

Real-time Optimisation for Industrial Internet of Things (IIoT): Overview, Challenges and Opportunities

Long D. Nguyen^{1,*} and Ayse Kortun^{2,†}

¹Duy Tan University, Da Nang 550000, Vietnam and Dong Nai University, Dong Nai 810000, Vietnam

²Queen's University Belfast, Belfast BT7 1NN, UK

Abstract

Industrial Internet-of-Things (IIoT) with massive data transfers and huge numbers of connected devices, in combination with the high demand for greater quality-of-services, signal processing is no longer producing small data sets but rather, very large ones (measured in gigabytes or terabytes), or even higher. This has posed critical challenges in the context of optimisation. Communication scenarios such as online applications come with the need for real-time optimisation. In such scenarios, often under a dynamic environment, a strict real-time deadline is the most important requirement to be met. To this end, embedded convex optimisation, which can be redesigned and updated within a fast time-scale given sufficient computing power, is a candidate to deal with the challenges in real-time optimisation applications. Real-time optimisation is now becoming a reality in signal processing and wireless networks of IIoT. Research into new technologies to meet future demands is receiving urgent attention on a global scale, especially when 5G networks are expected to be in place in 2020. This work addresses the fundamentals, technologies and practically relevant questions related to the many challenges arising from real-time optimisation communications for industrial IoT.

Received on 23 September 2020; accepted on 14 December 2020; published on 16 December 2020

Keywords: Industrial Internet-of-Thing, convex optimization, realtime optimization, cloud computing, 5G network.

Copyright © 2020 Long D. Nguyen *et al.*, licensed to EAI. This is an open access article distributed under the terms of the Creative Commons Attribution license (<http://creativecommons.org/licenses/by/3.0/>), which permits unlimited use, distribution and reproduction in any medium so long as the original work is properly cited.

doi:10.4108/eai.16-12-2020.167654

1. Introduction to Industrial IoT (IIoT)

The Internet-of-things (IoT) technology is considered as crucial in many areas of engineering and wireless communication. A wide range of products, machines, smartphones, and sensors are networked via this technology under the advancements of computing power and network interconnections to offer new wireless capabilities. This leads to many open issues for wireless communication systems and new market opportunities for high technical interoperability [21].

There is no universal definition of the IoT. The IoT represents scenarios where many objects from

personal electronics to industrial machines and sensors are allowed to connect wirelessly to the internet, and exchange and consume data through the network connectivity and computing capability. The purpose of combining IoT technologies and wireless networks is to monitor and control many devices in a large-scale scenario of the IoT in wireless communication. In fact, the recent confluence of several technologies, e.g., ubiquitous connectivity, IP-based networking, data analytics, cloud computing and fog computing, is bringing the IoT closer to wireless communication systems. The implementations of the IoT can utilise various technical connectivity models, each with its own flexibility. There are four general models including Device-to-Device (D2D), Device-to-Cloud (D2C), Device-to-Gateway (D2G), and Back-End Data-Sharing [21]. The transformational potential of the IoT once realised can bring a hyperconnected network. This is an illustration of the general purpose of modern Internet architecture, which provides the concept of

*Long D. Nguyen is with the Institute of Research and Development, Duy Tan University, Da Nang 550000, Vietnam and the Department of Engineering, Dong Nai University, Dong Nai 810000, Vietnam. Email: nguyendinhlong1@duytan.edu.vn

†Ayse Kortun is with the School of Electronics, Electrical Engineering and Computer Science, Queen's University Belfast, Belfast BT7 1NN, UK. Email: akortun@gmail.com

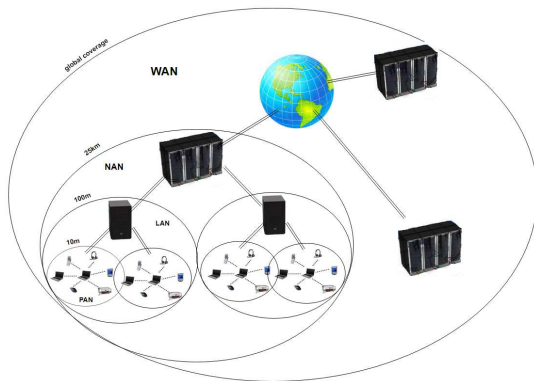


Figure 1. A network range of IoT applications.

unlimited interconnection of applications or services for a massive number of devices.

