Channel State Information (CSI) analysis for Predictive Maintenance using Convolutional Neural Network (CNN)

Prachi Bagave
Jeroen Linssen
Wouter Teeuw
Saxion | University of Applied Sciences
Enschede, Netherlands
{p.bagave,j.m.linssen,w.b.teeuw}@saxion.nl

Jeroen Klein Brinke
Nirvana Meratnia
University of Twente
Enschede, Netherlands
{j.kleinbrinke,n.meratnia}@utwente.nl

Abstract

With the onset of the fourth industrial revolution, predictive maintenance using wireless sensing technologies has been in high demand. This motivates to investigate the potential of WiFi CSI as a sensor for understanding the operation of machines. Since rotating motors are one of the fundamental elements in many complex machines, this paper focuses on the classification of CSI signals influenced by rotating motors at different speeds. As WiFi CSI technology is still not mature, we focus on data collection and study the sensitivity and reliability of data for this type of applications. We observe that CNNs are suitable to classify the speeds of motors and is also sensitive to speeds close to each other when operated in ideal network condition. However, in practical network conditions, unreliability of the data and the inability of CNN to classify it remains a challenge.

Keywords WiFi Channel State Information (CSI), convolutional neural networks, servo motor, stepper motor, wireless predictive maintenance

ACM Reference Format:
Prachi Bagave, Jeroen Linssen, Wouter Teeuw, Jeroen Klein Brinke, and Nirvana Meratnia. 2019. Channel State Information (CSI) analysis for Predictive Maintenance using Convolutional Neural Network (CNN). In *The 2nd Workshop on Data Acquisition To Analysis (DATA'19), November 10, 2019, New York, NY, USA.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3359427.3361917

1 Introduction

With the surge in the fourth generation of industrial revolution, predictive maintenance has become a hype. Companies consider it as a necessity these days [3]. As IoT is also emerging at the same time, the two technologies merge together into the so called Industrial IoT (IIoT). With IIoT, it is possible to have a wireless network connecting IoT sensor devices to collect data continuously and share it over the network for the purpose of maintenance [4]. Later, machine learning techniques can be used to analyze the data from the sensors to detect any anomaly in the machines.

One of the challenges in predictive maintenance of machines is installation of sensors. Wireless predictive maintenance provides a solution which is unobtrusive and realtime compared to the wired and manual data collection methods of predictive maintenance. Compared to reactive maintenance (done after the machine is broken down) and preventive maintenance (done before the machine is broken down), predictive maintenance is done just in time. Thus it reduces the downtime of production, thereby saving the costs involved with it. The existing market solutions for wireless predictive maintenance cost about a few hundreds to thousand dollars per sensor, use Bluetooth to communicate wirelessly or are based on vibration and temperature sensing [1, 5, 6]. As WiFi is readily available at most places these days, in this paper we investigate the potential of channel state information as a rich and inexpensive source of wireless sensing for predictive maintenance which senses movement of objects and has a range of a few metres.

In the past decade, a number research papers have been published on using the WiFi CSI for human activity detection. These include coarse grain activity recognition like identification of people on the basis of their gait [10], classification of activities like jumping, walking and running [11], and fine grained activity recognition like keystroke detection [7] and heart rate detection [9]. These studies prove that WiFi CSI is sensitive to human activities on a broad scale and could be successfully used for such applications. To the best of our knowledge no study exists in the domain of predictive maintenance using WiFi CSI.

The presence of metals, electronic appliances and large floor areas are a few challenges which makes it difficult to study this area of research. Thus, to study this approach, we

DATA'19, November 10, 2019, New York, NY, USA

ACM ISBN 978-1-4503-6993-0/19/11.
https://doi.org/10.1145/3359427.3361917

start off with a simple office environment to conduct the experiments. Industrial machines usually perform activities like churning, mixing, grinding and transportation [4]. These motions involve combinations of rotational motion. To imitate this behaviour in a simple way, we use rotating motors for our experiments. This paper performs experiments to investigate the WiFi CSI behaviour in terms of complexity, reliability and sensitivity. To analyze complexity, we use two different motors which vary in number of moving elements attached to them. To analyze sensitivity we let motors operate with different speed ranges and to analyze reliability of the system we examine consistency of the data sets under real-life network conditions.

