MSc ASE Project Report

Audio Location System

Simon King

A thesis submitted in part fulfilment of the degree of

MSc in Advanced Software Engineering

Supervisor: Chris Bleakley

UCD School of Computer Science and Informatics
College of Science
University College Dublin

February 26, 2018

Table of Contents

Abstract

This document is a layout and formatting template for your ASE project. It's there to help you, but change it as much as your want!

What is an abstract? The abstract should provide a short overview of your project that enables a reader to decide if your report is of interest to them or not. It should be concise, to-the-point and interesting. Avoid making it read like a verbose table of contents. Avoid references, jargon or acronyms, as the reader may not be familiar with them. An abstract usually contains a brief description of:

- The project and its context;
- How the project work was carried out;
- The major findings or results.

One paragraph is plenty. The main thing to remember is the principle that the abstract must be short, and a person reading it should be able to determine if they want to read more. For example, if your project involves building a compiler for Java, and a major section of your work is focussed on developing an efficient parser (rather than say code-generation), make this clear in the abstract. Then a reader who is interested in efficient parsing techniques knows that your report may be of interest to them.

Acknowledgements

In your Acknowledgements section, give credit to all the people who helped you in your project.

Chapter 1: **Introduction**

Chapter 2: **Indoor Localisation System**

Indoor localisation technologies are a widely researched topic in recent years, with a focus on developing a method for widescale deployment. With the growing availability of advanced mobile devices and wireless infrastructure in public areas, accurate indoor localisation systems within places such as hospitals, shopping centres, warehousing etc. without the need for bespoke hardware is more feasible than ever. Although global positioning system (GPS) works extremely well for an open-air Localisation, it does not perform effectively in indoor environments due to the disability of GPS signals to penetrate in-building materials (*1*). Being able to pinpoint the location of a wireless devices requires higher resolution for indoor environments than in outdoor applications (*2*). In this section, we will examine some of the approaches implementing solutions for the indoor localisation problem and some of the main issues in implementing such systems. We focus on a solution that uses ultrasonic audio signals to attempt to locate a device with millimetre accuracy. There is no system that can be used for every application under all environmental conditions. From the point of view of usability and accuracy, it is preferable to use a system that performs localisation by using the propagation delay time of electromagnetic waves, based on a principle like that used in GPS (*3*). Ultrasonic is a well-known ideal candidate for indoor positioning that relies on the time of flight (TOF) scheme. The key idea is to use an ultrasonic transceiver to emit and detect ultrasonic signals (*4*). Ultrasound is already commonly used for distance measuring in ultrasound parking assist. By recording the time it takes for an ultrasonic signal to travel between a transmitter and receiver, it is possible to compute the distance, given the medium travelling speed. Using c to represent the speed of sound together with the time of flight, t_f, the distance, d, can be calculated using

$$d = ct_f$$

The TOF of each vector between one of the anchors and the mobile device is measured through its time of arrival (TOA) by finding the peak of a correlation result (*5*).

