Information technology applications

ITA

1/2024

PANEURÓPSKA VYSOKÁ ŠKOLA
FAKULTA INFORMATIKY

(ITA) International Journal of Information Technology Applications

Volume 13, Number 1, July 2024

AIMS AND SCOPE OF ITA

The primary aim of the International Journal of Information Technology Applications (ITA) is to publish high-quality papers of new development and trends, novel techniques, approaches and innovative methodologies of information technology applications in the broad areas. The International Journal of ITA is published twice a year. Each paper is refereed by two international reviewers. Accepted papers will be available online with no publication fee for authors. The International Journal of ITA is being prepared for the bibliographic scientific databases.

(ITA) International Journal of Information Technology Applications

Instructions for authors

The International Journal of Information Technology Applications is welcoming contributions related with the journal´s scope. Scientific articles in the range approximately 10 standard pages are reviewed by two international reviewers. Reports up to 5 standard pages and information notices in range approximately 1 standard page are accepted after the decision of editorial board. Contributions should be submitted via e-mail to the editorial office. The language of contributions is English. Text design should preserve the layout of the template file, which may be downloaded from the webpage of journal. Papers in this journal are provided under diamond acess policy (no fee, open website) and authors declare their own Creative Commons license.

Deadlines of two standard issues per year (special issues are possible)

paper submission deadline	– continuous process
review decision	– continuous process
release date	– July/December

Editorial office address

Faculty of Informatics, Pan-European University, Tematínska 10, 851 05 Bratislava, Slovakia
juraj.stefanovic@paneurouni.com

Published by

Pan-European University, Slovakia, https://www.paneurouni.com/
Paneurópska vysoká škola, n.o., Tomášikova 20, 821 02 Bratislava, IČO 36 077 429
Civil Association EDUCATION-SCIENCE-RESEARCH, Slovakia, https://v-v-v.eu/
OZ VZDELÁVANIE -VEDA-VÝSKUM, Andrusovova 5, 851 01 Bratislava,
IČO 42 255 180
Slovak Society for Cybernetics and Informatics (SSKI) at the Slovak Academy of Sciences, Slovakia
Slovenská spoločnosť pre kybernetiku a informatiku pri SAV (SSKI),
Ústav automobilovej mechatroniky, Fakulta elektrotechniky a informatiky STU,
Ilkovičova 3, 812 19 Bratislava 1, IČO 00 178 730, https://www.sski.sk/

Electronic online version of journal

https://www.itajournal.com/
(Open Journal System)

visit Archive and Instructions for authors:

Electronic backup and preservation

www.webdepozit.sk

digitálne pramene

Print version

Multigrafika s.r.o., Rajecká 13, 821 07 Bratislava

Print issues

Contact the editorial office,
print issues are available until they are in stock.

ISSN: 2453-7497 (online)
ISSN: 1338-6468 (print version)

Registration No.: EV 4528/12

Contents

Editorial

Dear readers,

Thank you for your contributions, we welcome your latest research results and solutions and we look forward to further collaborations.

Ing. Juraj Štefanovič, PhD.
ITA Executive editor

This time, we would like to announce this next event in winter:

32th International Conference Cybernetics & Informatics 2025

K&I '25 Mikulov

Mikulov, Czechia, February 2-5, 2025
International Conference | Cybernetics & Informatics 2025 - Mikulov (sski.sk)

organised by Slovak Society of Cybernetics and Informatics
under the auspices of Faculty of Electrical Engineering and Information Technology, Slovak University of Technology in Bratislava, Slovakia
and Faculty of Informatics, Pan-European University in Bratislava, Slovakia

Draft paper submission:	November 4, 2024
Notification of acceptance:	December 16, 2024
Final paper submission:	January 8, 2025

The conference focuses on presentation of latest development in control engineering, information technologies and related multidisciplinary fields in line with current development trends in Industrial Internet of Things. Papers are invited within the following fields:

- Methods and algorithms for modelling and control
- New information and communication systems
- Embedded, distributed and networked control systems
- Cyber-physical systems, cloud computing, Big Data and extended reality
- Artificial Intelligence

International Journal
Information Technology Applications (ITA)
Volume 13, Number 1, September 2024

HISTORY OF RESEARCH, DEVELOPMENT AND APPLICATION OF COMPUTER NETWORKS IN SLOVAKIA

Štefan Schill, Štefan Kozák

Abstract:

The paper focuses on the description and characteristics of research, development and application of computer networks in the Slovak Republic, following the development of computer and telecommunication technology since the seventies. In the seventies of the last century, research and development of technical and software means for the creation and use of computer networks was initiated in Slovakia at the Institute of Applied Cybernetics (UAK). The result of these efforts was the launch of the first UAKNET computer network in the countries of Eastern Europe in 1984. The computer network used datagram technology. It was used primarily by professionals in the field of research and development to exchange information and to access library information. On the basis of the experience gained, there was a growing demand for interconnection with foreign computer networks. This triggered the need to switch to X.25 technology.

Keywords:

Computer, computer networks, software development, node computers, host computers, front end computers, terminals, datagram technology, X.25 technology, Internet.

Introduction

Computers originated in the first half of the 1940s in the USA with the development of the atomic bomb and the requirements of industrial modernization. The existence of computers sparked the continuation of research and development in computing and computer technology mainly in the USA and Germany. The development of telecommunication technology was strongly influenced by space research. The use of computing also presupposed the availability of appropriate software and, last but not least, skilled personnel who were in a position to develop software for the field. A prerequisite for the emergence and development of computer networks was the necessary requirement to have computing at a certain higher level. Another prerequisite for the development of computer networks was modern telecommunications technology.

In Slovakia, the Department of Computers at Slovak Technical University in Bratislava (today: Slovak University of Technology) was the first to deal with the preparation of people for the field of computer technology. This department had computers at its disposal since the early 1960s (type Ural 2 and LGP 30). To a lesser extent, this issue was also addressed by the institutes of the Slovak Academy of Sciences and other universities in Slovakia. Experts for this field were mainly trained in research organisations. These included the Institute of Technical Cybernetics at the Slovak Academy of Sciences, the Research Institute of Computing in Žilina, and the United Nations Research Computing Centre (INFOSTAT) in Bratislava, which already in 1969 had at its disposal a modern powerful Control Data 3300 computer equipped with relatively good communication technology, which at that time already allowed remote input of programs and printing of outputs. A research group on remote data transmission and processing was also set up at the institute.

In 1971, the Slovak Commission for Scientific, Technical and Investment Development (actually a ministry) established the Institute of Applied Cybernetics (ÚAK). The main focus of the ÚAK was research and development of software systems for the national economy. Another area was computing, mainly focusing on the research, construction and application of computer networks.

1 Research and Development of Computer Networks in Slovakia in 1975-1990

The only research centre for computer networking in Slovakia and Czechoslovakia was the institute of Applied Cybernetics (UAK). In this period, computers with communication technology were not produced in Eastern European countries. Another disadvantage was that Western companies, due to the embargo, could supply less powerful computing equipment without the means to transmit data. In Slovakia at that time, east-European EC computers compatible with IBM 360 and SMEP (small computers) compatible with DEC products were available. The SMEP computers manufactured in Slovakia were also produced with communication devices and elements, which created suitable conditions for building data and computer networks. As far as potential users were concerned, there was no interest in data communication at that time. Even the telecommunications in Slovakia did not realise the economic importance of data communication at the beginning and did not create the necessary technical prerequisites for its development.

In the late 1970s, a research group of experts at the UAK began to deal with the problem of computer communication. The aim was to build a computer network that could be used in the future mainly for scientific purposes. At that time there were no technical means for computer network operation available in Slovakia. There was also no network software available and there was a lack of experience in the operation of computer networks. The main objective was to develop network software for equipment produced in Eastern European countries. For hardware development was limited to add-on devices. The research and development resulted in the design of the network architecture, the development of software for the node computer, for the terminal concentrator, the front-end computer, the network metering system and the intermediate link to connect the SMEP computers to the EC computer channel. The following first-generation products were used to build the experimental computer network. The experimental computer network was put into operation in 1984. The Department of Telecommunications made fixed lines available free of charge. The main objective of the network was to allow direct access to databases in Bratislava and Prague. The main users were universities, research institutes of the Slovak Academy and the Czech Academy of Sciences. Among the first users were also research institutes and several manufacturing companies. The UAK Computer Network (UAKNET) was the first computer network in the countries of the east-European Council for Mutual Economic Assistance (RVHP). Based on its objectives and the composition of its users, it was a research network.

In 1988 a connection to the EUNET network in Vienna was implemented. The connection was made via a switched telephone line twice a day. Information was downloaded from the EUNET network - mail for UANET users to access and use. Their "documents", which were intended for EUNET subscribers, were stored on UAKNET and sent twice a day during the telephone connection to EUNET. The configuration of the UAKNET network, including the subscribers at the time, is shown in (Fig.1).

