3·4·2019

ЕЛЕКТРОТЕХНИКА И ЕЛЕКТРОНИКА

ELECTROTECHNICA & ELECTRONICA

E+E

FEDERATION OF THE SCIENTIFIC ENGINEERING UNIONS

WWW.FNTS.BG

ELECTROTECHNICA & ELECTRONICA E+E

Vol. 54. No 3-4/2019

Monthly scientific and technical journal

Published by:

The Union of Electronics, Electrical Engineering and Telecommunications /CEEC/, BULGARIA

Corresponding address:
108 Rakovski St.
Sofia 1000
BULGARIA
Tel. +359 2 987 97 67
e-mail: epluse@mail.bg
http://epluse.fnts.bg

ISSN 0861-4717

C O N T E N T S

In 2019 the Journal Electrotechnica & Electronica is published with the financial support of the Bulgarian Science Fund, Ministry of Education and Science.

Comparative analysis of the possibilities of building a decentralized control plane of a Software Defined Networks

Jordan Raychev, Diyana Kinaneva, Georgi Hristov, Plamen Zahariev

The objective of the current paper is to be conducted a comparative analysis of the available possibilities for building a decentralized controller plane of a Software Defined Network (SDN). The SDN networks are an emerging paradigm, which aims to change telecommunications networks as we known them today. The constantly growing social media, mobile communications and technologies like server virtualization that can be found in the modern data centers are pushing the limits of traditional computer networks. That led to a lot of researching in the field of network virtualization and more specifically the field of software defined networking. One of the most important challenges in front of the new concept for software defined networking is the single point of failure that is created due to the centralization of the control. Unfortunately, as every new technology and this one have its flaws. One of the biggest research fields is investigating the possibilities of building a decentralized control plane to avoid issues related to the single point of failure, scalability, high availability and security.

Сравнителен анализ на възможностите за изграждане на децентрализирана контролна равнина при софтуерно дефинираните мрежи (Йордан Райчев, Дияна Кинанева, Георги Христов, Пламен Захариев). *Целта на настоящият доклад е да бъде извършен сравнителен анализ на познатите до момента решения за създаване на децентрализирана контролна равнина при софтуерно дефинираните (СД) мрежи. Софтуерно дефинираните мрежи представляват съвременна парадигма, чиято основна цел е да промени телекомуникационните мрежи, които познаваме днес. Непрекъснато разрастващите се социални мрежи, мобилните комуникации и технологии като сървърната виртуализация, намерила широко приложение в съвременните центрове за съхранение на данни и публичните облачни услуги достигат лимита на възможностите на традиционните телекомуникационни мрежи. Това налага изследването и имплементирането на технологии като мрежовата виртуализация и по-конкретно софтуерно дефинираните мрежи. Един от основните недостатъци на първоначалната концепция на софтуерно дефинираните мрежи е създаването на единична точка за отказ на мрежата поради наличието на един единствен управляващ контролер. Съществуват редица изследвания, които са съсредоточени върху този проблем и неговото отстраняване. Всички тези изследвания се свеждат до създаване на децентрализирана контролна равнина, която има за цел, както да отстрани изцяло проблем с отпадането на управляващия контролер и загуба на пълна свързаност в софтуерно дефинираната мрежа, така и увеличаване мащабируемостта и сигурността на дадената мрежа.*

Keywords – software defined networks, decentralization, single point of failure

Introduction

The requirements of modern application and their users are pushing the conventional communication networks to their limits. Dynamic development of the modern information and communication technologies during the last decade is the major factor, which forces current network architecture to be revised. Providing high quality of services, dynamic,

centralized and programmable management of telecommunication networks are part of the issues that arise when it comes down to building and standardizing new network architectures like software defined networks [1].

Software defined networks is a new technology which provide programmable management of computer networks. They aim to minimize the problems that accompany traditional telecommunication networks. Such problems are management difficulty, the lack of flexibility and scalability, heterogeneity of network devices and others. The idea of creating programmable computer networks was born more than a decade ago [2] with a proposal to increase the level of programmability of the communication networks that can be achieved by separating functionality of control plane from the data plane. This in turn would allow rapid and effective deployment of new network services, coexistence of network devices from different manufacturers without creating conflicts, virtualization of network equipment and others. The major and most important stage from the development process of the new architecture is the separation of control plane from the data plane which sits at the physical layer. Physical and logical separation of the two planes provides the possibility for dynamic deployment of new functions to networks, something that would be practically impossible with conventional networks. In software defined networks this is achieved by physical separation of the control plane and centralizes its functions to a separate network device called network controller. Using a centralized control plane has its benefits, for example, the controller has a single point of view that facilitate network management which is expressed in traffic optimization, easy and rapid functional deployment, reducing management cost and so on.

Software-defined network architecture

As it can be noticed in fig. 1, which shows network architecture with centralized controller, all control functions from the physical layer are moved to the control plane as a separate device known as controller. Such separation of the control logic leaves the devices at the physical layer with no control functionalities, just a simple flow table, which entries are populated via the SDN controller.

Flow tables that are illustrated in fig. 1 represent pseudo flow tables which reflect packet switching from high level of abstraction. In this line of thought the real SDN switches and in particular their flow tables are composed of multiple elements (not only these shown in fig.1) such as match field, priority, packet counter, instruction and so on, that are not discussed in this paper.

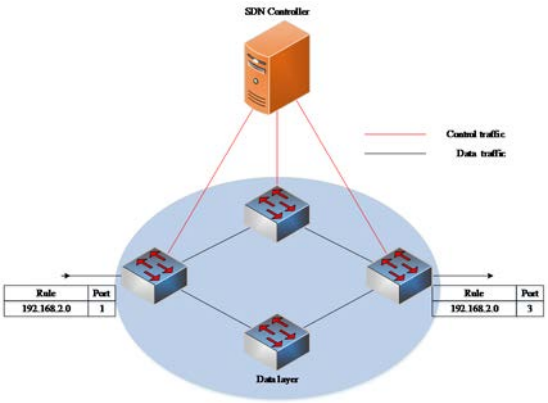


Fig. 1. Simple Architecture of SDN network.

For the purpose of our explanation it is only important to be noticed that when a packet arrive at the ingress port, the SDN switch looks-up the flow table in order to decide how to process the packet. There are two outcomes in such situation. First, if the switch has an entry in its flow table for the corresponding packet (there is a match) will be switched to egress port regarding the rule found in the entry. In the second case the switch does not have any entry for the destination of the incoming packet. In that case the switch will query the controller and as a response the controller will install an entry with a corresponding rule to the flow table of the switch and then the switch will have the capability to forward all subsequent packets. As it can be seen, the SDN switch is just a forwarding device that can switch or route packets regarding the rule installed from the controller. Routing and switching are interchangeable terms in the context of software defined networks and the only difference from conventional networks is in the way how packets and frames are processed. In conventional networks each device has its own control plane and makes decisions by itself, while in SDN networks the controller hold the control and decides how the network will operate generally.

The architecture of software defined networks presented in [1] and its newer version [3] is just a reference and it is used as base for most modern implementation of SDN controllers. However this architecture does not give a description how the control place could be implemented – distributed or centralized. The answer to this question might seem easy to determine based on the analysis provided above, but it is actually a quite difficult task. Using single management controller raises questions related to scalability of the network and control plane

performance issues. Researches show for example that the NOX controller managed to process 30,000 flows per second [4]. With the constantly increasing requirements of the modern data centers and even the needs of most Internet service providers (ISP) this flow processing speed is insufficient. Scalability of the control plane is a major issue for successful implementation of SDN in modern communication networks. The control plane of SDN networks is scalable only if its architecture is able to dynamically adapt to the constant changes occurring in the network as well as satisfying the quality of services. In order to achieve this level of scalability two approaches are proposed. The first approach is related to transferring some of the function of control plane to the switches in forwarding plane. This will transfer some of the tasks from the SDN controller to the switches, which in turn will reduce the queries sent to the control plane and at the same time will increase the overall network performance. However, the technological implementation of that approach is quite difficult. The feasibility of such solution is accompanied by creating integrated motherboards that would enhance the functionalities of the forwarding elements. However, moving part of the control function back to forwarding devices breaks the ideology behind the software defined networks, turning them back to conventional communication networks.

The second approach is to create a distributed control plane architecture where tasks are distributed among multiple SDN controllers. This approach allows queries to be process in parallel by different SDN controllers while at the same time the controllers communicate between each other in order to synchronize the information and avoid conflicts. The advantage of this approach is that technological simplicity of network elements from forwarding plane is kept. Another advantage of this approach is that the idea behind SDN networks and the centralized management are also kept and in addition the single-point of failure is not an issue any more since there are more than one controllers that can continue working if one controller fails.

Models of managing the control plane of software defined networks

The management of the control plane can be classified into three main categories – centralized, decentralized and hierarchical. Decentralized and hierarchical management can be classified into a single category – distributed management structures.

Model of centralized control management

The centralized control management is a model that conceptually describes the relationship between individual nodes of a particular structure. In the context of computer networks centralized control management means the management of a particular group of resources through a single master control unit. The first architecture of software defined networks is represented by a centralized control management model, as shown in fig.2. In this management model all queries are sent to the controller, which is responsible for their processing and giving a response back to network elements of forwarding plane. The controller is able to manage the network since it is the only element of the network that has a global view of the topology. Having a global view of the network gives the controller the ability to perform correct routing decisions.

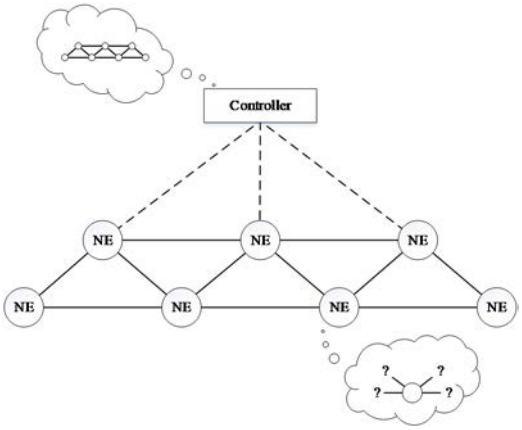


Fig. 2. Centralized control management of SDN networks.

Centralized control management has the following features and advantages:

- The ability for easy management due to the fact that in most cases the controller is implemented into a single physical server;
- Improved network security due to the minimum of logical and physical connection from and to the controller;
- Low cost of the elements forming the data plane;
- Ability to make better use of available physical resources.

There are also some disadvantages of centralized control management:

- Using a one centralized controller opens a vulnerability place in the network even in software and hardware. In case of software vulnerably, for example found in the operating system of the controller would compromise the entire network. Hardware vulnerability could be

seen in case of controller failure where the network becomes unmanageable;

- Lack of scalability. Increasing the number of forwarding devices in the data plane will increase the number of queries sent to the controller. In this case the controller would not have the ability to process the query on time due to network congestion. The entire network performance decrease as a result.

Model of decentralized control management

Decentralized computer system is a group of autonomous network elements, which communicate with each other in order to implement particular task. A distinctive feature of decentralized control management is that control elements can be located in a physical room but most often they are located in geographically different locations. An example of decentralized system is telephone system. The telephone system can be perceived as a heterogeneous system due to the fact that it is made up of many different devices (computers, terminals, etc.) [5].

In the context of software defined networks decentralized system has a different meaning. Due to the fact that in SDN networks the control plane is physically separated for data plane it is necessary to find an approach that will distribute the management among network elements from the control plane. In case of a large amount of queries the overall load can be distributed among the controllers, which would remove the problem with overloading found in the centralized approach for management. In case of controller failure there will always be another controller which will take over the control. In that case the operation of the network will not be interrupted and it will continue working. All of the aforementioned are only part of the advantages of decentralized control management. There are other advantages such as:

- Scalability;
- Security;
- Ability to fast recovery;
- High availability.

There are several disadvantages of this category management:

- Higher cost of network maintenance;
- Lower efficiency compared to centralized management due to the fact it is almost impossible to use all of the available resource;
- The network becomes more complex.

Decentralized control management in software defined network is conditionally divided in two subgroups – management with global and local view of the network topology. In case of management with global view of the network topology the control plane is implemented by N SDN controllers, each of which has global view of the network topology. One of the main advantages of this type of network architecture is the small delay of responses to the queries coming from the physical plane and going to the controllers. The architecture of decentralized control plane is given in fig. 3.

Decentralized management with local view of network topology has the following characteristics: the control plane is again implemented by multiple controllers in order to provide scalability but this time the controllers do not have a global view of the network topology. Instead of having global view of the network topology the controllers only manage the devices that are in their administrative domain.

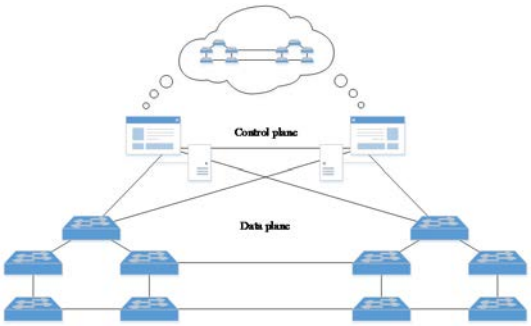


Fig. 3. Architecture of decentralized control management with global view of network topology.

In opposed to decentralized control management with global view of the network topology, where the delay for replay the queries is significantly low, the delay in decentralized management with local view of the topology would be higher due to the fact that in some cases the controller in one domain have to communicate with controllers in other domains. In this case the time for synchronizing the information between the controllers in different domain would add additional delay.

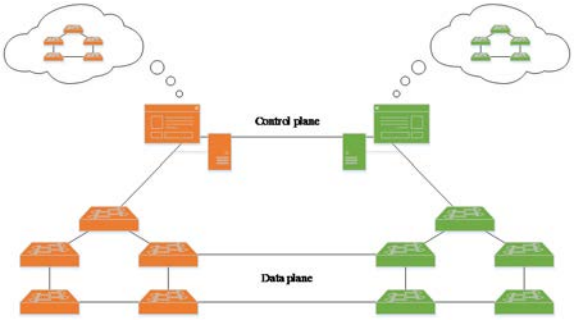


Fig. 4. Architecture of decentralized control plane with local view of the network topology.

The architecture of control plane implementation with decentralization and local view of the network topology is given in fig. 4. In order to communicate the controllers from different domain use the so called east-west interfaces. East-west communication is used for synchronizing control information between controllers in different domains.

Model of hierarchical control management

The model of hierarchical implementation of control plane can be referred to the model with decentralized control plane implementation with local topology view. The difference between the two is that in hierarchical implementation there is one additional master controller that will control all of the other controllers. The architecture of the hierarchical model is given in fig. 5. As it can be noticed in the figure, the network topology again is separated to several administrative domains and each domain is managed by the corresponding for the domain controller. In addition to the domain controllers the architecture adds additional controller which main purpose is to monitor and manage the domain controllers.

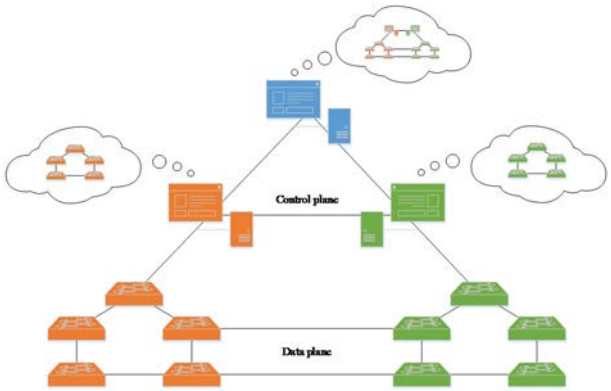


Fig.5. Architecture of decentralized control plane with hierarchical view of network topology.

A major disadvantage of this type of control plane implementation is the existence of additional controller which adds complexity to the architecture of SDN networks and creates a prerequisite for overlapping the controllers' functionalities.

The hierarchical structure of the control plane separates the controllers in two main categories – global and local. The local SDN controller manages only one administrative domain. They generate rules and install them in local flow tables. The global SDN controller on the other hand manages multiple administrative domains and participates in the routing decision if and only if there is a need for global view of network topology.

Types of architectures for building a decentralized control plane in SDN

Different models for control plane management in Software defined networks have been examined. In this section a comparative analysis will be carried out between existing solutions for building a decentralized control plane.

Kandoo

The architecture of Kandoo control plane [6] is based on the model with hierarchical management. This type of network architecture differs greatly from the abovementioned solutions for control plane management. The control plane in Kandoo architecture is realized based on two levels of hierarchy, which is achieved by using additional controller also called local controller. This network architecture type is shown in fig.6. The local controller is responsible for queries that occur more frequently while the main controller response to all other queries.

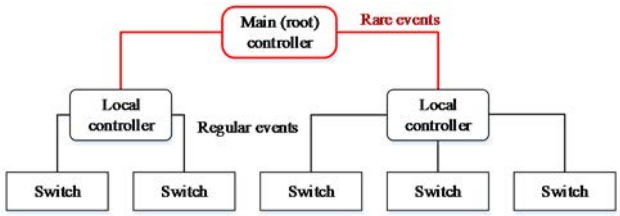


Fig.6. Two-layered hierarchy of control plane architecture.

With the structure shown in fig.6 the local controllers manage one or more switches in their domains, while the main (root) controller is responsible for managing all local controllers or in particular the main controller manage more rarely happened events. A major characteristic of main controller is that it is the only device in the network than has a global view of network topology. Another feature of the architecture is the ability to implement local controller functions in device from the access layer (the switches) which will significantly simplify the structure of the network. The approach of Kandoo architecture is a diversion from the ideology of software defined networking where the main approach relays on physical separation of control logic from data plane.

ONIX

The decentralized architecture of ONIX controller [7] is composed of four main elements – physical and communicational infrastructure; application programming interface of ONIX, and control logic.

The **physical infrastructure** consists of network devices (switches and routers) that work at datalink and network layers of the OSI (Open System Interconnection) other devices that are used as traffic balancers. There are requirements for ONIX devices – they have to support ONIX application layer protocol with read and write permissions.

The **Communication infrastructure** is an element from the architecture which is responsible for network connectivity between physical infrastructure and the application of ONIX. The control channel that is built between the physical infrastructure and the applications can be implemented in two different ways – in-band and out-of-band. In in-band implementation the control channel is shared between the control traffic and the data traffic, while out-of-band implementation separates the control traffic from data traffic via a separate communication channel.

The distributed control system is implemented via a cluster of multiple physically connected servers. On each server several application programming interfaces (APIs) can be installed. The APIs give read and write permissions to network resources.

The control logic is the last element of the system and it is implemented above the application programming interface of ONIX. The major function of control logic is to determine the state of each object from the physical infrastructure. A copy of each determined state for each physical object is stored in a database which is called NIB or Network information base. The NIB database is similar to the routing tables which are essential for network devices that work in layer 3 of the OSI model.

HyperFlow

The architecture of HyperFlow [8] consists of OpenFlow switches (which built-up the data plane) and NOX SDN controllers (which built-up the control plane). HyperFlow is running on each controller and there are also an application and a system which are responsible for the communication between the controllers. A characteristic of this architecture is the fact that each controller has a global view of the network topology and each controller work as an independent unit such as it is the only controller in the network. Fig. 7 illustrates an example network which uses the HyperFlow network architecture. As it can be noticed in fig. 7 each controller manages a group of switches usually the nearest ones. In case of controller failure the switches under its management are reconnected to other controller again the nearest one.

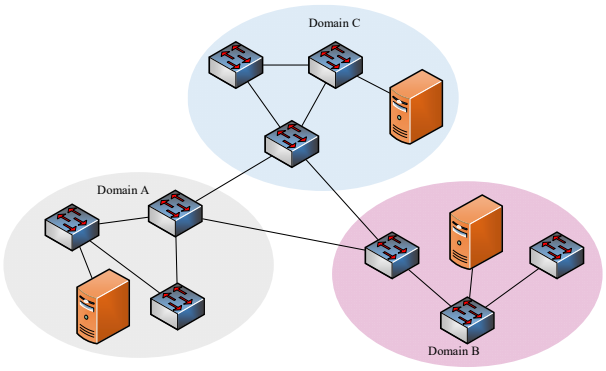


Fig.7. HyperFlow architecture.

DIFANE

The architecture called DIFANE (Distributed Flow Architecture for Network Enterprises) [9] is a different approach for control plane management and implementation. DIFANE architecture increases network performance and scalability by moving part of the responsibilities of the controller back to the access layer – in particular to a specialized switch called Authority switch. The role of Authority switch is to manage a group of switches from data layer. In a standard SDN architecture when a packet arrives in the switch ingress port, the switch looks for a match in its flow table. If there is a match the switch forwards the packet to its destination, otherwise the switch sends a query to the SDN controller and the controller response with a rule how the switch should process the packet. In DIFANE architecture when a switch receives a first packet from a flow it automatically sends the packet to Authority switch. The Authority switch is responsible for packet delivery to its destination, as shown in fig. 8. This is possible due to the controller pre-installs rules in the flow table of Authority switch. The second packet from the flow is forwarded by the switch from the data plane since it already has a cashed entry in its flow table.

The architecture illustrated in fig. 8 has two main ideas:

- The main (root) SDN controller distributes and pre-installs rules in the Authority switches with the goal to improve the scalability of the network. The rules are equally distributed among all authority switches;
- Switches in the data plane process the traffic based on cashed rules. In case of no cashed rules the switch redirect packets to authority switches.

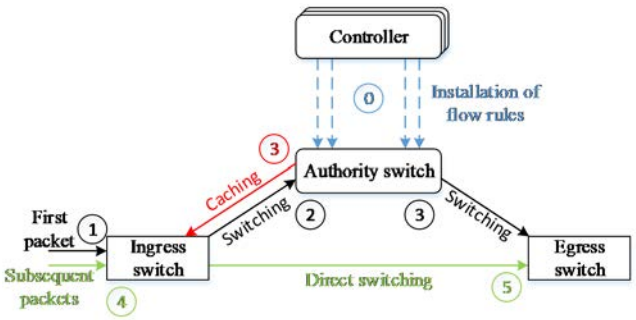


Fig. 8. DIFANE architecture.

DevoFlow

The idea of DevoFlow architecture presented in [10] is similar to DIFANE architecture – moving part of the responsibilities of the controllers back to the switches or in particular applying additional control logic to the switched which the goal to keep most of the traffic flows in the data plane. The authors represent several new mechanisms for distinguishing the traffic between control plane and data plane. The idea behind DevoFlow breaks the concept of SDN networks for separating control logic from data plane.

ONOS

ONOS (Open Network Operating System) [11] is distributed system for building decentralized control plane with high performance, availability, scalability, and security. ONOS is extremely flexible system, which allows implementing centralized or decentralized control plane with global view of network topology. Researches [12] show that the architecture of the control plane with global view of the network topology does not provide the highest level of scalability, however the authors choose to implement such architecture and on top of that divide the topology into logical administrative domains. The ONOS approach transforms the control plane architecture to architecture with local view of the network topology, which has a better scalability.

Conclusion

Software defined networks rely on the idea for logical separation of the control plane from the data plane while at the same time the control plane is physically move to a centralized server also known as SDN controller. Despite the numerous advantages and applications of such approach it is important to remember that centralized architectures do not scale well and have scalability and availability issues.

A lot of researches have been carried out during the past several years exploring the possibilities of building distributed management for the control plane that would increase the availability, security and scalability of SDN networks.

The main characteristics of the different architectures – centralized, decentralized with local or global view of the network topology and hierarchical, have been revealed in this paper. In addition different control plane solutions that are available were also reviewed. From the analysis it can be concluded that building decentralized control plane (no matter of its type) is obvious and a must in order to meet the requirements for scalability and availability of networks of large sizes (data centers, ISPs and others) which also have demand for high quality of services, availability and low latency.

Acknowledgement

The work presented in this paper is completed as partial fulfilment of a PROJECT 2018 - FEEA - 02 “Development and evaluation of a methodology for sensing, analysis and visualization of open data through LoRaWAN based communication infrastructure”, financed under the Scientific Fund of the University of Ruse.

REFERENCES

[1] Open Networking Foundation, SDN architecture, Issue 1, June 2014.

[2] Campbell, A., De Meer, H., Kounavis, M., Miki, K., Vicente, J., and D. Villela, "A Survey of Programmable Networks", ACM SIGCOMM Computer Communication Review, Volume 29, Issue 2, pp. 7-23, September 1992.

[3] Open Networking Foundation, SDN Architecture, Issue 1.1, 2016.

[4] A. Tavakkoli et al., “Applying NOX to the Datacenter,” Proc. ACM HotNets-VIII Wksp., 2009.

[5] Gray J., An Approach to Decentralized Computer Systems, Technical Report 85.4, June 1985.

[6] S. Yeganeh , Y. Ganjali, Kandoo: a framework for efficient and scalable offloading of control applications, Proceedings of the first workshop on Hot topics in software defined networks, August 13-13, 2012, Helsinki, Finland.

[7] T. Koponen, M. Casado, N. Gude, J. Stribling, L. Poutievski, M. Zhu , R. Ramanathan, Y. Iwata, H. Inoue, T. Hama, S. Shenker, Onix: a distributed control platform for large-scale production networks, Proceedings of the 9th USENIX conference on Operating systems design and implementation, p.1-6, October 04-06, 2010, Vancouver, BC, Canada.

[8] A. Tootoonchian and Y. Ganjali, “HyperFlow: a distributed control plane for OpenFlow,” in Proceedings of the 2010 internet network management conference on Research on enterprise networking, ser. INM/WREN’10. Berkeley, CA, USA: USENIX Association, 2010, pp. 3–3.

[9] M. Yu, J. Rexford, M. J. Freedman, and J. Wang, “Scalable flow-based networking with difane,” SIGCOMM Comput. Commun. Rev., vol. 41, no. 4, pp. –, Aug. 2010.

[10] A. R. Curtis, J. C. Mogul, J. Tourrilhes, P. Yalagandula, P. Sharma, S. Banerjee, “DevoFlow: scaling flow management for high- performance networks,” Comput. Commun. Rev., vol. 41, no. 4, pp. 254–265, Aug. 2011.

[11] P. Berde, M. Gerola, J. Hart, Y. Higuchi, M. Kobayashi, T. Koide, B. Lantz, B. O’Connor, P. Radoslavov, W. Snow, and G. Parulkar, “ONOS: Towards an open, distributed SDN OS,” in Proceedings of the Third Workshop on Hot Topics in Software Defined Networking, ser. HotSDN ’14. New York, NY, USA: ACM, 2014, pp. 1–6.

[12] Hu, Jie et al. “Scalability of control planes for Software defined networks: Modeling and evaluation.” 2014 IEEE 22nd International Symposium of Quality of Service (IWQoS) (2014): 147-152.