2 Experimental Setup and Data Acquisition

Experiments were performed in an office environment, keeping the motors in direct line of sight between the transmitter and receiver at a distance of about 25cm. The motors were controlled by Arduino board using the software drivers. We used 'Gigabyte Brix IoT' miniPC for setting up the WiFi network. One PC was dedicated for the purpose of transmission while the other for reception throughout the experiments. These WiFi nodes were kept on carton boxes on an office table. Two to three people were working on adjacent tables at a distance of about 2-3 metres while the experiments were performed. To maintain the height of the motor so as to keep it in the direct line of sight from the transmitter and receiver, it was mounted on a stand with PVC and/or cardboard material. Figure 1 shows an example of the setup for servo motor. Experiments were done in the same setup to avoid influence of other factors.

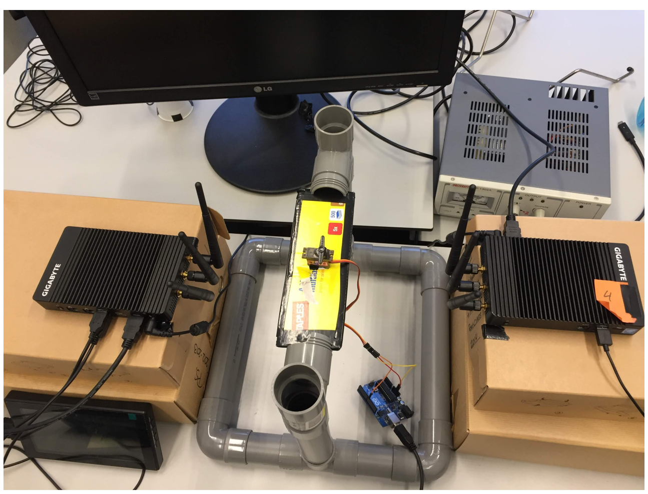


Figure 1. Experimental setup.

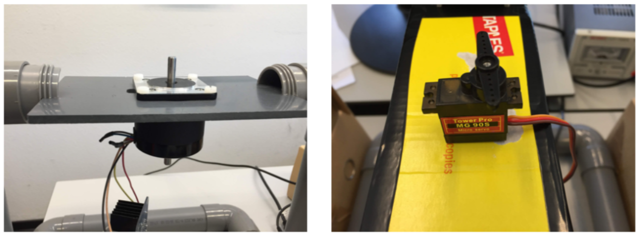
In a real life scenario, a WiFi connection could be interrupted due to a number of factors. This could be when connection is lost due to signal interference or even when the power is switched off during non working hours. Once the system is reconnected, system settings could be a bit different than the previous connection. A reliable system is expected to work irrespective of such interruptions. To study the reliability of the system to such interruptions, we deliberately stop the connection and connect it again multiple times while collecting data. This data is referred to as data with re-connections or under practical network conditions hereafter in the paper. We also collect data without any interruption in the connection and later compare the two types of data. This study is significant to understand the behaviour of the system to be used over multiple days, where we expect the network to have re-connections.

2.1 Connectivity

For setting up the WiFi connection, the mini-PCs were configured with Ubuntu 14.04.4. They had Intel N Ultimate Wifi Link 5300 as the network interface card and CSI drivers provided by a 3rd party [2]. Experiments were performed with 802.11n protocol at 5GHz bandwidth with 64 QAM as the multiplexing technique. Two transmitting and three receiving antennas were used for communication. The communication was set up in injection mode were the transmitter has control over number of packets, size of packets, sampling rate, data rate and the channel number for communication. The packets were sent in broadcast mode. Any receiver listening on that channel could receive the packets. For the experiments, channel 64 was used with 100 bytes packet size, 1kHz sampling rate and 50 mbps data rate. For the experiments with stable network connection, once the connection was set up, data was collected for 5 minutes continuously, which was later separated into samples of 3 seconds each. On the other hand, for the experiments with network re-connections, data was collected for 3 seconds, the WiFi was stopped, reconnected with the same transmission parameters and the next sample was collected. This process was done iteratively.

