Extended Abstract:

Safety-oriented flexible design of Autonomous Mobile Robot systems

Nitin Desai and Sasikumar Punnekkat
Mälardalen University
Västerås, Sweden

***Abstract*—Current industrial automation applications particularly within the smart manufacturing domain require mobility, flexibility of deployment, and scalability. In addition to these, it is important to mitigate the risk of safety hazards. In this paper we discuss a flexible, granular, and software-based system design framework that aims to improve both security and safety of an autonomous mobile robot (AMR) based industrial automation systems. The decentralized control architecture ensures that safety-critical functions are distributed throughout the network. To this end, we first define system-level security/safety requirements and identify security procedures required to satisfy safety-critical functions such as emergency-stop (E-Stop). We then explain the benefits provided by the proposed system architecture vis-á-vis its resilience towards potential safety and security hazards.**

***Index Terms*—Safety, Fog computing, mobile robots, industrial automation**

I. Introduction

Ensuring safety is a multi-dimensional problem that depends not just on the correct functioning of the system but also in the way a system responds to changes in the environment, unanticipated inputs, changes to its functioning after hardware and/or software upgrades, amongst other things. Safety assurance is a guarantee that the system is acceptably safe to its users (humans or other systems) at all times, under all operating conditions. Such a guarantee (Safety certificate) is based on provision of a detailed safety case including safety arguments and supporting evidence (test results, proofs of verification, adherence to standards etc.). The evidences are gathered from conducting numerous trials and subjecting the system to different input combinations to ensure that its behaviour is along expected lines. Thus, assuring system safety is a precise and highly complex activity that demands huge efforts as well as a mindset that can anticipate failure modes and understand its effects.

Safety requirements vary with each application domain. For instance, mission-critical control and safety requirements of time sensitive applications in the oil and gas industry need deterministic communications so that data packets reach the destination within the cycle time of the process typically 100 ms or less [1]. However, in advanced driver assistance systems (ADAS) in automotive applications should be in the range of a few ms, typically around 3 to 4 ms [2].

In order to guarantee safe operation, safety standards like IEC 61508 [3] define measures and techniques to be applied to the development and life cycle of control systems that reduce the residual process risk to a tolerable level [4]. An important requirement of safety critical systems is the ability to guarantee safe operation when any configuration changes are made to the system while it is running. Therefore, in this paper, we develop a novel hybrid software architecture that can adapt to changes in the system configuration at run-time. The hybrid aspect combines the best of centralised and decentralised design paradigms geared towards enhancing the ability of the robotic system to respond to such run-time changes.

The rest of this paper is organized as follows. Section II discusses the current state-of-the-art in automation software design. In Section III, we describe a concrete use-case. Section IV deals with the safety requirements in a typical factory automation setting with specific focus on the networking dimension. In section V, we describe limitations of present system designs and key desirable architectural considerations followed by the proposed hybrid architecture in section VI. Section VII concludes this extended abstract with our intentions regarding future work both for extending to the full paper and beyond.

II. Related works

Ahmed *et al* [5] provide a comprehensive mapping study of the recent advances in robotic software architectures. The authors categorize current robotic state-of-the-art based on generic classifications robotic evolution, operations and development. Broadly, we have objected oriented (OO), component-based (CB) and service driven (SD) software design approaches. Remy *et al* [6] discusses service-oriented architectures, while Yung *et al* [7] discusses an interesting domain of medical robotics. The paper describes a component-based architecture that seamlessly bridges the gap between real-time robot control and a distributed, integrated system.

The AMR-based wireless industrial automation systems can be subject to a wide variety of security attacks. Each of these security vulnerabilities can affect the system safety in different ways. For instance, promising technologies such as update over-the-air (OTA) for IoT devices can potentially compromise security by installing firmware from a malicious source [8].

In the worst case, this can endanger the safety by disabling it altogether or by running amok in the factory causing collisions. Other ways to induce safety hazards is by jamming the safety-critical data packets by using a high power transmitter. For a comprehensive list of security attacks that can be made on IoT devices, readers are referred to [9].

To the best of our knowledge, related literature do not discuss any fine-grained hybrid design paradigm with focus on safety and security whilst supporting flexibility.