Received on: 05.03.2019

Spectra of multilayer networks - mathematical foundations, metrics, spectral properties

Mircho J. Mirchev

This article reviews a generalize approach of network models of complex systems and networks by introducing layers, thus defining multilayer networks. Having the multilayer property on the network model, it can better describe the complex system and give better results in the analysis of network science. In this paper are given the mathematical foundations and definitions of generalized multilayer networks, as well as typical classes of multilayer networks with example references. The approaches of deriving the spectra of such networks are also given. Based on the mathematical foundations and given approaches, it is possible to analyze and compare networks with different layers and from different classes.

Keywords – multilayer network, network model, network science, mathematical foundation, spectral property.

Спектрални характеристики на многонивови мрежи - математическа база, метрики и спектрални свойства (Мирчо Й. Мирчев). *Настоящата статия разглежда генерализиран модел на мрежи, като въвежда отделни слоеве в мрежовото описание на комплексните системи. Въвеждането на слоеве в мрежовите модели на комплексните системи позволява по точното им описание и анализ с инструментариума на науката за мрежи. В разработката са представени математически описания и дефиниции на слоестите мрежови модели, както и различните класове такива с примери от литературата. Показани са насоките за изчисление и анализ на спектралните характеристики на моделите. На база на посочените математически модели е възможно и сравнението на мрежови модели с различни слоеве и от различни класове.*

Introduction

Network theory is an important tool for describing and analysing complex systems throughout the social, biological, physical, information and engineering sciences. The broad applicability of networks and their success in providing insights into the structure and function of both natural and designed systems have thus generated considerable excitement across myriad scientific disciplines. For example, networks have been used to represent interactions between proteins, friendships between people, hyperlinks between Web pages, and much more. Importantly, several features arise in a diverse variety of networks. For example, many networks constructed from empirical data exhibit heavy-tailed degree distributions, the small-world property, and/or modular structures; such structural features can have important implications for information diffusion, robustness against component failure, and many other considerations. Traditional studies of networks generally assume that nodes are connected to each other by a single type of static edge that encapsulates all connections between them. So we can initially think each network layer as a separate analysis and optimization problem. This assumption is almost always a gross oversimplification, and it can lead to misleading results and even the sheer inability to address certain problems. Most real and engineered systems usually have multiple subsystems and layers of connectivity, and the data produced by such systems are very rich [1]–[4]. To represent such systems consisting of networks with multiple types of links, or other similar features, we consider structures that have layers in addition to nodes and links. In its more general form, in a multilayer framework a node u in layer α can be connected to any node v in any layerβ. In practice problems become coupled as upper layer “links” become lower layer “demands”. In technological multilayer networks (examples of which are given below) the layers are usually defined be the OSI model and upper layer “links” are implemented

via lower level "paths" [5]. In this case also often each upper layer node is associated with one lower layer node (adaptation "links"), while the opposite is not true [6].

Examples of technological networks that can and should be considered multilayer in their analysis and optimizations are:

- IP over some L2 protocol
- IP over MPLS over some L2 protocol
- Ethernet over WDM
- Ethernet over TDM
- WAN tunnelling
- TDM over WDM
- etc.

As above examples usually follow the OSI layered model, this is not always the case. Today's technological advances led to the concept of overlay networks, which is considered as a lower level layer network over a higher level network, e.g. L2 (Ethernet) transport over IP (VXLAN/EVPN).

In the terms of today's social networks and our *hyper connected* [7], [8]world, multilayer network are also broadly covered. In example one person can know others (i.e. is connected) via several social networks - Facebook, LinkedIn, etc. In essence this is also considered as a multilayer network, where nodes are people, links are social connections and layers are the different means of social networks. Usually in the case of social multilayer networks there is strict one-to-one mapping of nodes between layers and such networks are a subset of multilayer networks called *multiplex networks* [3], [9], [10].

In the case of transportation networks, different means of transportation, or different airlines or operators also create multilayer networks. Such networks can be multiplex (in the air transportation, e.g.), but this is not always the case. In land and sea transportation networks, there is middle (inter-layer) network that defines the relation between nodes in the different layers [11]–[13].

Most recently, there have been increasingly intense efforts to investigate networks with multiple types of connections and so-called "network of networks [3], [4], [14]". Such systems were examined decades ago in disciplines like sociology and engineering, but the explosive attempt to develop frameworks to study multilayer complex systems and to generalize a large body of familiar tools from network science is a recent phenomenon.

So studying the properties of multilayer networks as such, and not simply studying them layer by layer, is the key to successful understanding of the structure, dynamics and behaviour to the real life complex systems represented by networks.

Types of multilayer networks

In this article we start by presenting the most general notion of a multilayer-network structure and by defining various constraints for that structure. We then show how this structure can be represented as an adjacency tensor and how the rank of such a tensor can be reduced (i.e. order) by constraining the space of possible multilayer networks or by 'flattening' the tensor. Taken to its extreme, such flattening process yields "supra-adjacency matrices" (i.e. "super-adjacency matrices"), which have the advantage over tensors of being able to represent missing nodes in a convenient way - when implementing methods of computation, especially on eigenvalues and eigenvectors, there are much more tools and mechanisms for working with matrices than with tensors[15], [16].

Multilayer networks generally inherit the properties of graphs, but also introduce new properties. The new properties are related to the additional term of layers and the relationship between them. The properties of various multilayer-network structures from the literature, represented using the general multilayer-network structure, are shown in Table 1.

Some the properties that differentiate the types of multilayer networks are:

- Aligned - Is the network node aligned?
- Disjoint - Is the network layer-disjoint?
- Equivalent Size - Do all of the layers have the same number of nodes?
- Diagonal - Are the couplings diagonal?
- Layer couplings - Do the inter-layer couplings consist of layer couplings?
- Categorical - Are the inter-layer couplings categorical?
- Additionally each network has a number of possible layers "L|, and also different number of "aspects" (*d*).

Also some network structures can be represented in multiple different ways using the multilayer-network framework. For example, a network structure that consists of a sequence of graphs that share the same set of nodes can be represented using either categorical or ordinal couplings - this is usually dependant on the goal of the analysis.

Table 1 - Various types of multilayer networks with different properties and references in the literature

Name	Algn	Disj	Eq.size	Diag.	Lcoup.	Cat	|L|	d	Example refs.
Multilayer network				✓	✓	✓	Any	1	[13][1]
Multiplex network	✓		✓	✓	✓	✓	Any or 2	1	[17][18][19][20][10][21][22][23][24]
Multivariate network	✓		✓	✓	✓	✓	Any	1	[25]
Multinetwork	✓		✓	✓	✓	✓	Any	2	[26][27]
Multirelational network	✓		✓	✓	✓	✓	Any	1	[11][28][29][30]
Multislice network	✓		✓	✓			Any	1	[31][32][33][34]
Multiplex of independent networks	✓		✓	✓	✓	✓	Any	1	[35]
Hypernetwork	✓		✓	✓	✓	✓	Any	1	[36][37]
Overlay network	✓		✓	✓	✓	✓	2	1	[38][39]
Composite network	✓		✓	✓	✓	✓	2	1	[40]
Multilevel network		✓					Any	1	[41][42]
Multiweighted graph	✓		✓	✓	✓	✓	Any	1	[43]
Heterogeneous network		✓					2	1	[44][28]
Multitype network		✓					Any	1	[45][46][47]
Interconnected networks		✓	✓				2	1	[48][49]
Interdependent networks		✓	✓				2	1	[50][51][52][53][54][55][56]
Network of networks			✓				Any	1	[14]
Coupled networks				✓	✓	✓	Any	1	[57]
Interconnecting networks				✓	✓	✓	2	1	[58]
Interacting networks		✓					Any or 2		[59][60]
Heterogeneous information network		✓					Any	2	[61] [61], [62] [63][64]
Meta-matrix, meta-network							Any	2	[65][66][67]

Mathematical representation

The issues posed by the multiscale modelling of both natural and artificial complex systems call for a generalization of the "traditional" network theory, by developing a solid foundation and the consequent new associated tools to study multilayer and multicomponent systems in a comprehensive fashion.

A lot of work has been done during the last years to understand the structure and dynamics of these kind of systems. Related notions, such as networks of networks, multidimensional networks, multilevel networks, multiplex networks, interacting networks, interdependent networks, and many others have been introduced, and even different mathematical approaches, based on tensor representation or otherwise, have been proposed [1], [9]. The purpose of this section is to survey and discuss a general framework for multilayer networks and review some attempts to extend the notions and models from single layer to multilayer networks. As we will see, this framework includes the great majority of the different approaches addressed so far in the literature.

General form

A multilayer network is the pair $\mathcal{M} = (\mathcal{G}, \mathcal{C})$, where $\mathcal{G} = \{G_\alpha;\ \alpha \in \{1, \dots, M\}\}$ is a family of (directed or undirected, weighted or unweighted) graphs $G_\alpha = (N_\alpha, E_\alpha)$, called layers of $\mathcal{M}$, and

$$\mathcal{C} = \left\{E_{\alpha\beta} \subseteq X_\alpha \times X_\beta; \alpha, \beta \in \{1, \dots M\}, \alpha \neq \beta\right\} \quad (1)$$

is the set of interconnection between nodes in different layers G_α and G_β with $\alpha \neq \beta$. The elements of $\mathcal{C}$ are called *crossed layers,* and the elements of each E_α are called *intralayer* connections of $\mathcal{M}$ in

contrast with the elements of each $E_{\alpha\beta}$ $(\alpha \neq \beta)$ that are called *interlayer* connections.

The set of nodes of the layer G_α is denoted by $N_\alpha = \{n_1^\alpha, \cdots, n_{N_\alpha}^\alpha\}$ and the adjacency matrix of each layer G_α is denoted by $A^{[\alpha]} = \left(a_{ij}^\alpha\right) \in \mathbb{R}^{N_\alpha \times N_\alpha}$, where

$$a_{ij}^\alpha = \begin{cases} 1, & if\ \left(n_i^\alpha, n_j^\alpha\right) \in E_\alpha, \\ 0, & otherwise \end{cases} \quad (2)$$

for $1 \leq i, j \leq N_\alpha$ and $1 \leq \alpha \leq M$. The *interlayer adjacency matrix* corresponding to $E_{\alpha\beta}$ is the matrix $A^{[\alpha,\beta]} = \left(a_{ij}^{\alpha\beta}\right) \in \mathbb{R}^{N_\alpha \times N_\beta}$ given by:

$$a_{ij}^{\alpha\beta} = \begin{cases} 1, & if\ \left(n_i^\alpha, n_j^\beta\right) \in E_{\alpha\beta}, \\ 0, & otherwise. \end{cases} \quad (3)$$

The projection network of $\mathcal{M}$ is the graph $proj(\mathcal{M}) = (N_\mathcal{M}, E_\mathcal{M})$, where:

$$X_\mathcal{M} = \bigcup_{\alpha=1}^{M} X_\alpha,$$

$$E_\mathcal{M} = \left(\bigcup_{\alpha=1}^{M} E_\alpha\right) \bigcup \left(\bigcup_{\substack{\alpha,\beta=1 \\ \alpha \neq \beta}}^{M} E_{\alpha\beta}\right). \quad (4)$$

The adjacency matrix of $proj(\mathcal{M}) = (N_\mathcal{M}, E_\mathcal{M})$, is denoted as $\overline{A_\mathcal{M}}$.

Tensor representation

There have been some attempts in the literature for modelling multilayer networks properly by using the concept of tensors.[9] There are two main ways to think about tensors:

- tensors as multilinear maps;
- tensors as elements of a tensor product of two or more vector spaces.

The former is more applied. The latter is more abstract but more powerful. The tensor product of two real vector spaces $\mathcal{V}$ and $\mathcal{L}$, denoted by $\mathcal{V} \otimes \mathcal{L}$ (i.e., the Kronecker product), consists of finite linear combinations of $v \otimes w$, where $v \in \mathcal{V}$ and $w \in \mathcal{L}$.

The dual vector space of a real vector space $\mathcal{V}$ is the vector space of linear functions $f: \mathcal{V} \longrightarrow \mathbb{R}$, indicated by $\mathcal{V}^*$. Denoting by $Hom(\mathcal{V}^*, \mathcal{L})$ the set of linear functions from $\mathcal{V}^*$ to $\mathcal{L}$, there is a natural isomorphism between the linear spaces $\mathcal{V} \otimes \mathcal{L}$ and $Hom(\mathcal{V}^*, \mathcal{L})$. Moreover, if $\mathcal{V}$ is a finite dimensional vector space, then there exists a natural isomorphism (depending on the bases considered) between the linear spaces $\mathcal{V}$ and $\mathcal{V}^*$. In fact, if $\mathcal{V}$ is finite-dimensional, the relationship between $\mathcal{V}$ and $\mathcal{V}^*$ reflects in an abstract wat the relation between $(1 \times N)$ row-vectors and the $(N \times 1)$ column-vectors of a $(N \times N)$ matrix. However, if $\mathcal{V}$ and $\mathcal{L}$ are finite-dimensional, them the linear spaces $\mathcal{V} \otimes \mathcal{L}$ and $Hom(\mathcal{V}, \mathcal{L})$ can be identified. Thus a tensor $\sigma \in \mathcal{V} \otimes \mathcal{L}$ can be understood as a linear function $\sigma: \mathcal{V} \longrightarrow \mathcal{L}$, and therefore, once the two bases of the corresponding vector spaces are fixed, a tensor may be identified with a specific matrix.

Multilayer networks, multidimensional networks, hypergraphs, and some other objects can be represented with tensors, which represent a convenient formalism to implement different models. Specifically, tensor-decomposition methods and multiway data analysis have been used to study various types of networks. These kind of methods are based on representing multilayer networks as adjacency higher-order tensors and then applying generalizations of methods such as Singular Value Decomposition (SVD), and the combination of CANDECOMP (canonical decomposition) and PARAFAC (parallel factors) leading to CANDECOMP/PARAFAC (CP). These methods are used to extract communities, to rank nodes and to perform data mining [1], [68].

Supra-adjacency representation

The adjacency-matrix representation for monoplex (single-layer) networks is powerful because of the numerous tools, methods and theoretical results that have been developed for matrices can be exploited. To get access to these tools for investigations of multilayer networks, such networks can be represented using supra-adjacency matrices. An additional advantage of supra-adjacency matrices over adjacency tensors is that they provide a natural way to represent multilayer networks that are not node-aligned without having to append empty nodes. However, this boon comes with a cost: flattening a multilayer network to obtain a supra-adjacency matrix loses some of the information about the aspects. Partitioning a network's edge set into intra- layer edges, inter-layer edges and coupling edges makes it possible to retain some of this information.

If the multilayer network is a multiplex network (i.e. there each node participates in all layers), it can be represented as network $\mathcal{M}$ made of M layers $\{G_\alpha; 1 \leq \alpha \leq M\}, G_\alpha = (N_\alpha, E_\alpha)$, with $N_1 = \cdots = N_M = \{n_1, \ldots, n_N\} = N$ and described as a tensor product:

$$\mathcal{M} = \langle N \rangle \otimes \langle \{\ell_1, \ldots, \ell_M\} \rangle, \quad (5)$$

where ℓ_i represents the layer G_i.

Then according to [xx], a multilayer network $\mathcal{M}$ of N nodes and with layers $\{\ell_1, \ldots, \ell_M\}$ can be

identified with a linear transformation $\sigma: \mathcal{V} \otimes \mathcal{L} \longrightarrow \mathcal{L} \otimes \mathcal{V}$ and therefore $\mathcal{M}$ is completely determined by the matrix associated with σ on the basis

$$n_i \otimes \ell_\alpha; 1 \leq i \leq N, 1 \leq \alpha \leq M \quad (6)$$

Also if A and B are two different matrices assigned to the same tensor, there exists a permutation matrix P such that $B = P \cdot A \cdot P^{-1}$, and thus both matrices have the same spectral properties.

So for any tensor of a multilayer network we can get an adjacency matrix, called *supra-adjacency matrix* of $\mathcal{M}$, which can be written in the usual form

$$\mathcal{A} = \begin{pmatrix} A_1 & \cdots & I_N \\ \vdots & \ddots & \vdots \\ I_N & \cdots & A_M \end{pmatrix} \quad (7)$$

Note that, unlike the case of multiplex networks, the blocks in the supra-adjacency matrix of a general multilayer networks are not necessarily square matrices.

Now a supra-Laplacian matrix can be derived from a supra-adjacency matrix in a manner that is analogous to the way that one derives a Laplacian matrix from the adjacency matrix of a monoplex graph. For example, the combinatorial supra-Laplacian matrix is $\mathcal{L}_\mathcal{M} = D_\mathcal{M} - \mathcal{A}_\mathcal{M}$, where $D_\mathcal{M}$ is the diagonal supra-matrix that has node-layer strengths (i.e. weighted degrees) along the diagonal and $\mathcal{A}_\mathcal{M}$ denotes the supra-adjacency matrix that corresponds to the graph $G_\mathcal{M}$. Hence, each diagonal entry of the supra-Laplacian $\mathcal{L}_\mathcal{M}$ consists of the sum of the corresponding row in the supra-adjacency matrix $\mathcal{A}_\mathcal{M}$, and each non-diagonal element of $\mathcal{L}_\mathcal{M}$ consists of the corresponding element of $\mathcal{A}_\mathcal{M}$ multiplied by -1. The eigenvalues and eigenvectors of this supra-Laplacian are important indicators of several structural features of the corresponding network, and they also give crucial insights into dynamical processes that evolve on top of it [69], [70]. The second smallest eigenvalue and the eigenvector associated to it, which are sometimes called (respectively) the "algebraic connectivity" and "Fiedler vector" of the corresponding network, are very important diagnostics for the structure of a network [71], [72].

Spectra of multilayer networks

Supra-Laplacian matrix

Generally, the Laplacian matrix of a graph with adjacency matrix A, or simply the
Laplacian, is given by:

$$L = D - A \quad (8)$$

where $D = diag(k_1, k_2 \cdots)$ is the degree (or identity) matrix.

Thus, it is natural to define the *supra-Laplacian* matrix of a Multiplex network as the Laplacian of its supra-graph:

$$\mathcal{L} = \mathcal{D} - \mathcal{A} \quad (9)$$

where $\mathcal{D} = diag(K_1, K_2, \dots)$ is the degree matrix. Besides the Laplacian for each layer graph G_α can be defined as

$$L^\alpha = D^\alpha - A^\alpha \quad (10)$$

and the Laplacian of the coupling graph:

$$\boldsymbol{\mathcal{L}_C} = \boldsymbol{\Delta} - \boldsymbol{\mathcal{C}} \quad (11)$$

where $\boldsymbol{\Delta} = \boldsymbol{diag(c_1, c_2, \dots, c_N)}$ is the coupling-degree matrix.

By definition:

$$\boldsymbol{\mathcal{L}} = \underbrace{\bigoplus}_{\alpha} \boldsymbol{\mathcal{L}^\alpha} + \boldsymbol{\mathcal{L}_C} \quad (12)$$

Eq. (12) takes a very simple form in the case of a node-aligned multiples, i.e.:

$$\boldsymbol{\mathcal{L}} = \underbrace{\bigoplus}_{\alpha} . (\boldsymbol{L^\alpha + cI_N}) - \boldsymbol{K_m} \otimes \boldsymbol{I_n} \quad (13)$$

where $\boldsymbol{K_m}$ is the adjacency matrix of a complete graph of m nodes, $\boldsymbol{I_n}$ is the $k \times k$ identity matrix and $\boldsymbol{c_i = c}, \forall \boldsymbol{i \in N}$ is the coupling degree of a node-layer pair.

Individual layer spectra

In [68] using perturbation theory, is showed that the largest eigenvalue of the multiplex network is equal to the one of the adjacency matrix of the dominant layer of the system at a first order approximation. Similarly, in [73] is studied thee eigenvalues of the Laplacian in the context of multilayer networks. They show that the eigenvalues of the quotient (a coarsening of the original network) are interlaced with the eigenvalues of its parent network. This fact has deep consequences as, for example, the relaxation time on a multiplex network is at most the one of the aggregated network, which in turn can result in faster diffusion processes on multiplex networks than in their aggregated counterparts.

Combined spectrum

The eigenvalues and eigenvectors of this supra-Laplacian are important indicators of several structural features of the corresponding network, and they also give crucial insights into dynamical processes that evolve on top of it [15], [74]. The second smallest eigenvalue and the eigenvector associated to it, which are called (respectively) the "algebraic connectivity" and "Fiedler vector" of the corresponding network, are very important diagnostics for the structure of a network.

For example, the algebraic connectivity of a multilayer network with categorical couplings16has

two distinct regimes [3], [15], [48] when examined as a function of the relative strengths of the inter-layer edges and the intra-layer edges. Additionally, there is a discontinuous (i.e. first-order) phase transition — a so-called "structural transition" — between the two regimes. In one regime, the algebraic connectivity is independent of the intra-layer adjacency structure, so it is determined by the inter-layer edges. In the other, the algebraic connectivity of the multilayer network is bounded above by a constant multiplied by the algebraic connectivity of the unweighted superposition of the layers. Combinatorial supra-Laplacian matrices have also been used to study a diffusion process on multiplex networks [1].

Eigenvalues and eigenvectors of tensors

The Z-eigenvalue problem for tensors involves finding nontrivial solutions of in homogeneous polynomial systems in several variables. The Z-spectrum of A, denoted $\mathcal{Z}(A)$ is defined to be the set of all Z-eigenvalues of $\mathcal{A}$. It is proven in [75], that for a symmetric tensor $\mathcal{A}$, the set of *E*-eigenvalues of $\mathcal{A}$ is nonempty and finite. We therefore define the Z-spectral radius of a symmetric tensor A, denoted $\varrho(\mathcal{A})$, to be $\varrho(\mathcal{A}) := max\{|\lambda| \mid \lambda \in \mathcal{Z}(\mathcal{A})\}$.

Conclusion and future works

Based on the mathematical representation of multilayer networks, much better multilayer models can be created and analysed. Moreover complex systems can be modelled in different aspects and also can be related to each other based not only on typical graph properties, but also based on the spectral properties. The generalized model of multilayer networks using tensors is good in the modelling complex systems, but on the other hand spectrum is very difficult to compute. The supra-Laplacian matrix is a good way of reduction the tensor to a matrix and for it to be done spectral analysis.

Future works in this field include better eigenvalue and eigenvector decomposition of tensors, so the information for the systems that is lost in flattening the tensors to supra-Laplacian and supra-adjacency matrices is preserved, while keeping the computation to bearable degrees.

This paper is reported in the XXVI National conference with international participation TELECOM, Sofia, Bulgaria, 2018.

REFERENCES

[1] M. De Domenico, A. Solé-Ribalta, E. Cozzo, *et al.*, "Mathematical formulation of multilayer networks," *Physical Review X*, vol. 3, no. 4, p. 041022, 2013.

[2] A. Jamakovic, "Characterization of complex networks: application to robustness analysis," Next Generation Infrastructures Foundation, 2008.

[3] G. D'Agostino and A. Scala, *Networks of networks: the last frontier of complexity*. Springer, 2014.

[4] N. R. Council and others, *Network science*. National Academies Press, 2006.

[5] S. Hanna, "Representation and generation of plans using graph spectra," 2007.

[6] I. G. Branca, "Architectures and Technologies for Quality of Service Provisioning in Next Generation Networks," no. March, 2012.

[7] A. Bigdeli, "Robustness and Vulnerability Design for Autonomic Management," 2012.

[8] S. Boccaletti, V. Latora, Y. Moreno, *et al.*, "Complex networks: Structure and dynamics," *Phys. Rep.*, vol. 424, no. 4–5, pp. 175–308, 2006.

[9] S. Boccaletti, G. Bianconi, R. Criado, *et al.*, "The structure and dynamics of multilayer networks," *Phys. Rep.*, vol. 544, no. 1, pp. 1–122, 2014.

[10] G. Bianconi, "Statistical mechanics of multiplex networks: Entropy and overlap," *Physical Review E*, vol. 87, no. 6, p. 062806, 2013.

[11] G. Chen, "Introduction to Complex networks," *IWCSN 2009*. 2009.

[12] R. J. Mondragon, J. Iacovacci, and G. Bianconi, "Multilink communities of multiplex networks," *PLoS One*, vol. 13, no. 3, p. e0193821, 2018.

[13] A. Cardillo, M. Zanin, J. Gómez-Gardenes, *et al.*, "Modeling the multi-layer nature of the European Air Transport Network: Resilience and passengers re-scheduling under random failures," *The European Physical Journal Special Topics*, vol. 215, no. 1, pp. 23–33, 2013.

[14] J. Gao, S. V. Buldyrev, S. Havlin, and H. E. Stanley, "Robustness of a network of networks," *Physical Review Letters*, vol. 107, no. 19, p. 195701, 2011.

[15] M. Kivelä, A. Arenas, M. Barthelemy, *et al.*, "Multilayer networks," *Journal of complex networks*, vol. 2, no. 3, pp. 203–271, 2014.

[16] A. Aleta and Y. Moreno, "Multilayer Networks in a Nutshell," *arXiv preprint arXiv:1804.03488*, 2018.

[17] M. De Domenico, A. Solé-Ribalta, S. Gómez, and A. Arenas, "Navigability of interconnected networks under random failures," *Proceedings of the National Academy of Sciences*, vol. 111, no. 23, pp. 8351–8356, 2014.

[18] O. Ya\ugan and V. Gligor, "Analysis of complex contagions in random multiplex networks," *Physical Review E*, vol. 86, no. 3, p. 036103, 2012.

[19] C. D. Brummitt, K.-M. Lee, and K.-I. Goh, "Multiplexity-facilitated cascades in networks," *Physical Review E*, vol. 85, no. 4, p. 045102, 2012.

[20] V. Nicosia, G. Bianconi, V. Latora, and M. Barthelemy, "Growing multiplex networks," *Physical review letters*, vol. 111, no. 5, p. 058701, 2013.

[21] D. Cellai, E. López, J. Zhou, *et al.*, "Percolation in multiplex networks with overlap," *Physical Review E*, vol. 88, no. 5, p. 052811, 2013.

[22] F. Battiston, V. Nicosia, and V. Latora, "Structural measures for multiplex networks," *Physical Review E*, vol. 89, no. 3, p. 032804, 2014.

[23] B. Min and K. Goh, "Layer-crossing overhead and information spreading in multiplex social networks," *seed*, vol. 21, no. T22, p. T12, 2013.

[24] B. Min, S. Do Yi, K.-M. Lee, and K.-I. Goh, "Network robustness of multiplex networks with interlayer degree correlations," *Physical Review E*, vol. 89, no. 4, p. 042811, 2014.

