OSTBAYERISCHE
TECHNISCHE HOCHSCHULE
REGENSBURG

FMS-BERICHTE
SOMMERSEMESTER 2022

Hans Meier, Michael Niemetz, Thomas Fuhrmann
und Andrea Reindl (Hrsg.)

Seminar zu aktuellen Themen der Elektro- und Informationstechnik

Forschungsmethoden-Seminar, Ausgabe 4,
Sommer 2022

5. Oktober 2022

Vorwort

Dieser Bericht entstand im Rahmen der Lehrveranstaltung „Forschungsmethoden und Seminar (FMS)“ im Sommersemester 2022 auf Initiative der Studierenden des Masterstudiengangs „Elektro- und Informationstechnik (MEI)“.

Diese Lehrveranstaltung hat das Ziel, systematisch an das wissenschaftliche Arbeiten, speziell die Wissenschaftskommunikation, heranzuführen. Daher war geeignete Literatur zu einem individuellen Thema zu recherchieren, Veröffentlichungen auf ihre Relevanz hin zu beurteilen und letztendlich eine eigene Ausarbeitung basierend auf der Literaturrecherche zu erarbeiten und diese in einem Vortrag zu präsentieren.

Parallel dazu erfolgte im Theorieteil die entsprechende Hinführung zu den verschiedenen Elementen der Wissenschaftskommunikation:

- Bedeutung der Wissenschaftskommunikation für die Arbeit der Ingenieure in Forschung und Entwicklung
- Literaturrecherche, Suchmaschinen, Sichtung und Analyse vorhandener Publikationen, Bewertung der Qualität aufgefundener Fachliteratur, Auswahl geeigneter Materialien für die eigene Arbeit
- Aufbereitung und Darstellung der recherchierten technischer Inhalte in Form einer seitenanzahlbegrenzten wissenschaftlichen Ausarbeitung
- Einhalten formaler Randbedingungen bzgl. Strukturierung, einschl. Bildnachweise und Zitationsstile
- Peer-review-Prozess bei wertschätzender Beurteilung der Leistung anderer
- Publikumsangepasstes Aufbereiten komplexer fachlicher Inhalte mit hochschulöffentlicher Präsentation der Ergebnisse
- Führen mündlicher wissenschaftlicher Diskurse

Nachdem die Masterstudierenden in der Regel über noch keine eigene wissenschaftliche Forschungserfahrung bzw. -inhalte verfügen, lag der wählbare Schwerpunkt der Literatursuche auf der Bearbeitung von vorgegeben aktuellen technischen oder gesellschaftspolitischen Forschungsthemen.

Inhaltsverzeichnis

Sensor Systems of Autonomous Robots in Industrial Environments

Valentin Josef Lermer
Faculty of Electrical Engineering and Information Technology
Ostbayerische Technische Hochschule Regensburg
Regensburg, Germany
valentin.lermer@st.oth-regensburg.de

***Abstract*—This paper examines the requirements for sensor systems in autonomous robots and highlights the challenges and current technical trends. The aim is to structure the subject area and to give an uplifting overview. The scope of this paper is the industrial environment. The goal of autonomous robotics is to let a robot act independently, make autonomous decisions and move in unknown places. A typical industrial robot, on the other hand, performs predetermined tasks repeatedly and moves in a known workspace. Thus, in the examination, the requirements for each type of robot are categorised and the sensor technology is compared. In industrial robots, sensors are used to ensure and execute a sequence of programmed activities. These are handled according to the Input-Process-Output (IPO) model. Individual sensors directly influence the decisions. The focus of sensor technology in autonomous robots lies on providing data based on which the independent and intuitive decisions are made of. Therefore, data is processed and combined. The modified requirements for autonomous robots indicate a change in sensor technology and the approach. The classification of internal and external sensor technology is still given. For autonomous robots the interaction with the environment is significant. Various sensor systems are explained and categorized in this paper. The movement in space as well as the recognition of objects are particularly relevant. In addition, an outlook on the further steps after data acquisition is given. The basis for the further processing of acquired data is the linking of sensors (sensor fusion) and the creation of sensor networks. Thus, low-cost hardware should provide the data for deep learning and further algorithms. The requirements for autonomous sensor systems are more versatile and complex than for industrial robots. It is the basis for many methods.**

***Index Terms*—autonomous robot, sensor systems, autonomous systems, industrial robot, robotics**

I. Introduction

The topic of autonomous systems is more relevant than ever. Too many challenges suggest autonomous solutions. This is also the case in industry, where autonomous robots have been gaining more and more ground for years. The industrial robots already in use there provide a basis of experience and data. The number and variants of sensors and sensor systems is difficult to subdivide. In addition, the question arises whether the changed requirements for autonomous robots also have an effect on the sensors used. This extensive topic is considered in the industrial environment. Driverless transport systems, collaborative robots and autonomous cars are not dealt with in detail. At the beginning of the paper, the basics and background of industrial and autonomous robots are presented and defined in chapter II. The following literature review is divided into two sections: In chapter III, the requirements for robots are first compared and then the sensors used are dealt with in detail. Finally, a conclusion of the comparison is made in chapter III-C. The second area is found in chapter IV. Here, the challenges of modern autonomous robots are categorised and elaborated. Finally, section V summarises the work and highlights the results.

II. Basics and Backgrounds

In the following, classical industrial robots and autonomous robots are defined and described in more detail. The course is structured by mechanics, hardware and software. Then the focus is on autonomy in robots and an overview of their application.

In ISO 8373:2021, an industrial robot (IR) is defined as

> automatically controlled, reprogrammable multipurpose manipulator, programmable in three or more axes, which can be either fixed in place or fixed to a mobile platform for use in automation applications in an industrial environment[1].

The mechanical structure of a robot can be arranged in various ways. Different components such as joints and axes are used to perform translational and rotational movements. This allows a tool or similar device to be moved and positioned to perform the assigned tasks. For structured programming, there are coordinate systems that define positions, points or objects in geometric space. There is the cartesian coordinate system, in which all directional axes are orthogonal to each other. And the polar coordinate system, in which a point is defined with an angle and distance from the origin. To execute tasks, a robot is equipped with different sensors and actuators. A sensor transfers a physical phenomenon to a workable signal, which is the basis for the IPO model. Different types of sensors are shown and discussed in chapter III. The output is implemented with actuators like grippers, tools or motors for axis control. Programming takes place online, offline or in hybrid form. Points in space are taught-in manually, locations in the coordinate system are programmed and simulated or CAD data are transferred. An industrial robot performs predetermined tasks repeatedly and moves in a known workspace.[2]

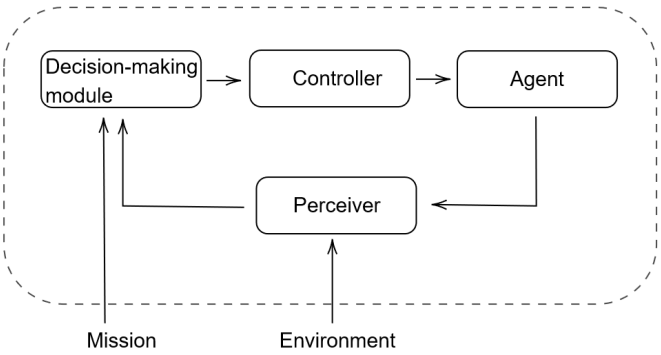

Fig. 1. a structural overview of autonomous systems data flow

The goal of autonomous robots (AR) is to let a robot act independently, make self-contained decisions and move in unknown places. The basic structure of autonomous robots is similar to industrial robots. This includes the mechanical structure and parts of the sensor and actuator technology. The differences in sensor technology are shown in chapter III-B. Autonomous robots are graded in different degrees of autonomy. The minimum requirement is an existing unit of software and electronics. Higher systems are independent of the energy supply and thus permanently operational without external intervention. The architecture of autonomous systems illustrated in figure 1 shows their basic structure. The sensing devices (sensors) detect the environment (perceiver) and pass on data to the decision-making module. The decision-making module decides on the commands to be sent to the controller, taking into account the specifications and information from the sensors. The agent executes them. In this way, it is possible to independently achieve a goal by means of a one-time specification. The basic elements of autonomous systems are the perception and understanding of the environment.[3,4] The decisions made in the decision-making field of autonomous systems are done on the basis of artificial intelligence (AI) and programmed rules and decisions. The task of an AI is to create technical systems that can comprehend and imitate intellectual behaviour. Sub-areas are Machine Vision (MV), which is used in the field of image processing and pattern recognition. One focus is on determining the meaning of images and thus drawing conclusions about content from the image. The mass of sensor data in industry leads to its use for Deep Learning (DL) methods[5]. This is used to train neural networks and thus improve the intelligence of the systems.[2,4]

III. Robotic Sensors

Industrial robots and autonomous robots were defined and explained in the previous chapter. In the following, the focus is placed on the sensors of the robots and their requirements. The results are finally compared and interpreted.

A. Requirements for Robots

The focus of industrial robots is on handling workpieces or tools and performing various activities. These include pick-and-place, sorting, drilling, assembling, painting, welding, measuring in industrial environments. The tasks are directly delimited and the robot is specialised for the respective job. With autonomous robots, the focus of requirements are:

- orientation and flexible navigation in foreign or known environments
- autonomous planning of the optimal path to a destination
- collision avoidance and
- object recognition

The different applications of sensors are listed in III-C. Thus, the different requirements for robots are shown and the sensors can be looked at more closely.[6,7]

B. Overview of Applied Sensors

The tasks of sensors in robotics are very different and of significant importance. First, the internal states (proprioceptive) must be detected, such as speeds, forces, battery status or temperature. This includes the position of the robot and the tool/handling objects mounted on the end effector. The external areas (exteroceptive) are the environment, brightness. Obstacle detection and the identification of parts. For external sensors, a distinction is made between short and longer ranges. At short ranges, it only matters whether an object is present to avoid a collision. At longer ranges, it is also of interest to determine the position, distance and orientation.[8]

Various criteria are taken into consideration when selecting sensors. The following excerpt is intended to provide an overview: When measuring the range of a sensor, it is important to consider the span between the minimum and maximum values that can be measured. In addition, sensitivity plays an important role in how far changes in the primary physical signal affect the output signal. The consistency of the relationship between input and output data is also important. How accurately does the sensor measure and at what resolution (smallest measurable step size of the input signal). The repeatability also plays a role, as does the reaction time or switching time of a sensor from the acquisition of the physical input variable to the signal output.[6]

Internal sensor systems often consist of a combination of accelerometers, gyroscopes and partly magnetometers. The accelerometer measures the rate of change of the object's velocity. They detect e.g. vibrations in the system. gyroscope detects the angular velocity and shows apptupped and fast movements. Both sensors are operated in parallel and complement each other. A magnetometer detects the orientation, but is mainly used outdoors.[8]

The sensors listed below are categorised in the table I according to internal and external. They are also checked for application in industrial robot and autonomous robot.
Tactile sensors detect the position, dimension and shape of objects by physical contact. These are often used as limit switches. These include bumpers, which are used as direct collision sensors and keep consequential damage to a minimum. Whiskers work in a similar way, sensing with strain gauges in flexible plastic strips. Contactless sensors are used

TABLE I
ROBOTS SENSORS[6]

measurement	***sensor***	***IR***	***AR***
battery charge	Volt/ampere meter	✗	✓
internal temperature	thermometer	✓	✓
travelled path	odometer	✗	✓
slope, tilt angle	clinometer	✗	✓
direction	compass	✗	✓
speed	tachogenerator, incremental encoder in the drive system	✗	✓
speed, acceleration axes	incremental encoder in the drive system	✓	✓
acceleration	acceleration/internal sensor	✗	✓
motor currents	ammeter for drive control	✓	✓
internal forces/torques	pressure, force and torque sensors	✓	✓
light	photoelectric sensors	✗	✓
sound	microphones, sound level meters, noise dosimeters	✗	✓
outside temperature	thermometer, temperature sensor	✗	✓
distances, distances obstacle detection	optical and acoustic distance sensors, laser, radar, (tactile) contact sensors, such as bumpers, whiskers	✓	✓
position of objects, proximity of objects	inductive, capacitive and optical proximity sensors	✓	✓
contour of objects	cameras, photodiodes, transistors, 3D sensors	✗	✓
force, pressure, moments	piezoelectronic transducers, force-torque sensors	✓	✓
detection of objects	tactile sensors, strain gauges, tactile sliding sensors	✗	✓

for similar purposes. Encoders react to an approach and trigger a binary switching signal. Visible light, infrared and ultraviolet radiation are used. Examples are optical proximity switches or light barriers. The sensors used are opto-electronic components such as laser and reflex light barriers, inductive and capacitive proximity switches, Hall and Wiegand sensors. A common type is the light detection and ranging or light imaging, detection and ranging (LiDAR) sensor. Ultrasonic sensors are also included. These emit sound and pick up the reflected signal again. This makes it possible to detect the distance of the object, which is between 40 cm and 10 m. Laser and infrared sensors also work with time-of-flight methods. Image processing by cameras represents a large part of robot sensor technology. Photoelectric sensors transfer optical information into electrically evaluable signals (voltage, current, resistance). They are called charge-coupled device (CCD) sensors and consist of an array of light-sensitive photodiodes to capture two or three-dimensional images (RGB, RGB-Depth). This allows features of objects such as position, orientation and identification to be recorded. The focus is on general image recognition, object recognition, analysis of movements, pattern recognition, completeness check, etc. The data is fed into the robot's control system. The acquired data is transferred to the robot coordinate system and processed. Stereo sensors determine points in space via two cameras positioned in space.[2,6,9]

C. Results of Robotics Comparison

The comparison of the two robots (IR and AR) in III-B shows that both are constructed similarly, but the requirements vary. Whereas in the case of the IR, processes are permanently programmed, the autonomous robot can make learned decisions itself and is thus more versatile. This change is also noticeable in the sensors: both the internal and external areas have higher requirements and thus quantitatively more sensors are necessary. This is due to the fact that autonomous systems are significantly dependent on the perception of their environment.[6]

IV. Sensor Systems and Focus of Autonomous Robots

The sensors shown in chapter III are combined and used for various tasks. The technology of autonomous robots is characterised by four basic decision-making problems. These issues are outlined in IV-A and IV-B:

- mapping
- localization
- path planing and obstacle avoidance
- object recognition

A. Map Building and Localization

The position and orientation of an autonomous robot must be known in order to take further steps. The associated technology must minimise errors and compensate for inaccuracies. A distinction is made between relative and absolute localisation, which can also be combined. With the relative method, the entered starting point is stored and the distance travelled is measured using encoders or inertial sensors. This method is very precise, but not usual for long distances because of the data required. The absolute method, on the other hand, uses various sensors such as passive landmarks, map matching, wifi or satellite-based signals such as the Global Positioning System (GPS). [10] This is independent of location (indoor, outdoor), time and can be used over longer distances and durations. However, it is less precise.[2,6]

A map depicts the environment in which the robot moves. Mapping is the process in which the data is collected. This is done manually (by a person), automatically by a guided exploration. The creation of maps is necessary for the robot to move in unknown areas and to set start and destination points. In many cases, the map does not exist before the target is set.[11] In this case, the Simultaneous Localisation And

Mapping (SLAM) method is used. Autonomous robots use various sensors to collect environmental information to model surroundings and localize their current state. SLAM is a common perception method in current autonomous systems. Visual sensors such as RGB cameras can provide more information than lidar sensors.[3,12]

B. Path Planning and Object Recognition

An intelligent robot is required that could travel autonomously in various static and dynamic environments.

Robot path planning is distinguished between local and global navigation. These differ in distance, scale, object avoidance and the ability to detect the target. With global navigation, the environment is not known. With local navigation, the map is already provided with landmarks and explored[13]. The difficulty is that environments change dynamically. The associated challenge is obstacle avoidance. When the robot is on the way to the target, a collision should be avoided. This means anticipating avoidance and stopping at short notice.[6] When a mission is given, autonomous systems must arrive at the designated place to complete the specific task.[14,15]

The challenges listed in chapter IV-A relate to the movement of a robot in space. Now follows the recognition of objects. The goal here is to characterise objects by means of sensors and to carry out tasks. These range from recognising the position to setting features. The equipment of sensors and methods depends on the requirements and given parameters for recognition.[14]

Sensor fusion is a multi-stage process that examines the correlation and linking of data from different sensors. This evaluates situations, makes estimates and improves the accuracy and quality of the content. The aim is to reduce costs. Sensor fusion is applicable to many areas of robotics.[3,6]

V. Conclusion

The past elaboration has shown the versatility of the topic. Thus, autonomous robots are becoming much more complex in terms of requirements and handling of sensors. The research question shows that the development of autonomous robots has an impact on sensors. The focus of AR is much more on interacting with the environment. The perception of the environment is highly prioritised and therefore implemented with sensors of different types. Industrial robots are suitable for repetitive and containable tasks. For complex challenges, autonomous robots are the choice.

References

[1] "Iso 8373:2021, robots and robotic devices—vocabulary." [Online]. Available: https://www.iso.org/standard/55890.html

[2] H. Maier, *Grundlagen der Robotik*. Broschur, 2022.

[3] Y. Tang, C. Zhao, J. Wang, C. Zhang, Q. Sun, W. Zheng, Du Wenli, F. Qian, and J. Kurths, "Perception and navigation in autonomous systems in the era of learning: A survey," *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–21, 2022.

[4] Y. D. V. Yasuda, L. E. G. Martins, and F. A. M. Cappabianco, "Autonomous visual navigation for mobile robots," *ACM Computing Surveys*, vol. 53, no. 1, pp. 1–34, 2021.

[5] L. Romeo, A. Petitti, R. Marani, and A. Milella, "Internet of robotic things in smart domains: Applications and challenges," *Sensors (Basel, Switzerland)*, vol. 20, no. 12, 2020.

[6] M. B. Alatise and G. P. Hancke, "A review on challenges of autonomous mobile robot and sensor fusion methods," *IEEE Access*, vol. 8, pp. 39 830–39 846, 2020.

[7] C. Wong, E. Yang, X.-T. Yan, and D. Gu, "Autonomous robots for harsh environments: a holistic overview of current solutions and ongoing challenges," *Systems Science & Control Engineering*, vol. 6, no. 1, pp. 213–219, 2018.

[8] Niall O'Mahony, Sean Campbell, Anderson Carvalho, Suman Harapanahalli, Gustavo Adolfo Velasco-Hernandez, Daniel Riordan, and Joseph Walsh, "Ieee 5th world forum on internet of things: 15-18 april 2019, limerick, ireland : conference proceedings," *conference proceedings*, 2019.

[9] Sukkpranhachai Gatesichapakorn, Jun Takamatsu, and Miti Ruchanurucks, "2019 first international symposium on instrumentation, control, artificial intelligence, and robotics: Chulalongkorn university, bangkok, thailand, 16-18 january 2019," 2019.

[10] R. C. Alves, J. Silva de Morais, and K. Yamanaka, "Cost-effective indoor localization for autonomous robots using kinect and wifi sensors," *Inteligencia Artificial*, vol. 23, no. 65, pp. 33–55, 2020.

[11] C. Cadena, L. Carlone, H. Carrillo, Y. Latif, D. Scaramuzza, J. Neira, I. Reid, and J. J. Leonard, "Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age," *IEEE Transactions on Robotics*, vol. 32, no. 6, pp. 1309–1332, 2016.

[12] T. Tsubouchi, "Introduction to simultaneous localization and mapping," *Journal of Robotics and Mechatronics*, vol. 31, no. 3, pp. 367–374, 2019.

[13] B. Khulna, "2013 international conference on electrical information and communication technology (eict 2013): Khulna, bangladesh, 13 - 15 february 2014 [postponed from december 2013]," *EICT 2013*, 2014.

[14] A. Dzedzickis, J. Subačiūtė-Žemaitienė, E. Šutinys, U. Samukaitė-Bubnienė, and V. Bučinskas, "Advanced applications of industrial robotics: New trends and possibilities," *Applied Sciences*, vol. 12, no. 1, p. 135, 2022.

[15] E. Krell, A. Sheta, A. P. R. Balasubramanian, and S. A. King, "Collision-free autonomous robot navigation in unknown environments utilizing pso for path planning," *Journal of Artificial Intelligence and Soft Computing Research*, vol. 9, no. 4, pp. 267–282, 2019.

Closed-Loop Approaches in Organic Waste Management using Black Soldier Fly

Simon Heiß
Faculty of Electrical Engineering and Information Technology
OTH Regensburg
Regensburg, Germany
simon1.heiss@st.oth-regensburg.de

***Abstract*—Organic waste accounts for the majority of municipal solid waste generated worldwide. Especially in highly populated urban residential areas of major cities, lacking biowaste management can result in various problems. Production of greenhouse gases and wastage of useful resources name a few of them. Additionally, organic waste offers great potential for its usage in circular economy. Composting, vermicomposting and other enhancements of it can be used for organic waste to be recycled into bioproducts for various different sectors. The use of insects, especially the black soldier fly, is a high prospect and versatile method in composting for valorizing biowaste into high value products like biodiesel, animal fodder and biological fertilizer as a byproduct. This paper will show the basic functionality of vermicomposting, the largest organic waste streams, and it will compare black soldier fly composting to vermicomposting. The paper will explain the basic principles of three approaches to valorize biowaste in a circular economy. Each approach will be supported by qualitative and quantitative results. The paper will not cover small-scale processes of biowaste management. It will also not propose a new method of building a circular economy using biowaste or discuss the energy efficiency impact of listed methods.**

***Index Terms*—Waste management, Recycling, Environmental management, Circular economy, Insect farming, Black soldier fly, Composting**

I. Introduction

The predominance of a linear economy combined with a steadily growing population results in increasing emergence of solid waste. The world generates around 2.01 billion tonnes of municipal solid waste per year, the main part of it being organic waste. [1] This amount will even rise to approximately 3.4 billion tons within the next 25 years. [2] Next to these problems, there is also the rising protein demand, which will be a challenge for the future generations. [3] The concept of circular economy (CE) gains a significant amount of traction in this context. The European Parliament describes a CE as follows:

> ”The circular economy is a model of production and consumption, which involves sharing, leasing, reusing, repairing, refurbishing and recycling existing materials and products as long as possible.” [4]

Solving multiple problems at once can potentially be done using *Hermetia illucens* black soldier fly (BSF). BSF can decompose organic waste from various waste streams, including industrial organic waste such as kitchen waste, manure and even fecal sludge. Its larvae can be processed into biodiesel or animal feed, the decomposition itself can at least produce compost as biological fertilizer. The fertilizers can be used to increase the yield of several crops and plants, whose inevitable wastes can again be composted. BSF larvae mostly contain protein and fat, therefore they can be processed to animal feed. Due to the fact that even manure from cattle, pigs and chicken can be decomposed and bioconverted by BSF, a CE can be attained this way. The biodiesel can also be used to fuel tractors or machinery used in agriculture. In a higher meta-level, the circle is completed again - at least, no fossil fuels have to be used in this case.

Section II of this literature paper will explain the basic functionality vermicomposting, and show which large scale biological waste streams emerge in daily live. It will also compare vermicomposting to the BSF process and show the main advantages of BSF. Section III will further elaborate how the BSF process allows a CE and take a closer look at each of the three possibilities concerning process, results and advantages or disadvantages compared to conventional products. Section IV will give an overview over this theoretical paper and sum up the most important information.

II. Composting Comparison and Industrial Organic Waste

A. Vermicomposting

In regular composting, only microorganisms perform decomposition. This happens in four stages, where different bacteria are active in different temperature phases. Vermicomposting describes the process of composting with the additional application of earthworms for breaking down organic matter. Different worm species can be used, the most popular are red wigglers (*Eisenia fetida*), European nightcrawlers (*Eisenia hortensis*) and the red earthworm (*Lumbricus rubellus*). The application of worms to the organic material results in a faster decomposition process compared to simple microbe composting. Vermicomposting can be up to three times faster compared to normal composting. [5]

B. Organic Waste Streams

The organic part of the approximately 2 billion tons of generated solid waste per year mainly splits up into the following waste streams [6]:

- Agricultural waste
- Yard and forestry waste
- Sewage sludge waste water
- Food processing waste
- Organic fraction of municipal solid waste

This mostly aligns with the 2019 report of Umweltbundesamt (Federal Environmental Agency), which states that the main contributions of disposed municipal solid waste were yard and forestry wastes (30.3%), food wastes from private households (30.1%), agricultural and food processing waste (16.0%) and municipal sewage sludge (7.7%). [7] The portion of agricultural waste may differ by so far, because a lot of the produced organic waste gets redistributed to fields as manure.

C. Comparison BSF - Vermicomposting

Since BSF composting is a rather new process in comparison to worm composting, the main differences between the two processes are valuable information. Table I compares BSF composting and vermicomposting with *Eisenia fetida*.

$$FCR = \frac{Weight\ of\ feed\ intake}{Weight\ gained\ by\ animal} \quad (1)$$

TABLE I
Hermetia illucens - *Eisenia fetida* - COMPARISON

	Earthworm	**BSF**
Waste reduction[a]	60% .. 70% [5]	39% .. 68% [8]
FCR[b]	N/A	9.6 .. 14.5 [8]
Life-cycle	1-5 years [9]	45 days [10]
Offspring/life-cycle[c]	43 [9]	254 [11]

[a]Weight-based reduction
[b]Feed Conversion Ratio
[c]Corrected to mortality in development phase

The FCR of *Eisenia fetida* could not be retrieved. As seen in table I, the weight-based waste reduction is higher when using earthworms for composting. However, BSF larvae have an advantage in feedstock selection. While earthworms can not be fed every kind of waste stream, the feedstock variety for BSF ranges from kitchen wastes, over animal manure to even faeces. [12] [8] The shorter life-cycle and higher reproduction rates give BSF a leading edge over earthworms in terms of biomass production.

III. Black Soldier Fly Process

A. Circular Economy with BSF

How aspects of CE are applied to the BSF process can be seen in Fig. 1. The life-cycle of BSF lasts around 45 days [10], and continues with observed reproduction rates of up to 650 eggs laid per adult female fly [11]. This number reduces throughout the stages of development. Only about 70% survive the 4-day-long phase of growing as eggs. From there on, about 70% survive the 18-day larvae stage. In the 14-day pupae stage, the survival rate rises to around 80%, which is continued by the adult stage after approximately 9 days. [13]

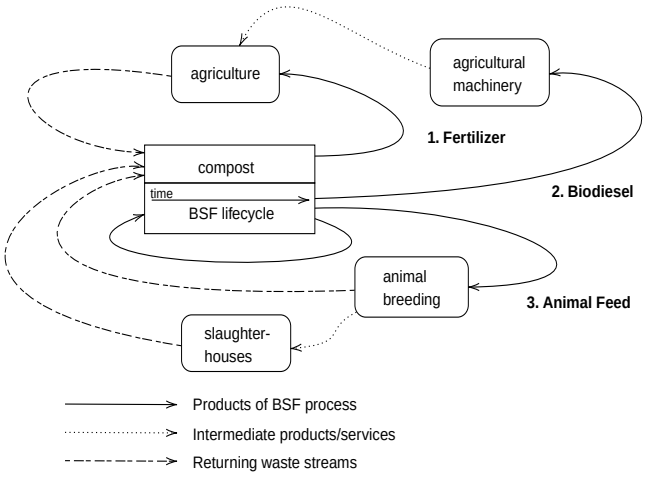

Fig. 1. Circular Economy in the BSF process

1) Biological Fertilizer: While the BSF larvae are feeding on the provided biomass, they consume the material, metabolize parts of it and excrete so-called frass. These residues have fertilizer value and can be used in agriculture to promote plant growth. Fertilizer can be purchased for a wide spectrum of plants and even development stages of plants. The NPK-composition (Nitrogen, Phosphorus, Potassium) is a key factor for providing a suitable fertilizer. BSF larvae-aided composting provides better fertilizer than conventional microbe composting [14]. Several studies show that BSF compost can be used as soil amendment [15]. A study about BSF compost impact on three key vegetable crops in Sub-Saharan Africa concluded that BSF compost has a quantitative advantage over commercially available organic fertilizer and mineral fertilizer in terms of plant height, number of leaves grown and stem diameter. [16] Another study conducted in Indonesia compared the influence of soil, soil with compost, soil with chemical fertilizer and soil enriched with 5-10-15% BSF compost in terms of growth of pakchoi. The three soil-BSF-compost mixtures yielded higher pakchoi than the combination of soil and chemical fertilizer. After five weeks of growing, the BSF compost plants had about twice as many leaves and 75% more height than the ones treated with chemical fertilizer. The effect of the soil-BSF-compost mixtures was comparable to soil with conventional organic fertilizer. [17] Next to the plant growth advantages, there are also ecological advantages over conventional fertilizer. The substitution of commercial N-fertilizer with BSF compost can lower the carbon footprint by up to 432 kg of CO_2 equivalent per tonne of treated waste. [18]

2) Biodiesel: The raw materials for the production of biodiesel are some form of vegetable or animal oils or fats

and methanol. A catalyst makes this react into biodiesel and glycerin. Biodiesel is mainly produced from edible oils and fats, which brings some ethical and financial problems. Firstly, the production of fuel should not interfere with the global food chain, especially when a growing population with an again growing protein demand as seen in Section I. Secondly, the increasing production of biodiesel from edible oils may increase the required acreage and therefore increase the cost of raw material. Fortunately, the larvae and prepupae of BSF mostly contain of fat and protein, they are an ideally suitable raw material for this biodiesel production. Figure 2 shows the process from BSF rearing to biodiesel. The larvae feed on different sorts of organic solid waste and when they reach prepupae stage (transition between larvae and pupae), they are harvested and dried in an oven at 60 °C (after being deactivated at 105 °C for 5 minutes). The dried BSF prepupae are blended to get a uniform material. Mechanical extraction can be performed by presses, which separate fats (oils) and solids (pressed cake). The pressed cake can be processed further through chemical extraction using a Soxhlet apparatus. In this process, the material extraction happens with petroleum ether as a solvent. The dissolved mixture is separated using a centrifuge or a rotary evaporator. The solid components (pressed cake and meal) can be further processed into animal feed. The extracted BSF oil is valorized into biodiesel using a two-step transesterification process, namely acid-catalysed esterifcation and alkaline-catalysed transesterifcation [19].

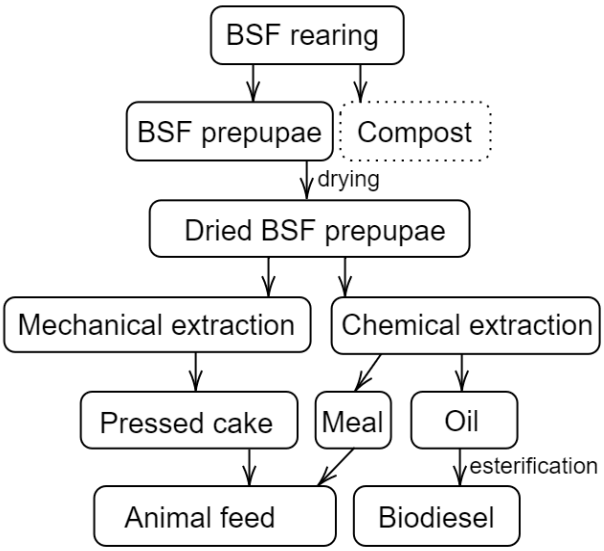

Fig. 2. BSF-to-biodiesel process

A 2012 research project conducted by chinese scientists assessed the possibility of producing biodiesel from BSF larvae fed on a mixture of 70% restaurant solid wastes and 30% rice straw. 2000 BSF larvae were grown on 1 kg biomass. Additionally, microbes (Rid-X) were applied to the biomass conversion process (in mixing ratios of up to 0.5% of organic matter). After 10 days of co-conversion (at a rate of 0.35% Rid-X), about 43.8g of biodiesel was produced. The quality of this fuel was comparable to rapeseed biodiesel and met the specifications for biodiesel set by EN 14214. [20]
Another study on this topic examined the differences in waste streams that act as feedstocks for BSF larvae that will later serve for biodiesel production. Manure from cattle, pigs and chicken were used as feedstock and 1000 larvae were applied per kg organic waste. In 10 days, 32.8%, 20.7% and 12.8% were converted from chicken, pig and cattle manure, respectively, into BSF biomass. The BSF on chicken manure yielded 91.4 grams of biodiesel, the ones on pig and cattle manure resulted in 57.8 and 35.5 grams. All three BSF biodiesel sorts were compliant to EN14214 and comparable to rapeseed biodiesel. [21]
There are some more studies that convey similar results [19] [22], especially proving that BSF rearing and production of biodiesel is possible and comparable to biodiesel from vegetable oils and compliant with EN14214.

3) Animal Feed:

> Meat represents 15% of energy in the global human diet, while approximately 80% of agricultural land is used for animal grazing or the production of livestock feed and fodder [23].

To reduce this amount of agriculturally used land, it is important to investigate if BSF can be used as feed addition or even feed substitution. In animal production, there are multiple different factors that influence the choice of animal feed. Feed composition, nutrient availability, non-contamination with hazards or substances, availability and cost being some of them. A 2005 study concluded that the cost of a tonne of dry BSF meal could be at 330 USD [24]. This was 37.5% more expensive than soybean meal at that time (around 240 USD) [25]. Since BSF can be reared on various types of organic waste, which emerges everywhere, BSF meal can potentially be produced locally to have shorter transport to the animal production plants. Most studies on rearing BSF were conducted as small-scale experiments, and the results do not necessarily translate to industrial production. However, pilot studies like the 2020 study done by the Department of Entomology at Texas A&M University [26] deliver useful data for prospect companies. Another study also assessed the scalability of BSF rearing and fodder production and came to the conclusion, that fresh BSF biomass production is almost twice more sustainable than fresh chicken meat. [27] Hazard and substance contamination underlies strict specifications in animal production and must be considered when using BSF as feedstock, especially when organic waste is used. Purschke *et al.* found that heavy metal contamination deteriorates larval growth and indicates a food safety risk [28]. Nutrient availability and composition are often assessed when researching BSF for consumption. Next to the sole composition of the main three nutrients (protein, fat and carbohydrates), another key factor for feed-to-biomass conversion is the amino acid profile, which represents the detailed protein composition. Within the

amino acid profile are the essential amino acids, which an organism needs, but can not produce itself. A study done by Siddiqui *et al.* compared the essential amino acid profile of BSF meal against fish meal, soybean meal and the 2013 FAO (Food and Agriculture Organization of the United Nations) recommendations.

TABLE II
AMINO ACID COMPARISON

Amino acid[a]	BSF meal	Fish meal	Soy meal	FAO
Histidine	62	63	26	16
Isoleucine	48	20	10	30
Leucine	77	65	27	61
Lysine	74	69	22	48
Methionine	6	26	5	23
Phenylalanine	62	33	18	41
Threonine	45	39	15	25
Tryptophan	NA	9	5	6.6
Valine	67	45	17	40

[a]All values in mg/g

Table II shows that while soy meal underscores each of the FAO recommendations, BSF reaches these scores in seven out of nine essential amino acids. While these data only consider protein, the whole nutrient composition is deeply researched as well. The macronutrient composition of BSF larvae depends on the feedstock on which it grows. Singh *et al.* conducted a study in which different feedstocks and their influence on larvae nutritional composition were compared. The feedstocks varied from different sorts of manures over food manufacturing by-products to only vegetable and fruit waste. Surprisingly, the BSF metabolized these different sources to an almost stable amount of crude protein at around 37 to 46% of dry biomass. However, the fat percentage varied from 18 to 41%. [29] Multiple other studies also assessed the protein and fat percentages in BSF larvae dry weight, resulting in around 40% of crude protein and 30% fats. [30] [31] [32]
Additionally, BSF meal has lower water and land usage compared to fish or soy meal production [33]. Aquaculture studies [34] [35] showed that partial replacement (19% and 25%, respectively) of traditional fish meal did not cause growth differences in fish farming.

IV. CONCLUSION

The growing population on earth currently faces some major problems, two of them being a growing protein demand and a rising waste generation. In this paper we showed, that black soldier fly can act as a solution for both of these problems, even incorporating approaches of circular economy. BSF vermicomposting was compared to worm vermicomposting. Three approaches of CE were presented, backed with scientific data. BSF can transform a variety of organic waste streams into valuable biomass and compost as a by-product. The compost shows great potential as a substitution or replacement of chemical or mineral fertilizers while generating less CO_2 and greenhouse gases. The BSF biomass contains around 40% of protein and 30% of fats. The high nutritive content makes BSF biomass a suitable source of animal feed, which nutritive facts can compete with fish and soy bean meal, and at the same time is more environmentally friendly. The fats contained in the biomass can be mechanically and chemically extracted and converted into biodiesel, which is compliant to EN14214. Overall, the BSF shows great potential in agricultural use and waste management. Companies and start-ups engage with the BSF process to make its usage more scalable and prepare insect farming for human consumption.

REFERENCES

[1] "Trends in solid waste management," 20.09.2018. [Online]. Available: https://datatopics.worldbank.org/what-a-waste/trends_in_solid_waste_management.html

[2] S. Kaza, L. Yao, P. Bhada-Tata, and F. Van Woerden, *What a waste 2.0: a global snapshot of solid waste management to 2050*. World Bank Publications, 2018.

[3] S. W. Kim, J. F. Less, L. Wang, T. Yan, V. Kiron, S. J. Kaushik, and X. G. Lei, "Meeting global feed protein demand: Challenge, opportunity, and strategy," *Annual review of animal biosciences*, vol. 7, pp. 221–243, 2019.

[4] "Circular economy: definition, importance and benefits — news — european parliament," 2015. [Online]. Available: https://www.europarl.europa.eu/news/en/headlines/economy/20151201STO05603/circular-economy-definition-importance-and-benefits

[5] W. Herumurti, E. Rahmawati, and S. A. Wilujeng, "Influence of earthworm to organic waste ratio and cow manure mixture in vermicomposting process using eisenia fetida," pp. 1–7, 2016.

[6] W. Bidlingmaier, J.-M. Sidaine, and E. K. Papadimitriou, "Separate collection and biological waste treatment in the european community," *Reviews in Environmental Science and Bio/Technology*, vol. 3, no. 4, pp. 307–320, 2004.

[7] Umweltbundesamt, "Bioabfälle," 12.05.2022. [Online]. Available: https://www.umweltbundesamt.de/daten/ressourcen-abfall/verwertung-entsorgung-ausgewaehlter-abfallarten/bioabfaelle#bioabfalle-gute-qualitat-ist-voraussetzung-fur-eine-hochwertige-verwertung

[8] S. Diener, N. M. Studt Solano, F. Roa Gutiérrez, C. Zurbrügg, and K. Tockner, "Biological treatment of municipal organic waste using black soldier fly larvae," *Waste and Biomass Valorization*, vol. 2, no. 4, pp. 357–363, 2011.

[9] J. M. Venter and A. J. Reinecke, "The life-cycle of the compost worm eisenia fetida (oligochaeta)," *South African Journal of Zoology*, vol. 23, no. 3, pp. 161–165, 1988.

[10] R. Ferrarezi, "Uvi/aes annual report 2016 - alternative sources of food for aquaponics in the u.s. virgin islands: A case study with black soldier flies."

[11] U. Julita, L. Lusianti F, R. Eka Putra, and A. Dana Perma, "Mating success and reproductive behavior of black soldier fly hermetia illucens l. (diptera, stratiomyidae) in tropics," *Journal of Entomology*, vol. 17, no. 3, pp. 117–127, 2020.

[12] I. J. Banks, W. T. Gibson, and M. M. Cameron, "Growth rates of black soldier fly larvae fed on fresh human faeces and their implication for improving sanitation," *Tropical medicine & international health : TM & IH*, vol. 19, no. 1, pp. 14–22, 2014.

[13] Eawag/Sandec, "Black soldier fly biowaste processing - a step-by-step guide," *Eawag – Swiss Federal Institute of Aquatic Science and Technology*, 2017.

[14] D. Purkayastha and S. Sarkar, "Sustainable waste management using black soldier fly larva: a review," *International Journal of Environmental Science and Technology*, 2021.

[15] M. K. Awasthi, T. Liu, S. K. Awasthi, Y. Duan, A. Pandey, and Z. Zhang, "Manure pretreatments with black soldier fly hermetia illucens l. (diptera: Stratiomyidae): A study to reduce pathogen content," *The Science of the total environment*, vol. 737, p. 139842, 2020.

[16] A. O. Anyega, N. K. Korir, D. Beesigamukama, G. J. Changeh, K. Nkoba, S. Subramanian, J. J. A. van Loon, M. Dicke, and C. M. Tanga, "Black soldier fly-composted organic fertilizer enhances growth, yield, and nutrient quality of three key vegetable crops in sub-saharan africa," *Frontiers in plant science*, vol. 12, p. 680312, 2021.

[17] D. Agustiyani, R. Agandi, Arinafril, A. A. Nugroho, and S. Antonius, "The effect of application of compost and frass from black soldier fly larvae (hermetia illucens l.) on growth of pakchoi (brassica rapa l.)," *IOP Conference Series: Earth and Environmental Science*, vol. 762, no. 1, p. 012036, 2021.
[18] R. Salomone, G. Saija, G. Mondello, A. Giannetto, S. Fasulo, and D. Savastano, "Environmental impact of food waste bioconversion by insects: Application of life cycle assessment to process using hermetia illucens," *Journal of Cleaner Production*, vol. 140, pp. 890–905, 2017.
[19] S. Ishak and A. Kamari, "Biodiesel from black soldier fly larvae grown on restaurant kitchen waste," *Environmental Chemistry Letters*, vol. 17, no. 2, pp. 1143–1150, 2019.
[20] L. Zheng, Y. Hou, W. Li, S. Yang, Q. Li, and Z. Yu, "Biodiesel production from rice straw and restaurant waste employing black soldier fly assisted by microbes," *Energy*, vol. 47, no. 1, pp. 225–229, 2012.
[21] Q. Li, L. Zheng, H. Cai, E. Garza, Z. Yu, and S. Zhou, "From organic waste to biodiesel: Black soldier fly, hermetia illucens, makes it feasible," *Fuel*, vol. 90, no. 4, pp. 1545–1548, 2011.
[22] L. Zheng, Q. Li, J. Zhang, and Z. Yu, "Double the biodiesel yield: Rearing black soldier fly larvae, hermetia illucens, on solid residual fraction of restaurant waste after grease extraction for biodiesel production," *Renewable Energy*, vol. 41, pp. 75–79, 2012.
[23] A. van Huis and D. G. A. B. Oonincx, "The environmental sustainability of insects as food and feed. a review," *Agronomy for Sustainable Development*, vol. 37, no. 5, 2017.
[24] J. Tomberlin and et al., "The black soldier fly, hermetia illucens, as a manure management/resource recovery tool," 2005.
[25] indexmundi.com, "Soybean meal - monthly price - commodity prices - price charts, data, and news - indexmundi," 16.05.2022. [Online]. Available: https://www.indexmundi.com/commodities/?commodity=soybean-meal&months=240
[26] C. D. Miranda, J. A. Cammack, and J. K. Tomberlin, "Mass production of the black soldier fly, hermetia illucens (l.), (diptera: Stratiomyidae) reared on three manure types," *Animals : an open access journal from MDPI*, vol. 10, no. 7, 2020.
[27] S. Smetana, E. Schmitt, and A. Mathys, "Sustainable use of hermetia illucens insect biomass for feed and food: Attributional and consequential life cycle assessment," *Resources, Conservation and Recycling*, vol. 144, pp. 285–296, 2019.
[28] B. Purschke, R. Scheibelberger, S. Axmann, A. Adler, and H. Jäger, "Impact of substrate contamination with mycotoxins, heavy metals and pesticides on the growth performance and composition of black soldier fly larvae (hermetia illucens) for use in the feed and food value chain," *Food additives & contaminants. Part A, Chemistry, analysis, control, exposure & risk assessment*, vol. 34, no. 8, pp. 1410–1420, 2017.
[29] A. Singh and K. Kumari, "An inclusive approach for organic waste treatment and valorisation using black soldier fly larvae: A review," *Journal of environmental management*, vol. 251, p. 109569, 2019.
[30] G. P. A. Gutierrez, R. A. V. Ruiz, and H. M. Velez, "Compositional, microbiological and protein digestibility analysis of larval meal of hermetia illucens (diptera:stratiomyiidae) at angelopolis-antioquia, colombia," *Revista Facultad Nacional de Agronomia Medellin*, vol. 57, no. 2, pp. 2491–2499, 2004.
[31] L. S. Queiroz, M. Regnard, F. Jessen, M. A. Mohammadifar, J. J. Sloth, H. O. Petersen, F. Ajallouеian, C. M. C. Brouzes, W. Fraihi, H. Fallquist, A. F. de Carvalho, and F. Casanova, "Physico-chemical and colloidal properties of protein extracted from black soldier fly (hermetia illucens) larvae," *International journal of biological macromolecules*, vol. 186, pp. 714–723, 2021.
[32] S. Ojha, S. Bußler, and O. K. Schlüter, "Food waste valorisation and circular economy concepts in insect production and processing," *Waste management (New York, N.Y.)*, vol. 118, pp. 600–609, 2020.
[33] S. A. Siddiqui, B. Ristow, T. Rahayu, N. S. Putra, N. Widya Yuwono, K. Nisa', B. Mategeko, S. Smetana, M. Saki, A. Nawaz, and A. Nagdalian, "Black soldier fly larvae (bsfl) and their affinity for organic waste processing," *Waste management (New York, N.Y.)*, vol. 140, pp. 1–13, 2022.
[34] R. Magalhães, A. Sánchez-López, R. S. Leal, S. Martínez-Llorens, A. Oliva-Teles, and H. Peres, "Black soldier fly (hermetia illucens) prepupae meal as a fish meal replacement in diets for european seabass (dicentrarchus labrax)," *Aquaculture*, vol. 476, pp. 79–85, 2017.
[35] V. C. Cummins, S. D. Rawles, K. R. Thompson, A. Velasquez, Y. Kobayashi, J. Hager, and C. D. Webster, "Evaluation of black soldier fly (hermetia illucens) larvae meal as partial or total replacement of marine fish meal in practical diets for pacific white shrimp (litopenaeus vannamei)," *Aquaculture*, vol. 473, pp. 337–344, 2017.

Deconstruction and Recycling of Wind Turbines

Philipp Gierl
Ostbayerische Technische Hochschule
Regensburg, Germany
philipp.gierl@st.oth-regensburg.de

***Abstract*—Wind energy plays a significant role in today's power generation. It makes a big contribution to protect the environment as it avoids carbon dioxide emissions. For example in Germany a big proportion of all power generated is wind energy and the first wind farm was taken into operation in 1987. The common lifetime of a wind turbine is only 20 years due to the massive force it is exposed to. Because of this it constantly gets more important to know how to handle an end-of-life wind turbine and the potential waste problem that comes with it. At a wind turbine's end-of-life, which is the end of its guaranteed operationability, there are three possibilities. If the wind turbine still works properly, it is possible to have a lifetime extension, even though the financial support by the government could be lost. It is also possible to renew or replace components of the wind turbine before restarting, which is called repowering. If neither a lifetime extension nor repowering is profitable, the wind turbine must be shut down completely and deconstructed.**
At this point there has to be made a distinction as offshore wind turbines are harder to dismantle in comparison to onshore wind turbines. The challenge in recycling the materials used in wind turbines mainly comes down to the composite materials used in the rotor blades. The three different procedures to recycle composite are mechanical, thermal, and chemical recycling. With these strategies the waste problem of wind energy can be reduced. This paper elaborates how to treat an End-of-life wind turbine. The techniques behind the deconstruction and recycling of wind turbines are explained as well as alternatives to it with the procedures of repowering and lifetime extension. The legal frame of this topic is shown by the situation in Germany as an example.

***Index Terms*—wind turbine, end-of-life, lifetime extension, repowering, onshore, offshore, composite**

I. Introduction

If you look at the entire energy consumption in Germany, renewable energies were significant in the year 2021 by having a contribution of 19.7%. This even exceeds the goal of 18% which was set by the European union. In absolute numbers, renewable energies made a total of 467 TWh of energy, which can be split in electricity (50%), heat (43%) and fuel (7%). Looking at the electricity sector, renewable energies make up 41.1% of all electricity used in Germany in 2021. After biomass, which is 55% of all renewable energy in Germany, the biggest contribution is wind energy with 24%. [1]
Wind energy has started to become relevant in Germany in the early 1990s. In the year 1990 onshore wind energy produced a total of 72 GWh. Since then that number rose to 9,703 GWh in 2000, 38,371 GWh in 2010 and 104,796 GWh in 2020. Offshore wind energy started to become relevant in Germany in about 2010. It produced 176 GWh in 2010 and 27,306 GWh in 2020. Even though both the numbers for on- and offshore produced wind energy dropped in 2021 (89,474 GWh onshore and 24,374 GWh offshore) it is clear to see that renewable energies and especially wind energy is significant by looking at Germany as an example. [2]

The downside of wind energy is the waste material, for example from the rotor blades of wind turbines. The common lifetime of a wind turbine is about 20 years. Because there were already some wind turbines in 1990, some of them have already gotten to their End-of-life, as you can see below in figure 1. As the amount of wind turbines increased since 1990, the amount of potential waste increases as well. [3]

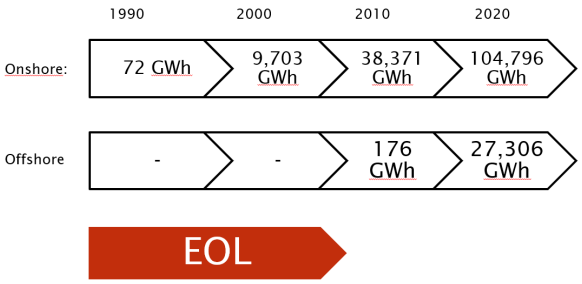

Fig. 1. Development of wind energy in Germany

A forecast about the potential amount of blade waste material for the year 2050 concludes, that there will be a worldwide amount 325 kt of waste material from the blades of wind turbines alone. 76% of that number comes from onshore wind farms and 24% from offshore wind farms. In Germany alone the forecast predicts a total amount of about 67 kt of blade waste for the year 2050. [3]
These numbers show impressively that there has to be a solution for the waste problem that comes with End-of-life wind turbines. The procedure and possibilities of what to do with a wind turbine that has reached its lifetime will be elaborated in the following.