Time difference of arrival (TDOA) works similarly to the system described above, but uses a radio synchronization packet followed by an ultrasonic pulse. The distance between nodes is calculated by the travel time of the ultrasound signal. Within a localised system, this measurement can be performed between a device and at least three beacons and the position of the mobile device can be found using trilateration (*4*). These systems are made up of a number of fixed nodes or beacons and mobile nodes, where the mobile nodes try to calculate their position based on the known positions of the fixed nodes. One disadvantage of this system is the considerable amount of fixed position nodes needed, which increases the setup cost (*6*). Ultrasound can provide high Localisation precision using this system however, it suffers from the line-of-sight restrictions. Given the system requires the mobile device to communicate with at least 3 fixed nodes to get an accurate measurement, the placement of fixed nodes must account for obstacles like desks and chairs to communicate with the mobile devices. Beacon placement thus becomes challenging for ultrasound-based indoor Localisation in environments with various obstacles (*7*). Determining the node positions also requires manual calibration which is time consuming, since each anchor has to be measured individually, and fault-prone, because of inaccurate measurement methods and human error (*8*). The angle of arrival (AOA) technique uses an array of receivers to evaluate the incoming reception angle. Calculating the location of the source is done by combining the angles of different receivers (*6*). Relative Received signal strength (RSSI) is a WiFi based indoor Localisation system which uses the signal strength of radio communication to estimate the distance between devices (*6*). The main idea is to measure a set of signals signatures, known as fingerprints, based on different locations in the area of interest and build a fingerprint database. The location is then estimated by mapping the measured fingerprints against the database. This approach requires a considerable manual effort to build the fingerprint database and the resulting system is relatively inflexible to changing environments (*9*). RSSI is not well suited to tracking users in real time, due to the lengthy time taken to calibrate for channel propagation parameters (*10*). (*11*) and (*9*) propose methods of tracking users movements within a space with wearable devices embedded on the person or user motions from mobile phones. Implementing accurate indoor Localisation faces a number of problems technologically to implement. A good solution should be low cost, scalable, robust and easy to deploy. It should also be able to cope with changing environments. The localisation accuracy depends mainly on four factors: Accuracy of the range measurements, location errors of the anchor nodes and geometric configuration of the system. There also exist trade-offs among the positioning accuracy, computational complexity, cost and power consumption (*12*). (*2*) finds the deployment cost of these various systems to be one of the largest problems to overcome. Out of 22 solutions compared, the average setting and calibration time is 5 hours for two rooms covering 300 square meters. This may be unrealistic and intrusive when deploying these Localisation systems in large deployment sites like shopping malls. (*3*) proposes an indoor Localisation which is realized with as few initial references as possible, based on the idea of iterative multilateration. When such a Localisation method is used, deterioration of localisation accuracy due to no line-of-sight signals and to accumulated errors is a problem.

(*7*) describes many of the issues in deploying an ultrasound based Localisation system. In general, ultrasonic wave emission is usually directional, which introduces difficulties in orienting the transceiver precisely. Only when the listener is inside the transmitters beam pattern, can the listener derive distance measurement value from the beacon. One solution to this issue is described in (*13*). They developed a 2-D isotropic ultrasound transmitter with a beam width of 360 using an array of eight ultrasound transducers placed in a round body. The angle of aperture of one transducer is 45. By using eight ultrasound transducers, the coverage range can be increased by a factor of 8. (*7*) also determines that more beacons are required grows linearly with the size of the area of interest.

Chapter 3: **Measuring distances using audio**

Measuring distance using ultrasonic signals offers an inexpensive solution to the indoor localisation problem. Measurements of the distance of an object from fixed landmarks can be used to calculate the precise location of that object. The basic idea is to send an acoustic signal through the air from a transmitter to a receiver. The time it takes for the signal to reach the receiver is known as the time of flight (TOF), which can be used to measure the distance by the following equation:

$$d = ct_f$$

Where d is the distance, t_f is the time of flight and c is the speed of sound in air. Many ultrasonic applications use a single transducer, which both emits and receives, to calculate the distance to an object by emitting a signal and measuring the time it takes for an echo to be received. The distance is calculated similarly to the calculation above, but having the result, as the signal has travelled the distance to the object and back;

$$d = (ct_f)/2$$

For this comparison, we will consider the measurements are using the second technique, which allows us to know when the signal was sent, although in more complex ranging systems that use independent transmitters and receivers, synchronisation becomes an important factor, which we will examine later. The accuracy of the distance measurement depends on the accuracy of the measurements of the TOF and the speed of sound in air. Choosing the measurement technique of the TOF depends on a number of factors such as cost, ease of implementation, environment. We will examine some common methods used for these TOF measurements (*14*). Methods are usually compared by the accuracy, or error in the distance measurement, repeatability, or the variance in repeated measurements, cost of implementation and performance under noise.