[25] P. Pattison and S. Wasserman, "Logit models and logistic regressions for social networks: II. Multivariate relations.," *Br J Math Stat Psychol*, vol. 52 (Pt 2), pp. 169–93, 1999.

[26] M. Barigozzi, G. Fagiolo, and D. Garlaschelli, "Multinetwork of international trade: a commodity-specific analysis.," *Phys Rev E Stat Nonlin Soft Matter Phys*, vol. 81, no. 4 Pt 2, p. 046104, 2010.

[27] M. Barigozzi, G. Fagiolo, and G. Mangioni, "Identifying the community structure of the international-trade multi-network," *Physica A: statistical mechanics and its applications*, vol. 390, no. 11, pp. 2051–2066, 2011.

[28] D. Cai, Z. Shao, X. He, *et al.*, "Community mining from multi-relational networks," in *European Conference on Principles of Data Mining and Knowledge Discovery*, 2005, pp. 445–452.

[29] A. Harrer and A. Schmidt, "An approach for the blockmodeling in multi-relational networks," in *Advances in Social Networks Analysis and Mining (ASONAM), 2012 IEEE/ACM International Conference on*, 2012, pp. 591–598.

[30] V. Ströele, G. Zimbrão, and J. M. Souza, "Modeling, mining and analysis of multi-relational scientific social network," *Journal of Universal Computer Science*, vol. 18, no. 8, pp. 1048–1068, 2012.

[31] M. Kurant, P. Thiran, and P. Hagmann, "Error and attack tolerance of layered complex networks," *Physical Review E*, vol. 76, no. 2, p. 026103, 2007.

[32] P. J. Mucha and M. A. Porter, "Communities in multislice voting networks," *Chaos: An Interdisciplinary Journal of Nonlinear Science*, vol. 20, no. 4, p. 041108, 2010.

[33] V. Carchiolo, A. Longheu, M. Malgeri, and G. Mangioni, "Communities unfolding in multislice networks," in *Complex Networks*, Springer, 2011, pp. 187–195.

[34] D. S. Bassett, M. A. Porter, N. F. Wymbs, *et al.*, "Robust detection of dynamic community structure in networks," *Chaos: An Interdisciplinary Journal of Nonlinear Science*, vol. 23, no. 1, p. 013142, 2013.

[35] J. Gómez-Gardenes, I. Reinares, A. Arenas, and L. M. Flor'\ia, "Evolution of cooperation in multiplex networks," *Sci. Rep.*, vol. 2, p. 620, 2012.

[36] D. Irving and F. Sorrentino, "Synchronization of dynamical hypernetworks: Dimensionality reduction through simultaneous block-diagonalization of matrices," *Physical Review E*, vol. 86, no. 5, p. 056102, 2012.

[37] F. Sorrentino, "Synchronization of hypernetworks of coupled dynamical systems," *New Journal of Physics*, vol. 14, no. 3, p. 033035, 2012.

[38] S. Funk and V. A. Jansen, "Interacting epidemics on overlay networks," *Physical Review E*, vol. 81, no. 3, p. 036118, 2010.

[39] V. Marceau, P.-A. Noël, L. Hébert-Dufresne, *et al.*, "Modeling the dynamical interaction between epidemics on overlay networks," *Physical Review E*, vol. 84, no. 2, p. 026105, 2011.

[40] X. Wei, N. Valler, B. A. Prakash, *et al.*, "Competing memes propagation on networks: a case study of composite networks," *ACM SIGCOMM Computer Communication Review*, vol. 42, no. 5, pp. 5–12, 2012.

[41] P. Wang, G. Robins, P. Pattison, and E. Lazega, "Exponential random graph models for multilevel networks," *Soc. Networks*, vol. 35, no. 1, pp. 96–115, 2013.

[42] E. Lazega, M.-T. Jourda, L. Mounier, and R. Stofer, "Catching up with big fish in the big pond? Multi-level network analysis through linked design," *Soc. Networks*, vol. 30, no. 2, pp. 159–176, 2008.

[43] M. Rocklin and A. Pinar, "On clustering on graphs with multiple edge types," *Internet Mathematics*, vol. 9, no. 1, pp. 82–112, 2013.

[44] D. Zhou, S. A. Orshanskiy, H. Zha, and C. L. Giles, "Co-ranking authors and documents in a heterogeneous network," in *Data Mining, 2007. ICDM 2007. Seventh IEEE International Conference on*, 2007, pp. 739–744.

[45] A. Vazquez, "Spreading dynamics on heterogeneous populations: multitype network approach," *Physical Review E*, vol. 74, no. 6, p. 066114, 2006.

[46] J. Hindes, S. Singh, C. R. Myers, and D. J. Schneider, "Epidemic fronts in complex networks with metapopulation structure," *Physical Review E*, vol. 88, no. 1, p. 012809, 2013.

[47] A. Allard, P.-A. Noël, L. J. Dubé, and B. Pourbohloul, "Heterogeneous bond percolation on multitype networks with an application to epidemic dynamics," *Physical Review E*, vol. 79, no. 3, p. 036113, 2009.

[48] M. Dickison, S. Havlin, and H. E. Stanley, "Epidemics on interconnected networks," *Physical Review E*, vol. 85, no. 6, p. 066109, 2012.

[49] V. Louzada, N. Araújo, J. Andrade Jr, and H. Herrmann, "Breathing synchronization in interconnected networks," *Sci. Rep.*, vol. 3, p. 3289, 2013.

[50] S. V. Buldyrev, R. Parshani, G. Paul, *et al.*, "Catastrophic cascade of failures in interdependent networks," *Nature*, vol. 464, no. 7291, p. 1025, 2010.

[51] R. Parshani, S. V. Buldyrev, and S. Havlin, "Interdependent networks: Reducing the coupling strength leads to a change from a first to second order percolation transition," *Physical review letters*, vol. 105, no. 4, p. 048701, 2010.

[52] J. Martin-Hernandez, H. Wang, P. Van Mieghem, and G. D'Agostino, "On synchronization of interdependent networks," *arXiv preprint arXiv:1304.4731*, 2013.

[53] C. D. Brummitt, R. M. D'Souza, and E. A. Leicht, "Suppressing cascades of load in interdependent networks," *Proceedings of the National Academy of Sciences*, vol. 109, no. 12, pp. E680–E689, 2012.

[54] A. Bashan, Y. Berezin, S. V. Buldyrev, and S. Havlin, "The extreme vulnerability of interdependent spatially embedded networks," *Nature Physics*, vol. 9, no. 10, p. 667, 2013.

[55] G. Baxter, S. Dorogovtsev, A. Goltsev, and J. Mendes, "Avalanche collapse of interdependent networks," *Physical review letters*, vol. 109, no. 24, p. 248701, 2012.

[56] D. Zhou, J. Gao, H. E. Stanley, and S. Havlin, "Percolation of partially interdependent scale-free networks," *Physical Review E*, vol. 87, no. 5, p. 052812, 2013.

[57] O. Yagan, D. Qian, J. Zhang, and D. Cochran, "Conjoining speeds up information diffusion in overlaying social-physical networks," *IEEE Journal on Selected Areas in Communications*, vol. 31, no. 6, pp. 1038–1048, 2013.

[58] X.-L. Xu, Y.-Q. Qu, S. Guan, *et al.*, "Interconnecting bilayer networks," *EPL (Europhysics Letters)*, vol. 93, no. 6, p. 68002, 2011.

[59] E. Leicht and R. M. D'Souza, "Percolation on interacting networks," *arXiv preprint arXiv:0907.0894*, 2009.

[60] J. F. Donges, H. C. Schultz, N. Marwan, *et al.*, "Investigating the topology of interacting networks," *The European Physical Journal B*, vol. 84, no. 4, pp. 635–651, 2011.

[61] Y. Sun and J. Han, "Mining heterogeneous information networks: a structural analysis approach," *Acm Sigkdd Explorations Newsletter*, vol. 14, no. 2, pp. 20–28, 2013.

[62] D. Davis, R. Lichtenwalter, and N. V. Chawla, "Multi-relational link prediction in heterogeneous information networks," in *Advances in Social Networks Analysis and Mining (ASONAM), 2011 International Conference on*, 2011, pp. 281–288.

[63] Y. Sun, J. Han, X. Yan, *et al.*, "Pathsim: Meta path-based top-k similarity search in heterogeneous information networks," *Proceedings of the VLDB Endowment*, vol. 4, no. 11, pp. 992–1003, 2011.

[64] J. Han, Y. Sun, X. Yan, and P. S. Yu, "Mining heterogeneous information networks," in *Tutorial at the 2010 ACM SIGKDD Conf. on Knowledge Discovery and Data Mining (KDD'10), Washington, DC*, 2010.

[65] K. M. Carley and V. Hill, "Structural change and learning within organizations," *Dynamics of organizations: Computational modeling and organizational theories*, pp. 63–92, 2001.

[66] K. M. Carley, J. Diesner, J. Reminga, and M. Tsvetovat, "Toward an interoperable dynamic network analysis toolkit," *Decision Support Systems*, vol. 43, no. 4, pp. 1324–1347, 2007.

[67] M. Tsvetovat, J. Reminga, and K. Carley, "DyNetML: Interchange format for rich social network data," 2004.

[68] E. Cozzo, G. F. de Arruda, F. A. Rodrigues, and Y. Moreno, "Multilayer networks: metrics and spectral properties," in *Interconnected Networks*, Springer, 2016, pp. 17–35.

[69] A. E. Brouwer and W. H. Haemers, *Spectra of graphs*. Springer Science & Business Media, 2011, p. 255.

[70] D. Cvetković and S. Simić, "Graph spectra in computer science," *Linear Algebra Appl.*, vol. 434, no. 6, pp. 1545–1562, 2011.

[71] M. Fiedler, "Algebraic Connectivity of Graphs," *Czechoslovak Mathematical Journal*, vol. 23, no. 2, pp. 298–305, 1973.

[72] D. A. Spielman, "Spectral Graph Theory and its Applications Daniel A . Spielman," 2007.

[73] A. Tizghadam and A. Leon-Garcia, "Betweenness centrality and resistance distance in communication networks," *IEEE network*, vol. 24, no. 6, 2010.

[74] A. Barrat, M. Barthelemy, and A. Vespignani, Dynamical processes on complex networks, vol. 1. Cambridge University Press Cambridge, 2008.

[75] D. Cartwright and B. Sturmfels, "The number of eigenvalues of a tensor," *Linear Algebra Appl.*, vol. 438, no. 2, pp. 942–952, 2013.

Dipl. eng. Mircho J. Mirchev *– received the M.Sc. degree in telecommunications in 2006 and currently is a Ph.D. candidate in the Technical University of Sofia. Since 2001, he has been with Smartcom-Bulgaria AD, Sofia, where he is currently a Project Manager.*

tel.:+359887412248 e-mail: mircho@mirchev.eu

Received on: 20.04.2019

Analysis of non-preemptive priority single-server queueing systems with peaked traffic flows

Seferin Mirtchev, Rossitza Goleva, Dimitar Atamian, Ivan Ganchev

In this article, single-server priority queueing systems with a peaked arrival process, generally distributed service time and infinite waiting position is analyzed by using the Polya distribution to describe the peaked traffic flow. The model of such queueing systems is obtained using a generalized Pollaczek-Khinchin formula. In the article, the dependence of the mean waiting time in the systems with four priority classes and non-preemptive priority from the offered traffic with different values of the peakedness coefficient of the arrival process and different values of the variation coefficient of service process is presented. It is shown that the performance of such single-server priority queueing systems varies vastly depending on the peakedness of the arrival and service processes.

Keywords – non-preemptive priority, single-server queue, peaked arrival process.

Анализ на едноканални системи с чакане и относителен приоритет при неравномерен входящ поток (Сеферин Мирчев, Росица Голева, Димитър Атамян, Иван Ганчев). *В статията се анализират едноканални телетрафични системи с чакане и относителен приоритет при неравномерен процес на постъпване на заявките, произволно разпределение на времето на тяхното обслужване и безкрайна опашка за чакане, като се използва разпределението на Пойа, за да се опише неравномерният трафичен поток. Моделът за анализ на тази телетрафична система е получен чрез използването на обобщената формула на Pollaczek-Khinchin. В статията е представена зависимостта на средното време за чакане в система с четири приоритетни класа с относителен приоритет от постъпващия трафик при различен коефициент на неравномерност на постъпващия процес и различен коефициент на вариации на процеса на обслужване. Показано е, че характеристиките на тази едноканална телетрафична система с чакане и относителен приоритет се променят значително в зависимост от неравномерностите на процесите на постъпване и на обслужване.*

1. Introduction

The queueing systems represent an interesting area that is widely exploited in a number of real-life situations. In order to provide different service levels to different users, the queueing systems often utilize priority mechanisms. Using priorities can easily provide the distinction between these different levels of service [1]. For example in telecommunications networks, priority classes can be employed. The priority is usually identified in an appropriate header field of the transmitted protocol units, e.g. in the DiffServ Code Point (DSCP) field in IP packets, or the first three bits in the Ethernet frame's Q-tag field, or the Cell Loss Priority (CLP) flag in the ATM cells, the channel priority mechanisms in the IEEE 802.15.4 standard, and etc. Priority management is also widely used in some production processes, transport control, healthcare, and population protection. The new telecommunications technologies, such as Bluetooth, ZigBee, etc., used in the Internet of Things (IoT) area, allow the interconnection of a large number of devices that are seen as uneven sources of traffic. The variety of applications using IoT requires servicing with different priorities of the individual tasks in a single system and can be determined by the Service Level Agreement (SLA) between users and service providers [2].

Priority service issues arise in many practical situations in telecommunications networks. Practically in packet switched networks, it is important to define the strategy of sharing the communications resources provided to a large number of different traffic flows with different service requirements. Priority mechanisms can be either static or dynamic. The serving devices can handle tasks using either preemptive or non-preemptive priority disciplines. Regarding the use of priority services in data transmission networks, in-depth research is being carried out to improve the average delay for short

messages at the expense of the long ones or to meet the stringent requirements of traffic flows that are sensitive to delays or losses [3].

One of the fundamental dependencies in the queueing theory defines the average queue length and the average waiting time in a single-server queueing system M/G/1 with a Poisson arrival process, generally distributed service time and an infinite queue is given by the Pollaczek-Khinchin formula [4]. Many authors offer a different generalization of Pollaczek-Khinchin's formula, for instance, a M/G/1 queue with an occasional service failure [5], throughput analysis of input-buffered switches [6], two-phase Markov modulated processes [7], point-to-point communications networks [8], etc.

The Polya/G/1 teletraffic system is a generalization of the M/G/1 queue, with a Polya arrival flow. This generalization leads to a significant increase in the complexity of the analysis.

Various models based on the M/G/1 teletraffic system have been suggested, such as a state dependent one[9], processor sharing disciplines [10], rest periods [11], etc. In [12], a new teletraffic model of a multichannel waiting system with a peaked arrival flow, described by Polya distribution, and constant service time is offered. All these models make it possible to take into account the influence of the peaked arrival flows in IP networks more accurately.

In [13], a single-channel Polya/D/1 teletraffic system is proposed, and all its interesting features and parameters are evaluated. The idea is based on an analytical continuation to the classical single channel M/D/1 system, using a Polya distributed arrival process.

In [14], a M/G/1 teletraffic system is investigated, where all sources are the same, but they are assigned a randomly selected priority level before requests are entering the system. A transformation of Laplace-Stieltjes is used to determine the parameters of the assigned priorities. It is shown that the model determines the average waiting time, limited by the mean time using of the two service mechanisms - FIFO and LIFO. Lastly, it is pointed out that this new approach can increase the efficiency of the server when a new source appears.

In [15], an optimal strategy behavior of high priority users for a M/G/1 queue with two priority classes is presented. When a highly priority request is received, it can be decided whether it should be served with preemptive priority or wait for the completion of a low-priority request that is currently served. Optimal strategies and numerical results are given.

A priority M/G/1 model with accumulation is analyzed in [16]. It allows to control the waiting time for each class of requests. The authors present an in-depth analysis of a dynamic queue behavior with a preemptive priority for two disciplines serving low-priority requests – repeating the same request and repeating the next one.

In this article, the model of a single-channel teletraffic system with a priority service, a peaked traffic flow, a generally distributed service time and an infinite queue is described [17]. The model is developed based on a generalization of the classical M/G/1 model and of the Pollaczek-Khinchin formula [18]. The analysis of this non-preemptive priority single-server queueing system, based on the numerical results, is presented.

2. Arrival process with Polya distribution

The peaked arrival processes is described by Polya distribution with two parameters – intensity λ and peakedness β [19]. The probability $P_i(t)$ for the arrival of i requests with in the time interval t is determined by the following formula:

$$(1) \quad P_i(t) = \left(\frac{\lambda t}{1+\beta\lambda t} \right)^i \frac{1(1+\beta) \ \ldots \ [1+(i-1)\beta]}{i!} P_o(t) \ ,$$

where $P_o(t) = (1+\beta\lambda t)^{-\frac{1}{\beta}}$.

The mean value $M(t)$ and the variance $V(t)$ of the number of arrivals for the time interval t are respectively:

$$(2) \quad M(t) = \lambda t; \quad V(t) = \lambda t(1+\beta\lambda t) \ .$$

The coefficient of peakedness z of the number of arrivals is:

$$(3) \quad z = \frac{V(t)}{M(t)} = 1+\beta\lambda t > 1 \ .$$

3. Pollaczek-Khinchin's generalized formula for Polya/G/1 system

The M/G/1 teletraffic system model is one of the most frequently studied models in the telecommunications and computer networks. The model of a teletraffic Polya/G/1 system is a generalization of the above model. The Polya/G/1 system has a peaked input process, described by the Polya distribution, with an arrival intensity – λ and a coefficient of peakedness – z, generally distributed service time (independent of the input process) with a mean value – τ and a coefficient of variation – C_t. The offered traffic $A = \lambda\tau$ must be less than 1 Erl in order for the teletraffic system to be stationary.

The Pollaczek-Khinchin formula for a Polya/G/1 teletraffic system is obtained using the Kendal recursion [18]. The mean waiting time for a single-channel system with a peaked traffic flow (i.e. the time a request has to wait in the queue for a service) is:

$$W_q = \frac{\tau(A+z-1)\left(C_t^2+1\right)}{2(1-A)}. \quad (4)$$

As shown in [1], the mean value of the residual time t_R for servicing a request at a random point of time during its service (i.e. the time left to finish servicing the current request) is:

$$E(t_R) = \frac{\tau\left(C_t^2+1\right)}{2}. \quad (5)$$

The average residual service time R of a server at a random point of represents the average release time of the server, if occupied at the moment. The probability of the server having a request currently being served is equal to the offered traffic A (as in a single-channel system with an unlimited queue the probability of the system not being occupied is $1 - A$). Therefore, the fraction of the average residual service time R taken by the mean residual time t_R of a request is determined by the offered traffic. Then the average residual service time R of the server, which may be busy or free at a random point of time, becomes:

$$R = \frac{A\tau\left(C_t^2+1\right)}{2}. \quad (6)$$

The average waiting time W_q for any request can be divided into two parts:

1. The average residual service time R of the server;
2. The average service time of the previous requests that have already arrived and are waiting in the queue:

$$W_q = R + \tau L'_q, \quad (7)$$

where L'_q is the average number of requests waiting in the queue (i.e. the average queue length) when new request arrives.

With conversion and substitution, one can get:

$$L'_q = \frac{(A^2+z-1)\left(C_t^2+1\right)}{2(1-A)}. \quad (8)$$

Using (4), the average queue length at the arrival of a new request can be expressed by the average waiting time, i.e.:

$$L'_q = \frac{(A^2+z-1)}{\tau(A+z-1)} W_q. \quad (9)$$

From (7) and (9) one can get the following:

$$W_q = R + kW_q; \quad (10)$$

$$W_q = \frac{R}{1-k}, \quad (11)$$

where: $k = (A^2+z-1)/(A+z-1)$.

4. Teletraffic Polya/G/1 system with non-preemptive priority

In the telecommunications networks, the arriving requests are often divided into N priority classes, e.g. by postulating that requests of class p have a higher priority than requests of class $p+1$. In a single-channel system with a non-preemptive priority, a newly arriving request will have to wait until the server becomes 'free', even if a lower priority request is being currently served. Then the new request will also have to wait until all higher priority requests are served followed by the earlier arrived requests of the same priority class.

In a Polya/G/1 single-channel waiting system with a non-preemptive priority [18], the requests of class i arrive with an arrival intensity λ_i, a coefficient of peakedness z_i, and an average service time τ_i. The offered traffic is $A_i = \lambda_i \tau_i$. The coefficient of variation of the service time is C_{ti}. In the present study the FIFO discipline is used for serving the requests within each of the priority classes, Fig.1.

Instead of considering the individual arriving processes, one can use the general arrival process described by the Polya distribution with the following intensity:

$$\lambda = \sum_{i=1}^{N} \lambda_i. \quad (12)$$

Then the obtained coefficient of peakedness of the general arrival process becomes a weighted sum of the coefficients of peakedness of the individual classes (combined in parallel):

$$z = \sum_{i=1}^{N} \frac{\lambda_i}{\lambda} z_i. \quad (13)$$

The total average service time of the requests from the general arrival process is:

$$\tau = \sum_{i=1}^{N} \frac{\lambda_i}{\lambda} \tau_i. \quad (14)$$

The coefficient of peakedness of the requests from the general arrival process is:

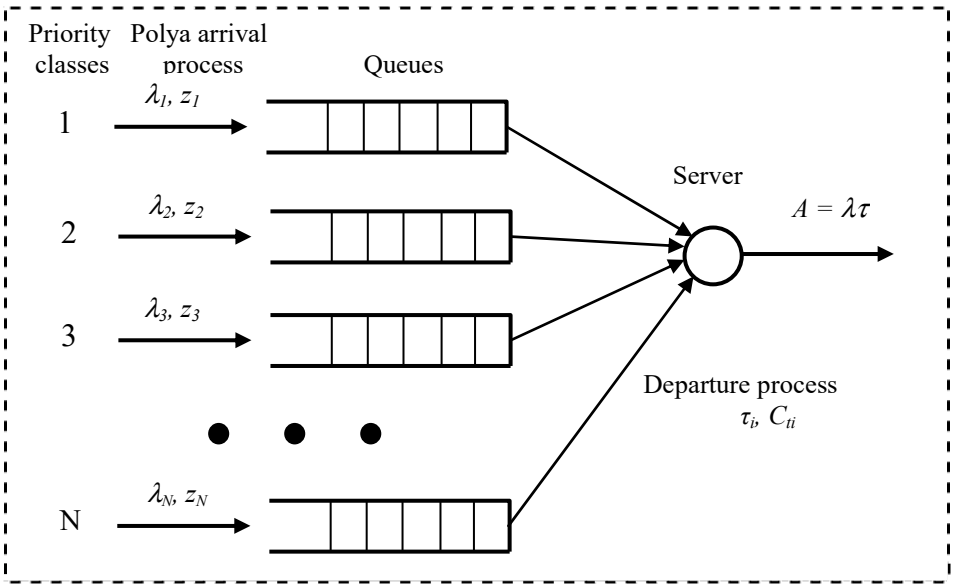


Fig.1. The Polya/G/1 single-server teletraffic system priority model.

(15) $$C_t = \sum_{i=1}^{N} \frac{\lambda_i}{\lambda} C_{ti}.$$

The total offered traffic is:

(16) $$A = \lambda\tau = \sum_{i=1}^{N} A_i = \sum_{i=1}^{N} \lambda_i \tau_i .$$

The average residual service time of the server at a random point of time is:

(17) $$R = \sum_{i=1}^{N} R_i = \sum_{i=1}^{N} \frac{A_i \tau \left(C_{ti}^2 + 1\right)}{2}.$$

The mean waiting time W_{q1} for the highest priority requests (i.e. class 1) is the sum of the average residual service time R of the server and the average time for waiting the server to finish servicing other, earlier arrived requests of the same class that are already in the queue ($\tau_1 L'_{q1}$):

(18) $$W_{q1} = R + \tau_1 L'_{q1} = R + k_1 W_{q1};$$

(19) $$W_{q1} = \frac{R}{1 - k_1},$$

where $k_1 = (A_1^2 + z_1 - 1) / (A_1 + z_1 - 1)$.

The mean waiting time W_{q2} for the requests of priority class 2 is the sum of the following two components:

1. The mean waiting time due to servicing the previously arrived requests of class 1 and 2 that are already in the queue. This case is similar to the case of non-priority requests entering the system with an intensity equal to the sum of intensities of the two classes – 1 and 2. Therefore, this component may be represented by formula (11);

2. The mean service time for all new requests of class 1, arriving while there are still requests of class 2 waiting in the queue, i.e. $\lambda_1 \tau_1 W q_2$.

By combining these two components, one can get the following expressions:

(20) $$W_{q2} = \frac{R}{1 - k'_2} + \lambda_1 \tau_1 W_{q2};$$

(21) $$W_{q2} = \frac{R}{(1 - k'_2)(1 - A_1)},$$

where

(22) $$k'_2 = \frac{\left(A_1 + A_2\right)^2 + \frac{z_1\lambda_1 + z_2\lambda_2}{\lambda_1 + \lambda_2} - 1}{A_1 + A_2 + \frac{z_1\lambda_1 + z_2\lambda_2}{\lambda_1 + \lambda_2} - 1}.$$

Similarly, the mean waiting time W_{q3} for the requests of priority class 3 is:

(23) $$W_{q3} = \frac{R}{1 - k'_3} + (\lambda_1\tau_1 + \lambda_2\tau_2) W_{q3}.$$

where k_3' is obtained by analogy with (20) for the total offered traffic from all the three priority classes and the weighted coefficients of peakedness of the arrival streams.

From (23) after conversion, one can get the following formula:

$$W_{q3} = \frac{R}{(1-k_3')(1-A_1-A_2)}. \quad (24)$$

With a corresponding generalization, the mean waiting time for class p with a non-preemptive priority can be represented as:

$$W_{qp} = \frac{R}{1-k_p'} + \sum_{i=1}^{p-1} A_i W_{qp}. \quad (25)$$

$$W_{qp} = \frac{R}{(1-k_p')(1-\sum_{i=1}^{p-1} A_i)}, \quad (26)$$

where

$$k_p' = \frac{\left(\sum_{i=1}^{p} A_i\right)^2 + \sum_{i=1}^{p} \frac{z_i \lambda_i}{\sum_{j=1}^{p} \lambda_j} - 1}{\sum_{i=1}^{p} A_i + \sum_{i=1}^{p} \frac{z_i \lambda_i}{\sum_{j=1}^{p} \lambda_j} - 1}. \quad (27)$$

5. Numeric Results

Using a computer program and the formulas from the previous sections, results for the mean waiting time have been obtained for a given value of the coefficient of peakedness – z, the coefficient of variation of the service time – C_t and the offered traffic – A is obtained.