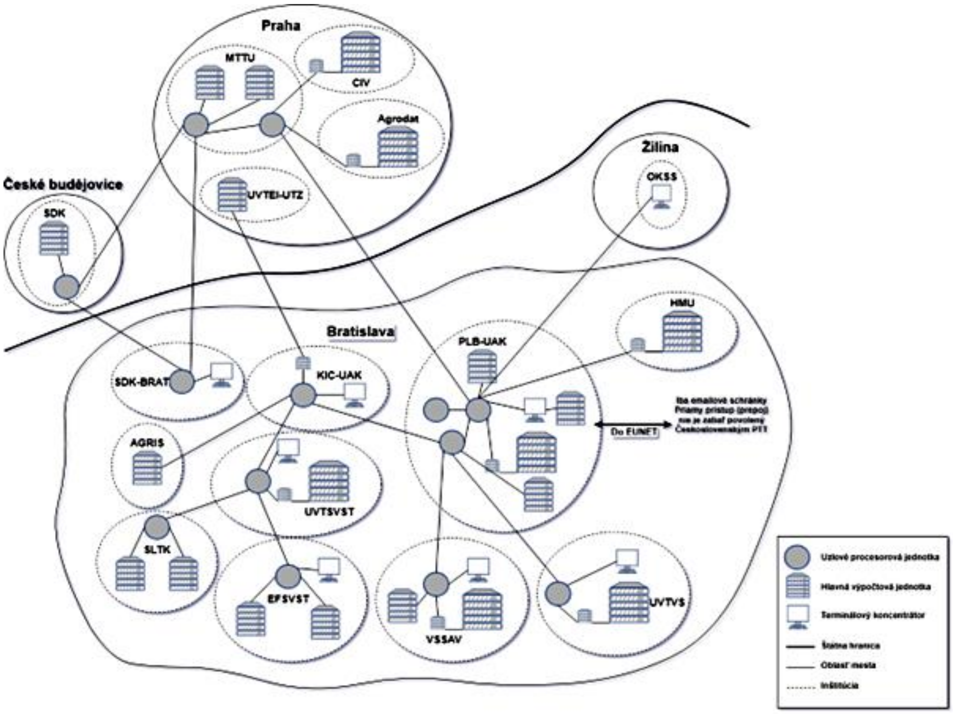


Fig.1. Netwok UAKNET configuration.
Symbols: node, computer center, terminal hub.
Lines: state border, city border (Praha, etc.), institution area.

The following organisations were connected to the UAKNET computer site:

- Institute of Applied Mathematics and Computer Technology of Comenius University (UVTVS)
- Faculty of Electrical Engineering (EF SVŠT)
- Slovak Technical Library (SLTK)
- Information Centre of Agriculture and Food Industry (AGRIIS and AGRODAT)
- Long Distance Cable Administration - Telecommunications (SDK)
- Research Institute of Telecommunications (MTTU)
- Information Centre of the Federal Office for Patents and Inventions (CIV)
- Institute of Scientific, Technical and Economic Information - Central Technical Base (UVTEI-ÚTZ)
- Bratislava Long Distance Cable Administration (SDK BRAT)
- Computer Laboratory of the IAC (PLB-ÚAK)
- Communication and Information Centre of ÚAK (KIC-ÚAK)
- Institute of Computer Technology of the Slovak University of Technology (UVT SVŠT)
- Computer Centre of the Slovak Academy of Sciences (VSSAV)
- Slovak Hydrometeorological Institute (HMU)
- Žilina District Communications Administration (OKSS)

2 UAKNET's Entry on the International Scene

Thanks to the cooperation and contacts with the RWTH Technical University in Aachen, West Germany, in December 1989, a representative of Slovakia (Mr. Schill) was invited to West Berlin for the annual meeting of the DFN Verein (Association of Research Networks of Germany), where he presented the state of the art of computer networking in Czechoslovakia and especially in Slovakia, became its representative, and introduced the participants to the NET network (Schill, 1990). On the basis of the information translated by him, Czechoslovakia was admitted to the association of operators of corporate, university and research networks RARE Working group 8, later TERENA .

The Association subsequently invited a representative of the Czechoslovak Republic (CSSR) to present the state of computer network development in the Slovak Republic at the 1990 annual meeting in Killarney, Ireland (Schill, 1990). The paper presented there was published in the professional journal Computer Networks and ISDN Systems. In the following year, another expert meeting was again held for the purpose of building modern computer networks. This meeting featured a discussion by the world's leading computer networking expert from the USA - Vint Cerf, who had built and introduced the Pentagon's DARPA-NET computer network in the early 1970s. In his paper, he recommended that the isolated computer networks then in operation should be interconnected by operators using IP addressing. With this paper (May 1991) he initiated the worldwide build-up of the INTERNET network, especially in the USA and Europe.

3 Further Development and Use of Computer Networks in Slovakia

The political changes after 1989 brought significant changes also in the field of computing. With the lifting of the embargo on the import of more powerful computing equipment from Western countries, professional literature became more accessible and experts also in the field of computing were able to travel to industrially developed countries and learn on the spot about the state of "networking" in Western Europe. For the UAK, this meant the legalisation of cooperation with EUNET and the possibility to connect to networks in Western Europe. This resulted in an increase in the number of NET users. As UAK was a budgetary organisation, it was not able to finance the operation of the network from its own resources. It therefore introduced user charges.

Connection to the Internet was only possible with X.25 technology. For this reason, UAK concentrated all its possible capacities on the development of X.25 technology. Slovak Telecommunications also realised the economic importance of the provision of its services for the operation of computer networks and therefore intensified its work on the creation of a data network based on X.25 technology. It set itself the goal of commissioning a public data network in 1992. It should also be noted that in 1992, the UAK won a tender for the construction of a computer network for the Government bodies of the Slovak Republic. The computer network was built and put into operation under the name GOVNET in mid-1993.

Research, development and operation of the UAKNET network was carried out until mid-1995, when the managers of the Statistical Office of the Slovak Republic, under which the UAK belonged, decided to abolish it as a legal entity as of January 1st, 1996. Its activities, assets and staff were transferred to INFOSAT body. This sealed the future fate of computer network research and development. However, the main focus of INFOSTAT's activities was statistics. Shortly after this event, the operation of GOVNET was "handed over" to a private company. After the merger, a substantial part of the UAK staff moved to various other research organisations in Slovakia.

Conclusion

The research, development, design and operation of the computer network at the IAC was of considerable importance for the development of computer communication in Slovakia. It was here that experts were "grown up" who significantly influenced the development and use of computer technology in the future. The existence of UAKNET also resulted in the fact that Telecommunications also realized the economic importance of computer communication and began to intensively build communication channels using X.25 technology. These facts significantly influenced and accelerated the introduction of the INTERNET network into operation in Slovakia.

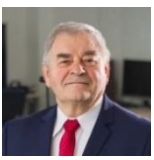
References

[1] Stefan Schill: CSSR - Hoffnung auf die Zukunft (CSSR - The Hope for the Future). Annual Meeting, Verein, West Berlin. December 1989, in DFN Mitteilung, Berlin 1990.

[2] Stefan Schill: Networking in Czechoslovakia: Efforts and Results, Computer Networks and ISDN Systems, 19, (1990), pp. 186-188

Authors

Ing. Štefan Schill
Slovakia
stefan@schill.sk
Graduated in Automation Technology at the Faculty of Electrical Engineering of the Slovak University of technology in Bratislava in 1966. In 1966-1967 he worked at the Department of Computers of Faculty of Electrical Engineering EF STU (today: www.fei.stuba.sk). In 1968-1980 he was involved in research and development at the Research Computing Centre of the United Nations Programme - now INFOSTAT. From 1981 -1995 he was head of the computing laboratory and researcher in the field of computer networks at the Institute of Applied Cybernetics in Bratislava.



prof. Ing. Štefan Kozák, PhD.
Faculty of Informatics,
Pan-European University in Bratislava, Slovakia
stefan.kozak@paneurouni.com
His research interests include system theory, linear and nonlinear control methods, numerical methods, algorithms and software for modeling, control, signal processing, IoT, IIoT and embedded intelligent systems applications in a wide range of industrial, healthcare, banking and service sectors.

International Journal
Information Technology Applications (ITA)

Volume 13, Number 1, September 2024

HYDROGEN FUEL CELL SYSTEM MODELLING APPROACHES

Rastislav Putala, Viktor Ferencey

Abstract:

This article describes methods to model the functions of a PEM fuel cell stack system. Fuel cells are electrochemical devices that convert chemical energy of inputs reactants into output electrical energy. The proton exchange membrane fuel cell (PEM FC) is considered one of the most promising candidates for next-generation energy sources due to its high power density, zero emissions, and low operating temperature. Modeling has gained enormous attention and interest in recent years. This paper briefly reviews the types of PEM FC models such as Empirical/semi-empirical, analytical and mechanistic models, zero to three-dimensional models, multiphase flow models including optimization methods in PEM fuel cell models. In particular, this study focuses on the simulation of a standard semi-empirical mathematical model for an air-cooled PEM FC, utilizing the Simscape Electrical toolbox in the Matlab Simulink environment. The processes by which this model simulates the behavior of an air-cooled PEM FC are described in detail, illustrating the dynamic interactions between key physical and electrochemical parameters. This paper highlights the importance of combining different parameter modeling strategies in PEM FC modeling in order to achieve accurate results and reduce experimental costs.