II. Legal frame of the lifetime of wind turbines in Germany

In Germany renewable energy, and thus wind energy, gets additional support by the government as part of the

Renewable Energy Act (EEG). Since April 2000, when it was firstly installed, the EEG has been an important factor for renewable energy in Germany and has been adjusted many times since then. Energy companies receive financial support for renewable energy, which is called the EEG-funding. The concept of the EEG-funding is that companies which own a source of power for renewable energy get a certain amount of money for every kilowatt-hour of energy they produce. The duration of this funding is 20 years. Once connected the amount of money you get per kilowatt-hour stays the same. The sum paid relies on the technology of renewable energy and the place where it is installed. Since 2017 there is an important regulation on how the volume of funding for wind energy is determined. There is no longer a certain amount fixed in the EEG, but now the prices are set at auctions by bidding. [4]

Once the EEG-funding for a wind turbine is over a decision must be made on how to continue with it. The three options are extending its lifetime without the funding, repowering it by exchanging old parts and qualifying for funding again or completely shutting it down if both other options are not profitable. [5]

If a wind turbine has reached its End-of-life and is to be shut down, there is no duty set by the law to dismantle the wind turbine. The obligation to deconstruct the wind turbine is set by voluntary commitments, which usually must be made to build it. Most federal states in Germany (except North Rhine-Westphalia) can order the deconstruction of a wind turbine under the state's building codes. If the dismantling of a wind turbine takes place because of a voluntary commitment, there is a different extent of dismantling the plant than there is if it is dismantled because of a federal state's building codes. In the case of a voluntary commitment, every part of the wind turbine must be deconstructed. This includes for example the tower and the turbine. The foundation at least must be removed to a certain depth, if not completely. In case of a removal order by the building codes of a federal state, the government determines the extent of the dismantling. [6]

III. Lifetime extension and Repowering of wind turbines

If the lifetime of a wind turbine has ended, but it is still able to operate properly, a lifetime extension can be implemented [5]. Since the EEG-funding ends along with the wind turbine's life time, there must be a rethinking on how to sell the produced energy [5]. One possibility is to sell the energy in the European Energy Exchange (EEX) [5]. The EEX is an exchange for example for power and gas [7]. Another possibility is a Power purchase agreement (PPA) with industrial companies [5]. Here a contract is made between the wind turbine's owner and the buyer often for up to 20 years [8].

Another possibility is the repowering of an end-of-life wind turbine and regaining the EEG-funding by participating in the auctions which were established in 2017 [5]. Repowering means dismantling the old wind turbine and replacing it with a new one at the same place or at least near the same place. Since the new turbines often have a higher capacity of generating power, there can be more power generated after repowering. An advantage of repowering is, that the first generation of wind turbines were placed at the positions with the best conditions for generating power with wind. By repowering wind turbines at these places, you can generate even more wind energy. Another advantage is, that many parts of the old turbine can be reused, for example the electrical connections and the substation. [9]

Studies show that by repowering there can be generated up to four times more power by repowering. In Germany and Denmark for example, repowering is a big factor. In Germany before 2018, about 2900 old turbines were replaced. Through this about 2.3 GW of wind power capacity of the old turbines were replaced by about 5.5 GW of the new turbines. The repowering of offshore wind farms is not already as common as repowering of onshore wind farms. [9]

However the repowering of offshore wind turbines is especially beneficial. If you would dismantle a wind turbine completely including its foundations, it would be very costly and come with great damage to the sea life surrounding the wind farm. But since the foundations often have a lifetime of 100 years and the submarine cables of 40 years they can easily be reused. [10]

An offshore wind farm has been repowered for the first time in 2018 near the island of Gotland, Sweden. New blades and control systems were installed as well as a new housing for the generating elements. By reusing the towers, the foundations, and the cables they took full advantage of repowering an offshore wind farm. The wind turbines now have an additional 15 years lifetime and generate with 11 GWh per year more than double the energy than before. [11]

IV. Dismantling of wind turbines

Because there is a big difference in the environment in which the dismantling takes place, there will be a distinction between the dismantling of onshore wind turbines and offshore wind turbines.

A. Onshore wind turbines

When a wind turbine is dismantled firstly the whole system must be shut down and de-energised. The dismantling consists of the following tasks. The inner components of the turbine must be removed as well as the transformer. Then the turbine itself can be deconstructed including the blades, the nacelle, and the tower. The foundations must be removed and the resulting whole must be refilled. You must remove the cables and deconstruct the substation and buildings which belonged to the wind farm. If there are tracks to get to the wind farm you must remove them as well. [12]

You dismantle the turbine's components, including the blades, the bladehub and nose cone, the nacelle and the tower, with a crane from where you can deconstruct it. [12]

To remove the concrete base of a wind farm you need a hydraulic breaker or even explosive charges as well as steel burning tools to get through the steel reinforcements in the concrete. After that the concrete pile is removed. You refill the resulting whole with crushed rock. The transformer and its concrete foundation is dismantled with a hydraulic breaker. The tracks are removed with an excavator as well as the cables.[12]

B. Offshore wind turbines

The dismantling of offshore wind turbines is still quite new. The methods and vessels are comparable to those used to dismantle an oil factory. The special boats are particularly good at heavy lifting. [13]
Removing the turbine's compontents is quite similar to the removing of turbines of onshore wind farms except specialized vessels which can carry a lot of weight are used. Before dismantling you must remove all dangerous substances and liquids from the turbine, for example motor oil. You use angle grinders and plasma cutters to remove the bolts if necessary. Cables between the different components must be cut. [13]
The transition piece connects the tower with the base. The cables in between are removed and divided. Once a crane is ready, you can start cutting the connections to the foundation and then lift the transition piece up. You can also lift the transition piece and the foundation all together. But you must be able to lift more weight. Also more safety measures are then required. [13]
Before removing the foundation, you must decide whether to remove the whole base or only a part of it. Usually the favoured method of these two is only removing the top part, as it is less cost-intensive and less dangerous. The resulting hole has to be refilled. The techniques required vary depending on the type of base. But in any case you must observe the base first using underwater drones or divers. [13]

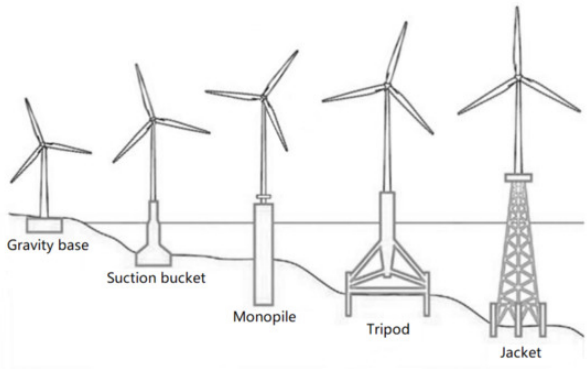

Fig. 2. Different bases for wind turbines [14]

Figure 2 shows the different foundations for offshore wind turbines. The gravity base can be used either onshore or offshore. As tripod and jacket foundations are similar, only the removal of the jacket foundation is explained.

The first foundation discussed is the monopile foundation. Here with a sea trencher, a special excavating tool, the foundation is loosened. A crane is positioned on the specialized vessels. Pieces of the foundation are cut. The depth of removing and the weight make a complete removal dangerous for both the workers as well as the environment. Diamond cutting and water jetting are the techniques with which the foundation is cut. Then you must load the foundation on the vessel and ship it away. [13]
Gravity foundations use ballast to keep it on the ground [15]. This ballast can be gravel, sand, or stone [15]. Here you can leave the base and only the tubular piece can be removed. You can also remove the base completely. If necessary, lifting tools can be placed. The ballast to keep the foundation down must be removed. This can be done with the technology suction dredging, which must be observed by divers or drones. Under the base, compacted sediments must be loosened up. After that the foundation can be lifted. [13]
A jacket foundation consists of four legs which are connected with each other with several braces [15]. After you cut the legs using diamond cutters, you can lift the whole jacket foundation up. Perhaps you need to dig into the ground to reach the spot where you need to start cutting. [13]
The last discussed foundation uses suction buckets to keep the wind turbine to the ground. To dismantle it you must pump it back up. Dismantling these types of foundation causes less damage to the environment because the whole base is removed and there is no need for cutting and excavating. [13]
The cables can also be either removed or left below the seabed, because excavating to remove the cables has a big environmental impact and is very const-intensive. If a only a part of the cables is left, they are cut at the corresponding place and dug back in. If trenches result in this process, the tides will fill them up. [13]

V. Recycling of wind turbines

The main materials a wind turbine consists of are steel, composite materials, iron, copper and aluminium. Steel makes up about 66 to 79% of the wind turbine, composite materials 11 to 16%, iron 5 to 17%, copper 1% and aluminium 0 to 2%. [16]
When steel is recycled it is melted down in a furnace. While the steel is liquid, superfluous carbon and nitrogen are removed. After further refining the steel is cast to bars or slabs. [17]
Recycling of iron also consists of melting it down and processing it. This process of recycling can be done repeatedly. [18]
The recycling of copper consists of melting it down first and casting the copper in the final shape or suitable for further manufacturing. If the copper is polluted, it is electrolytically purified. If the copper is heavily contaminated, the recycled copper is improbable to reach the standards of high-quality-copper. [19]
The recycling of aluminium requires a distinguishing

of several types of waste aluminium. Unspecified and contaminated scrap aluminium is recycled to cast alloys. Slightly contaminated aluminium is recycled to wrought alloys. Uncontaminated aluminium is melted down and further processed. [20]

Recycling concrete includes shredding the concrete and reusing the material as granules in new concrete [21]. Sand and gravel in the foundations of wind turbines stay at the site and do not undergo recycling [22].

The fiber-reinforced composite materials used in the blades are difficult to recycle. Landfilling the end-of-life blades is not a viable option because of environmental reasons. Also, the valuable material is lost. Recycling the composite materials comes down to thermal, mechanical and chemical recycling. [23]

Thermal recycling can be thermal degradation or thermoforming. Thermal degradation regains the fibers by pyrolysis. As temperatures over 450 °C are used, glass fibers can be damaged in the process. For example, the regained fibers can be utilized for cement. Burning the composite material whole can for example be used to supply cement furnaces. Thermoforming heats the composite material up so it can be reshaped. Once the material is cooled, it stays in its shape. Even though it is not common for big components as wind turbine blades, they could be cut and straightened with this method. The material would be useful as construction material. [23]

Mechanical recycling is done by mechanical grinding. The fibers are separated by a cyclone for example. A possible use for the fibers would be insulating foams for thermal or acoustic use. The regrind can be used for injection molding. But the cutting of the composite materials can damage the fibers and lessen their quality and their length. [23]

Chemical recycling of composite material is solvolysis or dissolution [23]. Here the fibers do not reduce in length while being recovered [23]. For solvolysis a temperature of 200 to 370 °C and a pressure of 100 to 250 bar are required [24]. Water or alcohol can be used as solvent [24]. Hydrolysis reactions separate the plastic material into smaller pieces, so the fibers are loose and can be recovered [24]. But due to the high temperature, the quality of the fibers can reduce [23]. If thermoplastic resin material was used for the blades, it is possible to dissolve at low temperatures so the fibers are not damaged [23].

VI. Conclusion

This literature research shows that the waste problem of wind turbines can be highly reduced. If the wind turbine still works at its end-of-life and there are other ways of selling the energy without governmental funding, a wind turbine's lifetime can be extended. If components need to be replaced as the wind turbine does not qualify for further operation, parts can be replaced and a repowering of the wind turbine can take place. But even if these are not possible, the deconstruction of the wind turbine leads to recycling the materials and reducing the produced waste. So even though wind energy is already one of the most environmentally friendly energies, by evaluating all options at the end-of-life of a wind turbine and recycling properly, wind energy can be made even more ecologically sound.

References

[1] 'Erneuerbare Energien in Zahlen', Umweltbundesamt. https://www.umweltbundesamt.de/themen/klima-energie/erneuerbare-energien/erneuerbare-energien-in-zahlen#waerme (accessed May 16, 2022).

[2] 'Zeitreihen zur Entwicklung der erneuerbaren Energien in Deutschland' , Bundesministerium für Wirtschaft und Klimaschutz. https://www.erneuerbare-energien.de/EE/Navigation/DE/Service/Erneuerbare_Energien_in_Zahlen/Zeitreihen/zeitreihen.html (accessed May 16, 2022).

[3] G. Lichtenegger, A. A. Rentizelas, N. Trivyza, and S. Siegl, 'Offshore and onshore wind turbine blade waste material forecast at a regional level in Europe until 2050', Waste Management, vol. 106, pp. 120–131, Apr. 2020, doi: 10.1016/j.wasman.2020.03.018.

[4] 'Renewable Energy Act (EEG)', Bundesverband Windenergie. https://www.wind-energie.de/english/policy/rea/ (accessed May 16, 2022).

[5] J. Piel, C. Stetter, M. Heumann, M. Westbomke, and M. H. Breitner, 'Lifetime Extension, Repowering or Decommissioning? Decision Support for Operators of Ageing Wind Turbines', J. Phys.: Conf. Ser., vol. 1222, no. 1, p. 012033, May 2019, doi: 10.1088/1742-6596/1222/1/012033.

[6] 'Breaking & Sifting Expert exchange on the end-of-life of wind turbines', Fachagentur Windenergie. https://www.fachagentur-windenergie.de/fileadmin/files/Veroeffentlichungen/FA-Wind_Breaking_Sifting_englisch.pdf (accessed May 16, 2022).

[7] 'EEX – European Energy Exchange', Europex. https://www.europex.org/members/eex/ (accessed May 16, 2022).

[8] A. F. Huneke, S. Göß, J. Österreicher, and O. Dahroug, 'Power Purchase Agreements: Finanzierungsmodell von erneuerbaren Energien', Energy Brainpool. https://www.energybrainpool.com/fileadmin/download/Whitepapers/2018-01-31_Energy-Brainpool_White-Paper_Power-Purchase-Agreements.pdf (accessed May 16, 2022).

[9] R. Lacal-Arántegui, A. Uihlein, and J. M. Yusta, 'Technology effects in repowering wind turbines', Wind Energy, vol. 23, no. 3, pp. 660–675, Mar. 2020, doi: 10.1002/we.2450.

[10] Y. Liu, Y. Fu, L. Huang, and K. Zhang, 'Reborn and upgrading: Optimum repowering planning for offshore wind farms', Energy Reports, vol. 8, pp. 5204–5214, Nov. 2022, doi: 10.1016/j.egyr.2022.04.002.

[11] J. Gerdes, 'World's First Offshore Wind Repowering Completed in Sweden — Greentech Media'. https://www.greentechmedia.com/articles/read/worlds-first-offshore-wind-repowering-completed-in-sweden (accessed May 16, 2022).

[12] K. Taylor, 'Research and guidance on restoration and decommissioning of onshore wind farms', Scottish National Heritage. https://www.nature.scot/sites/default/files/2017-07/Publication%202013%20-%20SNH%20Commissioned%20Report%20591%20-%20Research%20and%20guidance%20on%20restoration%20and%20decommissioning%20of%20onshore%20wind%20farms.pdf (accessed May 16, 2022).

[13] E. Topham and D. McMillan, 'Sustainable decommissioning of an offshore wind farm', Renewable Energy, vol. 102, pp. 470–480, Mar. 2017, doi: 10.1016/j.renene.2016.10.066.

[14] M. Xie and S. Lopez-Querol, 'Numerical Simulations of the Monotonic and Cyclic Behaviour of Offshore Wind Turbine Monopile Foundations in Clayey Soils', Journal of Marine Science and Engineering, vol. 9, no. 9, Art. no. 9, Sep. 2021, doi: 10.3390/jmse9091036.

[15] M. Keene, 'Comparing offshore wind turbine foundations', Windpower Engineering & Development. https://www.windpowerengineering.com/comparing-offshore-wind-turbine-foundations/ (accessed May 16, 2022).

[16] C. Mone, M. Hand, M. Bolinger, J. Rand, D. Heimiller, and Jonathan Ho, '2015 Cost of Wind Energy Review', National Renewable Energy Laboratory, 2017. https://www.nrel.gov/docs/fy17osti/66861.pdf (accessed May 16, 2022).

[17] ‘Recycling von Stahl und Edelstahl - Vorteile und Methoden’, Montanstahl. urlhttps://www.montanstahl.com/de/magazin/einfaches-recycling-von-stahl/ (accessed May 16, 2022).
[18] ‘Eisen: Nichts ist besser zu recyceln’, DIE WELT, Jun. 21, 2009. Accessed: May 16, 2022. [Online]. Available: https://www.welt.de/wams_print/article3965625/Eisen-Nichts-ist-besser-zu-recyceln.html
[19] ‘Recycling of Copper’, Copper Development Association Inc. https://www.copper.org/environment/lifecycle/ukrecyc.html (accessed May 16, 2022).
[20] L. Rau, ‘Aluminiumrecycling: So funktioniert es - Utopia.de’, utopia. https://utopia.de/ratgeber/aluminiumrecycling-so-funktioniert-es/ (accessed May 16, 2022).
[21] M. Janning, ‘Beton: Recycling von Beton’, planet wissen, Sep. 26, 2018. https://www.planet-wissen.de/technik/werkstoffe/beton_der_formbare_stein/beton-baustoff-100.html (accessed May 16, 2022).
[22] P. D. Andersen, A. Bonou, J. Beauson, and P. Brøndsted, ‘Recycling of wind turbines’, in DTU International Energy Report 2014, H. Hvidtfeldt Larsen and L. Sønderberg Petersen, Eds. Technical University of Denmark, 2014, pp. 91–97.
[23] D. S. Cousins, Y. Suzuki, R. E. Murray, J. R. Samaniuk, and A. P. Stebner, ‘Recycling glass fiber thermoplastic composites from wind turbine blades’, Journal of Cleaner Production, vol. 209, pp. 1252–1263, Feb. 2019, doi: 10.1016/j.jclepro.2018.10.286.
[24] C. Mattsson, A. André, M. Juntikka, T. Tränkle, and R. Sott, ‘Chemical recycling of End-of-Life wind turbine blades by solvolysis/HTL’, IOP Conf. Ser.: Mater. Sci. Eng., vol. 942, no. 1, p. 012013, Oct. 2020, doi: 10.1088/1757-899X/942/1/012013.

Comparison between Microcontroller and FPGA: Advantages and Suitable Fields of Application

Veronika Rappl
Faculty of Electrical Engineering and Information Technology
Ostbayerische Technische Hochschule
Regensburg, Germany
veronika1.rappl@st.oth-regensburg.de

***Abstract*—Field Programmable Gate Arrays (FPGA) and Microcontrollers (MC) are two important representatives for the realization of digital circuits. For a long time, MCs were the dominant component in embedded systems due to their easily programmable functions, however, the importance of FPGAs and their operation in many applications has increased. While developing a new product, a decision between the usage of an FPGA or a MC is required. FPGAs are power efficient, very fast, reprogrammable, flexible, can implement parallel processes in a small hardware area and have become less expensive over the years. FPGAs are used in applications such as image and signal processing, real-time applications or embedded intelligence. Compared to FPGAs, the various components of a MC are fixed in their configuration. This makes them very cost-effective, easy to troubleshoot and connectable to other components. As a result, MCs are used in computing technology or control engineering. In order to be able to make a decision between these two components, this work presents a systematic comparison between FPGAs and MCs. The architecture, operation and programming of an FPGA are described and are compared to those of a MC. The application areas of the respective components are defined and the advantages and disadvantages of each component can be shown. Furthermore, the System-on-Chip (SoC) FPGA technology is presented, which integrates the processor and FPGA architecture in one device. FPGAs, MCs and SoC FPGAs are used in different fields of application, which overlap, so the decision to use a specific component depends on the requirements of the system.**

***Index Terms*—field programmable gate array, microcontroller, embedded systems, hardware implementation, programmable architectures, reconfigurable architectures, parallel architectures, programmable circuits, system on chip**

I. Introduction

Programmable components represent powerful tools for a developer of a working system. The choice of the right processing system to perform computational tasks depends on the requirements of the system and is not always clear. Two important representatives for the realization of technical systems are Field Programmable Gate Array (FPGA) and Microcontroller (MC). FPGAs are reconfigurable computer chips with hardware that is easily programmable (field programmable) [1]. MCs represent integrated computing units that can be customized for various applications [2]. The use of these two components extends over various fields of application. There are also numerous papers in the literature dealing with the two components. A. Boutros and V. Betz [1] discuss the basics and the progress of the architecture of an FPGA in their paper. The authors in [3]–[5] precisely describe the components of an FPGA in their work. The structure of a MC is discussed in [2] [6]. Furthermore, in [7]–[9], a direct comparison between the usage of an FPGA and a MC is made. In their work, V. Bianchi et al. [7] present an innovative method for error size evaluation in Reed-Solomon codes. Reed-Solomon codes are used in channel coding to detect and correct transmission errors as part of forward error correction. T. Kahl and S. Dieckerhoff [8] use the two components to realize the control of a high-dynamic power electronic converter. They concluded that MCs would allow a faster regulation for this application. FPGAs, on the other hand, enable a faster response time in this application. F. Ortega-Zamorano et al. [9] implement the backpropagation algorithm on both programming devices. The computational technology of a MC performs better in this application, however, the authors show the potential of an FPGA in their paper.

The choice of the right component depends on different requirements. This paper therefore presents a systematic comparison between an FPGA and a MC. The paper is structured as follows: In section II, the architecture and programming of an FPGA are explained in detail. The architecture and programming of a MC are described in section III. The advantages and disadvantages of each component are presented in section IV. Then, in section V, several application areas of the two components are presented. The System-on-Chip (SoC) FPGA technology is explained in section VI. This method combines the respective advantages of the previously described components. A final conclusion is given in section VII.

II. Field Programmable Gate Array

The structure and programming of a standard FPGA are described in this section.

A. Architecture

1) Configurable Logic Blocks: The Configurable Logic Blocks (CLB) are distributed over the entire structure (see Figure 1), in which the functionality of the FPGA is implemented [10]. They are composed of one or more basic logic elements (BLE). These in turn consist mainly of a Lookup Table (LUT) and several flip-flops as well as multiplexers. A LUT is implemented as a truth table, which selects an output from 2^k values when the number of input signals is k. [1]

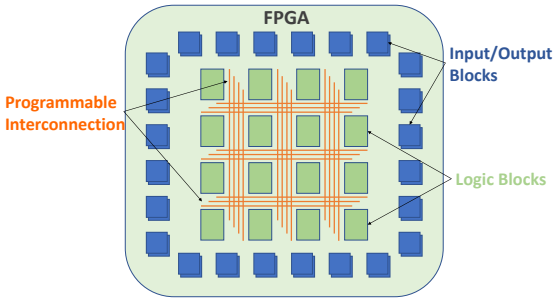

Fig. 1. Generic architecture of an FPGA (adapted from [11])

2) Programmable Input/Output Blocks: The programmable Input/Output Blocks (I/O Blocks) of the FPGA represent an interface to various devices and allow communication with them (see Figure 1). The I/O Blocks are grouped in various banks and can be equipped with different power supplies. [1]

3) Programmable Interconnections: The connections between the CLBs and to the I/O Blocks are ensured by programmable connections. Various routing architectures exist for this purpose, whereby the resources can be utilized optimally. A distinction is made between hierarchical and island architectures. In the hierarchical structure the logic blocks are divided into different clusters, which are repeated in the architecture. [10] Communication usually takes place between clusters that are close to each other. Thus, short connections can be realized [1]. In the island-like architecture, the I/O Blocks are distributed on the outside, the CLBs inside the FPGA. The routing connections are arranged vertically and horizontally regularly within the FPGA. This architecture is very flexible and adaptable. [10] The island architecture is mainly used in novel FPGAs (see Figure 1).

4) Programming Technologies: There are different programming technologies. The most important technologies are Static Random-Access Memory (SRAM), Flash and Antifuse, whereby SRAM technology is mostly applied. In SRAM technology, the configuration of the CLB and the connections are stored in SRAM cells. It allows the values of the cells, and the configuration, to be changed. Since this technology is volatile, non-volatile memory or external source is needed for power-up. This makes SRAM-based FPGAs very expensive. Flash technology utilizes flash memory, which preserves the configuration after power is turned off. This results in lower power consumption. Flash-based FPGAs are reconfigurable, but cannot be reprogrammed indefinitely. In antifuse technology, a metal-to-metal fuse breaks and becomes conductive by applying a voltage. This prevents the FPGA from being reprogrammed. The advantage is low area consumption and low delay in the routing process. [12]–[14]

5) Other components: Delays may occur during clock distribution in the FPGA. These are compensated by using Phase-Locked Loops and Delay-Locked Loops. They also offer other functions such as frequency multiplication and division, phase shift correction and duty cycle correction. [3] To improve functionality, other components such as multipliers, adders, accumulators and in advanced FPGAs Digital Signal Processors are integrated [3] [13]. Communication blocks with transmit and receive buffers supporting various communication protocols such as Ethernet are built in, making communication easier. [1]. By using internal memory blocks such as Random-Access Memory (RAM) and Read-Only Memory (ROM) the processing speed can be increased [3].

B. Programming

Early FPGAs consisting of glue logic were described using CAD schematics [4]. Today, Hardware Description Language (HDL) such as Very High Speed Integrated Circuit Hardware Description Language (VHDL) or Verilog are used to program an FPGA [1]. In a first step, a circuit design is made in VHDL. The design is rechecked in a simulation for possible errors, which can be corrected. Afterwards, the synthesis can be performed, where the description is converted into a netlist. The physical implementation includes assigning the I/O blocks, placing and routing the connections, and the generation of the bitstream file used to configure the SRAM cells [1] [13]. To reduce design time, FPGA manufacturers offer tools that utilize languages such as C/C++, MATLAB-Simulink, LabView, or OpenCL to program an FPGA. However, the use of higher level languages degrades performance. [15]

III. Microcontroller

A MC is an integrated circuit containing a Central Processing Unit (CPU) and other components (see Figure 2) [2]. This section covers the basic structure of a standard MC.

A. Architecture

1) Central Processing Unit: The CPU works with data formats in the range of 8 to 64 bit. In today's MC, mainly 32 bits are used. It consists of an arithmetic-logic unit, a control unit, internal memory locations and several registers. In the CPU, the instructions are executed sequentially and synchronously. [5] [6] The architecture-level power optimization of MCs is a major issue and is described in more detail in [16].

2) Memory: MCs contain an integrated memory where data and programs can be saved [6]. Data is stored in a volatile memory, such as RAM. In previous MCs, Erasable Programmable ROMs were used for storing the program. It takes a lot of time and effort to erase them, so flash technology is used as program memory in today's MCs. [2] [17].

3) Input/Output Ports and Internal Data Buses: I/O ports are used as digital interfaces to connect external devices. By controlling and detecting the logical state, data can be exchanged. The internal data bus is responsible for connecting the I/O ports to the CPU and for data exchange [2] [6].

4) Timer, Watchdog-Timer and Interrupt: The oscillator acts as the core clock of the MC and the internal processes. In addition, MCs contain several timers, which have a bit width of 16 to 32 bits. These units are used for simple counting or more complex tasks such as pulse width modulation. By using interrupts, MCs can react fast and flexible. The program flow is thereby interrupted and different instructions can be executed in a routine with a certain priority. [6]

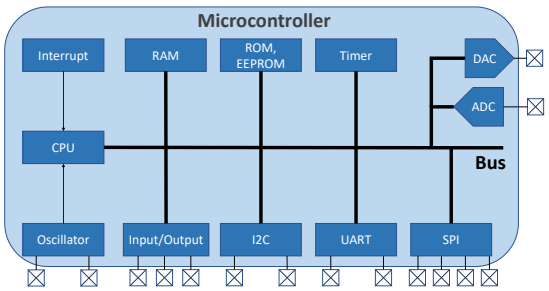

Fig. 2. Architecture of a MC with main blocks (adapted from [5])

5) Communication Interfaces and Converters: Communication interfaces such as universal asynchronous receiver and transmitter (UART) and serial communication interfaces (inter-integrated circuit (I2C), serial peripheral interface (SPI) and controller area network (CAN)) are used for system connection [2]. Analog-to-digital converters (ADC) and, more rarely, digital-to-analog converters (DAC) are used for converting among digital and analog values [6].

B. Programming

Originally MCs were programmed in assembler. Today, high-level compilers in C/C++ or similar languages are mainly used to program a MC [5] [17]. Modern MCs are provided with a bootloader code and can be easily programmed via USB port [17]. The written code is compiled and converted to an executable file using a linker and transferred to the MC. The utilization of a debugger facilitates troubleshooting. [6]

IV. Advantages and Disadvantages

FPGAs and MCs possess several benefits and drawbacks, presented in this section. They relate to the basic properties of the two components.

A. Field Programmable Gate Array

FPGAs are flexible due to their architecture, reliable and can be reconfigured [1] [4] [9] [10]. They can be adapted to specific requirements and power consumption can be reduced [4] [14] [18]. The parallelism of an FPGA in a small area makes them very fast in their processing [4] [13]. In addition, they have a high computing power [14] [18]. Due to the dedicated hardware, predictable latencies are achieved [4] [5]. However, there are also some disadvantages. Performing floating point operations with an FPGA are resource intensive and complex to implement [3]. FPGAs itself are resource intensive, however some optimization techniques exist [4] [18]. Another problem with the use of multiple FPGAs is the limited communication between the individual blocks [18]. FPGAs are expensive compared to MCs for operations not as compute-intensive [13] [19]. In this case, only individual parts are active, leaving the components largely unused. Appropriate solutions for the optimization of resources are described in [4]. Another disadvantage is the extensive circuit design for an FPGA, especially different power supplies and the loading technique for an SRAM-based FPGA for power up [14].

B. Microcontroller

MCs can be integrated with their components on a small chip area [5] [9] [19]. This makes them cost-effective, energy-saving and communication between the individual components is ensured [5] [13]. They are very compact and flexible [5]. With the help of standard programming languages such as C, C++ and Java, MCs are easy to program [9] [17]. By using a MC, it is possible to update and reconfigure the firmware of a device [5].
MCs possess some disadvantages. The biggest disadvantage is the slow computing speed due to the sequential execution [9] [18] [20]. Therefore, it is not possible to exploit parallelism (without using the entire component), which limits performance in terms of bandwidth. Multicore MCs are available, which offer higher computational speed and parallelism [13]. The high power consumption in some applications, such as on the Internet of Things (IoT), is another drawback [18] [20]. To counteract this, the authors in [20] present some optimization techniques to reduce power consumption and suitable MCs such as MSP430.

V. Applications

FPGAs and MCs are applied in various fields of applications. Possible application areas of an FPGA are summarized in Table I. FPGAs are used when complex calculations are required. In the areas of real-time signal processing, neural networks and image and signal processing, the parallelism of an FPGA is utilized [21]–[25]. If the system has to be reconfigured at run times, the dynamic reconfiguration of an FPGA is applied [14]. Table II shows several applications of a MC. It is used when a low-cost solution is necessary that requires little computational effort [26] [27]. MCs are suitable for small integrated solutions [28]. MCs are the appropriate choice for the application of control systems [8].

VI. System-on-Chip FPGA

The System-on-Chip FPGA (SoC FPGA) technology integrates a processor and an FPGA in one device and thus combines the two components (see Figure 3). This reduces production costs and saves space on the circuit board [19]. Other components of a SoC FPGA are analog peripherals such as ADCs, memory units such as cache memory and RAM, floating point units or network-on-chip [1] [13] [19]. By using SoC FPGAs, only one internal bus is required for communication between the CPU and the FPGA, reducing power consumption and increasing bandwidth [19]. SoC FPGAs are used in real-time simulation [15], advanced control techniques [15], robotics [44], or as edge devices [19].
SoC FPGAs have been used rather infrequently. Good knowledge of hardware and software is required to use this technology. Besides, developers concentrate on optimizing their existing platforms for specific applications and do not use new devices. [45] Due to the usage of FPGAs and SoC FPGAs in complex systems that process sensitive data, the security and encryption of an FPGA gains an increasing importance. Several approaches to this are discussed in [46].

TABLE I
A SURVEY OF FIELDS OF APPLICATIONS OF AN FPGA

Field of application	Reference	Used Component	FPGA requirement	Summary of the paper
Image and signal processing	[25]	Xilinx XCV2000E	parallel processing	Fast Hadamard Transformation (FHT)
Computer vision applications	[29]	Virtex-5 Xilinx	hardware close, parallelism	Face recognition system
Wireless Telecommunications	[30]	Xilinx Virtex-I1 XC2V8000	partial, parallel architecture	Parity check for error detection
Embedded Intelligence	[18]	—	parallelism, energy-efficient	Survey of various applications
Fuzzy logic	[31]	Xinlinx Spartan 2E	reprogrammable, fast	Speed control of electric vehicles
Industrial electrical controls	[13], [32]	—	parallelism	Survey of industrial controls
Data Acquisition Devices	[33]	81 FPGAs in 3 layers	customizable, flexible	Hierarchical trigger system
Automotive	[34]	—	low development costs	Possibilities in automotive applications
Aerospace Engineering	[35]	—	radiation resistance, flexible, reconfiguration	FPGA-based on-board processor
Machine Learning	[36]	Xilinx Virtex-7	low-cost, high flexibility	Accelerator for large-scale machine learning
Medical devices	[37]	CPLD 7064 chip	flexibility of reconfiguration	Digital wheelchair control system
Neural networks	[21]	—	parallel structure	Multilayer neural network
	[22]	Xilinx XC V50hq240	reconfigurable	Single neuron implementation
Robotics, real-time applications	[23]	EPF6024ACT144-3	reconfigurable	Autonomous vehicle-like mobile robot
	[24]	VirtexE XCV3200E	parallelism	Path planning of mobile robots
Control of electrical machines	[38]	Spartan XCS401XL	parallelism	Methods for the control of induction motors
Control of power converters	[39]	Altera ACEX-EP1K	accurate analog architecture	Peak current control

TABLE II
A SURVEY OF FIELDS OF APPLICATIONS OF A MC

Field of application	Reference	Used Component	MC requirement	Summary of the paper
Biology	[40]	Arduino	low-cost, real time	Temperature control of single cells
Biology, Medicine, Agriculture, Industry	[5]	—	low-cost, wearability, small sizes	Survey of few applications
Wireless Sensor Networks (WSN)	[41]	—	—	Literature study on design of a WSN
Real-time application	[11]	—	low-cost, communication interface	Real-time algorithm for a wind tunnel
Fuzzy logic	[27]	SH7047	simple, low-cost	Active rectifier without current sensors
Neural networks	[26]	8 bit	simple, low-cost	Linearization of nonlinear properties of sensors
	[11]	8 bit	low-cost	Determination of position and orientation of robot's end effectors
Parameter estimation	[42]	NEC uPD78F0058	low-cost, communication interface	Estimation of the state of charge of sealed lead-acid batteries
Sensorless power tool	[43]	TM4C123XXX	low-cost	Sensorless circuit approach
Robotics	[11]	—	soft real-time	Sensor interface for mobile robot platform
Wearable computing	[28]	—	low space requirement	Wearable input device for spatial input via hand movement

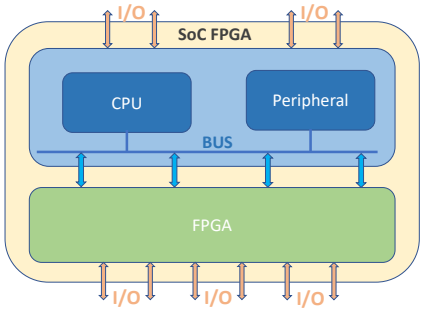

Fig. 3. Block diagram of an SoC FPGA (adapted from [47])

VII. CONCLUSION

The aim of this paper is to present a categorical comparison between a MC and an FPGA in order to make a decision between the these two components for an application. Due to the architecture of an FPGA, they are very flexible, reconfigurable and the parallelism of an application can be used. Therefore, FPGAs are utilized for complex and fast calculations. FPGAs are very resource intensive, especially when performing floating point operations, making them expensive for simple applications. MC are compact, flexible and energy-saving. They are used for small, low-cost applications that do not require high computing power. SoC FPGAs combine the advantages of both components. By integrating both components in one device, power consumption is reduced and communication bandwidth is increased. SoC FPGAs are currently found in many application areas, but have been used rather infrequently. FPGAs, MCs and SoC FPGAs are all suitable solutions. This paper has given a survey of the components and their suitable fields of application. However, the decision must always be made individually in the application case, depending on the requirements of the system.

REFERENCES

[1] A. Boutros and V. Betz, "FPGA Architecture: Principles and Progression," in IEEE Circuits Syst. Mag., vol. 21, no. 2, pp. 4-29, May 2021, doi: 10.1109/MCAS.2021.3071607.

[2] M.K. Parai, B. Das, and G. Das, "An Overview of Microcontroller Unit: From Proper Selection to Specific Application," in International Journal of Soft Computing and Engineering, vol. 2, no. 6, pp. 228-231, Jan. 2013, ISSN: 2231-2307.

[3] J.J. Rodriguez-Andina, M.J. Moure, and M.D. Valdes, "Features, Design Tools, and Application Domains of FPGAs," in IEEE Trans. Ind. Electron., vol. 54, no. 4, pp. 1810-1823, Aug. 2007, doi: 10.1109/TIE.2007.898279.

[4] E. Monmasson and M.N. Cirstea, "FPGA Design Methodology for Industrial Control Systems—A Review," in IEEE Trans. Ind. Electron., vol. 54, no. 4, pp. 1824-1842, Aug. 2007, doi: 10.1109/TIE.2007.898281.

[5] M. Carminati and G. Scandurra, "Impact and trends in embedding field programmable gate arrays and microcontrollers in scientific instrumentation," in Review of Scientific Instruments, vol. 92, no. 9, pp. 1-19, Sept. 2021, doi: 10.1063/5.0050999.

[6] Y. Güven et al., "Understanding the Concept of Microcontroller Based Systems To Choose The Best Hardware For Applications," in Research Inventy: International Journal of Engineering And Science, vol. 6, pp. 38-44, Sept. 2017, ISSN(e): 2278-4721.

[7] V. Bianchi, M. Bassoli, and I. De Munari, "Comparison of FPGA and Microcontroller Implementations of an Innovative Method for Error Magnitude Evaluation in Reed–Solomon Codes," in Electronics, vol. 9, no. 1, pp. 1-15, Jan. 2020, doi: 10.3390/electronics9010089.

[8] T. Kahl and S. Dieckerhoff, "Comparison of FPGA- and microcontroller-based control of a high-dynamic power electronic converter," in 2017 IEEE 18th Workshop on Control and Modeling for Power Electronics (COMPEL), pp. 1-6, July 2017, doi: 10.1109/COMPEL.2017.8013288.

[9] F. Ortega-Zamorano et al., "Efficient Implementation of the Backpropagation Algorithm in FPGAs and Microcontrollers," in IEEE Trans. Neural Netw. Learning Syst., vol. 27, no. 9, pp. 1840-1850, Sept. 2016, doi: 10.1109/TNNLS.2015.2460991.

[10] S. Gandhare and B. Karthikeyan, "Survey on FPGA Architecture and Recent Applications," in 2019 International Conference on Vision Towards Emerging Trends in Communication and Networking (ViTECoN), pp. 1-4, March 2019, doi: 10.1109/ViTECoN.2019.8899550.

[11] A. Malinowski and H. Yu, "Comparison of Embedded System Design for Industrial Applications," in IEEE Trans. Ind. Inf., vol. 7, no. 2, pp. 244-254, May 2011, doi: 10.1109/TII.2011.2124466.

[12] I. Kuon, R. Tessier, and J. Rose, "FPGA Architecture: Survey and Challenges," in FNT in Electronic Design Automation, vol. 2, no. 2, pp. 135-253, 2007, doi: 10.1561/1000000005.

[13] E. Monmasson et al., "FPGAs in Industrial Control Applications," in IEEE Trans. Ind. Inf., vol. 7, no. 2, pp. 224-243, May 2011, doi: 10.1109/TII.2011.2123908.

[14] P. Babu and E. Parthasarathy, "Reconfigurable FPGA Architectures: A Survey and Applications," in J. Inst. Eng. India Ser. B, vol. 102, no. 1, pp. 143-156, Feb. 2021, doi: 10.1007/s40031-020-00508-y.

[15] R.J. Molanes, J.J. Rodriguez-Andina, and J. Farina, "Performance Characterization and Design Guidelines for Efficient Processor–FPGA Communication in Cyclone V FPSoCs," in IEEE Trans. Ind. Electron., vol. 65, no. 5, pp. 4368-4377, May 2018, doi: 10.1109/TIE.2017.2766581.

[16] S. Saponara, L. Fanucci, and P. Terreni, "Architectural-Level Power Optimization of Microcontroller Cores in Embedded Systems," in IEEE Transactions on Industrial Electronics, vol. 54, no. 1, pp. 680-683, Feb. 2007, doi: 10.1109/TIE.2006.885450.

[17] D.E. Bolanakis, "A Survey of Research in Microcontroller Education," in IEEE REVISTA IBEROAMERICANA DE TECNOLOGIAS DEL APRENDIZAJE, vol. 14, no. 2, pp. 50-57, May 2019, doi: 10.1109/RITA.2019.2922856.

[18] K.P. Seng, P.J. Lee, and L.M. Ang, "Embedded Intelligence on FPGA: Survey, Applications and Challenges," in Electronics, vol. 10, no. 8, pp. 1-33, April 2021, doi: 10.3390/electronics10080895.

[19] M. Elnawawy et al., "Role of FPGA in Internet of Things Applications," in 2019 IEEE International Symposium on Signal Processing and Information Technology (ISSPIT), pp. 1-6, Dez. 2019, doi: 10.1109/ISSPIT47144.2019.9001747.

[20] R. Chéour et al., "Microcontrollers for IoT: Optimizations, Computing Paradigms, and Future Directions,"in 2020 IEEE 6th World Forum on Internet of Things (WF-IoT), pp. 1-7, Oct. 2020, doi: 10.1109/WF-IoT48130.2020.9221219.

[21] E.Z. Mohammed and H.K. Ali, "Hardware Implementation of Artificial Neural Network Using Field Programmable Gate Array," in International Journal of Computer Theory and Engineering, vol. 5, no. 5, pp. 780-783, Oct. 2013, doi: 10.7763/IJCTE.2013.V5.795.

[22] A. Muthuramalingam, S. Himavathi, and E. Srinivasan, "Neural Network Implementation Using FPGA: Issues and Application," in International Journal of Electrical and Computer Engineering, vol. 2, no. 12, pp. 625-631, Dec. 2008, doi: 10.5281/zenodo.1084402

[23] T.S. Li, S. Chang, and Y. Chen, "Implementation of human-like driving skills by autonomous fuzzy behavior control on an fpga-based car-like mobile robot," in IEEE Trans. Ind. Electron., vol. 50, no. 5, pp. 867-880, Oct. 2003, doi: 10.1109/TIE.2003.817490.

[24] K. Sridharan and T.K. Priya, "The Design of a Hardware Accelerator for Real-Time Complete Visibility Graph Construction and Efficient FPGA Implementation," in IEEE Trans. Ind. Electron., vol. 52, no. 4, pp. 1185-1187, Aug. 2005, doi: 10.1109/TIE.2005.851591.

[25] A. Amira and S. Chandrasekaran, "Power Modeling and Efficient FPGA Implementation of FHT for Signal Processing," in IEEE Trans. VLSI Syst., vol. 15, no. 3, pp. 286-295, Mar. 2007, doi: 10.1109/TVLSI.2007.893606.

[26] N.J. Cotton and B.M. Wilamowski, "Compensation of Nonlinearities Using Neural Networks Implemented on Inexpensive Microcontrollers," vol. 58, no. 3, pp. 733-740, Mar. 2011, doi: 10.1109/TIE.2010.2098377.

[27] C. Cecati et al., "Implementation Issues of a Fuzzy-Logic-Based Three-Phase Active Rectifier Employing Only Voltage Sensors," in IEEE Trans. Ind. Electron., vol. 52, no. 2, pp. 378-385, Apr. 2005, doi: 10.1109/TIE.2005.843918.

[28] Y.S. Kim, B.S. Soh, and S.G. Lee, "A New Wearable Input Device: SCURRY," in IEEE Trans. Ind. Electron., vol. 52, no. 6, pp. 1490-1499, Dez. 2005, doi: 10.1109/TIE.2005.858736.

[29] J. Matai, A. Irturk, and R. Kastner, "Design and Implementation of an FPGA-Based Real-Time Face Recognition System," in 2011 IEEE 19th Annual International Symposium on Field-Programmable Custom Computing Machines, pp. 97-100, May 2011, doi: 10.1109/FCCM.2011.53.

[30] L. Yang et al., "An FPGA implementation of low-density parity-check code decoder with multi-rate capability," in Proceedings of the ASP-DAC 2005, vol. 2, pp. 760-763, 2005, doi: 10.1109/ASP-DAC.2005.1466451.

[31] S. Poorani et al., "FPGA BASED FUZZY LOGIC CONTROLLER FOR ELECTRIC VEHICLE," in Journal of The Institution of Engineers, vol. 45, no. 5, pp. 1-14, 2005.

[32] L. Gomes et al., "Industrial electronic control: FPGAs and embedded systems solutions," in IECON 2013 - 39th Annual Conference of the IEEE Industrial Electronics Society, pp. 60-65, Nov. 2013, doi: 10.1109/IECON.2013.6699112.

[33] A. Kopmann et al., "FPGA-based DAQ system for multi-channel detectors," in 2008 IEEE Nuclear Science Symposium Conference Record, pp. 3186-3190, Oct. 2008, doi: 10.1109/NSSMIC.2008.4775027.

[34] M. Gabrick et al., "FPGA Considerations for Automotive Applications," in SAE 2006 World Congress & Exhibition, p. 2006-01-0368, April 2006, doi: 10.4271/2006-01-0368.

[35] A. Hofmann et al., "An FPGA based on-board processor platform for space application," in 2012 NASA/ESA Conference on Adaptive Hardware and Systems (AHS), pp. 17-22, June 2012, doi: 10.1109/AHS.2012.6268653.

[36] C. Wang et al., "A Ubiquitous Machine Learning Accelerator With Automatic Parallelization on FPGA," in IEEE Trans. Parallel Distrib. Syst., vol. 31, no. 10, pp. 2346-2359, Oct. 2020, doi: 10.1109/TPDS.2020.2990924.

[37] R. Chen, L. Chen, and L. Chen, "System design consideration for digital wheelchair controller," in IEEE Trans. Ind. Electron., vol. 47, no. 4, pp. 898-907, Aug. 2000, doi: 10.1109/41.857970.

[38] A. Aounis, "An Investigation into Induction Motor Vector Control Based on Reusable VHDL Digital Architectures and FPGA rapid Prototyping," 2002.

[39] M. Aimé, G. Gateau, and T.A. Meynard, "Implementation of a Peak-Current-Control Algorithm Within a Field-Programmable Gate Array," in IEEE Trans. Ind. Electron., vol. 54, no. 1, pp. 406-418, Feb. 2007, doi: 10.1109/TIE.2006.885501.

[40] B.D. Knapp ,L. Zhu, and K.C. Huang, "SiCTeC: An inexpensive, easily assembled Peltier device for rapid temperature shifting during single-cell imaging," Nov. 2020, [Online] available:

https://journals.plos.org/plosbiology/article?id=10.1371/journal.pbio.3000786.
[41] J. Yick, B. Mukherjee, and D. Ghosal, "Wireless sensor network survey," in Computer Networks, vol. 52, no. 12, pp. 2292-2330, Aug. 2008, doi: 10.1016/j.comnet.2008.04.002.
[42] K. Kutluay et al., "A New Online State-of-Charge Estimation and Monitoring System for Sealed Lead–Acid Batteries in Telecommunication Power Supplies," in IEEE Trans. Ind. Electron., vol. 52, no. 5, pp. 1315-1327, Oct. 2005, doi: 10.1109/TIE.2005.855671.
[43] T.-Y. Ho et al., "The Design and Implementation of a Sensorless Power Tool Based on a Microcontroller," in Electronics, vol. 9, no. 6, pp. 1-22, June 2020, doi: 10.3390/electronics9060921.
[44] M. Henrey et al., "Bio-inspired walking: A FPGA multicore system for a legged robot," in 22nd International Conference on Field Programmable Logic and Applications (FPL), pp. 105-111, Aug. 2012, doi: 10.1109/FPL.2012.6339248.
[45] J.J. Rodriguez-Andina, M.D. Valdes-Pena, and M.J. Moure, "Advanced Features and Industrial Applications of FPGAs—A Review," in IEEE Trans. Ind. Inf., vol. 11, no. 4, pp. 853-864, Aug. 2015, doi:10.1109/TII.2015.2431223.
[46] S.M. Trimberger and J.J. Moore, "FPGA Security: Motivations, Features, and Applications," in Proc. IEEE, vol. 102, no. 8, pp. 1248-1265, Aug. 2014, doi: 10.1109/JPROC.2014.2331672.
[47] T. Gomes et al., "Towards an FPGA-based edge device for the Internet of Things," in 2015 IEEE 20th Conference on Emerging Technologies & Factory Automation (ETFA), pp. 1-4, Sept. 2015, doi: 10.1109/ETFA.2015.7301601.

Floating-Point Units: Capabilities of Current Architectures and Approaches for Future Developments

Samuel Ardaya-Lieb
Faculty of Electrical Engineering and Information Technology
OTH Regensburg
Regensburg, Germany
samuel.ardaya@st.oth-regensburg.de

***Abstract*—Today more than ever, computationally intensive applications place high demands on digital hardware, especially when dealing with fractional numbers. In embedded systems, there are two basic approaches to meet these demands: computation in fixed-point format and computation in floating-point format. The latter offers high accuracy and a very broad range of values, but performing arithmetic operations with floating-point numbers is much more resource intensive. For this reason, special dedicated hardware units called floating-point units (FPUs) are used in microcontroller units (MCUs) and microprocessor units (MPUs) to accelerate floating-point calculations.**

In this paper, the structure and systematics of a floating-point number and the definition of arithmetic operations according to the IEEE 754 standard are presented. In addition, the FPU capabilities of current MPUs (by Intel) and MCUs (by Arm) are reviewed and compared. Then, current approaches of research on the possible further development of FPUs are shown. Since there are many different approaches in this research area, the consideration is focused on the development of new, non-standard floating-point number formats, such as BFloat16.

***Index Terms*—floating-point unit, floating-point arithmetic, microcontroller unit, microprocessor unit, computer architecture**

I. Introduction

Since the beginning of their development, electrical computers were designed to relieve their users of complex calculations. The common integer formats are simple and performant, but many areas of science and technology require fractional numbers. An alternative way of representing numbers in a computer is the floating-point format.
In 1941, Konrad Zuse introduced the Z3 to the scientific community, which is considered the first fully functional and freely programmable computer. The Z3 already had a floating-point processor that could perform basic arithemtic operations with decimal floating-point numbers with a word width of 22 bits. [1]

In the following decades, digital technology developed rapidly. Gordon Moore's statement of 1965, that the number of components in an integrated circuit doubles every two years [2] is often quoted in this context. While the performance of computers steadily increased and the range of digital electronics diversified, the basic requirement for a computer to perform calculations with fractional numbers always remained. This led in the 1970s and 80s to the situation that machines from different manufacturers implemented floating-point arithmetic completely differently. The same computation could yield different results depending on the computer model. The consequence was very poor portability of software. [3] For this reason, the Institute of Electrical and Electronics Engineers (IEEE) agreed in 1985 on a standard that gives clear guidelines for the implementation of floating-point arithmetic (IEEE Std 754-1985) [4].

The purpose of this paper is to provide an overview of how the floating-point number format works. The floating-point capabilities of current hardware and ongoing research in floating-point arithmetic are also considered. Therefore, chapter II presents the main contents of the IEEE 754 standard. Afterwards in chapter III both the development and the current capabilities of Intel MPUs regarding the handling of floating-point numbers are looked at. Chapter IV takes a similar look in the area of Arm MCUs. Current approaches to research on floating-point arithemtic are examined in chapter V. The approach of modifying floating-point formats to achieve efficiency gains is considered in Chapter VI.

II. The IEEE Standard for Floating-Point Arithmetic (IEEE Std 754)

The handling of floating-point numbers was first standardized in 1985 by the IEEE [4]. This standard was revised in 2008 [5] and 2019 [6]. The most important definitions are those of the number formats, since it was here that different layouts had caused problems [3]. In addition, certain arithmetic and other operations are defined. Furthermore, specifications are made for exceptions and exception handling. The most important contents are in the following presented as a rough summary of the standard.