Figure 2 shows the mean waiting time W_{qi} in the Polya/G/1 teletraffic model with four priority classes and a non-preemptive priority as a function of the offered traffic A. The results have been obtained for different values of the coefficient of peakedness z (i.e. 1.0, 1.2 and 1.4) and zero coefficient of the variation of the service time, i.e. C_t=0.

The offered traffic to each priority class has the same value and the service time in each class equals the others, i.e. τ_i=*0.001 s*. For comparison, the figure also shows the mean waiting time W_q for the same teletraffic system but without priorities. It can be seen that the increase in the number of arrival requests leads to a significant increase in the mean waiting time of the low-priority requests. In addition, the mean waiting time for class 4 is several times higher than that in the non-priority system.

Figure 3 depicts the mean waiting time W_{qi} for the same values of the coefficient of peakedness z as in Figure 2 but with a coefficient of variation C_t equal to 2. It can be seen that the peakedness of the service process significantly increases the mean waiting time for both low- and high-priority classes. The results show that the mean waiting time for low-priority classes increases steeply when the offered traffic approaches 1 Erl and has a much higher value in the case of greater peakedness of the incoming and servicing processes.

6. Conclusion

This article has described a model for analyzing a single-channel teletraffic system with a peaked arrival traffic flow, a generally distributed service time, an infinity queue, and a non-preemptive priority, based on the generalized Pollaczek-Khinchin's formula, a peaked arrival traffic flow described by the Polya distribution, and the classical queueing M/G/1 system with priorities.

It has been shown that the peakedness of arrival requests and the variation in service time lead to a significant increasing of delays in the single-channel system and, consequently, to a significant increasing of the queue length for low-priority traffic classes.

The analysis of teletraffic systems with peaked arrival and service processes provides guidelines for traffic flow analysis that are typical of modern telecommunication networks and systems, which is an important element of their design.

The presented results make it possible to assess the characteristics of priority queueing systems in fixed and mobile networks with traffic classification, in specific applications in cloud technologies and in point-to-point communications. Other possible applications include e-health, disaster and emergency protection, emergency calls, highly reliable robots, etc.

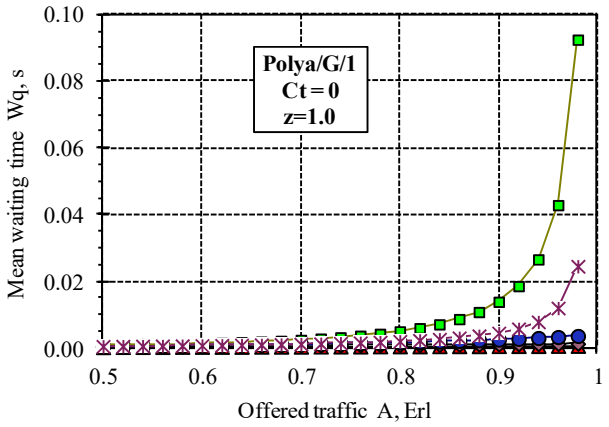
A version of this article (in Bulgarian) was reported at the XXVI National Conference TELECOM (with international participation), Sofia, Bulgaria, 2018.

REFERENCES

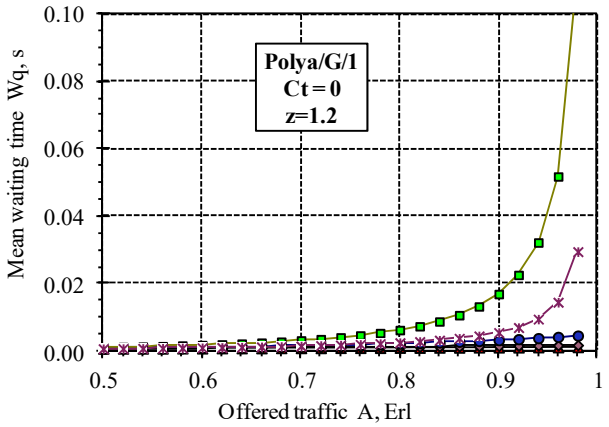
[1] Iversen, V. *Teletraffic engineering and network planning.* DTU Fotonik, 2015, http://orbit.dtu.dk/files/118473571/Teletraffic_34342_V_B_Iversen_2015.pdf

[2] Goleva, R., Garcia, N., Mavromoustakis, C.X., Dobre, C., Mastorakis, G., Stainov, R., Chorbev, I., Trajkovik, V., AAL and ELE Platform Architecture, in Dobre, C., Mavromoustakis, C.X., Garcia, N., Goleva, R., Mastorakis, G. (Editors), Ambient Assisted Living and Enhanced Living Environments: Principles, Technologies and Control, 1st Edition, ISBN: 9780128051955, Elsevier, Butterworth-Heinemann, 2016, pages 544, Biomedical engineering book series, pp. 171-210.

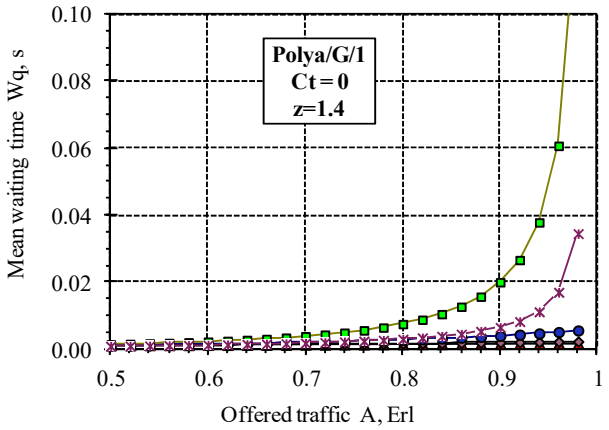
[3] Sztrik J. *Basic Queueing Theory*. GlobeEdit, 2016.



a). For coefficient of peakedness z=1.0 and coefficient of variation C_t=0

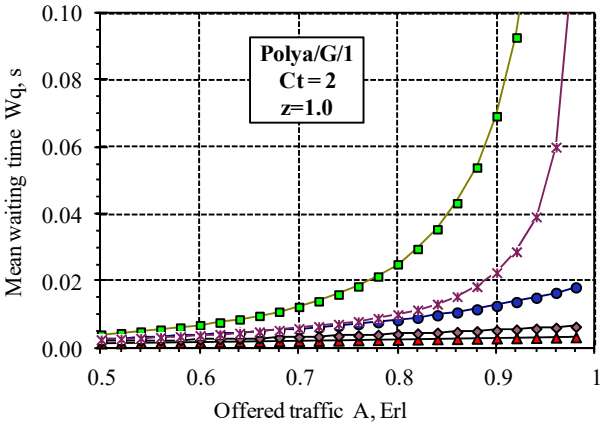


b). For coefficient of peakedness z=1.2 and coefficient of variation C_t=0

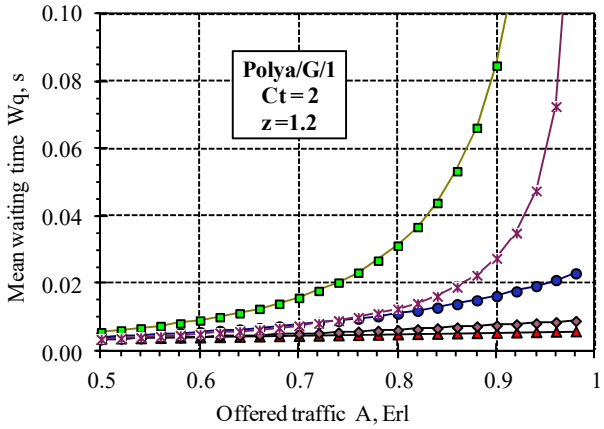


c). For coefficient of peakedness z=1.4 and coefficient of variation C_t=0

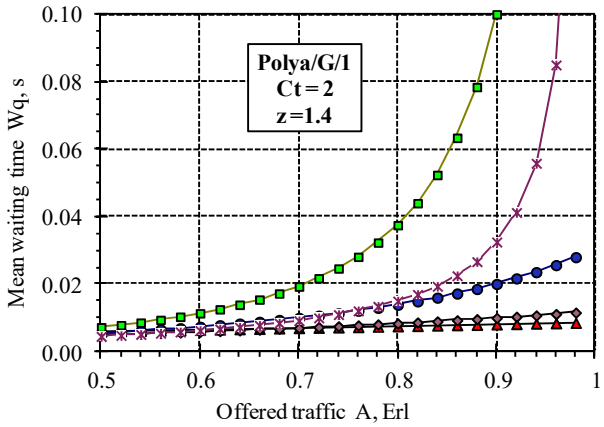
Fig.2. The mean waiting time in a Polya/G/1 queue with four non-preemptive priority classes as a function of the total offered traffic, for defined values of the coefficient of peakedness of the number of arrivals and fixed service time.



a). For coefficient of peakedness z=1.0 and coefficient of variation C_t=2



b). For coefficient of peakedness z=1.2 and coefficient of variation C_t=2



c). For coefficient of peakedness z=1.4 and coefficient of variation C_t=2

Fig.3. The mean waiting time in a Polya/G/1 queue with four non-preemptive priority classes as a function of the total offered traffic, for defined values of the coefficient of peakedness of the number of arrivals and peaked service process.

[4] Zukerman M. *Introduction to Queueing Theory and Stochastic Teletraffic Models*, City University of Hong Kong, 2016, https://arxiv.org/pdf/1307.2968.pdf

[5] Jain G., K. Sigman. A Pollaczek-Khintchine Formula for M/G/1 Queues with Disasters, Journal of Applied Probability, Vol. 33, No 4, 1996, pp. 1191-1200, doi: 10.2307/3214996

[6] Chang C., D. Lee, C. Yu. Generalization of the Pollaczek-Khinchin formula for throughput analysis of input-buffered switches, Proceedings IEEE 24th Annual Joint Conference of the IEEE Computer and Communications Societies, Vol. 2, 2005, pp. 960-970, doi: 10.1109/INFCOM.2005.1498325

[7] Huang L., T. Lee. Generalized Pollaczek-Khinchin formula for Markov channels, IEEE Transactions on Communications, Vol. 61, No 8, 2013, pp. 3530–3540, doi: 10.1109/TCOMM.2013.061913.120712

[8] Zhang J., T. Lee, T. Ye, W. Hu. On Pollaczek-Khinchine Formula for Peer-to-Peer Networks, Eprint arXiv:1605.08146, 2016, bib. code:2016arXiv160508146Z

[9] Abouee-Mehrizi H., O. Baron. State-dependent M/G/1 queueing systems, Queueing System, Vol. 82, No 1, 2016, pp. 121-148, doi: 10.1007/s11134-015-9461-y

[10] Sigman K. Using the M/G/1 queue under processor sharing for exact simulation of queues, Annals of Operations Research, Vol. 241, No 1-2, 2016, pp. 23-34, doi: 10.1007/s10479-013-1408-2

[11] Giambene G., S. Puzovic. Non-Saturated Performance Analysis for WiMAX Broadcast Polling Access, IEEE Transactions on Vehicular Technology, Vol. 62, No 1, 2013, pp. 306-325, doi: 10.1109/TVT.2012.2216296

[12] Mirtchev S., R. Goleva. New constant service time Polya/D/n traffic model with peaked input stream, Simulation Modelling Practice and Theory, Vol. 34, No 11-12, 2013, pp. 200-207, doi: 10.1016/j.simpat. 2012.08.004

[13] Mirtchev S., R. Goleva, V. Alexiev. Evaluation of Single Server Queueing System with Polya Arrival Process and Constant Service Time, Proceedings of the International Conference on Information Technologies (InfoTech-2010) Varna, Bulgaria, 2010, pp. 203-212.

[14] Haviv M. The performance of a single-server queue with preemptive random priorities. Performance Evaluation, No103, 2016, pp. 60–68.

[15] Xu B., X. Xu, X. Wang. Optimal balking strategies for high-priority customers in M/G/1 queues with 2 classes of customers. J. Appl. Math. Comput., No 51, 2016, pp. 623–642.

[16] Fajardo V. A Generalization of M/G/1 Priority Models via Accumulating Priority. Thesis for the degree of Doctoral of Philosophy in Statistics. Waterloo, Ontario, Canada, 2015.

[17] Mirtchev, S., Goleva, R., Atamian, D., Ganchev, I. Investigation of priority queue with peaked traffic flows. Proceedings of the ACM Symposium on Applied Computing, 2018, pp. 1017-1019.

[18] Mirtchev S., I. Ganchev. A Generalized Pollaczek-Khinchin formula for the Polya/G/1 queue. Electronics Letters, Vol. 53, No 1, 2017, pp. 27-29.

[19] Ramos H., D. Almorza, J. Garcia-Ramos. 'On Characterizing the Polya Distribution', ESAIM: Probability and Statistics, No 6, 2002, (6), pp. 105-112, doi: 10.1051/ps:2002005

Prof. DSc Eng. Seferin T. Mirtchev *has graduated telecommunications at Technical University of Sofia (TUS) in 1981. He is with Department of Communication Networks TUS, vice president of the Union of Electronics, Electrical Engineering and Telecommunications (CEEC), member of IEEE and has research interest in teletraffic engineering, switching systems, quality of service, cloud computing.*

tel.: +359 2 965 2254 e-mail: stm@tu-sofia.bg

Assist. Prof. PhD Eng. Rossitza I. Goleva *has graduated computer science at Technical University of Sofia in 1982. She is with Department of Informatics, New Bulgarian University, Vice Chair of the IEEE section Bulgaria, member of Union of Electronics, Electrical Engineering and Telecommunications (CEEC) and has research interest in cloud computing, quality of service, teletraffic engineering.*

tel.: +359 887991919 e-mail: rgoleva@gmail.com

Assist. Prof. PhD Eng. Dimitar K. Atamian *- assistant professor in the Technical University of Sofia, Bulgaria.*
e-mail: dka@tu-sofia.bg

Prof. PhD Ivan Ganchev *received his doctoral and engineering (summa cum laude) degrees from the Saint-Petersburg University of Telecommunications in 1995 and 1989, respectively. He is an IEEE Senior Member, an ITU-T Invited Expert, and an IET Invited Lecturer. Prof. Ganchev is with the University of Limerick (Ireland), the University of Plovdiv "Paisii Hilendarski" (Bulgaria), and the Institute of Mathematics and Informatics, Bulgarian Academy of Sciences (Bulgaria).*

e-mail: ivan.ganchev@ul.ie

Received on: 20.04.2019

Simple PV battery charger without MPPT based on SEPIC converter

Hristo Antchev, Anton Andonov

This article describes a specific implementation of a SEPIC converter for charging an accumulator battery from a photovoltaic panel and which is characterized by simple control without MPPT. Here are presented some basic mathematical dependencies on the transmission functions that can be used in the implementation of the control and regulation systems of other DC-to-DC converters without galvanic isolation. The article shows a schematic diagram and the way it work. Here are presented experimental research results as well as conclusions on the feasibility of the presented scheme and the dependencies.

Keywords – battery, charger, control, converter, photovoltaic.

Зарядно устройство за акумулаторна батерия от фотоволтаичен панел на основата на SEPIC преобразувател (Христо Анчев, Антон Андонов). *В статията се разглежда зарядно устройство за акумулаторна батерия на базата на SEPIC преобразувател (Single Ended Primary Inductance Converter). Устройството се характеризира с опростена и надеждна структура без следене на точката на максимална мощност MPPT (Maximum Power Point Tracking). Изведени са основните предавателни функции за входната мощност от фотоволтаичния панел по отношение на изходното съпротивление на преобразувателя, неговото входно напрежение и коефициента на запълване на управляващите импулси за транзистора. Представена е пълната принципна схема на зарядното устройство и е пояснена работата й. Зарядното устройство реализира алгоритъм на зареждане с константен ток и ограничение по напрежение за акумулаторната батерия. Представени са осцилограми от експериментални изследвания при начално зареждане, напълно заредена батерия, както и при сработване на защитата по ток при различни напрежения на фотоволтаичния панел.*

Introduction

With the widespread use of renewable energy sources, various applications of energy storage elements have been developed - supercapacitor and mainly accumulator batteries [1], [2]. DC-to-DC converters with or without galvanic isolation are used for their charging. For those without galvanic isolation, the basic application is for the SEPIC Converter and more rarely Zeta converter [3], [4], [5]. A research on a system of parallel co-working converters is done in [6]. The volt-ampere characteristic of photovoltaic panels, changing under temperature and solar radiation, generally requires Maximum Power Point Tacking (MPPT) for optimal power utilization. For this purpose, there are different methods described, for example, in [7], [8], with application example for SEPIC converter [9]. Significantly more limited are the descriptions of MPPT-free variants, popularly called Pulse Width Modulation (PWM) variants. They are characterized by a simple solution and lower price. At the same time, there are researches showing that, for example, in the temperature range of the panel and for lower power, the option without MPPT is more inexpensive and is almost as efficient as the one with MPPT [10].

This study presents a research of a SEPIC converter for charging an accumulator battery from a photovoltaic panel without MPPT. For creating the circuit diagram are used some of the ideas presented in [11], [12]. Part II presents the mathematical link between the major values. A description of the circuit diagram is made in Part III. The experimental results are shown in Part IV.

Mathematical description

Assuming that the energy efficiency of the converter is equal to 1, then:

(1) $P_O = P_I,$

(2) $U_O.I_O = U_I.I_I$

where P_O – output power, P_I - input power, U_O - output voltage, I_O - output current, U_I - input voltage, I_I - input current.

(3) $\frac{U_O}{U_I} = K = \frac{\delta}{1-\delta}; \quad \delta = \frac{t_{ON}}{T},$

where δ is the duty cycle, t_{ON} - the time the transistor is switched on, T- the period of operation.

From (2) and (3) follows

(4) $\frac{I_O}{I_I} = \frac{1}{K} = \frac{1-\delta}{\delta}$

and from (3) and (4)

(5) $\frac{R_I}{R_O} = \left(\frac{1}{K}\right)^2 = \left(\frac{1-\delta}{\delta}\right)^2$

Therefore, for the input power can be recorded:

(6) $P_I = \frac{U_I^2}{R_I} = U_I^2.\left(\frac{\delta}{1-\delta}\right)^2.\frac{1}{R_O}$

The full input power P_I differential is:

(7) $dP_I = 2.U_I.\left(\frac{\delta}{1-\delta}\right)^2.\frac{1}{R_O}.dU_I + 2.\frac{\delta}{(1-\delta)^3}.U_I^2.\frac{1}{R_O}.d\delta - \frac{1}{R_O^2}.U_I^2.\left(\frac{\delta}{1-\delta}\right)^2.dR_O$

Here can be figured the change of input power P_I at the changing of one of the values U_I, δ, R_O and the invariable value of the other two.

For example, in relation to the variation of the output resistance:

(8) $dP_I = -\left[\frac{U_I}{R_O}.\left(\frac{\delta}{1-\delta}\right)\right]^2.dR_O = -\left[\left[\frac{U_O}{R_O}.\left(\frac{1-\delta}{\delta}\right).\left(\frac{\delta}{1-\delta}\right)\right]\right]^2.dR_O = -I_O^2.dR_O$

Finally:

(9) $dP_I = -I_O^2.dR_O$

In relation to input voltage changes:

(10) $dP_I = 2.U_I.\left(\frac{\delta}{1-\delta}\right)^2.\frac{1}{R_O}.dU_I = 2.\frac{U_O}{R_O}.\frac{\delta}{1-\delta}.dU_I = 2.I_O.\frac{\delta}{1-\delta}.dU_I = 2.I_I.dU_I$

or:

(11) $dP_I = 2.I_O.\frac{\delta}{1-\delta}.dU_I = 2.I_I.dU_I$

Regarding the changes in the duty cycle:

(12) $dP_I = 2.\frac{\delta}{(1-\delta)^3}.U_I^2.\frac{1}{R_O}.d\delta = 2.\frac{U_O^2}{\delta.(1-\delta)}.\frac{1}{R_O}.d\delta = 2.\frac{I_O.U_O}{\delta.(1-\delta)}.d\delta = 2.\frac{P_O}{\delta.(1-\delta)}.d\delta$

Finally:

(13) $dP_I = 2.\frac{P_O}{\delta.(1-\delta)}.d\delta$

Fig. 1 shows the approximate change of the output current and the output voltage at charging algorithm with constant current and voltage limits.

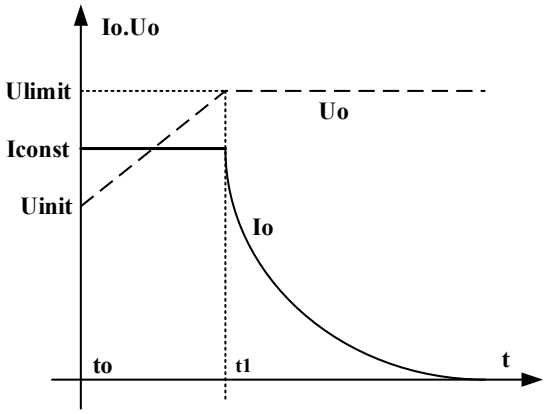


Fig.1. Change of output current and output voltage.

The initial power corresponds to the moment t_o:

(14) $P_{init} = U_{init}.I_{const}$

The biggest output power value will be at the moment t_1:

(15) $P_{limit} = U_{limit}.I_{const}$

For example, for a 12V nominal voltage accumulator and 50Ah capacity, the typical values are:

$U_{init} = 10V, U_{limit} = 13.8V, I_{const} = 5A$

For the interval $t_0 \div t_1$ the change of the output resistance is

$dR_O \approx \Delta R_O = \frac{U_{init}-U_{limit}}{I_{const}} = \frac{10-13.8}{5} = -0.76\Omega.$

For the change of the input power assuming that the energy efficiency is equal to 1, the formula (9) leads to $dP_I \approx \Delta P_I = -5^2.(-0.76) = 19W.$

A similar result is achieved in the following way

$$\Delta P_I = P_{t_1} - P_{t_0} = U_{limit}.I_{const} - U_{init}.I_{const} = 13.8.5 - 10.5 = 19W$$

The presented dependencies can also be used in other non-galvanic isolation converters, in which the input and output voltages are connected with (3) - buck/ boost, Cuk converter, Zeta converter.

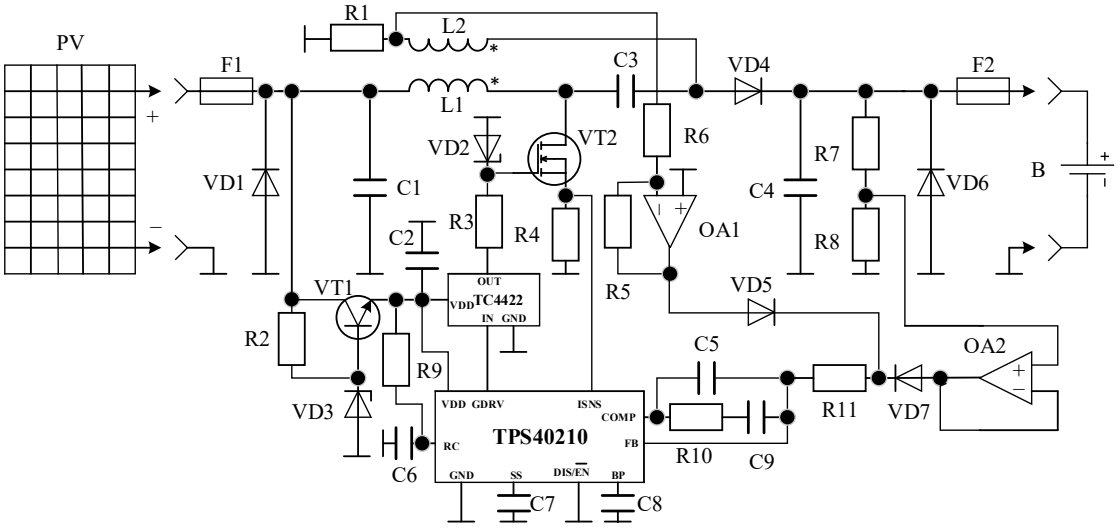


Fig.2. Schematic diagram of the converter.

Scheme and description of the operation

The schematic diagram is shown in Fig. 2. The main elements of the power scheme of the SEPIC converter are: *C1, L1, L2, VT2, C3, VD4, C4.*

The operation control is done through the integrated circuit TPS40210 [13]. To its output at the GDRV transistor is connected TC4422 driver [14]. The supply voltages VDD for the two integrated circuits are obtained by the stabilizer with the elements *VT1, R2, VD3,* and *C2*. The frequency of operation of the converter is set with the elements R9,C6 [13] and in this case, it is selected equal to 100 kHz. A battery-charging mode with constant current and voltage limitation is implemented. The elements of the voltage feedback circuit are *R7, R8, OA2, VD7*, with the output voltage value set to 3.8V. The value of the output charging current is monitored indirectly via *R1*, and the current feedback circuit also includes *OA1, R6, R5, VD5*. The indicators of the closed loop system (accuracy, sustainability, transition process parameters, etc.) are established by the elements connected between the COMP and FB terminals: *R11, R10, C5,* and *C9*. The connected capacitor *C7* to SS terminal determines the soft start time of the converter [13]. Additionally, the following protections are implemented: current overload and short-circuit at the output protections (via the resistor *R4* connected to ISNS terminal); change of polarity of the photovoltaic panel or the battery protections (respectively with *F1, VD1* and *F2, VD6*). For the design of the elements of the power circuit diagram can be used [15] for example.