Keywords:

Fuel cell stacks, fuel cell power system, distributed power generation, current converter, modelling, empirical/semi-empirical, multiphase flow model, MATLAB/Simulink.

Introduction

Global climate changes have been raising a great amount of concerns since the last two decades. The planet's average surface temperature has risen to about 1.18 °C above the late 19th century level [3]. In fact, scientists have been largely attributing such climate changes to the expansion of greenhouse gas emission, particularly carbon dioxide emission, which have been largely generated due to human activities in the last 150 years. Various solutions for this pollution reduction have been proposed, such as greener fuels for internal combustion (IC), more efficient IC with more potent exhaust aftertreatment systems, electric power systems, or hybridization of IC and electric power systems. Fuel cells (FCs) are devices that convert fuels and oxidants into electricity via electrochemical reactions, that means, there is no direct combustion between fuels and oxidants. Thanks to the ability to convert chemical energy to electrical energy with low to zero emission, Proton Exchange Membrane Fuel Cell (PEM FC) is the most common choice for power generation due to its low temperature operation and fast start-up/shutdown relative to other types of fuel cells. In addition, when pure hydrogen is used as fuel, only water and heat are produced.

Models of PEM FC at different system levels have long been developed and are largely available. PEM FC model considering four phenomena, which were species transport, flow through pores under potential and pressure fields, electrochemical kinetics, and gas-phase transport.

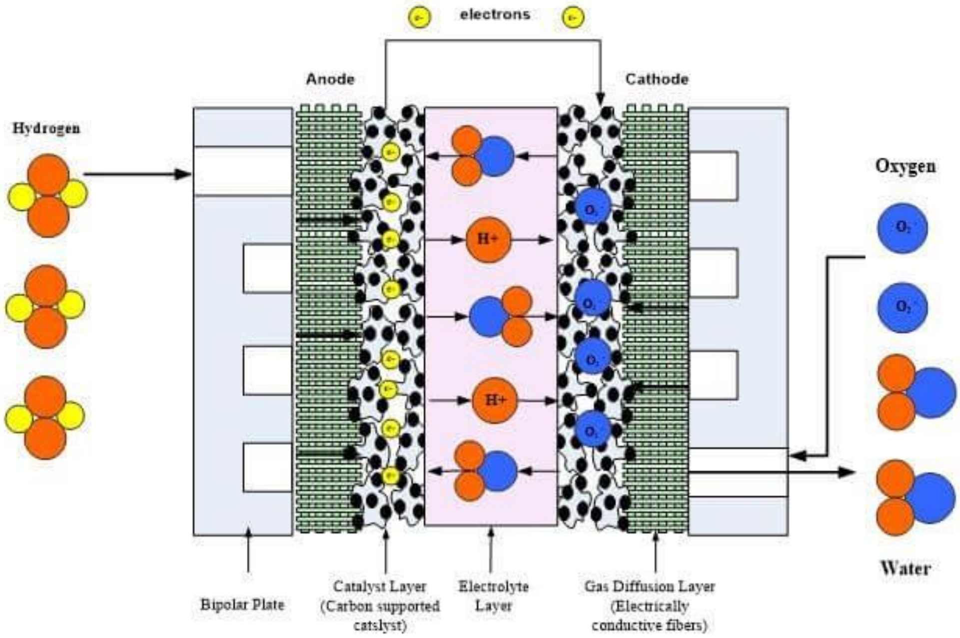


Fig.1. Diagram of a polymer electrolyte fuel cell (PEMFC) stack.

1 Fundamentals of PEM FC

A PEM FC is an appealing choice as a strong contender of an alternative clean energy generation for stationary and automotive applications. This is mainly because of PEM FC's rapid startup, low operating temperature, low pressure, high efficiency in energy conversion with zero greenhouse gas emission, and high power density [4]. The cell performance of PEM FCs is mainly determined by numerous factors including the manufacturing process, the mechanical design, the electrochemical reaction kinetics, the transport phenomena in the cells, and the operating conditions [5].

A) Basic equations for PEM FC modeling

In PEM FC hydrogen fuel H2 enters a cell through the anode flow path and reacts electrochemically at the anode - membrane interface in the reaction [15]:

$$H2 \rightarrow 2H^{+} + 2e^{-} \quad (1)$$

The electrons are attracted to conductive materials and travel to the load when needed. In the same moment, oxidant O2 is reduced at the cathode - membrane catalyst interface in the reaction [15]:

$$2H^{+} + 2e^{-} + 0{,}5O_2 \rightarrow H_2O \quad (2)$$

Oxygen reacts with protons and electrons, forming water and producing heat. Both the anode and cathode contain a catalyst to speed up the electrochemical processes, as shown in (Fig.1).

The overall reaction is [15]:

$$H_2 + 0{,}5O_2 \rightarrow H_2O + \text{electric energy} + \text{heat} \quad (3)$$

Mass, momentum, chemical species, ionic and electrical current and thermal energy are the physical quantities that are transported in a PEM FC [6]. To understand the dynamic and static behavior of fuel cells, the mathematical models help in estimating strategies of control and optimizing the design of fuel cells. By applying mathematical models to study fuel cells, there is a reduction in experimental tests, which save the time and effort. In these models, the influence of temperature distribution, thermal stress, operating conditions, and other variables could be understood simply through the simulation [6].

2 PEM FC Models

There are three types of fuel cell models in terms of parametric which are empirical/semi-empirical, analytic, and mechanistic models. The methods that have been applied recently to optimize these parametric models are available. The two other modelling approaches which explain the models in terms of dimensional and phase change are respectively zero to three-dimensional models and multiphase models.

1) Empirical – semi empirical models: The empirical – semi empirical models are designed for particular fields and should be changed for every specific operating condition or application. The polarization curve of PEM FC, which is described by 0-dimensional models are empirical and simple models [7]. Semi-empirical and 0-dimensional models are suitable for beginner researchers, who try to find a model for the fuel cell. The main emphasis of empirical/semi-empirical models is on the correlation between input and output. Besides, these models provide calculation for performance of PEM FC in a short time [8]; therefore, the empirical/semi-empirical models are more suitable for problems that focus on controlling.

2) Analytical models: The analytical models are useful for simple and one-dimensional designs with short time computing processes and the calculation of water management and voltage losses [9]. The advantage of analytical models is introducing the practical surveys for certain conditions, which provide information easily without necessitating for a numerical program to run in many cases. By utilizing just mathematical models because of nonlinear problems, it is hard to solve the PEM FC problems; though, they can be solved by numerical approaches.

3) Mechanistic models: The difficulty of modeling the PEM FC systems is because they are electrochemical, nonlinear, and multivariable systems [10]. Thermodynamic, fluid dynamics, and electrochemistry are the fields that mechanistic model focuses on. Mechanistic models explain the basic actions of a fuel cell including flow pattern, pressure drop, distribution of current density, and voltage. empirical and mechanistic techniques as combination models, predicting the performance of the PEM FCs become more effective [10].

A) Optimization methods in PEM FC models

The latest researches have obtained huge attention to the optimization of fuel cells by classifying different parameters and models like a genetic algorithm (GA) and other evolutionary computation techniques. These techniques are used to increase the precision of model parameter identification in the fuel cell.

The particle swarm optimization (PSO) is an effective algorithm, which is in comparison with GA, has better and precise performance in the convergence of problems by providing global and stable optimization solutions [10]. With the aim of solution for optimization of real problems, Modified Particle Swarm Optimization (MPSO), as a nature-inspired algorithm, is suitable for many engineering applications [11].

The Simulink technique, which is applied in Matlab-SIMULINK, is suitable for various operating conditions and different fuel cell systems. It is easy to apply for different applications; because this modeling is simple, and there is no need for extensive time for computation [12].

B) Zero to three dimensional PEMFC models

In terms of dimension there are four models for PEM FC. The dimensions of the problem, which should be zero, one, two, or three-dimensional, is dependent on the number of spatial independent variables of the differential equations. Zero-dimensional and semi-empirical PEM FC models are simple and helpful for examiners at the start point [7].

One dimensional model is the first model that researchers established to study the fuel cell. It was a complicated model with a sandwich domain in the y-direction. These models give massive particular details in different operating conditions for fuel cells by examined the temperatures, mass concentrations, electrical potentials, and fluxes [7].

Two dimensional (sandwich) model in the y-z or x-y directions is the upgraded version of the one-dimensional model. The influence of channel geometry, bipolar plates, heat, mass transfer, and fluxes on fuel cells could be examined by these two-dimensional sandwich models.

The most appropriate model to analyze PEM FC in every detail, for instance, current density distribution, the influence of flow field design on fuel cell performance, or bipolar plate blockage impact, is *three-dimensional model*, which is in the x-y-z direction [7].

Two-Phase Flow Models to understand water transport within PEM FC [13], various models based on water flow in MEA and the gas channels have been investigated in recent years. It appears that high temperature has an effect on kinetics rise and liquid water reduction and increases PEM FC's performance. Fuel cell modelling is a combination of electrochemistry and thermodynamics in porous media over transportation phenomena to the science of material [14].