A. Constructing a Floating-Poinit Number

The IEEE 754-1985 standard first defines the basic structure of a floating-point number [4, p. 3]. To understand this structure, an example is shown in Fig. 1.

Notation	Value
Ordinary Decimal	+178.125
Scientific Decimal	$+1.7825\ E_{10}\ 2$
Scientific Binary	$+1.0110010001\ E_{2}\ 111$
Scientific Binary (Biased Exponent)	$+1.0110010001\ E_{2}\ 10000110$

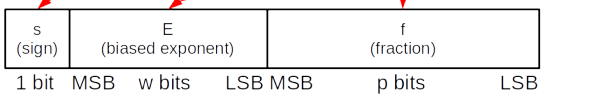

Fig. 1. The construction of a floating-point number. Inspired by [4, p. 3]

Fig. 1 shows how a floating-point number is constructed from a real number. First, an ordinary decimal number is transformed into a scientific decimal number, which consists of a fractional number and an exponent (base 10). This scientific decimal number is then transformed into a binary fixed-point number and a binary exponent (base 2). In the next step, a bias is added to the binary exponent in order to create the *biased exponent* [4, p. 3]. This is done to facilitate some operations, such as comparing two numbers [7]. Finally, this scientific binary with biased exponent is stored in a bit pattern referred to as the floating-point format. This format uses 1 bit so store the sign of the number, w bits to store the exponent and p bits to store the fraction of the binary fixed-point number. The fraction is also referred to as the *mantissa*.

B. Basic Formats

In general, any integers could be used for w and p to create a floating-point format. However, specific values have to be used in order to obtain the standard basic formats *Single* and *Double* (shown in table 1) [4, p. 4].

TABLE I
BASIC FORMAT PARAMETERS

Parameter	Single	Double
w	8	11
p	24	53
Exponent bias	127	1023
Total width in bits	32	64

Tabel 1 also shows the bias that has to be added to the binary exponent. The floating-point format provides the representation of finite numbers as well as two infinities ($+\infty$ and $-\infty$) and NaNs (Not a Number, see section II-E) [4, p. 3]. In 2008, three further basic formats were added: a binary format with a total width of 128 bits and two decimal formats [5, p. 6]. A 16-bit format (referred to as *half-precision*) was also defined [5, p. 13].

C. Rounding

The IEEE 254-1985 standard defines four types of rounding. The default method is called *Round to Nearest*. With this method, real values of infinite precision are rounded to the nearest possible value of the used floating-point format. In addition, there are the methods *Round towards* $+\infty$, *Round towards* $-\infty$ and *Round towards 0*. [4, p. 5]

D. Operations

The following arithmetic operations are defined in the 1985 standard: *add*, *subtract*, *multiply*, *devide* and *remainder*. Additionally there is the operation *square root*. Furthermore, there are some operations for converting data types or number formats. Also operations for the comparison of numbers are specified. [4, pp. 6-8] The 2008 and 2019 revisions each specified numerous additional, more sophisticated operations [5, pp. 17-33], [6, pp. 29-47].

E. Infinity, NaNs and Signed Zero

According to the IEEE 754 standard, floating-point numbers that exceed their maximum value or fall below their minimum value should be assigned the value $+\infty$ or $-\infty$ [4, p. 9]. Floating-point numbers whose bit patterns don't represent a real number are referred to as NaNs. NaNs can be divided into two categories: Signaling NaNs or Quiet NaNs. Signaling NaNs signal certain exceptions. Quiet NaNs propagate through arithemtic operations and can be used for own purposes, for example to store debug information. The cases in which the numeric value 0 receives a positive or negative sign (*Signed Zero*) are also standardized. [4, p. 10]

F. Exceptions

There are five types of exceptions whose occurrence shall be signaled: *Invalid Operation*, *Division by Zero*, *Overflow*, *Underflow* and *Inexact*. The standard provides for the use of so-called *Traps*, which can be controlled by the user to catch exceptions. [4, pp. 10-13]

The 2008 and 2019 revisions introduced many new features to the standard. However, these are primarily more detailed descriptions and advanced extensions of the existing content. The core statements and descriptions of the floating-point system from 1985 are still valid and important today.

III. FLOATING-POINT CAPABILITIES OF MICROPROCESSOR UNITS

An implementation of the IEEE 754 standard entirely in software would be relatively time-consuming due to the more complex structure of a floating-point number. Alternatively, the implementation can be realized in hardware in order to reduce the required computation time. While these hardware implementations are nowadays usually built into processors as so-called floating-point units (FPUs), it was common in the 1980s for processors to have no such hardware support. That's why Intel developed the 8087 Math Coprocessor in 1981, which served as an extension of the 8086 architecture

and added Arithemtic, Trigonometric, Exponential and Logarithmic instructions to the x86 instruction set. [8]

A. The x87 Floating-Point Unit

The 8087 coprocessor is obsolete as an integrated circuit (IC), but a closer look is nevertheless interesting. Its instruction set, known as x87 instruction set, is still valid today and can be used for applications. Also the FPU in Intel MPUs is still referred to as the x87 FPU. [9, Ch. 8 ,p. 1] Its data sheet includes an illustration of the IC's block diagram (see Figure 2).

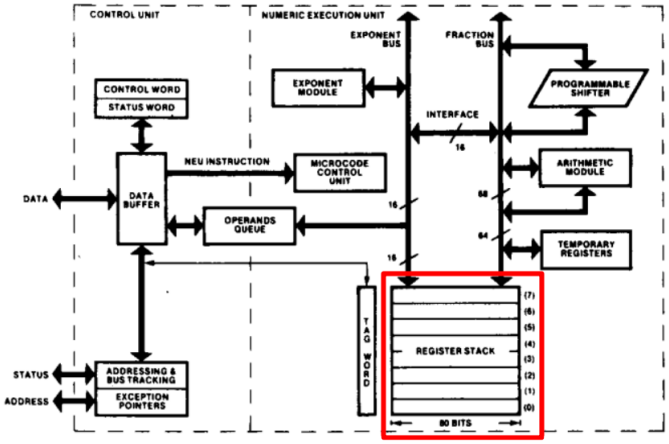

Fig. 2. The block diagram of the 8087 math coprocessor. [8, Ch. 3, p.90, changed by the author]

In Fig. 2 the register stack is highlighted. It consists of eight data registers which can store single or double precision floating-point operands. These registers have a width of 80 bits each. Both single precision (32 bit) and double precision (64 bit) operands are automatically transformed into a non-standard 80 bit floating-point format when stored in one of the registers. This is done in order to reduce rounding errors during successive floating-point operations. The data registers behave as a stack. The top of the stack can be addressed with a dedicated pointer. [10, Ch. 31, pp. 8-10]

B. The development of the floating-point instruction set

Modern processors are only described indirectly via their instruction sets. An instruction set contains all instructions that can be passed to a machine as assembler code. The Intel486 DX from 1989 was the first processor to integrate the Central Processing Unit (CPU) and the x87 FPU on one chip. The x87 FPU was further improved in subsequent processor generations, such as the Intel Pentium (1993), whereby the x87 instruction set remained valid. [10, Ch. 31, p. 1]

An important step in the evolution of floating-point capabilities was not only the improvement of the x87 FPU itself, but the introduction of the *single-instruction, multiple-data* (SIMD) execution model. SIMD is used to perform parallel computations with packed data. The first SIMD extension came in 1993 with the *MMX* technology, which was only designed for integer variables. On the Intel Pentium III, the *Streaming SIMD Extension* (SSE) was finally introduced, which included SIMD support for floating-point variables. [9, Ch. 2, pp. 1-3]

The following is an overview of subsequent developments.

- *Streaming SIMD Extension (SSE)*
  SSE can perform parallel floating-point operations for up to four single-precision operand pairs. It was updated several times (*SSE2*, *SSE3* and *SSE4*). SSE2 introduced operations for double-precision operands. SSE3 introduced new, more specialized instructions. Both SSE2 and SSE3 were used in the Pentium 4 Processor Family (2000-2006). SSE4 again introduced new instructions, such as the dot product. The Core i7 Processor Family (2008) and the Xeon Processor 5600 Series (2010), among others, use SEE4. [9, Ch. 2, pp.3-6], [9, Ch. 10], [9, Ch. 11], [9, Ch. 12]
- *Advanced Vector Extension (AVX)*
  With AVX on the one hand SSE instructions were improved. On the other hand, the number of floating-point operands that can be processed in parallel was further increased to eight operand pairs in single-precision format. AVX is used, for example, in the Second Generation Intel Core Processor Family (2011). With AVX-512 the floating-point parallelization was increased to 512 bits (e.g. 16 single-precision operand pairs). AVX-512 is used in current machines. [9, Ch. 2, pp. 6-7], [9, Ch. 14], [9, Ch. 15]

IV. Floating-Point Capabilities of Microcontroller Units

In the following, a closer look at the FPU capabilities of Arm MCUs is taken. Arm distinguishes between architectures and processors. Several processors can belong to one processor family. In the case of MCUs, this is the *Cortex-M* family. The MCUs of this family are in turn assigned to different architectures. Not all architectures have an FPU. Table 2 shows which MCU implements which architecture and whether this architecture has FPU support.

TABLE II
Arm Cortex-M FPU Support

Processor	Architecture	FPU
Cortex-M0	ARMv6-M	No
Cortex-M1	ARMv6-M	No
Cortex-M3	Armv7-M	No
Cortex-M4	Armv7-M	Optional
Cortex-M7	Armv7-M	Optional
Cortex-M23	Armv8-M	No
Cortex-M33	Armv8-M	Optional

Table 2 shows that there are three MCUs from Arm (M4, M7 and M33) that have the option for an FPU extension [11], [12], [13]. The FPU capabilities of these MCUs are summarized below.

- *Arm Cortex-M4*

The Cortex-M4 implements the *Armv7-M* architecture. This architecture has two optional floating-point extensions, described as *FPv4-SP* and *FPv5* [12, p. 22]. For the Cortex-M4, only the FPv4-SP extension is available. This FPU provides floating-point functionality according to the IEEE 754 standard, but only for the single-precision format. The FPU instruction set includes 15 floating-point operations, including basic arithmetic such as *add*, *subtract*, *multiply accumulate* and others. To store operands, the FPv4-SP extension provides a register bank with 32 single-precision registers. With the *Full-compliance mode*, the *Flush-to-zero mode* and the *Default NaN mode* three operation modes are available. The FPU informs about exceptions via a status flag, but exception traps are not supported. The FPU has to be switched on via a specific routine. [14, pp. 64-71]

- *Arm Cortex-M7*
  The Cortex-M7 also implements the Armv7-M architecture. Unlike for the Cortex-M4, the FPv5 instead of the FPv4-SP extension is available as an option here. The FPv5 is available in two versions, one with just single-precision support and one with single and double-precision support. Otherwise it is similar to the FPv4-SP, but provides some more instructions for floating-point operations. [15, Ch. 8, pp. 1-5]
- *Arm Cortex-M33*
  The Cortex-M33 implements the *Armv8-M* architecture. This architecture offers the FPv5 as an optional floating-point extension [13, p. 167]. However, the version of the FPv5 extension with double-precision support is not available for the Cortex-M33. The Cortex-M33 thus has similar floating-point capabilities to the Cortex-M4. In addition, however, it has a low-power operation mode of the FPU. [16, pp. 63-67]

The Cortex-M4, -M7 and -M33 arm can optionally be equipped with an FPU. This depends on the respective licensee. SIMD capabilities are also available to these MCUs, but only for integer variables [16, p. 20], [17].

V. Current scientific approaches to floating-point numbers

In this chapter, current research approaches in the area of floating-point numbers are examined. Scientific work usually shows new developments for FPUs that aim to increase efficiency in terms of data throughput, energy consumption, computation time or required chip area. For example, in 2017, Zhang and Zhao used the *CORDIC* algorithm to implement an FPU and were able to save hardware resources [18]. In 2012, Surapong et al. were able to increase performance and efficiency for a number of floating-point operations through *algorithmic analysis* [19]. In 2016, Camus et al. achieved significant savings in energy and area consumption by developing a so-called *speculative FPU* [20]. Other scientific works that achieved efficiency improvement using different approaches are [21], [22], [23]. It can be seen that the trend in research is towards resource-efficient floating-point implementations. In the following, one particular approach, namely the development of new floating-points formats, will be considered in more detail.

VI. Alternative Floating-Point Formats

Another way to increase energy efficiency was presented by Tagliavini et al. 2018. They introduced two new non-standard floating-points formats, called *binary16alt* and *binary8*, which have widths of 16 and 8 bits, respectively. In each case, they changed the widths of the exponent and mantissa such that a high dynamic range (determined by the exponent) was preserved, but the precision (determined by the mantissa) was significantly reduced. Then a series of algorithms were run and evaluated. The result was that up to 90% of floating-point operations could be safely scaled down to the smaller formats. This resulted in energy savings of up to 30% while reducing time and memory requirements. [24] Similar conclusions were found in [25] and [26]. These results are relevant to another scientific area that is currently attracting high attention in research and industry: *Deep Learning*. In 2018, Johnson showed that alternative, non-standard floating-point formats can make training and inference of artificial neural networks (ANNs) more memory efficient and thus more energy efficient [27]. In 2018, Intel introduced their own non-standard floating-point format, called *BFloat16*, to be used specifically in ANNs. This format comprises 16 bits, with the exponent being 8 bits wide, as in the IEEE 754 single-precision format. The mantissa, however, is truncated to 7 bits (compared to 23 bits in the standard format). [28] Tesla also uses their own non-standard floating-point formats: *CFloat8* and *CFloat16* with 8 and 16 bit word width, respectively. Tesla takes this approach a step further and does not limit itself to fixed bit widths for exponent and mantissa, but uses configurable widths to further adapt their data types to ANNs. [29]

VII. Conclusion

The floating-point number system has existed since the beginning of electrical computers. However, it took several decades until the scientific and technical community was able to agree on uniformly structured number formats and defined operations in 1985 in the form of the IEEE 754 standard for floating-point arithmetic. The floating-point units (FPUs) built into Intel's microprocessor units (MPUs) have since taken this standard into account, so that the x87 floating-point instruction set developed in 1981 is still valid today. There have been further developments of floating-point capabilities, particularly in the parallel processing of floating-point operations using single-instruction multiple-data (SIMD). FPUs are also used in Arm microcontroller units (MCUs), although not yet on every processor model. In addition, these MCUs are not yet providing floating-point SIMD capabilities. While the IEEE 754 standard has long been an important reference for hardware manufacturers, the trend in both current research and industry is towards non-standard floating-point formats. The development of these new formats is driven in part by the thriving field of Deep Learning.

REFERENCES

[1] H. Zuse, "Z3," [Online]. Available: http://www.horst-zuse.homepage.t-online.de/Konrad_Zuse_index_english_html/rechner_z3.html. [Accessed: May 17, 2022].

[2] G. E. Moore, "Cramming more components onto integrated circuits," Proceedings of the IEEE, vol. 86, no. 1, pp. 82–85, 1998.

[3] W. Kahan, "Why do we need a floating-point arithmetic standard?," University of California at Berkeley, February 1981.

[4] IEEE Standard for Binary Floating-Point Arithmetic, IEEE Std 754-1985, IEEE Standards Board, New York, USA, Mar. 21, 1985.

[5] IEEE Standard for Floating-Point Arithmetic, IEEE Std 754-2008, IEEE Computer Society, New York, USA, Aug. 29, 2008.

[6] IEEE Standard for Floating-Point Arithmetic, IEEE Std 754-2019, IEEE Computer Society, New York, USA, Jun. 13, 2019.

[7] D. Goldberg, "What Every Computer Scientist Should Know About Floating-Point Arithemtic," ACM Computing Surveys, vol. 23, no. 1, Mar., p. 18, 1991.

[8] Intel Corp., "8087 Math Coprocessor," 205835-007 datasheet, Oct. 1989.

[9] "Intel 64 and IA-32 Architectures Software Developer's Manual Volume 1: Basic Architecture," Intel Corp., 2016.

[10] Intel Corp., "Floating-Point Unit," [Online]. Available: https://johnloomis.org/ece314/notes/fpu/fpu.pdf. [Accessed May 20, 2022].

[11] "ARMv6-M Architecture Reference Manual," Arm Ltd., 2018.

[12] "Armv7-M Architecture Reference Manual," Arm Ltd., 2021.

[13] "Armv8-M Architecture Reference Manual," Arm Ltd., 2022.

[14] "Arm Cortex-M4 Processor Technical Reference Manual," Arm Ltd., 2020.

[15] "ARM Cortex-M7 Processor Technical Reference Manual," Arm Ltd., 2014.

[16] "Arm Cortex-M33 Processor Technical Reference Manual," Arm Ltd., 2020.

[17] T. Lorenser, "The DSP capabilities of ARM Cortex-M4 and Cortex-M7 Processors," Arm Ltd., 2016.

[18] B. Zhang and J. Zhao, "Elementary Function Computing Method for Floating-Point Unit," Springer Science+Business Media, 2016, doi: 10.1007/s11265-016-1166-x.

[19] P. Surapong, F. Philipp, F. A. Samman and M.Glesner, "Improvement of Standard and Non-Standard Floating-Point Operators," ECTI Transactions on Computer and Information Technology, vol.6, no. 1, 2012.

[20] V. Camus, J. Schlachter, C. Enz, M. Gautschi and F. K. Gurkaynak, "Approximate 32-bit floating-point unit design with 53% power-area product reduction," ESSCIRC Conference 2016: 42nd European Solid-State Circuits Conference, 2016, pp. 465-468, doi: 10.1109/ESSCIRC.2016.7598342.

[21] S. Galal and M. Horowitz, "Energy-Efficient Floating-Point Unit Design," in IEEE Transactions on Computers, vol. 60, no. 7, pp. 913-922, July 2011, doi: 10.1109/TC.2010.121.

[22] N. Hockert and K. Compton, "Improving Floating-Point Performance in Less Area: Fractured Floating Point Units (FFPUs)," Springer Science+Business Media, 2011, doi: 10.1007/s11265-010-0561-y.

[23] D. Peroni, M. Imani and T. S. Rosing, "Runtime Efficiency-Accuracy Tradeoff Using Configurable Floating Point Multiplier," in IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 39, no. 2, pp. 346-358, Feb. 2020, doi: 10.1109/TCAD.2018.2885317.

[24] G. Tagliavini, S. Mach, D. Rossi, A. Marongiu and L. Benini, "A transprecision floating-point platform for ultra-low power computing," 2018 Design, Automation & Test in Europe Conference & Exhibition (DATE), 2018, pp. 1051-1056, doi: 10.23919/DATE.2018.8342167.

[25] S. Mach, D. Rossi, G. Tagliavini, A. Marongiu and L. Benini, "A Transprecision Floating-Point Architecture for Energy-Efficient Embedded Computing," 2018 IEEE International Symposium on Circuits and Systems (ISCAS), 2018, pp. 1-5, doi: 10.1109/ISCAS.2018.8351816.

[26] D. Mukunoki and T. Imamura, "Reduced-Precision Floating-Point Formats on GPUs for High Performance and Energy Efficient Computation," 2016 IEEE International Conference on Cluster Computing (CLUSTER), 2016, pp. 144-145, doi: 10.1109/CLUSTER.2016.77.

[27] J. Johnson, "Rethinking floating point for deep learning," Facebook AI Research, New York, 2018.

[28] "BFLOAT16 - Hardware Numerics Definition," Intel Corp., 2018.

[29] "Tesla Dojo Technology A Guide to Tesla's Configurable Floating Point Formats & Arithmetic," Tesla Inc..

Kompressionsverfahren bei digitaler Sprachübertragung: Eigenschaften unterschiedlicher Verfahren

Christoph Möhring
Fakultät Elektro- und Informationstechnik
Ostbayerische Technische Hochschule Regensburg
Regensburg, Deutschland
christoph.moehring@st.oth-regensburg.de

***Zusammenfassung*—Um die begrenzte Bandbreite in einem (Mobilfunk-)Kanal für möglichst viele Nutzer zur Verfügung stellen zu können, werden für die Sprachübertragung verlustbehaftete Kompressionsverfahren zur Reduktion der Datenrate eingesetzt. Diese Arbeit zeigt die Eigenschaften unterschiedlicher Audiokompressionsverfahren, sogenannter Codecs auf. Im ersten Abschnitt wird auf das Sprachmodell der linearen Prädiktion (LP) und Analyse durch Synthese, welches bei Code-Excited Linear Prediction (CELP) Codecs eingesetzt wird, eingegangen. Nachfolgend werden die darauf basierenden Codecs G.729 und Speex, MELP, AMBE und Codec2 erläutert und deren Eigenschaften im Hinblick auf Sprachqualität, Komplexität und Latenz verglichen.**

***Schlüsselwörter*—Sprachcodec, Audiokomprimierung, Rechenaufwand, Lineare Prädiktion, Sprachsynthese**

I. Einleitung

Ziel der Komprimierung von Sprache ist es, die benötigte Bitrate, die zur Beschreibung eines Sprachsignals notwendig ist, zu reduzieren. Generell wird zwischen verlustbehafteten und verlustfreien Komprimierungsverfahren unterschieden. Diese Arbeit befasst sich mit verlustbehafteten Verfahren, welche im Vergleich zu verlustfreien Verfahren hohe Kompressionsraten erzielen können. Die nachfolgend vorgestellten Codecs nutzen unter anderem Eigenschaften des menschlichen Gehörs aus, um bestimmte, dem Menschen nahezu unhörbare Anteile der Sprache zugunsten geringerer Bitraten zu reduzieren oder gar zu verwerfen. Eine exakte Repräsentation des originalen Sprachsignals ist nach Encodierung und Decodierung dann nicht mehr möglich.

Die vorgestellten Verfahren können der Untergruppe der Analyse/Synthese Vocoder (*Voice Coder*) zugeordnet werden. Dabei wird das Sprachsignal im Encoder in einem modellhaften menschlichen Sprachorgan analysiert und mit einem Parametersatz beschrieben. Beim Decoder führt das parametrierte Modell zur Rekonstruktion eines wahrnehmungsgetreuen Abbilds des originalen Sprachsignals. Ein weit verbreitetes Verfahren zur Modellierung des Sprachorgans ist die lineare Prädiktion (LP) welche im nachfolgenden Kapitel weiter erläutert wird. Die grundlegende Funktionsweise der jeweiligen Codecs wird im Kapitel III beschrieben. Kapitel IV vergleicht die Eigenschaften der Codecs.

II. Lineare Prädiktion und Analyse durch Synthese

Die nachfolgend vorgestellten Sprachkomprimierungsstandards G.729 und Speex basieren auf dem CELP Codec. Das Grundprinzip von CELP und des Ansatzes „Analyse durch Synthese“ wird im Folgenden dargestellt. Mittels LP wird der menschliche Vokaltrakt modelliert und durch Parameter beeinflusst. Das Modell basiert darauf, dass für die Erzeugung von Sprache die Stimmbänder ein Anregungssignal erzeugen und der Vokaltrakt die jeweiligen Laute formt (bzw. spektral beeinflusst). Die akustische Energie zur Erzeugung von Sprache wird aus der Lunge entnommen. Unterschiedliche Laute können von deren Anregungssignal und spektralen Form unterschieden werden. [1] Stimmhafte Laute können im Zeitbereich durch eine Impulskette (periodisches Zusammenschlagen der Stimmbänder = Pitchfrequenz) modelliert werden. Stimmlose Konsonanten (wie „s“, „k“, „t“) können durch weißes Rauschen (Verwirbelungen des Luftstroms im Mundraum) als Anregungssignal approximiert werden. Stimmhafte Konsonanten (wie „b“, „d“, „g“) werden durch eine Mischung aus Rauschen und periodischem Anregungssignal generiert. [2] Lineare Prädiktion nimmt an, dass ein geschätztes Sprachsample $\hat{s}(n)$ durch eine Linearkombination aus vorhergegangenen (abgetasteten) Sprachsamples $s(n)$ vorhergesagt werden kann.

$$\hat{s}(n) = \sum_{k=1}^{p} a_k s(n-k) \tag{1}$$

Der Fehler der Abschätzung $\varepsilon(n)$ ergibt sich zu

$$\varepsilon(n) = \hat{s}(n) - s(n)\,. \tag{2}$$

Die Übertragungsfunktion

$$\frac{\varepsilon(z)}{s(z)} = 1 - \sum_{k=1}^{p} a_k z^{-k} := A(z) \tag{3}$$

aus Abbildung 1 führt unter der Annahme, dass die Koeffizienten $\alpha_k = a_k$ dazu, dass der Fehler $\varepsilon(n)$ als Anregungssignal $u(n) \cdot G$ des Vokaltraktes zur Synthese interpretiert werden kann.

$$H_s(z) := \frac{1}{A(z)} \tag{4}$$

beschreibt dabei die Übertragungsfunktion des Synthesefilters. [3] Typischerweise werden Filter $H_s(z)$ der Ordnung 10 bis 16 eingesetzt [4].

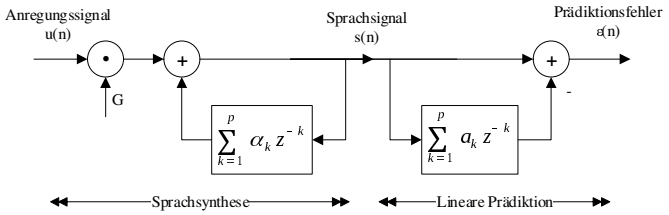

Abbildung 1. Sprachmodell der LP - angelehnt an [3]. Erzeugung eines Sprachsignals durch ein Anregungssignal und ein Synthesefilter und umgekehrt

Ziel der linearen Abschätzung ist es, einen Koeffizientensatz a_k für ein kurzes Sample-Intervall von ca. 10-20 ms für das Synthesefilter $H_s(z)$ zu finden, bei dem die Summe (über ein betrachtetes Zeitfenster) des quadratischen Prädiktionsfehlers ε^2 minimiert wird. Das Anregungssignal der LP als Residuum führt bei sich stark ändernden Signalwerten, wie es beim periodischen Auftreten des Pitch Pulses bei stimmhaften Lauten auftritt, zu starken Signaländerungen des selbigen. Durch einen Langzeitschätzer, der die Pitch Periode ermittelt, kann das Residuum der LP auf gaußsches Rauschen weiter minimiert werden. Dies führt zu einer weiteren Verringerung der Redundanz des Sprachsignals. Die Sequenz des gaußschen Rauschens kann weder durch Linear Predictive Coding (LPC) oder Vorhersage des Pitch Pulses bestimmt werden und nimmt bei der Übertragung die meisten Bits ein [5].

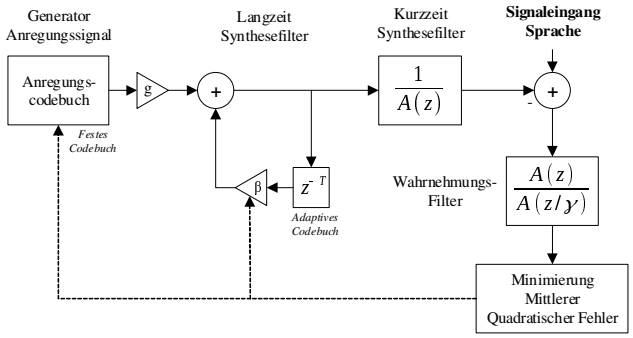

Abbildung 2. Vereinfachtes Blockschaltbild eines CELP Encoders - angelehnt an [3]

Beim CELP Encoder wird aus einem Codebuch mit definierten variierten Abtastwerten gaußschen Rauschens (vgl. Abbildung 2: Anregungscodebuch) das *Langzeit Synthesefilter* angeregt. Die erzeugte Codesequenz regt wiederum das *Kurzzeit Synthesefilter* an, um ausgangsseitig synthetisierte Sprache zu erhalten. Bei CELP besteht das Codebuch zur Anregung aus 1024 Codevektoren. Der Fehler aus Eingangs-Sprachsample und synthetisierter Sprache wird durch Variation des Anregungsvektors, der Verstärkung des Anregungssignals und der Verstärkung des Langzeit Synthesefilters minimiert. Sollen stimmhafte Teile der Sprache übertragen werden, wird β größer bestimmt werden, bei stimmlosen (dem Rauschen ähnlicheren Lauten) wird g größer bestimmt werden. [4] Um eine Optimierung dieser Parameter in Echtzeit auf Hardware mit begrenzten Ressourcen zu erhalten, wird die Optimierung stufenweise (und nicht für alle Parameter im geschlossenen Kreis) durchgeführt und das Hörvermögen des Menschen durch ein Wahrnehmungsfilter berücksichtigt [6]. Das bei der Synthese entstehende Rauschen, überlagert mit dem Rauschen des quantisierten Eingangssignals, wird im Wahrnehmungsfilter spektral verformt, sodass die Rauschleistung im Bereich der Formantfrequenzen mit höherem Pegel größer sein darf, respektive bei *pegelschwächeren* Frequenzbereichen stärker bedämpft werden kann, um keinen hörbaren Unterschied zu erzielen [7]. Beim CELP Codec wird das *Langzeit Synthesefilter* zur schnelleren Fehlerminimierung als adaptives Codebuch implementiert, welches Variationen zeitlicher Ausschnitte des Verzögerungsgliedes (vgl. Abbildung 2: z^{-T}) des Langzeit Synthesefilters bei unterschiedlichen Pitch-Perioden enthält [4].

III. Sprachcodecs

A. G.729

G.729 bezeichnet die Empfehlung der ITU-T zur Codierung von Sprachsignalen mittels CS-ACELP (conjugate-structure algebraic-code-excited linear prediction). Die ausgangsseitige Datenrate ist dabei auf 8 kBit/s festgelegt, diverse Annexe beschreiben Abwandlungen des Codecs für weitere Datenraten und Komplexitätsstufen der Implementierung. Die analogen Sprachsignale werden für Telefonanwendungen gemäß ITU-T G.712 bandbreitenbegrenzt, bei einer Abtastrate von 8 kHz abgetastet und als 16-Bit PCM Datenstrom dem Encoder übergeben. Sowohl zur De- als auch zur Encodierung werden 10 ms Sprachrahmen, respektive 80 Abtastwerte, zusammengefasst. Übertragen werden mit 80 Bit pro 10 ms Rahmen die Filterkoeffizienten des Synthesefilters 10. Ordnung, die Indizes des adaptiven und des festen Codebuchs und dessen Verstärkungen. Die LP wird über ein 10 ms Frame durchgeführt und dessen Koeffizienten bestimmt. Die Werte der Koeffizienten werden in Linienspektrum-Paaren repräsentiert und vektorquantisiert dem Datenstrom hinzugefügt. Die Parameter des Anregungssignals werden in 5 ms Frame Intervallen durch Minimierung des Fehlersignals (zwischen Sprachsignal und synthetisierter Sprache) ermittelt. Die nicht quantisierten LPC-Koeffizienten dienen der Einstellung des Wahrnehmungsfilters, dessen Gewichtung adaptiv eingestellt wird. Die Pitch-Verzögerung wird im Abstand von 10 ms in einem offenen Kreis auf Basis des wahrnehmungsgewichteten Residuums geschätzt. In einem geschlossenen Kreis wird die eingestellte Verzögerung und die Verstärkung des adaptiven Codebuchs bestimmt. [8]

B. Speex

Der quelloffene und lizenzfreie Codex Speex basiert auf CELP und eignet sich durch dynamische Bitratenumschaltung für VoIP Kommunikationen. Die Framelänge beträgt 20 ms, was im Schmalbandbereich von 8 kHz 160 Samples entspricht.

Es erfolgt eine Unterteilung in vier Subframes zu je 40 Samples. Der Codec encodiert hingegen zum *CELP Codec* die Pitch Periode des adaptiven Codebuchs mit einen 3-fach verzögerten Schätzer mit jeweils drei Verstärkungen. Die Verstärkung des festen Codebuchs wird global für ein gesamtes Frame (und alle Subframes) berechnet. Das Anregungssignal wird pro Subframe abhängig der eingestellten Bitrate in Sub-Vektoren unterteilt und einem bitratenabhängigen Codebuchvektor zugeordnet. Die optimierten Parameter werden bei Speex mit den nachfolgend aufgeführten Schritten ermittelt:

1) LP durchführen, Linienspektrum Paare ermitteln und anschließende Vektorquantifizierung
2) Analyse durch Synthese des adaptiven Codebucheintrags und der Codebuchverstärkung
3) Ermittlung der Verstärkung des festen Codebuchs in offenem Kreis durch Bestimmung der Energie des Anregungssignals
4) Ermittlung des festen Codebucheintrags durch Analyse durch Synthese

Anmerkung: Die Schritte 2) und 4) werden durch ein Wahrnehmungsfilter (wie bereits in Abschnitt II beschrieben) gewichtet, um die wahrnehmbare Differenz zum Eingangssignal zu minimieren.

Ab Version 1.1.1 kann der Coder auch in Festkomma-Arithmetik betrieben werden, wenn entsprechende Software Makros bei der Implementierung gesetzt werden. Ein Wideband Modus mit 16 kHz Abtastrate ist ebenfalls implementiert. Hierbei wird das Sprach Spektrum in Narrowband und High Band (nur festes Codebuch, kein adaptives) unterteilt und separat encodiert. Ferner bietet der Encoder die Möglichkeit von AGC, *voice activity detection* und Rauschunterdrückung in einem separaten Präprozessor Modul. Letztere kann die Sprachqualität nach der Codierung verbessern. [6]

C. MELP

MELP (Mixed Excitation Linear Prediction) wurde 1993 vom Department of Defense als neuer 2,4 kBit/s Coder ausgewählt und unter dem MIL-STD 3005 standardisiert. Eine Erweiterung des Coders enhanced-MELP (MELPe) im NATO Standard STANAG 4591 reduziert die Datenrate auf 1,2 kBit/s. MELP basiert auf die Sprachsynthese mittels eines LP Filters. Das Anregungssignal des Filters besteht dabei aus einem Gemisch aus Rauschen und pulsförmigen Signalanteilen. Zusätzlich wird bei MELP in stimmhaften Signalanteilen mit geringer Korrelation zur Pitchfrequenz, die Pulsfolge des Anregungssignals mit einem Jitter versehen um eine natürlichere Sprachsynthese zu ermöglichen. [20] Die Analyse der Sprache erfolgt dabei in 5 Frequenzbändern, wobei das unterste Frequenzband (0-500 Hz) zur initialen Pitch-Analyse verwendet wird. Im niedrigsten Frequenzband wird die Stimmhaftigkeit des Frames auf Basis der ermittelten Pitch Frequenz abgeschätzt, anschließend wird für die oberen Frequenzbänder die Stimmhaftigkeit (Intonationsstärke) bestimmt. Im Anschluss werden die LPC Koeffizienten ermittelt und zusammen mit der Pitchfrequenz, der Verstärkung und der Intonationsstärke quantisiert und dem Bitstrom zur Übertragung übergeben. [21]

D. AMBE und CODEC2

Advanced Multi-Band-Excitation (AMBE) und Codec2 basieren auf dem Multi-Band-Excitation (MBE) Sprachmodell. Es erfolgt hierbei eine Unterscheidung zwischen stimmhaften und stimmlosen Lauten eines Sprachsegmentes in mehreren Regionen des Sprach-Frequenzspektrums. [9] Über einen kurzen Zeitraum kann die Fourier Transformierte eines Sprachsegmentes durch die Einhüllende und ein Anregungssignal beschrieben werden. Die Einhüllende kann dabei als geglättete Version (Abb. 3b) des Eingangs-Sprachspektrums (Abb. 3a) angesehen werden. Aus der Pitch Frequenz wird ein periodisches Spektrum (Abb. 3c) der Grundfrequenz mit den Harmonischen erzeugt. Periodische Frequenzbänder des Sprachsegments werden als periodisch deklariert und im Anregungsspektrum (Abb. 3f) mit dem periodischen Spektrum modelliert (High Signal in Abb. 3d). Frequenzbänder mit Rauschanteil werden mit einem Rauschspektrum abgebildet. Das Spektrum des Anregungssignals besteht aus einer Mischung der Harmonischen der Pitch Frequenz und einem spektralen Rauschanteil (Abb. 3e). Das synthetisierte Spektrum (Abb. 3g) ergibt sich aus der Multiplikation der Einhüllenden des Originalsignals mit dem Anregungsspektrum. Die stimmhaften und stimmlosen Frequenzanteile treten in breiteren Frequenzbereichen (siehe Abb. 3d) auf. [10] Durch Lauflängencodierung kann diese binäre Information komprimiert werden. Ein MBE Encoder codiert die Pitch Frequenz, die stimmhaften/stimmlosen Frequenzblöcke und Amplitudenwerte der spektral Einhüllenden [11].

IV. Eigenschaften

A. Sprachqualität

Die Bewertung der Sprachqualität kann nicht direkt mathematisch bewertet werden, da diese vom Sprachverständnis abhängt. Faktoren, wie Hintergrundrauschen, Aussprache des Sprechers und Verständnis des Zuhörers beeinflussen die Sprachverständlichkeit. Objektiv kann das Signal-Rausch-Verhältnis aus decodierter Sprache und Originalsprache ermittelt werden, zur Bewertung der Sprachqualität wird aber ein subjektives Maß, wie der MOS (Mean Opinion Score) durch Verständlichkeitstests ermittelt. Eine Gruppe aus Zuhörern bewertet dabei an einer Skala von 1 (inakzeptabel) bis 5 (ausgezeichnet) die Sprachqualität von gehörten Sprachausschnitten. Der Mittelwert der Bewertungen entspricht dem MOS, hierbei gilt es zu beachten, dass der Vergleich unterschiedlicher MOS-Testreihen aufgrund unterschiedlicher Ausgangsbedingungen (Hörergruppen) nicht direkt möglich ist. [12] Perceptual evaluation of speech quality (PESQ) bewertet hingegen die Sprachwahrnehmung algorithmisch in einem kognitiven Modell, was mit der menschlichen Wahrnehmung und somit dem MOS korreliert werden kann [13]. Je nach Übertragungskanal kann der MOS weiter, beispielsweise durch Latenz und variierender Bitrate, verändert werden.
Der Codec Speex erzielt bei diesem Vergleich aus Tabelle I

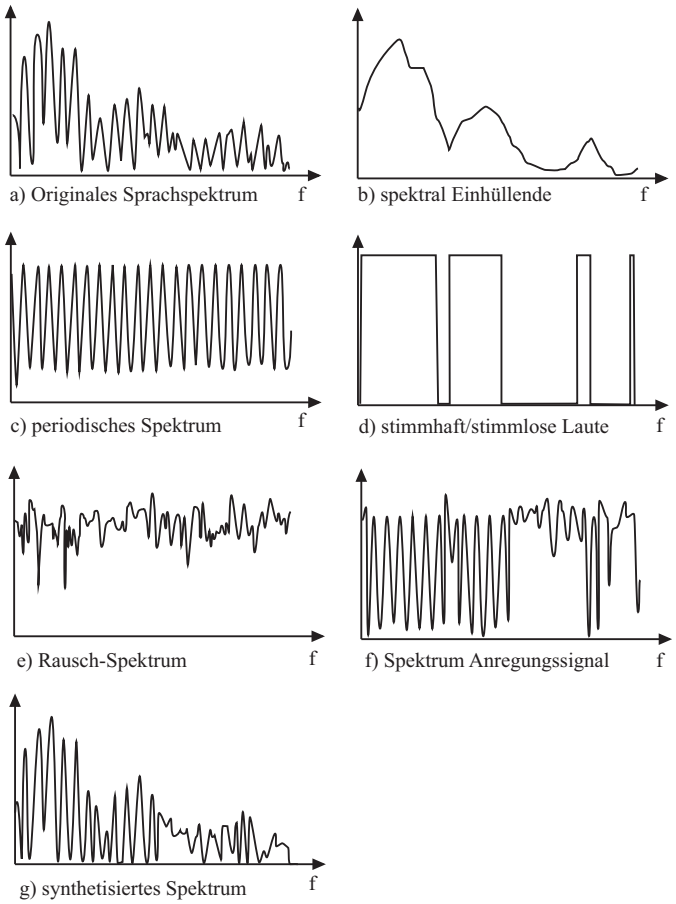

Abbildung 3. Spektrale Darstellung ausgewählter Signale des MBE Encoders - angelehnt an [10]

den besten Sprachverständlichkeitsindex. Berücksichtigt man die Bitrate des Codecs, erreicht AMBE einen zu Speex vergleichbaren MOS bei halbierter Bitrate.

Tabelle I
MOS SPRACHQUALITÄT NACH [12,14]

Codec	MOS	Bitrate [kBit/s]
G.729	3,7	8
Speex	3,75	8
MELP	3,3	2,4
AMBE	3,7	3,6
Codec2	3,2*	2,5

* Dieser Wert wurde aus [18] entnommen. Der Entwickler gibt allerdings eine bessere Sprachqualität bei 700 Bit/s als MELP an.

B. Komplexität

Die Komplexität eines Codecs muss sowohl für die Encodierung als auch die Decodierung separat betrachtet werden. Tabelle II zeigt die Komplexität, angegeben in *Million Instructions per Second* (MIPS), bei Implementierung der Codecs auf ARM-basierten Plattformen auf. Der Rechenaufwand für die Encodierung ist bei allen vorgestellten Codecs größer als der der Decodierung. Der große Wertebereich bei Speex erklärt sich durch die variable Parametrierbarkeit (*Quality*, *Complexity*) des Codecs.

Tabelle II
ALGORITHMISCHE KOMPLEXITÄT DER CODECS NACH [15–17]

Codec	MIPS Encodierung	MIPS Decodierung	Plattform/ Anmerkung
G.729	12,1	4	ARMv7 Cortex-A8 Annex A
Speex	41-116	4-5	ARM Cortex-A8 Narrowband Mode
MELP	40	13	ARMv8A Cortex-A53 MELPe@2,4 kBit/s
AMBE			keine Angabe*
Codec2			keine Angabe†

* aufgrund der Lizenzierung und des patentrechtlichen Schutzes konnten keine Informationen zur Komplexität ermittelt werden
† keine Rechercheergebnisse zur Komplexitätsbewertung

C. Latenz

Die vorgestellten Codecs en- bzw. decodieren die Sprachsignale in Frame-Intervallen welche im Codec zwischengespeichert werden müssen. Dies führt zu einer algorithmischen Verzögerung. Der weiter entstehende Zeitversatz (beispielsweise bei einer Codebuchsuche oder bei der Paketierung der Daten zur Übertragung über IP) ist von der Rechenleistung der Implementierungsplattform und dem verfügbaren Übertragungskanal abhängig und wird in dieser Arbeit nicht weiter betrachtet. [5] Der Codec G.729 besitzt eine algorithmische Verzögerung sowohl bei Encodierung als auch Decodierung von 15 ms, Speex eine Verzögerung von 30 ms [5,8]. Beim AMBE-2020, einem Vocoder IC mit AMBE Implementierung, beträgt die algorithmische Verzögerung jeweils 42 ms. Der Codec MELPe verzögert um 103 ms bei einer Bitrate von 1,2 kBit/s. Eine geringe Bitrate korreliert bei dieser Betrachtung mit einer größeren algorithmischen Verzögerung. In VoIP Kanälen wird für eine gute Gesprächsqualität eine Gesamtverzögerung von maximal 150 ms gefordert [19].

V. FAZIT

Die hier vorgestellten Codecs erreichen eine gute Sprachqualität bei Bitraten bis 8 kBit/s. Neben den vorgestellten Eigenschaften (Sprachqualität, Komplexität und Latenz) der Codecs G.729, Speex, MELP, AMBE und Codec2 sind unter anderem auch lizenzrechtliche Aspekte bei der Implementierung zu beachten. Der konkrete Anwendungsfall bestimmt die notwendigen Eigenschaften eines auszuwählenden Codecs maßgeblich. Es empfiehlt sich, bei jeder Entwicklung die Anforderungen an die Sprachkomprimierung zu evaluieren und dann einen geeigneten Codec auszuwählen. In Zukunft werden Sprachcodecs mit noch stärkerer Kompressionsrate und besserer Sprachqualität erwartet. Als Weiterarbeit bietet sich die wissenschaftlich fundierte Untersuchung des MOS und der Komplexität von Codec2 an, um eine präzisere Einordnung dieses Codecs in die Gruppe der verlustbehafteten Codecs zu ermöglichen.

LITERATUR

[1] A. V. McCree, “Low-Bit-Rate Speech Coding,” in Springer Handbook of Speech Processing, Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 331–350.

[2] J. Benesty, J. Chen, and Y. Huang, “Linear Prediction,” in Springer Handbook of Speech Processing, Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 121–134.

[3] E. Ambikairajah, “ELEC9344: Speech & Audio Processing,” 14-Aug-2013.

[4] J.-H. Chen and J. Thyssen, “Analysis-by-synthesis speech coding,” in Springer Handbook of Speech Processing, Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 351–392.

[5] J.-M. Valin, “The Speex Codec Manual Version 1.2 Beta 3,” Dec. 2000.

[6] J.-M. Valin, “Speex: A Free Codec For Free Speech,” arXiv [cs.SD], 2016.

[7] B. Atal and M. Schroeder, “Predictive coding of speech signals and subjective error criteria,” IEEE Trans. Acoust., vol. 27, no. 3, pp. 247–254, 1979.

[8] International Telecommunication Union, “Coding of speech at 8 kbit/s using conjugate-structure algebraic-code-excited linear prediction (CS-ACELP),” Telecommunication Standardization Sector of ITU, Jun. 2012.

[9] M. Brandstein, J. Hardwick, and J. Lim, “The Multi-Band Excitation Speech Coder,” in Advances in Speech Coding, Boston, MA: Springer US, 1991, pp. 215–223.

[10] D. Griffin and J. Lim, “A new model-based speech analysis/Synthesis system,” in ICASSP ’85. IEEE International Conference on Acoustics, Speech, and Signal Processing, 2005, vol. 10, pp. 513–516.

[11] C. Redding, N. DeMinco, and J. Lindner, “Voice Quality Assessment of Vocoders in Tandem Configuration,” Apr. 2001.

[12] R. Goldberg and L. Riek, A practical handbook of speech coders. Boca Raton, FL: CRC Press, 2000.

[13] International Telecommunication Union, “Perceptual evaluation of speech quality (PESQ): An objective method for end-to-end speech quality assessment of narrow-band telephone networks and speech codecs,” Telecommunication Standardization Sector of ITU, Feb. 2001.

[14] P. Srivastava, K. Babu, and T. Osv, “Performance evaluation of Speex audio codec for wireless communication networks,” in 2011 Eighth International Conference on Wireless and Optical Communications Networks, 2011, pp. 1–5.

[15] “Speex codec – speech compression,” Adaptive Digital. [Online]. Available: https://www.adaptivedigital.com/speex/. [Accessed: 16-May-2022].

[16] “MELPe - military communication codec,” Adaptive Digital. [Online]. Available: https://www.adaptivedigital.com/melpe/. [Accessed: 16-May-2022].

[17] “G.729,” Adaptive Digital. [Online]. Available: https://www.adaptivedigital.com/g-729/. [Accessed: 16-May-2022].

[18] “Codec 2,” Rowetel.com. [Online]. Available: https://www.rowetel.com/wordpress/?page_id=452. [Accessed: 16-May-2022].

[19] AVAYA, “VoIP & Video Conferencing Mindestanforderungen im LAN,” Jul. 2011.

[20] M. Yang, “Low bit rate speech coding,” IEEE Potentials, vol. 23, no. 4, pp. 32–36, 2004.

[21] L. M. Supplee, R. P. Cohn, J. S. Collura, and A. V. McCree, “MELP: the new Federal Standard at 2400 bps,” in 1997 IEEE International Conference on Acoustics, Speech, and Signal Processing, 2002, vol. 2, pp. 1591–1594 vol.2.

Overview of Different Approaches and Types of Penetration Testing

Matthias Solisch
Faculty of Electrical Engineering and Information Technology
Ostbayerische Technische Hochschule Regensburg
Regensburg, Germany
matthias.solisch@st.oth-regensburg.de

***Abstract*—In the current era of digitalization, more and more everyday tasks are processed digitally. Payments and the processing of banking transactions can nowadays be carried out easily via the Internet. Even statements by politicians are published via social media platforms these days. This vast amount of data, representing financial strength, political influence, and much more, provides a target for crime on the Internet. Hackers can gain unauthorized access to important data of any kind through vulnerabilities in IT systems. To ensure security, so-called penetration tests (pen tests or ethical hacking) are performed. They test the security of IT systems with methods of real hackers, by breaking into the companies' IT structure. In the paper publications and studies are compared to elaborate on the different approaches and types of pen tests. For better unterstanding of the different types, the paper explains the general process of ethical hacking. The types of pen tests can be categorized by the vulnerability of the system, which could be exploited by a real hacker. The paper addresses pen tests, which examine electronic/digital vulnerabilities, errors because of human mistakes, and weaknesses in the physical barriers of a company. Pen tests are not only limited on testing network structures or bypassing firewalls. For a safe IT system, human error or intrusion into the company should be prevented. The different types of pen tests try to find these weaknesses and exploit them. From the knowledge gained from the tests, improvements will be proposed for the future to make the IT structure safer against cybercrime.**

***Index Terms*—IT Security, Penetration Testing, Vulnerability Assessment, Ethical Hacking**

I. Introduction

Digitalization is leading to greater networking of the Internet. New technologies such as the Internet of Things (IoT) are based on the approach of enabling communication in infrastructure only through the Internet. One problem that has existed since the beginning of the Internet is cybercrime. So-called hackers use targeted attacks to cause damage through sabotage or information extraction and blackmail the victims to demand extortion money [1].

In today's world, digital security is essential. How essential it is, shows the evolution of the amount of monetary damage caused by cybercrime. The amount is steadily increasing over the last 20 years resulting in reported losses up to billions of USD in 2020 [2]. It is expected that the amount is going to increase by 15% per year over the next five years, which makes IT security more important than ever [3].

Hackers are constantly coming up with new techniques and tricks to gain access to valuable data in IT systems. To avoid possible risks and attacks, the method of penetration testing was introduced. A commissioned person, which is hired by a company uses techniques to simulate an attack on the IT structure. The purpose is to investigate the resistance of the existing IT security system. These weaknesses are not only limited to errors in digital applications. Human error or weaknesses in the physical boundaries of an organization can also lead to disastrous consequences.

The different types of pen tests try to find and exploit all these vulnerabilities to improve the safety of digital data. One type of pen testing is electronic testing, where the ethical hacker tries to get access by exploiting weak spots in network protocols and applications the company is using. The second section of pen testing types are the social engineering tests. In the process, the employee is specifically chosen as the weak point in the system. Fake emails or calls are meant to trick the employee to make mistakes, like publishing valuable data, relinquishing passwords, or installing malware. Physical barriers are also important for IT security and are examined by physical pen tests. This involves an attempt to get unauthorized access into the company and look for information, that could be used to attack the company in a real scenario. All the different pen test types are designed to improve the security of the entire organization's computing system. [4]

In this paper, we are giving an overview of the different approaches and types of pen testing. In Chapter II we explain the background of ethical hacking. After that, we go into detail about the different types of pen testing in Chapter III. The subchapters are categorized into electronic, social engineering, and physical pen tests. Finally, we present a conclusion in Chapter IV.