2. Fundamental of IIoT

2.1. Radio access technologies for IIoT

Given a wide variety of applications in various environments and diverse demands/requirements, it is obvious that no single wireless standard can adequately prevail in an IIoT scenario. As shown in Fig. 1, a common network includes personal area networks (PANs) with a range of about 10m, local area networks (LANs) with 100m coverage, neighbourhood area networks (NANs) with 25km coverage and wide area networks (WANs) for global coverage. Therefore, a difficult task is to exploit good wireless connectivity technology for each of the IIoT applications since numerous standards have been deployed via multiple frequency bands and communication protocols [11].

The potential capabilities of the IIoT have been significantly enhanced by using various advanced technologies in wireless communication protocols, e.g., RFID, Wi-Fi-Direct, LoWPAN, and 5G. Moreover, with the development of sensing, wireless energy harvesting, big data exchanging, and signal processing, the integration of the IIoT with emerging technologies has been also accelerated. However, the interconnection of these technologies based on the IIoT is to bring great benefits on scalability, QoS, energy efficiency, and cost-effectiveness. Privacy and security issues also have to be considered in IIoT applications. The presence of these challenges in the IIoT is still open to study and necessarily calls for novel solutions.

2.2. Cloud computing

Cloud computing refers to the use of computer technology that harnesses the power of many inter-networked processing computers to handle complex database tasks without requiring a computer for each user. This technique creates the current backbone of

communication networks for IIoT applications [23]. The deployment of cloud computing can be attributed to the variation of emerging technologies on the Internet with the huge amounts of the database that need to be mined and organised for users. This leads to the supply of enhanced performance and processing power by the economies of scale in larger data centres.

Cloud Technology. Some several major technologies of cloud computing are investigated as below [14].

- *Application:* is a component stored on remote servers and runs in real time for clients. Access to the application is made through a user interface (UI) that an end user will use the features of cloud computing.
- *Client:* is a UI platform, i.e., a web browser likes Mozilla Firefox, Google Chrome or Microsoft Internet Explorer.
- *Infrastructure:* consists of computer hardware and the buildings containing computer hardware. The server environment runs virtualisation technology, meaning that the server framework inside is not depending on the number of specific machines. Hence, it entirely runs software by harnessing processing power of multiple machines to reap more benefits of processing ability from each customer.
- *Platform:* is the deployment approach of applications via Platform as a Service (PaaS). This would include a web application framework in the form of an open source web application with web design programming.
- *Service:* refers to users' experiences which are reaped from their cloud computing. There are a variety of services on the Internet that support users. These services would require a lot of database storage and processing power to perform tasks.
- *Storage:* is one of the biggest features of cloud computing that can be expensive. It is important that cloud vendors should assure the safety of customer's data by providing the agreements of service and letting with customers.
- *Processing power:* is expanded immensely with cloud computing. Companies are able to use this capacity for testing out new markets and trying out new applications over the web. Cloud-based vendors provide networked users who are connected to the Internet with the processing power of cloud computing. The servers should be utilised efficiently to avoid wasting valuable processing resources.

Cloud Management. Cloud computing is more flexible in terms of technology resources, e.g., the idea of the server rental model. In particular, with cloud computing, customers can decide how much storage space is used, and how much processing power is required so that they can choose when to update an application enterprise-wide all in real-time. Scalability in cloud computing allows each user to quickly go from small to large processes while being able to guarantee the processing of heavy amounts of data at a specified time without requiring large servers. There are still a lot of issues with cloud computing in IoT applications since any innovations always come with their problems. Below are potential challenges in cloud computing that call for novel solutions [14].