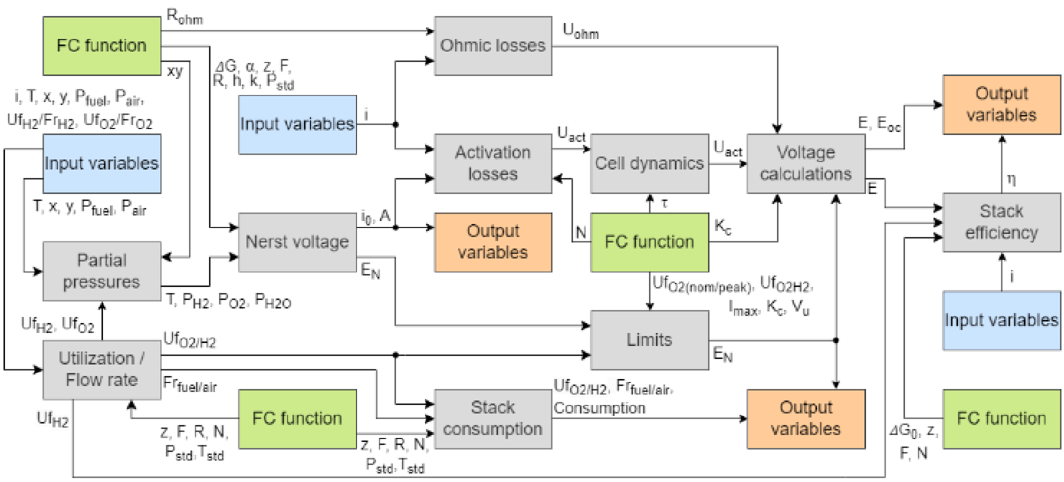


Fig.2. Process scheme of mathematical simulation PEM fuel cell model.

3 Mathematical Model of Air Cooled PEM Fuel Cell

This chapter describes a standard semi-empirical mathematical simulation model of a PEM fuel cell (Fig. 2) from Simscape Electrical in the Matlab Simulink environment. Specifically, the processes by which this model simulates the behavior of an air-cooled PEM fuel cell will be described (Fig. 3).

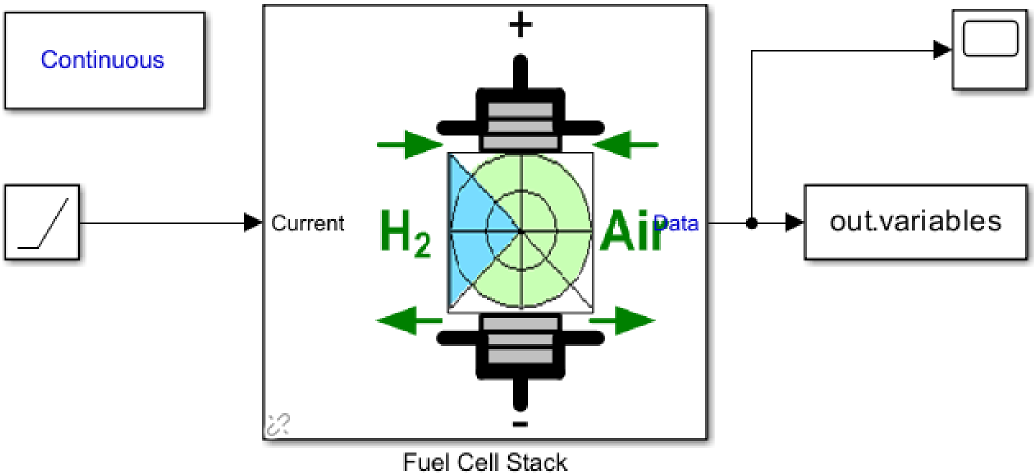


Fig.3. Described mathematical simulation model of a PEM fuel cell [2].

A) Inputs and outputs of mathematical model

The subsection describes all input and output parameters, as well as the tasks of the FC function block (function FuelCellInit()).

The FC function block has several functions. It sets the basic constants (Tab.1), approximates the missing parameters and reconstructs the polarization curve based on its 4-point description in the steady state. Also, this function has a controlling role in the selection of dynamic / steady inputs.

Table 1. Constants from function FuelCellInit() [2]

Description	Constant	Value
Boltzman constant	k	$1.381*10^{-23}$ [J*K]
Planck constant	h	$6.626*10^{-34}$ [J*s]
Faraday constant	F	96487 [C/mol]
Gas constant	R	8.314 [$J*K^{-1}*mol^{-1}$]
Standard atmospheric pressure	P_{std}	101325 [Pa]
Standard temperature	T_{std}	273 [K]
Standard water enthalpy	$\Delta h^0(H_2O(gas))$	$241.83 * 10^3$ [J/mol]
Number of transferred electrons	z	2 [-]

The FuelCellInit() function requires 4 voltage points at certain current loads (0A, 1A, I_{nom}, I_{end}) and approximation of the parameters E_{oc}, i_0, NA, and R_{ohm} to reconstruct the polarization curve. Approximation of constants α, ΔG, K and K_c is required for parameter calculation with dynamic inputs. [1] To demonstrate the functionality of this feature in these tasks, we will use the PEM values of the H-500 fuel cell as inputs and a test script:

```
clc;
close all;
clear;
```

Inputs and Constants:

```
% Definition of 4 points of the polarization curve
V0 = 23.5;          % [V] maximum voltage (0A)
V1 = 21;            % [V] voltage at 1A current
Vnom = 16.2;        % [V] voltage at ~1/2 maximum current
Vend = 13.4;        % [V] minimum voltage
Inom = 20;          % [A] ~1/2 maximum current
Iend = 40;          % [A] maximum current

% Definition of non-electrical quantities
T = 55;             % [°C] operating temperature of the system
N = 24;             % [-] number of cells
eta = 40;           % [%] nominal efficiency of the stack
Pfuel = 1.5;        % [atm] nominal fuel pressure
Pair = 1;           % [atm] nominal air pressure
Vair = 19.12;       % [l/min] nominal air flow rate
x = 99.995;         % [%] nominal air composition(H2)
y = 21;             % [%] nominal air composition(O2)
Uf_O2peak = 70;     % [%] peak oxygen utilization
Vu = 5;             % [%] stack nominal voltage undershoot

% Definition of constants
R = 8.314;          % [J*K^(-1)*mol^(-1)] gas constant
F = 96487;          % [C/mol] Faraday constant
k = 1.381*10^(-23); % [J*K] Boltzmann constant
h = 6.626*10^(-34); % [J*s] Planck constant
d_h = 241.83*10^3;  % [J/mol] standard enthalpy of water
z = 2;              % [-] number of transferred electrons
```

Inputs for FC function block are nominal and steady-state, and by default also Input variables except current value. However, all input variables can be changed to dynamic in Simulink scheme. Values of dynamic variables can be changed during the simulation run. [1], [2]

1) Ohmic Losses - Reconstruction of the Polarization Curve:

```
% Calculation of the NA value
NA = ((V1 - Vnom) * (Iend - 1) -    ...
     (V1 - Vend) * (Inom - 1)) /    ...
     (log(Inom) * (Iend - 1) -      ...
     log(Iend) * (Inom - 1));
```

```
% Calculation of ohmic resistance
Rohm = (V1 - Vnom - NA * log(Inom)) / (Inom - 1);

% Definition of current range from a value 0 to 40 with a step of 0.1
I = 0:0.1:Iend;

% Calculation of Uohm for each current
Uohm = Rohm * I;
```

2) Activation Losses - Reconstruction of the Polarization Curve:

```
% Calculation of i0
i0 = exp((V1 - V0 + Rohm) / NA);

% Calculation of Uakt for each current
Uakt = NA * log(I / i0);

% Removal of potential warnings for log(0)
Uakt(I == 0) = 0; % Set Uakt to 0 for I=0

% Calculation of total voltage
% Total voltage after subtracting both losses
U_total = V0 - (Uohm + Uakt);
% Subtracting activation losses from V0
Uakt_adj = V0 - Uakt;
% Subtracting ohmic losses from V0
Uohm_adj = V0 - Uohm;
```

3) Calculation of Charge Transfer Coefficient:

```
% Tafel slope
A = NA/N;

% Charge transfer coefficient
T_K = T + 273.15;
alfa = (R*T_K)/(z*A*F);
```

4) Calculation of Gibbs Free Energy Change:

```
% Nominal reactant utilization
eta = eta/100;
Uf_H2 = (eta*d_h*N) / (2*F*Vnom);
Pair = Pair*101325;
y = y/100;
Uf_O2 = (60000*R*T_K*N*Inom) / (2*z*F*Pair*Vair*y);

% Calculation of partial pressures
x = x/100;
P_H2 = x*(1-Uf_H2)*Pfuel;
P_O2 = y*(1-Uf_O2)*Pair/101325;

% Gibbs free energy change
K1 = (2*F*k*(P_H2+P_O2)*101325) / (h*R);
d_G = -R*T_K*log(i0/K1);
```

5) Calculation of Voltage Constant and Undershoot Constant:

```
% Voltage constant
En = 1.229 + (T_K-298.15)*(-44.43)/(z*F) + ...
     (R*T_K)/(z*F)*log(P_H2*sqrt(P_O2));
Kc = V0/En;

% Voltage undershoot constant
Vu = Vnom/100*Vu;
K = Vu / (Kc*(Uf_O2peak-Uf_O2));
```