III. Brief overview of use-case

We consider a factory automation scenario consisting of autonomous mobile robots that are programmed to complete specific missions towards the accomplishment of system-level goals. The AMR nodes are controlled by a centralized controller which takes care of a subset of functions such as path planning, scheduling and policy-based decision making. The nodes communicate over standard industrial wireless communication protocols such as wirelessHART. Although each manufacturer is free to choose his/her own architecture, there are certain commonalities w.r.t. the design principles - particularly a fixed/static architecture - which gives very little or almost no room to adapt to on-demand and run-time requirements. By means of this use-case, we aim to explore possibilities that go beyond the current static architectures with added focus on safety and security.

The ultimate objective of the proposed architecture is to provide resilience by ensuring safety functions that are distributed among the nodes themselves and not confined to a centralized controller. By doing so, the single-point dependence on any device such as a controller can be avoided. This reduces the probability of system failure even when a key component malfunctions. The underlying principle is based on a holography wherein complete information is retained even when major parts of a hologram are removed [10].

IV. Safety requirements in Factory Automation

Even though every factory automation application has its own specific set of safety requirements, there are few common ones. The primary aim is to ensure safe operations and one focus area is collision avoidance between robots and between robots and the infrastructure(including human operators). Secondly, safe state definitions in the event of a collision is needed to ensure how the system must react in the event of a collision. In our use-case, an emergency-stop (E-stop) mechanism is necessary in the likelihood of a collision wherein the colliding robots stop immediately and return to a safe state, to minimize damage to the system and its users. Considering the dynamic and high interference environment in which these robots operate, one of the challenges is ensuring that the E-stop mechanism is always available.

A wireless factory automation system has to receive its inputs within a well-bounded, deterministic time interval and with a guaranteed latency due to the typical real-time requirements of such systems and the associated safety implications. If for some reason the input is not received, the emergency-stop is triggered. At the same time, in an interference rich environment such as the factory environment, there is a strong possibility that in general, a transmitted signal does not reach the receiver on time which can initiate the E-stop mechanism unnecessarily. Such false positive scenarios must also be tackled appropriately.

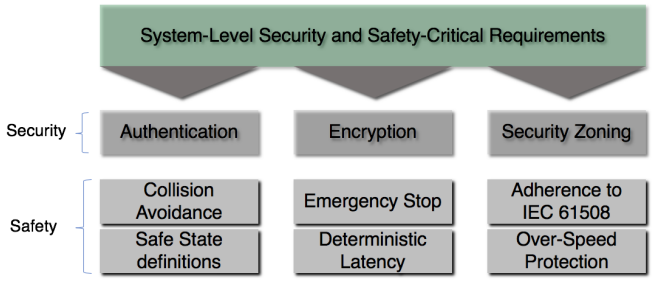

Fig. 1. A subset of security and safety requirements

Fig. 1 provides a brief overview of the safety and security requirements from a system-level perspective, some of which have been elucidated above. The security requirements broadly can be categorized under authentication, encryption and security zoning. The concept of security zoning is described in IEC 62443 [11]. In short, security zoning is a method to segment a system into zones with different security levels. A security level is complied with by implementing a combination of security countermeasures [12]. In a wireless system, since the transmitted signals are available to any receiver near the vicinity (based on the physical characteristics of the signal such as Tx power, range and obstacles etc.), there exists a strong possibility to extract this signal for malicious purposes. The signal can very well be a safety-critical data packet and access to this can potentially impact safety in a serious way [13].

There are specific attacks that can be targeted at each layer of the wireless interface such as PHY, MAC, Network layer etc. [14]. While conventional logic aims to strengthen security loopholes, [14] explains how even a sufficiently secure system can make way for an intrusion by triggering its emergency response mechanism (coined ”DISASTER” by the authors). Once the emergency mechanisms are in place, the system is in fact in a weakened state from a security perspective. This can be exploited by a malicious user to gain entry [15]. This is an interesting and different view of security, which usually goes unchecked since the conventional view of security is to ensure that an intruder cannot gain access.

It is almost certain that security can never be foolproof. And consequently, safety hazards arising out of security vulnerabilities can never be prevented totally. Therefore, the onus is on system designers, to develop architectures that are resilient to security attacks in a way that does not endanger safety especially for, but not limited to, critical applications in industrial automation.

V. Architectural considerations

In this section, we focus on some of the emerging trends of contemporary systems and describe some of the key systems level architectural considerations.