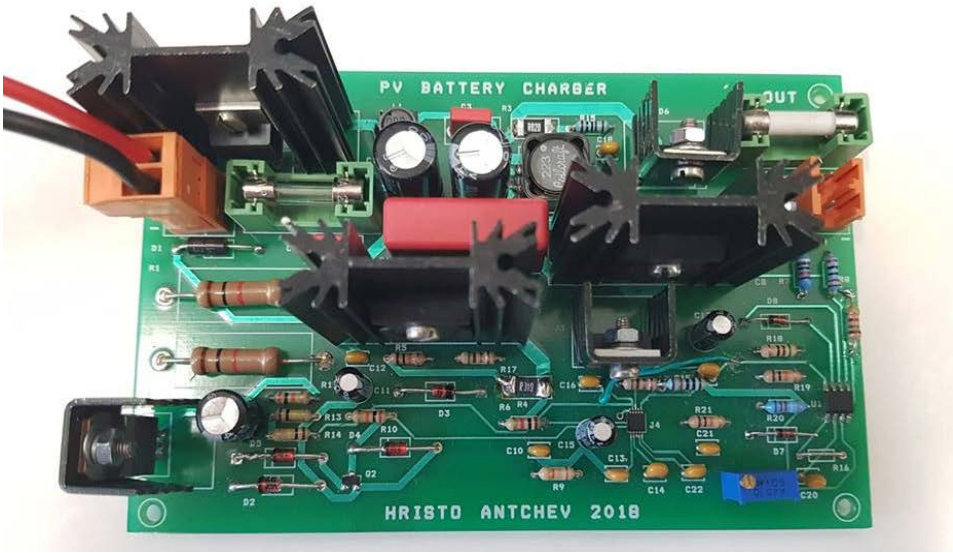


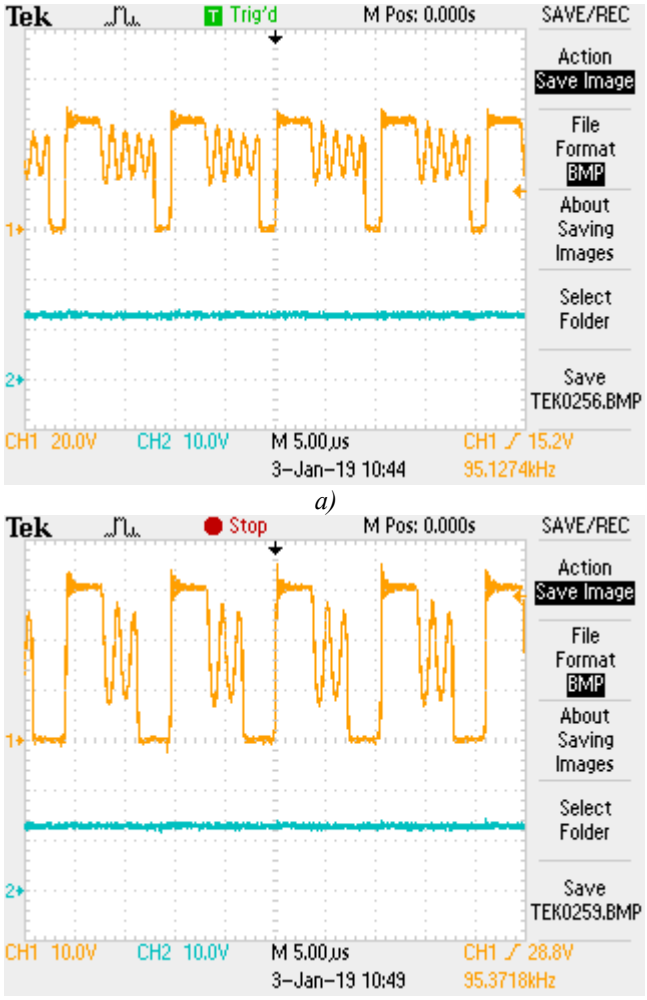
Fig.3. Appearance of the converter.

Experimental results

Fig.3 shows the appearance of the converter. The operating range of the input voltage is $9V - 36V$, as the low limit is determined by the minimal supply voltage of TPS40210.

The experiment is done with a *12V/50Ah* accumulator powered by a PV-PV120-1-18 photovoltaic panel with the following values: maximum power - $P_{MAX} = 120W$, open circuit voltage - $V_{OC} = 22V$, short circuit current - $I_{SC} = 7.34A$, and voltage at maximum power point - $V_{MPP} = 17.6V$, and current at maximum power point - $I_{MPP} = 6.8A$.

Fig. 4, Fig. 5 and Fig. 6 show some oscillograms of the experiments.



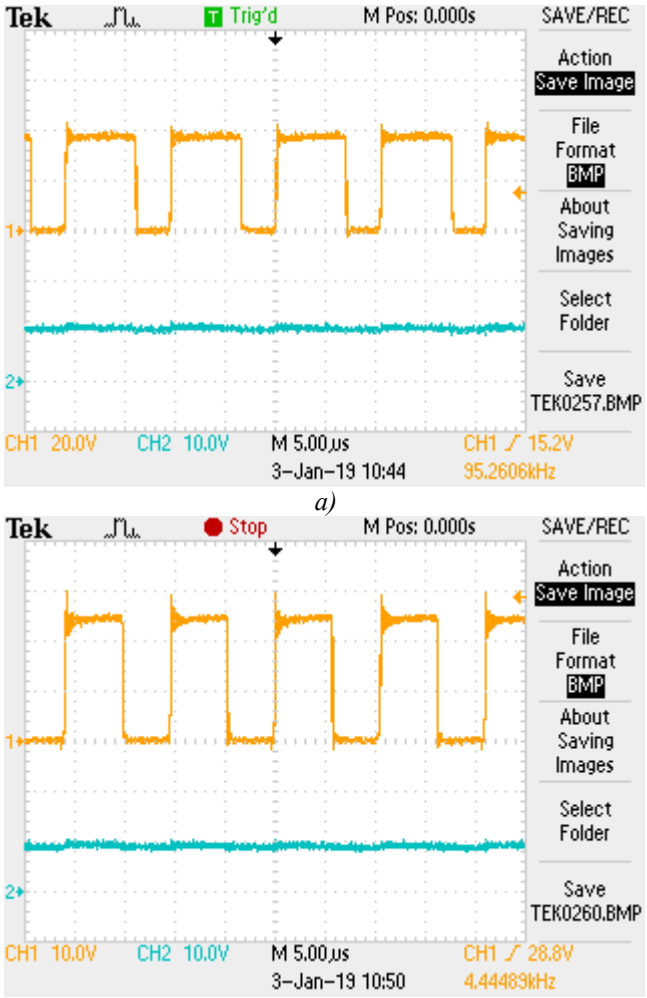
a)

b)

Fig.4. CH1 – drain-source voltage of VT2, CH2 – accumulator battery voltage,
a) input voltage 22V, b) input voltage 17V.

The experiments are made at input voltages around the maximum power point. CH2 on Fig. 4 shows that the battery voltage has reached the limit of ≈13.8V and the charging current has dropped to 0.35A (measured value).

CH2 on Fig. 5 shows that the battery voltage is at the beginning of the charge ≈ ≈10V and the charging current is constant 5A (measured value).



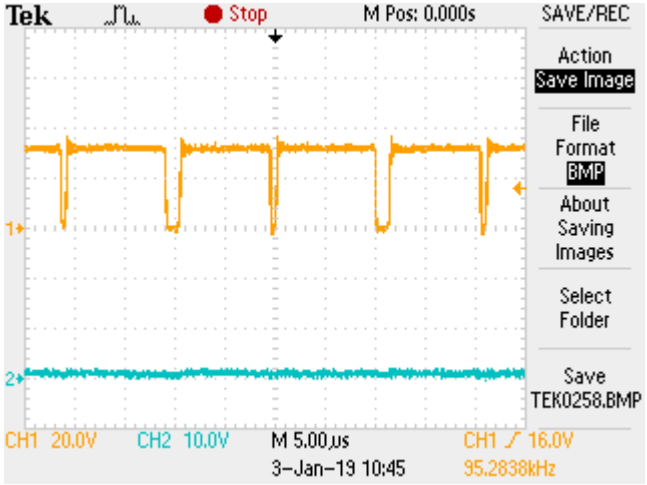
a)

b)

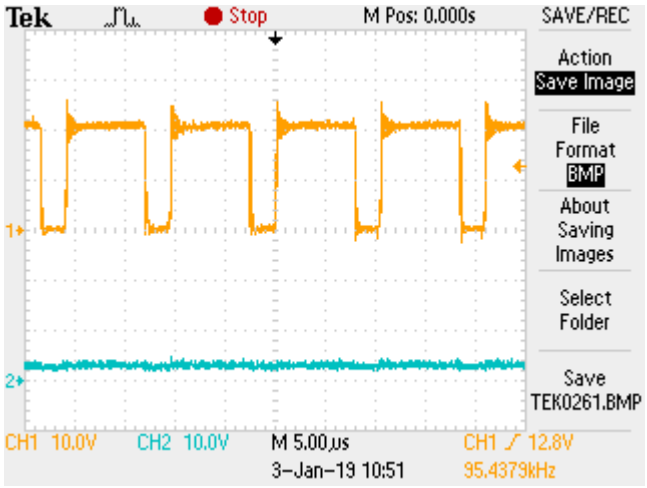
Fig.5. CH1 - drain-source voltage of VT2, CH2 - accumulator battery voltage,
a) input voltage 20V, b) input voltage 13V

Fig. 6 shows the oscillograms at short circuit at the output, i.e. the short-circuit protection. The current value at which the protection starts working is ≈5.5A. Fig. 6a shows that the duty cycle of the control impulses for the transistor has been reduced to a minimum value (at the time of switching on the transistor), resulting in a decrease of the output voltage approximately to 0. Therefore, the short-circuit current is measured about 6A (measured value). Fig. 6b shows that the duty cycle of the controlling impulses of the

transistor has not yet dropped to minimum value (at the time of switching on the transistor), resulting in a decrease in the output voltage but not yet equal to 0. At this, the current is measured around 5.8A.



a)



b)

Fig.6. CH1 - drain-source voltage of VT2, CH2 – accumulator battery voltage
a) input voltage 13V, b)input voltage 11V

Conclusion

For the analyzed in the present study SEPIC converter are obtained the transmission functions of the input power to the output resistance (9), the input voltage (11) and the duty cycle of the control impulses (13) that can be used in various methods and circuits. Here is described a concrete implementation of an accumulator battery charge converter of a photovoltaic panel without MPPT and a schematic diagram and experimental results are presented.

REFERENCES

[1] D. Ganesh, S. Moorthi, H. Sudheer, A Voltage Controller in Phot-Voltaic System with Battery Storage for Stand-Allone Applications, International Journal of Power Electronics and Drive Systems, Vol.2, No.1, 2012, pp.9-18.

[2] N. Sujitha, S. Krithiga, RES based EV battery charging systems: A review, Renevable and Sustainable Energy Reviews, 75(2017), pp.978-988.

[3] Microchip Technology, High-Power CC/CV Battery Charger Using an Inverse SEPIC(Zeta) Topology, AN1467, 2012.

[4] Texas Instruments, Jeff Falin, Designing DC/DC Converters based on SEPIC Topology, Analog Applications Journal, 4Q, 2008, derived from www.ti.com/aaj.

[5] S.J.Chiang, H.J.Shieh, M.C.Chen, Modeling and Control of PV Charger System With SEPIC Converter, IEEE Transactions on Industrial Electronics, Vol.56, Issue 11, 2009, pp.4344-4353, DOI:10.1109/TIE.2008.2005144.

[6] Antchev H., Investigation of systems of power electronic converters with common output DC voltage, PhD thesis, S., 2012, (in Bulgarian).

[7] B. Bendib, H. Belmili, F. Krim, A survey of the most used MPPT methods: Conventional and advanced algorithms applied for photovoltaic systems, Renevable and Sustainable Energy Reviews, Vol.45, May 2015, pp.637-648.

[8] S.E. Babaa, M. Armstrong, V. Pickert, Overview on Maximum Power Point Tracking Control Methods for PV Systems, Journal of Power and Energy Engineering, 2014,2 pp.59-72.

[9] T. Ramalu, M.A.M. Radzi, M.A.A.M. Zainuri, N.I.A. Wahab, R.Z.A. Rahman, A Photovoltaic-Based SEPIC Converter with Dual Fuzzy Maximum Power Point Tracking for Optimal Buck and Boost Operations, Energies, 9(8), 2016, DOI:10.3390/en9080604.

[10] Victron Energy B.V., Which Charge Controller: PWM or MPPT?, 24 June, 2014, The Nedherlands, derived from www.victroenergy.com.

[11] Texas Instruments, Fast Capacitor Charger with SEPIC Topology (42V to 52Vin, 160V/180Vout@900 mA) Reference Design, derived from www.ti.com/tool/PMP30162.

[12] Texas Instruments, 20W SEPIC Low-Cost Lead Acid Battery Charger reference Design, derived from www.ti.com/tool/PMP10081.

[13] Texas Instruments, TPS4021x 4.5V to 52V Input Current Mode Boost Controller, SLUS772F, March 2008-Revised March 2015.

[14] Microchip Technology, TC4421/TC4422, 9A High-Speed MOSFET Drivers, DS20001420F, 2002-2013.

[15] Texas Instruments, Designing a SEPIC Converter, Application Report, SNVA168E-May 2006-Revised April 2013.

Dr. H. Antchev *awarded Bachelor and Master degrees at Technical University – Sofia, Bulgaria, in specialization "Power Electronics. He is a senior assistant at the University of Chemical Technology and Metallurgy in Sofia. He works in the field of design of power electronic converters and control systems.*

e-mail: hristo.antchev@gmail.com.

Assoc. Prof. Dr. A. Andonov *is Head of the Department of Electrical Engineering and Electronics at the University of Chemical Technology and Metallurgy in Sofia. He graduated from the Technical University of Sofia, specialty electrical power supply and electrical equipment.*

e-mail: andonov@uctm.edu

Received on: 11.02.2019

Review of hardware-in-the-loop - a hundred years progress in the pseudo-real testing

Nikolay Brayanov, Anna Stoynova

In the last two decades, the electronics and mechatronics systems became undividable part of human life. With increasing of the number of application fields and functionalities, they became more complex and a lot more efforts need to be put in the process of verification. Hardware-in-the-loop (HIL) is an approach that optimizes this process. In many cases, it is the only efficient or even possible way for testing and thus a lot of efforts are put in it. There are a big number of proposals on how to use HIL for different applications. Maybe because its highly practical nature, the scientific definitions are still not clarified, the used terms are not summarized but used on a case by case bases. This paper presents an analysis of HIL systems' academicals and industrial publications and experiences. It goes through the HIL history and application, trying to clarify what it is for and how it is used. Based on this, it comes up with the definition of a HIL system and further classifies HIL systems' types. The biggest challenges are identified along with the applicable requirements for HIL systems engineering.

Keywords – Hardware-in-the-loop, HIL design, HIL architecture, HIL application

Обзор на hardware-in-the-loop – сто години прогрес в псевдо-реалното тестване (Николай Браянов, Анна Стойнова). *През последните две десетилетия, електронните и мехатронните системи се превърнаха в неделима част от битието на човека. С увеличаването на броя на приложенията и техните функционалности, този тип проблеми стават все по сложни и изискващи повече усилия във връзка с тяхната валидация. Hardware-in-the-loop (HIL) предлага решение за оптимизация на този процес. В много случаи, това е единствената възможност за тестване, поради тази причина подобрението на методологията е продължава да бъде важна задача. Публикувани са значителен брой предложения за реализацията и приложението на този тип системи. Въпреки това, терминологията все още не е прецизирана, и се използва във връзка с конкретна реализация. Тази публикация представя анализ на академичните и индустриални публикации за HIL системи. Чрез историята и приложенията на HIL се достига до същината на подхода. Ная тази база се предлага дефиниция за HIL система и класификация на видовете такива системи. Идентифицират се изискванията към такъв тип системи и се изброяват и анализират предизвикателствата пред тях.*

Introduction

Hardware-in-the-loop (HIL) is already widely used for verification and integration of electronics and mechatronics systems. It is applied in a rising number of industries and different stages of the development's process. What makes it unchangeable is its usage for validation in cases it is very expensive, dangerous or even impossible to do it in a real environment. Thus, it is commonly applied in automotive, railways, aerospace industries.

A number of publications claim that HIL systems have become an irreplaceable, integral part of the development and testing of complex devices. Even if it is widely used, the terminology is not clear. Different terms are used for same or very similar systems and there is no common definition for HIL, because the term is still not included in IEEE taxonomy [1].

HIL history

In spite of its significant importance, it is not easy to find and arrange in the time an accurate history of HIL systems. Perhaps it is because of its highly industrial "in the kitchen" application, or maybe because of non-formalized and non-academic form

that this approach still has. Most probably, the initial need and development of such kind of systems has started in high-technology dynamic systems used for military, aerospace or airborne applications, where high quality, but also confidentiality of applied technology is required.

The first known example of HIL system is a flight simulator. It could be considered as a HIL system with a physical control system (a pilot) placed in a physical sub environment (a real airplane mounted on the ground with a face towards the wind) and experiences the behavior of a virtual simulation of the environment. It was created in 1910 by the "Sanders Teacher" [2] in order to protect human's life but also the machine. In this case, the virtual part of the HIL system is the wind, which of course can't be controlled. This is the reason why the system was not very useful. Later, in 1917, more functional simulator was presented [3]. It includes controllable body which is used to represent different response and the feeling of speed. During 40's this kind of HIL system utilized all up coming technologies – analog computers, servo systems, hydraulics [4], etc. During the 50's and 60's the digital computers were introduced and gaining popularity, but they were not capable to fulfil simulator's needs. In the beginning of 70's started their integration in hybrid systems. Anyway, HIL technology has been in wide use in Defense and Aerospace industry as early as the 1950s [5]. It was continuously utilized in flight and missiles control industry as in the Sidewinder program [6], NASA highly maneuverable aircraft technology (HiMAT) [7].

In the automotive industry, the HIL application has started with a vehicle driving simulator [8], [9]. Consecutively, HIL was integrated for testing and development purposes of different functionalities [4] - 1987 Dynamic motor test stands with a real engines and simulated by hardware(an electrical motor) vehicles gears controlled by a digital process computer; 1987 HIL is integrated for the needs of ABS (antiblock braking systems) system development; 1988 performance and quality evaluation are done for ASR(Anti-Slip Regulation); 1992 it is applied for simulation of vehicles' systems dynamics

During the 90s the commercial HIL system were proposed [4] and HIL got widely integrated in number of industries. The HIL systems, started from a method of learning new pilots, became an unchangeable solution for complex devices development and testing. From very custom device, it ended up as standard industrial semi-automatic solutions. Its application defines its main objectives – correct recreation of the real life phenomenon and thus in hard real-time manner.

Application industries

In recent years, HIL systems are applied in all industries, related to safety and impossibility of real live test. As already reviewed they are widely integrated in automotive (powertrain control module [10], brake system [11], suspension systems [12]; general control system [13], [14]; dynamics [15]) and aerospace (flight simulation [16], fan rocket control [17], general verification and validation of flight and mission-critical software [18], power electronics [19])

Apart, HIL is also used in the railways (wheel slide protection systems [20], braking [21] but also vehicle dynamic and electric systems [22]); power electronics and electrical systems(output power converters [23], power electronics [24], solar power station [25], microgrids [26], power converters and electric machines [27], thermal power plant control systems [28]); manufacturing and distributed automation [29]; underwater vehicles [30]; robotics [31].

Currently, HIL simulation is generally applied:

- in safety related industries where the process of verification is complicated and thus expensive
- in industries, where verification could cause a damage of expensive components
- in industries, where functionalities can not be tested in the real environment, either because it is dangerous(autonomous vehicles) or impossible(space vehicles)

All other industries, related to hardware/software interaction could benefit from HIL. Later in this paper are discussed the key challenges for the HIL systems, which could enable their application.

HIL systems as part of the development process

HIL could be applied literally everywhere, and different users find it helpful in different stages of the development process. Next review is based on the V-model, as commonly used and widely accepted process. Particularly used is the model defined in the automotive safety standard ISO26262 [32]. The reason is that it is based on the general standard IEC 61508, Functional safety of electrical / electronic / programmable electronic safety-related systems [33], but provides more detail representation of the V diagram Fig. 1.

During development and implementation of system and control software [34], HIL systems are used for "construction and implementation", "System integration" and "Field tests", as described in their V

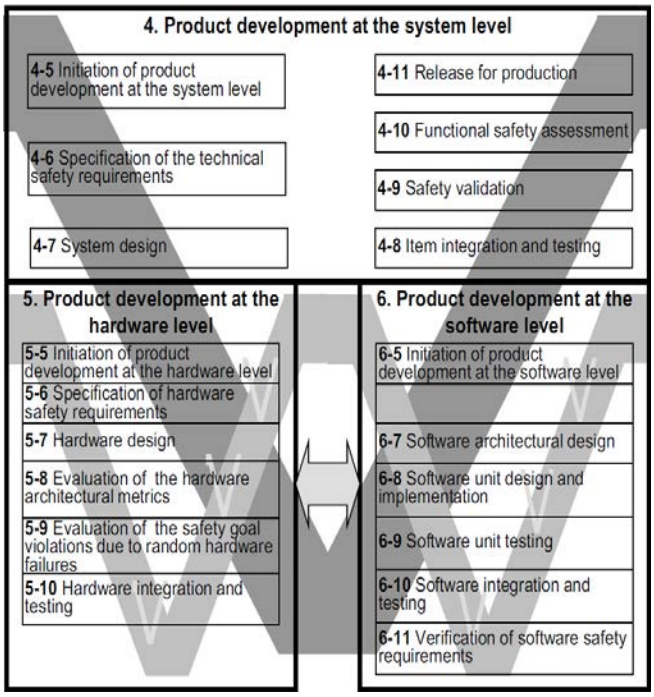


Fig. 1. Product development process V-style diagram. Credit ISO26262.

diagram. Mapped to already present diagram, the HIL is used for unit test, integration test, and product integration test. According to "HIL for automotive vehicle control systems development and testing" [35], it is used for item integration test and safety validation. A HIL simulation for electric drives and power electronics [36] uses HIL for "Subsystem Test" and "Subsystem integration", which corresponds to used V diagram as unit test and integration test. A case study in application HIL during the development of ECU for hybrid electric vehicle [37] proposes a HIL system for "Unit test", "System Test" and "Integration/Release test". For development of ADAS, HIL is applied for "Module verification", "System integration testing" and "System verification" [38]. For "Robustness and safety testing" and "System validation", they use the so-called VEHIL, or vehicle HIL, discussed later, as a variation of HIL.

Depending on its integration, HIL could be used all over the right part of the V diagram, evaluating all different kinds of tests. This enables the usage of the same test cases during whole the process of development, enabling its integration into a consistent toolchain.

Definition for Hardware-in-the-loop

State of the art definitions

In the literature HIL systems are known with various names as well as descriptions:

- technique for combining of a mathematical simulation system model with actual physical hardware, such that the hardware performs as though it were integrated into the real system [39];
- system with primarily objectives to control and observe the interfaces of the system under test (SUT), often done on semi-automatic way [40];
- technique where real signals from a controller are connected to a test system that simulates reality, tricking the controller into thinking it is in the assembled product [41].
- real-time simulator constructed by hardware and software, which is configured for the control SUT and connected to the target system or component through appropriate interface. During testing with an HIL simulator the target system or component does not experience significant difference from its integration in the real system [42].
- system, operating real components in connection with real-time simulated components [4].
- combination of physical target electronics units and physical communication bus, where real-time simulation is performed based on data, loaded to the electronic control units (ECUs) [43].
- real-time simulation for embedded control systems, in presence of hardware and other control systems in which a dynamic simulator is replaced by the real system [34]
- non-intrusive test mechanism where the environment of a SUT is simulated in order to perform tests on the SUT [44].
- method in which one or more real sub-systems interact in a closed loop with sub-systems that are simulated in real time to test them intensively in this virtual environment [45].
- synergistic combination of physical and virtual prototyping or a setup that emulates a system by immersing faithful physical replicas of some of its subsystems within a closed-loop virtual simulation of the remaining subsystems [46].
- combination of simulated and real components, alternatively, a real component can be emulated, i.e. replaced by an artificial component that has the same input and output characteristics in a closed-loop configuration [38]

The main aspects of a HIL system are easily notable – A combination of virtual and physical(real-world) systems; Based on a model of the environment; Executes in real time; Utilize the control signals, thus creating very close to the real environment for the

controller; A possibility to replace a simulator with an artificial real time hardware system. This gives a general description about HIL as a system type and approach for testing. To come up with a definition, few questions need to be clarified:

- Which part of the system should be a real physical system and respectively – which virtual?
- Is it black/grey/white box testing
- Is HIL always a matter of closed loop system, or open loop is also applicable?

HIL system as combination between physical and virtual

The Greek word "systema" means the organized relationship among the functioning units, a collection of elements is discernible within the total reality. Thus, a system includes control part, processing the input data and responding with output data and the process itself, containing the environment together with sensors and actuators. Based on already proposed HIL descriptions, one could state that HIL system either has real controlling and virtual processing or virtual controlling and real processing. However, the majority of the papers claim that HIL system is a system with a real controller and possibly virtual sensors, actuators or environment. They state that if a virtual controller is used, the evaluation is called control prototyping [4] or rapid control prototyping (RCP) [47], [48], [49], but it could also be found described another way [50]. General graph of both systems in given on Fig. 2.

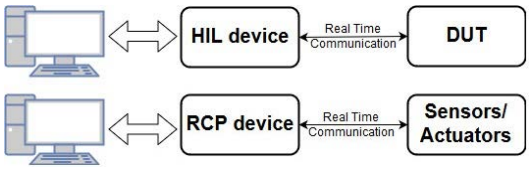


Fig. 2. HIL system (up) vs RCP system (down).

HIL system contains a PC that configures the HIL device. Once the simulation is ran, the HIL device executes simulation models and emulates the DUT's interface signals. RCP system is a platform used to evaluate the control algorithm, when the real one is not available [51]. It is configured with the DUT's software. When the simulations starts, the device executes DUT's software and attached real sensors and actuators interact with the environment. On Table 1 are described the main resources that both systems occupy. Obviously there is not that big difference between them. The fact is proved once again by the industry, which currently provides systems, capable to be used in HIL and RCP loop [52].

Table 1

Comparison of resources required for HIL and RCP systems

Resourse	HIL	RCP
Communication with the PC	Initially intensive, later floating	Initially intensive, later floating
Communication with DUT	real-time, intensivety depends on the system interface	na
Communication with Sensors/ Actuators	na	real-time, intensivety depends on the system interface
CPU load	simulation of the environment	processing the enironment data and control
Real time execution	Yes	Yes

The systems are very similar,but their application is essentially different. Recognition of the difference sticks to the common definition of HIL as a real controller system tested in simulated environment.

HIL test in the box

The type of testing is very important either as technology of the HIL system and as applicability. This is a consequence from the definitions – white box testing (WBT) is structural testing that is possible because of the deep knowledge for the SUT; black box testing (BBT) is a functional testing regarding the system's specification and with no understanding about its realization [53]. The term grey box testing (GBT) stands at the middle when the SUT is partially known.

Literature references HIL as a BBT systems, generally because they are used for system functional tests, which is a BBT by definition. Additionally most of the HIL systems are not able to access the SUT in real time in order to synchronize with its internal states and thus assure white-box treatment. However, proceeding WBT could have many advantages [54] and thus white box HIL systems are implemented. In such cases, synchronization between SUT and HIL is a big challenge, thus such kind of solutions are rare. It could be achieved by adding extra functionality in the main system [55] or by real time ECU access [56]. The solutions are very custom and thus expensive. Additionally, their implementation changes the execution of the SUT, which could introduce issues.