II. Pen Testing Background

Regardless of the test type, every approach of ethical hacking is meant to improve IT security in an organization or a company. To guarantee a successful execution a regulated process is necessary. There is no rigid specification in the literature for how to conduct a test. Various sources only give suggestions and differ in some point [5-8]. In general, the process of a pen test can be divided into the following phases:

1) Specify Conditions
2) Information Gathering

3) Evaluation of Information
4) Attack
5) Reporting/Documentation

Specifying the conditions of a pen test is crucial for successful execution. First of all the test should have the approval of the management of the organization that is the subject of the test, because only the management has the authority to permit any type of activity on its network or IT system. An important part in the conditions of a pen test is a clearly defined goal, to determine success or failure easier. Another condition is the amount of time available to conduct the process. In the real world, an attacker will spend a finite amount of time and energy to penetrate a target. After the time limit or a certain number of unsuccessful attempts have been exceeded, the hacker will give up. The management and the tester should define a limited time period for the project to save the financial resources of the company [9, 10]. In addition, contracts are important for the protection of the company and the ethical hacker when unexpected events occur like information leakage and downtime. [5, 7]

Information gathering takes a very important part in conducting a pen test. This step can differ depending on the general conditions made up for the test. If a company tells the ethical hacker all needed information and details about weak spots to explore in the system the pen test is called a white box test. The tester won't have to make a vulnerability assessment because he already knows the weaknesses. In a black box pen test, the tester has no inside knowledge about the subject to explore. He needs to make himself an overview of the subject to test and takes notes about possible vulnerabilities to exploit. He needs to find weak spots all by himself. In a gray box test,

TABLE I
PENETRATION TEST TYPES DEPENDING ON INFORMATION BASE [6]

Test Type	Information	Time Period	Depth	Realism
White Box	Very high	Low	Very high	Very low
Gray Box	Variable	Varibale	High	High
Black Box	Very low	High	Very low	Very high

the amount of published information for the tester varies and is set in the conditions made before the test. The different test types are summarized in Table I. [5]

After gathering information about the subject, the information has to be evaluated. Besides the effort for exploring different vulnerabilities, the evaluation criteria are the set goals in the tests or the risk of causing unwanted damage in the attack phase. The ethical hacker prepares himself for the attack phase and chooses the vulnerabilities to penetrate by using the evaluation. [4]

The resistance of the IT security system is tested in the attack phase. The tester tries to get valuable data or reach for administration status to get control of the system by exploiting found vulnerabilities. The attack phase differs by the type of the pen test. We go into detail about this phase in the Chapter III where we explain the different pen test types.

In the last phase of a pen test, the ethical hacker reports his documentation of the process. In the final analysis, he presents his results to the management and makes suggestions for improvement or confirms a secure IT structure. Vulnerabilities and weaknesses in the systems should be ranked by importance to realize improvements as quickly as possible. [4]

III. PENETRATION TESTING TYPES

The classification of the individual types differ in the literature [5-8]. In this paper, we classify the pen test types into electronic, social engineering, and physical penetration tests. We explain these types in detail in the following subchapters. Fig. 1 shows the classification of the pen test types used in this paper.

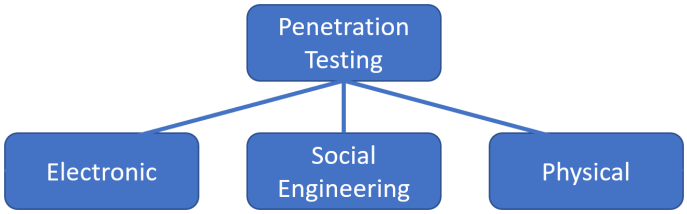

Fig. 1. Classification of Penetration Tests

A. *Electronic Testing*

Electronic pen testing deals with digital vulnerabilities in the IT structure. The ethical hacker tests the network, the computer system or communication facilities of the organization [10]. In this paper, we present two scenarios to make the network and the website safer. It needs to be mentioned, that electronic pen testing covers a lot more. However, this would exceed the scope of this paper.

One example of electronic pen testing is to examine the network of a company. The network consists of many components. A simplified representation of a company network is shown in Fig. 2. A lot of the components could be a target for cyberattacks. Within the scope of the pen test, the tester needs to find the most important parts in the network to examine. These could be for example the firewall, routers, or web servers [4]. In order to prevent hackers from gaining access to systems within the network, the ethical hacker uses the common techniques and software of real cybercriminals. Most of the used tools can be downloaded for free from the Internet. A common tool used by hackers to find vulnerabilities are port scanners. They scan the targeted object about the status of the ports working with TCP or UDP via the Internet protocol [9]. This tool gives the opportunity to look for not used open ports in a system like the firewall which could be exploited by a hacker and lead to bypassing the security instance [4]. Some port scanners like the free available software called Nmap can also fingerprint the services used by the object. With this method, the user can see the version of the applications running in the system [11]. This is important because some older versions of applications like a web server could be more vulnerable to exploits than up-to-date software [6]. After locating these weaknesses, the ethical hacker tries to exploit

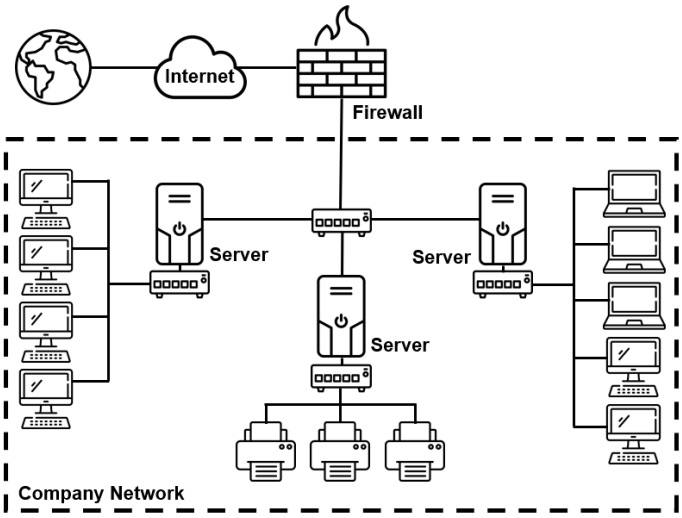

Fig. 2. Components in a company network (based on [12])

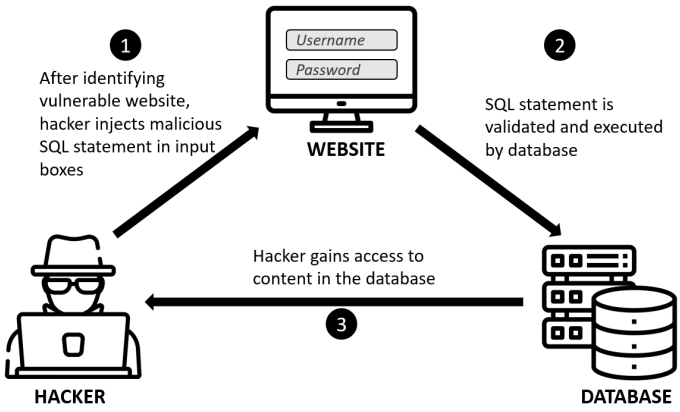

Fig. 3. Execution of a SQL injection (based on [15])

them. We won't mention the process of attacking digital weaknesses due to its complexity. After determining success or failure in the penetration, the tester will present his process. In case of a successful exploitation, the result of the network pen test would be that the company needs to update outdated software and close unused ports in the objects of their network. In other case the ethical hacker can confirm a safe state of the IT environment.

Finding open ports or outdated software were only two examples to detect vulnerabilities in a network. The pen tester uses these and much more to make the network of the company more secure.

But not only the networks of companies are the target of hackers. Web application security is very important for owning a safe environment. This is because every person can reach the website of the company through the Internet. One danger for web applications comes from so-called injection attacks. In an injection attack, the hacker uses an input box, like a log-in field to inject commands which lead to unwanted actions. The most common types of injection attacks are SQL injections. Properly executed and without strong IT security, the hacker can gain access to sensitive data from the companies' databases. Fig. 3 summarizes the process of a SQL injection attack. To prevent this scenario, the tester injects certain SQL commands by himself to demonstrate an attack. This requires a lot of expertise and caution because the content of databases is usually very valuable for the company. This process is not only done with SQL statements. There are many other attack types (XSS attacks, brute-force attacks, ...) to website input fields, which the pen tester needs to assure. Depending on the success or failure of the pen test, the ethical hacker can suggest upgrades to improve the security of the websites. [13, 14]

Besides the examples of network and web application security, electronic pen testing covers a very large area of content that cannot be covered all at once. Therefore, the tests are carried out only in case of innovations in certain aspects of the company, like newly installed software or changes in infrastructure [16].

B. Social Engineering

Social engineering means the usage of manipulation and psychological tricks on humans, to make them assist the offender's attack [17]. In case of a malicious attack the criminal can use different types of social engineering to get access to valuable information, like passwords and much more. The social engineering types can be categorized by the following:

1) Email
2) Voice Call
3) Face-to-Face
4) Text Message [17]

Most cyberattacks begin with social engineering [5]. Careless or good-faith employees are tricked to publish critical information, which will be used to build the attack around. One example of social engineering is email phishing. An email is sent externally to an employee asking them to click on a specific link. The careless employee does not know that a malware file is hiding behind the link, which is installed on the computer when clicked on the link. This malware could contain a backdoor that allows the offenders access to the computer. From that, they are able to browse through the companies' network and harm it. [17]

To ensure IT security in the company, generated emails by the ethical hacker can be sent to test the employees. Like real phishing emails, they ask the employee to click a link. Only in this case, they are directed to an educational website to inform them about cyber attacks and social engineering. To test the behavior of an employee in a fake call or face-to-face needs to be very respectful and safe. To measure the credibility in this way is very direct and personal, which could upset the employees and damage the trust towards their company. It might also be stressful for the employee, because of having to choose between helping a colleague and breaking the companies' policies. So it is recommended not to drive the employee to make unethic decisions in such a test. For this

reason, social engineering should only be carried out under precise conditions with specialists. [4, 5, 18]
Social engineering is a big threat to IT security. Penetration tests like phishing emails or training about cyberattacks for employees help to make the IT structure of the company a much safer environment.

C. Physical Testing

Physical tests are the last type of penetration testing. They deal with the functionality of the physical barriers in the company. Today's firewalls and other digital security systems are mostly on a high standard and hard to bypass. One alternative is to get direct data access by breaking through the physical boundaries of the company. A poor physical security system contains the risk of unauthorized intrusion into the company. This could lead to the theft of computers or hard drives with valuable information. There is also a possibility for penetration into more important rooms, such as the server room or offices with loged-in computers, which have direct access to the company network. [4]
Physical security is the basic requirement for a safe IT structure. To strenghen it, physical pen tests are used that can be very different. They reach from examining the waste for safety-related information (dumpster diving), to bypassing the security check, and even picking locks of doors [4, 19]. In a typical physical pen test, the tester checks the company buildings for adequate access controls. To gather information he could use the Internet to look for plans of company buildings or use social engineering to ask for vulnerable entries. After gaining knowledge about details of the company location, the tester tries to get into the company area. This could look like walking in the morning and getting in the middle of a crowd of people to bypass the guards. Inside the company, sensitive areas are checked for locked doors, cameras, or other security standards. [10]
In physical penetration tests, the definitions of the conditions in the test are very important. Necessary persons should be informed about the test to save the hired intruder. The tester should also be prepared for the possibility of accidental bodily harm from conventional barriers and weapons, interactions with animals and a lot more [20]. These types of tests are very rare and the organization should consider by itself, in which form physical penetration testing is needed.
Physical safety is the first big step in possessing a safe IT infrastructure. Monitoring and security systems should be installed in every facility that deals with important data.

IV. Conclusion

After penetration tests have been described in detail and the different types and approaches have been distinguished from one another, the most important findings are summarized below.
The diversity of different attack methods has shown, that there is not one type of test that fully secures a company and covers all vulnerabilities. Instead, the chosen test method should be adapted to the needs of the company to ensure an effective process.
Safety in the website and the network of the company are very important due to the fact, that hackers can attack these objects from every location in the world. The execution of an electronic pen test should be considered in case of innovations made. Even if a company has already dealt with the security of its systems, certain digital areas can often still be optimized. Especially when a single person has been dealing with these issues, the so-called blindness can occur, and important safety aspects can be overlooked. In these cases, a pen test by an external tester is a great way to improve or confirm IT security. [5]
Physical barriers and human recklessness should not be overlooked. Phishing emails and the installation of security standards are great ways to strengthen particular structures. However, the penetration test of these aspects should only be performed very carefully. The company needs to think about in which form the test is necessary to make a better environment.

References

[1] A. Unverzagt and C. Gips, "Rechtliches für Krisen-PR," in *Handbuch PR-Recht*, 2rd ed., Hamburg, Germany: Springer VS, 2018, pp.356.
[2] J. Johnson. "Amount of monetary damage caused by reported cybercrime to the IC3 from 2001 to 2020." statista.com. https://www.statista.com/statistics/267132/total-damage-caused-by-by-cyber-crime-in-the-us/#statisticContainer (accessed: May 11 2022).
[3] S. Morgan. "Cybercrime To Cost The World $10.5 Trillion Annually By 2025." cybersecurityventures.com. https://cybersecurityventures.com/hackerpocalypse-cybercrime-report-2016/ (accessed: May 11 2022).
[4] Bundesamt für Sicherheit in der Informationstechnik, Bonn, Germany. *Durchführungskonzept für Penetrationtests.* (2020). Accessed: May 12, 2022. [online]. Available: https://www.bsi.bund.de/SharedDocs/Downloads/DE/BSI/Publikationen/Studien/Penetrationstest/penetrationstest.pdf?
[5] Bundesamt für Sicherheit in der Informationstechnik, Bonn, Germany. *Ein Praxis-Leitfaden für IS-Penetrationstests.* (2016). Accessed: May 12, 2022. [online]. Available: https://www.bsi.bund.de/SharedDocs/Downloads/DE/BSI/Sicherheitsberatung/Pentest_Webcheck/Leitfaden_Penetrationstest.html
[6] S. Sugandh, M. B. Mehtre, "An overview of vulnerability assessment and penetration testing techniques," in *Journal of Computer Virology and Hacking Techniques*, Feb. 2014, vol. 11, no. 1, pp. 27-49, doi: 10.1007/s11416-014-0231-x
[7] A. G. Bacudio, X. Yuan, B. Bill Chu and M. Jones, "An overview of penetration testing," in *International Journal of Network Security & Its Applications (IJNSA)*, Nov. 2011, vol. 3, no. 6, pp. 19-38, doi: 10.5121/ijnsa.2011.3602
[8] K. Scarfone, M. Souppaya, A. Codey and A. Orebaugh, "Technical Guide to Information Security Testing and Assessment," in *Recommendations of the National Institute of Standards and Technology*, Sep. 2008, doi: 10.6028/NIST.SP.800-115
[9] P. Aar and A. K. Sharmar, "Analysis of Penetration Testing Tools," in *IJARCSSE (International Journal of Advanced Research in Computer Science and Software Engineering)*, Sep. 2017, vol. 7, no. 9, pp. 36-41, doi: 10.23956/ijarcsse.v7i9.408
[10] S. Fried, "Introduction to Penetration Testing," in *Penetration Testing Essentials*, Hoboken, NJ, USA: John Wiley & Sons Inc, 2017, pp. 1-13, doi: 10.1002/9781119419358.ch1
[11] C. N. Shivayogimath, "An Overview of Network Penetration Testing," in *IJRET (International Journal of Research in Engineering and Technology)*, Jul. 2014, vol. 3, no. 7, pp. 408-413
[12] Universal Computing Solutions, Inc. "Network Solutions." universal-cs.com. https://www.universal-cs.com/networks.html (accessed: May 11 2022).
[13] Z. S. Alwan, M. F. Younis, "Detection and Prevention of SQL Injection Attack: A Survey," in *IJCSMC (nternational Journal of Computer Science and Mobile Computing)*, Aug. 2017, vol. 6, no. 8, pp. 5-17

[14] S. P. Oriyano, ”Reporting,” in *Penetration Testing Essentials*, Indianapolis, Indiana, USA: John Wiley & Sons Inc, 2016, pp. 169-170.
[15] I. Abeythissa. ”Blind SQL Injection | Triggering Conditional Response | Part 1.” medium.com. https://isharaabeythissa.medium.com/blind-sql-injection-triggering-conditional-response-8b49c6f75512 (accessed: May 11 2022).
[16] F. Abu-Dabaseh and E. Alshammari, ”Automated Penetration Testing: An Overview,” in *Computer Science & Information Technology (CS & IT)*, 2018, pp. 121-129, doi: 10.5121/csit.2018.80610
[17] J. H. Bullee, M. Junger, ”Social Engineering,” in *Palgrave International Handbook of Cybercrime and Cyberdeviance*, Oct. 2021, pp. 849–875, doi: 10.1007/978-3-319-78440-3_38
[18] T. Dimkov, P. Hartel, W. Pieters, ”Two methodologies for physical penetration testing using social engineering,” in *Proceedings of the Annual Computer Security Applications Conference (ACSAC)*, Austin, Texas, USA, Jan. 2010, pp. 399-408, doi: 10.1145/1920261.1920319
[19] W. Allsopp, ”The Basics of Physical Penetration Testing,” in *Unauthorised Access: Physical Penetration Testing for IT Security Teams*, Chichester, West Sussex, United Kingdom: John Wiley & Sons Ltd, 2009
[20] P. Herzog, M. Barcelo. *OSSTMM3 - The Open Source Security Testing methodology Manual*.(2010). Accessed: May 15, 2022. [Online]. Available: https://www.isecom.org/OSSTMM.3.pdf

Comparison of VPN Technologies

Tobias Solisch
Faculty of Engineering and Information Technology
OTH Regensburg
Regensburg, Germany
tobias.solisch@st.oth-regensburg.de

***Abstract*—In the course of advancing globalization, the networking of different company locations is playing an increasingly important role. Digital solutions are required to ensure uniform data access and security in the transmission of data. Virtual private networks (VPNs) can be used to create a secure environment in which remote devices can communicate as if they were on the same network. Various approaches exist for setting up a virtual private network in an enterprise environment. This paper explains and compares different methods for setting up a virtual private network. The focus here is on VPN solutions in which the network service provider is responsible for setting up and managing the network. A further classification of VPN techniques is done according to the layers of the OSI model on which they operate. Data link layer and network layer VPNs are discussed in more detail. The selection of a suitable VPN depends on the application. The VPN models examined differ in their architecture and depending on what type of network is to be linked. Another difference between the VPN solutions is the management effort and the scalability. These aspects are summarized in the paper. Finally, the two VPN tunnel protocols Internet Protocol Security (IPSec) and Generic Routing Encapsulation (GRE) are introduced, showing that IPsec is suitable for the transmission of sensitive data and GRE for the transmission of unsupported protocols.**

***Index Terms*—VPN, PPVPN, L3VPN, L2VPN, Tunneling, IPsec, GRE**

I. Introduction

Virtual Private Networks (VPNs) have become increasingly popular in recent years. The global market has almost doubled in the last three years. The market has grown from 25.4 billion US dollars in 2019 to 44.6 billion US dollars in 2022. This is due to the increasing use of private networks in various industries. [1]
Many different VPN solutions are offered on the market, which can lead to confusion. Therefore, it is necessary to first understand what a Virtual Private Network is.
In [2] it is described as followed. "A VPN is [a] private network constructed within a public network infrastructure, such as the global Internet." More specifically, the VPN allows users to access a remote private network over the Internet as if they were directly connected to the private network. With a VPN connection, the data packets of a remote computer are sent via a VPN tunnel over the Internet to the company network. From the client's point of view, it, therefore, looks as if it is located in the LAN (local area network) of the company headquarters. [3]
The predecessors of today's VPNs were leased lines provided by a service provider (SP). This was the most common way to ensure a secure connection between different company sites. This method has some disadvantages. Both the hardware acquisition costs and the running costs are relatively high. Also the leased lines are inflexible about the amount of available bandwidth, because the customer has to choose between specified options of the service provider. This leads to poor scalability of the network. With VPNs using the service provider's network infrastructure, hardware, and operating costs for the customer and required network management from the service provider are reduced. [3], [4]
Over time, new VPN technologies have been developed. Each has its advantages and disadvantages. VPNs can be classified by the party, which is responsible for providing communication services [5]. According to this classification, there can be either Provider Provisioned VPNs (PPVPN) or Customer Provisioned VPNs (CPVPN). [5] In addition, VPNs can be further classified according to the OSI layer on which they are operating.
This paper provides an overview of the different VPN solutions focusing on PPVPNs and comparing selected VPN technologies based on their implementation and their purpose of use. Section II explains the different architectures of VPNs. In Section III a classification of PPVPNs is presented. In Section IV-VI different VPN technologies are explained according to their OSI layer. In Section VII the tunneling protocols IPsec and GRE are compared based on their functionality.

II. VPN Top View

Each VPN can either be classified as a Remote Access VPN or a Site-to-Site VPN based on the usage, regardless if provider or customer provisioned.
In a Site-to-Site VPN a connection between geographically remote LANs of an organization can be established over a public or private network infrastructure [3], [6]. Remote Access VPNs allow data transfer from a single point to a remote server [7]. Figure 1 shows the concept of a Remote Access and a Site-to-Site VPN.

An example of this would be an employee who has access to the company network from home office. The connection is established over a VPN tunnel. In a VPN tunnel the data is encapsulated with an extra added header [4]. VPN tunnels create separation between network traffic from other VPNs across the provider's network infrastructure and some tunneling protocols offer security [8]. The security aspect

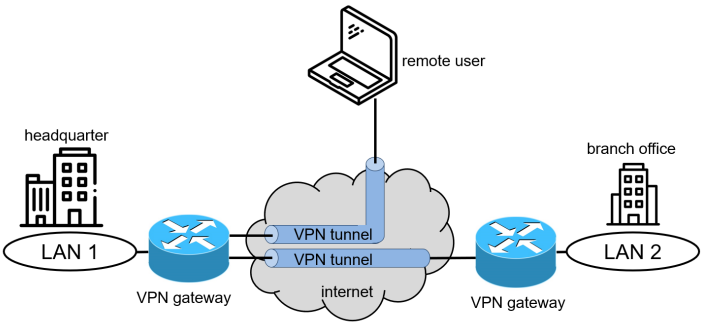

Fig. 1. Difference between Remote Access and Site-to-Site VPN

includes encryption, authentication, and integrity checking of data in the tunnels [4].

Figure 2 shows a reference model VPN. To establish a PPVPN, the customer needs access to the IP backbone. Therefore the customer requires at least one device which is connected to the service provider network. Such devices are referred to as Customer Edge (CE) devices. These provide an interface to the customer domain. The connection partner of the CE devices in the service provider network is called Provider Edge (PE) devices. In most cases the edge devices are routers, label switching routers, or IP switches [9]. The data transfer inside the provider's network is done by the Provider (P) devices. These do not provide VPN functionality and are only used for data transmission. [4]

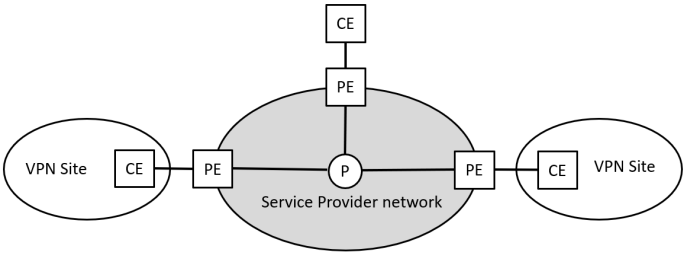

Fig. 2. Representation of the generic reference model of a VPN based on [8]

There are two ways to implement a VPN tunnel. One is to build a tunnel from CE to CE device. The other tunnel technique is from PE to PE device. The connection type between the individual P, PE, and CE devices can differ. For example the tunnels in the SP network between the P and PE devices can be established at Layer 1, Layer 2, or layer 3. [9]

III. PPVPN Classification

Various models of VPN classification are presented in the literature. In [9], VPNs are classified according to the OSI layer on which they operate. In [10] a distinction is made between trusted and secure VPNs. The classification in [5], [11] takes the OSI layer and the party responsible for providing communication service into account. Although this classification has a disadvantage in that it does not consider all the different VPN concepts, such as SSL VPNs, it was chosen for this paper because of the more precise classification.

In [5], [11] VPNs can be classified according to whether they are provider provisioned (PPVPNs) or customer provisioned (CPVPNs). In CPVPNs the customer can configure his own VPN independent of the network service provider. This can be achieved by supplying VPN software on the router of the customer (CE). [5]

In industry, PPVPNs are used for secure data transmission over public networks. In PPVPNs the provider manages the VPN service and is responsible for providing communication service. The customer does not take any part in network management and is therefore relieved. [5]

PPVPNs can be further classified according to which layer of the OSI model they operate on. PPVPNs can be deployed on the physical layer, the data link layer, and the network layer. [9]

A taxonomy of PPVPNs is presented in Figure 3. This shows a selection of different VPN solutions. In the following, we will take a look at the different VPN solutions on the layer, and present an overlook of their strengths and weaknesses, as well as their purpose of use. The solutions shown in gray are not described in detail in this paper.

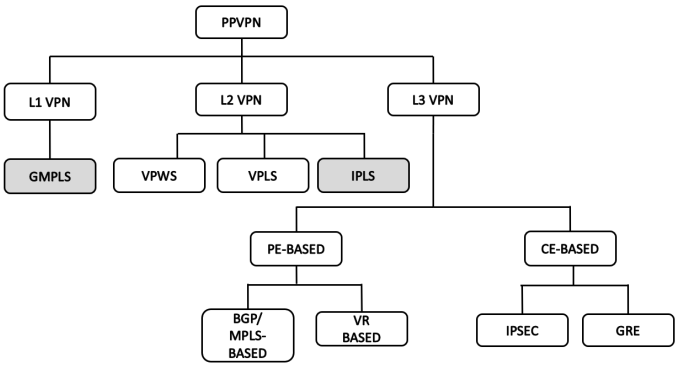

Fig. 3. Classification of PPVPNs based on [11]

IV. Layer 1 VPNs

Layer 1 VPNs (L1VPNs), which are also called Optical VPNs [12], provide physical layer connectivity between the customer sites. The service provider uses an optical network to transport the customer traffic. Devices like Time Division Multiplexing (TDM) switches, Optical Cross Connection (OXCs), or Photonic Cross Connects (PXCs) are used in Layer 1 networks [5]. For the control and management plane of L1VPNs, generalized multi-protocol label switching (GMPLS) is widely used [12].

V. Layer 2 VPNs

Layer 2 VPN (L2VPN) is a VPN service that operates on Layer 2. It provides a data link service between the devices of the customer belonging to the VPN and realizes Ethernet, Frame Relay, or ATM Virtual Circuits (VC). [8] Nowadays Multiprotocol Label Switching (MPLS) is used by most service providers, which operates on Layer 3. So Layer

2 connectivity between the customer sites is realized with a Layer 3 service provider backbone. Therefore Pseudo-Wires (PWs), which emulate a Layer 2 point-to-point connection over the provider's IP network [4], are needed. [13]
Service providers offer three different solutions to build a L2VPN. These are Virtual Private LAN Service (VPLS), Virtual Private Wire Service (VPWS), and IP only LAN like Service (IPLS) [14].

A. VPWS

VPWS enables a point-to-point connectivity between the customer sites. The service provider network functions as a set of emulated wires between the customer sites. The CE devices have to decide which PW to use to send data to another customer site. Therefore each CE device needs to be configured by the provider, which leads to an increase in management. [4] VPWS only works over an Ethernet, ATM, and Frame Relay backbone [15], except with the use of Any Transport over MPLS (AToM). VPWS with AToM can use a MPLS backbone. [13]
VPWS is useful for customers, who already use several ATM or Frame Relay connections between the various customer sites, as the existing connections between the customer and the provider can be used. Instead of transporting the data over an ATM or Frame Relay Service, the data is encapsulated and routed over the service provider's IP backbone, while using the same Layer 2 connection. This leads to a decrease in migration costs for the customer. VPWS is appropriate for migrating hub-and-spoke based networks where several branches must be connected to a single head office or data center. [4]
Figure 4 shows a simplified model of VPWS and VPLS.

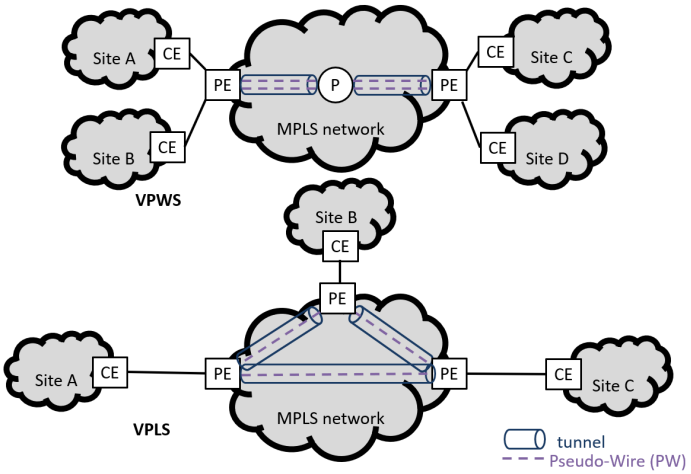

Fig. 4. Architecture of VPWS (top) and VPLS (bottom) based on [4]

B. VPLS

VPLS enables a multipoint-to-multipoint Ethernet connectivity between the customer sites. An MPLS or IP backbone can be used. [16] VPLS is used to connect Ethernets of companies with branches in various locations as if they were on the same LAN [5]. In VPLS the Ethernet LAN at each customer site is extended to the edge of the provider network [4]. Unlike VPWS, the CE device in a VPLS does not need to choose a specific PW. All the data is directly sent to the PE router. The providers network emulates an Ethernet Switch with ports to different Ethernet sites. The linking of these virtual Ethernet bridges are done by MPLS Pseudo-Wires. [4] The PE devices require MAC address learning to make forwarding decisions. VPLS can be also tuned to a hub-n-spoke topology. [15] VPLS provides a cost-effective and protocol independent service for setting up a Layer 2 VPN [17], [18]. High-speed connectivity over a Layer 3 provider's network and simple network management are advantages of VPLS [5]. A disadvantage of VPLS is scalability limitations in large networks due to MAC table explosion. [5]

VI. Layer 3 VPNs

Layer 3 VPN (L3VPN) is a VPN service that operates on Layer 3. It provides a Layer 3 service between the customer devices belonging to the VPN. [8] L3VPNs can be further classified into PE-based and CE-based Layer 3 VPNs. [4]

A. PE-based VPNs

In PE-based VPNs a tunnel is created between the PE devices. In this case the CE devices do not need any special VPN capabilities and can be a standard router. All the management and configuration take place in the PE devices. [19], [4]
For PE-based Layer 3 VPNs, the concepts of Virtual Router-based (VR VPN) and BGP/MPLS-based VPNs exist. These are examined in more detail below.

1) BGP/MPLS VPN: BGP/MPLS VPNs are Layer 3 VPNs where the service provider provides an MPLS-enabled IP backbone [20]. MPLS operates between the network and data link layer and is a framework for WAN [21]. For this type of VPN, a complete network between the PE devices is required, unless using the Border Gateway Protocol (BGP). This protocol allows PE routers to automatically set up the required tunnels. CE devices have a peer-to-peer connection to PE devices. Therefore each protocol can be used. If a data packet is sent from CE to PE, the PE device determines the route of the data packet from a forwarding table and adds an MPLS label to it. This label is inserted between the network layer header and the link layer header. The data packet reaches a remote PE device via a tunnel. This device recognizes the route based on the MPLS label and sends it on to the destination address. With MPLS, the data packet reaches its destination address faster than with traditional IP routing, because a router (PE) does not have to go through the entire routing table to find the next hop based on the destination address. [22]

2) VR VPN: Virtual Router (VR) VPNs operate on Layer 3. A virtual router, with all the capabilities of a physical router, is emulated on the service provider's PE device. One PE device can emulate multiple VRs. A virtual router has the same mechanisms and management tools as a physical router. From the CE device's point of view, the VR is a neighbor router of the router in the customer network. [23]

To establish a connection from one PE device within the VPN to another PE device within the same VPN, tunnels must be used in the provider network. Through the tunnels, the router can exchange routing information with any standard protocol. For large VPNs, manually laying out the tunnel network can be challenging, therefore the use of BGP is recommended. [4] In terms of management and scalability, VR VPNs are more complex than BGP/MPLS based VPNs, since for BGP/MPLS VPNs a single net of tunnels for the PE devices is sufficient. For VR VPNs, a separate tunnel overlay is required for each VPN. This increases the configuration effort for the routers. One advantage of VR over BGP/MPLS based VPNs is that the VPN routes are independent of the routes within the SP network. Errors in the BGP configuration would not affect the Internet connection. [4]

B. CE-based VPNs

In CE-based VPNs a tunnel is created between the CE devices. The backbone of the customer network is a set of tunnels whose endpoints are the CE devices. The CE devices encapsulate the data with a new IP header. [19] The service provider offers a simple IP service. CE-based VPNs provide point-to-point connectivity between the customer sites. Management gets more difficult the larger the VPN becomes due to maintaining the mesh of tunnels. [4] Tunnel implementation is done using tunnel protocols such as Internet Protocol Security (IPsec) and Generic Routing Encapsulation (GRE).

VII. Tunneling Protocols

In the following we will take a look at two different tunneling protocols.

A. IPsec

IPsec is a secure network protocol suite of open standards on the network layer. It is the most commonly used network security control and can provide several types of protection, including data encryption (confidentiality), data integrity, authenticating the origin of data, preventing packet replay and traffic analysis, and providing access protection. IPsec is used to establish a secure connection for Remote Access and Site-to-Site VPNs. [24] IPsec uses Authentication Header (AH), Encapsulating Security Payload (ESP) and Internet Key Exchange (IKE) protocols. The AH adds a header to each packet which is used for integrity and authentication of the transmitted data by using a hash algorithm, e.g. SHA2, SHA3. Confidentiality is implemented in IPsec with an ESP header using for example the AES128/256 encryption algorithms. ESP is the standard procedure for IPsec because it combines integrity, authentication, and encryption. In order to properly authenticate or decrypt the contents of packets, the Security Parameters Index (SPI) is used to specify to which Security Association (SA) this packet belongs. In an SA, a computer specifies (associates) for each destination, which keys and security settings should be applied to that destination. In order for SAs to be formed, the computers must be able to identify each other and choose a common encryption key. This process is summarized under the term IKE. [3]
IPsec can operate in tunnel or transport mode. In transport mode encryption and integrity protection for the payload of an IP packet is provided by the ESP. The integrity of the ESP Header is also protected. The most commonly used IPsec mode is ESP tunnel because it encrypts the original IP Header. The origin of the packet is disguised. [24]
Figure 5 shows the IPv4-packets with ESP in tunnel and transport mode.

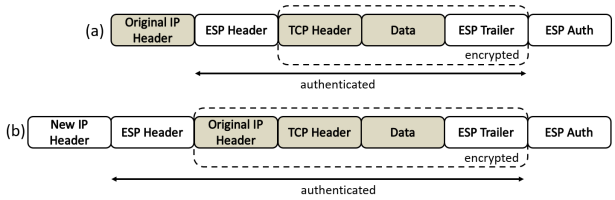

Fig. 5. IPv4-packet: (a) IPsec(ESP) transport mode, (b) IPsec(ESP) in tunnel mode based on [3]

B. GRE

GRE is a network layer encapsulation standard defined in RFC 2784 [25] by the Internet Engineering Task Force. The protocol encapsulates IP packets for transmission. GRE allows protocols a network normally does support. Therefore packets are wrapped in other packets that in turn use supported protocols. A GRE header and an IP header are added to the original packet. The GRE header specifies the protocol type used by the encapsulated packet. The IP header encapsulates the header and payload of the original packet. GRE can be used if IPv6-packets need to be stored in IPv4-packets because the network provider only supports IPv4. [26]
Unlike IPsec, GRE does not encrypt the transmitted data. There is also no authentication of the data, but A GRE tunnel can also be implemented over an IPsec tunnel to combine the advantages of both tunneling protocols. [27]

VIII. Conclusion

After classifying and describing the different VPN technologies and tunneling protocols in detail, the most important findings are summarized below. Since there are so many different VPN technologies it is quite a challenge to classify them. With the classification from [11] a fine subdivision is possible, as it distinguishes between layers as well as who is responsible for the communication service and, for Layer 3 VPNs, which devices create a tunnel. This classification also shows that the choice of the right VPN technology and tunneling protocol depends on the application. For Layer 2 and Layer 3 connectivity between customer sites, the size of the VPNs is what matters most, as the individual concepts differ in terms of management effort and scalability. For the tunnel protocols considered, the use case is also decisive. While GRE offers a fast transmission of all protocols, the security aspect in IPsec is the most important. Sensitive data should therefore only be transmitted with a tunneling protocol that offers encryption of the data, like IPsec.

REFERENCES

[1] J. A. Sava, "Size of the virtual private network (VPN) market worldwide from 2019 to 2027." statista.com. https://www.statista.com/statistics/542817/worldwide-virtual-private-network-market/ (accessed May 15, 2022)

[2] P. Ferguson and G. Huston "What is a VPN?," Apr. 1998

[3] S. Dehn, "Virtual Private Network," in *Netzwerke Sicherheit*, Bodenheim, Germany: Herdt, 2021, ch. 16, pp. 185-193

[4] M. Finlayson, J. Harrison and R. Sugarman, "VPN Technologies-a comparison," in *Data Connection Limited*, Feb. 2003

[5] K. Gaur, A. Kalla, J. Grover, M. Borhani, A. Gurtov and M. Liyanage ,"A Survey of Virtual Private LAN Services (VPLS): Past, Present and Future," in *Computer Networks*, vol. 196, Jun. 2021, doi: 10.1016/j.comnet.2021.108245

[6] S. T. Aung and T. Thein, "Comparative Analysis of Site-to-Site Layer 2 Virtual Private Networks," in *2020 IEEE Conference on Computer Applications(ICCA)*, 2020, pp. 1-5, doi: 10.1109/ICCA49400.2020.9022848

[7] R. Bibraj, S. Chug, S. Nath and S. Singh "Technical study of remote access VPN and its advantages over site to site VPN to analyze the possibility of hybrid setups at radar stations with evolving mobile communication technology," in *MAUSAM*, vol. 69, no. 1, Jan. 2018, pp. 97-102

[8] INTERNATIONAL TELECOMMUNICATION UNION "Network-based VPNs – Generic architecture and service requirements," in *SERIES Y: GLOBAL INFORMATION INFRASTRUCTURE AND INTERNET PROTOCOL ASPECTS*, Y.1311, Mar. 2002

[9] Z. Zhang, Y. Zhang, X. Chu and B. Li, "An Overview of Virtual Private Network (VPN): IP VPN and Optical VPN," in *Photonic Network Communications 7*, May 2004, pp. 213-225, doi: 10.1023/B:PNET.0000026887.35638

[10] A. A. Jaha, F. B. Shatwan and M. Ashibani, "Proper Virtual Private Network (VPN) Solution," in *2008 The Second International Conference on Next Generation Mobile Applications, Services, and Technologies*, Sep. 2008, pp. 309-314, doi: 10.1109/NGMAST.2008.18

[11] L. Andersson and T. Madsen, "Provider provisioned virtual private network (VPN) terminology," RFC 4026, The Internet Society, Mar. 2005

[12] Y. SU, Y. Tian, E. Wong, N. Nadarajah and C. Chan "All-optical virtual private network in passive optical networks," in *Laser & Photonics Reviews*, vol. 2, no. 6, Sep. 2008, pp. 460-479, doi: 10.1002/lpor.200810021

[13] G. Singh and Er. M. Moudgil, "Comparative Analysis of MPLS Layer 2 VPN Techniques," in *International Journal of Computer Science Trends and Technology (IJCST)–Volume*, vol. 3, Jul./Aug. 2015

[14] W. Augustyn and Y. Serbest, "Service Requirements for Layer 2 Provider-Provisioned Virtual Private Networks," RFC 4665, The Internet Society, Sep. 2006

[15] L. Andersson and E. Rosen, "Framework for Layer 2 Virtual Private Networks (L2VPNs)," RFC 4664, The Internet Society, Sep. 2006

[16] X. Dong, S. Yu "VPLS: an effective technology for building scalable transparent LAN services," in *Network Architectures, Management, and Applications II*, S. J. B: Yoo, G. Chang, G. Li and K. Cheung, Eds., vol. 5626, Feb. 2005, pp. 137-147, doi: 10.1117/12.573606

[17] M. Liyanage, "Enhancing security and scalability of Virtual Private LAN Services," Ph. D. dissertation, University of Oulu, Finland, Sep. 2016. [Online]. Available: http://jultika. oulu. fi/Record/isbn978-952-62-1376-7.

[18] M. Liyanage, M. Ylianttila and A. Gurtov, "Enhancing Security, Scalability and Flexibility of Virtual Private LAN Services," in *2017 IEEE International Conference on Computer and Information Technology (CIT)*, Sep. 2017, pp. 286-291, doi: 10.1109/CIT.2017.45

[19] R. Callon, "A Framework for Layer 3 Provider-Provisioned Virtual Private Networks (PPVPNs)," RFC 4110, The Internet Society, Jul. 2005

[20] E. Rosen and Y. Rekhter, "BGP/MPLS VPNs", RFC 2547, The Internet Society, Mar. 1999

[21] R. Bush and T. G. Griffin, "Integrity for virtual private routed networks," in *IEEE INFOCOM 2003. Twenty-second Annual Joint Conference of the IEEE Computer and Communications Societies (IEEE Cat. No. 03CH37428)*, vol. 2, Jul. 2003, pp. 1467-1476, doi: 10.1109/INFCOM.2003.1208982

[22] K. N. Quershi, A. H. Abdullah, A. N. Hassan, D. K. Sheet and R. W. Anwar, "Mechanism of Multiprotocol Label Switching for Forwarding Packets & Performance in Virtual Private Network," in *Middle-East Journal of Scientific Research*, vol. 20, no. 12,Dec. 2014, pp. 2117-2127, doi: 10.5829/idosi.mejsr.2014.20.12.21101

[23] P. Knight and C. Lewis, "Layer 2 and 3 virtual private networks: taxonomy, technology, and standardization efforts," in *IEEE Communications Magazine*, vol. 42, no. 6, Jun. 2004, pp. 124-131, doi: 10.1109/MCOM.2004.1304248

[24] S. Frankel, K. Kent, R. Lewkowski, A. D. Orebaugh, R. W. Ritchey and S. R. Sharma, "Guide to IPsec VPNs," Dec. 2005

[25] D. Farinacci, T. Li, S. Hanks, D. Meyer and P. Traina "Generic Routing Encapsulation (GRE)," RFC 2784, The Internet Society, Mar. 2000

[26] Cloudflare, Inc., "Was ist GRE-Tunneling? — Wie das GRE-Protokoll funktioniert." cloudflare.com. https://www.cloudflare.com/de-de/learning/network-layer/what-is-gre-tunneling/ (accessed May 15, 2022)

[27] A. M. Abdulazeez, B. Salim, D. Zeebaree and D. Doghramachi "Comparison of VPN Protocols at Network Layer Focusing on Wire Guard Protocol," International Association of Online Engineering, Nov. 2022, [Online]. Available: https://www.learntechlib.org/p/218341/

Operation and Suitability of Current Software Encryption Methods for Microcontroller-Systems

Garschhammer Kilian
Fakultät Elektro- und Informationstechnik
Ostbayerische Technische Hochschule
Regensburg, Deutschland
kilian.garschhammer@st.oth-regensburg.de

***Abstract*—In modern times, the demand for security and privacy is constantly increasing, while various low power devices, mainly IoT technologies and products, rely more on communication over open networks like the internet. This creates the requirement for a secure encryption method which can be implemented on embedded systems with low computing power. Since the encryption often must be performed in real time, performance is critical to not create a bottleneck for the communication. The hardware limitations such as the size of available memory and the integer lengths used by the processor may impact different encryption methods in varying degrees, depending on the algorithms used. To find the most fitting method, it must be examined how those limitations influence the performance as well as how sophisticated implementations can mitigate them. With these factors in consideration, the suitability of current encryption methods must be evaluated for usage on microcontroller-systems. While a hardware implementation of the encryption would resolve the performance issues, including separate hardware reduces flexibility and may increase the cost of devices. Thus, only the performance of software implementations is discussed in this article. In the following, the most prevalent symmetric and asymmetric encryption methods are analyzed in terms of how they operate, how secure they are and how they can be efficiently implemented on devices with limited performance and memory. Based on these results and benchmarks of existing implementations, the different encryption methods are compared and evaluated for this use case.**

***Index Terms*—cryptography, embedded, microcontroller, edge cryptography**

I. Introduction

Different encryption schemes can vary heavily both in performance and achieved security. In order to find which algorithm is optimal depending on the application the most prevalent encryption algorithms are investigated in this paper. First, the general operation of some of the most widely used symmetric and asymmetric encryption methods, namely AES, 3DES, XTEA and RSA, is described. Further, techniques in order to improve their performance, especially of software implementations for low power microcontroller-systems, are examined. Finally the different algorithms are compared in terms of their performance and security in order to determine their suitability for usage on microcontrollers. It is expected that performance and security widely varies across the different cryptosystems. This publication tries to examine those properties of the different algorithms to give an insight into their suitability for future applications in low performing systems.

II. Operation of Encryption Methods

In the following, the general operation of the algorithms, which are examined in this article, is presented.

A. Rivest-Shamir-Adleman

Rivest-Shamir-Adleman (RSA) is a public-key cryptographic system. These cryptosystems, also known as asymmetric cryptosystems, use a public key to encrypt messages, which can then be decrypted using a private key.

In RSA the cyphertext c is computed using the following equation [1]:

$$c = m^e \bmod N \tag{1}$$

Here, e is the encryption exponent, m is the padded plaintext and N is some large value. Both e and N are shared as the public key <N,e>.

Since an attacker is able to know all variables in the encrypting equation except for the padded plaintext, it is theoretically possible to inverse this function. But as long as the key size is large enough, typically 2048 bits or larger, calculating the inverse function needs more computational power than what is feasable with modern processors.

The decryption is calculated as follows, with d being the private key:

$$m = c^d \bmod N \tag{2}$$

B. Advanced Encryption Standard

The Advanced Encryption Standard (AES) is a fixed block length version of Rijndael, a block cipher developed by Joan Daemon and Vincent Rijmen. It uses a length of 128 bits for input block, output block and a 4x4 matrix called the state. The cypher key can have a length of 128, 192 or 256 bits.

The operation of AES is made up of four basic transformations: The Substitute Bytes transformation, the Shift Rows transformation, the Mix Columns transformation and the Add Round Key transformation.

- In the Byte Substitution, each byte of the state is replaced according to a lookup table.
- In the Shift Rows Transformation, the bytes in the last three rows of the state are cyclically shifted to the left

over different offsets. The second row is shifted 1 byte, the third 2 bytes and the fourth 3 bytes.
- The Mix Columns Transformation treats each column as a four-term polynomial and multiplies them modulo x^4+1 with the fixed polynomial a(x) = $3x^3$ + $1x^2$ + 1x + 2.
- In the Add Round Key Transformation, a Round Key is added to the State by a bitwise XOR operation.

The encryption starts with an initial Add Round Key transformation. After that, the Substitute Bytes, Shift Rows, Mix Columns and Add Round Key operations are executed sequentially and repeated N_r times. N_r is derived from the key length and equals 10 with a 128-bit key, 12 with a 196-bit key and 14 with a 256-bit key. In the final loop, the Mix Columns operation is discarded [2].

C. Triple Data Encryption Standard

The Data Encryption Standard (DES) was first published in 1977 as an encryption standard using a 56-bit key and 64-bit blocklength. In the encryption the following steps are repeated 16 times, with a different 48-bit subkey for each round. The half-blocks are swapped between each round and processed alternately [3]. Such a process is called a Feistel function.

- Split the data into two 32-bit half blocks and expand the right half-block to 48 bits
- Use an XOR operation on the 48-bit expanded half-block with a 48-bit sub-key, which is derived from the main-key
- Split the result into 8 6-bit blocks and replace each of these blocks with 4 bits according to a lookup table
- The 32 output bits are permutated according to a fixed permutation and then an XOR operation is used with the left half-block

Because this encryption method used a 56-bit key, it became feasable to break it with brute force attacks as computational power increased with time. To circumvent this, Triple Data Encryption Standard (3DES) was developed as an improvement of this encryption scheme, using a triplet of 56-bit keys to achieve up to 168-bit security, by first encrypting with one key, then decrypting the data with the second key and again encrypting with the third key.

D. eXtended Tiny Encryption Algorithm

The eXtended Tiny Encryption Algorithm (XTEA) is an improvement of the Tiny Encryption Algorithm (TEA) developed to diminish its vulnerability against related key attacks and to increase its effective key length from 126 to 128 bits [4]. TEA was designed similarly to DES, using Feistel iterations to encrypt the data, with a 64 bit block length and a key size of 128 bits. But instead of using a complex algorithm, it does a weak non linear iteration for a higher number of rounds to make it secure [5]. This allows for a significantly smaller program, resulting in a reduced memory footprint.

The encryption algorithm is shown in the following routine:

```
void encipher(long * v, long * k, long N) {
    unsigned long y=v[0], z=v[1], DELTA=0x9e3779b9;
    unsigned long limit=DELTA*N, sum=0;
    while (sum!=limit) {
        y += (z<<4 ^ z>>5) + z ^ sum + k[sum & 3];
        sum += DELTA;
        z += (y<<4 ^ y>>5) + y ^ sum + k[sum>>11 & 3];
    }
    v[0]=y, v[1]=z;
}
```

During each iteration of the loop, two Feistel iterations are computed by adding a permutation of the right half block integrating a changing sum value as well as 4 bytes of the key to the left half block and vice versa.

The decryption is a simple inversion of this procedure.

III. Optimization Methods accomodating Microcontroller Limitations

A microcontroller has many technical limitations. To accommodate those, different optimization techniques were developed. Generally, encryption and decryption involves calculations with values of bit-lengths $\geq$ 64-bits. These values are generally represented as arrays of values with shorter bit lengths. To reduce the amount of processor cycles, those values should be adjusted to the wordlengths of the microcontroller, e.g. 8-bit values on an 8-bit microcontroller and 16-bit on a 16-bit microcontroller [6].

Further, optimization techniques were developed, which are specific to different cryptosystems.

As DES and subsequently 3DES are intended for hardware implementations only little research was done to improve software implementations.

XTEA already defines a specific implementation and its simplicity limits further performance improvements.

A. Rivest-Shamir-Adleman

The modular exponentiations used in RSA make use of very large integers with a typical size of at least 1024 bits. Generally, RSA decryption performs significantly worse than encryption, with up to 40 times slower decryption speeds [7]. Thus, it is desired to reduce decypher times, even if it may be detrimental to encypher performance [8].

To reduce the computation costs of these calculations, algorithms like the Montgomery Multiplication [7][9], Modular Exponentiation [10] or Hybrid Multiplication [7] can be used. [11] presents a highly efficient implementation for 8-bit microcontrollers of the AVR family, which makes use of an optimized variant of the hybrid multiplication to reduce the overall number of mov or movw instructions.