3.1 Time Domain methods with single frequency signals

The first method we look at is thresholding. A signal is sent, and the time of flight is the time it takes for the amplitude of the echo signal to surpass a certain threshold. The threshold is usually selected to be well above the noise standard deviation. This method is relatively straightforward and does not require complex circuitry or calculations, and can be implemented with inexpensive transducers. Its limitation is it naturally introduces a bias into the measurement. If the threshold level is set to low, noise interference can cause false positive to be detected. Increasing the threshold will improve detection of real echoes, however the time it takes for the incoming signal amplitude to surpass the threshold will be delayed further. This is especially true if too low a sampling frequency is chosen. Although noise and sampling frequency selection are issues that all TOF measurements must deal with. Curve fitting is a method of TOF estimation that attempts to fit a parabolic curve to the echo signal envelopes leading edge to provide a measurement without bias. This uses threshold estimate as above, but then fits a parabolic curve in the form $a_0(t - t_0)^2$ where t_0 is the estimation of the result of simple thresholding and a_0 is estimated from a second derivative approximation around this threshold point. A nonlinear least-squares method is then applied to fit the curve, and the vertex of the parabola is used as the measurement of the TOF. Sliding window is a method that can be used to make detection more robust. A window of length N is slid through the echo signal one sample at a time. As the window slides through the sample, it counts the number of samples which exceed the threshold value. If this count exceeds second threshold the signal is considered present and the TOF estimate is produced. Cross-correlation is an unbiased measure of TOF. The echo signal is correlated with a matched filter that contains the waveform and the delay will be the peak. This method has a few drawbacks in comparison with those outlined above. In real-time processing, the entire echo must be observed before the correlation process can being which can add a significant delay to producing the estimate. It is also computationally much more complex than the previous methods. However, this method should significantly reduce noise interference. A study performed by (*14*) shows that Correlation gives by far the best results in terms accuracy, which is what would be expected, however the less complex methods offer some acceptable performance results at much lower cost. Curve fitting performing best when it came to bias and total error, whereas sliding window performed best with standard deviation. Other methods of measurement Time difference of arrival (TDOA) can be used as part of a multilateration system to predict the location of an object. A transmitter sends a signal which is received at receiver stations 1 and 2, which are at known locations. When both receivers have received the signal, the can cross-correlate them to determine the time shift between the two waves which is the difference in time it took the signal to at each station. This time shift can be used in equation 1 to get a measure of distance. As both stations are fixed, we now have an infinite number of points along a curve that satisfy the transmitters location. If we had a second pair of stations, we would get a second curve of possible locations that intersects the first. This produces a small number of locations that the transmitter could be. The ultrasonic waves from a small mobile device with an ultrasonic transmitter are received by the receiver array. The ultrasonic receiver array and the mobile device are synchronized by a wireless connection. The listener can then derive the distance from the beacon by multiplying the ultrasound propagation time by the speed of sound. Since the location of the receiver is known when the mobile device was deployed, the listener must be located at the surface of a sphere that is centred at the beacon and with a radius of the derived distance from the beacon to the listener (*7*). By multiplying the measured propagation delay time by the speed of sound, the distance between the mobile device and each of the ultrasonic receivers is derived. Since the locations of the individual receivers are accurately given beforehand, the location of the mobile device can be derived three-dimensionally by solving a set of simultaneous equations involving the measured distance to each receiver and the locations of the receivers. (*3*) However, this does require the system to be synchronised for accurate measurements between devices. Another approach using TDOA is described in (*4*). Distance is measured using the difference in time-of-ight of RF signals and US signals. A RF and US signal are sent simultaneously from a transmitter to a receiver. The RF signal travels much faster than the US signal. Over a distance of 10m, the radio signal takes 30ns and the ultrasound signal, will take 30ms. As the RF signal is much lower, the TDOA can be measured as the difference in time from when the RF signal arrives to when the US signal arrives. Unfortunately, the speed of sound is not constant. Indoors it varies mainly with temperature and can be approximated using

$$v_{US} = (331.6 + 0.6T)m/s$$

where T is the temperature in degrees Centigrade. The multilateration approach is popular in indoor localisation systems, however its accuracy is dependent on many factors such as the devices in the system being synchronised, interference from noise or multi-path interference, inaccuracies in the measurement of the locations of the fixed devices. Angle of Arrival (AoA) is used to determine the direction a received signal arrives at an array. It is measures by taking the TDOA between elements within the array. For example, lets consider an array of two microphones separated by half the wavelength of an incoming sinusoidal wave. If the wave was emitted from directly in front of the array, there would be no phase difference between the two measured waves. However, if the wave was emitted from the right of the array, the microphones would receive the signals half a wavelength apart, resulting in a phase difference on 180. If this was used alongside the multilateration system we described earlier, we could immediately disregard many of the values of the curve produced.