2.2 Motor Details

We used two types of motors, a stepper motor Astrosyn MY180 and a servo motor Tower Pro MG 90S. The stepper motor needed 12V power supply, had a torque of 0.53N-m and was $57 \times 57 \times 91mm$ in size. And the servo motor weighed 13.4g, needed 4.8-6V power supply, had a torque of 0.17-0.21 N-m was $22.8 \times 12.2 \times 28.5mm$ in size and had a gearbox attached with it. Thus the stepper motor was powerful in terms of size, torque generation and power supply requirements. And the servo motor was complex in terms of moving elements since it had gears. While working with the Arduino software drivers, it was noticed that the servo motor could be configured with a limited range of operating speeds with fine tuning, thus the experiments were performed on 0, 2.67, 3.26, 4.05, 5.38, 8.02 and 14.45 rpm. On the other hand, stepper motor was used with speeds of 0, 50, 150, 250, 350 and 450 rpm. The stepper motor could make full rotations, but



Stepper motor Servo motor

FIGURE 2. A close view of motors at the experiment.

the servo motor moved alternately 180 degrees clockwise and then 180 degrees anticlockwise with the same speed. Figure 2 shows the motors at experiment. Both the motors had a single shaft rotating along its axis. Rotating motor without load is like an axis rotating along itself in a single dimension. This might have different effects when compared to a load (plastic arm as in Figure 2, servo motor) rotating in a plane (two dimensional). To study these two scenarios we performed experiments with and without the plastic arm for both the motors.

2.3 Experimental Data and Primary Analysis

In total we studied the motors for two load conditions (with and without arm propeller) and two network conditions. Thus we had four scenarios for each motor. For each scenario, motors were experimented on 6-7 different speeds. We used a classification model which classified these speeds. Hence each speed was a separate class for the model. For each class, 100 samples of data were collected, each 3sec long with a sampling frequency of 1kHz. Thus each data sample was 3000 packets of CSI data. The dataset could be found on [8].

To understand the CSI response to different speeds, we studied the overall stability of data by calculating the statistics of all the classes and compare with each other. One of the best ways to visualize such a big data is to have box plots where edges of the box represent $25th$ and $75th$ percentiles, centre of the box is the mean and the red marks are outliers as in Figure 3. The overall variance, standard deviation and mean for data(all classes combined) with and without network re-connections is [68.7696, 8.2927, 20.31] and [93.75, 9.68, 21.81] respectively. The overall variance of values for data with network re-connections is a bit lower than the data with stable network conditions. The variance and standard deviation for individual classes for the non-reconnecting network data was [93.97, 92.39, 92.02, 107.66, 75.78, 96.76] and [9.69, 9.61, 9.59, 10.37, 8.71, 9.83] respectively. Similarly, for reconnecting network data, it was [65.6835, 68.6108, 72.7494, 71.4834, 70.1162, 75.6701] and [8.1045, 8.2832, 8.5293, 8.4548, 8.3735, 8.6989]. These values for individual classes do not show any pattern with the speeds of motors and are also very close to each other which makes it difficult to differentiate with each other. This motivates the use of noise reduction techniques to help in the classification process.

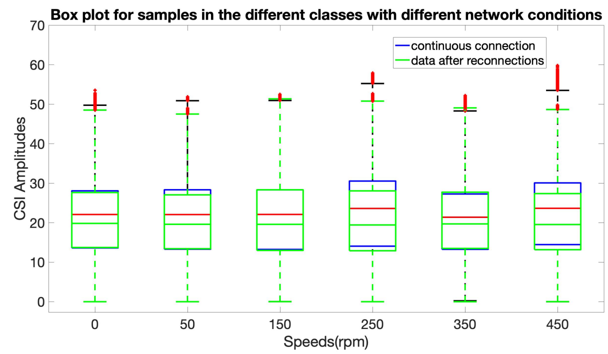


FIGURE 3. Box plots for individual classes of stepper motor under different network conditions.

In order to have a visualization of the actual signals we plot the time series and frequency domain data for both the motors in Figure 4. It shows the data for 30 sub-carriers of the servo motor at 14.45 rpm and the stepper motor at 50 rpm after 50-point average smoothening and its corresponding Short Time Fourier Transform (STFT) spectrograms. The mean of the variances for individual signals for the servo and stepper motors are 0.639 and 1.625 respectively. From the figure also we can see that stepper motor shows more amplitude changes than the servo motor. This may be due to the higher power consumption of the stepper motor since it had higher voltage and current ratings. It is also noted that the raw time series data seems to be much more distinctive compared to the respective spectrogram data.