Commonly, the HIL test is GBT. The reason is that in the most of the cases the tested system is well known. Even though the functional testing requires BBT, in many cases using the knowledge about the system is beneficial, since one could improve the amount of covered code and execution paths.

HIL system in the loop

The topic if the HIL is close- or open- loop system is not particularly discussed in the literature. The most of the reviewed publications [46], [45], [57] etc. describe HIL as a close-loop system. However, there are papers state it other way [58], [59]. They describe:

- open-loop HIL - the generation of simulation data by the HIL simulator is independent of the previous output data of the SUT. The output of the SUT is only captured for future evaluation purposes but has no influence on the simulation data.
- close-loop HIL - the previous output of the SUT directly influences the calculation of subsequent input data. The HIL simulator must calculate the simulation data in real-time.

The benefits from open-loop HIL system are easier performing of unit level software verification and verification of the software and hardware interface of the low-level platform software [5]. Generally, it is simpler to integrate an open-loop systems and they are less resource hungry. There is no reason to neglect this kind of testing, even though the close-loop is more common.

Similar Terms

Further literature review enumerates variations of the HIL systems. For it, a car-environment system is presented Fig. 3.

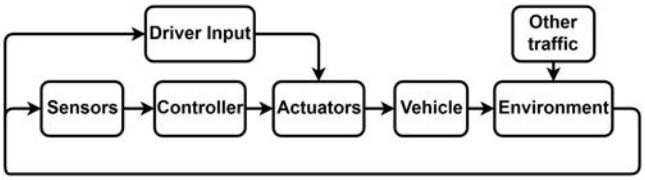


Fig. 3. Complex car-environment system.

As already reviewed, generally the HIL system contains a real controller in a loop with virtual or semi-virtual sensors and actuators. In this situation, if a human driver is included in the loop, the system could be called HIL [60], but alternatively driver- or human-in-the-loop [61], that could be also presented as type of HIL [62]. Later, if vehicle is added in the loop, so real car is on a real road and all its sensors are stimulated, then it is VIL (Vehicle-in-the-loop) [63]. And further - a real car, mounted on a stand with dynamometer and stimulating its sensors corresponding to artificial environment, this is called VeHIL (Ve stands for vehicle) [64], [65]. Alternatively VIL is described as placing the entire vehicle into a test facility, 'fooling' the sensors so that it senses it like real environment [66]. Finally, in case an engine is tested, using dynamometer, it is called Engine-in-the-loop [46], even though another paper [57] says, it is still HIL. Apart of these, one of the leaders in HIL systems defines Engine Environment HIL, Vehicle Environment HIL and Complete System Environment HIL up to nowadays distributed HIL solutions [67].

As reviewed behind, there are different claims about HIL systems and alternative pseudo-real simulations. In their research on Hybrid Powertrains, the authors claims that they are just different kinds of HIL [68]. From vocabulary point of view, hardware is mechanical equipment necessary for conducting an activity, usually distinguished from the theory and design that make the activity possible [69]. The phrase in-the-loop means included in a group that receive information about something [69]. Thus, generally either if the engine or vehicle is in the loop it should be called HIL.

HIL types

Referenced HIL types

The literature distinguishes several types of HIL, according to the level of the integration of the test platform. They are going to be analyzed, using a common mechatronics system graph Fig. 4. On it, the sensors measure the environmental signals, related to the environment, and provide low power electrical signals. It is later processed by Signal process and control, which in result generates actuators controlling low power signals. After the power amplifier, the signals are able to feed the actuators. Their act effects the mechanics and environment that closes the loop.

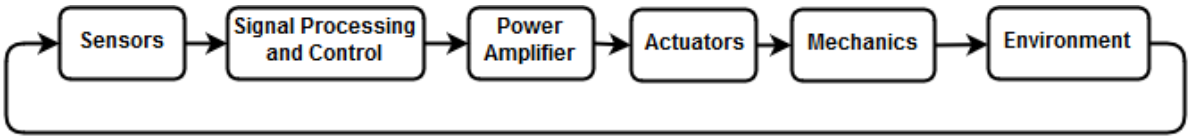


Fig. 4. Common mechatronics system form.

According to [70], HIL system could be established on signal level, power level and mechanical level. The authors describe signal level HIL (sHIL) as interconnection of a physical sub-system and a virtual residual system at signal level, which related to already proposed system, means that Signal Process and Control is real and the rest is simulated. The same definition is also supported by [56], [57] and [71].The concept of (electrical) power level HIL simulation (pHIL) summarizes HIL procedures which include a significant exchange of electrical power within the interface between the physical sub-system and the residual simulated system. Thus, aside from processing the real power amplifier is used, connected with some kind of physical loads. Mechanical level HIL simulation (mHIL) enables the close analysis of the mechanical interaction between a yet unrealized mechanical structure and an existing actuator system in the HIL test environment, so additionally in this case real actuators and mechanics are used.

For their proposal for hardware-in-the-loop model for electric vehicles [71] the authors analyze different HIL types. Power HIL system's realization includes real actuators. Additionally is proposed reduced-scaled HIL, similar to pHIL but loaded with equivalent subsystems with reduced power. The replaced subsystems have the same characteristic with the original ones. Similar definitions are given in Industrial Electronics Handbook [72].

At the electric power level the real power electronics could be simulated using electric loads [56]. At the mechanical level, simulation is done on a mechanical test bench, considering the effects of the real motor and additional real mechanical parts using the real controller and simulating environment. In difference with previously described mHIL, the DUT uses real sensors. This requires that HIL system contains reciprocal actuators, so it could stimulate the sensors. The same idea is published [44], however without having integrated real mechanics. Same HIL system is referenced as system for intelligent sensors and actuators [73].

pHIL is also definition as a test proceed over some of the real actuators [57]. Loads should be either real or with very close characteristics. Additionally, reduced scale power HIL is presented. It checks the operation principles, based on a load that is with significantly lower power, but similar load characteristics. Further, they describe "Mechanical power HIL" used to study the electric drive on a static bench. It fits to mHIL description. Regarding the last paper, "reduced scale mechanical power HIL" uses loads very similar to the original, but with smaller power.

Another term is Component-in-the-loop (CIL) [74]. Authors define it as a system used for testing of a real entire sub-system (hardware / mechanics / software) emulating its environment interface based on models. Also a rarely mentioned term is platform- and hardware-in-the-loop (PHILS) [75], describing a system that evaluates not only the hardware performance, but also cooperative performance of a group of autonomous robots.

Signal HIL

sHIL is usually chosen, when the physical transducers are not available or a certain test scenario cannot be established interfacing it. For example, if a fault injection campaign involves transducers to exhibit a particular erroneous behaviour, the use of real one could be very hard or even impossible. Additionally it is the simplest and the most universal HIL system, thus the cheapest and the fastest to be integrated. The review found that publications are in consensus, that sHIL is a system, that consist of real controller and all the rest of the system is simulated Fig. 5. sHIL could be found with other names as

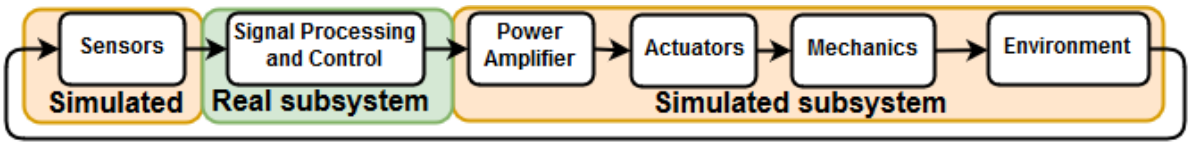


Fig. 5. Signal hardware-in-the-loop (sHIL).

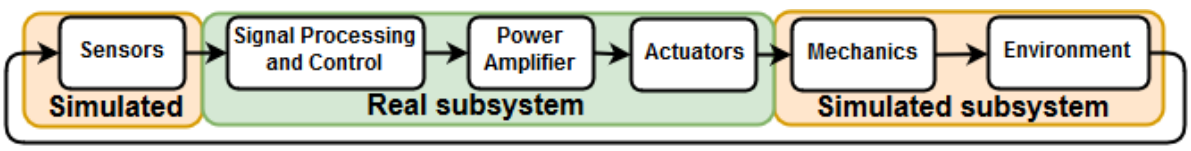


Fig. 6. Power hardware-in-the-loop (pHIL).

Controller-In-the-Loop (CIL) [46], Controller Hardware-In-the-Loop (CHIL) [76], [74] and even component level HIL [77].

Power HIL

pHIL moves the system in more realistic conditions [70]. The focus of the power level HIL tests is over power amplifiers and power electronics Fig. 6. In the context of the development of a controlled electromechanical system, an electrical power level HIL simulation enables testing of an existing power amplifier under realistic operating. There are number of variations in its definition and integration - reduced pHIL uses smaller loads; scaled pHIL with smaller load type with similar characteristics; pHIL with real loads.

In case the pHIL is used for evaluation of the electronics, it doesn't make sense to be used different loads. In some cases, using smaller loads with similar characteristics could be beneficial, especially in order to protect some high power devices, while reproducing the real system's behaviour. Because of the very similar characteristics of both pHIL and scaled pHIL could not be classified as different kind of HIL systems.

Mechanical HIL

mHIL adds extra level of reality, controlling the real actuators and investigating its real behaviour. Its evaluation is the closest to the reality test, the final one that could be done before the final product testing Fig. 7.

In this case, the border of integration is not that defined and solutions vary:

- real actuators simulated mechanics and sensors, based on the measured behaviour of the actuators.
- real actuators and real mechanics with additionally integrated sensors able to capture the systems dynamics. Thus, only the DUTs sensors and environment are simulated. In this case one should be caution on the way the mechanics is driven, since improper actions could damage it.
- real sensors and actuators and simulated subsystem of the environment. In this case pairing between the SUT's actuators and HIL sensors and SUT's sensors and HIL actuators should assure proper capturing of the environment and sensor's stimulation.

mHIL systems are very custom, and thus very slow for integration, complex and expensive. However their usage provides the closest to the real live simulation. This pseudo-real live testing is the only possibility if the environment is not accessible (the space), the equipment is very expensive (spaceships, airplanes, hi power modules) and as lately in autonomous vehicles, where there is high risks for the people's live.

Others

Based on the already defined types of HIL systems, component HIL [74] cannot fit to any of this groups. Its level of abstraction is undefined, so actually it could be part of any or all of the already classified groups. The given definition is very close to the general explanation of HIL. Platform-hardware-in-the-loop (PHILS) is a special case of a HIL system, that calculates a mathematical parameter based on hardware signals. Since the communication is on the level of standard communication signals, it could be encountered as one sHIL system.

HIL system designs

Non-standard solutions

In simplest decisions, the authors manage to proceed HIL tests, only with a personal computer (PC) [78]. This type of HIL system is possible, only because the required interface between DUT and the environment is RS232, commonly available on the PCs. Since Windows is not a hard real time operation system, such implementation cannot be used in case high simulation frequency or accuracy are required. Possible solution is HIL simulator based on a PC with Real-Time Linux [79]. Additionally, a programmable logic controller (PLC) based system is able to simulate the system even faster. The latest is applied only in HIL systems with PLC based system under test (SUT).

A distributed HIL system based on smart virtual transducers Fig. 8 is implemented over Atmel 4433 micro controller unit (MCU) [44]. Even rocket's HIL system is developed on a very simple ATMEGA 2560 8-bit MCU [17]. HIL realisations could be done on a low cost hardware [80]. The platform is not clarified,

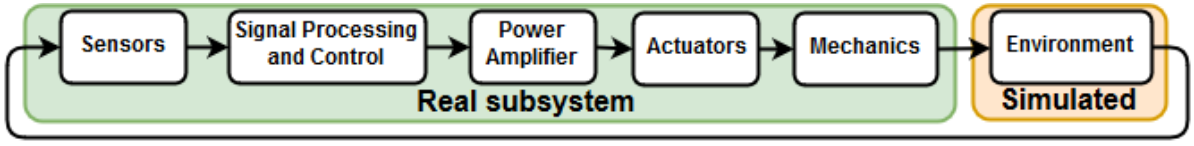


Fig. 7. Mechanics hardware-in-the-loop (mHIL).

but what is known, it is based on a target micro controller that is actually tested.

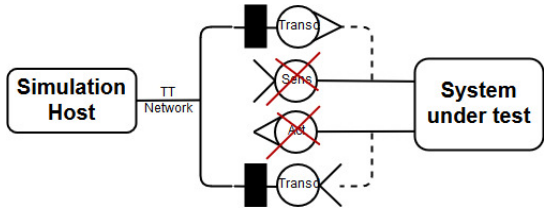


Fig. 8. MCU based smart virtual transducer HIL.

Mechanical level HIL on Arduino mega [81]. The SUT is unmanned aerial vehicle (UAV). A PC based flight simulator executes the simulation. The PC is connected to the MCU via RS232 and the controller takes care of the environment simulation.

Another paper proposes a distributed HIL system [82] using MSP430 series explorer board Fig. 9.

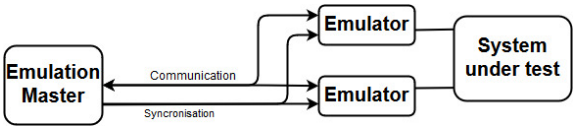


Fig. 9. Master-slave MCU based architecture for distributed HIL.

Development of a hardware-in-loop attitude control simulator [83], uses a computer connected via its USBs to a bunch of Arduino boards Fig. 10.

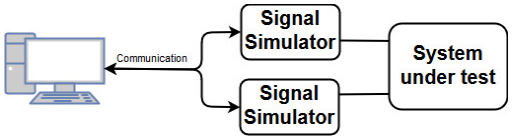


Fig. 10. PC mastered MCU based distributed HIL.

HIL systems for different power applications are implemented on field-programmable gate array (FPGA) [23], [24], [26], [84]. Aside from the FPGA, a digital signal processor (DSP) is used stimulate a solar systems [43]. Another proposal is a combination of FPGA and MCU [85]. Rarely but effectively, graphics processing units (GPUs) are used [75], in this case in combination with central processing units (CPUs) Fig. 11.

Standard HIL device are integrated in different ways. For example, they could be utilized with model based technics and code generation [50]. A low-cost real-time HIL testing approach [86] is based on a PC with bunch of specific tools and standard data acquisition devices. A HIL System for Active Brake Control Systems [11] contains two PCs. The Windows based host computer is used for simulation

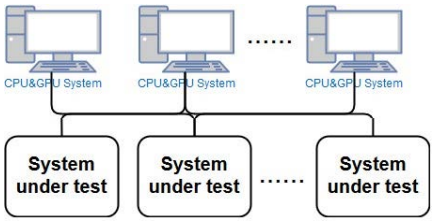


Fig. 11. CPU&GPU distributed HIL.

development and configuration. The target computer uses QNX operation system, capable to run the simulation and simulate the signal with standard data acquisition boards in real time manner. Advancing Subaru hybrid vehicle testing through HIL simulation [72] uses standard HIL devices to execute the simulation and output required signals. Additionally they synchronise an "ECU RAM monitor" interface, able to dump the ECUs memory Fig. 12. Thus, white box testing is possible.

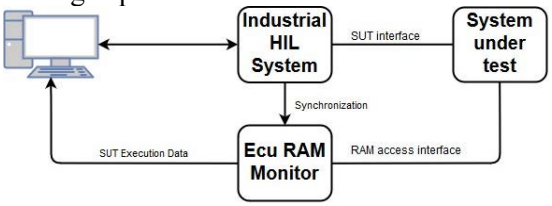


Fig. 12. ECU RAM monitor integration for white box HIL application.

HIL systems could be implemented over different computational platforms – CPUs, MCUs, GPUs, FPGAs, DSPs, PLCs. Authors prefer to choose the PC, as it is standard equipment. Further integrations use MCU platforms, often the same as the target. The implementation platform is chosen mainly because of authors experience and availability and current review of the nonstandard approaches is not able to demonstrate any dependency between used hardware platform and the application. Thus, industrial solutions for particular applications are going to be analysed.

Standard solutions

Standard solutions classification

Commercially available HIL simulation systems are classified based on their architecture, as simple simulators and complex simulators [87]. Simple HIL simulators contains a single hardware target (FPGA, MCU, etc.) connected to a development PC that proceeds an environment simulation. The complex HIL simulators are separated as monolithic and distributed HIL simulators. The monolithic HIL simulator is a single device that is modularly configured to offer all required interfaces for a

particular SUT. In contrast to the monolithic, a distributed HIL simulator consists of several interacting nodes that are capable to execute a distributed simulation model [49].

HIL systems are also classified as close and open architecture solutions. Close architectures, are hardly expandable, but offer whole hardware/software solutions for a particular problem. Opened systems provide different interfaces and thus could be integrated with many technologies. The cost of the openness is the more complicated set up and in result – increased configuration time [88].

Standard solutions platforms

DSpace proposes large line of HIL devices. A processor board based on CPU is designed for complex calculations [89]. It enables the combination of up to 20 modules of this kind, working distributed in parallel, in order to multiply the performance. CAN interface board is based on DSP [90]. Generally, the measurement of common signals like digital, analog, PWM, etc. are hardware implemented with so called "piggy modules" [91], [92]. To achieve fast parallel emulation of electrical signals are used FPGA based boards [93]. Another possibility is MCU based I/O board [94] and A/D board [95]. When it comes to the simulation and measurement of special automotive signals, the same provider proposes I/O board, combining an Angular processing unit (APU) and a DSP [96].

National Instruments does not provide particular information about the implementation. They claim their HIL system uses a CPU for computation and real time simulation processing and FPGA for electrical signals emulation [97].

Speedgoat uses a CPU base line real time target machine [98] and I/O modules, based on either FPGAs (hi-end realisation [99]) or MCUs (for the low-end products [100]).

ETAS proposes CPU based carrier board or housing [101] and FPGA I/O boards.

Currently all types of HIL system are in production, mainly because the simplest architecture defines the lower price that in many cases is the most important parameter. The implementation platform is also very important for the price, so the cheapest HIL systems are implemented over MCU. This defines their limitation, regarding calculation power as well as parallel signals emulation. The most of today's HIL system contain multi CPU part responsible for the calculations and FPGA, capable to handle very fast emulation of big number of signals in parallel.

HIL pros and cons

Waeltermann summarizes the benefits of HIL [45] - with HIL a reduction of development costs is achieved by moving function tests and diagnostics tests from tests drives or test bench experiments to the laboratory. The result is reduction of the number of expensive prototypes and time spent at the test bench. HIL tests can be often reproduced and automated, so it could automatically run, evaluate, and document. This allows the test operators to concentrate on assessing tests, implementing and adjusting tests. Automation provides better test coverage than manual tests, enhancing the quality.

HIL is a testing method that enables testing in different virtual situations, keeping the context on maximum possible reality level that is the last option before the real test. It is a trade-off between accuracy of test and cost and time consumption Figure 13. Because of its semi-virtual nature, HIL provides many benefits:

- Repeatable and stable - In contrast to road vehicle test, HIL is able to test complete feedback control systems in laboratory environment without disturbances from unrelated systems [25], [102].
- Cost effective - Extensive and expensive testing periods may be reduced, if HIL testing is applied for complex sub-systems [70]
- Verisimilitude, fidelity - By prototyping in hardware components, which dynamics or other attributes are not fully understood, HIL simulators often achieve higher fidelity levels [46].
- Non-destructive testing - HIL simulation often makes it possible to simulate destructive events eliminating the possibility for costly destruction [4].
- Comprehensiveness - HIL simulation often makes it possible to simulate a given system over a broader range of operating conditions than with purely physical prototyping [103].
- Flexibility – a minor changes in the HIL system could acknowledge changes in virtual system's behaviour or if SUT has to be evaluated for a multiple applications [70], [34]
- Parameter study, sensitivity analysis and optimization are easily proceeded [70].
- Safety - HIL simulators can often be used to train human operators of safety-critical systems in significantly safer environments [46].
- Concurrent systems engineering – A sub-system may be tested even if the remaining components

are only partially or not at all available, without losing sight of integration issues [29].

- Accelerates shift left - HIL enables the application of a consistent toolchain along the development and validation, thus using same test scenarios [65].
- Enables additional savings in personnel and equipment [104].

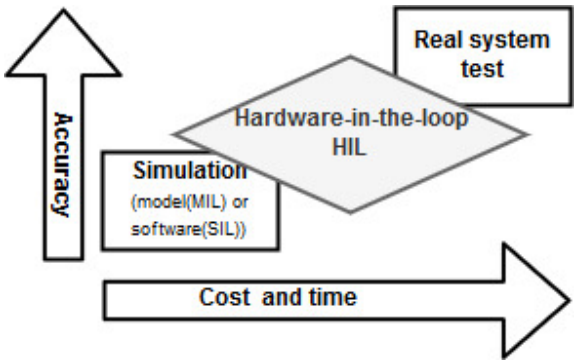


Fig. 13. Testing methods as a trade-off between accuracy of test and cost and time consumption.

Because of its great benefits and because there is no alternative for particular situation, one could hardly criticize it as approach. There are just few identified critics:

- No internal SUT information - HIL simulator will not directly give any information about the state of the tested control system since it only acts as a black box tester [85].
- Simulation speed is constant and actually the real execution one - The HIL simulators are able to run in real time and cannot be paused or slowed down [85], neither accelerated.
- No standard solutions, slow integration [85].
- Particular issues, related to non-perfect virtualization and parameters like bus length, terminations (for communication), back-feeding voltages(power circuits), etc. [35]

Generally, the HIL system could hardly be capable to utilize information for internal states of SUT, but as already reviewed it is possible and WBT could be done with it. As a matter of execution time, the real-time work of any system requires it can't tilt it. Thus the testing process could be accelerated only indirectly, e.g. improving HIL integration time. It is considered later as requirement for future improvement, since it is key enabler for wider HIL integration and price reduction. In any case, for faster integration time one would need standardization of the problem, and thus particular tight cases could not be in the focus.

HIL Requirements and challenges

A HIL simulation framework enumerates minimal set of functions that are present in all the HIL systems with similar terms [102], [105], [17]: numerical simulation, hardware interface, and real-time constraints. These functions should satisfy a number of requirements:

- Hi computational power –It could be a matter of possibility to integrate a high-fidelity simulation of the environment and virtualized subsystems, with resolution suitable for the SUT [106]. Computational power should assure sampling rates as low as 1ms [73]. Even real-time operating systems (RTOS) is pointed as mandatory for the hard real-time work of the HIL system [46]. But generally it is a trade-off between acceptable accuracy and achievable simulation time-step [36].
- Flexibility - Hardware platform and specially the interfaces should be very flexible so that a HIL system could be used either for rapid prototyping or for testing [106]. It also should be easily integrated in different applications [10] and capable to abuse special IOs e.g. for simulation of crank and camshafts, generation of knock signals, PWM signals, CAN interface), while using HIL Simulation to Test Mechatronic Components [73].
- Hi level of automation – Automation is needed on each level of the simulation, from injection of representative faults [106] up to the entire HIL system, e.g. in a script for signal generation and capturing in real time, complex test sequence structures, access to external devices [73]
- The price – even not engineering, it is always an important argument [106]

Apart from the general requirements, there are others, based either on particular needs or on different points of view:

- The computational platform should be expandable. A modular interface is going to enable cost effective modifications [10]
- Integration with different tools - The software architecture must be flexible enough to allow the use of products from existing tools, and products from other areas of research [106], [73]
- Enabled for scripting configuration for easier management and documentation [106].
- Modularity, standardization and expandability at microscopic and macroscopic levels [73], [44].
- Tight integration deadlines, more importantly, deadlines that can be planned and met [73], [85].

- Supplier competence in the particular industry [73].
- And others like Open- and close- loop testing capability [44]; Functional and non-functional testing [62]; Equipped with system debugging tools [10]; Connecting the test cases to requirements to ensure test coverage; Evaluating regression tests for faster and qualitative validation [72].

Conclusions

The idea for estimation or simulation of the unknown or unavailable is not new. It is applied in many ways, but HIL is definitely an enabler for integration and testing of nowadays high complex mechatronics and cyber physical systems. Even if a hundred years have been spent, even if last 20 years it is a hot topic, the application of this approach is still very time consuming, very custom and very expensive. Moreover, this field still doesn't have its state of the art terminology, which introduces a big number of different terms for very same or identical phenomena referenced in the publications.

The paper proposes a definition for HIL system: **Hardware-in-the-loop system is a non-intrusive test approach, containing physical controller connected in open- or close-loop with virtual or semi-virtual subsystems, providing faithful physical replicas of the real world and evaluating the SUT in either black / grey / white box manner.** Three main types of HIL systems were identified, based on the level of their integration – signal, power and mechanical HIL. The last one, is the most complex, but provides closest to the real live test, where everything but the environment is real. Obviously it is not possible to fully validate today's complex systems, without a faith in the models, so this is a very important issue, even not directly related.

The HIL systems are applied in many fields of the industry and thus it needs to implement a big number of requirements. However the most common identified parameter is the integration time. Small or complex, calculation intensive or with a large number of different signal interfaces, everyone needs to setup his HIL system fast and iterate with it. The best way to have this is higher level of automation and thus easily configurable interfacing with different tools and hardware platforms. There should be enough processing power to assure required frequency and accuracy of virtual world simulation. Last, but not the least, the price is important parameter, enabling the mass integration.

REFERENCES

[1] IEEE Taxonomy V1.01 , IEEE standard 2017

[2] M.Baarspul, "A Review of Flight Simulation Techniques", Progress in Aerospace Science, vol. 27,pp. 1-120, 1990.

[3] H.Valdere, "Flight simulators. A review of the research and development", 1908, Aerospace medical research laboratory

[4] R.Isermann, J.Schaffnit, S.Sinsel, "Hardware-in-the-Loop Simulation for the Design and Testing of Engine-Control Systems", IFAC Proceedings, 1998, Vol 31, Issue 4, pp 1-10

[5] S.Nabi, M.Balike, J.Allen, K.Rzemien, "An Overview of Hardware-In-the-Loop Testing Systems at Visteon", SAE 2004 World Congress & Exhibition, SAE Technical Paper 2004-01-1240, 2004

[6] M.Baileyand, J.Doerr, "Contributions of hardware-in-the-loop simulations to Navy test and evaluation", SPIE Proceedings Vol. 2741: Technologies for Synthetic Environments: Hardware-in-the-Loop Testing, 1996, pp.33-43.

[7] M.B.Evansand, L.J.Schilling, "The role of simulation in the development and flight test of the HiMAT vehicle", NASATM-84912, 1984

[8] J.Drosdol, F.Panik, "The Daimler-Benz Driving Simulator A Tool for Vehicle Development", SAE International Congress and Exposition, 1985, Technical Paper 850334.