3.2 Synchronisation Signals

So far, we have looked at generic sinusoidal waves for transmission across the network. When these signals are cross-correlated with an expected waveform, we can see a peak at the lag position where the two signals overlap, giving us a measure for the delay time. What we can also see is a wide sideband of peaks which get larger the closer to the true peak. If we were to take the measurement for the lag be the maximum value of the waveform, we can see how a noisy signal could lead to incorrectly choosing the maximum. There are signals with autocorrelation properties which mitigate against this. One of the properties of Additive white Gaussian noise (AWGN) is that the auto-correlation values for any non-zero delay, i.e. where the signals do not align perfectly, is effectively zero. It also has zero cross-correlation with any other AWGN waveform. This is a very useful property in peak detection. Pseudo-random noise (PRN) sequences also display similar autocorrelation properties. At zero time-delay there is a peak, and at non-zero time-delays the values are very small. These PRN signals also carry timing information as well, which is extremely useful in distributed systems to coordinating device transmissions. In wireless transmission, a PRN sequence is used in setting up connections to detect and synchronise devices. A wireless access point will transmit a synchronisation signal. A matched filter in devices check incoming frames for this PRN sequence and aligns its local clock to the access points and sends back a signal, which the access point then scans for the PRN sequence, detecting the devices timing and instructs the device to adjust its transmit timing to account for round trip propagation. So far, we have determined that a sequence with good autocorrelation properties are useful in time-of-arrival detection. However, another useful property of these sequences is having zero or very low cross-correlation with the same signal at any delay. A complex PRN sequence has a periodic autocorrelation of

$$\gamma_{PN}\left(K_1, K_2\right) = \begin{cases} \frac{1}{N}\sum_{n=0}^{N-1} x\left[n-K_1\right] x^*\left[n+K_2\right] = -\frac{1}{N} \\ 1 \end{cases}$$

where N is the period of the PN sequence. Therefore, cyclically shifted PN sequences have a correlation $-1/N$ with the original sequence.

Gold Code - A Gold code, also known as Gold sequence, is a type of binary sequence, used in telecommunication (CDMA) and satellite navigation (GPS). Gold codes are named after Robert Gold. Gold codes have bounded small cross-correlations within a set, which is useful when multiple devices are broadcasting in the same frequency range. A set of Gold code sequences consists of 2n 1 sequences each one with a period of 2n 1. Gold codes are used in GPS. The GPS C/A ranging codes are Gold code of period 1,023. A Zadoff-Chu Sequence is a complex-valued sequence with some very useful properties in signal transmission. It is given by the equation

$$x_u\left(n\right) = \exp\left(-j\frac{\pi u n\left(n+1+2q\right)}{N_{ZC}}\right)$$

$$0 \leq \ n \leq \ N_{ZC}$$

$$0 \leq \ u \leq \ N_{ZC}$$

$$\gcd\left(N_{ZC}, u\right) = 1$$

Where N_{ZC} is the length of the sequence. When N_{ZC} is odd, the sequence is periodic If N_{ZC} is prime, the Discrete Fourier Transform of a ZadoffChu sequence is another ZadoffChu sequence conjugated, scaled and time scaled. The auto correlation of a ZadoffChu sequence with a cyclically shifted version of itself is zero, i.e., it is non-zero only at one instant which corresponds to the cyclic shift. The cross-correlation between two prime length ZadoffChu sequences. Zadoff-Chu is used in 3GPP LTE services for both synchronisation and random access preambles ZadoffChu sequences are an improvement over the WalshHadamard codes used in UMTS because they result in a constant-amplitude output signal, reducing the cost and complexity of the radio's power amplifier. Supposing we had a single nodes whose location is unknown is communicating with multiple fixed point nodes whose locations are known. The orthogonal nature of the ZC signals means that multiple cyclically shifted signals can be combined and sent simultaneously in a single transmission. If each receiver had a matched filter to look for the signal with a particular phase shift, the other signals in the transmission would not be detected.

Chapter 4: **Detailed Design and Implementation**

Chapter 5: **Testing/Evaluation**

Consult (*2*) for some excellent advice on how to present the results of your experiments.