- *Dependability:* To receive resources from cloud centres, users have to prove their credibility by showing an image projection through an electric utility model. This level of energy should be a dependable resource to allow cloud computing to provide certain level of service.
- *Security:* Securing data has always been a high priority in IT cloud because of massive amount of data stored at cloud centres. Moreover, with a wide range of technologies and various applications, there are risks in communicating and accessing services and applications through the Internet, and thus, thin clients could be interfered or hacked when the cloud organiser fails to keep track of resources, or when users are careless with data transfer and the services they use.
- *Supporting services:* The vendors are required to support their clients with novel solutions which could be different from traditional IT solutions. They should ensure stable service levels for many demands of their customers - being able to provide 24-hour on-call duty is a good illustration for this. The reliability of connectivity for all clients and the guarantee of data security are important in order to improve their services.
- *Capacity:* With the IT tasks organised in the cloud, the system must be operated in the way that capacity resources are effectively utilised for the benefit of both cloud centres and clients. Resource management aims to achieve a high level of performance and functionality to keep high quality of services. Resource or capacity management would be important to resource planning in the implementation of cloud computing.

2.3. Fog (Edge) computing

It is envisioned that an exceeding volume and variety of data in IoT has been generated and continuously exploded. As a result, the operating time for data processing in the cloud has significantly increased. To tackle this issue, a new technology for IoT applications and services, known as fog computing, inherits and extends the cloud computing paradigm for supporting large-scale systems, e.g., a large number of devices/nodes [1, 3]. Fog computing has been promising many benefits such as low latency, wide-spread geographical distribution, high mobility and location awareness, predominant role of wireless access, strong presence of streaming and real-time applications, and heterogeneity.

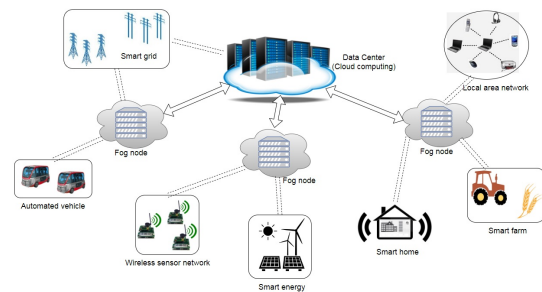


Figure 2. Applications in IoTs with fog computing.

IIoT Fog Computing. For the communication of systems, fog computing nodes operate as small cells, smart gateways, routers, or dedicated fog devices, which involve application components running both in the cloud and/or end devices between the network edge and the cloud as shown in Fig. 2. There are many benefits of fog computing in supporting the mobility, computing resources, communication protocols, interface heterogeneity, and distributed data analytics to enable low-latency applications within a wide and dense geographical distribution. The main characteristics of fog computing are as follows:

- *Reduction in network traffic:* A huge volume of data that is generated, sent, and received from the billions of smart devices, introduces an opportunity for putting computing capabilities closer to devices, rather than having all data sent over to data centres at high latency. Fog computing provides a platform for gathering and analysing data close to the network edge through local data views. This drastically decreases the network traffic being sent to and from the cloud.
- *Supporting mobility and geographical distribution:* It is obvious that fog computing applications can communicate directly with end devices, and thus support mobility techniques. In sharp contrast to

the centralised cloud, the targeted services and applications in fog paradigm can be deployed in a distributed manner. As such, fog computing will play an active role in delivering high-quality real-time services, for example, to support smart transportation, through proxies and access points positioned.