[9] T.Suetomi, A.Horiguchi, Y.Okamoto, S.Hata, "The driving simulator with large amplitude motion system.", International Congress & Exposition, SAE International, 1991, Technical Paper 910113.

[10] S.Raman, N.Sivashankar, W.Milam, W.Stuart, S.Nabi, "Design and implementation of HIL simulators for powertrain control system software development", Proceedings of the 1999 American Control Conference

[11] Hwang, J.Roh, K.Park, J.Hwang, K.Lee, K.Lee, S.Lee, Y.Kim, "Development of HILS Systems for Active Brake Control Systems," 2006 SICE-ICASE International Joint Conference, Busan, 2006, pp. 4404-4408.

[12] H.Sahin, N.Fukushima, T.Mochizuki, I.Hagiwara. "Hil simulation evaluation of a novel hybrid-type self-powered active suspension system", IEEE International Conference on Industrial Technology, Chile, 2010

[13] M.Short, M.J.Pont, "Hardware in the loop simulation of embedded automotive control systems", Proceedings. IEEE Intelligent Transportation Systems, Austria, 2005.

[14] H.Hanselmann, "Hardware-in-the-loop simulation as a standard approach for the development, customization, and production test of ECU's", International Pacific Conference On Automotive Engineering, SAE Technical Paper 931953, 1993,

[15] H.Kim, H.Yang, Y.Park, "Robust Roll Control of a Vehicle: Experimental Study Using a Hardware-in-the-

Loop Set-up", Proceedings of the Institution of Mechanical Engineering Part D: Journal of Automobile Engineering, vol. 216, pp. 1-9, 2002.

[16] H.Yamasaki, T.Matsumoto, K.Itakura, S.Miyamoto, K.Yonemoto, "Development of a hardware-in-the-loop simulator and flight simulation of a subscale experimental winged rocket." 2013 IEEE/ASME International Conference on Advanced Intelligent Mechatronics(AIM). 2013, p. 1510–1515.

[17] R.A.Yulnandi, C.Machbub, A.Prihatmanto, E.M.I.Hidayat, "Design and implementation of hardware in the loop simulation for electric ducted fan rocket control system using 8-bit microcontroller and real-time open source middleware, Journ. of Mechatronics, Electrical Power and Vehicular Technology, Vol 8,No 1,2017

[18] E.Johnson, "Use of flight simulation to complement flight testing of low-cost UAV's", AIAA Modeling and Simulation Technologies Conference and Exhibit, 2001, Canada

[19] J.Vejlupek, J.Chalupa , R.Grepl, "Model Based Design of Power HIL System for Aerospace Applications".) Mechatronics 2013. Springer, Cham, ISBN 978-3-319-02293-2

[20] Kim, H.Y., Lee, N.J., Lee, D.C., Kang, C.G.. "Hardware-in-the-loop simulation for a wheel slide protection system of a railway train.", Proceedings of the international federation of automatic control, Cape Town, South Africa 2014;:24–29.

[21] R. Conti, E. Meli, A. Ridolfi, A. Rindi, "An innovative hardware in the loop architecture for the analysis of railway braking under degraded adhesion conditions through roller-rigs", Mechatronics, Vol 24, Issue 2, 2014, Pp 139-150, ISSN 0957-4158, ScienceDirect

[22] M.Spiryagin, C.Cole, "Hardware-in-the-loop simulations for railway research", Vehicle System Dynamics, Vol. 51, Issue 4, pp 497-498, Taylor & Francis, 2013

[23] O.Lucía, O.Jimenéz, L.Barragán, I.Urriza, J.Burdío, D.Navarro. "Real-time fpga-based hardware-in-the-loop development test-bench for multiple output power converters", IEEE Applied Power Electronics Conference and Exposition (APEC), USA, 2010

[24] O. Jimenéz, I. Urriza, L.A. Barragán, D. Navarro, J.I. Artigas, O. Lucía. "Hardware-in-the-loop simulation of fpga embedded processor based controls for power electronics", IEEE International Symposium on Industrial Electronics, Poland , 2011

[25] T. Debreceni, T. Kökényesi, Z. Süt, I. Varjasi. "Fpga-based real-time hardware-in-the-loop simulator of a mini solar power station". IEEE International Energy Conference (ENERGYCON), Croatia, 2014

[26] M.Panwar, B.Lundstrom, J.Langston, S.Suryanarayanan, S.Chakraborty. "An overview of real time hardware-in-the-loop capabilities in digital simulation for electric microgrids", North American Power Symposium (NAPS), 2013, USA

[27] L. Herrera, X. Yao, "FPGA-Based Detailed Real-Time Simulation of Power Converters and Electric Machines for EV HIL Applications", IEEE Energy Conversion Congress and Exposition, 2013, USA

[28] M.Iacob, G.D.Andreescu, Real-Time Hardware-in-the-Loop Test Platform for Thermal Power Plant Control Systems, IEEE Intern. Symposium on Intelligent Systems and Informatics, 2011, Serbia

[29] F.Gu, W.Harrison, D.Tilbury, C.Yuan. "Hardware-in-the-loop for manufacturing automation control: Current status and identified needs", IEEE International Conference on Automation Science and Engineering, 2007, USA

[30] O.Parodi, L.Lapierre, B.Jouvencel. "Hardware-in-the-loop simulators for multi-vehicles scenarios: survey on existing solutions and proposal of a new architecture", IEEE/RSJ International Conference on Intelligent Robots and Systems, 2009, USA

[31] A.Martin, E.Scott, M.R.Emami. "Design and development of robotic hardware-in-the-loop simulation", International Conference on Control, Automation, Robotics and Vision, 2006, Singapore

[32] Functional safety of road vehicles, ISO 26262, 2011

[33] Functional safety of electrical/electronic/programmable electronic safety-related systems, IEC 61508, 2010

[34] P. Sarhadi, S. Yousefpour , "State of the art: hardware in the loop modeling and simulation with its applications in design, development and implementation of system and control software", International Journal of Dynamics and Control 2015, Vol 3, Issue 4, pp 470–479

[35] P.J.King, D.G.Copp, "Hardware in the loop for automotive vehicle control systems development", Mini Symposia UKACC Control, 2004, UK

[36] S. Abourida , C. Dufour , J. Bélanger, "Real-Time and Hardware-In-The-Loop Simulation of Electric Drives and Power Electronics: Process, problems and solutions", Proceedings of the International Power Electronics Conference, 2005 , Japan

[37] D.Ramaswamy , R.McGee, S.Sivashankar , A.Deshpande, J.Allen , K. Rzemien, W.Stuart, "A Case Study in Hardware-In-the-Loop Testing: Development of an ECU for a Hybrid Electric Vehicle", SAE 2004 World Congress & Exhibition, SAE Technical Paper 2004-01-0303, 2004

[38] O. Gietelink, J. Ploeg, B. De Schutter, M. Verhaegen, "Development of advanced driver assistance systems with vehicle hardware-in-the-loop simulations", International Journal of Vehicle Mechanics and Mobility, 2006, Vol 44, Issue 7, pp 569-590

[39] L.Michaels, S.Pagerit, A.Rousseau, P.Sharer, S.Halbach, R.Vijayagopal, M.Kropinski, G.Matthews, M.Kao, O.Matthews, M.Steele, A.Will, "Model-Based Systems Engineering and Control System Development via Virtual Hardware-in-the-Loop Simulation", SAE

Convergence 2010, SAE Technical Paper 2010-01-2325, 2010

[40] M.Eider, S.Kunze, R.Pöschl, "FPGA based emulation of multiple 1-wire sensors for hardware in the loop tests", 2016 IEEE Sensors Applications Symposium (SAS), Italy

[41] "What Is Hardware-in-the-Loop", White Paper, National Instruments Corporation, Austin, USA, Available: www.ni.com/white-paper/53958/en/ [Accessed: 14 March 2019]

[42] "Hardware in the loop testing (HIL)", standard DNVGL-ST-0373, DNV GL AS 2016

[43] J. Chang, Y. Liu, "Evaluation of hardware-in-the-loop framework for intelligent safety systems verification", Master's Thesis, Dept. of Mech. and Maritime, Chalmers University of Technology, Göteborg, Sweden, 2018

[44] M. Schlager, W. Elmenreich, I. Wenzel, "Interface Design for Hardware-in-the-Loop Simulation", IEEE International Symposium on Industrial Electronics, 2006, Canada

[45] P.Wältermann. "Hardware-in-the-loop: The technology for testing electronic controls in automotive engineering", 6th Paderborn Workshop" Designing Mechatronic Systems", Paderborn 2016

[46] H.Fathy, Z.Filipi, J.Hagena, J.Stein, "Review of Hardware-in-the-Loop Simulation and Its Prospects in the Automotive Area". Proceeding SPIE 6228, Modeling and Simulation for Military Applications, 2006

[47] "Rapid Control Prototyping", Speedgoat, Bern, Switzerland, Available: https://www.speedgoat.com/applications-industries/applications/controller-prototyping [Accessed: 14 March 2019]

[48] "Rapid Control Prototyping", DSpace Gmbh, Paderborn, Germany, Available: https://www.dspace.com/en/inc/home/applicationfields/our_solutions_for/bussimulation/bussimulation_usecases/rapid_control_prototyping.cfm [Accessed: 14 March 2019]

[49] K. Kulkarni, "Rapid Control Prototyping for Automotive Control Software", White Paper, 2013, ETAS GmbH, Stuttgart, Germany, 2017, Available: https://www.etas.com/data/group_subsidiaries_india/webinar_2013-06-27_RCP_for_Autmotive_controls_1.pdf [Accessed: 14 March 2019]

[50] P.Baracos, G.Murere, C.A.Rabbath, W. Jin, "Enabling PC-based HIL simulation for automotive applications," IEMDC 2001. IEEE International Electric Machines and Drives Conference (Cat. No.01EX485), Cambridge, MA, USA, 2001, pp. 721-729.

[51] J. Khan, "Rapid Control Prototyping (RCP) solutions for the validation of motor control applications," 2016 International Conference on Emerging Technological Trends (ICETT), Kollam, 2016, pp. 1-6.

[52] "RT LAB OP4200, Affordable entry-level solution for your RCP/HIL needs", brochure, OPAL-RT, Montreal, Canada, 2018, Available: https://www.opal-rt.com/wp-content/uploads/2018/07/OPAL_FICHES_BDL42-100_RTLAB.pdf [Accessed: 14 March 2019]

[53] S. Nidhra, J. Dondeti, "Black box and white box testing techniques – a literature review", International Journal of Embedded Systems and Applications (IJESA) Vol.2, No.2, 2012

[54] M.E. Khan, F. Khan, "A Comparative Study of White Box, Black Box and Grey Box Testing Techniques", International Journal of Advanced Computer Science and Applications(IJACSA), Vol 3, Issue 6, 2012

[55] M. Iacob , G.D. Andreescu , "Implementation of Hardware-in-the-Loop System for Drum-Boiler-Turbine Decoupled Multivariable Control", 6th IEEE International Symposium on Applied Computational Intelligence and Informatics (SACI), 2011, Romania

[56] "dSPACE Hardware-in-the-Loop Test Systems", Brochure 2018, DSpace Gmbh, Paderborn, Germany, Available: https://www.dspace.com/shared/data/pdf/2018/dSPACE_HIL-Systems_Business-Field-Brochure_2018_English1.pdf [Accessed: 14 March 2019]

[57] A.Bouscayrol, "Hardware-In-the-Loop simulation", Industrial Electronics Handbook, second edition, tome Control and mechatronics, Taylor and Francis March 2011, Chapter M33

[58] M.Lauritzsen, "Hardware-in-the-Loop Testing Systems for ROV Control Systems", Master Thesis, Dep. of Marine Technology, Norwegian University of Science and Technology, Trondheim, Norway, 2014

[59] T. VanGilder, "What is Hardware-in-the-Loop (HIL) Testing?", White Paper, Wineman Technology Incorporated, USA, Available: https://www.winemantech.com/blog/what-is-hardware-in-the-loop-hil-testing [Accessed: 14 March 2019]

[60] G. Reymond, A. Heidet, M. Canry, A. Kemeny. "Validation of Renault's dynamic simulator for adaptive cruise control experiments" Proceeding of the Driving Simulation Conference, p181–192, France, 2000.

[61] K.Driggs-Campbell, G.Bellegarda, V.Shia, S.S.Sastry, R.Bajcsy, "Experimental Design for Human-in-the-Loop Driving Simulations", CoRR 2014 abs/1401.5039

[62] A.Soltani, F.Assadian, "A Hardware-in-the-Loop Facility for Integrated Vehicle Dynamics Control System Design and Validation", IFAC Symp. on Mechatronic Systems, 2016, UK, Vol 49, Issue 21, pp32-38

[63] S.A.Fayazi, A.Vahi, "Vehicle-in-the-loop (VIL) Verification of a Smart City Intersection Control Scheme for Autonomous Vehicles", IEEE Conference on Control Technology and Applications, 2017, USA

[64] R.Rossi, C.Galko, H.Narasimman, X.Savatier, "Vehicle Hardware-In-the-Loop System for ADAS Virtual Testing", Intern. Conf. on Embedded Computer Systems: Architectures, Modeling, and Simulation, 2014

[65] S.Riedmaier, J,Nesensohn, C.Gutenkunst, T.Duser, B. Schick, H.Abdellatif, "Validation of X-in-the-Loop

Approaches for Virtual Homologation of Automated Driving Functions", White Paper, AVL Deutschland GmbH, Available: https://www.avl.com/documents/10138/2699442/GSVF18_Validation+of+X-in-the-Loop+Approaches.pdf/0b13c98a-7e6d-45e7-baab-2a8d80403c38 [Accessed: 14 March 2019]

[66] Z.Saigol, A.Peters, M.Barton , M.Taylor, "Regulating and accelerating development of highly automated and autonomous vehicles through simulation and modelling", Technical report, Catapul, Swindon , UK, 2018, Available: https://s3-eu-west-1.amazonaws.com/media.ts.catapult/wp-content/uploads/2018/03/23113301/00299_AV-Simulation-Testing-Report.pdf [Accessed: 14 March 2019]

[67] G. Gackel, "Implementing a Global, Modular HIL Strategy for Automotive Electronics at PSA", National Instruments week 2015, Available: ftp://ftp.ni.com/pub/branches/us/tlf/automotive/psa_peugeot_citroen_tlf_2015.pdf [Accessed: 14 March 2019]

[68] Vora, H.Wu, C.Wang, Y.Qian, "Development of a SIL, HIL and Vehicle Test-Bench for Model-Based Design and Validation of Hybrid Powertrain Control Strategies", SAE 2014 World Congress & Exhibition, SAE Technical Paper 2014-01-1906, 2014

[69] Dictionary, Available: https://www.dictionary.com/ [Accessed: 14 March 2019]

[70] D.Mayer, T.Jungblut, J.Millitzer, S.Wolte, "Hardware-in-the-Loop Test Environments for Vibration Control Systems", NAFEMS Seminar Practical Aspects of Structural Dynamics, Germany, 2015

[71] T.Vo-Duy, M.C.Ta, "A signal hardware-in-the-loop model for electric vehicles", ROBOMECH Journal, 2016, Vol 3, Issue 29

[72] T.Morita, "Advancing Subaru Hybrid Vehicle Testing Through Hardware-in-the-Loop Simulation", Case study, Fuji Industries, Available: http://sine.ni.com/cs/app/doc/p/id/cs-15982 [Accessed: 14 March 2019]

[73] K.Lamberg, P.Wältermann, "Using HIL Simulation to Test Mechatronic Components in Automotive Engineering", Translation of "Einsatzder HIL-Simulation zum Testvon Mechatronik-Komponenteninder Fahrzeugtechnik",2nd Congress on Mechatronikim Automobil, Hausder Technik, Munich, 2000

[74] R. Vijayagopal, N. Shidore, S. Halbach, L. Michaels, A. Rousseau, "Automated Model Based Design Process to Evaluate Advanced Component Technologies", SAE 2010 World Congress & Exhibition, 2010, SAE International, Technical paper 2010-01-0936

[75] T.Furukawa, L.C.Mak, K.Ryu , X.Tong, "The Platform- and Hardware-in-the-loop Simulator for Multi-Robot Cooperation", Proceedings Performance Metrics for Intelligent Systems, pp 347-354, 2010, USA

[76] Y.Deng, H.Li, S.Foo, "Controller Hardware-In-the-Loop simulation for design of power management strategies for fuel cell vehicle with energy storage", IEEE Vehicle Power and Propulsion Conf.,2009, USA

[77] J.Khan, "A Standardized Process Flow for Creating and Maintaining Component Level Hardware in the Loop Simulation Test Bench", SAE World Congress and Exhibition, SAE International 0148-7191, 2016

[78] W.Adiprawita, A.S.Ahmad, J.Sembiring, "Hardware in the Loop Simulation for Simple Low Cost Autonomous UAV (Unmanned Aerial Vehicle) Autopilot System Research and Development", Proceedings of the International Conference on Electrical Engineering and Informatics, Indonesia, 2007

[79] C. Kleijn, "Introduction to Hardware-in-the-Loop Simulation", White Paper, Controllab, Enschede, the Netherlands, Available: https://www.hil-simulation.com/images/stories/Documents/Introduction%20to%20Hardware-in-the-Loop%20Simulation.pdf [Accessed: 14 March 2019]

[80] A.Taksale, V.Vaidya, P.Shahane, G.Dronamraju, V.Deulkar, "Low cost hardware-in-loop for automotive application", International Conference on Industrial Instrumentation and Control, 2015, India

[81] E.Atlas, M.Erdogan, O. Ertin, A. Guclu, Y. Saygi, U. Kaynak,"Hardware-in-the-Loop Test Platform Design for UAV Applications", Applied Mechanics and Materials, Vols. 789-790, pp. 681-687, 2015

[82] N. Brayanov, A. Stoynova, "Multi-Microcontroller Emulation of Sensors, Actuators and Systems for Flexible Integration of Complex Devices," Intern. Spring Seminar on Electronics Technology,2018,pp. 1-6.

[83] W.Tapsawat, T.Sangpet, S.Kuntanapreeda, "Development of a hardware-in-loop attitude control simulator for a CubeSat satellite", IOP Conf. Series: Materials Science and Engineering, Vol. 297, conf. 1, 2018

[84] E.Duman, H.Can, E.Akin, "FPGA based Hardware-in-the-Loop (HIL) simulation of induction machine model", International Power Electronics and Motion Control Conference and Exposition, 2014, Turkey

[85] M. Imani, "Hardware-in-the-Loop simulation of servo drivers on an embedded system", Master Thesis, Dep. Of Machine Design, Royal Institute of Technology, Stockholm, Sweden, 2014

[86] B. Lu, X. Wu, H. Figueroa, "A Low-Cost Real-Time Hardware-in-the-Loop Testing Approach of Power Electronics Controls," in IEEE Transactions on Industrial Electronics, vol. 54, no. 2, pp. 919-931, 2007.

[87] M.Schlager, R.Obermaisser, W.Elmenreich, "A Framework for Hardware-in-the-Loop Testing of an Integrated Architecture", Software Technologies for Embedded and Ubiquitous Systems, Springer Berlin Heidelberg , pp 159-170, 2007

[88] L.Long, G.Gefke, B.McKay, "Open Architecture Solution for Hardware-in-the-Loop Testing", White Paper, 2008, Mathworks Inc., Natick, USA, Available: https://pdfs.semanticscholar.org/

ab5e/4f55b4b3589e618154b01feb05ecd4342274.pdf [Accessed: 14 March 2019]

[89] "DS1006 Processor Board", DSpace Gmbh, Paderborn, Germany, Available: https://www.dspace.com/en/pub/home/products/hw/phs_hardware/processor_boards/ds1006.cfm [Accessed: 14 March 2019]

[90] "DS4302 CAN Interface Board", DSpace Gmbh, Paderborn, Germany, Available: www.dspace.com/en/inc/home/products/hw/phs_hardware/i_o_boards/can_interface.cfm [Accessed: 14 March 2019]

[91] "PWM Measurement Solution", DSpace Gmbh, Paderborn, Germany, Available: https://www.dspace.com/shared/data/pdf/2018/dSPACE_PWM_Measurement_Solution_Catalog2018.pdf[Accessed:14March 2019]

[92] "Electric motor HIL solutions catalog", DSpace Gmbh, Germany, Available: https://www.dspace.com/shared/data/pdf/2018/dSPACE_EMH_Solution_Catalog2018.pdf [Accessed: 14 March 2019]

[93] "FPGA Base Board For high-resolution signal preprocessing", DSpace, Paderborn, Germany, Available: https://www.dspace.com/shared/data/pdf/2018/dSPACE_FPGA_Base_Board_DS5202_Catalog2018.pdf [Accessed: 14 March 2019]

[94] "HIL Digital I/O Board", DSpace Gmbh, Paderborn, Germany, Available: https://www.dspace.com/shared/data/pdf/2018/dSPACE_DS4004_Catalog2018.pdf [Accessed: 14 March 2019]

[95] "A/D Boards", 2018, DSpace, Paderborn, Germany, Available: https://www.dspace.com/shared/data/pdf/2018/dSPACE_DS2002_DS2003_Catalog2018.pdf [Accessed: 14 March 2019]

[96] "Position Sensor Simulation Solution",DSpace Gmbh,Paderborn,Germany,Available:www.dspace.com/en/inc/home/products/hw/phs_hardware/i_o_boards/position_sensor_simulation.cfm Accessed:14March2019]

[97] "HIL Simulators", White Paper, 2016, National Instruments Corporation, Austin, USA, Available: http://www.ni.com/pdf/products/us/HIL_simulators_flyer_2016_FG.pdf [Accessed: 14 March 2019]

[98] "Baseline real-time target machine", Speedgoat, Bern, Switzerland, Available: www.speedgoat.com/products-services/real-time-target-machines/baseline/capabilities#compatible-io [Accessed:14March 2019]

[99] "FPGA I/O Module", Speedgoat, Bern, Switzerland, Available: https://www.speedgoat.com/products/simulink-configurable-fpgas-fpga-i-o-modules-io391 [Accessed: 14 March 2019]

[100] "Simultaneous Sampling 16-bit Analog I/O Module", Speedgoat, Bern, Switzerland, Available: https://www.speedgoat.com/products/io-connectivity-analog-io183 [Accessed: 14 March 2019]

[101] "Housing, Users guide", ETAS GmbH, Stuttgart, Germany, 2017, Available: https://www.etas.com/download-center-files/products_ES5000/ES5300.1A_ug_r04_en.pdf [Accessed: 14 March 2019]

[102] M.A.A.Sanvido, "Hardware-in-the-loop simulation framework", Doctoral Thesis, Swiss Federal Institute of Technology (ETH), 2002, Zurich DOI:10.3929/ethz-a-004317368

[103] Wilhelm, E.Fowler, M.W.Fraser, R.A.Stevens, "Hardware in the loop platform development for hybrid vehicles", 2007, Conference Proceeding

[104] F.Jiang, S.Gao, J.Zhang, "Prototyping Hardware-in-the-loop Simulation System of Diesel Engine on Linux System with Automatic Code Generation", Powertrains, Fuels and Lubricants Congress, ISSN 0148-7191, 2008 SAE International

[105] R.O.Sanchez, R.Horowitz , P.Varaiya, "Hardware-In-The-Loop On-ramp Simulation Tool to Debug and Test the Universal Ramp Metering Software", International Federation of Automatic Control Proceedings, Vol 42, Issue 15, 2009, pp 211-216

[106] M.Short, M.J.Point, "Assessment of high-integrity embedded automotive control systems using hardware in the loop simulation", Journal of Systems and Software, Vol 81, Issue 7, 2008, pp 1163-1183

Nikolay P. Brayanov is *a PhD student in the Technical University of Sofia. He studies the possibility for automation of the process of development, integration and configuration of hardware-in-the-loop systems in the context of the automotive industry. He is interested in embedded systems and model-based development.*

e-mail: npb@ecad.tu-sofia.bg

Prof. Anna V. Stoynova *is with Faculty of Electronic Engineering and Technology, Technical University of Sofia. Her scientific interests are in the fields of Quality and Reliability, Thermal Nondestructive Evaluation, Thermal management, Model-based Development. e-mail: ava@ecad.tu-sofia.bg*

Received on: 30.04.2019

Simulation of the temperature distribution and modelling of molten pool parameters at electron beam drip melting of copper

Tsvetomira Tsonevska, Elena Koleva,
Lilyana Koleva, Georgi Mladenov

In this article quasi steady-state two-dimensional heat model is implemented for the simulation of the temperature distribution in the cast copper ingots through electron beam drip melting. Regression models are estimated for the dependence of the shape of the crystallization front (molten depth, width, volume of the molten metal) on the variation of the process parameters - electron beam power, beam radius and the casting velocity. The molten pool crystallization surface and its reaching the outer wall of the ingot are discussed. The form of the crystallization front controls the dendrite structure formation, the uniform impurities' removal, the process of refining of the metal, the surface to volume ratio of the molten metal, the roughness of the obtained ingot side-walls and thus the quality of the obtained ingots.

Keywords: electron beam drip melting, heat model, temperature distribution, regression models.

Симулиране на температурното разпределение и моделиране на параметрите на течната вана при капково електроннолъчево топене на мед (Цветомира Цоневска, Елена Колева, Лиляна Колева, Георги Младенов). *В тази статия е използван квазистационарен двуизмерен топлинен модел за симулиране на температурното разпределение в отлетите медни слитъци чрез капково електроннолъчево топене. Оценени са регресионни модели за зависимостта на формата на кристализационния фронт (дълбочина на разтопяване, ширина, обем на разтопения метал) от изменението на параметрите на процеса - мощност на електронния лъч, радиус на лъча и скорост на леене. Дискутират се повърхността на кристализиране на стопилката и достигането ѝ до външната стена на слитъка. Формата на кристализационния фронт контролира: формирането на дендритната структура, равномерното отстраняване на примесите, процеса на рафиниране на метала, съотношението повърхност към обем на разтопения метал, грапавостта на страничните стени на получените слитъци и по този начин качеството на получени слитъци.*

Ключови думи: капково електроннолъчево топене, топлинен модел, температурно разпределение, регресионни модели.

Introduction

The drip melting is a classical method for electron beam melting and refining (EBMR) of metals. The raw material in the form of bars is fed horizontally (or vertically) and drip-melted directly into the withdrawal mold (Fig. 1). During the EBMR on the surface of the molten pool that is situated on the upper part of the cast ingot, the droplets, created on the front surface of feeding rod, go into the liquid pool in the crucible. In the same time the electron beam heats this surface, keeping it molten. The liquid pool surface is maintained by the operator on a constant level, by withdrawing the bottom of the growing ingot.