Chapter 6: **Ongoing and future work**

A number of ongoing experiments with the various methods mentioned above are ongoing. The current system is being developed on a Raspberry Pi (*15*), a low cost computer which is widely available. Although it is low cost, it is powerful with processing and memory capabilities comparable of modern smartphone, but with the versatility of being able to expand functionality by adding additional components using the USB and GPIO connections. It also has ethernet, wireless and Bluetooth connectivity built in, which makes it ideal for selection in its use in a distributed, interconnected system such as the indoor localisation system. This project is using a single Raspberry Pi 3 initially, which is the latest, most powerful iteration, but does not cost much more than the earlier models available. In addition, a PiFi audio expansion board is being used to for additional audio capability. For the final system, an array of ultrasound transmitters and receivers will be used for the devices, but for early development and testing, audible frequencies are being used. A USB microphone acting as a receiver and a standard speaker connected by auxiliary cable as the transmitter. The code is being developed using Python. Python is a widely adopted, open source, general purpose programming language with the quick development process that is possible when using interpreted languages that are focused on signal processing applications (*16*). Programming in Python offers access to many open-source libraries which can provide functionality for most applications at a broad level, which can then be tailored to suit the needs of the developer. For example, MATLAB is a widely used program in digital signal processing, but using libraries such as NumPy (*17*), SciPy (*18*), and MatPlotLib (*19*), a lot of the features can be implemented in Python freely. Playing audio can be done in many ways. The most straightforward is to create an array with audio data, and use the SoundDevice (*20*) and SoundFile (*21*) libraries to transmit the signals. These libraries are built on the PortAudio (*22*) and provided simple and convenient methods for basic processing. Solutions for the Time of Arrival and Time Difference of Arrival methods are currently in development. The Time of Arrival measurement can be made by generating a short, single channel sinewave signal. This signal is played over the speaker and recorded simultaneously by the microphone. The recorded wave is then cross-correlated with the original wave and using these results the lag position is found. This lag position represents the time delay between the signal being sent and is then multiplied by an approximated speed of sound to give the distance measurement. The Time Difference of Arrival uses a similar approach, however this time, the output signal is split using a 3.5mm splitter. The audio is sent to the speaker where the signal is recorded by the microphone as before. The signal is also sent to an input jack on the soundcard to give a second waveform. The Time of Arrival is calculated for both of the recorded waves. As the signal is sent directly to an input via a wire, we can assume the signal arrives instantly from the output. We can then assume that the time difference between the wire input and the microphone input is more representative of the true distance measurement as it alleviates any propagation delays in the hardware before the output is transmitted. One of the main issues with this however, is that the SoundDevice libraries do not allow inputs from multiple sources on the same stream. Two input streams, which will almost certainly never be exactly in sync, must be created and processed which introduces uncertainty in the measurements and erroneous behaviour. Ongoing work is being done to modify the code which SoundDevice uses to provide a more reliable measurement. Using the RPiTx (*23*) library is it possible to transmit FM signals from the Raspberry Pi. All that is required to attach a wire to the GPIO18 pin to act as an antenna. Early local tests have been able to transmit signals which are received using a standard FM radio. Further investigation in the areas of improving the power of the sent signal is ongoing. Future development looks to modify the systems above to use Ultrasonic signal transmitters and receivers for accurate range measurements. Use a second device as a transmitter, synchronize transmissions and measure distance from receiver. Develop a reliable synchronisation system that ensures the accuracy of the range measurements. With additional devices, locate the receiver in a 1D space between 2 transmitters, and a 2D space between 3 or more transmitters.

```
static public void main(String[] args) {
  try  {
    UIManager.setLookAndFeel(UIManager.getSystemLookAndFeelClassName());
  }
  catch(Exception e) {
    e.printStackTrace();
  }
  new WelcomeApp();
}
```

