient showing random variations.

**C. Measuring Body Temperature**

We use LM35 to detect the body temperature in our proposed project. This gives an exact temperature regardless of the constant relation of the output voltage to the temperature in Celsius. No proper calibration tests are required. The temperature scale is from -55 c to +150 c.

```

temp
const int sensor_A02; // Assigning analog pin A0 to variable sensor
float temp; //variable to store temperature in degree Celsius
float temp_f; //variable to store temperature in Fahrenheit
float vout; //temporary variable to hold sensor reading
void setup()
{
  pinMode(sensor_A02); // Configuring pin A0 as input;
  Serial.begin(115200);
}
void loop()
{
  vout=analogRead(sensor_A02);
  temp=(vout*5.0)/1023; // Converting value to Degree Celsius;
  temp_f=(temp*1.8)+32; // Converting to Fahrenheit;
  Serial.println("Temperature:");
  Serial.println(temp);
  Serial.println(temp_f);
  Serial.println("Fahrenheit:");
  Serial.println(temp_f);
  Serial.println("Celsius:");
  Serial.println(temp);
  delay(1000); //Delay of 1 second for ease of viewing
}

```

Fig. 11. Reading from body temperature sensing.



Fig. 13. Display of health parameter monitoring page.

#### D. Android Application of the Project

All the sensor implementations were integrated into a single operating device after all the individual implementation. The data has been saved for access by an Android App on the Google Firebase Real-Time database.



Fig. 12. Android Application titled Alzheimer's Assistant.

The Android framework has been created with Android Studio. This has tools to monitor a patient's health status and enables personal supervision. Yet something in the cloud is likely to leak. To stop this, the code is encrypted for authentication purposes with login credentials. The user can also select the tracking types. The server has a special ID which a patient or caregiver can himself/herself only authorize. The figure above shows the function of this type. The Android phone shows a screen of real-time details of body temperature, the heart rate and the SpO2, as seen in the figure above, once the caregiver monitoring the patient can access the patient's ID. Therefore, by actually tracking trends in the health condition, the caregiver will look after the patient by reviewing the previous data.

#### IV. CONCLUSION AND FUTURE PLAN

The measured values were obtained from a typical patient with the sensors during the experiment. However, these parameters vary accordingly and are co-related with each other with variations with their physical activities. Through the use of the app and the use of the database, the caregiver can access these data and take proper steps and give them the necessary care and medication. As this project is based on a microcontroller, the performance will be somewhat constant, and it will be easy to add updates to the code. We have also made a printed circuit unit for complex wiring to minimize the use of jumper cables so that there is a little chance of having loose connections and connection errors. In addition to that, the Device will meet its expectations in the long term and will have a significant impact on society in the future.

#### REFERENCES

- [1] M. Al-khafajiy et al., "Remote health monitoring of elderly through wearable sensors," *Multimed. Tools Appl.*, pp. 24681–24706, 2019.
- [2] M. Susha Cheriyaedath, "Photoplethysmography (PPG)", *News-Medical.net*, 2019. [Online]. Available: [https://www.news-medical.net/health/Photoplethysmography-\(PPG\).aspx](https://www.news-medical.net/health/Photoplethysmography-(PPG).aspx). [Accessed: 07-Dec-2019].
- [3] Y. Sun and N. Thakor, "Photoplethysmography Revisited: From Contact to Noncontact, From Point to Imaging", *Semanticscholar.org*, 2019. [Online]. Available: <https://www.semanticscholar.org/paper/Photoplethysmography-Revisited%3A-From-Contact-to-to-Sun-Thakor/a7c2cf9bcda344787ac6abca38f3077dc0d0e71/figure/1>. [Accessed: 04- Dec- 2019].
- [4] "Galvanic Skin Response (GSR)", *Brainsigns*, 2019. [Online]. Available: <https://www.brainsigns.com/en/science/s2/technologies/gsr>. [Accessed: 05- Dec- 2019].
- [5] M. Hamim, S. Paul, S. I. Hoque, M. N. Rahman and I. Baqee, "IoT Based Remote Health Monitoring System for Patients and Elderly People," 2019 International Conference on Robotics, Electrical and Signal Processing Techniques (ICREST), Dhaka, Bangladesh, 2019, pp. 533-538, doi: 10.1109/ICREST.2019.8644514.
- [6] R. Strogonovs, "Implementing pulse oximeter using MAX30100 - MORF - Coding And Engineering," *morf.lv*. [Online]. Available: <https://morf.lv/implementing-pulse-oximeter-using-max30100>. [Accessed: 01-Jul-2020]