Create the plot (Fig. 4):

```
% Plot for subtracted losses
hold on; % Allows multiple plots on the same axis
plot(I, U_total, 'r-',                        ...
     'LineWidth', 2, 'DisplayName',           ...
     'Total Voltage after Losses (U_{total})');
plot(I, Uakt_adj, 'g-',                       ...
     'LineWidth', 1, 'DisplayName',           ...
     'Voltage after Activation Losses (U_{akt})');
plot(I, Uohm_adj, 'b-',                       ...
     'LineWidth', 1, 'DisplayName',           ...
     'Voltage after Ohmic Losses (U_{ohm})');

% Setting labels and title
xlabel('Current [A]');
ylabel('Voltage [V]');
title('Voltage Analysis and Fuel Cell Losses');
grid on;
legend('Location','best'); % Display legend
hold off; % End hold mode
```

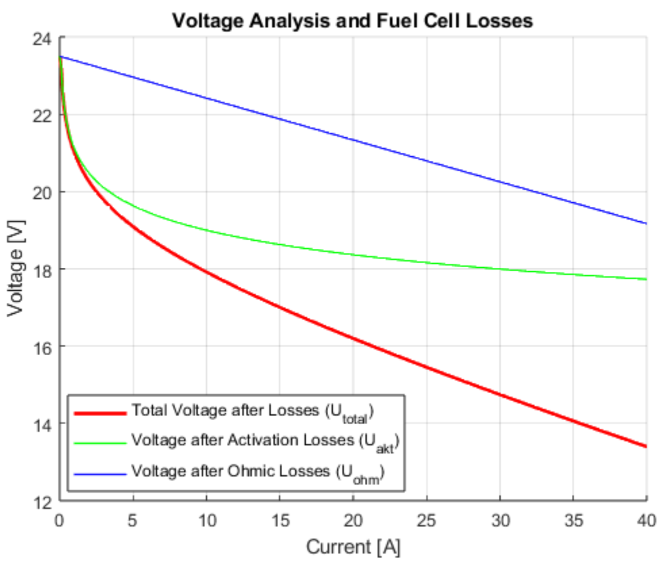


Fig.4. Voltage polarization curve and voltage losses.

Based on (Fig.4) we see that the output voltage is affected by activation and ohmic losses. Concentration losses have the greatest impact at high current loads, but a real fuel cell cannot reach these values, as it is limited by the control unit. [2]

Create a table for outputs:

```
% List of variable names
variables = {'Ohmic Resistance (Rohm)';         ...
            'Exchange Current (i0)';           ...
            'Tafel Slope (A)';                 ...
            'Charge Transfer Coefficient (a)'; ...
            'H2 Utilization (Uf_H2)';          ...
            'O2 Utilization (Uf_O2)';          ...
            'Partial Pressure of H2 (P_H2)';   ...

            'Partial Pressure of O2 (P_O2)';   ...
            'Gibbs Free Energy Change (d_G)';  ...
            'Nernst Voltage (En)';             ...
            'Voltage Constant (Kc)';           ...
            'Voltage Undershoot Constant (K)'};

% Create a column of values
values = [Rohm; i0; A; alfa; Uf_H2*100; ...
    Uf_O2*100; P_H2; P_O2; d_G; En; Kc; K];

% List of units
units = {'[ohm]'; '[A]'; '[-]';
    '[-]'; '[%]'; '[%]'; '[atm]';
    '[atm]'; '[J/mol]'; '[V]';
    '[-]'; '[-]'};

% Create the table
resultsTable = table(variables, values, units);

% Display the table
disp('Calculation Results:');
disp(resultsTable);
```

Table 2. Calculated results of test script.

variables	values	units
{'Ohmic Resistance (Rohm)' }	0.10827	{'[ohm]' }
{'Exchange Current (i0)' }	0.073375	{'[A]' }
{'Tafel Slope (A)' }	0.03815	{'[-]' }
{'Charge Transfer Coefficient (a)'}	0.37058	{'[-]' }
{'H2 Utilization (Uf_H2)' }	74.262	{'[%]' }
{'O2 Utilization (Uf_O2)' }	50.041	{'[%]' }
{'Partial Pressure of H2 (P_H2)' }	0.38605	{'[atm]' }
{'Partial Pressure of O2 (P_O2)' }	0.10491	{'[atm]' }
{'Gibbs Free Energy Change (d_G)' }	1.2888e+05	{'[J/mol]'}
{'Nernst Voltage (En)' }	1.1927	{'[V]' }
{'Voltage Constant (Kc)' }	19.703	{'[-]' }
{'Voltage Undershoot Constant (K)'}	0.00059151	{'[-]' }

(Tab.2) gives us approximate values that will serve as inputs for simulating dynamic processes in the next subsection. The output of the simulation model are the values of voltage E, current I, efficiency η, flow rate Fr, consumption, utilization Uf, slope of the Tafel curve A, exchange current i_0, Nerst voltage E_N and open circuit voltage E_{oc}.

B) Processes equations of mathematical model

The subchapter describes the equations of individual process blocks (Fig. 2) that can model a fuel cell with both steady and dynamic inputs. Individual blocks represent parts of the program in the Matlab Simulink environment.

Block Utilization / flow rate (this block has two functions – (4), (5) for steady-state / (6), (7) for dynamic inputs):

$$V_{fuel(lpm)} = \frac{60000*R*T*N*i}{z*F*P_{fuel}*P_{std}*Uf_{H2}*x} \quad (4)$$

$$V_{air(lpm)} = \frac{60000*R*T*N*i}{2*z*F*P_{air}*P_{std}*Uf_{O2}*y} \quad (5)$$

$$Uf_{H2} = \frac{60000*R*T*N*i}{z*F*P_{fuel}*P_{std}*V_{fuel(lpm)}*x} \quad (6)$$

$$Uf_{O2} = \frac{60000*R*T*N*i}{2*z*F*P_{air}*P_{std}*V_{air(lpm)}*y} \quad (7)$$

Block Partial pressures:

$$P_{H_2} = (1 - Uf_{H2}) * x * P_{fuel} \quad (8)$$

$$P_{O_2} = (1 - Uf_{O2}) * y * P_{air} \quad (9)$$

$$P_{H_2O} = (w + 2 * y * Uf_{O2}) * P_{air} \quad (10)$$

Block Nerst voltage:

$$i0 = \frac{z*F*k*P_{std}*(P_{H_2}+P_{O_2})}{R*h} * e^{\frac{-\Delta G}{R*T}} \quad (11)$$

$$A = \frac{R*T}{z*\alpha*F} \quad (12)$$

$$E_N = 1.229 + (T - 298) * \frac{-44.43}{z*F} + \frac{R*T}{z*F} * ln\left(\frac{p_{H_2}*\sqrt{p_{O_2}}}{p_{H_2O}}\right) \quad (13)$$

at: ≥ 100 °C

$$E_N = 1.229 + (T - 298) * \frac{-44.43}{z*F} + \frac{R*T}{z*F} * ln\left(p_{H_2} * \sqrt{p_{O_2}}\right) \quad (14)$$

at: < 100 °C

Block Stack consumption ((15), (16) for standard conditions):

$$\text{Fuel consumption} = \frac{60000*N*i*R*T_{std}}{z*F*P_{std}*x} \quad (15)$$

$$\text{Air consumption} = \frac{60000*N*i*R*T_{std}}{2*z*F*P_{std}*y} \quad (16)$$

The consumption of reactants for dynamic values of temperatures and pressures are calculated as the multiplication of the utilization and flow rates of reactants. [2]

Block Ohmic losses:

$$U_{ohm} = R_{ohm} * i \tag{17}$$

Block Activation losses:

$$U_{akt} = N * A * ln\left(\frac{i}{i0}\right) \tag{18}$$

Block Cell dynamics:

$$Cd = \frac{1}{\tau s+1} \tag{19}$$

Block Voltage calculations:

$$E_{OC} = K_C * E_N \tag{20}$$
$$E = E_{OC} - U_{akt} - U_{ohm} \tag{21}$$

Block Limits (for Nerst voltage):

$$Limit\_O2 = \frac{E_N - V_u * (Uf_{O2} - Uf_{O2(nom)})}{K_c * (Uf_{O2(peak)} - Uf_{O2(nom)})} \tag{22}$$

if: $Uf_{O2} > Uf_{O2(nom)}$

$$Limit\_Im\,a\,x = Limit\_O2 * cos\left(\frac{5*\pi}{2} * \left(\frac{i}{i_{end}} - 1\right)\right) \tag{23}$$

if: $i > i_{end}$

$$Limit\_Uf = Limit\_Im\,a\,x * cos\left(\frac{5*\pi}{2*(Uf_{O2H2}-1)}\right) \tag{24}$$

if: $Uf_{O2H2} > 1$

Block Stack efficiency:

$$\eta = \frac{E*i*100*z*F*Uf_{H2}}{N*i*\Delta G_0} \tag{25}$$

where: $i \neq 0$

Table 3. Fuel Cell Model Parameters.