Bibliography

1. K. Wu, J. Xiao, Y. Yi, D. Chen, X. Luo, L. M. Ni, *IEEE Transactions on Parallel and Distributed Systems* **24**, 1300–1309 (2013).
2. J. Xiao, Z. Zhou, Y. Yi, L. M. Ni, *ACM Computing Surveys (CSUR)* **49**, 25 (2016).
3. M. Minami, H. Morikawa, T. Aoyama, *Electronics and Communications in Japan (Part III: Fundamental Electronic Science)* **90**, 17–26 (2007).
4. H. Piontek, M. Seyffer, J. Kaiser, *Personal and Ubiquitous Computing* **11**, 439–449 (2007).
5. J. Moutinho, D. Freitas, R. E. Araújo, *The Journal of Navigation* **69**, 1024–1040 (2016).
6. L. Segers, J. Tiete, A. Braeken, A. Touhafi.
7. C.-C. Hsiao, P. Huang, presented at the.
8. A. Runge, M. Baunach, R. Kolla, presented at the Indoor Positioning and Indoor Navigation (IPIN).
9. C. Wu, Z. Yang, Y. Liu, W. Xi, *IEEE Transactions on Parallel and Distributed Systems* **24**.
10. M. Ros, J. Boom, G. de Hosson, M. D'Souza, *International Journal of Navigation and Observation* **2012** (2012).
11. H. Huang, J. Zhou, W. Li, J. Zhang, X. Zhang, G. Hou, *IET Networks* **5**.
12. J Xu, M Ma, C. Law, *IET science, measurement & technology* **5**.
13. R. Zhang, F. Hoflinger, L. Reindl.
14. B. Barshan, *Measurement Science and technology* **11**, 45 (2000).
15. R. Pi, *Raspberry Pi* **1**, 1 (2013).
16. J. C. Glover, V. Lazzarini, J. Timoney (2011).
17. T. E. Oliphant, *A guide to NumPy* (Trelgol Publishing USA, 2006), vol. 1.
18. E. Jones, T. Oliphant, P. Peterson (2014).
19. J. D. Hunter, *Computing In Science & Engineering* **9**, 90–95 (2007).
20. M. Geier, *Play and Record Sound with Python*, (`https://python-sounddevice.readthedocs.io/en/0.3.8/index.html`).
21. B. Bechtold, *PySoundFile*, (`http://pysoundfile.readthedocs.io/en/0.9.0/`).
22. R. Bencina, P. Burk, presented at the ICMC.
23. E. Courjaud, *RPiTx*, (`https://github.com/F5OEO/rpitx`).
24. L. Segers, D. Van Bavegem, S. De Winne, A. Braeken, A. Touhafi, K. Steenhaut, *Sensors* **15**, 18641–18665 (2015).
25. J. Dondo, F. Villanueva, D. Garcia, D. Vallejo, C. Glez-Morcillo, J. C. Lopez, *Journal of Network and Computer Applications* **45**, 181–190 (2014).
26. L. Brás, P. Pinho, N. B. Carvaloh, *IET Microwaves, Antennas & Propagation* **7**, 679–685 (2013).
27. Y. Shu, Y. Huang, J. Zhang, P. Coué, P. Cheng, J. Chen, K. G. Shin, *IEEE Transactions on Industrial Electronics* **63**, 2424–2433 (2016).

28. A. Varshavsky, E. De Lara, J. Hightower, A. LaMarca, V. Otsason, *Pervasive and Mobile Computing* **3**, 698–720 (2007).

29. R. Górak, M. Luckner, M. Okulewicz, J. Porter-Sobieraj, P. Wawrzyniak, *Mobile Information Systems* **2016** (2016).

30. J. J. Robles, *AEU-International Journal of Electronics and Communications* **68**, 578–580 (2014).

31. G. Girard, S. Côté, S. Zlatanova, Y. Barette, J. St-Pierre, P. Van Oosterom, *Sensors* **11**, 7606–7624 (2011).

32. L. Li, W. Yang, G. Wang, *AEU-International Journal of Electronics and Communications* **70**, 1105–1113 (2016).

33. A. Ens, L. M. Reindl, T. Janson, C. Schindelhauer, presented at the Signal Processing Conference (EUSIPCO), 2014 Proceedings of the 22nd European, pp. 731–735.

34. D. Lymberopoulos, J. Liu, X. Yang, R. R. Choudhury, S. Sen, V. Handziski, *GetMobile: Mobile Computing and Communications* **18**, 24–31 (2015).