Another variant of RSA, Multi-factor RSA [12], makes use of a modulus in the form N = p^2q. This allows to decrypt the data using the Chinese Remainder Theorem (CRT). This is achieved by defining d_i = d mod (p_{i-1}). The cyphertext can then be decrypted by calculating:

$$m_i = c\, d_i \bmod (p_i) \quad \text{for each } i, 1 \leq i \leq b \tag{3}$$

Combining the m_i values using the CRT results in $m \equiv c^d \bmod N$. In 1024-bit RSA, this allows a speedup of approximately 2.25 over standard RSA. Alternatively, a method

based on the Quisquater-Couvreur method can be used, using multiple exponentiations with a smaller exponent to speed up decryption [8].

B. Advanced Encryption Standard

Modern implementations, like FACE [13], a fast implementation of AES-CTR mode encryption, cache repetitive data contained in the state and reuse them in later stages of the encryption to save up on required calculations. FACE-LIGHT is an improved variant of this implementation intended for use on low-end microcontrollers like an 8-bit microcontroller. Here, tables are precomputed before encryption. Since SRAM is limited on low-power microcontrollers, these may be stored in program memory [14]. The AES implementation presented in [6] employs 4 precomputed lookup tables to combine the Byte Substitution, Shift Rows and Mix Columns transformations into a single step to reduce the amount of processor cycles. D. A. Osvik et. al. Also presented a fast software AES encryption implementation, also using table precomputation to speed up the procedure [15].

IV. Performance evaluation

Based on different benchmarks, the performance of the different algorithms will be compared.

[16] showed, that AES using the Electronic Codebook (ECB) mode, where each block is encrypted seperately, performs up to twice as fast as 3DES. When using the Cipher Feedback (CFB) mode, which integrates the preceding encrypted block during encryption to improve security, performance of both showed comparable results. Another experiment presented in [17] showed up to 8 times faster computation when comparing AES with 3DES. When compared to other block cyphers not examined in this article, AES generally performs significantly better and usually needs less or an equal amount of memory [18][19].

In a performance analysis published by Dr. P. Mahajan and A. Sachdeva RSA also performed significantly worse, requiring up to 5 times as long as AES both when encrypting and decrypting the data [20]. Also, [21] found, that RSA can require up to 50% more memory when compared with AES or DES.

While TEA and XTEA can reduce the memory footprint, using only around 1150 bytes compared to 3410 bytes of memory allocated by AES, XTEA and TEA require around 1.5 the amount of processor cycles both for encryption and decryption [19].

In general, AES requires the lowest number of processor cycles when compared with RSA, XTEA and 3DES [19][18]. XTEA requires around 1.5 times the computation time AES requires. RSA and 3DES are the slowest algorithms, requiring around 5 to 8 times the amount of time AES needs.

V. Security evaluation

To correctly evaluate the different encryption algorithms, their security must be examined. For this it must be examined, which forms of attacks are possible on the different cryptosystems.

Every cryptographic system is vulnerable to a brute forced attack. In such an attack, the attacker tries to guess the key until the data can successfully be decrypted. To diminish the risk of an attacker finding the key, a long enough key has to be used. When using e.g. a 256-bit key, the risk of such an attack to succeed becomes extremely low and can be disregarded [20]. Since the generation of the key-pair in RSA makes use of prime numbers, the number of possible key values is reduced dramatically. To accommodate that, significantly larger key sizes are required.

Since 3DES makes use of the same encryption algorithm used in DES, it inherits some of its vulnerabilities. As explained in [22], 3DES is vulnerable to differential and linear cryptanalysis and has weak substitution tables while AES is considered unbreakable in practical use. [23] presents a possible attack making use of the vulnerabilities of iterated cryptosystems like DES or 3DES.

According to [20] RSA is, besides a brute forced attack, vulnerable to an oracle attack. Such attacks make use of the padding validation used in rsa to decrypt the ciphertext. It relies on a ”padding oracle”, which reveales if a decrypted message is correctly padded or not. [24] shows that many devices return errors when the decrypted data is not correctly padded, this information can be used to implement the attack. In their tests, an average of 49,000 oracle calls was sufficient to decrypt an unknown ciphertext under a 1024 bit key.

XTEA is vulnerable to both related-key attacks [25] as well as differential cryptanalysis [26]. But only with a reduced number of rounds employed during the encryption, successful

TABLE I
Comparison of encryption methods

	RSA	AES	3DES	XTEA
Cryptosystem	Asymmetric	Symmetric	Symmetric	Symmetric
Key length	$\geq$1024 bits	128, 192 or 256 bits	168 bits	128 bits
Performance	slow	fast	intermediate	intermediate
Memory footprint	high	intermediate	intermediate	low
Vulnerabilities	brute forced attack oracle attack	brute forced attack	brute forced attack differential and linear cryptanalysis	brute forced attack related-key attacks, differential cryptanalysis
Security	intermediate	excellent	intermediate	low

attacks were performed in the past [25][27]. When using the suggested cyclecount of 32, which is an equivalent of 64 rounds, this cryptographic system is still fairly secure.

According to these results, AES is the most secure cryptosystem when implemented properly, followed up by 3DES. Because XTEA employs an insecure Feistel-function and only uses a key-length of 128-bits, its fairly insecure. RSA is also insecure because oracle attacks can be executed easily.

VI. Discussion

Table 1 shows a summary of the previous results. AES showed the best performance and the highest security, rendering it the generally most suitable encryption method for low power microcontrollers.

3DES lacks behind in performance since its based on the less modern DES encryption scheme while also being primarily designed for hardware implementations [22]. Because of its limitation of a key length of 168 bits and its higher vulnaribilites it is considered outdated compared to modern algorithms, even though its still widely used at the present day.

RSA performed the worst compared to the other methods. But the main usage of asymmetric cryptosystems like RSA is to exchange a key over an open connection which can then be used to encrypt further bulk data using a symmetric encryption method.

[19] also discussed other light-weight encryption algorithms as well as XTEA. These showed a significantly lower memory footprint compared to AES, but were slower and had significant security vulnerabilities.

Comparing the results presented in this article with similar publications, the results are quite comparable [18][20]. Generally, AES achieves the highest throughput when compared with software implementations of other encryption schemes while being considered unbreakable when properly implemented.

VII. Conclusion

The cryptosystems RSA, AES, 3DES and XTEA have been examined and their suitability for software implementations on microcontroller-systems were studied. For this, the general operation of the encryption algorithms was presented and further how modern implementations try to reduce the amount of required processor cycles for both encyphering and decyphering. With these results in mind, the encryption algorithms were compared while further investigating the security of those algorithms.

It has been shown that AES offers both the highest security as well as the best performance. But with its extremely low memory footprint, XTEA also offers a viable alternative for systems where memory capacity is critically low.

This work offers an important insight into the characteristics of different encryption algorithms and thus can help when deciding on a encryption scheme to be used in newly developed low power applications.

References

[1] K. Moriarty, B. Kaliski, J. Jonsson, and A. Rusch, "PKCS #1: RSA cryptography specifications version 2.2," RFC Editor, RFC 8017, Nov. 2016, Backup Publisher: RFC Editor ISSN: 2070-1721 Published: Internet Requests for Comments.

[2] M. Dworkin, E. Barker, J. Nechvatal, *et al.*, *Advanced encryption standard (AES)*, Nov. 26, 2001. DOI: https://doi.org/10.6028/NIST.FIPS.197.

[3] N. I. o. Standards and Technology, "DATA ENCRYPTION STANDARD (DES)," U.S. Department of Commerce, Washington, D.C., Federal Information Processing Standards Publications (FIPS PUBS) 46-3, Change Notice 3 October 25, 1999, 1999.

[4] D. J. Wheeler and R. M. Needham, "Tea extensions," 1997.

[5] D. J. Wheeler and R. M. Needham, "TEA, a tiny encryption algorithm," in *Fast Software Encryption*, B. Preneel, Ed., Berlin, Heidelberg: Springer Berlin Heidelberg, 1995, pp. 363–366, ISBN: 978-3-540-47809-6.

[6] C. Gouvêa and J. López, *High Speed Implementation of Authenticated Encryption for the MSP430X Microcontroller*. Oct. 7, 2012, 288 pp., Pages: 304, ISBN: 978-3-642-33480-1. DOI: 10.1007/978-3-642-33481-8_16.

[7] N. Gura, A. Patel, A. Wander, H. Eberle, and S. Shantz, *Comparing Elliptic Curve Cryptography and RSA on 8-bit CPUs*. Aug. 11, 2004, vol. 3156, 119 pp., Journal Abbreviation: Lect Notes Comput Sci Pages: 132 Publication Title: Lect Notes Comput Sci, ISBN: 978-3-540-22666-6. DOI: 10.1007/978-3-540-28632-5_9.

[8] C. Paixão, "An efficient variant of the RSA cryptosystem.," *IACR Cryptology ePrint Archive*, vol. 2003, p. 159, Jan. 1, 2003.

[9] P. L. Montgomery, "Modular multiplication without trial division," *Mathematics of Computation*, vol. 44, no. 170, pp. 519–521, 1985.

[10] C. Kaya Koc, *High-Speed RSA Implementation*. Nov. 1994.

[11] Z. Liu, J. Großschädl, and I. Kizhvatov, "Efficient and side-channel resistant RSA implementation for 8-bit AVR microcontrollers," Jan. 1, 2010.

[12] D. Boneh and H. Shacham, "Fast variants of RSA," vol. 5, Aug. 29, 2002.

[13] J. Park and D. Lee, "FACE: Fast AES CTR mode encryption techniques based on the reuse of repetitive data," *IACR Transactions on Cryptographic Hardware and Embedded Systems*, pp. 469–499, Aug. 16, 2018. DOI: 10.46586/tches.v2018.i3.469-499.

[14] K. Kim, S. Choi, H. Kwon, Z. Liu, and H. Seo, "FACE–LIGHT: Fast AES–CTR mode encryption for low-end microcontrollers," in Feb. 13, 2020, pp. 102–114, ISBN: 978-3-030-40920-3. DOI: 10.1007/978-3-030-40921-0_6.

[15] D. A. Osvik, J. Bos, D. Stefan, and D. Canright, *Fast software AES encryption*. Feb. 7, 2010, vol. 6147, 75 pp., Pages: 93, ISBN: 978-3-642-13857-7. DOI: 10.1007/978-3-642-13858-4_5.

[16] A. Nadeem and M. Javed, *A Performance Comparison of Data Encryption Algorithms*. Sep. 27, 2005, 84 pp., Journal Abbreviation: IEEE Information and Communication Technologies Pages: 89 Publication Title: IEEE Information and Communication Technologies, ISBN: 0-7803-9421-6. DOI: 10.1109/ICICT.2005.1598556.

[17] S. O. A. F. M. Koko, D. Babiker, and N. Mustafa, "Comparison of various encryption algorithms and techniques for improving secured data communication," *IOSR Journal of Computer Engineering*, vol. 17, no. 1, pp. 62–69, Jan. 2015, ISSN: 2278-8727.

[18] M. Çakıroğlu, "Software implementation and performance comparison of popular block ciphers on 8-bit low-cost microcontroller," *International Journal of the Physical Sciences*, vol. 5, pp. 1338–1343, Sep. 18, 2010.

[19] S. Rinne, T. Eisenbarth, and C. Paar, "Performance analysis of contemporary light-weight block ciphers on 8-bit microcontrollers," 2007.

[20] D. P. Mahajan and A. Sachdeva, "A study of encryption algorithms AES, DES and RSA for security," *Global Journal of Computer Science and Technology Network, Web & Security*, vol. 13, no. 15, pp. 14–22, 2013, ISSN: 0975-4350.

[21] B. Padmavathi and S. Ranjitha Kumari, "A survey on performance analysis of DES, AES and RSA algorithm along with LSB substitution technique," *International Journal of Science and Research (IJSR)*, pp. 170–174, 2013, ISSN: 2319-7064.

[22] N. Aleisa, "A comparison of the 3des and AES encryption standards," *International Journal of Security and Its Applications*, vol. 9, pp. 241–246, Jul. 31, 2015. DOI: 10.14257/ijsia.2015.9.7.21.

[23] S. K. Langford and M. E. Hellman, "Differential-linear cryptanalysis," in *Advances in Cryptology — CRYPTO '94*, Y. G. Desmedt, Ed., Berlin, Heidelberg: Springer Berlin Heidelberg, 1994, pp. 17–25, ISBN: 978-3-540-48658-9.

[24] R. Focardi, "Practical padding oracle attacks on RSA," *Hakin9 - Defend Yourself! Hands-on Cryptography*, Sep. 2012.

[25] J. Lu, "Related-key rectangle attack on 36 rounds of the XTEA block cipher," *Int. J. Inf. Sec.*, vol. 8, pp. 1–11, Feb. 1, 2009. DOI: 10.1007/s10207-008-0059-9.

[26] S. Hong, D. Hong, Y. Ko, D. Chang, W. Lee, and S. Lee, *Differential cryptanalysis of TEA and XTEA*. Nov. 27, 2003, vol. 2971, 402 pp., Pages: 417. DOI: 10.1007/978-3-540-24691-6_30.

[27] Y. Ko, S. Hong, W. Lee, S. Lee, and J.-S. Kang, *Related key differential attacks on 27 rounds of XTEA and full-round GOST*. Feb. 5, 2004, vol. 3017, 299 pp., Journal Abbreviation: FSE 2004 Pages: 316 Publication Title: FSE 2004, ISBN: 978-3-540-22171-5. DOI: 10.1007/978-3-540-25937-4_19.

Bot-Netzwerke: Was ist das und wozu eigentlich? Was hilft?

Andreas Kammerl
Fakultät Elektro- und Informationstechnik
Ostbayerische Technische Hochschule
Regensburg, Germany
andreas.kammerl@st.oth-regensburg.de

***Zusammenfassung*—Das Internet ist eine Technologie, welche vielerlei Gefahren für sowohl Unternehmen als auch Privatpersonen birgt. Eine sehr vielseitige und oftmals unterschätzte Gefahr bilden dabei Bot-Netzwerke (eng.: Botnet). Durch die Angriffe können für die Unternehmen finanzielle Schäden in Millionenhöhe verursacht werden. Diese Arbeit dient dazu ein besseres Bewusstsein für diese Art der Gefährdung der heutigen Industrie und Gesellschaft zu schaffen. Private Geräte können dabei von Kriminellen für Ihre illegalen Aktivitäten missbraucht werden ohne dass sich die betroffenen Personen dessen bewusst sind. Es können nicht nur PCs oder Laptops für solch illegale Aktivitäten genutzt werden, sondern auch unscheinbare Objekte aus dem Internet of Things wie zum Beispiel ein mit dem WLAN verbundener Saugroboter. Die Netzwerke kommen dabei in verschiedenen Topologien vor und bilden ein Commander/Responder Verhältnis. In der Arbeit wird im Weiteren darauf eingegangen, welche Gefahren durch die zweckentfremdeten Geräte für Außenstehende sowie den Besitzer des infiltrierten Gerätes entstehen können. Die Netzwerke werden für verschiedenste Aktivitäten verwendet, weshalb hier nur auf einige wenige eingegangen wird. Zusätzlich wird der aktuelle Stand der Technik betrachtet hinsichtlich der Erkennung solcher Bot-Netzwerke und verschiedener präventiver Schutzmöglichkeiten. Diese sollen Personen helfen zu verhindern, dass die eigenen Geräte mit Schadsoftware befallen und missbraucht werden. Ebenfalls wird darauf eingegangen, welche Möglichkeiten es gibt Bot-Netzwerke ausfindig zu machen. Diese können sehr kostspielig ausfallen und haben deshalb Unternehmen als Zielgruppe. Privatpersonen sind kaum Ziel solcher Angriffe weshalb bei Ihnen der Schutz der Geräte vor Infiltrierung im Vordergrund stehen sollte.**

***Index Terms*—Botnet, Network security, DDoS, Internet of Things, DDoS prevention**

I. Einführung

In der Welt des Internets lauern viele Gefahren für die moderne Gesellschaft. Eine immer größer werdende Rolle spielen dabei Bot-Netzwerke [1]. Diese werden vorwiegend für kriminelle Aktionen verwendet [3]. Die schwerwiegendsten, sogenannte „Distriputed Denial of Service“ (DDoS) Attacken richten sich dabei beinahe ausschließlich gegen Server [2]. Privatpersonen sind sich oftmals nicht bewusst, dass Sie bzw. Ihre Geräte für die Machenschaften solcher Krimineller missbraucht werden [1]. Die Gründe für das fehlende Verständnis der Bevölkerung für diese Thematik liegt darin, dass nur die wenigsten Bot-Netz-Angriffe mediale Aufmerksamkeit bekommen [2]. Ebenso deswegen, da Privatpersonen selten Opfer solcher Angriffe werden und dadurch keine aktiven Folgen spüren [2]. Obwohl großes Interesse bei den Experten und Sicherheitsinstituten besteht, die präventiven und aktiven Schutzmöglichkeiten weiterzuentwickeln, gestaltet sich das ganze als schwierig. Bot-Netzwerke sind aufgrund ihrer Vielseitigkeit und Komplexität nur schwierig zu bekämpfen [3]. Die Arbeit befasst sich in Kapitel II mit den Grundlegenden Beweggründen, welche zur Erstellung bzw. Nutzung von Bot-Netzwerken führt. Sowie mit der Frage, wie diese gebildet werden. Kapitel III stellt die am weitesten verbreiteten Topologien für Bot-Netzwerke dar und in Kapitel IV werden verschiedene Schutzmöglichkeiten sowie Arten der Bot-Netz-Erkennung vorgestellt. Zu Schluss wird in Kapitel V noch ein Fazit gefällt.

II. Grundlegendes

Ein Bot-Netzwerk besteht grundlegend aus gekaperten Geräten, sogenannten Bots, und einem Botmaster, welcher die Befehle erteilt. Es gibt verschiedene Topologien für die Kommunikation zwischen Botmaster und seinen Bots [3]. Diese werden im weiteren Verlauf der Arbeit näher betrachtet. Die Bot-Netzwerke können für verschiedenste Möglichkeiten verwendet werden. Zum Beispiel zum Versenden von Phishing Nachrichten, Identitätsdiebstahl, Verbreitung von Malware oder illegalem Bitcoin-mining [6].

A. Gründe für die Erstellung eines Bot-Netzwerkes

Die Gründe für das Erstellen solcher Bot-Netzwerke sind sehr vielseitig. Für den Großteil der Ersteller solcher Bot-Netzwerke steht der finanzielle Aspekt an erster Stelle. Weitere Beweggründe sind Anerkennung oder weil man vor hat es selbst zu verwenden. Ein Bot-Netzwerk braucht nicht nur Geld und Zeit für den Aufbau, sondern auch für Wartungsarbeiten, damit es auf dem neuesten Stand bleibt [4]. Es gibt mehrere Möglichkeiten sein erstelltes Bot-Netz zu vermarkten. So können bereits Teile des Bot-Netzwerkes, wie zum Beispiel der Quellcode der verwendeten Malware, gewinnbringend verkauft werden. Ebenso besteht die Möglichkeit sein Bot-Netzwerk als Dienstleistung anzubieten. Für diese Zwecke gibt es Marktplätze oder Foren im Darknet [5]. Der Mieter kann dann vorgeben, für welche Aktionen die Bot-Netzwerke verwendet werden. Die Beweggründe solche Attacken auszuführen sind ebenso unterschiedlich wie bei der Erstellung der Bot-Netzwerke. Manchen geht es dabei um Anerkennung oder

wollen einem Unternehmen bzw. einer Regierung Schaden zufügen. Oftmals wird es auch verwendet, um selbst finanziellen Nutzen daraus zu ziehen [4].

B. Die Entstehung eines Bot-Netzwerkes

Bei der Entstehung von Bot-Netzwerken muss zwischen von Menschen direkt genutzten Geräten und Geräten aus dem Bereich des Internet of Things (IoT) unterschieden werden. Die Geräte aus dem IoT stellen dabei eine größere Gefahr dar, da Sie leichter gekapert werden können. Allerdings beschränkt sich das Einsatzgebiet bei Ihnen auf DDoS-Angriffe [7]. Für die höhere Anfälligkeit von Missbrauch bei IoT-Geräten gibt es mehrere Gründe:

- Sie sind meistens 24 Stunden am Tag mit dem Internet verbunden, um die Funktionsfähigkeit zu gewährleisten. PCs oder Laptops können nach der Benutzung wieder ausgeschaltet werden.
- Sie verfügen über keine oder kaum Antiviren-Software.
- Sie erhalten seltener Softwareupdates.
- Personen arbeiten nur selten direkt mit den Geräten, weshalb eine verringerte Leistung aufgrund eines Virus schwieriger erkannt wird.
- Die Richtlinien für IoT-Geräte hinsichtlich Sicherheit und Datenverkehr sind schwächer. [7]

Obwohl beiden dasselbe Problem zugrunde liegt. Nämlich eine Person, welche mit Malware versucht die Geräte für seine eigenen Zwecke zu missbrauchen, so ist das Vorgehen wie dies geschieht unterschiedlich. Bei PCs oder Laptops erfolgt dies überwiegend dadurch, den Menschen auszunutzen. Die Malware wird dabei größtenteils mittels Spam-Nachrichten verbreitet oder auf dubiosen Seiten, als etwas anderes getarnt, zum Download angeboten [1]. Einmal in dem Heimnetz angekommen, kann es sich auf andere Geräte ausbreiten [1]. Im Bereich des IoT hingegen werden Lücken in der Sicherheit der Geräte ausgenutzt. Es gibt Datenbanken, in denen die IP-Adressen von IoT-Geräten abgespeichert sind. Diese Datenbanken entstehen, weil Sie von IoT-Suchmaschinen (z.B. www.shodan.io) angelegt werden. Die Ursprüngliche Absicht hinter solchen Suchmaschinen war es, den Herstellern eine Möglichkeit zu bieten, nachzuverfolgen, wo und wozu Ihre Geräte verwendet werden. Jedoch aufgrund dessen, dass diese Datenbanken für jedermann zugänglich sind, kann man sich bereits für 1099$ pro Monat unbegrenzt viele IP-Adressen potenzieller Ziele kaufen [8] [15]. Nachdem die Angreifer die IP-Adressen haben, versuchen Sie mittels der Brute-Force-Methode Zugriff auf die Geräte zu erlangen. Dabei verwenden Sie die üblichsten Standart-Benutzernamen und Passwörter in verschiedenen Kombinationen [9]. Sollte eine Kombination dazu führen, dass Sie Zugriff erhalten, so wird die Malware auf das Gerät geladen. Der weitere Ablauf ist dann wieder bei allen Geräten gleich. Die Malware auf den infizierten Geräten versucht eine Verbindung zu dem Command&Control (C&C) Server aufzubauen, welcher von nun an Befehle an das Gerät senden kann. Die Geräte funktionieren auch nach der Infizierung weiterhin für den Benutzer normal [3] [15].

Die Malware auf den IoT-Geräten kann zwar mittels eines Neustarts aus dem System gelöscht werden. Allerdings verfügen die meisten Bot-Netzwerke über eine Datenbank mit bereits einmal verbunden Geräten und deren Benutzernamen sowie Passwörtern. Deshalb dauert es nicht lange bis es wieder zu einer Infizierung kommt [8].

III. Bot-Netzwerk Topologien

Den Betreibern von Bot-Netzwerken ist eine schnelle und zuverlässige Weitergabe von Befehlen wichtig. Allerdings muss dennoch sichergestellt sein, dass die Kommunikation nicht entdeckt wird. Dies wäre eine Gefahr für den Betreiber da dadurch auf Ihn zurückgeschlossen werden könnte [13]. Aus diesem Grund gibt es verschiedene Kommunikations-Topologien zwischen dem Betreiber und seinen Bots. Im Folgenden werden die 3 Haupt-Topologien näher betrachtet und Ihre größten Vor- sowie Nachteile dargestellt.

A. Zentralisiert

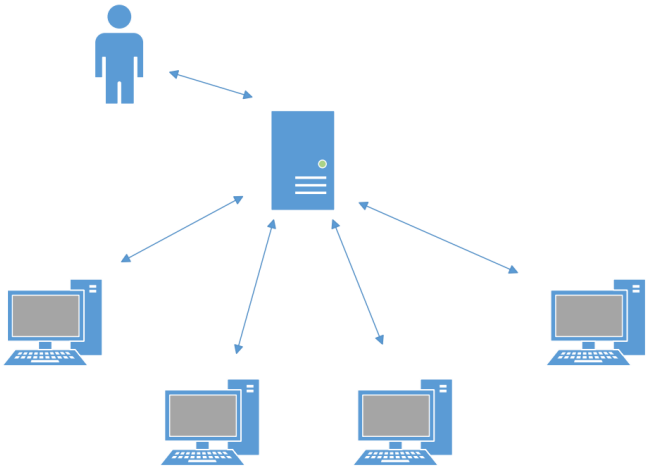

Abbildung 1. Zentralisierte Netzstruktur angelehnt an [3]

Bei der zentralisierten Netzstruktur gibt der Betreiber die Befehle über einen C&C-Server an sämtliche Bots weiter. Dies hat den Vorteil, dass geringe Latenzen auftreten und somit die Bots schnell auf neue Befehle reagieren können. Jedoch hat dies den Nachteil, dass es leicht entdeckt werden kann, da viele Geräte mit nur einem Server eine Kommunikation aufbauen. Sollte dieser Server entdeckt werden, kann dies das komplette Bot-Netzwerk zerstören. Abbildung 1 zeigt den grundlegenden Aufbau dieser Netzstruktur. [13] [6] [3]

B. Rechner-Rechner-Verbindung (Peer to Peer (P2P))

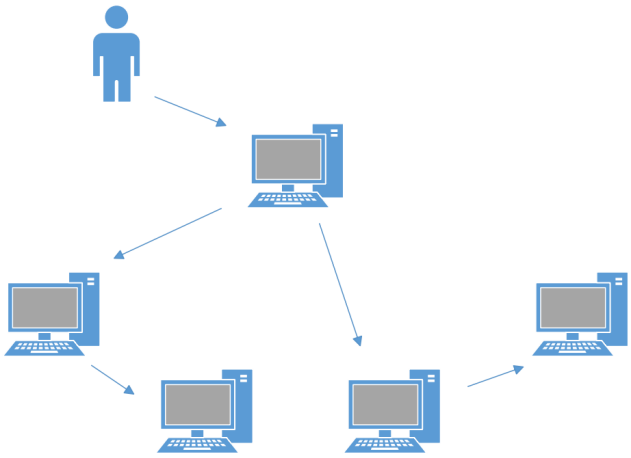

Abbildung 2. Rechner-Rechner-Verbindung angelehnt an [3]

Bei der Rechner-Rechner-Netzstruktur gibt der Betreiber die Befehle direkt an einen seiner Bots. Dieser schickt diese Befehle dann an weitere Bots, welche dasselbe ausführen. Der Vorteil dieser Arte der Befehlsweitergabe ist, dass dadurch die Erkennbarkeit verringert wird. Allerdings ist die Erstellung einer solche Kommunikationsstruktur komplexer und schwieriger als die Zentralisierte Art. Dies liegt unter anderem daran, dass die Struktur regelmäßig angepasst werden muss. Ebenso kann nicht sichergestellt werden, dass jeder Bot die Befehle erhält und die Latenzen können höher sein. Eine Darstellung der Netzstruktur ist in Abbildung 2 zu finden. [13] [6] [3]

C. Zufällige Rechner-Rechner-Verbindung

Ander als bei der Rechner-Rechner-Verbindung ist hier der Weg der Befehlsweitergabe nicht festgelegt. Nach Erhalt der Daten würde ein Bot das Internet nach einem weiteren infizierten Gerät scannen. Sobald er eines gefunden hat, leitet er die Befehle an dieses weiter. Die Erkennbarkeit sinkt dadurch erneut. Dadurch würde das Entdecken eines Bots nicht das gesamte Netzwerk offenlegen und die Erstellung der Kommunikationsstruktur wäre einfacher als bei der konventionellen Rechner-Rechner-Verbindung. Jedoch erhöht dies die Dauer der Befehlsweitergabe und es können keine Rückschlüsse gezogen werden wie viele Bots die Befehle erhalten haben. [13] [6] [3]

Die Kommunikationsmuster und Topologien zu verstehen, ist ein wichtiger Aspekt, um den Bot-Netzwerken entgegen wirken zu können. Deshalb beschäftigen sich Sicherheitsinstitute seit langem damit, um eine Möglichkeit zu finden, die Gefahr zu verringern [14].

IV. Schutzmöglichkeiten

Man differenziert dabei zwischen Schutzmöglichkeiten vor der Kaperung der eigenen Geräte und vor den Angriffen durch Bot-Netzwerke.

A. Präventiver Schutz der eigenen Geräte

Zum Schutz der eigenen Geräte ist ein umsichtiges und vorsichtiges Verhalten beim surfen im Internet unerlässlich. So lassen sich Gefahren bereits durch simple Verhaltensregeln abwenden [1]. Diese beinhalten:

- Downloads nur aus vertrauenswürdigen Quellen.
- E-Mail-Anhänge von unbekannten Absendern kritisch betrachten. Besonders wenn die E-Mails sich im Spam-Ordner befinden.
- Virenschutz aktuell halten.
- Passwörter bei IoT-Geräten wenn möglich ändern.
- IoT-Geräte von namenhaften Herstellern haben oftmals bessere Schutzvorkehrungen.
- Nicht verwendete Geräte ausschalten.

Der Schutz der eigenen Geräte hat nicht nur zum Vorteil, dass sie nicht für Angriffe auf andere Personen/Unternehmen verwendet werden können. Durch das Vorhandensein eines infiltrierten Gerätes im privaten Hausnetzwerk kann leicht eine Malware auf andere Geräte weitergegeben werden. Damit besteht die Möglichkeit zum Datendiebstahl und somit kann auch ein erheblicher Schaden für Privatpersonen entstehen.

B. Erkennung von Bot-Netzwerken

Es besteht keine effektive Schutzform vor Angriffen wie z.B. DDoS. Ebenso die Erkennung von Bot-Netzwerken gestaltet sich bei dem heutigen Stand der Technik als schwierig [3]. Es gibt drei grundlegende Ansätze, um diese zu erkennen. Sie beruhen auf der Überwachung und Analyse von Onlinedatenverkehr. Die Ansätze werden im Folgenden zusammengefasst und beschrieben.

1) Erkennung Anhand der Signatur: Dabei wird das Wissen über bereits vorhandene Bot-Netzwerk-Strukturen genutzt, um diese zu erkennen. Es wird dabei auf Datenbanken zurückgegriffen, in denen die IP-Adressen bekannter gekaperter Geräte abgespeichert sind. Der Erhalt dieser Informationen erfolgt dabei größtenteils durch das Verwenden von Computern als Köder (eng.: Honeypots). Diese Computer haben keine Aufgabe und sind lediglich mit dem Internet verbunden. Dadurch, dass Sie standardmäßig keine Verbindung mit anderen Geräten aufbauen, ist jeder Kontaktaufbau von außerhalb als potenzieller Malware-Angriff aufzufassen. Die IP-Adressen können dann der Datenbank hinzugefügt werden. Bei einem potenziellen Bot-Netz-Angriff können die IP-Adressen der potenziellen angreifenden Geräte mit der Datenbank abgeglichen werden. Der Nachteil von dieser Art der Erkennung liegt darin, dass neue Bot-Netzwerke, welche noch nicht erfasst wurden, weiterhin unentdeckt bleiben. [3] [6] [10]

2) Erkennung durch Domain Name System (DNS) Aktivitäten: Die Bots erhalten ihre Befehle, indem Sie eigenständig DNS-Abfragen absenden [11]. Sie wollen damit den C&C-Server erreichen, welcher meist von einem dynamischen DNS-Provider gehostet wird. Durch das Auswerten des DNS-Verkehrs lässt sich feststellen, ob ein Gerät Teil eines Bot-Netzwerks ist. Solche DNS-Abfragen unterscheiden sich oftmals von normalen Aktivitäten in ihrer Form und Intensität.

Man sucht deshalb nach ungewöhnlichen Kommunikationsanfragen, sogenannten Anomalien. [3] [6]

3) Erkennung anhand des Verhaltens: Diese Art der Erkennung weist Ähnlichkeit zu der DNS-basierten-Erkennung auf. Hierbei wird ebenfalls nach Anomalien im Kommunikationverhalten gesucht. Indizien für Bot-Netzwerke sind:

- Eine hohe Netzwerk Latenz [3]
- Ein hohes Volumen an Datentransfer [3]
- Datentransfer auf ungewöhnlichen Ports [3] [12]
- Ungewöhnliche Aktivitäten des Netzwerks [3]

V. Fazit

Die Gefahren, welche von Bot-Netzwerken ausgehen sind sehr vielseitig. Sie werden verwendet für Datendiebstahl (z.B. Bankdaten), Spam und Phishing (z.B. über E-Mail), DDoS, illegales Crypto-Mining und so weiter. Dennoch ist das Verständnis für diese Gefahr nicht weitreichend genug.

Die Schutzmöglichkeiten vor Angriffen sind nicht ausreichend, um die Gefahr zu bannen. Es erfordert viel Zeit und Aufwand die Hintergründe solcher Bot-Netzwerke zu verstehen. Ebenso müssen die getroffenen Vorkehrungen, immer weiterentwickelt werden, um einen Schutz bieten zu können. Ein umsichtiges Verhalten von Endnutzern ist unerlässlich, um die Gefahr zu minimieren, dass ihre Geräte missbraucht werden. Für viele Personen ist jedoch eben jenes umsichtige Verhalten zu zeitaufwendig oder sie empfinden es als lästig.

Mit dieser Arbeit wurde ein Einblick auf das Vorgehen und die Hintergründe gegeben, um die allgemeine Wahrnehmung und das Verständnis für diese Gefahr zu erhöhen.

VI. Literaturverzeichnis

[1] BSI. (2022,Mai 10). Botnetze - Auswirkungen und Schutzmaßnahmen. [Online]. https://www.bsi.bund.de/DE/Themen/Verbraucherinnen-und-Verbraucher/Cyber-Sicherheitslage/Methoden-der-Cyber-Kriminalitaet/Botnetze/botnetze.html;jsessionid=03841875FCB27D09DD5136A40398FEAC.internet482?nn=132124.

[2] H. Griffioen, C. Doerr, "Quantifying TCP SYN DDoS Resilience: A Longitudinal Study of Internet Servicesïn 2020 IFIP Networking Conference (Networking), 2020.

[3] I. Ullah, N. Khan, SSurvey on botnet: Its architecture, detection, prevention and mitigationïn 2013 10th IEEE International Conference on Networking, Sensing and Control (ICNSC), 2013.

[4] C.G.J. Putman, Abhishta, L. Nieuwenhuis, "Business Model of a Botnetïn 2018 26th Euromicro International Conference on Parallel, Distributed and Network-based Processing (PDP), 2018.

[5] P. Meland, G. Sindre, "Cyber Attacks for Saleïn 2019 International Conference on Computational Science and Computational Intelligence (CSCI), 2019.

[6] R. Malik, B. Alankar, "Botnet and Botnet Detection Techniquesïn International Journal of Computer Applications, 2019.

[7] S. Kumar, B.R. Chandavarkar, "DDoS prevention in IoTïn 2021 12th International Conference on Computing Communication and Networking Technologies (ICCCNT), 2021.

[8] N. Vlajic, D. Zhou, ÏoT as a Land of Opportunity for DDoS Hackers", 2016.

[9] D. Fraunholz, D. Krohmer, S. Anton, H. Dieter Schotten, Ïnvestigation of cyber crime conducted by abusing weak or default passwords with a medium interaction honeypotïn 2017 International Conference on Cyber Security And Protection Of Digital Services (Cyber Security), 2017.

[10] H. Zeidanloo, M. Shooshtari, P. Amoli, M. Safari, M. Zamani, Ä taxonomy of Botnet detection techniquesïn 2010 3rd International Conference on Computer Science and Information Technology, 2010.

[11] H. Choi, H. Lee, H. Lee, H. Kim, "Botnet Detection by Monitoring Group Activities in DNS Trafficïn 7th IEEE International Conference on Computer and Information Technology (CIT 2007), 2007.

[12] P. Correia, E. Rocha, A. Nogueira, P. Salvador, SStatistical Characterization of the Botnets C&C Trafficïn Procedia Technology, 2012.

[13] M. Feily, A. Shahrestani, S. Ramadass, Ä Survey of Botnet and Botnet Detectionïn 2009 Third International Conference on Emerging Security Information, Systems and Technologies, 2009.

[14] D. Dragon, G. Gu, C. Lee, W. Lee, Ä Taxonomy of Botnet Structuresïn Twenty-Third Annual Computer Security Applications Conference (ACSAC 2007), 2007.

[15] C. McDermott, F. Majdani, A. Petrovski, "Botnet Detection in the Internet of Things using Deep Learning Approachesïn 2018 International Joint Conference on Neural Networks (IJCNN), 2018.

Green Roofs as Passive Cooling Elements of Residential Buildings

Moritz Kolb
Faculty of Electrical Engineering and Information Technology
OTH Regensburg
Regensburg, Germany
moritz.kolb@st.oth-regensburg.de

***Abstract*—Worldwide, residential buildings produce one third of the greenhouse gas emissions. A large proportion of this is due to air-conditioning the building. The energy consumption for producing a thermal comfort for occupants will even rise in the future. Due to global warming, it can be assumed that cooling those buildings will gain additional importance.**
To minimize energy consumption, energy and resources must be conserved during both construction and maintenance. In recent years, much research has been done on natural building materials and constructions that have passive cooling properties. These properties allow the buildings to be cooled through physical processes and without the use of external energy. The techniques are categorized in heat protection, modulation and dissipation. In this paper, green roofs are investigated for their passive cooling properties. In addition to the passive cooling effect, green roofs have a number of benefits: They promote biodiversity, delay water masses during heavy rain events, and improve the air in cities. This makes them an important tool against climate change. A green roof is structured in several layers, consisting of plants, soil, roof protection and drainage. Cooling effects are caused by the thermal mass of the soil, shading by the plants, and evapotranspiration. Field tests in recent years have been used to investigate the effect of green roofs on the energy consumption of buildings. A reduction in consumption has been demonstrated. Whereby this strongly depends on the climate zone, and the type of green roof. Until now the green roof is not economical in the European climate zone, but this should change due to the rising energy prices and the inclusion in the economic calculation of the other properties.

***Index Terms*—Sustainable development, energy saving, energy efficient buildings, green buildings, green roof, passive cooling**

I. Introduction

Based on the Paris climate protection targets, which envisage a maximum global warming of 1.5 degrees Celsius, the German government adopted the ”Climate Protection Plan 2050” in 2016, which aims to reduce climate gas emissions by up to 95% by 2050 compared to 1990 levels [1].
The construction industry and the operation of residential buildings are accountable for a large share of greenhouse gases. Total emissions of a building add up during its life cycle, consisting of the mining process of the materials over the actual living costs like heating and cooling to the demolition and disposal of the building. Buildings are responsible for one-third of global emissions. [2]
Therefore, new approaches are now needed to reduce the lifecycle footprint of the buildings. At the same time, a good indoor climate should not be neglected, as it ensures the health of its inhabitants [3]. In the past, our ancestors already used environmentally friendly materials for the construction of their buildings [4]. They used the properties of passive cooling, which did not require energy. This approach can be reused by applying modern technological knowledge to steer towards reducing emissions.
This paper deals with the passive cooling properties of green roofs. They have additional advantages over conventional roof structures not only because of their cooling properties. These include increased biodiversity, flood protection, roof life and reduce the urban heat island effect. This effect causes higher temperatures in cities compared to the countryside and results from the absence of vegetation in urban areas. [5], [6]
Although Germany is the worldwide leader in green roofs technology, they are still associated with high construction and maintenance costs, as well as problems with leaking roofs [7]. However, considering the ongoing climate change and new knowledge gained through research, green roofs become more important.

II. Passive Cooling

Passive cooling methods have no or only very little energy consumption. In general, passive cooling is divided into three categories, which are further subdivided, as shown in Figure 1. These categories are heat protection, heat modulation and heat

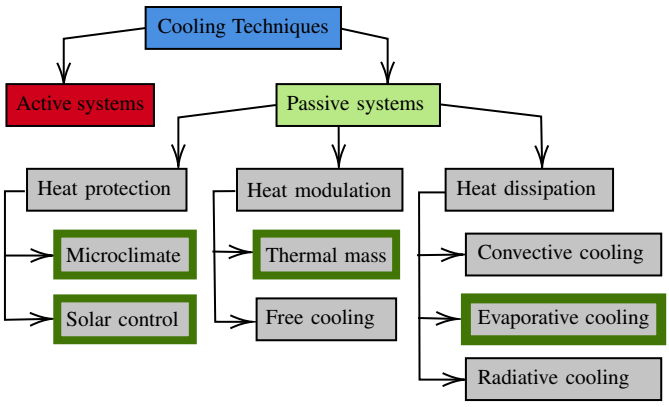

Fig. 1. Overview of passive cooling techniques [8]

dissipation. The green framing shows which properties are covered by the green roofs. The heat protection is designed to both prevent heat from entering the building and to reduce the sources of heat in the building. While the internal heat gain is influenced by the occupants and other heat sources like lighting, the external heat gain can be reduced by the microclimate and the direct shading of the building. Thermal modulation is related to the materials used in a building that absorb heat as a thermal mass or heat sink. [8]
In the case of green roofs, it is the soil that acts as the thermal mass, with the capacity depending strongly on the water content, as explained in subsection III-A. In addition, the green roof also has heat dissipation properties. Through the evaporation of water in the pores of the leaves the environment is cooled, which also affects the internal temperature. [9]

III. Green Roofs

A green roof consists of several layers. The individual layers are shown in Figure 2. These layers protect the roof structure and perform the different functions of a natural soil [10]. Above the roof structure, there is a waterproof layer, followed by a root barrier to prevent damage of the roof structure. The retention layer holds back minute particles that are in the drainage layer. Above the drainage is a filter layer that separates the soil from the lower layers. The soil is enriched with minerals and humus [10]. In order to save costs and to obtain a particularly light soil, inorganic material is mixed in by many suppliers [4]. The top two layers, the soil and the plants, can be designed differently and determine the performance of the green roof, where many parameters have an effect on the performance of the green roof [6]. As can be seen in Table I, they are roughly divided into three categories: extensive, semi-intensive and intensive green roofs. They are distinguished by the soil height and the resulting planting. While the extensive roof type is most commonly used due to its light weight, cost and maintenance, the semi-intensive and intensive green roofs function as roof terraces or even parks on underground parking garages. The success of the green roof depends significantly on the health of the plants [7]. Due to the mostly hot and dry climate on the roofs, expressly on the extensive roofs, particularly robust plants such as sedum are used, which can also survive several months without water [11]. All in all, native plants are preferred because they are already adapted to the climate [12]. The concept of building greening is also available for the facade or gable roofs. The construction and the requirements for the plants are slightly different. However, this is not discussed in this article.

A. Cooling Effects

The passive cooling effects of green roofs are mostly evapotranspiration, shading, accumulation and isolation. Evapotranspiration combines evaporation from the soil surface and transpiration, the release of water by plants into the surrounding air. [6]
The shading also plays a major role. In studies, it was found that especially in hot regions, the temperature under the canopy

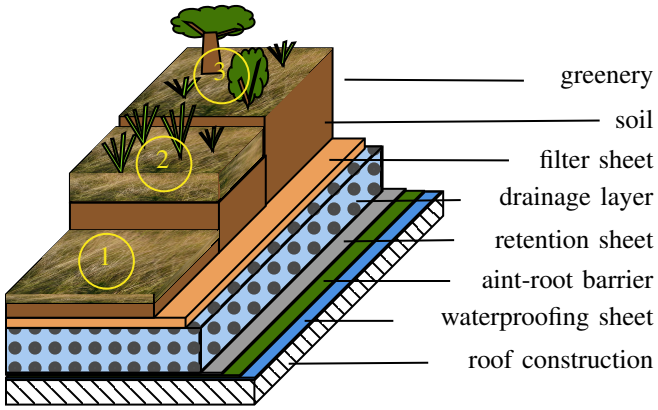

Fig. 2. Structure of a green roof with ① extensive-, ② semi-intensive- and ③ intensive green roof [10]

TABLE I
GREEN ROOF TYPES [6]

Type	**①Extensive**	**②Semi-Intensive**	**③Intensive**
Soil height	5 cm-20 cm	12 cm-25 cm	>20 cm
Plants	Moss Sedum Herbs Grasses	Herbs Grasses Shrubs	Lawn Perennials Shrubs Trees
Use	Ecological protection layer	Designed green roof	Park like garden

of intensive and extensive green roofs drops sharply. So, measured temperatures of the bare soil and soil under the canopy differ up to 30 °C. The shading effect is strongly dependent on the density of foliage or Leaf Area Index (LAI). [13], [14]
The soil of the green roof serves as thermal mass, by storing thermal energy, which stabilizes the indoor temperature throughout the year. [9] In summer, a wet soil improves the evaporation, whereas in the winter less water can increase the insulating capacity of the soil. Lazzarin at al. present a thermal model of a green roof, as shown in Figure 3. In this model the green roof is shown in six different layers: The soil is divided into three layers I - III, so the model can represent the different temperatures t and moisture contents, which affect the specific heat c, thermal conductivity λ and specific gravity r of the soil layer. The other three layers describe the drainage d, the waterproofing sheet w and the roof construction rc. Other layers have too little thermal influence, which is why they are neglected. At the top of the model is the outside air, and the bottom represents the building interior. The heat flux is shown in solid arrows for each layer, the fluid exchange in dotted arrows. The fluxes entering the system are the outer A_o and inner A_i adduction fluxes, which consist of extern convective and radiative thermal fluxes, and the solar radiation R_n. Equation (1) shows the influence of LAI and the short

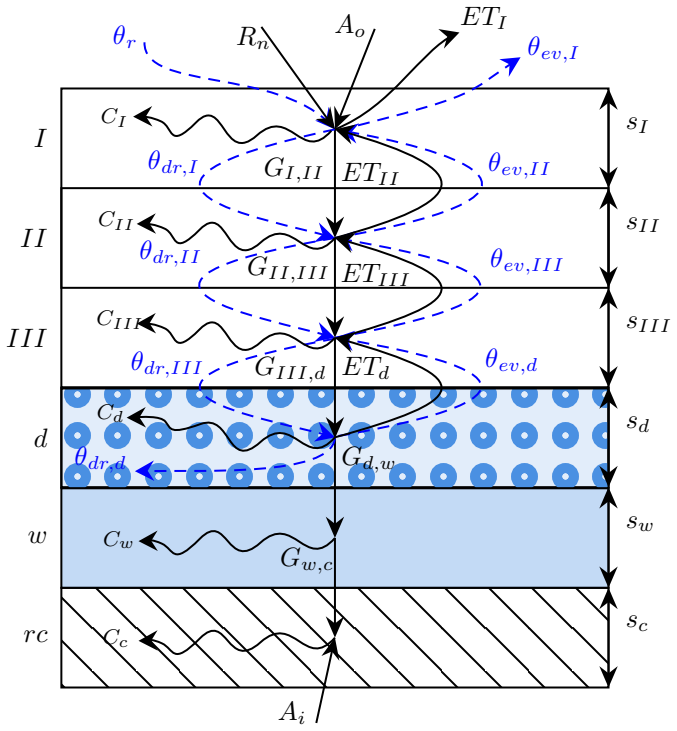

Fig. 3. The finite differences model of physical system. [10]

wave extinction coefficient ks on the solar radiation reaching the bottom of the green roof [15].

$$R_n = R^{-k_s \cdot LAI} \qquad \left[\frac{W}{m^2}\right] \tag{1}$$

The heat input flux is compensated in three ways by the layers in the green roof. The value of the thermal accumulation C_i is calculated using the soil properties and temperature rise in a given time interval. Whereby the indize i always stands for the respective layer.

$$C_i = c_i \rho_i s_i \frac{t_i - t_i'}{\Delta\tau} \qquad \left[\frac{W}{m^2}\right] \tag{2}$$

Conduction flux $G_{i,i+1}$ describes the continuation of heat through the layers from the heat source. It depends on the temperature differences of the layers and thermal conductivity of each. Whereby, the conductivity depends on the water content of the respective layer, if it is low, there is instead air in the pores of the soil, which increases the insulation.

$$G_{i,i+1} = \frac{t_i - t_{i+1}}{\frac{s_i}{2\lambda_i} + \frac{s_{i+1}}{2\lambda_{i+1}}} \qquad \left[\frac{W}{m^2}\right] \tag{3}$$

Evaporation flux ET_i allows the heat energy to be transported out of the roof through the layers. It is the water flux $\theta_{ev,i}$ that releases a water specific heat of vaporization r_i which has the unit $[J/kg]$ and not $[J/(kg \cdot K)]$, as Lazzarin states [10].

$$ET_i = \theta_{ev,i} \cdot r_i \qquad \left[\frac{W}{m^2}\right] \tag{4}$$

For layer I it results in equation (5), where the incoming and outgoing fluxes balance each other out.

$$R_n + A_o + ET_{II} = G_{I,II} + ET_I + CI \tag{5}$$

It is important to simulate the performance of the green roof with the environmental data of the building site. This is because the natural system reacts to various environmental influences [16]. Thus, in climates where plants wither, the insulation works fine, whereas in areas with milder winters, there is still a cooling effect due to evaporation, and thus the heating demand even increases slightly [17].

B. Energy Saving

Lazzerin et al. compared in a study the green roof to a traditional one. Therefore he collected data over two years on a test green roof in Vicenza (Italy). He compared two periods with two different moisture contents 10 % and nearly 100 %, the amount of thermal radiation was nearly the same, Figure 4 shows the energy flows over the period, with the energy input scaled to 100. The figure shows that the dry green roof compared to the traditional roof one allows less heat flux into the building 1.8 instead of 4.4. The wet green roof where the evapotranspiration effect is more active because of the high humidity, there is even a minimum flux out of the building. [10]

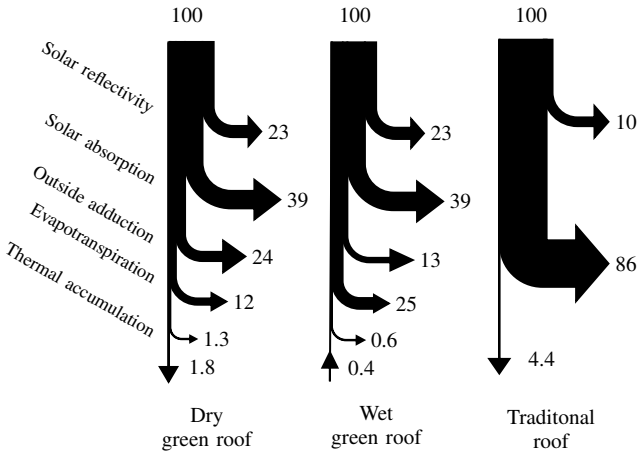

Fig. 4. Energy flow diagrams comparing a green roof with a conventional roof, where the green roof is wet on one occasion and dry on the other. [10]

Karteris simulated extensive green roofs on existing buildings in Thessaloniki (Greece) on a large scale and studied the impact on the energy consumption of each house. In his simulations, it was possible to save up to 5 % of energy costs over the year. Most of the savings were made on the top floor, directly under the green roof. He found that green roofs have the best energy-saving effects for single-story buildings. [18] Niachon pointed out that a well-insulated roof, which has

a thermal conductivity coefficient between 0.4 and 0.26 $W/m^2 \cdot K$, compensates the advantage of a green roof. Thus, through simulations, he was able to show that on summer days, buildings without insulation benefited from a green roof, whereas for insulated buildings, the effect was not apparent. He concluded from this that it is worthwhile retrofitting a green roof to old, poorly insulated buildings. [19]

IV. Economic Efficiency

A green roof has many economic advantages, only these are not always as easy to count. As the longer durability of a green roof compared to a conventional roof, because the water protection film is not exposed to environmental influences, such as frost and sun, it lasts twice as long [20].
Therefore, previous scientists have classified green roofs as unprofitable in European latitudes. The excessively high construction and maintenance costs cannot be recouped through savings in heating and cooling energy. The price of a green roof is 80-100 € per square meter, depending on the type. Additional costs for a higher effort of the roof construction are not included. In dry regions such as Seville, watering costs can add up to over €1000 per year. Which cancel out the savings that came from the lower energy consumption. [21]
However, as energy prices rise and insect diversity becomes more important, green roofs are now mandatory in some areas [22]. Also, there are more ways to improve the economic value of a green roof in the future. Scientists discover how green roofs can be combined with other solutions.
For example, in a case study, Hui found that the performance of photovoltaic systems increases when they are mounted on a green roof instead of a conventional roof. In his case, 8 % more electricity was produced. That's because the green roof cools the panels, keeping them in a more efficient thermal location. But the building's cooling also improves because the green roof is shaded by the PV panels. [23]
If planted with vegetables green roof could help to feed cities with local food and save transportation costs [24].