Symbol	Description
A	Tafel slope
E	Voltage of cell [V]
E_0	Theoretical voltage of one cell [V]
E_N	Nerst voltage [V]
E_{OC}	Open circuit voltage [V]
F	Faraday constant = 96487 [C/mol]
h	Planck constant = $6.626*10^{-34}$ [J*s]
i	Current [A]

i_0	Current exchange [A]
I_{end}	End value of current [A]
I_{nom}	Nominal current [A]
k	Boltzman constant = $1.381*10^{-23}$ [J*K]
K	Undervolt constant [-]
K_C	Voltage constant at nominal conditions of operation [-]
N	Number of cells [-]
P_{air}	Pressure of air supply [atm]
P_{fuel}	Pressure of fuel supply [atm]
P_{H2}	Partial pressure of hydrogen [atm]
P_{H2O}	Partial pressure of water vapor [atm]
P_{O2}	Partial pressure of oxygen [atm]
P_{std}	Standard atmospheric pressure = 101325 [Pa] (norm ISO 10780)
R	Gas constant = 8.314 [$J*K^{-1}*mol^{-1}$]
R_{ohm}	Internal resistivity [Ω]
T	Operationa temperature of system [K]
T_{std}	Standard temperature = 273 [K] (norm ISO 10780)
U_{akt}	Activation losses [V]
Uf_{H2}	Hydrogen utilization [%/100]
Uf_{O2}	Oxygen utilization [%/100]
$Uf_{O2(nom)}$	Nominal oxygen utilization [%/100]
$Uf_{O2(peak)}$	Peak oxygen utilization [%/100]
U_{ohm}	Ohmic losses [V]
$V_{fuel(lpm)}$	Flow rate of fuel [lpm]
$V_{air(lpm)}$	Flow rate of air [lpm]
w	Percent of water vapor in air [%/100]
x	Fuel composition [%/100]
y	Oxidant composition [%/100]
z	The number of electrons transferred by the electrode reaction = 2 [-]
α	Charge transfer coefficient [-]
ΔG	Change of Gibbs free energy [J/mol]
ΔG_0	Change of Gibbs free energy at standard conditions [J/mol]
$\Delta h^0(H_2O(gas))$	Standard enthalpy of water = $241.83 * 10^3$ [J/mol]
η	Overall efficiency of the fuel cell [%]

Conclusion

To reduce the time and cost of experiments, fuel cell designers and engineers could get the benefits of using parametric models to predict the fuel cell performance by given operating conditions, geometries, and properties of materials.

The empirical – semi empirical models are designed for particular fields and should be changed for every specific operating condition or application. The analytical models are useful for simple designs with short time computing processes and the calculation of water management and voltage losses. The mechanistic models are suited for optimization and design applications. A hybrid model that consists of mechanistic and empirical approaches is that the usage of both approaches has advantages on improving the model and defeats the disadvantages of each of them and improves the performance of the PEM FCs. The other advantages of the combined model over the empirical and mechanistic model are that it can be utilized for any operating condition and application, and it easily does not need large range calculation.

One dimensional model is the first model that researchers established to study the fuel cell. It was a complicated model with a sandwich domain in the y-direction. The Sandwich model in the y-z or x-y directions is a two-dimensional model, which is the upgraded version of the one-dimensional model. Three-dimensional model, which is in the x-y-z direction is the most appropriate model to analyze PEM FC in every detail, for instance, current density distribution, the influence of flow field design on fuel cell performance, or bipolar plate. The multiphase flow; there are two and three-phase flow in the PEM FC.

This article provides a comprehensive overview of various modeling approaches for PEM fuel cells, with a specific focus on the semi-empirical models. Modeling approaches offers significant advantages, such as a reduction in the need for extensive experimental procedures, thereby minimizing development time and costs. The central part of this study is the analysis and description of a semi-empirical model of an air-cooled PEM FC from Simscape Electrical toolbox in the Matlab Simulink environment. The functionality of this model is detailed, illustrating how it simulates the performance of the fuel cell. For those components of the model that are not directly available within Simscape Electrical, additional Matlab script were used to provide a deeper insight into behavior and interactions in this model, including the process of reconstruction of the polarization curve and approximation of key constants.

This approach not only enhances our understanding of the internal processes involved in modeling the air-cooled PEM fuel cell but also provides valuable insights into the complex interactions between different components of the model. By integrating software-based simulations with analytical methods, it allows for a more precise adjustment of model parameters, enabling a deeper understanding of their impact on the overall performance of the fuel cell.

References

[1] N. M. Souleman, O. Tremblay and L. -A. Dessaint, A generic fuel cell model for the simulation of Fuel Cell Power Systems; 2009 IEEE Power & Energy Society General Meeting, Calgary, AB, Canada, 2009, pp. 1-8, doi: 10.1109/PES.2009.5275853.

[2] Putala, Rastislav. Specifics of modeling and experimental verification of an air-cooled PEM FC system. Bratislava: FEI STU, 2024. pp. 70. Diploma thesis.

[3] U.Lucia, 2014: Renewable Sustainable Energy, Rev.30, 164-169, 2014. https://doi.org/10.1016/j.rser.2013.09.025.

[4] Y.Chang, Y.Qin, Y.Yin, J.Zhang, X.Li, 2018: Appl. Energy, 230, 2018, 643-662. https://doi/org/10.1016/.apenergy.2018.08.125.

[5] S. A. Atyabi, E. Asfhari, 2019: J.Therm. Anal. Calorim, 135, 2019, 1823-1833. https://doi.org/10.10O7/s10973-018-7270-3.

[6] J.P. Kone, X. Zhang, Y. Yan, G. Hu, G. Ahmadi, 2017: Multiphase Flows. 9, 2017, 3-25. https://doi.org/10.1177/175/7482X176923.

[7] S.Hamdollahi, L.Jun, 2023: Chem. Eng. 29, 2023, 61-74. https://doi.org/ 10.2298/CICEQ220126014H.

[8] M. Ohenoja, A. Sorsa, K. Leiviska, 2018: Computers 7, 2018. https://doi.org/10.3390/computers 7040060.

[9] A.U.Thosar, H. Agarwai, S. Govarthan, A.K Lele, 2019: Chem. Eng. Sci.206.,2019, 96-117. https://doi.org/10.1016/ces.2019.05.022.

[10] B.Grondin, S.Roche, C.Lebreton, M.Benne, C.Damour, J.A.Kadjo, 2014: Engineering 6. 418 - 426. https://doc.org/10.4236/eng.2014.68044.

[11] M.Servi, I.Soltani, 2012: Sci. Eng. Technol. 3, 2012, 285-378. https://ijosel.com/docs/IJCSET12-03-08-036.

[12] S.Z.Chen, Z.O.Bao, Y.C.Wang, 2015: Appl. Mech. Mater. 740, 2015, 474-478. https://doi.org/10.4028/740.474.

[13] N.S.Hassan, W.R.Daud, K.Sopian, J.Sahari, 2009: J. Power Sources 193,, 2009, 249-257. https://doi.org/10.1016/j.jpowsour.2009.01.066 .

[14] C. Siegel, 2008: Energy 33, 2008, 1331-1352. https://doi.org/10.1016/energy.2008.04.015.

[15] C. Spiegel, PEM fuel cell modeling and simulation using Matlab. 2008. doi: 10.1016/b978-0-12-374259-9.x5001-0.

Authors

Ing. Rastislav Putala
Faculty of Mechanical Engineering
Slovak University of Technology
Bratislava, Slovakia
rastislav.putala@stuba.sk

Modelling and control in automotive industry research and applications. PhD student.

prof. Ing. Viktor Ferencey, PhD.
Hydrogen Fuel Cells Slovakia, j.s.c.
Bratislava, Slovakia
viktor.ferencey@gmail.com

Research in the field of hydrogen technologies

International Journal
Information Technology Applications (ITA)
Volume 13, Number 1, September 2024

ROBUST CONTROLLER METHODS DESIGN AND APPLICATIONS

Ján Cigánek, Štefan Kozák

Abstract:

The paper deals with robust discrete controller design using new effective approach based on the reflection vectors techniques. We assume that the controller structure consists of feed-forward and feedback parts. Developed robust algorithms were tested on the case study examples for different dynamical processes stable, unstable and oscillating type. Simulations were realized in MATLAB-Simulink. Obtained numerical and simulation results confirm applicability of the theoretical principles for robust control of processes subject to parametric model uncertainty.

Keywords:

Robust control, robust stability, parametrical uncertainty, quadratic programming, pole-placement.

Introduction

During last ten years, development of robust control elementary principles and evolution of new robust control methods for different model uncertainty types are visible. Progress in new techniques and theories in control of processes with model uncertainty is necessary because of performance requirements on control of complex processes containing large number of loops, activities coordination of a many agents in hybrid and stochastic control of systems containing large plant model uncertainties. Based on theoretical assumptions, modeling and simulation methods, an effective approach to the control of processes with strong and undefined uncertainties is designed. Such uncertainties are typical for biotechnology processes, chemical plants, automobile industry, aviation etc. For such processes is necessary to design robust and practical algorithms which ensures the high performance and robust stability using proposed mathematical techniques with respect the parametric and unmodelled uncertainties. Solution to such problems is possible using robust predictive methods and „soft-techniques" which include fuzzy sets, neuron networks and genetic algorithms.