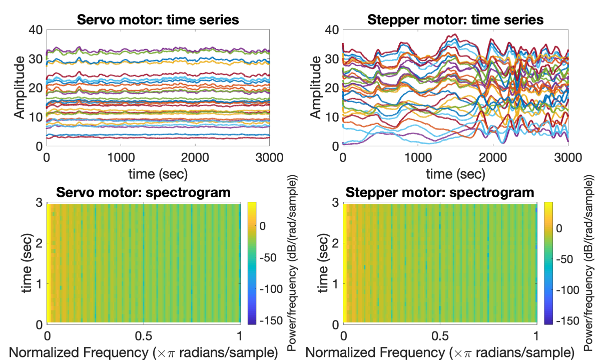


FIGURE 4. Time series and spectrogram plots for the stepper and servo motors.

3 Methodology for Data Analysis

Each received CSI packet was a matrix of 30 complex values (corresponding to the 30 sub-carriers) times the number of antennas. Since we used 2 transmitting and 3 receiving antennas, we got 2*3*30 = 180 channels of information. Instead of using the complex numbers, we used its amplitude information at all times. Each image in the CNN network was expected to be of the size $(Fs \times time(sec)) \times dimensions$ (3000*180). To simplify the the calculations with 180 dimensional data and reduce the noise factors following steps were performed with the raw data as shown in Figure 5.

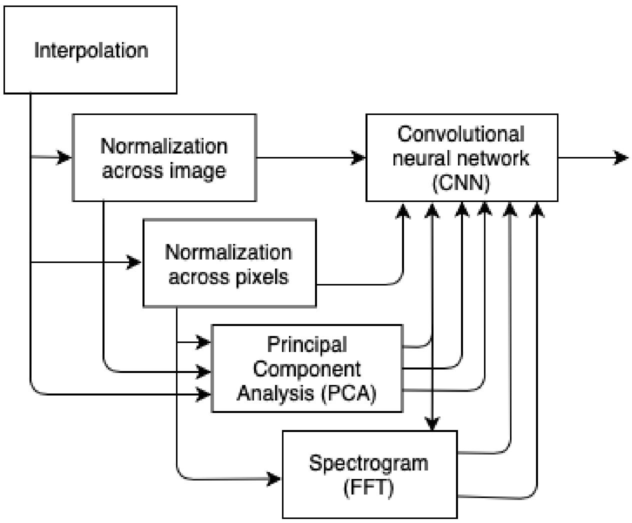


FIGURE 5. Block diagram for the methodology of data analysis.

1. Interpolation: Ideally, for each sample of 3 sec at a sampling frequency of 1kHz, 3000 samples are expected to be received. Due to packet loss this might not be possible for every sample. This causes uneven number of packets resulting in irregular image size. To reduce the bias towards uninformative data, only the samples having more than 2500 packets were interpolated to have 3000 equally spaced samples.
2. Normalization: Normalization of a data set could be helpful to remove redundancy of data which is spread over different scales. For image data set it could be performed in two ways.
   - Normalization across images: In this method we normalize each image individually.
   - Normalization across pixels: If we consider each pixel of an image to be an individual dimension, and normalize each dimension, each pixel gets normalized with respect to the corresponding pixels in other images.
3. Principle component analysis (PCA): PCA is used as a dimensionality reduction technique to reduce the 180 dimensional data to 20 by taking the first 20 significant components. It is performed on three inputs of data, namely, the raw data, data normalized across images and the data normalized across pixels. The image size after PCA reduces to 3000×20.
4. Spectrogram: Spectrogram is calculated by taking one dimensional Fast Fourier Transform (FFT) along the direction of time. The images after FFT are shaped as 3D images of size $samples \times subcarriers \times antennapairs$ for faster computations. This stage has two types of inputs, normalization across pixels and the same data after PCA with the first 50 significant components. The image size after this stage is either 3000*30*6 or 3000*50 (in case of PCA).
5. Convolutional neural network (CNN): There are a total of 7 models each corresponding to the input data. The input size of each model varies. The models have 6 layers with alternate layers of convolutional layer and average pooling layer. The convolutional layers have kernel sizes of [(100,10),(50,5) and (30,3)] with 70, 50 and 30 depth size. The model uses a Gradient Descent Optimizer with learning rate of 0.001.