- *Interoperability and federation in large-scale heterogeneous networks:* Fog computing can support many smart systems in monitoring the environment surrounding a very large number of devices/nodes by relying on distributed computing and storage resources in inherently distributed systems.
- *Suitability for IoT tasks:* Most service requests from a large number of smart devices pertain to their surroundings. These requests depend on the nature of the typical requests in each application, and should be processed in fog rather than cloud infrastructure. By bringing the processing closer to the network edge, fog computing makes the communication distance closer to the physical distance as well as reduces traffic loading.
- *Low-latency requirement and real-time applications:* Many mission-critical applications with real-time data processing, e.g., cloud robotics, video streaming, and automatic vehicles, are exploited by using fog computing. In fact, a control system based on the data collected and feedback, which runs on the cloud, may make the sense-process-actuate loop slow or unavailable as a result of communication failures. This issue can be handled by applying fog computing to perform the processing required for the control system very close to the devices and thus making real-time response possible.
- *Scalability:* The lack of finite resources in cloud computing prompts a bottleneck to occur when all the raw data generated by networked devices is continually sent to the cloud. Fog computing aims at processing incoming data closer to the data source itself, it reduces the burden of data processing on the cloud, thus addressing the scalability issues arising out of the increasing number of endpoints.

3. Challenges and Opportunities of Realtime Deployment IIoT networks

3.1. Energy efficiency in IIoT

Resource allocation is one significant aspect in any systems. In IIoT applying 5G wireless networks [15], the new systems should be more dynamic and

intelligent to simultaneously satisfy various network demands, by adopting new wireless technologies and approaches. To this end, resource allocation is faced with major challenges such as interference alignment, security attacks, or green communication. Consequently, together with the enhancement of spectral efficiency performance, an emerging trend of 5G wireless networks is to approach green communication via energy efficiency (EE), whose most significant challenge due to its belonging to the fractional programming in the optimisation field, i.e., nonconvex programming. This leaves many challenges in the efforts to improve network EE performance. The new works need to investigate the optimisation paradigm applied to tackling the EE performance in IIoTs [17].

3.2. Quality-of-Service for a large number of users

To serve many users, system's resource allocation must schedule its service because a large number of users are served at any given time. As such, it is not known if IIoT is able to deliver high quality-of-service (QoS) to many users simultaneously, no matter whether they are node-centre users with better channel conditions or node-edge users with poorer channel conditions [19].

In large-scale IIoT networks, node-edge users suffer not only poor channel conditions but also inter-tier interference, which cannot be rejected or mitigated by the current technologies. As such, the problem of providing high QoS for users in IIoT is much more technically challenging than ever.

3.3. Realtime computing and realtime optimisation

Realtime computing. A system can respond to real-time computing if its ability of processing tasks satisfies real-time bound responses for performing necessary computations. Physical components are controlled by functions of real-time computing framework and then they process tasks at periodic intervals and respond by sending signals to the main system controller and other components [12]. A time bound is required for all types of delivered responses. In critical missions, a computation schedule is essential to support the required time bounds for each response in a sequence of events. In theory, it may be unable to meet timing constraints within the limited processing capacity available. When the time bound of a response or the timing constraint is not met, there may be a catastrophic system failure. Therefore, any real-time program has to be considered and designed carefully on a real system with finite resources. Their computing platform should be adapted to a time-varying environment, interactive with predicted time-dependent behaviours and executed with time-bound responses.

When the number of tasks to perform with large processing loads increases, multiple processors as a potential technology for multiple tasks in real-time processing are required. On the other hand, a distributed computing framework is tremendously required for large processing loads in applications with multi-tasks [18]. Hence, the prediction and design of completion times are critical to observing real-time processing with time-varying task communication.

Realtime optimisation. Each optimisation application is a balance and trade-off embedded in the analysis and design by developers. However, there is a huge gap between the optimisation theory and optimisation implementation in their applications. In addition, there is a lack of research to link the theory and implementation of optimisation. The decision of optimisation application in reality is a complex decision making process which involves environment models, characteristics of the problem (variables, objectives), appropriate methods and trade-offs for an expected solution [6, 7, 17].