Robust control is used to guarantee stability of plants with parameter changes. The robust controller design consists of two steps [1], [2]:

- analysis of parameter changes and their influence for closed-loop stability,
- robust control synthesis.

In hybrid control structures that combine the discrete controller and continuous plant, it is difficult to assess the closed-loop stability. One possibility is transformation of the controller and the continuous plant to the discrete-time region and specifying requirements for the discrete controller design. The problem of the robust controller design can be solved as:

- time-optimal robust controller design,
- design of the robust controller based on the pole-placement.

In both parts of the robust controller design it is possible to evolve from the solution of Diophantine equations.

1 Problem Formulation

Consider the robust control synthesis of a scalar discrete-time control loop. Transfer function of the original continuous-time system is described by the transfer function:

$$G_P(s) = \frac{\bar{B}(s)}{\bar{A}(s)} e^{-Ds} =$$

$$= \frac{\bar{b}_m s^m + \bar{b}_{m-1} s^{m-1} + \ldots + \bar{b}_0}{\bar{a}_n s^n + \bar{a}_{n-1} s^{n-1} + \ldots + \bar{a}_0} e^{-Ds} \tag{1}$$

Transfer function of (1) can be converted to its discrete-time counterpart

$$G_P\left(z^{-1}\right) = \frac{b_0 + b_1 z^{-1} + \ldots + b_n z^{-n}}{1 + a_1 z^{-1} + \ldots + a_n z^{-n}} z^{-d} \tag{2}$$

For the plant (2) a discrete-time controller is to be designed in form

$$G_R(z) = \frac{q_0 + q_1 z^{-1} + \ldots + q_\upsilon z^{-\upsilon}}{1 + p_1 z^{-1} + \ldots + p_\mu z^{-\mu}} = \frac{Q(z)}{P(z)} \tag{3}$$

The corresponding closed-loop characteristic equation is

$$1 + G_P(z^{-1})G_R(z^{-1}) = 0 \tag{4}$$

Substituting (3) and (2) in (4) after a simple manipulation yield the characteristic equation

$$\begin{aligned} &1 + G_P G_R = \\ &= (1 + p_1 z^{-1} + \ldots + p_\mu z^{-\mu})(1 + a_1 z^{-1} + \ldots + a_n z^{-n}) + \\ &+ (q_0 + q_1 z^{-1} + \ldots + q_\upsilon z^{-\upsilon})(b_1 z^{-1} + \ldots + b_n z^{-n}) z^{-d} = 0 \end{aligned} \tag{5}$$

Unknown coefficients of the discrete controller can be designed using various methods. In this paper robust controller design method based on reflection vectors is used.

The pole assignment problem is as follows: find a controller $G_R(z)$ such that *C(z)=e(z)* where *e(z)* is a given (target) polynomial of degree *k*. It is known [8] that when $\mu = n - 1$, the above problem has a solution for arbitrary *e(z)* whenever the plant has no common pole-zero pairs. In general for $\mu < n - 1$ exact attainment of a desired target polynomial *e(z)* is impossible.

Let us relax the requirement of attaining the target polynomial *e(z)* exactly and enlarge the target region to a polytope *V* in the polynomial space containing the point *e* representing the desired closed-loop characteristic polynomial. Without any restriction we can assume that $a_n = p_0 = 1$ and deal with monic polynomials *C(z)*, i.e. $\alpha_0 = 1$.

Let us introduce the stability measure as $\rho = c^T c$, where

$$c = S^{-1} C \tag{6}$$

and *S* is a matrix of dimensions $(n + \mu + 1) \times (n + \mu + 1)$ representing vertices of the target polytope *V*. For monic polynomials holds

$$\sum_{i=1}^{k+1} c_i = 1 \tag{7}$$

where $k = n + \mu$. If all coefficients are positive, i.e. $c_i > 0,\ i = 1,\ldots, k + 1$, then the point *C* is placed inside the polytope *V*.

The minimum ρ is attained if

$$c_1 = c_2 = \ldots = c_{k+1} = \frac{1}{k+1} \tag{8}$$

Then the point C is placed in centre of the polytope V.

In matrix form we have

$$C = G\,x \tag{9}$$

where G is the Sylvester matrix of the plant with dimensions $(n+\mu+d+1)\times(\mu+\upsilon+2)$ and x is the $(\mu+\upsilon+2)$-vector of controller parameters: $x = \left[p_\mu, \ldots, p_1, 1, q_\upsilon, \ldots, q_0\right]^T$.

Now we can formulate the following control design problem: find a discrete controller $G_R(z)$ such that the closed-loop characteristic polynomial $C(z)$ is placed:

a. In a stable target polytope V, $C(z) \in V$ (to guarantee stability)

b. As close as possible to a target polynomial $e(z)$, $e(z) \in V$ (to guarantee performance).

$$G = \begin{bmatrix}
0 & 0 & \ldots & 0 & 0 & b_{n+\mu-\upsilon} & b_{n+\mu-\upsilon+1} & \ldots & 0 & 0 \\
0 & 0 & \ldots & 0 & 0 & b_{n+\mu-\upsilon-1} & b_{n+\mu-\upsilon} & \ldots & 0 & 0 \\
. & . & . & . & . & . & . & . & . & . \\
0 & 0 & \ldots & 0 & 0 & b_{n+\mu-\upsilon-d+1} & b_{n+\mu-\upsilon-d+2} & \ldots & 0 & 0 \\
a_n & 0 & \ldots & 0 & 0 & b_{n+\mu-\upsilon-d} & b_{n+\mu-\upsilon-d+1} & \ldots & 0 & 0 \\
a_{n-1} & a_n & \ldots & 0 & 0 & b_{n+\mu-\upsilon-d-1} & b_{n+\mu-\upsilon-d} & \ldots & 0 & 0 \\
. & . & . & . & . & . & . & . & . & . \\
a_1 & a_2 & \ldots & a_\mu & a_{\mu+1} & b_{\mu-\upsilon-d+1} & b_{\mu-\upsilon-d+2} & \ldots & b_{\mu-d} & b_{\mu-d+1} \\
1 & a_1 & \ldots & a_{\mu-1} & a_\mu & b_{\mu-\upsilon-d} & b_{\mu-\upsilon-d+1} & \ldots & b_{\mu-d-1} & b_{\mu-d} \\
0 & 1 & \ldots & a_{\mu-2} & a_{\mu-1} & b_{\mu-\upsilon-d-1} & b_{\mu-\upsilon-d} & \ldots & b_{\mu-d-2} & b_{\mu-d-1} \\
. & . & . & . & . & . & . & . & . & . \\
0 & 0 & \ldots & a_{d+1} & a_{d+2} & 0 & 0 & \ldots & b_1 & b_2 \\
0 & 0 & \ldots & a_d & a_{d+1} & 0 & 0 & \ldots & 0 & b_1 \\
0 & 0 & \ldots & a_{d-1} & a_d & 0 & 0 & \ldots & 0 & 0 \\
. & . & . & . & . & . & . & . & . & . \\
0 & 0 & \ldots & 1 & a_1 & 0 & 0 & \ldots & 0 & 0 \\
0 & 0 & \ldots & 0 & 1 & 0 & 0 & \ldots & 0 & 0
\end{bmatrix}$$

Let the polytope V denote the $(k+1)\times N$ matrix composed of coefficient vectors v_j, $j = 1, \ldots, N$ corresponding to vertices of the polytope V.

Then we can formulate the above controller design problem as an optimization task: Find x that minimizes the cost function

$$J_1 = \min_x x^T G^T G x - 2e^T G x = \min_x \left\| Gx - e \right\|^2 \tag{10}$$

subject to the linear constraints

$$G\,x = V\,w(x), \tag{11}$$

$$w_j(x) > 0, \quad j = 1, \ldots, N, \tag{12}$$

$$\sum_j w_j(x) = 1. \tag{13}$$

Here $w(x)$ is the vector of weights of the polytope V vertices to obtain the point $C = G\,x$. Fulfillment of the latter two constraints (12), (13) guarantees that the point C is indeed located inside the polytope V.

Then, finding the robust pole-placement controller coefficients represents an optimization problem that can be solved using the Matlab Toolbox OPTIM (quadprog) with constraints [9].