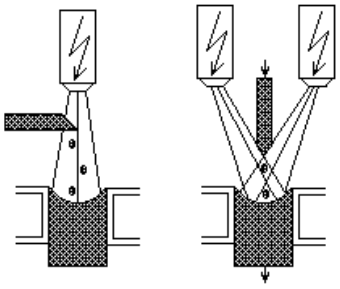


Fig.1. Drip melting.

Refining is based on degassing and selective evaporation of metallic and non-metallic constituents with vapor pressures higher than the base material.

Hard inclusions that are refined there could float or be dissolved in the molten metal. Usually repeated re-melting of the first melt ingots is required to achieve the desired final quality. For repeated re-melting often vertical feeding is applied due to more uniform irradiation of molten pool surface [1, 2].

The importance of the knowledge of the shape of the crystallization front is directly connected with the quality of the ingot – it is a key condition for producing ingots with a perfect crystal structure. A flat crystallization front permits formation of dendrite structures, parallel to the block axis as well as the uniform impurity displacement toward the ingot top surface.

The shape of the molten pool crystallization surface (bottom of the melting pool) controls the dendrite formation and the impurities removal. Uniform crystallization front surface, parallel to the top molten pool surface, avoids the typical for big diameter ingots segregation (namely, the impurities' concentration increase in the central region of the cast blocks) [1, 2].

In this paper quasi steady-state two-dimensional heat model is implemented for the simulation of the temperature distribution in the cast copper ingots through electron beam melting and refining. Regression models are estimated for the dependence of the shape of the crystallization front (molten depth, width and volume of the molten metal) on the variation of the process parameters - electron beam power, beam radius and the casting velocity. The molten pool crystallization surface reaching the outer wall of the ingot is discussed.

Heat model

The computer simulation of the temperature distribution at electron beam drip melting is based on the quasi steady-state two-dimensional heat model:

$$(1)\qquad \frac{1}{r}*\frac{\partial}{\partial r}*\left(r*\frac{\partial T}{\partial r}\right)+\frac{\partial^2 T}{\partial z^2}+\frac{v}{a}*\frac{\partial T}{\partial z}=0,$$

where the last term indicates the casting presented by the heat that is added by the poured molten metal into the crucible and given by the casting velocity of the ingot, moved with a speed v, coinciding with the z-axis. The temperature distribution in the ingot is described by the Poisson equation. It is used an axis-symmetrical thermal geometry. The stirring and mixing processes in the molten pool are presented by a modification (increase from 1 to 2 times) of the value of the ingot thermal conductivity.

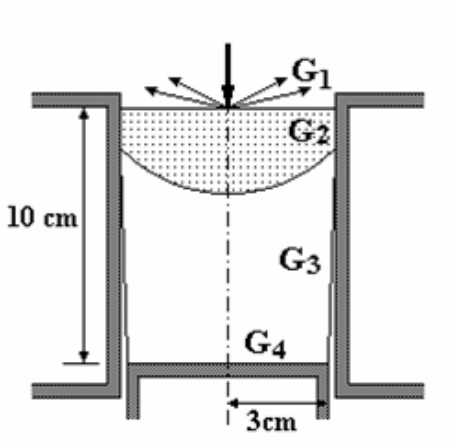


Fig.2. Geometrical conditions: G_1 – top ingot surface, G_2 – interface molten pool/crucible, G_3 – free side ingot surface, G_4 – interface – ingot/puller. The shown dimensions are typical radius and length for the used crucibles.

A set of appropriate boundary conditions takes into account the radiation losses and the heating beam energy distribution with a correction of the secondary electron energy losses and they are given according the interfaces G_1, G_2, G_3 and G_4 (Fig. 2) correspondingly:

$$(2)\qquad G_1:\ \lambda_1\times\frac{\partial T}{\partial z}|_{z=h}=-P_n+\alpha\times\sigma\times\left(T^4-T_{st}^4\right)$$

$$(3)\qquad G_2:\ \lambda_1\times\frac{\partial T}{\partial r}|_{r=R}=\lambda_2\times\frac{\partial T'}{\partial r}|_{r=R}$$

$$(4)\qquad G_3:\ \lambda_1\times\frac{\partial T}{\partial r}|_{r=R}=-\sigma\times\sigma\times\left(T^4-T_{st}^4\right)$$

$$(5)\qquad G_4:\ \lambda_1\times\frac{\partial T}{\partial z}|_{z=0}=\lambda_2\times\frac{\partial T''}{\partial z}|_{z=0}$$

where α is emissivity, σ is the Stefan-Bolzman constant, λ is the thermal conductivity, T_{st} is the ingot surface temperature. In Table 1 are given the material characteristics, used for the calculations. The heat transfer coefficients on the boundary areas (G_1, G_2, G_4) assumed are: $\lambda_1/\lambda_{Cu}=1.0$, $\lambda_2/\lambda_{Cu}=0.8$, $\lambda_4/\lambda_{Cu}=0.8$. Working with a real process these coefficients should be estimated first in order to apply the considered here approach. The height of the interface molten pool-crucible is constant – 8 mm.

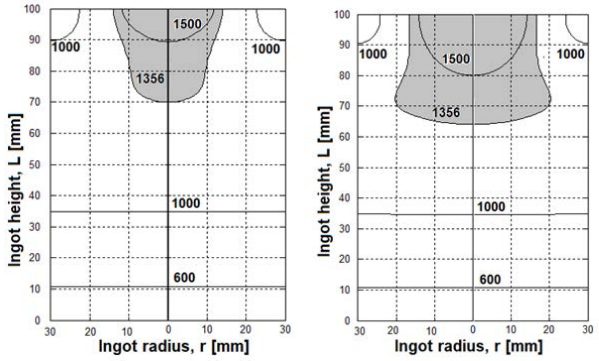
The copper material characteristics used for the simulations are presented in Table 1.

In Fig. 3 and Fig. 4 are presented the temperature distribution simulation results in a cylindrical ingot with height of 100 mm and diameter 60 mm.

Table 1

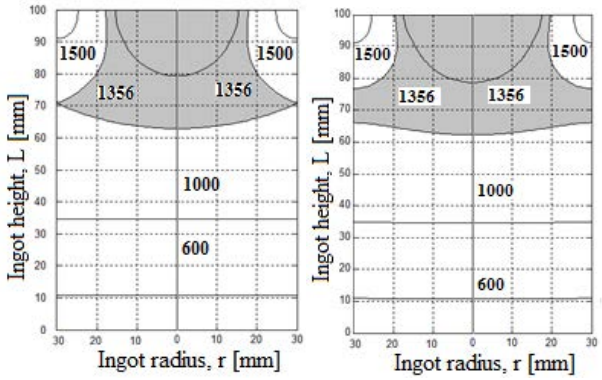
Material characteristics of Cu

Parameters	*Values*	*Dimension*
Thermal conductivity λ_i	318.1 (at 1280 K)	[W/m.K]
Melting temperature T_m	1356	[K]
Heat capacity C_p	0.38	[J/g.K]
Thermal diffusivity a	1.13×10^{-4}	[m^2/s]
Heat content $C_p.\rho.T_m$	4612	[J/cm^3]



a) *b)*

Fig.3. The temperature couture lines of EBM of Cu at: a) input electron beam power Z_1 = 10 kW, electron beam radius Z_2 =10 mm and crystallization speed Z_3 = 6 mm/min b) input electron beam power Z_1 = 16 kW, radius of the beam Z_2 =5 mm and crystallization speed Z_3 = 6 mm/min.



a) *b)*

Fig.4. The temperature couture lines of EBM of Cu at: a) input electron beam power Z_1 = 18.630 kW, electron beam radius Z_2 =10 mm and crystallization speed Z_3 = 6 mm/min b) input electron beam power Z_1 = 22 kW, radius of the beam Z_2 =15 mm and crystallization speed Z_3 = 3 mm/min.

It can be seen, that different molten pool shapes (the colored areas) are obtained at variation of the electron beam power, the radius of the electron beam and the crystallization velocity. In the cases, when the process parameters are appropriate, the molten pool crystallization surface can reach the outer wall of the ingot. The result is the increase of the roughness of the ingot walls, as well as obtaining more flat surface of the crystallization front, which helps the inclusions' removal toward the surface of the ingot and parallel to the ingot axis dendrite structure.

Experimental conditions

Simulated experiments are conducted on the base of the quasi steady-state two-dimensional heat model (1). Experiment was conducted to study the impact of process parameters on the geometry of the molten pool. Parameters that are changed during simulations are: Z_1 [kW] is the input electron beam power, Z_2 [mm] is the radius of the electron beam and Z_3 [mm/min] is the casting velocity (crystallization speed).

The process parameter variation regions for this electron beam melting experiment are presented in Table 2.

Table 2

Process parameter variation regions

Factor (Z_i)	*Dimension*	*Coded*	*Lower level ($Z_{min,i}$)*	*Upper level ($Z_{max,i}$)*
Z_1	kW	x_1	10	22
Z_2	mm	x_2	5	15
Z_3	mm/min	x_3	3	9

The data processing is carried out in a coded scale, in order to avoid problems related to a possible multicollinearity or other numerical problems in the evaluation of the coefficients of regression and bad prediction values of the quality indicators. The coded significations of the process parameters are given also in Table 2. The transformation from natural (Z_i) to coded (x_i) in the range from -1 to 1 values of the process parameters is done using the formula:

$$x_i = \frac{2Z_i-(Z_{max,i}+Z_{min,i})}{Z_{max,i}-Z_{min,i}}. \quad (6)$$

An augmented optimal composite design with one point in the center of the design is chosen and presented in coded and natural values in Table 3.

Modelling of the molten pool shape

The experimental design in Table 3 was conducted in order to investigate the temperature distributions in the cast ingot, obtained at different sets of the process parameters. The considered performance characteristics of the molten pool geometry are (Fig. 5): molten pool depth H (in the center of the ingot) the molten pool half-width B, the volume of the molten metal V, the height of the interface molten metal/crucible H_1 and the height of molten metal below the interface molten metal/crucible dH (if the molten pool has reached the outer wall of the ingot).

Table 3

Experimental design

№	x_1	x_2	x_3	Z_1	Z_2	Z_3
1	-1	-1	-1	10	5	3
2	+1	-1	-1	22	5	3
3	-1	+1	-1	10	15	3
4	+1	+1	-1	22	15	3
5	-1	-1	+1	10	5	9
6	+1	-1	+1	22	5	9
7	-1	+1	+1	10	15	9
8	+1	+1	+1	22	15	9
9	-1	0	0	10	10	6
10	+1	0	0	22	10	6
11	0	-1	0	16	5	6
12	0	+1	0	16	15	6
13	0	0	-1	16	10	3
14	0	0	+1	16	10	9
15	0	0	0	16	10	6

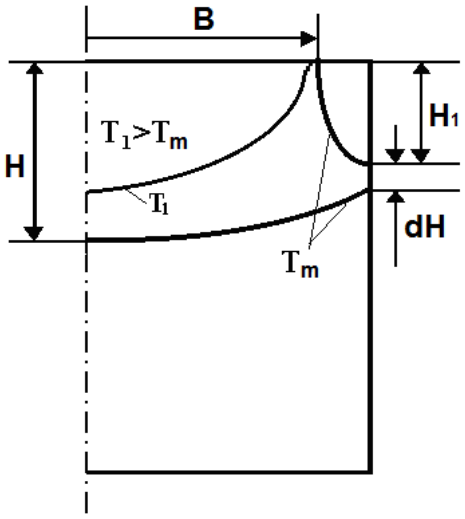


Fig.5. Molten pool geometry parameters.

Regression models are estimated for the volume of *V*, the depth *H* and the half-width *B* of the molten pool are presented in Table 4. The estimated regression models (Table 4) for the heights H_1 and *dH* (Fig. 5) are estimated after adding some additional experiments for clarification of the process parameter regions, where a molten zone below the solidified ring at the interface with the water cooled crucible appears.

In Table 4 together with the estimated regression models the corresponding determination coefficients R^2 (the square of the multiple correlation coefficient) and the adjusted R^2_{adj}, are presented also. They are measures for the accuracy of the models. The regression models are very good and they are implemented to study the dependencies of the obtained molten zone shapes on the process parameter variations.

Table 4

Regression models for the performance characteristics of the molten pool geometry

	Regression models	R^2, %	R^2_{adj} ,%
V	$36.975833+28.54335x_1-2.4725528x_2-2.9262405x_3+5.8807452x_1^2-3.0546324x_2^2-1.1358349x_1x_3$	99.62	99.48
B	$16.986267+2.2705905x_1+0.68586266x_2-0.32698233x_1^2+0.52430426x_2^2-0.24371278x_1x_2$	98.57	98.17
H	$36.248352+4.5351085x_1-1.3954508x_2-0.81773939x_3-1.581328x_1^2-0.89161009x_2^2+1.1045306x_1x_2+0.83676055x_1x_3$	98.37	97.66
dH	$-0.14181592-0.35336612x_2-0.7194417x_3+5.601067x_1^2-0.85137706x_2^2+6.3189943x_1^3-0.41915083x_1x_2-0.91706797x_1x_3-1.8686626x_1x_2^2$	98.07	96.96
H_1	$0.87489665+60.908981x_1-3.1486879x_3+10.482764x_1^2-49.508053x_1^3+3.5534275x_1^2x_3$	97.19	96.36

In Figs. 6-8 are presented contour plots of the molten pool depth *H*, the molten pool half width *B* and the volume of the molten metal *V*, depending on electron beam power (Z_1) and beam radius (Z_2) at crystallization velocity Z_3 = 6 mm/min.

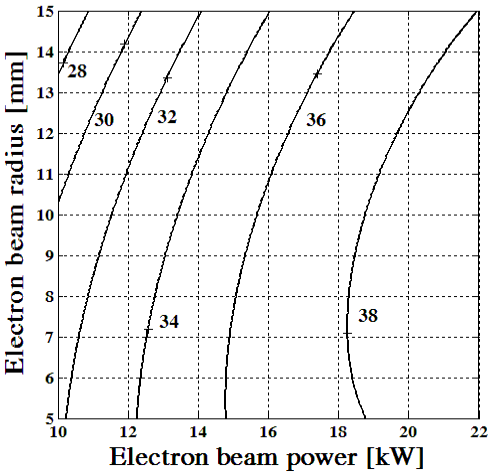


Fig.6. Contour plot of the molten pool depth H (mm), depending on electron beam power (Z_1) and beam radius (Z_2) at crystallization velocity Z_3 = 6 mm/min.

It can be seen that the volume of the molten pool depends mainly on the increase of the electron beam power, while the electron beam radius influences more on the values of the molten pool depth and half width. The desired form of the molten pool can be obtained through properly defined optimization requirements.

In Fig. 9 are presented the obtained contour plots

of the height of the molten zone *dH*, depending on electron beam power (Z_1) and beam radius (Z_2) at crystallization velocities Z_3 = 3 mm/min, Z_3 = 6 mm/min and Z_3 = 9 mm/min. The colored zones define process parameter combinations, which determine molten pools not reaching the outer walls of the ingots.

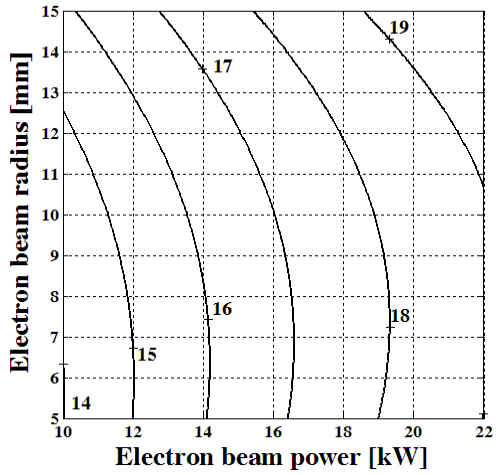


Fig.7. Contour plot of the molten pool half width B (mm), depending on electron beam power (Z_1) and beam radius (Z_2) at crystallization velocity Z_3= 6 mm/min.

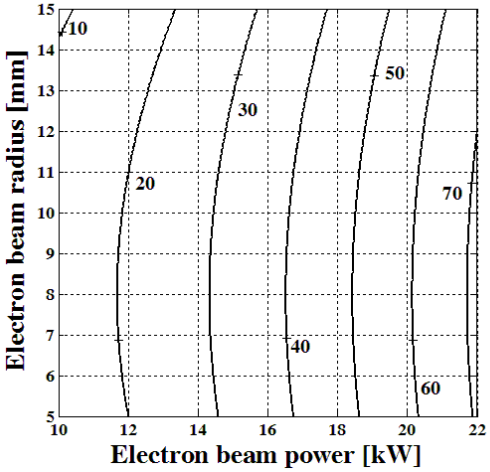
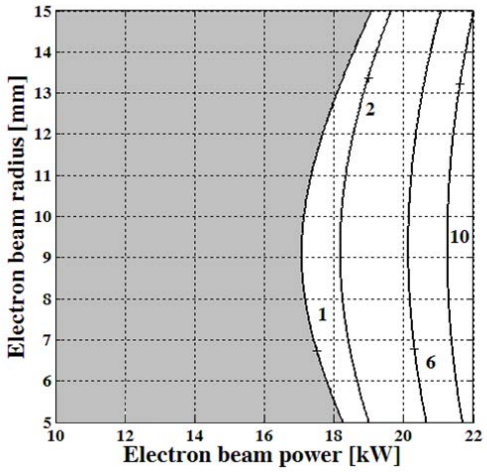
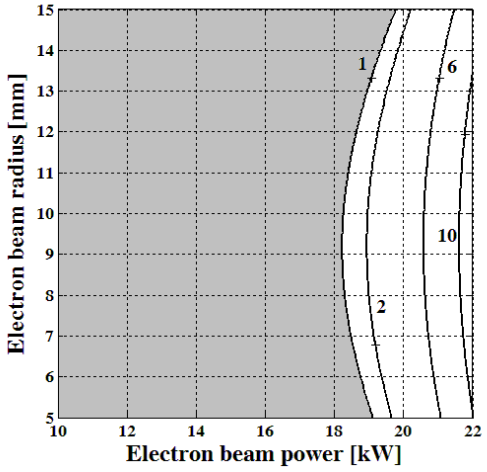


Fig.8. Contour plot of the volume of the molten metal V (mm^3), depending on electron beam power (Z_1) and beam radius (Z_2) at crystallization velocity Z_3 = 6 mm/min.

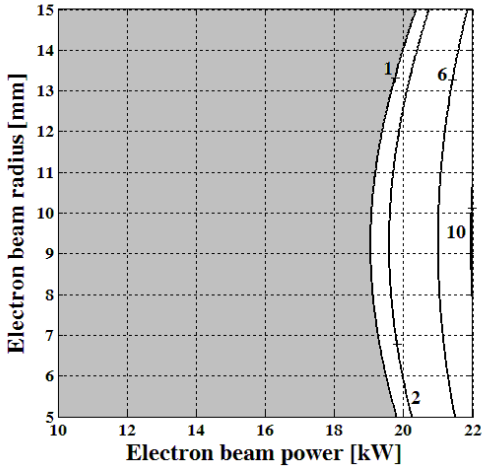
From the obtained results it can be seen, that there is certain level of the molten pool volume, which determines the appearance of the molten ring (with height *dH*) below the solidified ring at the interface with the water cooled crucible. The minimal molten metal volume is of order 53 mm^3. The height of the solidified ring at the interface with the crucible H_1, on the other hand, is not influenced by the electron beam



a)



b)



c)

Fig.9. Contour plots of the height of the molten zone dH (mm), depending on electron beam power (Z_1) and beam radius (Z_2) at crystallization velocity: a) Z_3 = 3 mm/min, b) Z_3 = 6 mm/min, c) Z_3 = 9 mm/min.

radius. At the chosen experimental process parameter regions the values vary in the region between 28 and 22 mm and it is decreasing with the increase of the molten pool volume.

In order to choose the minimal electron beam power (P_{min}), needed for obtaining the necessary molten pool volume that determines the appearance of the molten ring (like in Fig. 4a), at certain combinations of the electron beam radius (Z_2) and the casting velocity (Z_3), the following regression equation is estimated:

(7) $P_{min} = 18.606667+0.49166667x_2+0.84333333x_3 +1.0683333x_2^2$

The determination coefficients are: R^2 = 99.60% and the adjusted R^2_{adj} = 99.36% and the estimated model can be used for prediction and optimization. In Fig. 10 the contour plot of the minimal electron beam power P_{min} (kW) as a function of the electron beam radius (Z_2) and the casting velocity (Z_3) is presented. The expected values of the height *dH* at all the process parameter combinations ($Z_1 = P_{min}$, Z_2 and Z_3) is equal to 1 mm (like in Fig. 4a).

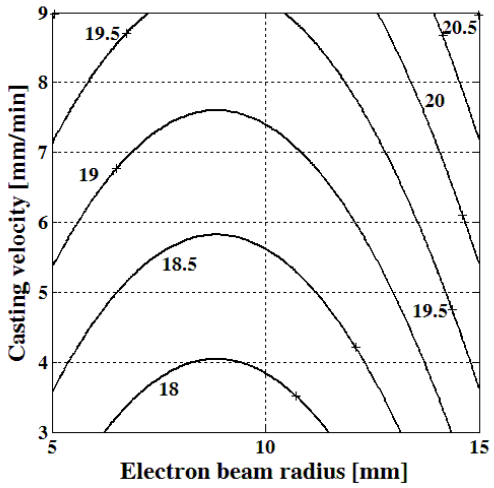


Fig.10. Contour plot of the minimal electron beam power P_{min} (kW) as a function of the electron beam radius (Z_2) and the casting velocity (Z_3).

Conclusion

In this paper simulation results from the temperature distributions in the cast copper ingots through electron beam melting and refining are implemented for the estimation of regression models for the dependence of the shape of the crystallization front (molten depth, width and volume of the molten metal) on the variation of the process parameters - electron beam power, beam radius and the casting velocity.

The obtained results show that, if the aim of the improvement of the electron beam melting process is to obtain more flat crystallization front by its reaching the outer wall of the ingot, it is necessary to obtain certain minimal volume of the molten pool *V*. It can be done by choosing the working regime for the electron beam radius and the casting velocity and then setting the electron beam power to its minimal value (Fig. 10) or higher.

There are other process conditions (not considered in this paper) that can influence the shape of the molten pool and in this way the dendrite structure, the refining process and the quality of the obtained ingot: the power distribution of the electron beam (here it is assumed to be uniform) and the electron beam movement along different trajectories. The detailed knowledge of the role of the beam scan trajectory and the frequency in every concrete case can be used to control the melt composition and structure. That knowledge could allow production of EBMR ingots from scrap materials with significant savings.

Acknowledgements

This research was supported by the Bulgarian National Fund "Scientific Investigations" under contracts KP-06-N27/18 and DN17/9.

REFERENCES

[1] Mladenov, G. M., E. G. Koleva, K. Zh. Vutova and V. I. Vasileva. Experimental and theoretical studies of electron beam melting and refining. Practical Aspects and Applications of Electron Beam Irradiation Transworld Research Network, India, Editors: Monica R. Nemtanu and Mirela Brasoveanu, 2011, pp.43-93.

[2] Mladenov, G. Electron and Ion Technologies, Academic Publ. House, Sofia, (in Bulgarian), 2009.

[3] Koleva, E., V. Vassileva, K. Vutova, G. Mladenov. Process analysis and quality control at EBMR of refractory and reactive metals. Proc. of the Intern. Conf. on High-Power Electron Beam Technology, Harrah's Reno, Nevada USA, 2006, pp.5/1-11.

[4] Koleva, E., V. Vassileva, K. Vutova, G. Mladenov. Electron beam melting and relining of refractory and reactive metals. "E+E", issue 5-6, 2006, pp.136-140.

Dipl. Mag. Tsvetomira S. Tsonevska *- PhD student to Institute of Electronics – Bulgarian Academy of Sciences, Bulgaria. Area of Interest - improving and optimization technological and electron beam processes on the basis of statistical methods and optimization. She is currently working at chief expert in manufacturing processes in the pharmaceutical company.*

e-mail: tsonevska@abv.bg

Assoc. Prof. Dr. Eng. Elena G. Koleva - *Institute of Electronics, laboratory "Physical problems of electron beam technologies" – Bulgarian Academy of Sciences, Bulgaria. Assoc. Prof and lecturer at University of Chemical Technology and Metallurgy – Sofia, Bulgaria. Director of Technological Center on Electron Beam and Plasma Technologies and Techniques – TC EPTT Ltd., Bulgaria. Scientific research areas: electron beam technologies – welding, melting and refining, lithography, selective melting, surface modification, electron beam characterization, automation, modeling, optimization, standardization.*

e-mail: eligeorg@abv.bg

Assist. Prof. Mag. Eng. Lilyana St. Koleva *- She is an Assistant at the University of Chemical Technology and Metallurgy, Sofia. Her fields of interest are: modeling, parameter optimization, automation, electron beam technologies.*

e-mail: sura@abv.bg

Corr. Memb. of BAS, Prof. DSc. Georgi M. Mladenov *- Institute of Electronics – Bulgarian Academy of Sciences, Bulgaria. Technological Center on Electron Beam and Plasma Technologies and Techniques, Bulgaria. He is the author of 10 books, 27 inventions and more than 350 articles. His research interests include electron beam microscope accelerators, electron beam technologies, electron device physics, electron beam welding, melting and refining of metals in vacuum, electron spectroscopy simulation, electron lithography, additive technologies vacuum technology.*

e-mail: gmmladenov@abv.bg

Received on: 30.04.2019

ЕЛЕКТРОТЕХНИКА И ЕЛЕКТРОНИКА Е+Е

54 год. 3-4/2019

Научно-техническо списание

Издание на: **Съюза по електроника, електротехника и съобщения /СЕЕС/**

Адрес:
ул. “Раковски” № 108
ет. 5, стая 506
София 1000
тел.: +359 2 987 97 67
e-mail: epluse@mail.bg
http://epluse.fnts.bg

ISSN 0861-4717

С Ъ Д Ъ Р Ж А Н И Е

През 2019 г. списание Electrotechnica & Electronica се издава с финансовата подкрепа на фонд „Научни изследвания“ към Министерството на образованието и науката.

ФНТС
1885

FEDERATION OF SCIENTIFIC UNIONS –
MORE THAN 130 YEARS HISTORY

ФЕДЕРАЦИЯ НА НАУЧНО-ТЕХНИЧЕСКИТЕ СЪЮЗИ –
ПОВЕЧЕ ОТ 130 ГОДИНИ ИСТОРИЯ

Зали за събития, обучения и концерти в идеалния център

Rooms for events, trainings and concerts in the ideal centre

Sofia, 108 G.S. Rakovski St | София, ул. "Г.С. Раковски" 108 | phone/тел: 02 987 7230 | www.fnts.bg