In the context of optimisation theory, a mathematical model optimisation problem from the real world can be transformed into a practical online optimisation problem. Then this problem can be programmed in embedded systems with online solutions. The online solutions found should be appropriate to the problem and satisfies the accuracy of performance. Meanwhile, the strict time deadline should be adhered to in real-time implementation. As a result, there are many challenges in real-time optimisation in the sense that "faster is better" [13]. In engineering, fast optimisation is used to calibrate some weights to improve the transmission strategy, optimise network performance or other objectives. However, practical optimisation problems in IIoT are frequently non-linear programming, large-scale problems with massive amounts of data (big data analytics), multi-objective and/or high-complex problems, stochastic problems, sparsity or low-rank problems (channel assignments) and dynamic programming (hybrid resource allocation problems) [16, 20].

3.4. Optimisation problem complexity

The computational complexity of a convex optimisation problem is calculated as

$$\mathcal{O}(N_{var}^2 N_{con}^{2.5} + N_{con}^{3.5}) \quad (1)$$

where N_{var} is the number of scalar variables and N_{con} is the number of constraints. A solution of the problem under algorithm programs with rational input is said to run in polynomial time. If a function is computed by some polynomial-time algorithm, it is said to be computable in polynomial time. Sometime,

the complexity of the problems is much more than that for (1). It leaves significant challenges for solving problems with appropriate solutions, especially for realtime optimisation solutions.

4. Realtime operating system and Embedded Realtime Optimisation

4.1. Realtime operating system

Concepts of real-time systems. One widely adopted definition of real-time systems describes them as real systems that are capable of meeting the requirements of real-world problems within a limited processing time. In other words, the implementation of real-time applications will have to satisfy timing constraints, e.g., solving deadlines and other timing requirements. This leads to the development of real-time programs in both software and hardware systems based on timing properties and characteristics. The most important features of time-critical systems are effective strategies, robust algorithms, parallelism techniques and final processing speed. The question is, how processing speed corresponds to timing demands and program statements. A carefully analysed problem statement and resource allocation of the system are tremendously needed to approach timing demands. In general, scheduling policies are useful in allocating the system's essential resources to embedded programming. By calculating the precision of development stages, schedulers can be determined. Then, specifications, verification and timing analysis will be collaboratively performed to provide an accurate prediction of the timing in real-time systems.

Real-time embedded systems. The development of embedded systems with integrated micro-controllers has been accelerated in order to replace conventional electronic control systems. Embedded real-time systems are an important prerequisite for a wide range of applications such as those in smart industry, manufacturing automation, smart city, and wireless communication. A real-time embedded system plays the role of an intelligent part in a well-designed large system, which consists of mechanical component, embedded computational controller, and human-machine interface. In fact, the variety of real-time applications of embedded systems and the size of real-time system markets have significantly expanded in the last few years, and are expected to continue to grow in the era of IIoT and 5G networks.

Intelligent decisions and predictable services are required of real-time application tasks in a real-time operating system. In real deployment, we focus on task management, inter-process communication, time management, and error detection [10] in real-time embedded systems. Task management refers to the provision

of a dynamic environment for the initialisation, execution, and termination of application tasks within a host. Inter-process communication is to ensure information exchange among concurrently executed tasks in order to achieve progress. Flexible time management services are provided by the operating system to simplify or reduce the computational operation of the application software in real-time applications. Error detection must be introduced in a well-designed, real-time system for monitoring task execution time and unexpected interruptions, and detecting fail-silent nodes.

4.2. Realtime Embedded Optimisation for IIoT

In this section, we focus on convex optimisation in real-time embedded systems. Real-time embedded optimisation is used in many areas such as signal processing, automatic control, real-time estimation, real-time resource allocation and fast decision making in trading. The considerations of "*real-time*" and "*embedded*" mean that optimisation algorithms execute much faster than a typical or generic method without any human intervention or action in the loop [13]. The embedded system using real-time optimisation algorithms is rapidly updated and fully automated with newly arriving data or changing conditions in time intervals that are measured in microseconds, milliseconds or a few seconds for small-, medium- or large-size problems, respectively. This approach will be beneficial in solving optimisation problems with realtime updated data sets.