A. Flexibility and granularity

Contemporary AMR node architectures typically have a fixed and rigid set of modules (software) for each function. There has been no way to distribute system capabilities between the nodes. The underlying system design is basically static and does not vary with the specific demands that the operational environment imposes. For example, a fixed communication protocol is employed to connect the nodes. From a safety and reliability perspective, this may not be very conducive, due to the over-dependence on a single frequency band which can face interference from other nodes or devices. Therefore, there is a need for more run-time flexibility to cater to the dynamic requirements of the system. This is one of the limitations which must be considered while designing future AMR nodes and factory automation systems based on AMRs.

B. Support for evolutionary system design

Earlier robots (and even some current ones) performed specific tasks such as lifting objects between two fixed points and navigation guided by laser tapes on the floor. The firmware with instructions about path planning and other parameters such as speed and distance would be burned on the ROM by an operator. The machines would then blindly execute the code. In the real sense, there was very little autonomy.

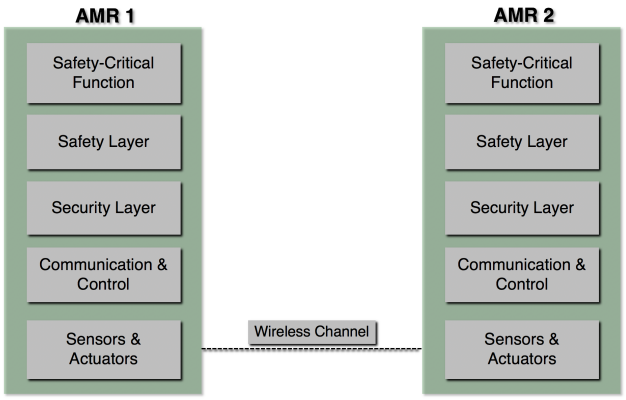

Fig. 2. High level AMR-based system design

Figure 2 depicts a system design that envisages separate modules for safety and security within the AMR nodes. The wireless chips connecting the AMR nodes are usually manufactured by a third party vendor and simply embedded into the AMRs. Hence, although the system comprised of the robots themselves have undergone safety certification, it is impossible to include the wireless channel into the safety process. The vagaries of the medium preclude such a guarantee due to the uncertainties of the environmental and operational conditions under which the system is expected to function.

C. Ensuring safety and security in evolving flexible systems

One of the main challenges in designing a system with a set of requirements that evolve is that safety and security certification cannot be provided easily. This is due to the fact that authorities that certify a system as safe or secure do so with full knowledge of the parameters that the system will operate. In a system that is flexible, there is no way as yet, to know how the system will react to changes in the operating environment.

In the sub-sections below, we discuss two trends in current AMR systems that can limit its growth or evolution into a flexible, software-defined version.

1) Centralized Cloud-based control: Current automation systems especially process automation heavily rely on cloud-based control wherein a local controller within the factory premises is connected to a cloud server via the Internet. Connectivity to the cloud helps with data storage and analytics to guide future policies w.r.t. process decisions. Furthermore, safety-critical as well as security functions from controller-to-nodes as well as between nodes themselves are processed in the cloud.

Whereas such a setup would be sufficient in a small factory having tens of nodes, it most certainly cannot cope well with ***scalability***. As factories expand in size and tasks to be handled grow more complex, robots with increasing intelligence needs to be developed and deployed in larger numbers. A centralized controller coordinates a set of robots within a section of the factory or the entire factory itself. For example, there are 15,000 Kiva robots spread across the 10 warehouses in the Amazon's logistics network [16]. Higher node densities within a confined space change the dynamics of the system in a non-trivial manner [17]. Latency increases due to multiple hops between the farthest node and the controller. Interference from other nodes in the vicinity can cause service outages and packet losses especially when they utilize the same frequency band. This causes performance issues which cannot be tolerated in a safety-critical application which demand low latency and deterministic data access. Although one solution is to have more powerful controllers connected to multi-/many-core cloud servers, such a solution may not viable from a cost perspective both in terms of infrastructure as well as operational costs of cloud services [18].

2) Dependence on a single technology: A central feature of current autonomous mobile robot systems in industrial automation is an over-dependence on a single and fixed mode of operation. Let us consider a simple controller-client node system as detailed above. The controller communicates with the client using a secure WiFi connection. The client has an MCU with sensors and actuators deployed as peripherals to communicate with the controller. Various control and navigation algorithms are embedded within the controller and the client to perform closed-loop feedback and course correction as the client node traverses the factory floor.