4 Results and Discussion

In this section we first discuss the results of stepper motor and later we move on to servo motor.

4.1 Stepper motor

Classification accuracies for the four cases namely, no load with ideal network conditions, arm propeller load in ideal network conditions, no load in practical network conditions and arm propeller load in practical network conditions are presented in Figure 6. For the stepper motor we have six classes. This means that for random allocation of any class, the accuracy could be as low as 16.66%. For the data without network reconnection, all the methods except normalization across pixels show low accuracy. For normalization across pixels accuracies are 54.16% and 70.00%. After applying PCA these are improved to 73.33% and 81.66%. Data with plastic arm shows higher accuracy than the no load condition. Results with network reconnections have almost random prediction for all cases. The highest accuracy achieved in this case is 32.53% for no load condition when applied normalization across pixels and PCA, which is also very low for practical use.

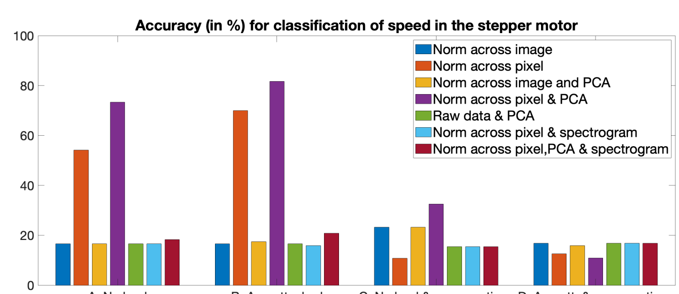


FIGURE 6. Classification accuracies for stepper motor.

The best classification accuracy for the stepper motor is 81.66% when the data is normalized across the pixels and PCA is used for dimensionality reduction. The distribution of values of an image when compared to images from other classes is informative. PCA is able to refine this information by clearing out the noisy part and by reducing the dimensions. PCA also reduces the training time from about 6 hours to 20 minutes while reducing the computational costs. The rotating fan in a two dimensional space (a plane) has more effects on the WiFi CSI when compared to no load condition. This could be due to the fact that the fan points at different directions while rotating, making a pattern in the signals. Comparing these results with the data with multiple reconnections, it is observed that data in this case gets stabilized in a way that the information cannot be gained even after normalization or by performing PCA. For a system to be reliable it is important that a model could be used over a long period of time irrespective of the network conditions which is not possible with this approach.

4.2 Servo Motor

Classification accuracies for the servo motors for the similar four cases is presented in Figure 7. The classification model for the servo motor had 7 classes. For random prediction of any one class, the model could have a prediction accuracy of around 14.28%. For experiments performed under stable network conditions, normalization across pixels gives fairly high accuracy of 81.42% and 90.00%. Applying PCA and performing spectrograms after normalization across pixels reduces the performance of the model. Motor with the plastic arm has slightly higher accuracy than the one without the arm. Experiments performed under the practical network conditions, show random prediction for all cases except the case with PCA on normalized data across pixels. For this case it has accuracies of 29.91% and 25.54% which is a little improvement over others but still low to be relied on for classification.

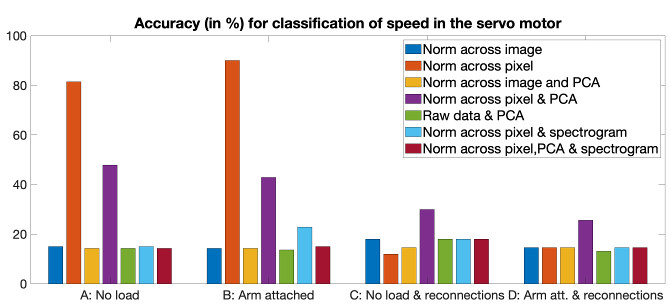


Figure 7. Classification accuracies for servo motor.

The differences between the speeds of servo motor were at magnitudes between 0.59 rpm for the slower speeds and 6.43 rpm for the faster ones. WiFi CSI is sensitive enough to predict motor speeds which are so close to each other. Since this motor had multiple gears attached to the shaft, received signal is a combination effect of all the moving elements. The first 20 components of PCA in the case of normalization across pixels might not be informative enough, thereby reducing the accuracy. The spectrograms provided might have a lot of noisy components, making the classificaion difficult. Since we get 90% accuracy for the motor with the attached arm for speeds as close as 0.59 rpm, CNN models could be said to be suitable for complex machines with multiple moving elements to detect change of motion smaller than one rpm. Whereas the CSI data itself does not seem to be consistent once the network is reconnected.