Embedded optimisation in MATLAB. Two widely used optimisation tools compatible with Matlab are CVX [9]. CVX is a Matlab-based tool that turns Matlab into a modelling language. Therein, the constraints and objectives of an optimisation problem is determined by standard Matlab expressions. In default mode, CVX provides a disciplined convex programming to approach convex optimisation. Convex functions and constraints are built up from the convex rules set in a base library. These rules address constraints and objectives following a canonical form after being automatically transformed and solved. It is not difficult to start CVX since it includes a growing library of examples from the book Convex optimisation [2] and from plenty of applications to help users.

4.3. Embedded optimisation in PYTHON

CVXPY [4] is a modelling language embedded in Python for convex optimisation problems. Python is a high-level programming language for many general programmings with various purposes. Python programming emphasises code readability, provides constructs for clearly programming on both small- and large-scale problems. CVXPY can express optimisation

problems in efficient ways following math models rather than restricted standard forms by solvers like CVX or YALMIP. In fact, CVXPY solves problems in simple ways and parallelism constructs and can be extended for non-convex optimisation in future. Similar to CVX, CVXPY is also easy to learn and use via many examples available.

4.4. Embedded optimisation in R

In R package, CVXR [8] is an integrated tool providing the object-oriented modelling language for convex optimisation, similar to CVX and CVXPY. Without restrictions to a standard form of solvers, CVXR users can formulate convex optimisation problems in a natural mathematical model by combining constants, variables, and parameters using a mathematical properties library. Once having verified the problem's convexity by disciplined convex programming, CVXR transforms the problem into a standard conic form and passes it to one of solvers such as ECOS or SCS solvers.

4.5. Embedded optimisation in JULIA

Solving optimisation in the high-level Julia programming is called JuliaOPT. It includes Convex.jl [22] and JuMP.jl [5] packages.

Convex.jl (CVX.jl) is a Julia package for solving linear programs, mixed-integer linear programs, and DCP-compliant convex programs under disciplined convex programming, through the MathProgBase interface. Convex.jl transforms the problem (objective functions and constraints) into a cone program and passes the problem to one of the solvers. JuMP is a domain-specific Julia modelling language for embedded optimisation in mathematics. JuMP supports a number of open-source solvers for a variety of problem classes, e.g., linear programming, (mixed) integer programming, second-order conic programming, semidefinite programming, and nonlinear programming with many popular optimisation solvers including Cbc, Clp, ECOS, GLPK, Gurobi, MOSEK, NLOpt, SCS. JuMP can be easily embedded in complex workflows including simulations and web servers.

5. Conclusions and Future works

We have introduced the practical optimisation related to a diversity applications of the IIoT systems based on the development of optimisation platform in real-time scenarios. Industrial IoT have been emerging to become a major trend in the next global innovation generation. With flexible deployment and diversity applications, IIoT can be more efficient and inexpensive for implementation, which is exploited in many applications. With a wide variety of platforms, IIoTs can operate in wide scenarios, i.e., ground, air, water,

or underground. This work opens a novel trend of optimisation for the modern industry generation, IIoT. As a result, there are many issues and future works to study and necessarily call for novel solutions in the realtime optimisation for IIoTs.

Acknowledgement

This work was supported in part by the Global Challenges Research Fund under the DfE-GCRF 2020-2021 scheme between Queen's University Belfast, UK and Dong Nai University, Vietnam.