Even though scheduling, path planning, and control is done in a flexible manner, there is a fundamental rigidity and inflexibility in the underlying system design. A single

wireless protocol is employed for communication between the controller and the client. The processing and storage capabilities are again fixed. This cannot fulfill the needs of the future automation scenarios where performance and scalability will be the key drivers to stay competitive. The benefits offered by a technology will be judged by the extent to which it can maximize performance without affecting system safety and security, amongst other critical factors. In addition, there is a growing trend towards on-demand, need-based, and flexible services at the right place and at the right time.

This motivates the need for a granular, flexible, and scalable system architecture that can maximize utilization of capabilities to match up to the specific requirements of the application.

VI. Proposed hybrid Architecture

We now present our hybrid design approach which can best be described as a predominantly decentralized architecture with an on-demand and need-based centralization capability.

Current AMRs in factory automation follow a centralized architecture with software agents within the controller and the robots to coordinate tasks. However, the flexibility brought about by the cooperation is only at the data exchange level and does not consider the capabilities of the controller and the nodes in terms of processing, control, storage, and available bandwidth, i.e., matching the available resources to the requirements imposed by the application.

A. Our Approach

The system architecture we propose is a hybrid between centralized and de-centralized. The controller wirelessly sends policy updates and is mostly used in long term decision making in collaboration with the central server. All major functions that were previously handled by the controller such as path planning, localization-based collision avoidance, safety and security functions will now be distributed within the nodes. Hence, the term decentralized.

Figure 3 shows a simplified diagram of the proposed architecture. The AMR nodes have modules as shown by various component blocks such as path planner, communication protocols, localization and processing. Each of these components can be flexible and the cognitive engine can decide which configuration of the software modules to use at run-time based on the system requirements. The bi-directional arrows signify the seamless flow of information between AMR nodes that is controlled by the cognitive engine within each node.

To enable such a shift, the AMR nodes must have a flexible, software-defined framework. A cognitive engine that can make decisions regarding the flow of nodal capabilities seamlessly. To the best of our knowledge, such a system has not been developed in factory automation as yet.

One of the methods to implement such a flexible, and granular architecture is Software-defined Radios (SDRs). [19] provides a good overview on the operational principles of SDR although the focus is towards security aspects. SDRs are becoming increasingly popular with not just academia but also in industry as it facilitates rapid prototyping as well as implementing multiple communication standards on a single device through software such as the GNU Radio Companion (GRC) [20]. Powerful FPGAs within the SDR can boost performance when it is needed at run-time.

B. Enabling technologies

The hybrid architecture takes the support of multiple technologies since all these can be implemented in the Software-Defined Radio (SDR). RF bands for improved reliability in data transmissions - one of the essential safety requirements for E-stop which needs to be transmitted with highest guarantees and within a deterministic time interval *regardless of the interference*.

1) Millimeter-wave high data rates: 60GHz millimeter-wave can perform high-speed, low latency communications between nodes. Although the path loss in the Non-Line-of-Sight (NLoS) is high, the reliability in the LoS direction is extremely good and provides almost no interference to surrounding nodes. Bandwidths of up to 5GHz are available while in Europe, bandwidths can reach up to 9GHz [21]. As a solution to ensure safety-critical information is received reliably, the data could be transmitted on millimeter-wave 60 GHz frequency bands when there is LoS between nodes to ensure guaranteed reception without errors. However, when there is no direct LoS path (due to obstacles or other nodes in between), a lower band such as ISM 5GHz could be used, since the low frequency bands have longer ranges and lower propagation losses.

2) FPGA-based processing: The FPGA in the Universal Software Radio Peripheral(USRP) can be customized for processing computationally intensive applications such as data encryption and scheduling between nodes. A singular advantage with such a run-time re-configurable system is that computational tasks can be shared between nodes. We shall see later how this can benefit our system design approach.

3) Mobile edge computing & Fog computing: Off late there has been considerable advances and research focus on mobile edge computing and fog computing paradigms [22]. The goal is to reduce latency and processing time by designating tasks to an edge device that is closer to the sensors and actuators, so that the cloud is not burdened with all tasks. Although there could be sufficient storage and processing capacities in the cloud, the data volume that is transferred to the cloud is going to be a challenge given the scalability requirements of a typical industrial/process automation application. Though often used interchangeably there are certain subtle differences between these two paradigms. From our perspective both of them provide more predictable platforms than the cloud to relocate safety relevant tasks.

C. Implications for safety and security

We now take two specific functions, E-stop and security zoning, and describe how the proposed architecture aims to improve the reliability of the system.