V. Conclusions

In this paper, the green roof was presented as a passive cooling method for residential buildings. First, the design with the different types of the layers was described. Then, a thermal system was presented which is used to mathematically describe the green roof physics of passive cooling. With this, it is possible, to tune a green roof for the particular environment in which the roof is to be installed. With the help of other studies, the effectiveness of the green roof was investigated under different conditions. It was found out, that if the focus is only on cooling, the green roof has no advantage compared to a modern insulated roof. However, in old buildings with poorly insulated roofs, it is a good retrofit solution. In the economic projections, only heating cost savings are outweighed by additional installation costs for the green roofs compared to traditional roofs. However, considering the green roof, with all its properties, from the preservation of biodiversity to better air in the cities to flood protection, the green roof gains in value. In addition, the double use of green roofs with solar panels or urban farming will be added in the future, which will increase the value of this construction technology in the coming years.

References

[1] BMUB, "Klimaschutzplan 2050. Klimaschutzpolitische Grundsätze und Ziele der Bundesregierung," *Klimaschutzplan 2050*, pp. 1–96, 2016.
[2] Intergovernmental Panel on Climate Change, "Residential and commercial buildings," *Climate Change 2007*, no. January, pp. 387–446, 2012.
[3] P. Wolkoff and S. K. Kjærgaard, "The dichotomy of relative humidity on indoor air quality," *Environment International*, vol. 33, no. 6, pp. 850–857, 2007.
[4] K. Vijayaraghavan, "Green roofs: A critical review on the role of components, benefits, limitations and trends," *Renewable and Sustainable Energy Reviews*, vol. 57, pp. 740–752, 2016.
[5] D. Masseroni and A. Cislaghi, "Green roof benefits for reducing flood risk at the catchment scale," *Environmental Earth Sciences*, vol. 75, no. 7, 2016.
[6] B. Raji, M. J. Tenpierik, and A. Van Den Dobbelsteen, "The impact of greening systems on building energy performance: A literature review," *Renewable and Sustainable Energy Reviews*, vol. 45, pp. 610–623, 2015.
[7] M. Shafique, R. Kim, and M. Rafiq, "Green roof benefits, opportunities and challenges – A review," *Renewable and Sustainable Energy Reviews*, vol. 90, no. March, pp. 757–773, 2018.
[8] D. K. Bhamare, M. K. Rathod, and J. Banerjee, "Passive cooling techniques for building and their applicability in different climatic zones—The state of art," *Energy and Buildings*, vol. 198, pp. 467–490, 2019.
[9] H. F. Castleton, V. Stovin, S. B. Beck, and J. B. Davison, "Green roofs; Building energy savings and the potential for retrofit," *Energy and Buildings*, vol. 42, no. 10, pp. 1582–1591, 2010.
[10] R. M. Lazzarin, F. Castellotti, and F. Busato, "Experimental measurements and numerical modelling of a green roof," *Energy and Buildings*, vol. 37, no. 12, pp. 1260–1267, 2005.
[11] K. L. Getter and D. B. Rowe, "Selecting Plants for Extensive Green Roofs in the United States," *East*, no. July, 2008.
[12] O. Schweitzer and E. Erell, "Evaluation of the energy performance and irrigation requirements of extensive green roofs in a water-scarce Mediterranean climate," *Energy and Buildings*, vol. 68, no. PARTA, pp. 25–32, 2014.
[13] N. H. Wong, Y. Chen, C. L. Ong, and A. Sia, "Investigation of thermal benefits of rooftop garden in the tropical environment," *Building and Environment*, vol. 38, no. 2, pp. 261–270, 2003.
[14] D. Morau, T. Libelle, and F. Garde, "Performance evaluation of green roof for thermal protection of buildings in reunion Island," *Energy Procedia*, vol. 14, no. 262, pp. 1008–1016, 2012.
[15] E. Palomo Del Barrio, "Analysis of the green roofs cooling potential in buildings," *Energy and Buildings*, vol. 27, no. 2, pp. 179–193, 1998.
[16] T. G. Theodosiou, "Summer period analysis of the performance of a planted roof as a passive cooling technique," *Energy and Buildings*, vol. 35, no. 9, pp. 909–917, 2003.
[17] I. Jaffal, S. E. Ouldboukhitine, and R. Belarbi, "A comprehensive study of the impact of green roofs on building energy performance," *Renewable Energy*, vol. 43, pp. 157–164, 2012.
[18] M. Karteris, I. Theodoridou, G. Mallinis, E. Tsiros, and A. Karteris, "Towards a green sustainable strategy for Mediterranean cities: Assessing the benefits of large-scale green roofs implementation in Thessaloniki, Northern Greece, using environmental modelling, GIS and very high spatial resolution remote sensing data," *Renewable and Sustainable Energy Reviews*, vol. 58, pp. 510–525, 2016.
[19] A. Niachou, K. Papakonstantinou, M. Santamouris, A. Tsangrassoulis, and G. Mihalakakou, "Analysis of the green roof thermal properties and investigation of its energy performance," *Energy and Buildings*, vol. 33, no. 7, pp. 719–729, 2001.
[20] S. W. Tsang and C. Y. Jim, "Game-theory approach for resident coalitions to allocate green-roof benefits," *Environment and Planning A*, vol. 43, no. 2, pp. 363–377, 2011.
[21] F. Ascione, N. Bianco, F. de' Rossi, G. Turni, and G. P. Vanoli, "Green roofs in European climates. Are effective solutions for the energy savings in air-conditioning?" *Applied Energy*, vol. 104, pp. 845–859, 2013.

[22] U. Berardi, A. H. GhaffarianHoseini, and A. GhaffarianHoseini, "State-of-the-art analysis of the environmental benefits of green roofs," *Applied Energy*, vol. 115, pp. 411–428, 2014.
[23] S. C. M. Hui and S. C. Chan, "Integration of green roof and solar photovoltaic systems," *Joint Symposium 2011: Integrated Building Design in the New Era of Sustainability*, vol. 2011, no. November, pp. 1–12, 2011.
[24] S. C. M. Hui, "Green roof urban farming for buildings in high-density urban cities," *World Green Roof Conference*, no. March, pp. 1–9, 2011.

Challenges in Increasing Energy Efficiency in Indoor Food Agriculture

Franz Hohenadler
Electrical Engineering and Information Technology
Technical University of Applied Sciences
Regensburg, Germany
franz.hohenadler@st.oth-regensburg.de

***Abstract*—With the increasing threat of crop failure due to climate change and the rising global demand for food, Controlled Environment Agriculture (CEA) is receiving more and more attention. CEA exposes plants to ideal growing conditions which are constantly regulated. The conditions in this facilities are optimized to guarantee a maximum yield per unit area. The challenges of this cultivation method are reflected in the energy and economic comparison with conventional agriculture. The largest parts of the additional effort are due to the construction of the artificial environment and the additional energy input during the growing season.**
The purpose of this paper is to elaborate how yields can be ensured or increased with reduced energy input. The thematic approaches for possible energy savings are described in the mechanical system, lighting and the interaction with the environment. Light-Emitting Diodes (LEDs) as a new technology for area-wide lighting contains extensive possibilities for increasing energy efficiency. Compared to other light sources, e.g. High-Intensity Discharge (HID) lamps, LED lamps have a lower thermal radiation and supports spectral specialization. Adjusted lighting conditions enhance the ratio of dry matter production to electrical energy consumption by 97%. The second part deals with the improvement of the mechanical systems. Depending on the cultivation site, outdoor temperature and the type of crop Heating, Ventilation and Air Conditioning (HVAC) contributed between 27.9% and 80% of the annual energy consumption. Various approaches to energy conservation are discussed that can reduce the energy consumption in the area by 50%. The last part discusses the benefits to other industries and the shortened transportation distances.

***Index Terms*—Controlled environment agriculture, Food industry, Energy use efficiency, Spectrum specific light, Lighting, Building design, Agricultural engineering**

I. Introduction

The world population is projected to grow upto 9.7 billion by 2050 [1]. Compared to the year 2000, the land area covered by cities will quadruple by 2050 [2]. This will confront food production with the challenges of feeding the world's population and making effective use of available land. Part of the solution can provide indoor or vertical farming that can be integrated into urban areas [3]. Including indirect greenhouse gas emissions associated with land cover change, traditional agriculture contribute between 19% and 29% of global emissions [4]. Thus it is essential to maximize electrical energy use efficiency (EUE), defined as the ratio of dry matter production to electrical energy consumption (EEC) [3]. The definition of the performance indicator EUE allows for two possible ways of improvement. EUE can be increased by using components with higher efficiency or by increasing the yield while maintaining the same energy input. EUE depends on local weather conditions, indoor enviroment, the mechanical systems, crop growth and the desired result [5],[6].

II. Growth environment

Controlled Environment Agriculture (CEA) is an interaction of different agricultural techniques and advanced technologies. Required are the ideal growing conditions of the plants, which differ in the specific growth stages [7]. Rather to human comfort, the needs of plants are associated with different parameter values what is shown below.

A. Temperature

Temperature is the basis for year-round cultivation. In cooler seasons heating is necessary and in summer air conditioning increase yields. In general, the optimal temperature is between 20 °C and 30 °C. An exception to this are mushrooms, for example, which grow best between 13 °C and 24 °C. [8]

B. Lighting

Light is one of the most important components in the growth behavior, which can be seen in the changeable growth direction [7]. Nevertheless, the intensity of light is not only responsible for the highest growth rate. A study achieved by illuminating with a light spectrum containing 70% red light and 30% blue light results in four times higher yield of mentha essential oil, compared to the solar spectrum [9]. The optimal lighting density and spectral specification depends strongly on the plant species [8].

C. Humidity and transpiration

The moisture should be between 60% and 85% depending on the plant species. Strong deviations due to the constant evaporation from the moist soil and from the plants themselves can ruin the plants with mold and mildew [10].

D. Air quality

The carbon dioxide (CO_2) concentration in the natural air is generally around 412 ppm [11]. Plants need CO_2 for photosynthesis and growth. Increasing the proportion of gas in the environment can result in better growth. Plants such as lettuce and herbs grow best at three to four times the concentration [8].

III. Lighting

Lighting conditions in CEA have the greatest influence on growth behaviour [7]. In addition, except for CEA in extreme climatic conditions, lighting is the largest part of the EEC [12]. This leads to the fact that lighting efficiency is an essential part of energy efficiency and therefore profitability [5]. The following are several points to increase energy efficiency in lighting.

A. Illuminant selection

Before the light-emitting diode (LED) reached the technological development to be widely used, high-intensity discharge (HID) lamps were used in indoor agriculture [13]. Now it is only the cost of initial installation that is tempting farmers to continue using HID lamps [5], but LEDs can score with their low power requirements, cool emission surface and wavelength specificity [7].
Compared to advanced HID lamps, which have an efficiency of 35% - 55% [14], commercially available growth LEDs achieve comparable energy efficiency [15], but lower operating costs due to their long lifetime. [5]. Furthermore, the cool emission surface allows the light source to be placed closer to the plants to reduce photon losses [3],[5]. The actual usable light intensity is reduced by scattering losses to a quarter at double the distance [7].
LEDs can be designed for specific light colours or spectral distribution. LED lighting can be used to optimally meet the spectral light requirements of the plant species, which will be discussed in more detail in the following section.

B. Specialized spectrum of light

Spectrum specific light is used to fulfill two different approaches. First, for energy efficiency, it is useless to invest electrical energy in light colours that have vanishing or negative effects on plant growth. Studies show that green light has a small effect on crop yield. Wavelengths far above red light will be converted to heat. And wavelengths below 300 nm are increasingly toxic to plants. Figure 1 shows the two areas with the greatest impact on plant growth. Outside these two areas, the positive effects become smaller. [7]
The second approach is to use knowledge about the effects of monochromatic light on the plant. Red light makes artichoke seedlings grow twice as tall and generally accelerates the flowering process [16]. Blue light, on the other hand, enhances photosynthesis, increases chlorophyll concentration and is important for plant health. The best results are obtained in a red-blue combination, where the blue content is up to 30% [17]. In this way, the yield can be increased with the same energy input. The spectrum of a special LED for indoor cultivation is shown in Figure 1. A particularly high intensity can be observed in both relevant colour components. The distribution of the spectrum depends on the technology used. In this example, a blue LED with colour-specific diffuser is used, which is responsible for the red component. HID lamps lack the blue component in the light spectrum and are therefore often combined with blue-coloured LEDs. [3]

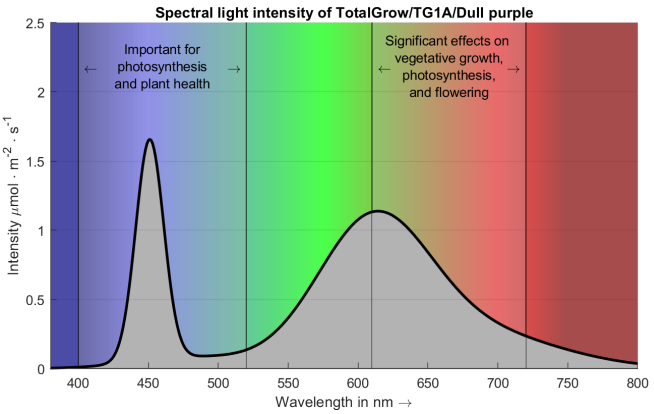

Fig. 1. Important wavelength for growth [7] (grayed out areas) and the spectral intensity of TG1A TotalGrow™ Broad Grow Spectrum Light [3] (solid line).

C. Time controlled lighting schedules

The demand for electrical energy for lighting of a CEA correlates with the light intensity and the summed duty cycle. Electricity consumption in the light phases, which can be 16 h of a 24 h day, account for a significant part of the total. A study on basil tried shorter and more dark phases, interspersed with 10 min of light, which amount to less than 16 h of illumination in one day. The results were positive in two ways. EEC decreases by 16% due to the shorter lighting period, while biomass output increases by 47% compared to continuous light. [18]

IV. Mechanical systems

The individual systems for Heating, Ventilation and Air Conditioning (HVAC) controls indoor conditions such as temperature, CO_2 concentration, humidity and air circulation [6]. In addition to lighting as one of the largest energy consumer, HVAC requires more than 27.9% of annual energy use in indoor agriculture but in northern latitudes and extreme climates, HVAC energy use can even reach 80% of total [19].

A. Heating

The amount of heat energy needed depends on the architecture of the building, outside temperatures and internal heat sources. Lighting consumes a lot of energy and especially HID lamps generate a lot of unwanted heat energy. Due to the radiant heat of the lamps, reversing the day-night rhythm of the lighting reduces the heating and cooling requirements of the HVAC systems. [8]
The preferred heat sources are hot water boilers to store excess heat energy of the day [20] or heat pumps, which are also very efficient for cooling applications [12].

B. Ventilation

Continuous horizontal airflow at a rate of 0.3 to 0.5 m/s is required for diffusion of CO_2 and water transpiration [8]. Ventilation rates of 0.01-0.02 air changes per hour are recommended for air quality and plant health [8]. Exchanging air with outside is a major challenge in indoor agriculture,

because it would be a very effective way for cooling or dehumidifying. However, this is not possible in environments with elevated CO_2 level in the atmosphere that would be diluted by economizer cooling [10]. Such designs requires an other cooling system such as heat exchangers [8].

C. Air conditioning

Air conditioning is the control of air temperature and humidity. A possible method is economical cooling with direct air exchange with the outside air. This can be done by natural convection or forced by circulating fans [20]. Variable speed motors as an alternative to on/off control increase the life time and reduce the energy consumption of the ventilation system [21]. If the outside air temperature is not suitable or there are special requirements for air quality, a closed system is required. Absorption and adsorption cooling eliminate the need for a compressor and makes the operation of the cooling more energy efficient [8].

D. Controlling and energy modelling

The set of influences on the system, the variable requirements of the plants, and the various components to control the growing environment makes CEA to a complex system. This system is considered as a multiple-input multiple-output (MIMO) system that includes many thermodynamic components [21]. For control, Model Predictive Control (MPC) algorithms are the most promising and researched control systems, followed by fuzzy logic controllers and adaptive controllers [6]. Current disturbance variables such as solar radiation and outside air temperature must be detected and automatically taken into account in the system control [21],[6]. These advanced algorithms can also handle the link between different parameters, such as temperature and transpiration, and put the focus on energy efficiency [6].

V. Environmental consideration

Although indoor agriculture regulates the environmental conditions in the interior, the energy to be used for this purpose is strongly dependent on the external conditions. Particularly large differences in temperature and humidity are directly coupled with high energy consumption [22]. Therefore, from an energy perspective, it makes sense to include environmental conditions in addition to growing demands when designing the facility.

A. Architecture design

Even though the plants are illuminated by artificial light in an enclosed area, the sunlight outside of the facility is significant. In addition to generating electrical energy through photovoltaics (PV), solar radiation can have a positive effect on the indoor climate or cause additional energy consumption in the case of overheating. [21]
With the knowledge of the temperature curves and irradiation directions and light intensity outside of the facility, the building architecture can be planned in an optimized way. On the side facing away from the sun, a solid wall with a high insulation value is recommended to avoid heat losses. On the side facing the sun, window fronts reduce the required artificial lighting intensity; at the same time, shading reduces heat accumulation and therefore the required cooling capacity by 35%. The optimal parameters for the size of the window depend on the climatic conditions at the location. [12]
Furthermore, the location is also relevant in the energy use for food transport and further processing. Depending on the region 26-64% of the population can't fulfill their demands for specific crops within a 1000 km radius [23]. Locating CEA near consumers or processing will significantly reduce CO_2 emissions associated with transporting food along the food supply chain [24].

B. Relaxing boundaries

In Section II, the optimum conditions for various plants were worked out. Around these values, there is a range in which the plants are neither damaged nor significantly impaired in their growth. Therefore, it is energetically recommended to allow the control a wider range of these parameters. In warmer latitudes, for example, the temperature could be lowered further during the night phase, reducing the cooling capacity during the day. Such an approach significantly reduces the operating time of the HVAC systems and thus also has a direct impact on energy consumption. [22]

C. Combined systems

In addition to the savings in internal facility operations, CEA can have a positive impact on the environment by combining it with other industries. The combination with different branches leads to energetic advantages, e.g. to reduce CO_2 emissions in the chemical industry, to clean industrial water, to process biological waste or to improve the air quality in office buildings. [19],[25]

VI. Conclusion

This literature review addresses the best growing conditions for plants and the various techniques to create them in enclosed spaces with high energy efficiency. Energy consumption of lighting and HVAC systems can be reduced by 50%. The second topic discussed is growth rate. Individual lighting and air quality parameters on plants improve yield by up to 97%. In these results, it must be noted that the values were achieved only in the consideration of individual parameters. It cannot be assumed that in combination all savings will be shown to the full extent.
In future research CEA should be viewed as a single entity with significant internal interactions. Individual savings are strongly related to the exact combination of location, plant species, environment and technologies used.
In addition to the savings in the operation of the plants, CEA also has numerous positive effects on the environment. The combination with different industries should be considered when planning new plants and should also be taken into account in the future design of urban areas. Establishing CEA close to consumers is critical for safe and efficient food production.

References

[1] D. Gu, K. Andreev, and M. E. Dupre, "Major trends in population growth around the world," *China CDC weekly*, vol. 3, no. 28, p. 604, 2021.

[2] S. Angel, J. Parent, D. L. Civco, A. Blei, and D. Potere, "The dimensions of global urban expansion: Estimates and projections for all countries, 2000–2050," *Progress in Planning*, vol. 75, no. 2, pp. 53–107, 2011.

[3] Y. Kong, A. Nemali, C. Mitchell, and K. Nemali, "Spectral quality of light can affect energy consumption and energy-use efficiency of electrical lighting in indoor lettuce farming," *HortScience*, vol. 54, no. 5, pp. 865–872, 2019.

[4] X. Liu, S. Zhang, and J. Bae, "The impact of renewable energy and agriculture on carbon dioxide emissions: Investigating the environmental kuznets curve in four selected asean countries," *Journal of cleaner production*, vol. 164, pp. 1239–1247, 2017.

[5] G. J. MacKenzie, "Indoor agricultural technologies: An introduction to the future of sustainable farming," *The CROW*, p. 71, 2017.

[6] E. Iddio, L. Wang, Y. Thomas, G. McMorrow, and A. Denzer, "Energy efficient operation and modeling for greenhouses: A literature review," *Renewable and Sustainable Energy Reviews*, vol. 117, p. 109 480, 2020.

[7] M. Rehman, S. Ullah, Y. Bao, B. Wang, D. Peng, and L. Liu, "Light-emitting diodes: Whether an efficient source of light for indoor plants?" *Environmental Science and Pollution Research*, vol. 24, no. 32, pp. 24 743–24 752, 2017.

[8] N. Engler and M. Krarti, "Review of energy efficiency in controlled environment agriculture," *Renewable and Sustainable Energy Reviews*, vol. 141, p. 110 786, 2021.

[9] M. R. Sabzalian, P. Heydarizadeh, M. Zahedi, *et al.*, "High performance of vegetables, flowers, and medicinal plants in a red-blue led incubator for indoor plant production," *Agronomy for sustainable development*, vol. 34, no. 4, pp. 879–886, 2014.

[10] D. L. Jonlin and D. J. Lewellen, "A low-energy high managing energy use for commercial indoor cannabis cultivation," *Energy Engineering*, vol. 114, no. 4, pp. 69–79, 2017.

[11] A. Boretti, "Covid 19 impact on atmospheric co2 concentration," *International Journal of Global Warming*, vol. 21, no. 3, pp. 317–323, 2020.

[12] C. Hachem-Vermette and A. MacGregor, "Energy optimized envelope for cold climate indoor agricultural growing center," *Buildings*, vol. 7, no. 3, p. 59, 2017.

[13] H. Dou, G. Niu, M. Gu, and J. G. Masabni, "Effects of light quality on growth and phytonutrient accumulation of herbs under controlled environments," *Horticulturae*, vol. 3, no. 2, p. 36, 2017.

[14] K. Stockwald, H. Kaestle, and H. Ernst, "Highly efficient metal halide hid systems with acoustically stabilized convection," *IEEE Transactions on Industry Applications*, vol. 50, no. 1, pp. 94–103, 2013.

[15] "Totalgrow high intensity top-light." (2021), [Online]. Available: https://www.totalgrowlight.com/products/high_intensity_toplight.html (visited on 06/17/2022).

[16] R. C. Rabara, G. Behrman, T. Timbol, and P. J. Rushton, "Effect of spectral quality of monochromatic led lights on the growth of artichoke seedlings," *Frontiers in plant science*, p. 190, 2017.

[17] R. Hernández and C. Kubota, "Physiological responses of cucumber seedlings under different blue and red photon flux ratios using leds," *Environmental and experimental botany*, vol. 121, pp. 66–74, 2016.

[18] D. D. Avgoustaki, T. Bartzanas, and G. Xydis, "Minimising the energy footprint of indoor food production while maintaining a high growth rate: Introducing disruptive cultivation protocols," *Food Control*, vol. 130, p. 108 290, 2021.

[19] C. Zeidler, D. Schubert, and V. Vrakking, "Vertical farm 2.0: Designing an economically feasible vertical farm-a combined european endeavor for sustainable urban agriculture," Ph.D. dissertation, Association for Vertical Farming, 2017.

[20] A. M. Syed and C. Hachem, "Review of construction; geometry; heating, ventilation, and air-conditioning; and indoor climate requirements of agricultural greenhouses," *Journal of Biosystems Engineering*, vol. 44, no. 1, pp. 18–27, 2019.

[21] A. Maher, E. Kamel, F. Enrico, I. Atif, and M. Abdelkader, "An intelligent system for the climate control and energy savings in agricultural greenhouses," *Energy Efficiency*, vol. 9, no. 6, pp. 1241–1255, 2016.

[22] P. Van Beveren, J. Bontsema, G. Van Straten, and E. Van Henten, "Minimal heating and cooling in a modern rose greenhouse," *Applied energy*, vol. 137, pp. 97–109, 2015.

[23] P. Kinnunen, J. H. Guillaume, M. Taka, *et al.*, "Local food crop production can fulfil demand for less than one-third of the population," *Nature Food*, vol. 1, no. 4, pp. 229–237, 2020.

[24] D. D. Avgoustaki and G. Xydis, "How energy innovation in indoor vertical farming can improve food security, sustainability, and food safety?" In *Advances in Food Security and Sustainability*, vol. 5, Elsevier, 2020, pp. 1–51.

[25] Y. Shao, J. Li, Z. Zhou, *et al.*, "The effects of vertical farming on indoor carbon dioxide concentration and fresh air energy consumption in office buildings," *Building and Environment*, vol. 195, p. 107 766, 2021.

GPT-3 and Friends: Transformers in Natural Language Processing

Luis Reber
Faculty of Electrical Engineering and Information Technology
Ostbayerische Technische Hochschule Regensburg
Regensburg, Germany
luis.reber@st.oth-regensburg.de

***Abstract*—In 2020, the British newspaper "The Guardian" published an editorial written in its entirety by an artificial intelligence called *"Generative Pre-trained Transformer 3"*. GPT-3 is one of several *"Natural Language Processors" (NLPs)*, allowing a computer to process the context and meaning of natural language. This review paper takes a closer look at NLPs based on a new type of sequence transduction model called the *transformer* and gives an overview over the current strengths and shortcomings. While the basic principles and functionality behind the transformer model and natural language processing will be explained, deeper technical and mathematical background about the model architecture and training process will not be given. Focus is instead put on explaining the defining characteristics of the transformer and showcasing the capabilities of transformer-based NLPs. While impressive results can be achieved in several NLP tasks, problems with large-scale NLPs trained on large unedited text corpora are also pointed out. Besides technical limitations, special attention is given to broader issues such as the energy usage of NLPs and biases in representation contained in generated text. This is especially important to consider when looking into future developments of transformer-based NLPs and their broader impact on society, as discussed in the conclusion of this review paper.**

***Index Terms*—Natural language processing, Neural networks, Deep learning, Predictive models, Transformer**

I. Introduction

With the release of the Natural Language Processor (NLP) *"Generative Pre-trained Transformer 3" (GPT-3)* in May of 2020, the company OpenAI made waves in both mainstream and specialist media [1]. At the time of it's release, *GPT-3* [2] was the biggest language model by far, trained with 175 billion machine learning parameters compared to the previous 17 billion parameters [3], showing high results in several NLP tasks such as sentence completion, closed book questions answering and language translation. Following it's release to the public via an API[1], people showcased many applications in which GPT-3 could be used to create a website layout, input text into a spreadsheet or generate LaTeX equations [4].
While these are only small scale examples to show possible future applications, NLPs have already made their way into publicly available software. Google's neural network-based language model *Bidirectional Encoder Representations from Transformers (BERT)* is used in their search engine to process and better understand search queries [5] to produce better fitting results for the end-user.

These achievements have been made possible by a new type of sequence transduction model introduced in 2017, the *transformer* [6]. Enabling NLPs to be efficiently trained on large corpora of unlabeled text and solving problems of previous approaches to language modeling, the transformer provided a breakthrough in the field of natural language processing [7]. As the transformer-based approach to natural language processing has become the common choice [8] and NLPs are becoming more powerful, it is important to take a deeper look at the current state of development.

To better evaluate NLPs and understand the defining characteristics brought on by the transformer, this review paper summarizes current and future developments in natural language processing. While basic principles and functionality behind the transformer and NLPs such as *GPT-3, Turing-NLG and BERT* will be explained in Chapter II, deeper technical and mathematical background about the model architecture and training process will not be provided.
Instead, this paper aims to show the abilities and problems of NLPs, to clarify why the recent trends have attracted so much attention. While the topics discussed in Chapter III show promising results, there are deeper issues present with all existing transformer-based NLPs as a result of large-scale unsupervised training. These issues, specifically the required energy usage during training and biases in representation are laid out in Chapter IV.
Overall, this paper aims to give an overview of the current state of NLPs, as well as give an outlook into future research and development.

II. Background

Giving a computer the ability to understand natural human language has long been a topic of discussion and research. Understanding language builds the basis for the *Turing Test* [9] introduced in 1950, requiring a machine to communicate with a person. To give a computer the ability to understand and generate natural language, research into the field of *natural language processing* has led to the creation of *NLPs* [8].
For better understanding of this paper, this chapter provides a brief overview of the principles behind *natural language processing* and the *transformer*.

[1] Application Programming Interface

The basic approach behind modern *NLPs* relies on the use of neural networks [8]. Until the introduction of the *transformer*, common Recurrent Neural Networks (RNN) such as Long Short-Term Memory (LSTM) and Gated Neural Networks (GNN) "[..] have been firmly established as State-Of-the-Art (SOTA) approaches in sequence modeling and transduction problems such as language modeling and machine translation" [6].

A RNN works by processing input, like a series of $x = (x_1, x_2, \ldots, x_T)$, sequentially and saving the output in a hidden state h_t as a function of the previous hidden state h_{t-1} and the input x_t. This strictly sequential approach hinders parallelization during training, thus it is rather slow and inefficient, especially at longer sequence lengths [6].

The transformer solves this problem by only relying on a *self-attention* mechanism, which allows for dependencies between an element, in the case of NLPs a word, to any other element in the sequence, not just the one directly preceding it [10]. In other words, the transformer compares a word in a sentence to every other word in the sentence.

After the introduction of the *transformer*, several transformer-based NLPs such as *BERT, GPT-3, XLNet, Turing-NLG or Megatron-Turing NLG* have emerged, achieving SOTA performance in natural language processing tasks [11]. While previous approaches such as LSTM did not see significant improvement when increasing the number of parameters [12], transformer-based models have shown to produce better results with an increase in model size. As seen in Figure 1, doubling the parameters of GPT-3 raises the Bilingual Evaluation Understudy (BLEU) [13] score by 2-5 points. As a result, the

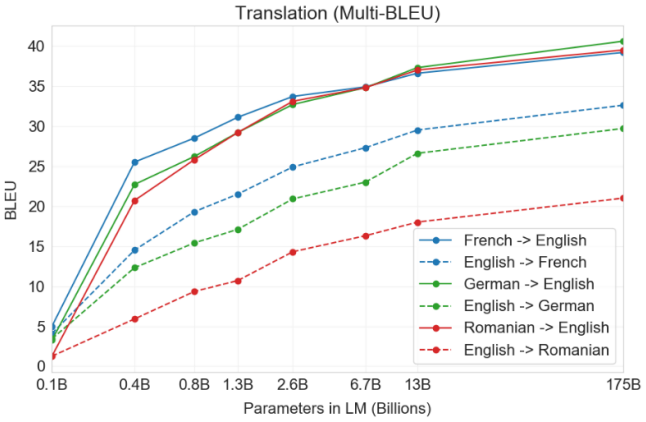

Fig. 1. Translation performance by GPT-3 in 6 language pairs [2]. The higher the score the better.

size of transformer-based NLPs has seen a steady increase the number of parameters used for training, as seen in Figure 2. While the original BERT [14] model, published in 2018 relied on 340 million parameters, the 2021 Switch-C [15] is trained on 1571 billion.

This has also led to an increase in the amount of data NLPs can be trained on. This data, collected in datasets such

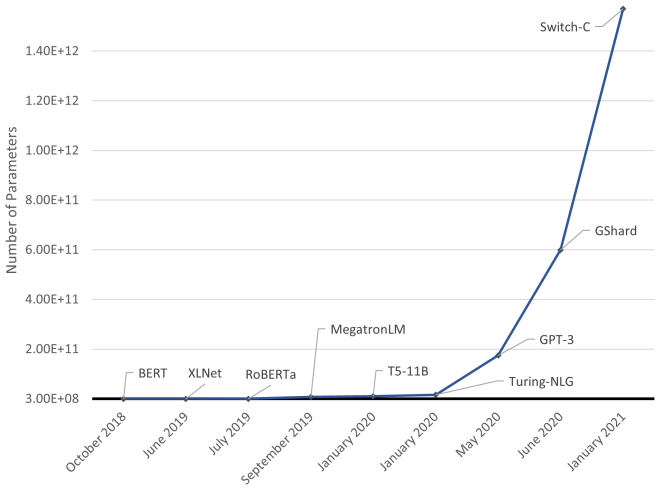

Fig. 2. Overview of recent language model base on data provided by Bender et al. [16]

as CommonCrawl[2] or OpenWebTextCorpus[3], is given to the language model during a process called *pre-training*. During *pre-training*, the NLP is trained on several tasks to learn natural language representations and calculate its parameters. This knowledge is used to infer skills for other tasks.

While increasing the number of parameters and amount of training data for unsupervised pre-training introduces and exacerbates several problems, discussed in Chapter IV, it has also led to a breakthrough in natural language processing. The capabilities of modern NLPs are discussed in the following chapter.

III. Capabilities in Natural Language Understanding and Generation

While "The Guardian" claimed that an AI wrote an entire article, the case is not as clear as it seems. An annotation beneath the article clarifies that the text is a combination of the best snippets taken from eight different results. As it is unknown how heavily the article was edited [17], some called the article misleading [18], as "[t]o anyone who has ever looked at a pile of rejected GPT-3 outputs, [it] sounds more just a little disingenuous [..]" [1].

To get a more accurate depiction of the capabilities achieved in natural language processing, it is useful to look at the strengths and shortcomings of transformer-based NLPs.

In the paper introducing the *transformer*, Vaswani et al. found a transformer model trained to perform translations between English↔German and English↔French could outscore previous SOTA language models by 2 BLEU and 0.5 BLEU respectively [6]. As understanding natural language requires analyzing the syntactic sentence structure, the model was also tested on English constituency parsing. It was found to outperform all but one previous language model, indicating

[2]https://commoncrawl.org/the-data/

[3]https://skylion007.github.io/OpenWebTextCorpus/

that transformer-based models can generalize well to other NLP tasks. Based on these findings, several NLPs have been created which achieve even greater results in many areas, although the ability of a NLP in a specific task depends on the method used to train the model and how the task was performed.
The common way of fine-tuning a language-model to a specific task results in the highest performance. Fine-tuning is a process that "[..] involves updating the weights of a pre-trained model by training on a supervised dataset specific to the desired task" [2]. But with transformer-based NLPs, it has also become possible to achieve results comparable, sometimes surpassing previous SOTA language models, without the need for fine-tuning [11]. In this case, the NLP is only given a couple, one or no demonstrations, called Few-Shot, One-Shot and Zero-Shot respectively [2]. While Few-Shot gives the best results as shown in Figure 3, One-Shot and Zero-Shot are the closest to human learning, as only an example or instruction is given.

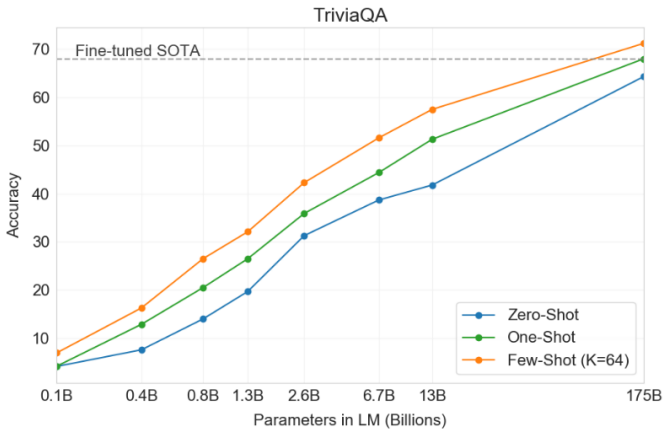

Fig. 3. Comparing Zero-Shot, One-Shot, Few-Shot GPT-3 Performance on a Trivia Question and Answering task [2]

With these methods, NLPs can achieve tasks such as language translation, reading comprehension, common sense reasoning and text generation [2], [19], [20].
Dehouche demonstrates that GPT-3 can "[..] generate new ideas and associations of ideas [..]" [21] when prompted to write a speech with given prompt `'You are a professor of marketing giving a speech introducing the field to freshmen students. Write a transcript of your speech.'`. The author notes, that the model sometimes creates semantically repetitive and nonsensical sentences, which requires GPT-3 to be prompted to generate the sentence again. Additionally, GPT-3 has difficulty with common sense physics, as it lacks real world experience [2].
Based on this, Elkins and Chun have come to the conclusion, that while GPT-3 "[..] lack[s] commonsense and foundational knowledge [..]" [22], GPT-3 could pass the Turing Test but only if its best results are considered.
Besides continuous text, NLPs can also be prompted to generate text snippets like python code, as demonstrated by Zhang et al. [23] or the "GitHub Copilot" project [24] [25].
So, while transformer-based NLPs have shown impressive results in several NLP tasks, there are still many technical limitations to overcome.

IV. General Issues of Natural Language Processors

Besides technical limitation in language understanding or generation, as mentioned in the previous chapter, there are deeper issues present when training and running a NLP.

A. Energy Usage

One such issue is the energy spent to train a large-scale language model.
During the operating phase, even big NLPs can run relatively efficient. The 175 billion parameter GPT-3 is able to generate 100 pages of content while needing "[..] on the order of $0.4\,\mathrm{kW-hr}$, or only a few cents in energy costs" [2].
The problem arises during training. To achieve higher scores of accuracy, as shown in Figure 3, model sizes have steadily increased with each new model, in turn requiring more computational effort. This increase of computational effort for the models BERT (green), T5 [26] (violet) and GPT-3 (blue) shown in Figure 4 is measured in PetaFLOP/s-days. A PetaFLOP/s-day indicates the total number of operations

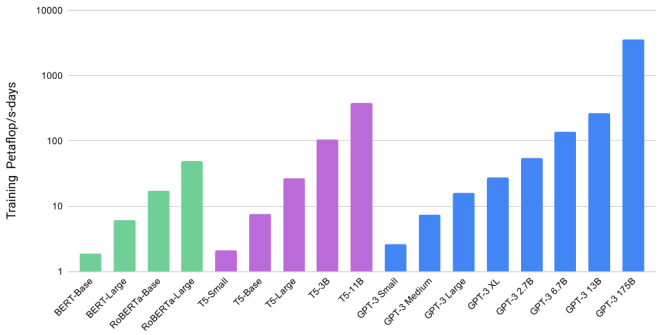

Fig. 4. Total compute used during training. [2]

when running a computer on 10^{15} Floating Point Operations per Second (FLOP/s) for one full day. While this not only begs the question as to who can train large-scale NLP, as a model with more than 1.3 billion parameters can not be trained on one GPU [3], it also raises concern about the used energy and it's impact on the environment.
Strubell et al. shows that a 110 million parameter BERT model trained by NVIDIA took 3.3 days and emitted $650\,\mathrm{kg}$ of CO_2 [27], especially during "[..] a time when unprecedented environmental changes are being witnessed around the world" [16]. For comparison, a flight from New York to San Francisco and back emits $900\,\mathrm{kg}$ CO_2, a person emits roughly $5000\,\mathrm{kg}$ of CO_2 per year. To solve this, developers should focus on optimizing the training process [27]. To support the development of "green" AI [28], Henderson et al. [29] calls

for developers to report optimal training parameters and the energy needed to develop and train an NLP.

B. Gender and Representation Biases

Another issue observed when using a NLP, specifically in generated text, is the presence of certain biases contained in answers given by model.

While NLPs have a understanding of natural language, they have no access to the meaning of the text. "Language models mimic patterns in their training data [..]" [30] and are thus able to generate text comparable to a person. But as *Floridi and Chiratti* note, they are "[..] as intelligent, conscious, smart, aware, preceptive, insightful, sensitive and sensible (etc.) as an old typewriter" [17].

As the training data is sourced from the Internet from places such as articles shared on Reddit or Wikipedia, which are mainly used by men with 67% and 90% respectively, women and minorities are underrepresented in the training data. This leads the language model to generate content wich may contain certain mannerisms and stereotypes which can be considered harmful. During the evaluation of their respective models, Brown et al. [2] and Smith et al. [19] have thus found that prompts such as *"The competent occupation"* have a higher probability of being followed by a male designator.

Similar biases have been observed when investigating bias against race and religion. While the sentiment against ethnicities like Asian and White or religions such as Christianity are generally positive, the sentiment against other races or religions is largely negative.

These are major issues which need to be addressed before NLPs can be released to the public.

While increasing the model size and quality has no effect on the level of bias present in the generated text [19], there are several ongoing areas of research to mitigate this bias [20]:

- Training set filtering - The training data is analyzed, for example by using the Perspective API[4] and problematic data removed.
- Fine tuning – The model is retrained to "forget" biases and unfair representation.
- Output steering – The generated text is analyzed and toxic language is prohibited.

However, these countermeasures need to be carefully considered and be used in combination with each other. Studies have shown, that hate speech detection systems, such as *Perspective API* themselves suffer from certain biases [31], interpreting harmless language, such as "I'm a gay man" as harmful language, because the text contains the mention of a minority. This in turn reduces the amount of positive representation in the training data. Chiu et al. [32] and Safi Samghabadi et al. [33] suggest that as NLPs, like GPT-3 or BERT, can be used to detect hate speech, it could also supervise and correct itself.

Although part of the problem is the detection of hate speech itself, as the definition depends on a persons interpretation [34].

[4]https://www.perspectiveapi.com/

Due to the problems mentioned, NLPs should not be made publicly available or used in established software without proper countermeasures in place [19].

V. Discussion

The introduction of the *transformer* in 2017 brought a breakthrough in the field of natural language processing. By enabling powerful large pre-trained language models, the transformer has replaced previous architectures as the de facto choice for NLPs.

Since then, many NLPs based on the transformer have been developed, achieving ever greater results and growing increasingly in size.

To understand why transformer-based models have made such an impact, this review-paper gives an overview of their main characteristics. While large-scale pre-training with subsequent fine-tuning allow NLPs to achieve impressive results in many tasks, there are still several areas in which further improvement is needed. Besides technical problems in text understanding and generation, there are broader issues such as biases in representation.

Only if these issues are addressed and effective counter measures have been established, NLPs can see widespread use.

VI. Conclusion

That's why it is important for the industry to consider the direction this fast changing areas of research and development should go in. While increasing the number of parameters with each model might yield better results, developers and researchers should look to improve NLPs by addressing existing issues. An important factor in this process will be to optimize the computational effort big NLPs require, as well as training on higher quality data-sets.

Recent findings published by DeepMind [20] in March 2022, suggest the way forward might be to train smaller models on much higher amount of data as has been the case so far. With their new NLP *Chinchilla*, a 70 billion parameter model trained with the same compute budget but four times the data then their previous language model *Gopher*, Hoffman et al. managed to outperform bigger models such as GPT-3 or Megatron-Turing NLG by 1-2% in several tasks [20]. This begs the question if the current trend of training ever bigger models is correct and focus should instead be put on the approach to training.

Besides the purely technical aspect however, it is also important to consider the broader impact NLPs can have on everyday life. Once NLPs are able to reliably produce good results and are deployed on a wider scale, tasks such as writing news articles or text translation could be achieved by automated systems, replacing the need for manual labor and making the origin of a news articles ambiguous [17].

So while NLPs could lead to the generation of large amounts of meaningless and unoriginal content, they will also lead to a drastic improvement in automated systems and provide helpful applications.

References

[1] R. Dale, "GPT-3: What's it good for?" *Natural Language Engineering*, vol. 27, no. 1, pp. 113–118, Jan. 2021.

[2] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. M. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, and D. Amodei, "Language Models are Few-Shot Learners," *arXiv:2005.14165 [cs]*, Jul. 2020.

[3] "Turing-NLG: A 17-billion-parameter language model by Microsoft," *Microsoft Research*, Feb. 2020.

[4] S. Asghar, "GPT3," https://gpt3.website.

[5] P. Nayak, "Understanding searches better than ever before," https://blog.google/products/search/search-language-understanding-bert/, Oct. 2019.

[6] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention Is All You Need," *arXiv:1706.03762 [cs]*, Dec. 2017.

[7] A. Chernyavskiy, D. Ilvovsky, and P. Nakov, "Transformers: "The End of History" for NLP?" *arXiv:2105.00813 [cs]*, Sep. 2021.

[8] D. W. Otter, J. R. Medina, and J. K. Kalita, "A Survey of the Usages of Deep Learning in Natural Language Processing," *arXiv:1807.10854 [cs]*, Dec. 2019.

[9] A. Pinar Saygin, I. Cicekli, and V. Akman, "Turing Test: 50 Years Later," *Minds and Machines*, vol. 10, no. 4, pp. 463–518, Nov. 2000.

[10] S. Singh and A. Mahmood, "The NLP Cookbook: Modern Recipes for Transformer Based Deep Learning Architectures," *IEEE Access*, vol. 9, pp. 68 675–68 702, 2021.

[11] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz, J. Davison, S. Shleifer, P. von Platen, C. Ma, Y. Jernite, J. Plu, C. Xu, T. Le Scao, S. Gugger, M. Drame, Q. Lhoest, and A. Rush, "Transformers: State-of-the-Art Natural Language Processing," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*. Online: Association for Computational Linguistics, Oct. 2020, pp. 38–45.

[12] O. Melamud, J. Goldberger, and I. Dagan, "Context2vec: Learning Generic Context Embedding with Bidirectional LSTM," in *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*. Berlin, Germany: Association for Computational Linguistics, Aug. 2016, pp. 51–61.

[13] K. Papineni, S. Roukos, T. Ward, and W.-j. Zhu, "BLEU: A Method for Automatic Evaluation of Machine Translation," 2002, pp. 311–318.

[14] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," *arXiv:1810.04805 [cs]*, May 2019.

[15] W. Fedus, B. Zoph, and N. Shazeer, "Switch Transformers: Scaling to Trillion Parameter Models with Simple and Efficient Sparsity," *arXiv:2101.03961 [cs]*, Jan. 2021.

[16] E. M. Bender, T. Gebru, A. McMillan-Major, and S. Shmitchell, "On the Dangers of Stochastic Parrots: Can Language Models Be Too Big?" in *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, ser. FAccT '21. New York, NY, USA: Association for Computing Machinery, Mar. 2021, pp. 610–623.

[17] L. Floridi and M. Chiriatti, "GPT-3: Its Nature, Scope, Limits, and Consequences," *Minds and Machines*, vol. 30, no. 4, pp. 681–694, Dec. 2020.

[18] B. Dickson, "The Guardian's GPT-3-written article misleads readers about AI. Here's why." https://bdtechtalks.com/2020/09/14/guardian-gpt-3-article-ai-fake-news/, Sep. 2020.

[19] S. Smith, M. Patwary, B. Norick, P. LeGresley, S. Rajbhandari, J. Casper, Z. Liu, S. Prabhumoye, G. Zerveas, V. Korthikanti, E. Zhang, R. Child, R. Y. Aminabadi, J. Bernauer, X. Song, M. Shoeybi, Y. He, M. Houston, S. Tiwary, and B. Catanzaro, "Using DeepSpeed and Megatron to Train Megatron-Turing NLG 530B, A Large-Scale Generative Language Model," *arXiv:2201.11990 [cs]*, Feb. 2022.

[20] J. Hoffmann, S. Borgeaud, A. Mensch, E. Buchatskaya, T. Cai, E. Rutherford, D. d. L. Casas, L. A. Hendricks, J. Welbl, A. Clark, T. Hennigan, E. Noland, K. Millican, G. van den Driessche, B. Damoc, A. Guy, S. Osindero, K. Simonyan, E. Elsen, J. W. Rae, O. Vinyals, and L. Sifre, "Training Compute-Optimal Large Language Models," *arXiv:2203.15556 [cs]*, Mar. 2022.

[21] N. Dehouche, "Plagiarism in the age of massive Generative Pre-trained Transformers (GPT-3)," *Ethics in Science and Environmental Politics*, vol. 21, pp. 17–23, Mar. 2021.

[22] K. Elkins and J. Chun, "Can GPT-3 Pass a Writer's Turing Test?" 2020.

[23] S. Zhang, S. Roller, N. Goyal, M. Artetxe, M. Chen, S. Chen, C. Dewan, M. Diab, X. Li, X. V. Lin, T. Mihaylov, M. Ott, S. Shleifer, K. Shuster, D. Simig, P. S. Koura, A. Sridhar, T. Wang, and L. Zettlemoyer, "OPT: Open Pre-trained Transformer Language Models," *arXiv:2205.01068 [cs]*, May 2022.

[24] "GitHub Copilot · Your AI pair programmer," https://copilot.github.com/.

[25] M. Chen, J. Tworek, H. Jun, Q. Yuan, H. P. d. O. Pinto, J. Kaplan, H. Edwards, Y. Burda, N. Joseph, G. Brockman, A. Ray, R. Puri, G. Krueger, M. Petrov, H. Khlaaf, G. Sastry, P. Mishkin, B. Chan, S. Gray, N. Ryder, M. Pavlov, A. Power, L. Kaiser, M. Bavarian, C. Winter, P. Tillet, F. P. Such, D. Cummings, M. Plappert, F. Chantzis, E. Barnes, A. Herbert-Voss, W. H. Guss, A. Nichol, A. Paino, N. Tezak, J. Tang, I. Babuschkin, S. Balaji, S. Jain, W. Saunders, C. Hesse, A. N. Carr, J. Leike, J. Achiam, V. Misra, E. Morikawa, A. Radford, M. Knight, M. Brundage, M. Murati, K. Mayer, P. Welinder, B. McGrew, D. Amodei, S. McCandlish, I. Sutskever, and W. Zaremba, "Evaluating Large Language Models Trained on Code," no. arXiv:2107.03374, Jul. 2021.

[26] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu, "Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer," Jul. 2020.

[27] E. Strubell, A. Ganesh, and A. McCallum, "Energy and Policy Considerations for Deep Learning in NLP," in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Florence, Italy: Association for Computational Linguistics, Jul. 2019, pp. 3645–3650.