References

- [1] Bonomi, F., Milito, R., Zhu, J. and Addepalli, S. [2012], Fog computing and its role in the internet of things, in 'Proceedings of the first edition of the MCC workshop on Mobile cloud computing', ACM, pp. 13–16.
- [2] Boyd, S. and Vandenberghe, L. [2004], *Convex optimization*, Cambridge university press.
- [3] Buyya, R. and Dastjerdi, A. V. [2016], *Internet of Things: Principles and paradigms*, Elsevier.
- [4] Diamond, S. and Boyd, S. [2016], 'CVXPY: A Python-embedded modeling language for convex optimization', *J. Machine Learning Research* 17(83), 1–5.
- [5] Dunning, I., Huchette, J. and Lubin, M. [2017], 'JuMP: A modeling language for mathematical optimization', *SIAM Review* 59(2), 295–320.
- [6] Duong, T. Q., Nguyen, L. D. and Nguyen, L. K. [2019], Practical optimisation of path planning and completion time of data collection for UAV-enabled disaster communications, in '2019 15th International Wireless Communications & Mobile Computing Conference (IWCMC)', IEEE, pp. 372–377.
- [7] Duong, T. Q., Nguyen, L. D., Tuan, H. D. and Hanzo, L. [2019], Learning-aided realtime performance optimisation of cognitive UAV-assisted disaster communication, in '2019 IEEE Global Communications Conference (GLOBECOM)', IEEE, pp. 1–6.
- [8] Fu, A., Narasimhan, B. and Boyd, S. [2017], 'CVXR: An R package for disciplined convex optimization', *arXiv preprint arXiv:1711.07582*.
- [9] Grant, M. and Boyd, S. [2014], 'CVX: MATLAB software for disciplined convex programming, version 2.1', <http://cvxr.com/cvx>.
- [10] Kopetz, H. [2011], *Real-time systems: design principles for distributed embedded applications*, Springer Science & Business Media.
- [11] Lethaby, N. [2017], 'Wireless connectivity for the internet of things: One size does not fit all', *Texas Instruments*.
- [12] Mattai, J. and Joseph, M. [1995], *Real-Time Systems: specification, verification, and analysis*, Prentice Hall PTR.
- [13] Mattingley, J. and Boyd, S. [2010], 'Real-time convex optimization in signal processing', *IEEE Signal Process. Mag.* 27(3), 50–61.
- [14] Menken, I. [2008], *Cloud Computing-The Complete Cornerstone Guide to Cloud Computing Best Practices Concepts, Terms, and Techniques for Successfully Planning, Implementing... Enterprise IT Cloud Computing Technology*, Emereo Pty Ltd.
- [15] Nguyen, L. D. [2018], 'Resource allocation for energy efficiency in 5G wireless networks', *EAI Endorsed Transactions on Industrial Networks and Intelligent Systems* 5(14).
- [16] Nguyen, L. D., Duong, T. Q., Ngo, H. Q. and Tourki, K. [2017], 'Energy efficiency in cell-free massive MIMO with zero-forcing precoding design', *IEEE Commun. Lett.* 21(8), 1871–1874.
- [17] Nguyen, L. D., Kortun, A. and Duong, T. Q. [2018], 'An introduction of real-time embedded optimisation programming for UAV systems under disaster communication', *EAI Endorsed Transactions on Industrial Networks and Intelligent Systems* 5(17).
- [18] Nguyen, L. D., Nguyen, K. K., Kortun, A. and Duong, T. Q. [2019], Real-time deployment and resource allocation for distributed UAV systems in disaster relief, in '2019 IEEE 20th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)', IEEE, pp. 1–5.
- [19] Nguyen, L. D., Tuan, H. D., Duong, T. Q. and Poor, H. V. [2019], 'Multi-user regularized zero-forcing beamforming', *IEEE Trans. Signal Process.* 67(11), 2839–2853.
- [20] Nguyen, M., Nguyen, L. D., Duong, T. Q. and Tuan, H. D. [2018], 'Real-time optimal resource allocation for embedded UAV communication systems', *IEEE Wireless Commun. Lett.* pp. 1–1.
- [21] Rose, K., Eldridge, S. and Chapin, L. [2015], 'The internet of things: An overview—understanding the issues and challenges of a more connected world. the internet society (isoc)'.
- [22] Udell, M., Mohan, K., Zeng, D., Hong, J., Diamond, S. and Boyd, S. [2014], 'Convex optimization in Julia', *SC14 Workshop on High Performance Technical Computing in Dynamic Languages*.
- [23] Velte, A. T., Velte, T. J., Elsenpeter, R. C. and Elsenpeter, R. C. [2010], *Cloud computing: a practical approach*, McGraw-Hill New York.