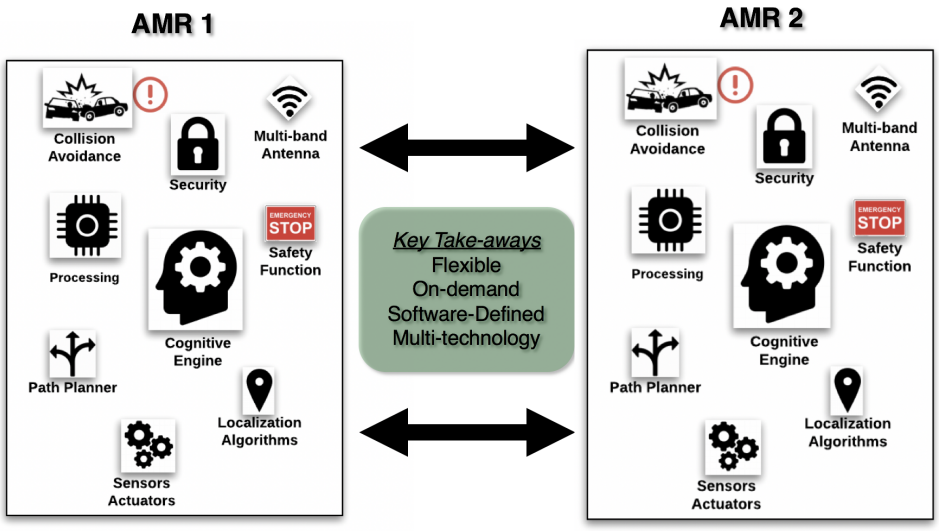

Fig. 3. System design approach using flexible software components

a) ***Safety Requirement****:* It is a high-level specification which defines the way in which a system must remain safe. In the present case, it means that a collision should be avoided at all costs. Collision is defined as the state in which two or more AMR nodes physically impact each other or an AMR node impacts a stationary object in the factory environment.

b) ***Safety trigger condition****:* To enforce this safety requirement, we need a specific function called an emergency-stop mechanism. This is done when the brakes of the AMR node fail, the node is unable to stop due to communication failure, the sensors fail or the momentum of the AMR nodes is too high to stop. For example, the emergency brakes should be triggered when any two AMR nodes are within 50 cm of each other and other trigger conditions with regard to speed, or communication packet loss rates.

The AMR nodes are assumed to be within 50 cm of each other (arbitrary) and approaching each other at speeds above the acceptable threshold. These conditions are conducive to invoke the wireless emergency stop safety function. The minimum distance indeed depends on the accuracy of the localization techniques used.

Two features of the proposed system design are considered - multi-band communication and FPGA-based run-time processing power. Typical industrial wireless bands for automation are in the ISM category - 2.4, 5 and 60GHz (milli-meter wave). In a highly scalable network with thousands of devices within a confined space, interference is bound to be high, especially when spectrum is shared by multiple users. In such a scenario, the cognitive engine in the design uses 60GHz band alongside a MIMO (Multiple Input Multiple Output) based antenna to produce sharp antenna beams (a.k.a millimeter-wave MIMO beam-forming) to direct the data towards a specific user. The advantage with millimeter-wave MIMO is the high antenna directivity it provides and low interference to other users. Therefore, the power delivered to the specific user in need of data is virtually guaranteed to receive it. This enhances reliability and safety packets can be sent by this means.

Secondly, let us consider the case when two or more nodes collide. From a performance perspective, there needs to be a mechanism in place for the system as a whole to recover from the loss of these nodes which now have to move to a safe state (and therefore, unable to continue their mission). The node closest to the affected node is empowered to become a controller. The FPGA inside the SDR provides the required processing power for the new controller. The controller chooses those nodes that are under-utilized and gives them the tasks that were originally assigned to the now quarantined nodes. It is essential to note that the controller mode is only temporary. The controller in addition to its management duties, also completes its own set of missions.

VII. Conclusion

The proposed hybrid design approach offers the best of centralised and decentralised design paradigms. We have seen how with various enabling technologies described in the paper, such a hybrid system can potentially convert an ordinary robot into a controller robot for a defined time duration in order to boost performance. The next step is to simulate the functioning of the cognitive engine and to evaluate its performance to achieve the desired flexible performance goals which we envisage for our hybrid system. Additionally, we wish to run a typical safety-critical automation application such as a motion planning algorithm to study the benefits in terms of reduced propensity for collisions.