35. G. Hollósi, C. Lukovszki, I. Moldován, S. Plósz, F. Harasztos, *Acta Universitatis Sapientiae, Informatica* **8**, 186–215 (2016).

36. H. Dai, W.-h. Ying, J. Xu, *IET Communications* **10**, 717–723 (2016).

37. G. Chen, L. Wang, *Electronics Letters* **50**, 1248–1250 (2014).

38. I. T. Haque, C. Assi, *IEEE Systems Journal* **9**, 76–85 (2015).

39. F. Sainjeon, S. Gaboury, B. Bouchard, presented at the.

40. L. Kong, G. Bauer, J. Hale, *Concurrency and Computation: Practice and Experience* **27**.

41. K. Mizutani, T. Ito, M. Sugimoto, H. Hashizume, *EURASIP Journal on Advances in Signal Processing* **2011**, 101 (2011).

42. O. Onalaja, M. Adjrad, M. Ghavami, *IET Communications* **8**.

43. F. Perez-Cruz, C.-K. Lin, H. Huang.

44. Z. Wang, L. Zhang, F. Lin, D. Huang, Y. Huang.

45. F Hammer, M Pichler, C Hesch.

46. D. Balcan, A. Sandryhaila, J. Gross, M. Puschel, presented at the Acoustics, Speech and Signal Processing, 2008. ICASSP 2008. IEEE International Conference on, pp. 3537–3540.

47. V. R. V. Remya B Ashish Francis, *International Journal of Innovative Research in Electrical, Electronics, Instrumentation and Control Engineering* **4** (2016).

48. M. Hua, M. Wang, K. W. Yang, K. J. Zou, *IEEE Transactions on Communications* **62**, 4024–4039 (2014).

49. A. A. Nasir, S. Durrani, R. A. Kennedy, *IET communications* **5**, 1028–1037 (2011).

50. K. F. Chen, S. L. Mei, *IEEE Transactions on Instrumentation and Measurement* **59**, 1571–1579 (2010).

51. R. Gusella, S. Zatti, "An election algorithm for a distributed clock synchronization program", tech. rep. (CALIFORNIA UNIV BERKELEY DEPT OF ELECTRICAL ENGINEERING and COMPUTER SCIENCES, 1985).

52. S. Denic, O. Georgiou, U. Spagnolini, *IEEE Communications Letters* (2017).

53. J.-C. Yoo, T. H. Han, *Circuits, Systems, and Signal Processing* **28**, 819–843 (2009).

54. G Baudoin, F Virolleau, O Venard, P Jardin, *ESIEE, Paris* (1996).

55. J. W. Kang, Y. Whang, B. H. Ko, K. S. Kim, *IEEE Transactions on Information Theory* **58**, 438–444 (2012).

56. M. D. Sacchi, T. J. Ulrych, C. J. Walker, *IEEE Transactions on Signal Processing* **46**, 31–38 (1998).

57. M Ferreira, J Bagarić, J. M. Lanza-Gutierrez, S. Priem-Mendes, J. Pereira, J. A. Gomez-Pulido, *International Journal of Distributed Sensor Networks* **11**, 720574 (2015).

58. M. Luise, U. Mengali, M. Moeneclaey, J. G. Proakis, E Strom, W. H. Tranter, *IEEE Journal on Selected Areas in Communications* **19**, 2293–2297 (2001).

59. R. H. Silverman, E. Vinarsky, S. M. Woods, F. L. Lizzi, D. J. Coleman, *Retina* **15**, 37–42 (1995).

60. S. Tripathy, S. Jakonen, *International Journal of Mathematics, Game Theory, and Algebra* **24**, 289 (2015).

61. J. C. Jackson, R. Summan, G. I. Dobie, S. M. Whiteley, S. G. Pierce, G. Hayward, *IEEE transactions on ultrasonics, ferroelectrics, and frequency control* **60**, 343–355 (2013).

62. A.-T. Jorge, *Ingeniería, Investigación y Tecnología* **16**, 383–390 (2015).

63. M. H. AlSharif, M. Saad, M. Siala, T. Ballal, H. Boujemaa, T. Y. Al-Naffouri, presented at the Signal Processing Conference (EUSIPCO), 2017 25th European, pp. 1250–1254.