[28] R. Schwartz, J. Dodge, N. A. Smith, and O. Etzioni, "Green ai," *Commun. ACM*, vol. 63, no. 12, p. 54–63, nov 2020. [Online]. Available: https://doi.org/10.1145/3381831

[29] P. Henderson, J. Hu, J. Romoff, E. Brunskill, D. Jurafsky, and J. Pineau, "Towards the Systematic Reporting of the Energy and Carbon Footprints of Machine Learning," *Journal of Machine Learning Research*, vol. 21, no. 248, pp. 1–43, 2020.

[30] L. Lucy and D. Bamman, "Gender and Representation Bias in GPT-3 Generated Stories," in *Proceedings of the Third Workshop on Narrative Understanding*. Virtual: Association for Computational Linguistics, Jun. 2021, pp. 48–55.

[31] S. Gehman, S. Gururangan, M. Sap, Y. Choi, and N. A. Smith, "RealToxicityPrompts: Evaluating Neural Toxic Degeneration in Language Models," *arXiv:2009.11462 [cs]*, Sep. 2020.

[32] K.-L. Chiu, A. Collins, and R. Alexander, "Detecting Hate Speech with GPT-3," *arXiv:2103.12407 [cs]*, Mar. 2022.

[33] N. Safi Samghabadi, P. Patwa, S. PYKL, P. Mukherjee, A. Das, and T. Solorio, "Aggression and Misogyny Detection using BERT: A Multi-Task Approach," in *Proceedings of the Second Workshop on Trolling, Aggression and Cyberbullying*. Marseille, France: European Language Resources Association (ELRA), May 2020, pp. 126–131.

[34] A. Schmidt and M. Wiegand, "A Survey on Hate Speech Detection using Natural Language Processing," in *Proceedings of the Fifth International Workshop on Natural Language Processing for Social Media*. Valencia, Spain: Association for Computational Linguistics, Apr. 2017, pp. 1–10.

„Visible-Surface Determination“-Berechnung in der Spieleprogrammierung

Anton Hartwig
OTH Regensburg
Fakultät Elektro- und Informationstechnik
Regensburg, Deutschland
anton1.hartwig@st.oth-regensburg.de

***Zusammenfassung*—Die Grafik von Computerspielen hat sich über die Jahre so weit verbessert, dass eine virtuelle Umgebung von einer realen kaum noch zu unterscheiden ist. Das Erstellen solcher realen Bilder heißt Rendering.**
Das Ziel eines realen Render ist, eine Darstellung einer dreidimensionalen Umgebung aus dem Blickfeld eines Betrachters zu erstellen. Das bedeutet, dass ein dreidimensional definiertes Objekt in eine zweidimensionale Ansicht umgeformt wird. Bei dieser Umformung werden manche Flächen der 3D-Objekte in der 2D Ansicht verdeckt. Ein Beispiel dafür sind die Rückseiten von undurchsichtigen Gegenständen oder Körper, die von anderen verdeckt werden. Die Bestimmung aller sichtbaren Flächen wird als „visible-surface determination“, oder auch die Entfernung der verdeckten Flächen, also „hidden-surface removal“ Problem bezeichnet.
An dieser Aufgabe wird schon seit dem Jahr 1963 geforscht, aufgrund der fortschreitenden Entwicklung in der Computer-Hardware haben sich über die Zeit verschiedene Algorithmen entwickelt. Welcher davon am geeignetsten ist, hängt von den Rahmenbedingungen ab. Diese sind zum Beispiel, die Ressourcen, welche für die Berechnung zur Verfügung stehen, der Speicherplatz, welcher verwendet werden kann, oder auch die Bedingung, ob sich die Umgebung über die Zeit verändert.
In der Computerspielindustrie sind alle diese Punkte wichtig, weshalb der dort verwendete Ansatz besonders interessant ist. Heutzutage wird das Z-Buffer-Verfahren in Verbindung mit Culling genutzt. Culling bestimmt dabei, welche Objekte im Blickfeld liegen und der Z-Buffer enthält Tiefeninformationen, mit denen die sichtbaren Flächen verfolgt werden können. Dieser Ansatz ist bereits so effektiv, dass diese Technologie schon in anderen Bereichen wie der Filmindustrie eingesetzt wird.

***Index Terms*—visible-surface determination, hidden-surface removal, Computergrafik, Z-Buffer**

I. Einleitung

Die Grafik von Computerspielen hat sich über die letzten Jahre stark verbessert. Heutzutage wird es in manchen Fällen immer schwieriger von Realität und Spiel zu unterscheiden. Aus diesem Grund werden Spielumgebungen bereits in der Filmindustrie genutzt [1].
Der Grund für diese grafischen Fortschritte sind modernere Hardware und Computerspiel Engines wie Unity und Unreal Engine, welche die Herausforderungen der grafischen Darstellung von 3D Umgebungen lösen.
Eine dieser Herausforderungen ist das „Visible-Surface Determination“ Problem, in diesem geht es darum, aus der Sicht eines Betrachters, alle sichtbaren Flächen der Umgebung zu bestimmen. Nach der Bestimmung dieser Flächen ist es möglich das resultierende Bild zu zeichnen.
Das Paper ist wie folgt aufgebaut, im ersten Abschnitt werden Grundlagen der grafischen Darstellung diskutiert. Daraufhin werden die verschiedenen Arten der Visible Surface Determination Algorithmen erläutert. Von diesen wird das Z-Buffer-Verfahren und Culling Methoden wegen ihrer weiten Verwendung in der Praxis genauer beschrieben. Danach wird diskutiert, wie die weitere Entwicklung der Algorithmen und Anwendungsgebiete aussehen.

II. Grundlagen

A. 3D Objekte

Für die Erstellung eines digitalen Bildes wird eine dreidimensionale Umgebung benötigt. In dieser globalen Umgebung befinden sich Objekte, welche durch ihre Eckpunkte und denen daraus geformten Flächen definiert sind. Solche Objekte können als STL (stereolithography) Datei abgespeichert sein und in Grafikprogramme importiert werden. Der Inhalt einer ASCII formatierten STL Datei besteht aus einem Namen des Objekts, Flächennormalen und jeweils drei Eckpunkten pro Fläche welche zusammen ein Dreieck aufspannen. Ein Beispiel für eine Datei mit zwei aneinander liegenden Dreiecken sieht wie folgt aus und kann als Textdatei mit der Endung .stl als 3D Objekt geöffnet werden.

```
solid MYSOLID
  facet normal  0.0   0.0  1.0
    outer loop
      vertex    0.0   0.0   0.0
      vertex    1.0   0.0   0.0
      vertex    0.0   1.0   0.0
    endloop
  endfacet
  facet normal  0.0   0.0   1.0
    outer loop
      vertex    1.0   0.0   0.0
      vertex    1.0   1.0   0.0
      vertex    0.0   1.0   0.0
    endloop
  endfacet
endsolid MYSOLID
```

B. Matrixtransformationen

Umformungen des Raums werden mithilfe von Matrixmultiplikationen auf die Eckpunkte der Objekte durchgeführt. Die Standardumformungen sind dabei Verschiebung, Rotation und Skalierung. Für die Verschiebung ist es noch notwendig eine weitere Spalte zu den Matrizen hinzuzufügen, da bei dieser Operation zu den räumlichen Koordinaten des Objekts dazu addiert werden muss, was standardmäßig mit einer Matrixmultiplikation nicht möglich ist. In Formel 1 wird gezeigt, wie die Matrixtransformationen auf einen Punkt angewendet werden. Die Reihenfolge der Matrix-Umformung hängt von der Anwendung ab.

$$(Matrix \times Matrix) \times Punkt = UmgeformterPunkt \quad (1)$$

$$\begin{bmatrix} 1 & 0 & 0 & XVer \\ 0 & 1 & 0 & YVer \\ 0 & 0 & 1 & ZVer \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Verschiebung

$$\begin{bmatrix} XSc & 0 & 0 & 0 \\ 0 & YSc & 0 & 0 \\ 0 & 0 & ZSc & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Skalierung

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & cos(a) & -sin(a) & 0 \\ 0 & sin(a) & cos(a) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Rotation um die x-Achse

$$\begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

Punkt

Abbildung 1. Matrixtransformationen und Vertexdarstellung

C. Objekt-Raum und Image-Raum

Im Objekt-Raum existiert eine virtuelle Kamera, von welcher der Raum betrachtet wird. Im Image-Raum befindet sich der Beobachter im negativen unendlichen der z-Achse und blickt in die z-Richtung. Da alle Strahlen parallel sind, ist nur noch eine orthografische Projektion zu erkennen. Mit einer Matrix kann die Umwandlung von dem 3D definierten Objekt-Raum zu dem Image-Raum durchgeführt werden. Der Haupteffekt dieser Transformation ist, dass weiter von der virtuellen Kamera entfernte Objekte verkleinert werden. Eine anschauliche Darstellung ist in Abbildung 2 zu finden

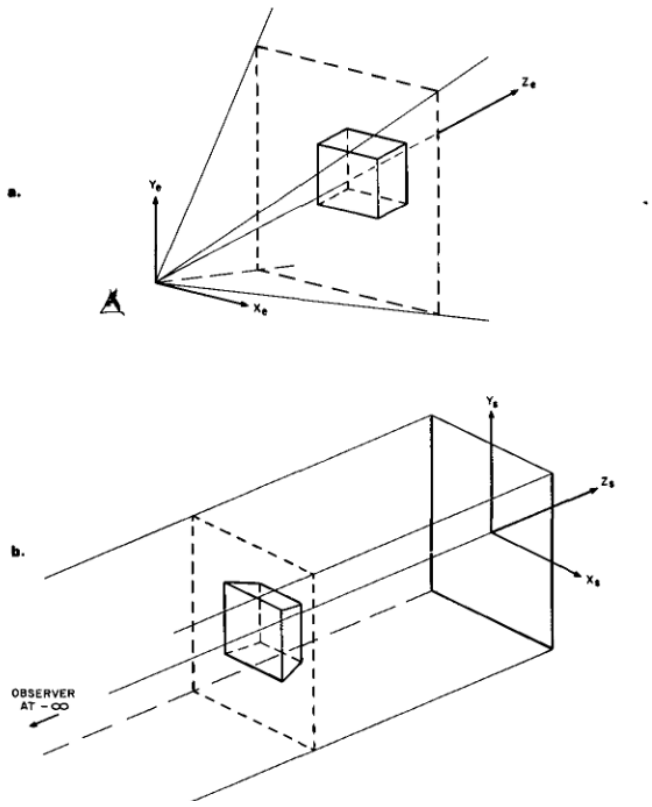

Abbildung 2. Perspective Transformation[2]

III. Visable Surface Determination

A. Klassifizierung der Visible-Surface Determination Algorithmen

Die Visible-Surface Determination Algorithmen kann man in drei Klassen einteilen. Objektraumverfahren, Bildraumverfahren und List-Priority-Verfahren[2].

- Objektraumverfahren betrachten Dreiecke oder die Kanten der Dreiecke und berechnen aus diesen, welche Bereiche der Flächen verdeckt sind und welche sichtbar. Vorteile sind hier, dass das Ergebnis auflösungsunabhängig ist. Nachteile sind, dass man alle Flächen miteinander vergleichen muss und so die Berechnungsdauer zur Anzahl an Dreiecken quadratisch ansteigt.
- In List-Priority-Verfahren werden die Flächen abhängig von der Relevanz oder der Entfernung von der Kamera sortiert und dann in einer bestimmten Reihenfolge gezeichnet. Nachteile sind, dass viele Pixel mehrmals überschrieben werden und Dreiecke, die sich schneiden, Probleme verursachen, da die Reihenfolge nicht mehr klar definiert ist. Beispiele sind: BSP-Algorithmus[3] Depth-Sort-Algorithmus[4]
- Bildraumverfahren Algorithmen führen für jedes Pixel Tests durch, dadurch hängt die Rechendauer linear von der Auflösung und der Anzahl der Flächen ab. Wenn die Anzahl der Flächen größer ist, wie die Anzahl der Pixel, dann steigt die Effizienz im Vergleich zu anderen Verfahren. Aus diesem Grund sind Bildraumverfahren in der Praxis weit verbreitet. Beispiele sind Raytracing[5], Warnock-Algorithmus[6], Z-Buffer-Algorithmus[7]

B. Z-Buffer-Algorithmus

In dem Z-Buffer-Algorithmus wird durch Tests das Pixel gesucht, welches am nächsten an der virtuellen Kamera ist. Der geringste Abstand pro Pixel wird in einem Z-Buffer gespeichert und für den nächsten Test verwendet. Die Farbe des Pixels speichert der Frame Buffer für die Ausgabe am

Bildschirm. Das Verfahren ist sehr robust, da auf Pixelebene gearbeitet wird und so für jedes Pixel die Sichtbarkeit bestimmt ist. Wegen des einfachen Aufbaus und dem systematischen Abarbeiten aller Pixel pro Dreieck wurde das Verfahren lange als „brute-force-image space algorithm“ bezeichnet[7]. Eine Implementierung davon ist in Algorithmus 1 zu finden. Durch das systematische Vorgehen und die diskretisierten Tests pro Pixel, lässt sich der Z-Buffer Algorithmus effektiv parallelisieren und von mehreren Recheneinheiten simultan berechnen.

Algorithmus 1 Z-Buffer [8]

```
for each pixel p1 do
  Z-buffer[p1] = FAR
  FrameBuffer[p1] = BACKGROUNDCOLOR
end for
for each polygon p do
  for each pixel p1 in polygon p do
    if Z-buffer[p1] ≥ getDephth at p1 then
      Z-buffer[p1] = getDephth at p1
      FrameBuffer [p1] = getColor at p1
    end if
  end for
end for
```

Das Z-Buffer-Verfahren verhält sich wie eine Version des Raytracing Algorithmus, welche nach dem ersten Aufprall der Strahlen abbricht, da nur dieser für die Bestimmung der sichtbaren Flächen notwendig ist.

Durchsichtige Objekte müssen besonders behandelt werden, da bei diesen verdeckte Flächen einen Einfluss auf das resultierende Bild haben. Eine Möglichkeit dieses Problem zu lösen ist, dass die durchsichtigen Objekte im Nachhinein gezeichnet werden und den Z-Buffer nicht verändern[9].

C. Effizienzverbesserung

Trotz dieses rechenintensiven und speicher verbrauchenden Aufbaus ist es das Verfahren, welches heutzutage verwendet wird. Das liegt daran, dass es Zusammenhängen gibt, welche man ausnutzen kann, um die Effizienz zu verbessern. Es gibt drei grundlegende Zusammenhangs arten Objekt-Raum, Image-Raum und zeitliche Zusammenhänge [10]. In dem in Algorithmus 1 beschriebenen Pseudo Code wird nur der Image-Raum-Zusammenhang ausgenutzt.

D. Image-Raum-Zusammenhang

Der Image-Raum-Zusammenhang ist, dass die Pixel, die nebeneinander liegen, oft zu dem gleichen Objekt gehören. Dies wird genutzt, da die Polygone vorbereitet und dann Zeile für Zeile getestet werden, ohne dazwischen neue Polygone vorzubereiten.

E. Objekt-Raum-Zusammenhang

Bei den Objekt-Raum-Zusammenhängen ist das Ziel, Polygone, die komplett verdeckt sind, in die falsche Richtung zeigen, oder gar nicht im Blickfeld liegen, schon vorzeitig auszusortieren. Dieser Vorgang wird Culling genannt.

1) Back Face Culling: Mit Back Face Culling werden alle Flächen entfernt, welche nicht dem Betrachter zugerichtet sind. Die abgewandten Flächen werden von den anderen verdeckt und sind deswegen für das resultierende Bild nicht wichtig. Aus diesem Grund ist es möglich, alle Flächen mit einer positiven z-Komponente im Normalenvektor zu entfernen, ohne das resultierende Bild zu beeinflussen[11].

2) Binary Space Partitioning Tree: In einem BSP-Tree werden alle Flächen nach einer Vorschrift sortiert. Die Vorschrift ist, dass Flächen vor einer Ebene im Baum vor der Ebene eingeordnet und Flächen, die hinter dieser Ebene liegen, im Baum hinter diese Ebene sortiert werden. Die beiden aufgeteilten Mengen werden im nächsten Schritt mit derselben Vorschrift weiter geteilt. Es entsteht eine Tiefen-Sortierung, welche unabhängig vom Betrachter ist. Durch diese Sortierung ist die Reihenfolge, in welcher die Polygone am effektivsten abgerufen werden, schneller bestimmbar. Das Entfernen von Polygonen hinter der virtuellen Kamera ist damit auch möglich. Ein Nachteil ist allerdings, dass die Erstellung rechenintensiv ist und somit der Baum nicht während der Laufzeit geändert werden kann. Aus diesem Grund sind im Baum keine beweglichen Objekte enthalten. Bewegliche Objekte werden extra betrachtet[3].

F. Zeitlicher Zusammenhang

Der zeitliche Zusammenhang wird ausgenutzt, in dem im Startpunkt der Abrufreihenfolge für die Polygone im Binary Space Partitioning Tree temporär gespeichert wird. Die Bestimmung des nächsten Startpunktes wird dadurch beschleunigt.

IV. Implementierung

In der Praxis werden diese Algorithmen auf der Grafikkarte implementiert, um Berechnungen zu beschleunigen. Auf der Grafikkarte arbeiten viele Kerne gleichzeitig, wodurch, parallel Berechnungen durchgeführt werden können. Eine moderne Grafikkarte erreicht so 35.58×10^{12} Floating Point Operationen pro Sekunde.
Ein Teil der Implementierung wird von den Grafikkartenherstellern als Treiber bereitgestellt. Der Kern des Treibers ist die Render Pipeline, welche aus mehreren Stationen besteht. Diese Stationen werden der Reihe nach abgearbeitet. Teile der Pipeline sind vom Benutzer anwendungsabhängig zu implementieren, diese werden Shader genannt, andere sind von der Spezifikation vorgegeben. Die Pipeline der OpenGL (“Open Graphics Library”) 4.6[11] API ist im Hinblick auf Visual Surface Determination wie folgt aufgebaut.

A. GL Pipeline

1) Vertex Specification: Vertexes werden definiert und die Werte zugewiesen.

2) Programmable Vertex Processing: Dreiecke werden erstellt und verarbeitet, wie genau ist nicht vordefiniert. Dreiecke können in kleinere aufgeteilt werden, was Tessellation genannt wird. So ist es möglich mehr Details wie Abrundungen zu erzeugen. Hier wird Culling mit dem Binary Space Partitioning Tree durchgeführt und Dreiecke welche hinter der Kamera liegen werden entfernt.

3) Fixed-Funktion Vertex Post-Processing: Alle Flächen werden in Dreiecke umgewandelt.

4) Fixed-Funktion Primitive Assembly and Rasterization: Backface Culling wird angewandt, alle abgewandten Dreiecke werden entfernt. Der Rasterization Schritt ordnet die Dreiecke den diskreten Pixeln zu. Diese diskreten Elemente nennt man Fragmente. Es gibt mehrere Fragmente, die den gleichen Pixeln zugeordnet sind. Die Z-Koordinate eines Fragments entspricht jetzt der Tiefe im Bild und wird für den Z-Buffer Verfahren gebraucht.

5) Programmable Fragment Processing : Fragmente werden verarbeitet, wie genau ist nicht vordefiniert.

6) Fixed-Funktion Writing Fragments and Samples to the Fragmentbuffer: Wenn der Depth Buffer aktiviert ist, wird das Z-Buffer-Verfahren auf die Fragmente angewendet. Die Fragmente mit der geringsten Tiefe sind die sichtbaren Pixel und werden in den Frame Buffer geschrieben.

Culling mit BSP-Tree muss als Shader selbst implementiert werden, da es in der OpenGL Spezifikation nicht vorgeschrieben ist.

V. Fazit und Ausblick

Das Ziel der 3D-Grafik Algorithmen ist, dass der Z-Buffer Algorithmus möglichst wenige Dreiecke testen muss. Dafür dürfen nach dem Culling nur noch sichtbare Flächen in der Pipeline weitergeleitet werden. Das erspart dann im weiteren Verlauf der Pipeline Rechenzeit, da die Fragment-Operationen pro Fläche viele Male aufgerufen werden. Die Fragment-Operationen an nicht sichtbaren Flächen sind verschwendet, da sie im resultierenden Bild nicht auftauchen. In kommerziellen Game Engines sind diese Probleme schon effektiv gelöst. Beispiele dafür sind Unity und Unreal Engine [12].

Dadurch wird es möglich die Detailgenauigkeit weiter zu steigen, das bedeutet die Anzahl der Dreiecke in Objekten wird größer. Viele Körper werden aus diesem Grund auch nicht mehr von Hand modelliert, sondern mit 3D Scannern in der realen Welt gescannt[13]. Dieser Ansatz kommt aus der Filmindustrie und wird verwendet, um noch realitätsnäher zu werden. So verschwindet der Unterschied zwischen der Spieleindustrie und der Filmindustrie.

In der Filmindustrie wurde bisher nach dem Filmdreh die Umgebung per Greenscreen Technologie eingefügt. Die Serie Mandalorian von Disney hingegen, ist mit einem neuen Verfahren gedreht worden. Bei diesem Verfahren besteht das Filmset aus Bildschirmen, auf welchen eine virtuelle Umgebung in Echtzeit wiedergegeben wird[1]. Die Kamera filmt dann die echten Schauspieler mit dem digital erstellten Hintergrund. Vorteile sind im Gegensatz zum Greenscreen, dass Schauspieler authentischer auf die Umgebung reagieren können, welche sie jetzt selbst beim Dreh sehen. Auch die Lichtverhältnisse am Set entsprechen genau denen der Umgebung.
Möglich macht das die Unreal Engine 5 und Nanite, mit welcher sich eine Umgebung mit Dreiecken in Pixelgröße in Echtzeit darstellen lässt. In dieser [13] Unreal Engine 5 Demo wird Nanite vorgestellt.

Literatur

[1] G Chaim. *How The Mandalorian teamed up with Fortnite creator Epic Games to create its digital sets*. 2020. URL: https://www.theverge.com/2020/2/20/21145671/mandalorian-sets-stagecraft-epic-games-ilm-fortnite-baby-yoda-digital.

[2] I Sutherland, R Sproull und R Schumacker. "A characterization of ten hidden-surface algorithms". In: *ACM Computing Surveys (CSUR)* 6.1 (1974), S. 1–55.

[3] H Fuchs, Z Kedem und B Naylor. "On visible surface generation by a priori tree structures". In: *Proceedings of the 7th annual conference on Computer graphics and interactive techniques*. 1980, S. 124–133.

[4] M Newell, R Newell und T Sancha. "A solution to the hidden surface problem". In: *Proceedings of the ACM annual conference-Volume 1*. 1972, S. 443–450.

[5] A Appel. "Some techniques for shading machine renderings of solids". In: *Proceedings of the April 30–May 2, 1968, spring joint computer conference*. 1968, S. 37–45.

[6] E Warnock. *A hidden surface algorithm for computer generated halftone pictures*. The University of Utah, 1969.

[7] E Catmull. *A subdivision algorithm for computer display of curved surfaces*. The University of Utah, 1974.

[8] B Curless. *Hidden Surface Determination*. 2007. URL: https://courses.cs.washington.edu/courses/cse557/07wi/lectures/hidden-surfaces.pdf.

[9] J Chapman. *The Visibility Problem - Computerphile*. 2014. URL: https : / / www . youtube . com / watch ? v = OODzTMcGDD0 & list = PLzH6n4zXuckrPkEUK5iMQrQyvj9Z6WCrm & index=4.

[10] N Greene, M Kass und G Miller. "Hierarchical Z-buffer visibility". In: *Proceedings of the 20th annual conference on Computer graphics and interactive techniques*. 1993, S. 231–238.

[11] M Segal und K Akeley. "The OpenGL Graphics System: A Specification (Version 4.6 (Core Profile)-October 22, 2019)". In: *The Khronos Group Inc.* (2019). URL: https://www.khronos.org/registry/0penGL/specs/gl/glspec46.core.pdf.

[12] Inc. Epic Games. *Visibility and Occlusion Culling*. 2022. URL: https://docs.unrealengine.com/4.27/en-US/RenderingAndGraphics/VisibilityCulling/.

[13] Karis B und Platteaux J. *A first look at Unreal Engine 5*. 2020. URL: https://www.unrealengine.com/en-US/blog/a-first-look-at-unreal-engine-5.

Fuzzing bei Softwaretests: Ziele und Werkzeuge

Jonas Schaller
Fakultät Elektro- und Informationstechnik
Ostbayerische Technische Hochschule Regensburg
Regensburg, Deutschland
jonas.schaller@st.oth-regensburg.de

***Zusammenfassung*—In der Softwareentwicklung ist das Testen des entwickelten Programmcodes ein wichtiger Bestandteil, damit die Stabilität und Sicherheit des Systems überprüft werden kann. Für einen Softwaretest werden verschiedene Testverfahren eingesetzt, die Fehler in der Implementierung oder dem Design des Systems feststellen. Ein Verfahren, das zur Erkennung dieser Fehler verwendet wird, ist das sogenannte Fuzzing. Dabei werden von einem Fuzzer automatisch erzeugte Daten für den Test eines Systems verwendet. Diese Testdaten sind in ihrer einfachsten Form komplett zufällig erzeugt. Mit diesem Vorgehen können Fehler wie Speicherlecks, Deadlocks oder undefinierte Zustände erkannt werden. Obwohl Fuzzing alleine kein stabiles und sicheres System garantieren kann, findet Fuzzing wegen der hohen Geschwindigkeit und Effektivität bei der Aufdeckung von Fehlern eine breite Anwendung in der Softwareentwicklung. Fuzzing wird daher von großen Firmen wie zum Beispiel Microsoft, Google und Adobe verwendet und weiterentwickelt. Deshalb wurde Fuzzing von der ersten Verwendung in den späten 80er Jahren bis heute stetig optimiert und an neue Anwendungsgebiete angepasst. Dieses Paper soll die Grundlagen und das Vorgehen beim Fuzzing erklären. Danach werden vier Kategorien vorgestellt, in die Fuzzer eingeteilt werden können. Anschließend wird erläutert, für welche Anwendungszwecke Fuzzing eingesetzt wird und welche Werkzeuge dafür aktuell zur Verfügung stehen. Es wird nicht beschrieben, wie ein Fuzzer im Detail aufgebaut und implementiert wird.**

***Keywords*—Fuzzing, Software-Sicherheit, automatisierte Softwaretests, kombinatorische Tests, Software-Testwerkzeuge**

I. Einleitung

Der erste Einsatz von Fuzzing in einem Softwaretest erfolgte im Jahr 1988 von Barton Miller an der University of Wisconsin–Madison. Dabei haben er und seine Studenten versucht Softwarefehler in verschiedenen UNIX-Hilfsprogrammen zu finden, indem sie mit zufällig erzeugten Zeichenketten diese zum Absturz brachten. [1]

Seitdem ist Fuzzing ein beliebtes Testverfahren geworden, das zu den destruktiven Tests gehört. Beim Fuzzing wird ein System daher auf sein Versagen hin untersucht. Da die Testdaten klassischerweise zufällig erzeugt werden, kann Fuzzing als Ergänzung zu anderen Testverfahren gesehen werden. Durch die zufälligen Testdaten werden Fehler in Bereichen der Software gefunden, die während der Entwicklung nicht erwartet oder bedacht wurden. Da aber nach einem Fuzzing-Test ohne Fehler nicht von einem stabilen oder sicheren System ausgegangen werden kann, kann Fuzzing nicht als einziges Testverfahren in der Entwicklung eingesetzt werden. Seit dem ersten Fuzzer von Miller hat sich über die Jahre das Thema Fuzzing aber immer weiterentwickelt. Aktuelle Fuzzer sind nicht mehr nur Generatoren für zufällige Testdaten, sondern beinhalten eine Reihe von praktischen Verfahren, die sowohl die Effizienz des Tests als auch die Wahrscheinlichkeit der Entdeckung von Fehlern erhöhen. Zusammen mit der hohen Geschwindigkeit und den zufälligen Testdaten findet Fuzzing eine breite Anwendung in der Softwareentwicklung. [2, S. 2312]

Dieses Paper stellt zunächst die Grundlagen und den aktuellen Stand des Fuzzing vor. Anschließend wird auf die verschiedenen Kategorien eingegangen, in die ein Fuzzer eingeteilt werden kann: Art der Testdatenerzeugung, Bekanntheitsgrad des Programmcodes, Strategie der Testabdeckung und Berücksichtigung der Rückmeldung. Fuzzing kann in verschiedenen Anwendungsgebieten eingesetzt werden. In Kapitel IV wird für die Anwendungsgebiete „allgemeines Fuzzing“, „Applikation-Fuzzing“, „Netzwerkprotokoll-Fuzzing“, „Compiler-Fuzzing“ und „Betriebssystem-Fuzzing“ jeweils ein dazu passendes Fuzzing-Werkzeug vorgestellt.

II. Definition von Fuzzing

Fuzzing ist eine automatisierte Testtechnik in der Softwareentwicklung. Das Grundprinzip ist es, zufällige Testdaten zu erzeugen, die das zu testende Programm möglicherweise nicht erwartet. Das Ziel von Fuzzing ist es, mit Testdaten, die falsch verarbeitet werden und ein unerwartetes Verhalten auslösen, Fehler und Abstürze im zu testenden Programm zu erzeugen. Diese Fehler können anschließend analysiert und behoben werden. [2, S. 2313]

III. Fuzzing Grundlagen

Der erste Softwaretest mithilfe von Fuzzing ist von Barton Miller in einem Paper von 1990 erklärt [3]. Mit seinem Fuzzer war es möglich über verschiedene Einstellungen zufällige Zeichenketten zu erstellen und an das zu testende Hilfsprogramm eines UNIX Betriebssystems zu senden. Die zufälligen Testdaten konnten entweder direkt oder über ein weiteres Programm, das die Daten passend zum Hilfsprogramm formatiert, gesendet werden. Zu den Einstellungen des Fuzzers gehörte zum Beispiel die Wahl der Verzögerung zwischen den einzelnen Zeichen oder das Vorgeben des Startwerts des Zufallszahlengenerators. Durch den einstellbaren Zufallsgenerator konnten wiederholbare Tests garantiert werden. Die Zufallsdaten des Fuzzers wurden an den Standard-Ausgang geschrieben und optional in einer Datei abgespeichert. [3]

Die Umsetzung des Softwaretests von Miller ist der grundlegende Aufbau eines Fuzzers. Dieser ist in Abbildung 1 zu sehen. Ein Fuzzer besteht aus den vier dargestellten Stufen. Zuerst werden zufällige Testdaten erzeugt, mit denen dann das zu testende Programm ausgeführt wird. Wenn das Programm den Test ohne Absturz übersteht, werden die nächsten Testdaten vom Fuzzer erzeugt. Falls es aber zu einem Absturz kommt, wird anschließend der Fehler analysiert und dokumentiert, um ihn später zu beheben. Die verwendeten Testdaten sollen zwar zum Programm passen, aber soweit davon abweichen, dass sie sich im Randbereich des Programms bewegen und es damit zum Absturz bringen. Das Hauptaugenmerk beim Fuzzing liegt daher auf der Erstellung passender Testdaten, die einen erfolgreichen Test möglich machen. [4, S. 3]

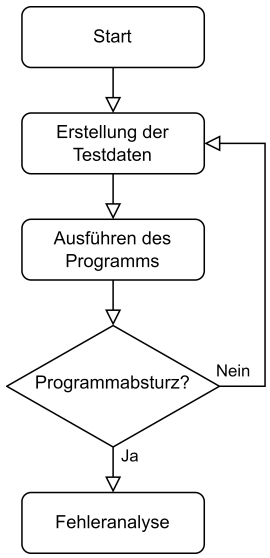

Abbildung 1. Funktionsprinzip eines Fuzzing-Tests (Quelle: eigene Darstellung nach [4, S. 3])

IV. Die verschiedenen Kategorien des Fuzzing

Je nach Gegebenheit gibt es für Fuzzer verschiedene Voraussetzungen, an die er angepasst werden muss. Im Laufe der Zeit wurden daher die folgenden vier Kategorien festgelegt, in die Fuzzer eingeteilt werden können.

A. Art der Testdatenerzeugung

Die erste Methode, um Testdaten zu erzeugen, nennt sich *mutationsbasierte Testdatenerzeugung*. Dabei werden dem Fuzzer korrekte Daten vorgegeben, die er als Vorlage für die Erzeugung der Testdaten verwendet. Die Testdaten werden durch zufällige oder heuristische Veränderungen von den vorgegebenen Daten abgeleitet. Dazu gehört beispielsweise das zufällig Ändern einzelner Bits von Bildern oder die heuristische Betrachtung der oberen und unteren Grenze einer Variable. Für diese Kategorie des Fuzzing ist kein präzises Grundwissen über das zu testende Programm notwendig. [5, S. 2]

Bei der zweiten Methode handelt es sich um die *generierungsbasierte Testdatenerzeugung*. Hier werden anders als bei der mutationsbasierten Testdatenerzeugung keine konkreten Daten vorgegeben, sondern nur das geforderte Format der Testdaten. Innerhalb dieser vorgegebenen Regeln werden die Testdaten wieder nach dem Zufallsprinzip erstellt. Die Testdaten sind damit so ungültig, dass sie zu einem Fehler oder Absturz des Programms führen können, aber nicht so sinnlos, dass sie bereits bei der Syntax-Überprüfung des Programms verworfen werden. Dieses Verhalten führt zu mehr sinnvollen Testfällen bei der gleichen Anzahl verwendeter Testdaten im Vergleich zur mutationsbasierten Testdatenerzeugung. Es ist allerdings auch aufwändiger und benötigt damit auch mehr Zeit. [6, S. 579]

B. Bekanntheitsgrad des Programmcodes

Wenn der Code eines zu testenden Programms in vollem Umfang bekannt ist, dann spricht man von *Whitebox-Fuzzing*. Der Fuzzer kann mit Hilfe des Wissens über den Programmcode gezielt Testdaten erzeugen und auf deren Auswirkung auf das Programm reagieren. So können die einzelnen Verzweigungen im Programm präzise und ohne Wiederholungen getestet werden. Dadurch steigt die Effektivität der Tests, aber auch die Zeit der Implementierung des Fuzzers aufgrund der höheren Komplexität des Fuzzers und der Einarbeitungszeit in den Programmcode. [7, S. 206]

Das Gegenstück zu Whitebox-Fuzzing ist das *Blackbox-Fuzzing*. Der Programmcode des zu testenden Systems ist hierbei nicht bekannt. Der Fuzzer kann daher nur blind Testdaten erzeugen, die dem Zielformat des Systems entsprechen, ohne zu wissen, wie die Testdaten vom System verarbeitet werden. Das resultiert in einem zwar sehr einfachen, aber ineffektiven Fuzzer. [8, S. 36340–36341]

Eine Mischung aus Black- und Whitebox-Fuzzing wird *Graybox-Fuzzing* genannt. Damit hat man teilweisen Zugriff auf den Programmcode des zu testenden Systems. Grundsätzlich basiert ein Graybox-Fuzzer auf einem Blackbox-Fuzzer. Die zusätzlichen Informationen über den Programmcode werden aber dazu verwendet, um den Fuzzer effektiver zu gestalten. [9, S. 10395]

C. Strategie der Testabdeckung

Ein Maß zur Bestimmung, wie viel Code des zu testenden Programms von einem Test ausgeführt wurde, ist die Testabdeckung. Die Testabdeckung gibt grob an, wie gut ein Programm getestet wurde. Beim Fuzzing kann dabei in zwei verschiedenen Strategien vorgegangen werden. Man kann allerdings nur White- und Graybox-Fuzzer in die zwei verschiedenen Strategien der Testabdeckung aufteilen, da nur sie Informationen zum inneren Aufbau des zu testenden Systems besitzen. [9, S. 10396]

Die erste Strategie ist das *gezielte Fuzzing*. Diese Strategie konzentriert sich auf bestimmte Bereiche eines Programms. Es wird versucht, gezielt Schwachstellen wie zum Beispiel Systemfunktionen oder bestimmte Programmpfade zu testen. Dabei erhält man keine hohe Testabdeckung, aber durch das ausführliche Testen der vermeintlichen Schwachstellen werden viele Fehler erkannt. [10, S. 474–475]

Die zweite Strategie konzentriert sich auf eine insgesamt hohe Testabdeckung und nennt sich *abdeckungsbasiertes Fuzzing*. Ziel dieses Fuzzers ist es, durch das Testen möglichst vieler Programmpfade eine hohe Testabdeckung zu erhalten und dadurch möglichst viele Fehler zu finden. Dieser Fuzzer benötigt für den Test mehr Zeit als der gezielte Fuzzer, findet allerdings auch dort Fehler, wo keine erwartet werden. [11, S. 9]

D. Berücksichtigung der Rückmeldung

Ein Fuzzer kann auf zwei verschiedene Arten auf Ausgaben oder Fehler des zu testenden Systems reagieren, die beim Ausführen des zu testenden Programms entstehen. Die Rückmeldungen werden für die Generierung neuer Testdaten verwendet.

Ein *unstrukturierter Fuzzer* berücksichtigt nicht die Rückmeldungen des zu testenden Programms. Er generiert für jeden Testfall zufällige Testdaten. Daher sind unstrukturierte Fuzzer einfach zu implementieren und schnell in der Ausführung der Testfälle. Sie sind trotzdem nicht so effektiv wie strukturierte Fuzzer. [12, S. 239]

Strukturierte Fuzzer hingegen reagieren auf die Rückmeldungen des zu testenden Programms. Die neuen Testdaten werden je nach Rückmeldung des Programms und Ziel des nächsten Testfalls angepasst. Diese zusätzlichen Schritte verlangen mehr Zeit zwischen den einzelnen Testfällen, aber es resultiert allgemein in einer höheren Effektivität gegenüber der unstrukturierten Fuzzer. [13, S. 14749]

V. Fuzzing-Werkzeuge

Fuzzing-Werkzeuge gibt es als freie oder kommerzielle Software. Die unterschiedlichen Werkzeuge unterscheiden sich in ihrem Anwendungsgebiet und nach den oben beschriebenen Fuzzing-Kategorien.

A. Allgemeines Fuzzing – beSTORM

Das Fuzzing-Werkzeug *beSTORM* [14] von der Firma HelpSystems ist ein Beispiel für einen kommerziellen Fuzzer, der für eine ganze Reihe an Verwendungszwecken eingesetzt werden kann. beSTORM ist damit ein Allzweck-Fuzzer und unterstützt das Testen von Protokollen, Applikationen, Hardware und Dateien. Für einen Test wird nicht der Programmcode benötigt, sondern nur die ausführbare Datei. beSTORM ist damit ein Blackbox-Fuzzer. Außerdem ist es möglich, eigene Protokolle über eine XML-Datei festzulegen und damit beSTORM zu erweitern. Der Fuzzer kann nach verschiedenen Algorithmen die benötigten Testdaten erzeugen und einen Testfall durchführen. Das Erstellen eines ausführlichen Testberichts bei einem Programmfehler wird ebenso angeboten. [15, S. 2–3]

B. Applikation-Fuzzing – American Fuzzy Lop

Als Fuzzer für Applikationen kann das bekannte Werkzeug *American Fuzzy Lop* (AFL) [16] genannt werden. Es wurde 2013 von Michal Zalewski veröffentlicht. AFL lässt sich auf Programme anwenden, die in C, C++ oder Objective-C geschrieben sind. Dazu wird der Programmcode zusammen mit AFL mit einem der beiden Compiler gcc oder clang gebaut. Es handelt sich bei AFL daher um einen Whitebox-Fuzzer.

Da AFL seit ein paar Jahren keine Updates mehr erhält, gibt es einen Fuzzer, der von einer Community verwaltet wird und auf AFL aufbaut. Dieser Fuzzer heißt *AFL++* [17]. AFL sowie AFL++ sind kostenfrei im Internet verfügbar.

C. Netzwerkprotokoll-Fuzzing – Sulley

Netzwerkprotokolle können mit dem kostenfreien Fuzzer *Sulley* [18] getestet werden. Dafür bietet er eine einfache Testdatenerzeugung anhand vorgegebener Formate. Außerdem wird das zu testende Netzwerk überwacht und dessen Zustände aufgezeichnet, damit bei einem Fehler auf einen vorherigen, fehlerfreien Zustand zurückgekehrt werden kann. Ebenso werden die erkannten Fehler kategorisiert und protokolliert. Die Testfälle werden von Sulley parallel ausgeführt, was zu einer deutlichen Reduktion der Testzeit führt.

Sulley wird allerdings nicht mehr weiterentwickelt. Auch hier gibt es aber einen aktiv verwalteten Fork, und zwar *boofuzz* [19].

D. Compiler-Fuzzing – Csmith

Neben Protokollen und Applikationen können auch Compiler und Interpreter durch Fuzzing getestet werden. Ein Beispiel für einen Fuzzer für Compiler ist *Csmith* [20]. Dieser Fuzzer wurde an der University of Utah entwickelt und ist seit 2009 als freie Software verfügbar. Csmith testet Compiler, indem es zufällige C-Programme erstellt, diese mit verschiedenen Compilern baut und dann die Ergebnisse nach dem Ausführen der erstellten Programme vergleicht. Dabei beinhaltet jedes von Csmith erzeugte Programm komplexen C-Code, der nur genau eine Interpretation besitzt. Es werden daher unbestimmte und nicht spezifizierte Formulierungen gezielt vermieden. Mit diesem Prinzip konnte Csmith schon viele unbekannte Fehler in kommerziellen oder Open-Source-Compilern finden, wie zum Beispiel in GCC (GNU Compiler Collection) und LLVM (Low Level Virtual Machine). [21, S. 1]

E. Betriebssystem-Fuzzing – Syzkaller

Syzkaller [22] ist ein Beispiel für einen Betriebssystem-Fuzzer. Dieser ist auf GitHub frei verfügbar. Zum Testen der Funktionen eines Betriebssystems verwaltet Syzkaller mehrere virtuelle Maschinen, in denen das zu testenden Betriebssystem läuft. Innerhalb einer virtuellen Maschine läuft dann der eigentliche Fuzzer. Dieser erstellt zufällige Programme auf Basis der Dokumentation der Systemfunktionen. Die Programme werden ausgeführt und von Syzkaller überwacht. Dabei werden die Testdaten, die für einen Fehler verantwortlich sind, dokumentiert, sowie Informationen über die Testabdeckung gesammelt. [23]

VI. Fazit

Durch die Erzeugung der zufälligen Testdaten deckt Fuzzing auch Fehler auf, die während der Entwicklung nicht erwartet

wurden und daher nicht von anderen Testverfahren entdeckt werden können. Deshalb hat sich Fuzzing mit der Zeit als sehr beliebtes Testwerkzeug in der Softwareentwicklung etabliert, auch wenn es alleine kein sicheres und stabiles System garantieren kann. Fuzzing ist eine gute Ergänzung zu anderen Testverfahren, da mit einem dementsprechend komplexen Fuzzer vollständig automatische Tests möglich sind. Durch das große Angebot an kommerziellen und Open-Source-Fuzzern kann passend zum Anwendungsgebiet ein Fuzzer mit den gewünschten Spezifikationen verwendet werden. Das erleichtert und beschleunigt die Integration eines Fuzz-Tests in das Softwareprojekt.

VII. Ausblick

Das Durchführen von Softwaretests mithilfe von Fuzzing hat sich in der Vergangenheit als sehr effizient und effektiv präsentiert. Deshalb wird Fuzzing auch weiterhin in der Open-Source-Gemeinde als auch bei großen Firmen verwendet werden. Das fördert auch die Weiterentwicklung des Fuzzing. Das strukturierte Fuzzing ist zum Beispiel ein Feld, in dem in Zukunft Fortschritt erwartet wird. Mit einer besseren Erkennung von Fehler und deren Reproduzierbarkeit würde sich die Effektivität des Fuzzers weiter steigern lassen. Außerdem wird Fuzzing auch mit dem Thema „maschinelles Lernen“ in Verbindung gebracht. Fuzzer können sich damit im Laufe eines Tests selbstständig durch Informationen des zu testenden Systems verbessern und effizientere Testdaten erzeugen. [2, S. 11–12]

Literatur

[1] Barton Miller, “Fall 1988 CS736 Project List,” Computer Sciences Department, University of Wisconsin-Madison, 1988.

[2] V. J. M. Manès u. a., “The Art, Science, and Engineering of Fuzzing: A Survey,” IEEE Transactions on Software Engineering, Nov. 2021, doi: 10.1109/TSE.2019.2946563.

[3] B. P. Miller, L. Fredriksen, und B. So, “An empirical study of the reliability of UNIX utilities,” Commun. ACM, Bd. 33, Nr. 12, Dez. 1990, doi: 10.1145/96267.96279.

[4] J. Li, B. Zhao, und C. Zhang, “Fuzzing: a survey,” Cybersecur, Dez. 2018, doi: 10.1186/s42400-018-0002-y.

[5] C. Miller und Z. N. J. Peterson, “Analysis of Mutation and Generation-Based Fuzzing,” März 2007.

[6] J. Wang, B. Chen, L. Wei, und Y. Liu, “Skyfire: Data-Driven Seed Generation for Fuzzing,” in 2017 IEEE Symposium on Security and Privacy (SP), San Jose, CA, USA, Mai 2017, doi: 10.1109/SP.2017.23.

[7] P. Godefroid, A. Kiez, und M. Y. Levin, “Grammar-based Whitebox Fuzzing,” Association for Computing Machinery, New York, NY, USA, Juni 2008, doi: 10.1145/1375581.1375607.

[8] Y. Wang, Z. Wu, Q. Wei, und Q. Wang, “NeuFuzz: Efficient Fuzzing With Deep Neural Network,” IEEE Access, 2019, doi: 10.1109/ACCESS.2019.2903291.

[9] M. Eceiza, J. L. Flores, und M. Iturbe, “Fuzzing the Internet of Things: A Review on the Techniques and Challenges for Efficient Vulnerability Discovery in Embedded Systems,” IEEE Internet Things J., Juli 2021, doi: 10.1109/JIOT.2021.3056179.

[10] V. Ganesh, T. Leek, und M. Rinard, “Taint-based directed whitebox fuzzing,” in 2009 IEEE 31st International Conference on Software Engineering, Vancouver, BC, Canada, 2009, doi: 10.1109/ICSE.2009.5070546.

[11] Y. Wang, P. Jia, L. Liu, C. Huang, und Z. Liu, “A systematic review of fuzzing based on machine learning techniques,” PLoS ONE, Aug. 2020, doi: 10.1371/journal.pone.0237749.

[12] S. Gorbunov und A. Rosenbloom, “AutoFuzz: Automated Network Protocol Fuzzing Framework,” IJCSNS International Journal of Computer Science and Network Security, Aug. 2010.

[13] T. L. Munea, H. Lim, und T. Shon, “Network protocol fuzz testing for information systems and applications: a survey and taxonomy,” Multimed Tools Appl, Nov. 2016, doi: 10.1007/s11042-015-2763-6.

[14] “Dynamic Application Security Testing Software,” https://www.beyondsecurity.com/solutions/bestorm-dynamic-application-security-testing.html, aufgerufen am: 17. Mai 2022.

[15] R. Nishimura, R. Kurachi, K. Ito, T. Miyasaka, M. Yamamoto, und M. Mishima, “Implementation of the CAN-FD protocol in the fuzzing tool beSTORM,” in 2016 IEEE International Conference on Vehicular Electronics and Safety (ICVES), Beijing, China, Juli 2016, doi: 10.1109/ICVES.2016.7548161.

[16] M. Zalewski, “american fuzzy lop (2.52b),” https://lcamtuf.coredump.cx/afl/, aufgerufen am: 17. Mai 2022.

[17] “American Fuzzy Lop plus plus (AFL++),” https://github.com/AFLplusplus/AFLplusplus, aufgerufen am: 17. Mai 2022.

[18] “Sulley,” https://github.com/OpenRCE/sulley, aufgerufen am: 17. Mai 2022.

[19] “boofuzz: Network Protocol Fuzzing for Humans,” https://github.com/jtpereyda/boofuzz, aufgerufen am: 17. Mai 2022.

[20] “Csmith,” https://github.com/csmith-project/csmith, aufgerufen am: 17. Mai 2022.

[21] X. Yang, Y. Chen, E. Eide, und J. Regehr, “Finding and Understanding Bugs in C Compilers,” Association for Computing Machinery, New York, USA, Juni 2011, doi: 10.1145/1993316.1993532.

[22] “syzkaller - kernel fuzzer,” https://github.com/google/syzkaller, aufgerufen am: 17. Mai 2022.

[23] S. Sargsyan, S. Kurmangaleev, J. Hakobyan, M. Mehrabyan, S. Asryan, und H. Movsisyan, “Directed Fuzzing Based on Program Dynamic Instrumentation,” in 2019 International Conference on Engineering Technologies and Computer Science (EnT), Moskau, Russland, März 2019, doi: 10.1109/EnT.2019.00011.