References

[1] K. Zhou, T. Liu, and L. Zhou, "Industry 4.0: Towards future industrial opportunities and challenges," in *2015 12th International Conference on Fuzzy Systems and Knowledge Discovery (FSKD)*, pp. 2147–2152, Aug 2015.

[2] R. Alieiev, A. Kwoczek, and T. Hehn, "Automotive requirements for future mobile networks," in *2015 IEEE MTT-S International Conference on Microwaves for Intelligent Mobility (ICMIM)*, 2015.

[3] International Electrotechnical Commission (IEC), "Functional safety of electrical/electronic/programmable electronic safety-related systems, IEC- 61508, 2010.," Accessed: October 18, 2018.

[4] D. Kuschnerus, A. Bilgic, F. Bruns, and T. Musch, "A hierarchical domain model for safety-critical cyber-physical systems in process automation," in *2015 IEEE 13th International Conference on Industrial Informatics (INDIN)*, 2015.

[5] A. Ahmad and M. A. Babar, "Software Architectures for Robotic Systems," *J. Syst. Softw.*, vol. 122, pp. 16–39, Dec. 2016.

[6] S. L. Remy and M. B. Blake, "Distributed Service-Oriented Robotics," *IEEE Internet Computing*, vol. 15, pp. 70–74, Mar. 2011.

[7] M. Jung, A. Deguet, and P. Kazanzides, "A component-based architecture for flexible integration of robotic systems," in *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 6107–6112, Oct 2010.

[8] E. Ronen, A. Shamir, A. Weingarten, and C. O. Flynn, "IoT Goes Nuclear: Creating a Zigbee Chain Reaction," *IEEE Security Privacy*, vol. 16, no. 1, 2018.

[9] A. Mosenia and N. K. Jha, "A Comprehensive Study of Security of Internet-of-Things," *IEEE Transactions on Emerging Topics in Computing*, vol. 5, no. 4, 2017.

[10] Howstuffworks, "https://science.howstuffworks.com/hologram.htm," Accessed: October 17, 2018.

[11] International Electrotechnical Commission (IEC), "Security for industrial automation and control systems, IEC - 62443, 2010.," Accessed: October 15, 2018.

[12] ABB News, "https://new.abb.com/control-systems/system-800xa/800xa-dcs/800xa-networks/security-zoning," Accessed: October 16, 2018.

[13] J. Åkerberg, "On Safe and Secure Communication in Process Automation," *Mälardalen University Press Dissertations No. 109*, November 2011.

[14] Y. Zou, J. Zhu, X. Wang, and L. Hanzo, "A Survey on Wireless Security: Technical Challenges, Recent Advances, and Future Trends," *Proceedings of the IEEE*, vol. 104, no. 9, 2016.

[15] A. Mosenia, S. Sur-Kolay, A. Raghunathan, and N. K. Jha, "DISASTER: Dedicated Intelligent Security Attacks on Sensor-Triggered Emergency Responses," *IEEE Transactions on Multi-Scale Computing Systems*, vol. 3, no. 4, 2017.

[16] Meet Amazon's busiest employee - the Kiva robot, "https://www.cnet.com/news/meet-amazons-busiest-employee-the-kiva-robot/," Accessed: April 26, 2019.

[17] S. V. Dhage, A. N. Thakare, and S. W. Mohod, "An improved method for scalability issue in wireless sensor networks," in *International Conference on Innovations in Information, Embedded and Communication Systems (ICIIECS)*, pp. 1–6, 2015.

[18] B. Martens, M. Walterbusch, and F. Teuteberg, "Costing of Cloud Computing Services: A Total Cost of Ownership Approach," 01 2012.

[19] G. Baldini, T. Sturman, A. R. Biswas, R. Leschhorn, G. Godor, and M. Street, "Security Aspects in Software Defined Radio and Cognitive Radio Networks: A Survey and A Way Ahead," *IEEE Communications Surveys Tutorials*, vol. 14, no. 2, 2012.

[20] GNURadio, "https://www.gnuradio.org," Accessed: October 18, 2018.

[21] E. Grass, K. Tittelbach-Helmrich, C.-S. Choi, F. Winkler, T. Ohlemller, and R. Kraemer, "Communication systems operating in the 60 GHz ISM band: Overview," vol. 3, pp. 89 – 97, 04 2011.

[22] M. Chiang, S. Ha, C. I, F. Risso, and T. Zhang, "Clarifying Fog Computing and Networking: 10 Questions and Answers," *IEEE Communications Magazine*, vol. 55, no. 4, 2017.