5 Challenges in analyzing the data

CSI is very sensitive to small changes. Thus small environmental changes have a lot of influence on the signals and may add noise. Proper noise reduction techniques, especially for sensitive data is a big challenge. Spectrograms for such data are also very alike which makes it very daunting for classification. Since CSI is 180 channel data, visualizing the data is difficult. Thus anticipating the important features and modelling them is impractical using this approach. As a preliminary approach, we studied the overall data by its statistical information like variation and standard deviation to design a simple approach for analysis.

6 Conclusion and Future Work

Classification of motor speeds with CSI data is possible with CNN models when the images are normalized across the pixels. Accuracy as high as 82% and 90% could be achieved for classifying motor speeds as close as 0.59 rpm. Thus CSI is sensitive to very small changes. But, noise reduction, dimensionality reduction and information extraction might be challenging while using this sensor technique. Since both the motors have different results with the same model, every equipment in the industry needs a separate analysis and no single model could fit all equipments. This system is also suitable for complex moving elements. But reliability of the data after the network reconnects still needs detailed investigation. While the possibility of such a technique is validated in this paper, the application in an actual industrial area remains open for future work.

7 Acknowledgement

Work done in this project is a part of RAAK MKB Data in Smart Industry project. This work was initiated at the University of Twente under the supervision of Paul Havinga & Nirvana Meratnia and later continued at Saxion, University of Applied Sciences.

References

[1] [n.d.]. HM11:Machine Health Monitor. https://www.samsara.com/products/models/hm11. Accessed: 2019-09-19.

[2] [n.d.]. Linux 802.11n CSI Tool. https://dhalperi.github.io/linux-80211n-csitool/. Accessed: 2018-04-25.

[3] [n.d.]. The Power of Wireless Predictive Maintenance. https://petasense.com/power-wireless-predictive-maintenance/. Accessed: 2019-07-30.

[4] [n.d.]. The Power of Wireless Predictive Maintenance. https://iiot-world.com/predictive-maintenance/the-power-of-wireless-predictive-maintenance/. Accessed: 2019-07-30.

[5] [n.d.]. Sensor Works: A health monitor for machines. http://sensor-works.com/sensorworks/. Accessed: 2019-09-19.

[6] [n.d.]. Wireless Predictive Maintenance System. https://axzon.com/wireless-predictive-maintenance-system/. Accessed: 2019-09-19.

[7] Kamran Ali, Alex X. Liu, Wei Wang, and Muhammad Shahzad. 2015. Keystroke Recognition Using WiFi Signals. In *Proceedings of the 21st Annual International Conference on Mobile Computing and Networking (MobiCom '15)*. ACM, New York, NY, USA, 90–102. https://doi.org/10.1145/2789168.2790109

[8] Wouter Teeuw Jeroen Klein Brinke Nirvana Meratnia Prachi Bagave, Jeroen Linssen. 2019. Dataset: Channel State Information (CSI) analysis for Predictive Maintenance using Convolutional Neural Network (CNN). https://doi.org/10.5281/zenodo.3452876

[9] P. Wang, B. Guo, T. Xin, Z. Wang, and Z. Yu. 2017. TinySense: Multi-user respiration detection using Wi-Fi CSI signals. In *2017 IEEE 19th International Conference on e-Health Networking, Applications and Services (Healthcom)*. 1–6. https://doi.org/10.1109/HealthCom.2017.8210837

[10] Wei Wang, Alex X. Liu, and Muhammad Shahzad. 2016. Gait Recognition Using Wifi Signals. In *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing (UbiComp '16)*. ACM, New York, NY, USA, 363–373. https://doi.org/10.1145/2971648.2971670

[11] W. Wang, A. X. Liu, M. Shahzad, K. Ling, and S. Lu. 2017. Device-Free Human Activity Recognition Using Commercial WiFi Devices. *IEEE Journal on Selected Areas in Communications* 35, 5 (May 2017), 1118–1131. https://doi.org/10.1109/JSAC.2017.2679658