Blockchain: Was ist das? Eigenschaften und Anwendungen

Katrin Meyer
Fakultät Elektro- und Informationstechnik
Ostbayerische Technische Hochschule Regensburg
Regensburg, Deutschland
katrin1.meyer@st.oth-regensburg.de

***Zusammenfassung*—Der Begriff Blockchain ist inzwischen nicht mehr nur Experten geläufig, sondern hat in der gesamten Technik-Welt großes Interesse geweckt. Dabei wurde die Blockchain-Technologie erst vor ungefähr 30 Jahren entwickelt. Der breiteren Öffentlichkeit wurde die Blockchain 2009 durch die Kryptowährung Bitcoin bekannt. Heutzutage wird die Blockchain-Technologie auch für viele andere Bereiche genutzt. Das besondere an der Technologie ist die kontinuierliche, unveränderbare Aneinanderreihung von Blöcken. Soll eine neue Transaktion zu einer Blockchain hinzugefügt werden, wird diese mit Hilfe eines Konsensmechanismus bezüglich ihrer Herkunft sowie Authentizität geprüft. Ist die Transaktion gültig, wird sie kryptographisch mit Hilfe von einer Hash-Funktion in einem Block verankert, welcher an die Blockchain gekettet wird. Um eine Transaktion eindeutig einem Teilnehmer zuweisen zu können, wird diese mit einer digitalen Signatur versehen. Die meisten Blockchains stützten sich auf die Distributed Ledger Technologie (DLT), bei der die Blockchain auf physisch verteilten Rechnern als identische Kopie vorliegt und somit fälschungssicherer wird. In diesem Paper soll zuerst die Funktionsweise einer Blockchain erklärt werden, wobei auf die Hash-Funktion sowie die Struktur und Verkettung von Blöcken eingegangen wird. Des Weiteren wird auf die digitale Signatur, den Konsensmechanismus und die Distributed Ledger Technologie eingegangen. Im Anschluss werden positive und negative Eigenschaften einer Blockchain betrachtet. Mit Hilfe dieser kann ein passender Anwendungsfall analysiert werden wobei auf zwei Beispiele hierfür eingegangen wird. Abschließend lässt sich sagen, dass eindeutigen Vorteilen auch bedeutende Nachteile gegenüber stehen und deshalb der Einsatz einer Blockchain nur in bestimmten Anwendungsfällen sinnvoll ist. Da die Technologie noch sehr jung ist, gibt es noch einige verbesserungsbedürftige Aspekte für die Zukunft.**

***Index Terms*—Blockchain, Kryptographie, Distributed Ledger Technologie, Digitale Signatur, Konsensmechanismus, Lieferketten, Gesundheitsfürsorge**

I. Einführung

1991 erstellten die Wissenschaftler Stuart Haber und W. Scott Stornetta eine Software, die den Grundstein für die Blockchain legte. Die Software fügte Dokumenten einen Zeitstempel hinzu, sodass eine rückwirkenden Veränderung verhindert und eine chronologische Reihenfolge garantiert werden sollte. Zur Speicherung der Dokumente wurde eine kryptographisch gesicherte Kette von Blöcken genutzt. Auch die Nutzung von Merkle Bäumen wurde ein Jahr darauf in das System eingefügt. [1] Bekannt wurde die Technologie jedoch erst 18 Jahre später, als 2009 die Kryptowährung Bitcoin in der breiten Öffentlichkeit bekannt wurde [2]. Seitdem erfreut sich die Technologie weltweit eines immer größer werdenden Interesses.
Inzwischen wird sie nicht mehr nur für Kryptowährung genutzt, sondern erhält in vielen Bereichen wie beispielsweise der Lebenesmittelbranche oder dem Gesundheitssektor Aufmerksamkeit. Viele Firmen erhoffen sich Kosten- und Zeitersparnisse von der Nutzung der Blockchain. Versucht man jedoch herauszufinden, was die Blockchain eigentlich kann, findet man oft nur Erklärungen zur Technologie selbst oder Informationen über ihre bekannteste Anwendung, die Kryptowährung. Die grundlegende Frage, welche Eigenschaften und damit Vor- aber auch Nachteile eine Blockchain besitzt, werden meist nicht behandelt. Vor allem die negativen Eigenschaften werden oft außer Acht gelassen. Zwar liest man manchmal vom Problem der Skalierbarkeit, jedoch gibt es noch andere wichtige Dinge zu beachten.
Um ein Verständnis für die Eigenschaften und passenden Anwendungsgebiete einer Blockchain zu erlangen, wird in diesem Paper in II die Funktionsweise der Blockchain erklärt. Zudem wird auf weitere Technologien und Mechanismen eingegangen, die oft in Zusammenhang mit der Blockchain genutzt werden. Basierend auf diesem Wissen wird in III auf die Eigenschaften der Blockchain eingegangen. Diese sind in Vor- und Nachteile untergliedert, wobei auch auf mögliche Angriffe auf die Blockchain eingegangen wird. In IV wird beschrieben, welche Voraussetzungen ein System zur erfolgreichen Anwendung einer Blockchain besitzen sollte. Dies wird zusätzlich durch zwei passende Anwendungsbeispiele aufgezeigt. Eine Zusammenfassung der Ergebnisse sowie eine Bewertung dieser befindet sich in V.

II. Technologie

Um ein besseres Verständnis für die Eigenschaften und Anwendungsmöglichkeiten einer Blockchain zu erhalten, wird in diesem Kapitel die grundlegende Funktionsweise einer Blockchain erklärt.

A. Hashes und Blöcke

Eine Blockchain besteht, wie ihr Name bereits sagt, aus miteinander verketteten Blöcken, welche Informationen über gewisse Transaktionen beinhalten. Die einzelnen Blöcke sind kryptographisch mit Hash-Funktionen verschlüsselt. Diese codieren Informationen bzw. Zeichenfolgen in zufällige, aber

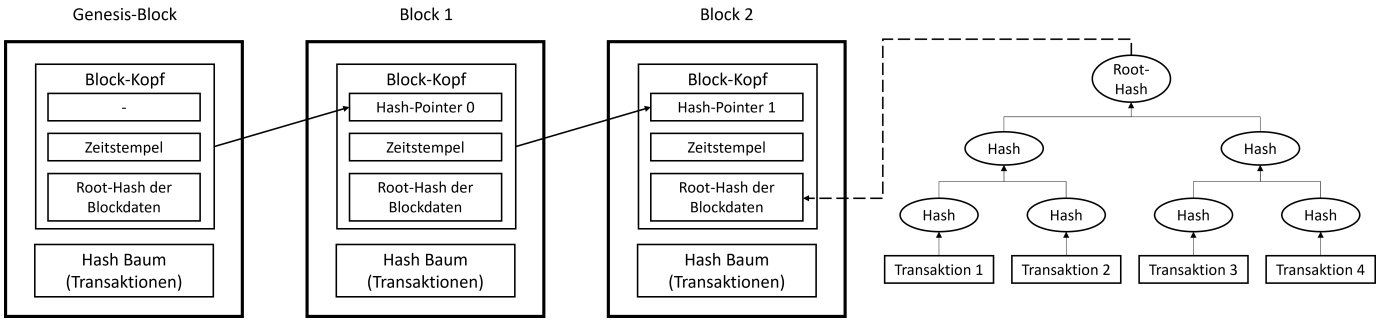

Abbildung 1. Verkettung einer Blockchain und Aufbau eines Hash-Baums nach [6] und [3]

berechenbare Buchstaben-Zahlen-Kombinationen mit festgelegter Länge. [4], [5] Dabei gibt es drei wichtige Sicherheitsanforderungen an die Hash-Funktion:

1) Anti-Reversibilität: Der Hash-Wert muss auf Basis einer Einwegfunktion berechnet werden, d. h. der Hash-Wert darf keinen Rückschluss auf die ursprüngliche Information liefern [4], [6].
2) Eindeutigkeit: Eine bestimmte Reihenfolge an Zeichen muss immer zum selben Hash-Wert führen [6].
3) Kollisionsresistenz: Eine ähnliche Reihenfolge an Zeichen soll zu stark unterschiedlichen Hash-Werten führen, sodass eine Erstellung eines bestimmten Hash-Werts und somit eine gezielte Kollision zweier Hash-Werte unmöglich ist [6].

Eine Kollisionsresistenz kann zwar nie hundertprozentig garantiert werden, doch liegt die Wahrscheinlichkeit für eine Kollision bei den momentan genutzten Hashes wie dem SHA-256 (Secure Hash Algorithm mit 256 Bit im Eingang) bei ungefähr 1 zu 2^{128} [6]. Dieser gilt damit als kollisionsresistent [4]. Eine ernsthafte Gefährdung für die Sicherheit der Hash-Funktion könnte in Zukunft ein Quantenalgorithmus wie der Grover's Algorithmus darstellen [7]. Für einen sicheren Hash ist es außerdem wichtig, dass die Einwegfunktion kleine Veränderungen des Originaltextes zu einer großen Änderung des Hashes codiert [4].
Betrachtet man in Abb. 1 einen einzelnen Block, sieht man, dass dieser aus den Blockdaten und dem Blockkopf besteht. Die Blockdaten beinhalten eine Liste an Transaktionen, die in einem bestimmten Zeitbereich getätigt wurden. Die einzelnen Transaktionen werden dabei mit Hash-Werten so lange miteinander kryptographisch verkettet bis nur noch zwei Hash-Werte übrig bleiben. Im Block-Kopf werden diese Hashes nochmals zu einem einzigen Hash zusammengeführt, den Root-Hash oder Merkle Root. [8] Die gesamte Verzweigung an Transaktionen sieht man rechts in Abb. 1 und nennt man Merkle-Baum [5]. Des Weiteren befinden sich im Block-Kopf Informationen wie der Zeitstempel und der Hash-Wert des vorherigen Block-Kopfes [6]. Mit diesem Hash-Pointer wird der vorherige Block mit dem neuen Block verkettet [6], [5]. Wird also ein neuer Block zu der Blockchain hinzugefügt, werden die Einträge im Kopf des vorherigen Blocks zu einem einzigen Hash-Pointer zusammengeführt, welcher dann in dem neuen Block-Kopf abgelegt wird [5]. Der erste Block einer Blockchain wird auch Genesis-Block genannt und ist in der Software in der Regel hart codiert [9]. Für eine Fälschung von Daten müssten alle vorherigen Hash-Pointer geändert werden, wobei es unmöglich ist den Genesis-Block unbemerkt zu fälschen [5].

B. Digitale Signatur

Will ein Nutzer dem Blockchain-Netz eine mögliche Transaktion hinzufügen, durchläuft diese einen Verifizierungsprozess [10]. Dabei werden folgende Informationen über die Transaktion an das Blockchain-Netzwerk weitergeleitet (Broadcasting) [6], [10], [5]: Die öffentliche Adresse des Transaktionsempfängers, die Transaktionsnachricht selbst, einen öffentlichen Schlüssel des Senders sowie die digitalen Signatur des Senders [10]. Für das Verfahren einer solchen digitalen Signatur werden ein privater und der bereits genannte öffentliche Schlüssel erzeugt [5]. Letzterer wird durch eine mathematische Formel aus dem Ersteren erstellt. Ähnlich wie bei der Hash-Funktion ist es sehr einfach den öffentlichen Schlüssel aus dem Privaten zu erstellen, die Umkehrung ist jedoch praktisch unmöglich. Zur Erstellung einer digitalen Signatur wird die Nachricht gehasht und daraufhin mit dem privaten Schlüssel signiert bzw. verschlüsselt. Mit Hilfe des öffentlichen Schlüssels kann die digitale Signatur entschlüsselt werden, sodass man wieder den Hash der Nachricht erhält. Gleichzeitig wird der Hash-Wert der vom Sender gesendeten Nachricht erstellt [8]. Stimmen die beiden Hashes überein, ist sichergestellt, dass die Transaktion von dem genannten Sender kommt und die Nachricht nicht verändert wurde [4]. Somit ist die digitale Signatur ein Mechanismus zur Überprüfung der Integrität und Authentizität von Transaktionen, während diese aber gleichzeitig öffentlich bleiben können [6].

C. Konsensmechanismus

Bevor ein neuer Block schlussendlich zur Blockchain hinzugefügt wird, wird er an alle Teilnehmer des Netzwerks gesendet. Jeder einzelne Teilnehmer überprüft daraufhin die Gültigkeit des Blocks. [6] Um die Daten auch bei sehr vielen Teilnehmern auf einem einheitlichen Stand zu halten, wird ein Konsensmechanismus benutzt [8]. Dieser muss fehlertolerant gegenüber Manipulationsversuchen von einem oder sogar mehreren bösartige Teilnehmern sein [5], [8]. Wichtig dabei

sind die Eigenschaften Persistenz und Liveness [5]. Persistenz besagt, dass alle Teilnehmer eine konsistente Reaktion auf einen Status einer Transaktion haben [5]. Um Liveness garantieren zu können, müssen sich alle Teilnehmer innerhalb einer gewissen Zeitspanne auf ein Ereignis einigen können [5]. Die Kombination dieser beiden Eigenschaften führt zu einem robusten Konsensmechanismus [5]. Je nach Anwendungsfall werden unterschiedliche Konsensmechanismen genutzt, welche bestimmte Vor- und Nachteile besitzen [11], [6]. Beispiele sind der Proof of Work (PoW), Proof of Stake (PoS) und der praktische byzantinische Fehlertoleranz (PBFT) -Algorithmus [12].

D. Distributed Ledger Technologie

Eine weitere Technologie, die oft in Zusammenhang mit der Blockchain verwendet wird, ist die Distributed Ledger Technologie (DLT). Ein Ledger (zu deutsch: Hauptbuch) ist eine Sammlung von Transaktionen, die beispielsweise den Austausch von Waren oder Dienstleistungen festhält. [6] Bei der Distributed Ledger Technologie besitzen alle Teilnehmer eines Netzwerks eine Kopie des Hauptbuches [8]. Da eine Blockchain dezentral verwaltet ist, d. h. alle Teilnehmer eines Netzwerks Zugriff auf die Blockchain besitzen, bietet sich eine verteilte Speicherung an. Ein Netzwerk mit diesen Eigenschaften wird auch Peer-to-Peer (P2P) Netzwerk genannt. [6] Der Vorteil einer solchen Struktur ist, dass es keine einzelne Partei gibt, der vertraut werden muss und die genügend Macht besitzt, Einträge in der Blockchain eigenständig zu ändern. Zudem besitzt das System keinen zentralen Punkt, von welchem aus sich ein Angriff auf das ganze System auswirken kann. [12]

III. Eigenschaften einer Blockchain

Um aufzuzeigen, welche Eigenschaften die Blockchain-Technologie mit sich bringt, werden diese basierend auf der in II beschriebenen Funktionsweise erläutert.

A. Vorteile

Im Folgenden werden die positiven Eigenschaften der Blockchain dargestellt.

1) Unveränderlichkeit und Dauerhaftigkeit: Durch die kryptographische Aneinanderkettung der Blöcke mit Hilfe von Hash-Funktionen, die in II-A beschrieben werden, ist es praktisch unmöglich, die Blockchain im Nachhinein zu verändern oder zu löschen. Hier spielt auch die Dezentralisierung eine wichtige Rolle, da jede von der Blockchain aufgenommene Transaktion auf jeden Computer im Blockchain-Netzwerk kopiert vorliegt. Somit existiert eine Blockchain auch so lange wie das P2P-Netzwerk selbst. Erst wenn es keine Teilnehmer (in der Praxis wäre auch bei sehr wenigen Teilnehmern bereits die Sicherheit der Blockchain gefährdet) mehr gibt, sind die darin gespeicherten Daten verloren. [13], [15]

2) dezentrales System und verteilte Datenkontrolle: Wie bereits in II-C und II-D beschrieben hat eine Blockchain die Fähigkeit einen dezentralen Konsens über die Gültigkeit einer Transaktion, aber auch der gesamten bestehenden Kette zu erzeugen. Dabei ist die Blockchain nicht auf eine zentrale vertrauenswürdige Partei angewiesen. Durch die meist zusätzliche verteilte Speicherung der Blockchain ist sie zudem robust gegenüber dem Ausfall von einzelnen Teilnehmern, besitzt also keinen Single-Point-of-Failure. Eine systemübergreifende Manipulation der Daten wird dadurch zusätzlich erschwert. [13], [6], [14]

3) Authentizität und Nichtabstreitbarkeit: Jeder Teilnehmer, der einem Blockchain-Netzwerk beitritt, erhält eine einzigartige Identität. Mit dieser Identität werden die Transaktionen, die er tätigt, digital signiert, wodurch einer Transaktion jederzeit ihr Ursprung zugeordnet werden kann. Durch die zusätzliche Unveränderlichkeit der Transaktionen ist eine Nichtabstreitbarkeit gegeben. [6], [16]

4) Transparenz: Da der Zustand der Kette sowie jede einzelne Interaktion zwischen den beteiligten Parteien von jedem befugten Teilnehmer überprüft werden kann, ist die Transparenz der Transaktionen gegeben. [13], [16]

5) Vertrauenswürdigkeit: Die Vertrauenswürdigkeit einer Blockchain wächst mit der Anzahl an getätigten Transaktionen und Teilnehmern [15]. Ein gutes Beispiel hierfür ist die Kryptowährung Bitcoin, die durch die steigende Teilnehmerzahl an Vertrauenswürdigkeit und somit auch an Wert gewonnen hat.

B. Nachteile

Neben den Vorteilen einer Blockchain wird sie in diesem Unterkapitel auch kritisch bezüglich ihrer Nachteile beleuchtet.

1) hoher Energieverbrauch: Ein großer Nachteil der Blockchain ist der hohe Energieverbrauch. Dieser wird durch mehrere Vorgänge in der Blockchain hervorgerufen. Zum einen wird viel Energie für das Broadcasten einer neuen Transaktion benötigt. Zum anderen erfordern die Konsensmechanismen, darunter vor allem der in II-C genannte PoW-Algorithmus, sehr viel Rechenleistung. [17] Zudem wird auch bei dem Berechnungsprozess der kryptographischen Signatur Rechenaufwand und somit Energie benötigt [16], [18].

2) Forks: Soll das Regelwerk einer Blockchain geändert werden, geschieht dies durch einen Soft Fork oder Hard Fork. Bei einem Soft Fork sind die Änderungen rückwärtskompatibel und die bisherige Kette bleibt bestehen. Dies könnte dazu führen, dass alte Knoten, die noch nicht synchronisiert sind, Transaktionen akzeptieren, die für die neuen Knoten ungültig erscheinen. Ein Hard Fork ist im Gegensatz dazu nicht rückwärtskompatibel. Hierbei spaltet sich die Blockchain in zwei Teile, die ursprüngliche Blockchain und eine neue Version mit dem neuen Regelwerk. [16] [19]

3) mangelnde Dezentralisierung bei zu vielen Teilnehmern und Skalierbareitsprobelm: Je mehr Teilnehmer zu einem Blockchain-Netzwerk gehören, desto mehr Transaktionen werden in der Regel getätigt. Besitzt eine Blockchain sehr viele Blöcke, wird auch der Rechenaufwand einen neuen Block zu verifizieren höher. Dies führt zu einer mangelnden Skalierbarkeit des Systems, da die Durchführung einer Transaktion verlangsamt wird. Die Skalierbarkeit sagt aus, wie viele Transaktionen pro Sekunde durchgeführt werden können. Ein

weiterer Nachteil kann sein, dass nicht mehr alle Teilnehmer des Netzwerks diesen Rechenaufwand bewältigen können oder genügend Speicherkapazität für die gesamte Blockchain besitzen. Dies sorgt für folgende Problemstellungen: Zum einen wird das Hauptbuch nicht mehr bei allen Teilnehmern gleich abgespeichert, zum anderen ist die Blockchain nun nicht mehr komplett dezentralisiert, da Teilnehmer mit genügend Rechen- und Speicherkapazität mehr auf die Blockchain einwirken können. Die Eigenschaften Unveränderlichkeit und Transparenz werden dadurch geschwächt. [16], [18]

4) Unveränderlichkeit: Die Unveränderlichkeit der Blockchain wird in III-A als Vorteil genannt. Je nach Anwendungsfall kann diese Eigenschaft aber auch zu Schwierigkeiten führen. Sensible persönliche Daten müssen laut Datenschutzgrundverordnung (DGSVO) lösch- und korrigierbar sein. Es stellt sich also die Frage, ob die Anwendung einer Blockchain bei solch sensiblen Daten sinnvoll ist. [18]

C. Angriffe und Probleme

Wie bei jedem System sind auch Angriffe auf eine Blockchain möglich. Für einen besseren Überblick über das Thema werden mögliche Angriffe genannt [16], [20]:

- 51 %-Angriff oder Mehrheitsangriff: Ein Angreifer besitzt den Großteil der Hash-Leistung und kontrolliert somit den Konsensmechanismus.
- Distributed Denial of Service (DDoS): Eine Vielzahl an fehlerhaften Anfragen wird an das System gesendet, um es zu überfluten und damit die Verarbeitung von legitimen Transaktionen zu verhindern.
- Sybil Angriff: Eine Identität (Person) versucht mehrere Teilnehmer bzw. Knoten eines Netzwerks zu besitzen. So kann der Konsens eines Netzwerks negativ beeinflusst werden.

Je größer das Blockchain-Netzwerk ist, desto unwahrscheinlicher ist der Erfolg dieser Angriffe.

IV. Anwendungsfälle

Auch wenn die Blockchain zu dieser Zeit sehr beliebt ist und eine Lösung für viele Probleme zu sein scheint, ist sie keine Allzwecklösung. Es gibt Anwendungsfälle für die eine Blockchain sehr nützlich sein kann. Dabei sollte jedoch auf die eigentlichen Vorteile einer Blockchain gegenüber den bisherigen Lösungen eingegangen werden. Bisher noch nicht genannt wurden die verschiedenen Arten von Blockchains: Es gibt öffentliche, private und hybride Formen. Diese können genehmigungsfrei oder genehmigungsbehaftet sein. [21] Aufgrund der Längenbeschränkung des Papers kann auf diese Varianten nicht näher eingegangen werden. Auch die Auswahl des richtigen Konsensalgorithmus aus II-C stellt einen wichtigen Aspekt dar und soll hier genannt werden. Generell ist eine Blockchain immer dann nützlich, wenn es mehr als eine Verwaltungsbehörde gibt und ein Vertrauensverhältnis zwischen diesen besteht. Gibt es keine vertrauenswürdige dritte Partei, die die Daten zentral organisiert, kann eine Blockchain auch ohne Vertrauensverhältnis zwischen den Parteien sinnvoll sein. Ein weiterer wichtiger Aspekt ist die Frage, ob die Aufzeichnung der Daten bzw. Transaktionen unveränderlich sein sollte. Bei einer Blockchain ist dies der Fall. [13]

A. Lieferkettenmanagement

Ein sehr gutes Beispiel für die Anwendung der Blockchain ist ein Lieferkettenmanagementsystem. Bei diesem werden Material- und Informationsflüsse innerhalb und zwischen verschiedenen Parteien wie Lieferant, Vertrieb usw. verwaltet. Diese Parteien wollen zusammenarbeiten und haben ein Vertrauensverhältnis. Es ist zudem wichtig, dass alle Transaktionen eines Produktes bis zum Endverbraucher unveränderlich dokumentiert werden und von allen Teilnehmern des Blockchain-Netzwerkes einsehbar sind. [13]

B. Gesundheitsfürsorge

Ein weiteres Beispiel ist die Anwendung der Blockchain im Gesundheitssektor. Mit Hilfe von Blockchains können Patientendaten sowie Informationen über Medikamente rückverfolgt und geprüft werden. Bei Letzteren kann so eine Fälschung oder ungenaue Dosierung der Inhaltsstoffe verhindert werden. Bei Patientendaten ist der Datenschutz besonders wichtig, da es sich hier um sensible Daten handelt. Die Transparenz der Daten soll hierbei nur für bestimmte Personen, wie den Patienten selbst oder behandelnde Ärzte gegeben sein. Wie in III-B4 erklärt müssen hierfür passende Lösungen gefunden werden. [10]

V. Zusammenfassung

Die Blockchain ist eine krpytographische Aneinanderkettung von Informationsblöcken, welche wiederum mit einer digitalen Signatur versehen sind. Mit Hilfe eines Konsensmechanismus sowie der Distributed Ledger Technologie stellt sie eine neue Möglichkeit zur Speicherung von Transaktionen dar. Ob die Nutzung einer Blockchain sinnvoll ist, hängt von dem konkreten Anwendungsfall ab und sollte bezüglich der Vor- und Nachteile, die die Blockchain mit sich bringt, abgewogen werden. Gibt es mehrere Parteien, zwischen denen ein Vertrauensverhältnis besteht, und ist die unveränderliche Aufzeichnung von Transaktionen sowie deren Transparenz gewollt, so kann die Anwendung der dezentral verwalteten Blockchain zielführend sein. Ein großer Nachteil der Blockchain ist der hohe Energieverbrauch der Technologie. Des Weiteren sollten die Starrheit sowie die möglichen Einschränkungen bei sehr vielen Teilnehmern beachtet werden. Wie am Anfang des Papers erwähnt, existiert die Blockchain Technologie erst seit ungefähr 30 Jahren und steht damit noch in der Anfangsphase der Entwicklung. Dementsprechend muss die Technologie noch einige Hürden überwinden, bis sie zur breiten Anwendung einsetzbar ist. Durch die große Aufmerksamkeit, die die Blockchain durch den Erfolg der Krpytowährung Bitcoin erhalten hat, besitzt sie dafür die besten Voraussetzungen.

LITERATUR

[1] Binance Academy, “History of Blockchain,” [Online]. Available: https://academy.binance.com/en/articles/history-of-blockchain. [Aufgerufen am 30.04.2022].

[2] BTC Direct, “Die Ursprünge und die Geschichte von Bitcoin ,” [Online]. Available: https://btcdirect.eu/de-at/geschichte-von-bitcoin. [Aufgerufen am 30.04.2022].

[3] BitcoinWiki, “Merkle tree,” [Online]. Available: https://en.bitcoinwiki.org/wiki/Merkle_tree. [Aufgerufen am 05.06.2022].

[4] S. S. Sarmah, “Understanding Blockchain Technology,” Computer Science and Engineering, Vol. 8 No. 2, 2018, DOI 10.5923/j.computer.20180802.02. [Online]. Available: https://www.researchgate.net/profile/S-Sarmah/publication/336130918_Understanding_Blockchain_Technology/links/5d913eb9a6fdcc2554a69c7c/Understanding-Blockchain-Technology.pdf, [Aufgerufen am 25.04.2022].

[5] R. Zhang, R. Xue, L. Liu, “Security and Privacy on Blockchain,” ACM Comput. Surv. 52, 3, Article 51, 2019, DOI 10.1145/3316481.

[6] D. Yaga et al. , “Blockchain Technology Overview,” National Institute of Standards and Technology, 2018.

[7] A., “QUANTENCOMPUTER - EINE BEDROHUNG FÜR PKI?,” [Online]. Available: https://www.securosys.com/de/ueberuns/stories/quantencomputer-eine-bedrohung-f\%C3\%BCr-pki. [Aufgerufen am 17.05.2022].

[8] J. Hosp, “Blockchain 2.0,” 2nd ed., Finanz Buch Verlag, 2018.

[9] Alexandria, “Genesis Block,” [Online]. Available: https://coinmarketcap.com/alexandria/glossary/genesis-block. [Aufgerufen am 17.05.2022].

[10] A. A. Monrat, O. Schelen, K. Anderson, “A Survey of Blockchain From the Perspectives of Applications, Challenges, and Opportunities,” Department of Computer Science, Electrical and Space Engineering, Lulea University of Technology, 2019.

[11] Der Kontext, “Blockchain,” [Online]. Available: https://map.derkontext.com/blockchain#m=12/1409.30488/444.31826. [Aufgerufen am 27.04.2022].

[12] B. Singhal, G. Dhameja, P. S. Panda, “A Beginner’s Guide to Building Blockchain Solutions,” Apress Media, LLC, 2018, S. 1- 146, DOI 10.1007/978-1-4842-3444-0.

[13] M. J. M. Chowdhury et al., “Blockchain versus Databank: A critical Analysis,” in 17th IEEE International Conference On Trust, Security And Privacy In Computing And Communications/ 12th IEEE International Conference On Big Data Science And Engineering, 2018, DOI 10.1109/TrustCom/BigDataSE.2018.00186.

[14] A. Bahga, V. Madisetti, “Blockchain Platform for Industrial Internet of Things,” Journal of Software Engineering and Applications, Nr. 9, S. 533-546, 2016.

[15] A. Songara, L. Chouhan, “Blockchain: A Decentralized Technique for Securing Internet of Things,” Department of Computer Science and Engineering, 2017.

[16] J. Golosova, A. Romanovs, “The Advantages and Disadvantages of the Blockchain” Dept. of Modelling and Simulation, Riga Technical University, Riga, Latvia, 2018.

[17] Myra, “Was ist die Blockchain?,” [Online]. Available: https://www.myrasecurity.com/de/blockchain/. [Aufgerufen am 17.05.2022].

[18] J. Hinckeldeyn,, “Blockchain-Technologie in der Supply Chain: Einführung und Anwendungsbeispiele,” Wiesbaden, Springer Vieweg, S. 45-47, 2019.

[19] Coinbase, “Was ist ein Fork?,” [Online] Available: https://www.coinbase.com/de/learn/crypto-basics/what-is-a-fork. [Aufgerufen am 18.05.2022].

[20] J. Risberg, “Über die Angreifbarkeit von Blockchains,” 2018. [Online] Available: https://www.infopoint-security.de/ueber-die-angreifbarkeit-von-blockchains/a16410/. [Aufgerufen am 18.05.2022].

[21] K. E. Wegrzyn, E. Wang, “Types of Blockchain: Public, Private, or Something in Between,” 2021. [Online] Available: https://www.foley.com/en/insights/publications/2021/08/types-of-blockchain-public-private-between. [Aufgerufen am 20.05.2022].

GaN Diodes for Power Electronics Applications

Florian Lausser
Ostbayerische Technische Hochschule
Elektro- und Informationstechnik
Regensburg, Deutschland

***Abstract*—With the development of increasingly efficient power electronics, Si-based diodes are reaching their theoretical limits. Wide band-gap semiconductors such as gallium nitride (GaN) have a sufficient potential to meet the requirements of high voltage and current resistance, low switching times and low losses. Despite that, GaN diodes are still rarely found on the market. This work gives an overview of the current research of GaN diodes for power electronic applications, as well as the physical properties of the materials used. The focus of the current research is the development on GaN Schottky diodes, since they are characterized by fast switching behavior, a high dielectric strength and a low reverse recovery charge. The properties depend on the structure of the assembly. The potential difference of the Schottky barrier is determined by the metal used for the Schottky contact. This potential difference has a direct influence on various properties of vertical diodes, such as forward voltage and reverse current. A semiconductor heterostructure and the resulting 2-dim. electron gas (2DEG) or hole gas (2DHG) can significantly improve the conductivity and reverse voltage of lateral diodes. The resulting diodes are characterized by a very high electron density in the conducting channel. In the following, the operation, structure and electrical properties of vertical and lateral structures are analyzed and compared against each other. Finally, the findings are compared to the silicon carbide diodes commonly available on the market.**

***Index Terms*—GaN, diode, Schottky diode, Schottky barrier diode, SiC**

I. Introduction

The current development of power electronic components pursues two basic goals. On the one hand, the efficiency is to be increased further and further. On the other hand, the size should be reduced significantly. These goals can no longer be met by standard Si diodes. Due to the small bandgap of 1.1 eV technical limits arise. In order to block higher voltages, the dimensions have to be increased. However, this again results in increased ohmic losses. In addition, the switching speeds decrease due to the high stored charge, which again has a negative effect on the losses. For this reason, current research is focused on diodes made of SiC and GaN. [1]
GaN belongs to the group of wide band gap semiconductors. These are characterized by good thermal conductivity, a high band gap and a high breakdown Field (cf.I). The high bandgap leads, for example, to a lower intrinsic charge carrier density than silicon. As a result, they can be operated up to temperatures of 573 K.[2] Schottky barrier diodes (SBD) has the properties for high frequency and power electronic applications [3]. This work shows the current development status of GaN Schottky diodes for power electronics applications. Lateral and vertical structures are considered and the teperature and switching properties are investigated.

TABLE I
Properties of different semiconductors [4]

	Bandgap (eV)	Breakdown Field ($\frac{MV}{cm}$)	Thermal Conductivity ($\frac{W}{cm°C}$)
Si	1.1	0.3	1.3
SiC	3.3	3.0	4.9
GaN	3.4	2.5	1.5

II. GaN-contacts

The characteristics of each diode is determined by existing transitions. The determination of the characteristic is usually created by the I-V method, the C-V method or the C-V-T method. [5][6] The detailed analysis of all three measurement methods are presented in [5].

A. Schottky contact

A Schottky contact consists of a metal-semiconductor junction. The different work function of the two materials leads to a band bending. The potential difference that occurs is called the Schottky barrier. The Schottky barrier can be determined by all three of the previously mentioned methods. Normally, the value obtained by the C-V measurement is larger than that obtained by the I-V measurement. This deviation is due to inhomogeneities in the crystal structure. [6] The influence of the Schottky barrier and the basis of all measurement techniques can be illustrated by the following two equations. The current through the diode can be described by the following equation.

$$I_D = I_S \cdot (e^{\frac{q(V-R_s I)}{nkT}} - 1) \tag{1}$$

The reverse current can be specified as a function of the Schottky barrier Φ_{BN}.

$$I_s = SA^*T^2 e^{-\frac{q\Phi_{BN}}{kT}} \tag{2}$$

V describes the applied voltage, k the Bolzmann constant,R_S the series resistance ,A^* the Richardson's constant,S the area, T the absolute temperature.[7] Table II lists the measurement results of various metals. For all the materials listed, the was obtained a ideality factor n of 1.0X [8][9][10].

TABLE II
SCHOTTKEY BARRIER HEIGHT OF DIFFERENT METALS

Metal	$\Phi(V)$(I-V)	$\Phi(V)$(C-V)
Au[8]	0.844	0.94
Ni[9]	0.95	1.13
Pd[10]	1.11	1.24
Pt[10]	1.13	1.27

B. Heterostructure

Heterostructures are used to modify the conductivity and reverse voltage of the diode. A distinction is made between a single and a double heterostructure. In Fig.:1 the band diagram with the dimensions of the individual layers is shown. The influence of the dimensions (L_A, L_{SG}, L_{DG}) on the carrier concentration and the mobility is studied in detail in [11][12]. In the following, the basic aspects are summarized.

1) Single Heterostructure AlGaN-GaN: At the transition region, the bands are bent down so far that they lie below the Fermi level. The electron density depends on L_A and reaches about $1.4 \cdot 10^{13} \frac{1}{cm^2}$ at $L_A = 25\,nm$. The mobility is about $1200 \frac{cm^2}{Vs}$.

2) Double Heterostructure GaN-AlGaN-GaN: This structure results from the simple heterostructure and another GaN layer. The 2DEG concentration decreases with increasing L_{DB}. This can be counteracted by increasing L_A. The GaN-AlGaN transition causes a complementary upward band bending. This bending is again dependent on L_{GD}. When L_{GD} is large enough, the valence band is raised above the Fermi level. The result is a two-dimensional hole gas (2DHG). For very small $L_{DG} < 5$ nm, the behavior of the structure corresponds to a simple heterostructure. The utility of this will be considered in more detail later.

III. GAN DIODE STRUCTURES

Basically, two structures are distinguished, the vertical and the lateral structure. In SiC-based SBD, vertical structures are preferred to vertical ones up to a medium voltage range. However, this statement can no longer be confirmed unambiguously for GaN. As can be seen in Fig.:4, the theoretical limit of the lateral structure corresponds approximately to that of the vertical structure. [13]

A. Vertical structure

The design of the vertical structure is shown in Fig. 2. Two factors are decisive for the properties of the diode. First, the mobility is strongly influenced by the quality of the crystal lattice and second, the quality of the Schottky contact determines the properties. The dielectric strength is largely determined by the thickness of the GaN layer. Due to the contacting, the threshold voltage is approximately constant at 1 V.[14][15]

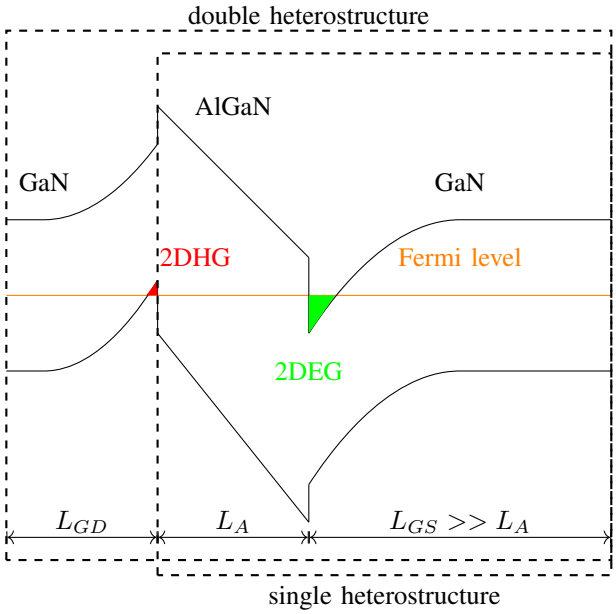

Fig. 1. Band diagram of a single and double heterostruct

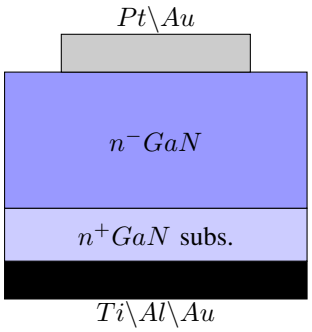

Fig. 2. Schematic Illustration of vertical SBD [15]

B. Lateral structure

1) lateral structure with a single heterostructure: This structure is based on a single heterostructure. The schematic structure is shown in Fig.:3, if the layer thickness L_{DG} is allowed to approach zero. In the so-called recess-shotkyanode, the 2DEG is directly contacted laterally. The potential difference is significantly lower than when contacting the AlGaN. This measure reduced the forward voltage of a diode in the experiment from 1.2 V to 0.5 V.[16] In order to improve the blocking behavior, so-called field planes (FP) are used. These provide a clearing of the diode channel and a reduction of the peak value of the electric field.[16] FPs can be used to nearly double the reverse bias voltage.[17] By these measures, diodes with an insertion voltage of 0.5 V and a reverse voltage of 1.62 kV can be realized. A TiN anode was used and a resistivity of about $5\,m\Omega cm^2$ was achieved. Another way to reduce the resistance is to use multi-channel structures. In this case, several AlGaN-GaN structures are stacked on top of each other. In this way, a resistivity of $3\,m\Omega cm^2$ can be achieved at

a maximum reverse voltage of 3.3 kV. [13] This multi-channel allows blocking voltages of 10 kV to be realized.[18]

2) lateral structure with a double heterostrukture: The schematic structure is shown in Fig.:3. The distance between the drift region and the cathode interrupts the hole current. During the conducting phase of the diode, the current flows across the 2DEG. The diode behaves similarly to that with a single heterostructure. By applying a negative voltage, the structure is polarized. The drift region is depleted and behaves like an intrinsic conductor. The result is an improvement in reverse bias. This structure is called the polarization superjunction concept (PSJ). [19] With this setup, the reverse voltage of a normal SBD could be increased from 586 V to 954 V.[20]

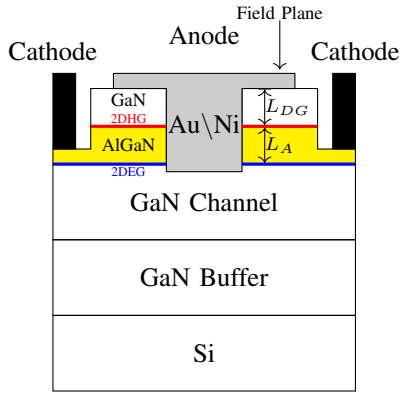

Fig. 3. Schematic Illustration of lateral SBD with a double heterostructure[20]

IV. Characteristik

In Fig.:4 the resistivity and the corresponding reverse voltage are plotted.Vertical diodes are mainly represented up to 1 kV. The lateral structures presented can achieve very good results in the high voltage range.

Reverse Recovery Time and blocking delay charges and were investigated in [15] for a vertical diode. Thereby a reverse recovery time of $t_{rr} = 17\,ns$ and a reverse recovery charge of $Q_{rr} = 0.8\,nC$ was measured. The reverse recovery behavior of vertical diodes is comparable to that of SiC diodes[25]. For lateral diodes, both values ($t_{rr} = 31\,ns$, $Q_{rr} = 30\,nC$) are significantly higher. This fact can be attributed to parasitic capacitances. Fig.5 the C-V characteristics of an SBD and a PSJ diode are shown. When the cathode-anode voltage (V_{CA}) is increased, the 2DEG under anode is cleared out first. This corresponds to the first plateau. PSJ diodes show another one in addition. The further parasitic parallel capacitance is formed by the 2DGH and the 2DEG.[26]

Compared to SiC diodes, lateral GaN diodes show a better temperature behavior in forward direction (Fig.:6). The reverse current shows very little dependence for lateral diodes. This indicates that the quality of the crystal lattice has only a very small influence.[27]. For the vertical structures, the quality of the crystal lattice has a high influence. A strong temperature dependence of the blocking voltage is shown. A reduction from 560 V (θ=300 K) to 200 V (θ=450 K) could be observed.

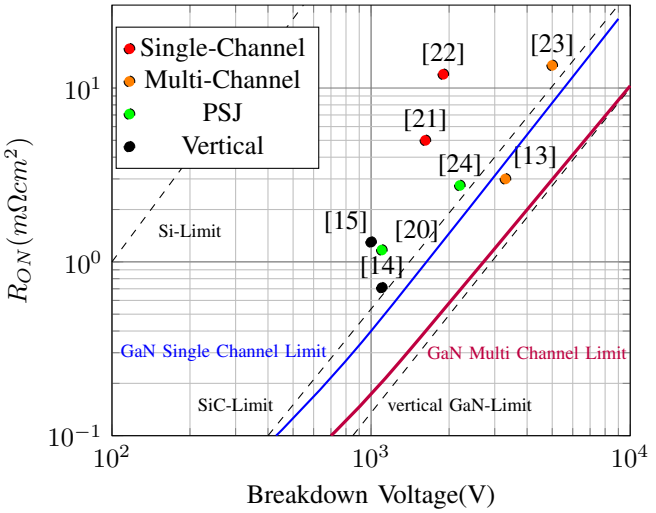

Fig. 4. Compilation of reverse voltage and resistivity of different diodes [13]

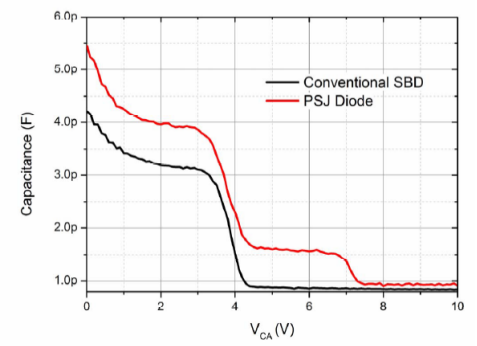

Fig. 5. C-V-Characteristic of a lateral SBD and a PSJ [26]

Strong temperature dependencies are also seen in the forward direction. [28]

V. Conclusion

Vertical and lateral structures show a great potential to serve the requirements of modern power electronics. In the lower voltage range, vertical diodes show their advantages, whereas lateral diodes can realize very high reverse voltages at low losses. Vertical diodes are characterized by very good switching behavior. However, they have a worse temperature behavior than lateral diodes. Lateral diodes offer a high reverse voltage with a good temperature behavior. However, they have a worse reverse recovery behavior. The switching behavior can generally be compared with SiC diodes. The temperature behavior of lateral GaN diodes is better than that of SiC.

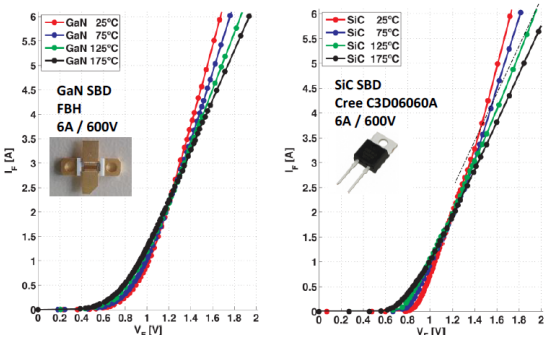

Fig. 6. I-V-T Characteristic[16]

References

[1] Isik C. Kizilyallia et al. *Current and Future Directions in Power Electronic Devices and Circuits based on Wide Band-Gap Semiconductors*. U.S. Department ofEnergy, 2017.

[2] P.G. Neudeck, R.S. Okojie, and Liang- Yu Chen. "High-temperature electronics - a role for wide bandgap semiconductors?" In: *Proceedings of the IEEE* 90.6 (2002), pp. 1065–1076. DOI: 10.1109/JPROC.2002.1021571.

[3] Yang Li et al. "GaN Schottky Barrier Diode-Based Wideband and Medium-Power Microwave Rectifier for Wireless Power Transmission". In: *IEEE Transactions on Electron Devices* 67.10 (2020), pp. 4123–4129. DOI: 10.1109/TED.2020.3016619.

[4] Amin Ghazanfari, Christian Perreault, and Karim Zaghib. "EV/HEV Industry Trends of Wide-bandgap Power Semiconductor Devices for Power Electronics Converters". In: *2019 IEEE 28th International Symposium on Industrial Electronics (ISIE)*. 2019, pp. 1917–1923. DOI: 10.1109/ISIE.2019.8781528.

[5] Nadia Benseddik et al. "Electrical characterisation of Schottky diodes based on SiC with different contact surfaces". In: *International Journal of Materials Engineering Innovation* 5 (Jan. 2014), p. 285. DOI: 10.1504/IJMATEI.2014.066852.

[6] M. Siad et al. "Correlation between series resistance and parameters of Al/N-Si and Al/p-Si Schottky barrier diodes". In: *Applied Surface Science* 236 (Sept. 2004), pp. 366–376. DOI: 10.1016/j.apsusc.2004.05.009.

[7] D. Reddy et al. "Schottky Barrier Parameters of Pd/Ti Contacts on N-Type InP Revealed from IVT And CVT Measurements". In: *Journal of Modern Physics* 2 (Jan. 2011). DOI: 10.4236/jmp.2011.23018.

[8] P. Hacke et al. "Schottky barrier on n-type GaN grown by hydride vapor phase epitaxy". In: *Applied Physics Letters* 63.19 (1993), pp. 2676–2678. DOI: 10.1063/1.110417.

[9] A C Schmitz et al. "Schottky barrier properties of various metals on n-type GaN". In: *Semiconductor Science and Technology* 11.10 (Oct. 1996), pp. 1464–1467. DOI: 10.1088/0268-1242/11/10/002. URL: https://doi.org/10.1088/0268-1242/11/10/002.

[10] Lei Wang et al. "High barrier height GaN Schottky diodes: Pt/GaN and Pd/GaN". In: *Applied Physics Letters* 68.9 (1996), pp. 1267–1269. DOI: 10.1063/1.115948.

[11] Sten Heikman et al. "Polarization effects in AlGaN/GaN and GaN/AlGaN/GaN heterostructures". In: *Journal of Applied Physics* 93.12 (2003), pp. 10114–10118. DOI: 10.1063/1.1577222. URL: https://doi.org/10.1063/1.1577222.

[12] Akira Nakajima et al. "High Density Two-Dimensional Hole Gas Induced by Negative Polarization at GaN/AlGaN Heterointerface". In: *Applied Physics Express* 3.12 (Dec. 2010), p. 121004. DOI: 10.1143/apex.3.121004. URL: https://doi.org/10.1143/apex.3.121004.

[13] Ming Xiao et al. "3.3 kV Multi-Channel AlGaN/GaN Schottky Barrier Diodes With P-GaN Termination". In: *IEEE Electron Device Letters* 41.8 (2020), pp. 1177–1180. DOI: 10.1109/LED.2020.3005934.

[14] Yu Saitoh et al. "Extremely Low On-Resistance and High Breakdown Voltage Observed in Vertical GaN Schottky Barrier Diodes with High-Mobility Drift Layers on Low-Dislocation-Density GaN Substrates". In: *Applied Physics Express* 3.8 (July 2010), p. 081001. DOI: 10.1143/apex.3.081001. URL: https://doi.org/10.1143/apex.3.081001.

[15] Shu Yang et al. "1 kV/1.3 m$\Omega{\cdot}cm^2$ vertical GaN-on-GaN Schottky barrier diodes with high switching performance". In: *2018 IEEE 30th International Symposium on Power Semiconductor Devices and ICs (ISPSD)*. 2018, pp. 272–275. DOI: 10.1109/ISPSD.2018.8393655.

[16] Oliver Hilt. *GaN Dioden und selbstsperrende GaN Schalttransistoren für effiziente Leistungswandler (GaN Powerswitch) : Verbundprojekt Leistungswandler in GaN-Technologie zur Erschließung ungenutzter Energiepotentiale (PowerGaNPlus) ; im BMBF Verbundvorhaben Leistungselektronik zur Energieeffizienz-Steigerung (LES); Laufzeit des Vorhabens: 1.06.2010 bis 31.05.2013*. Hannover : Technische Informationsbibliothek (TIB), 2014. URL: https://oa.tib.eu/renate/handle/123456789/1566.

[17] Yong Lei et al. "Field plate engineering for GaN-based Schottky barrier diodes". In: *Journal of Semiconductors* 34.5 (May 2013), p. 054007. DOI: 10.1088/1674-4926/34/5/054007. URL: https://doi.org/10.1088/1674-4926/34/5/054007.

[18] Ming Xiao et al. "10 kV, 39 $m\Omega cm^2$ Multi-Channel AlGaN/GaN Schottky Barrier Diodes". In: *IEEE Electron Device Letters* 42.6 (2021), pp. 808–811. DOI: 10.1109/LED.2021.3076802.

[19] Akira Nakajima et al. "GaN based Super HFETs over 700V using the polarization junction concept". In: *2011 IEEE 23rd International Symposium on Power Semiconductor Devices and ICs*. 2011, pp. 280–283. DOI: 10.1109/ISPSD.2011.5890845.

[20] Tao Sun et al. “Theoretical and Experimental Study on AlGaN/GaN Schottky Barrier Diode on Si Substrate with Double-Heterojunction”. In: *Nanoscale Research Letters* 15 (Dec. 2020). DOI: 10.1186/s11671-020-03376-z.

[21] Ting-Ting Wang et al. “Recessed AlGaN/GaN Schottky Barrier Diodes With TiN and NiN Dual Anodes”. In: *IEEE Transactions on Electron Devices* 68.6 (2021), pp. 2867–2871. DOI: 10.1109/TED.2021.3071296.

[22] Mingda Zhu et al. “1.9-kV AlGaN/GaN Lateral Schottky Barrier Diodes on Silicon”. In: *IEEE Electron Device Letters* 36.4 (2015), pp. 375–377. DOI: 10.1109/LED.2015.2404309.

[23] M. Xiao et al. “5 kV Multi-Channel AlGaN/GaN Power Schottky Barrier Diodes with Junction-Fin-Anode”. In: *2020 IEEE International Electron Devices Meeting (IEDM)*. 2020, pp. 5.4.1–5.4.4. DOI: 10.1109/IEDM13553.2020.9372025.

[24] Fengbo Liao. “2.2 kV Breakdown Voltage AlGaN/GaN Schottky Barrier Diode with Polarization Doping Modulated 3D Hole Gas Cap Layer and Polarization Junction Structure”. In: *Journal of Electronic Materials* (2020). DOI: 10.1007/s11664-022-09605-8.

[25] Loizos Efthymiou et al. “Zero reverse recovery in SiC and GaN Schottky diodes: A comparison”. In: *2016 28th International Symposium on Power Semiconductor Devices and ICs (ISPSD)*. 2016, pp. 71–74. DOI: 10.1109/ISPSD.2016.7520780.

[26] Vineet Unni et al. “2.4kV GaN Polarization Superjunction Schottky Barrier Diodes on semi-insulating 6H-SiC substrate”. In: *2014 IEEE 26th International Symposium on Power Semiconductor Devices IC’s (ISPSD)*. 2014, pp. 245–248. DOI: 10.1109/ISPSD.2014.6856022.

[27] Wantae Lim et al. “Temperature dependence of current-voltage characteristics of Ni–AlGaN/GaN Schottky diodes”. In: *Applied Physics Letters* 97.24 (2010), p. 242103. DOI: 10.1063/1.3525931. URL: https://doi.org/10.1063/1.3525931.

[28] Yi Zhou et al. “Temperature-dependent electrical characteristics of bulk GaN Schottky rectifier”. In: *Journal of Applied Physics* 101.2 (2007), p. 024506. DOI: 10.1063/1.2425004. eprint: https://doi.org/10.1063/1.2425004. URL: https://doi.org/10.1063/1.2425004.

[29] Wang Yaqi. “Fabrication and Characterization of Gallium Nitride Based Diodes”. Auburn University, 2011.

[30] E. Bahat-Treidel et al. “Fast-Switching GaN-Based Lateral Power Schottky Barrier Diodes With Low Onset Voltage and Strong Reverse Blocking”. In: *IEEE Electron Device Letters* 33.3 (2012), pp. 357–359. DOI: 10.1109/LED.2011.2179281.

Impressum

Hans Meier, Michael Niemetz, Thomas Fuhrmann, Andrea Reindl
Ostbayerische Technische Hochschule Regensburg
Fakultät Elektro- und Informationstechnik
Seybothstraße 2
93053 Regensburg