Generally J_1 is a kind of distance to the centre of the target polytope V. Is it better to use another criterion J_2, which measures the distance to the Schur polynomial $E(z)$

$$J_2 = (C-E)^T (C-E) = (Gx-E)^T (Gx-E). \qquad (14)$$

It is possible to use the weighted combination of J_1 and J_2

$$J = (1-\alpha)J_1 + \alpha J_2, \quad 0 \le \alpha \le 1 \qquad (15)$$

and to solve the following quadratic programming task

$$J = \min_x \left\{ x^T G^T \left[(1-\alpha)(SS^T)^{-1} + \alpha I_{k+1} \right] Gx - 2\alpha E^T Gx \right\},$$

$$S^{-1}Gx < 0. \qquad (16)$$

Assume the discrete robust controller design task with parametrical uncertainties in system description. Let us also assume that coefficients of the discrete-time system transfer functions $a_n, \ldots,$ a_1 and $b_n, \ldots, b_1$ are placed in polytope W with the vertices $d^j = \left[a_n^j, \ldots a_1^j, b_n^j, \ldots, b_1^j \right]$:

$$W = conv\{ d_j, j = 1, \ldots, M \} \qquad (17)$$

As (9) is linear in system parameters, it is possible to claim that for arbitrary vector of the controller coefficients x is the vector of the characteristic polynomial coefficients $C(z)$ placed in the polytope A with vertices $a^1, \ldots, a^M$:

$$A = conv\{ a^j, j = 1, ..., M \} \qquad (18)$$

where $a^j = D^j x$ and D^j is the Sylvester matrix of dimensions $(n + \mu + d + 1) \times (\mu + \upsilon + 2)$, composed of vertices set d^j, as in case of the exact model (9).

1.1 Problem Formulation

The digital controller $x = \left[p_\mu, \ldots, p_1, 1, q_\upsilon, \ldots, q_0 \right]^T$ is to be designed such that all its vertices a^j, $j = 1, \ldots, M$ are placed inside a stable desired target polytope V.

This problem can be effectively solved using quadratic programming procedure. It is necessary to find the controller x by minimization of the cost function

$$J = \min_x \left\{ x^T \bar{D}^T \begin{bmatrix} (1-\alpha)(I_M \otimes (S^T)^{-1})(I_M \otimes S^{-1}) + \\ + \alpha I_{(k+1)M} \end{bmatrix} \bar{D} x - 2\alpha E^T \bar{D} x \right\}, \quad S^{-1} D^j x > 0, \ j = 1, \ldots, M \qquad (19)$$

where I_M is identity matrix of dimension M, $\otimes$ is the Kronecker product and $\bar{D}^T = \left[D_1^T, \ldots, D_M^T \right]$

1.2 Stable Region Computation via Reflection Coefficients

Polynomials are usually specified by their coefficients or roots. They can be characterized also by their reflection coefficients using Schur-Cohn recursion.

Let $C_k(z^{-1})$ be a monic polynomial of degree k with real coefficients $c_i \in R$, $i = 0, \ldots, k$,

$$C(z^{-1}) = 1 + c_1 z^{-1} + \ldots + c_k z^{-k}. \qquad (20)$$

Reciprocal polynomial $C_k^*(z^{-1})$ of the polynomial $C_k(z^{-1})$ is defined in [11] as follows

$$C_k^*(z^{-1}) = c_k + c_{k-1}z^{-1} + \dots + c_1 z^{-k+1} + z^{-k} \tag{21}$$

Reflection coefficients r_i, $i = 1, \dots, k$, can be obtained from the polynomial $C_k(z^{-1})$ using backward Levinson recursion [12]

$$z^{-1}C_{i-1}(z^{-1}) = \frac{1}{1-\left|r_i^2\right|}\left[C_i(z^{-1}) - r_i C_i^*(z^{-1})\right] \tag{22}$$

where $r_i = -c_i$ and c_i is the last coefficient of $C_i(z^{-1})$ of degree i. From (22) we obtain in a straightforward way:

$$C_i(z^{-1}) = z^{-1}C_{i-1}(z^{-1}) + r_i C_{i-1}^*(z^{-1}). \tag{23}$$

Expressions for polynomial coefficients $C_{i-1}(z^{-1})$ and $C_i(z^{-1})$ result from equations (22) and (23):

$$C_{i-1}(z^{-1}) = \frac{1}{1-\left|r_i^2\right|}\left[\sum_{j=0}^{i-1}\left(c_{i,j+1} - r_i c_{i,i-j-1}\right)z^{-j}\right] \tag{24}$$

$$C_i(z^{-1}) = \sum_{j=0}^{i}\left(c_{i-1,j-1} + r_i c_{i-1,i-j-1}\right)z^{-j}. \tag{25}$$

The reflection coefficients r_i are also known as Schur-Szegö parameters [11], partial correlation coefficients [6] or k-parameters [13]. Presented forms and structures were effectively used in many applications of signal processing [13] and system identification [6]. A complete characterization and classification of polynomials using their reflection coefficients instead of roots (zeros) of polynomials is given in [11].

The main advantage of using reflection coefficients is that the transformation from reflection to polynomial coefficients is very simple. Indeed, according to (23) and (25), polynomial coefficients c_i depend multilinearly on the reflection coefficients r_i. If the coefficients $c_i \in R$ are real, then also the reflection coefficients $r_i \in R$ are real.

Transformation from reflection coefficients r_i, $i = 1, \dots, k$, to polynomial coefficients c_i, $i = 1, \dots, k$, is as follows

$$c_i = c_i^{(k)}, \quad c_i^{(i)} = -r_i,$$
$$c_j^{(i)} = c_j^{(i-1)} - r_i c_{i-j}^{(i-1)}, \qquad i = 1, \dots, k; \quad j = 1, \dots, i-1 \tag{26}$$

or in the matrix form

$$C = R(r)c^{(t)}, \quad t = 1, \dots, k-1,$$
$$C^{(t)} = \begin{bmatrix} 0^T \\ R_t(r_t) \end{bmatrix} C^{(t-1)}, \tag{27}$$

where

$$C = [c_k, \dots, c_1, 1]^T,$$
$$C^{(t)} = \left[0, c_t^{(t)}, \dots, c_1^{(t)}, 1\right]^T,$$
$$C^{(0)} = [0, 1]^T,$$

$$R(r) = R_k(r_k)\begin{bmatrix} 0^T \\ R_{k-1}(r_{k-1}) \end{bmatrix} \cdots \begin{bmatrix} 0^T \\ R_t(r_t) \end{bmatrix},$$

$$R_j(r_j) = I_{j+1} - r_j E_{j+1},$$

where I_k is a *k* x *k* identity matrix, $E_k = \begin{bmatrix} 0 \dots 1 \\ . \;\; . \;\; . \\ 1 \dots 0 \end{bmatrix}$ and 0^T is a row vector of zeros.

Lemma 1. A linear discrete-time dynamic system is stable if its characteristic polynomial is Schur stable, i.e., if all its poles lie inside the unit circle.

The stability criterion in terms of reflection coefficients is as follows [11].

Lemma 2. A polynomial $C(z^{-1})$ has all its roots inside the unit disk if and only if $|r_i| < 1,$ $i = 1, ..., k.$

A polynomial $C(z^{-1})$ lies on the stability boundary if some $r_i = \pm 1,$ $i = 1, ..., k$. For monic Schur polynomials there is a one-to-one correspondence between $C = [c_k, \dots, c_1]^T$ and $r = [r_1, \dots, r_k]^T$.

Stability region in the reflection coefficient space is simply the *k*-dimensional unit hypercube $R = \{r_i \in (-1,1), i = 1, \dots, k\}$. The stability region in the polynomial coefficient space can be found starting from the hypercube *R*.

1.3 Stable Polytope of Reflection Vectors

It will be shown that for a family of polynomials the linear cover of the so-called reflection vectors is Schur stable.

Definition 1. The reflection vectors of a Schur stable monic polynomial $C(z^{-1})$ are defined as the points on stability boundary in polynomial coefficient space generated by changing a single reflection coefficient r_i of the polynomial $C(z^{-1})$.

Let us denote the positive reflection vectors of $C(z^{-1})$ as $v_i^+(C) = (C|r_i = 1), i = 1, \dots, k,$ and the negative reflection vectors of $C(z^{-1})$ as $v_i^-(C) = (C|r_i = -1), i = 1, \dots, k.$

The following assertions hold:

1. every Schur polynomial has *2k* reflection vectors $v_i^+(C)$ and $v_i^-(C), i = 1, \dots, k;$

2. all reflection vectors lie on the stability boundary $(r_i^v = \pm 1);$

3. the line segments between reflection vectors $v_i^+(C)$ and $v_i^-(C)$ are Schur stable.

In the following theorem a family of stable polynomials is defined such that the polytope generated by reflection vectors of these polynomials is stable.

*Theorem 1***.** Consider $r_1^C \in (-1,1),$ $r_k^C \in (-1,1)$ and $r_2^C = \dots = r_{k-1}^C = 0$. Then the inner points of the polytope *V(C)* generated by the reflection vectors of the point *C*

$$V(C) = conv\{v_i^{\pm}(C), i = 1, \dots, k\} \tag{28}$$

are Schur stable.

1.4 Roots of Reflection Vectors

In this section we study the root placement of reflection vectors. It is useful for selecting a stable target simplex to solve the robust output control problem.

By definition, at least one root of a reflection vector $v_i(C)$ (i.e. root of $V_i(z^{-1}) = \left[z^{-k} \ldots z^{-1}\ 1\right]\begin{bmatrix} v_i(C) \\ 1 \end{bmatrix}$) must lie on the unit circle, and the number of unit circle roots is determined by the number *i* of the reflection vector $v_i(C)$ and the character of the roots (real or complex) is determined from the sign of the boundary reflection coefficient $(r_i^V = \pm 1)$.

1.5 Robust Controller Design

A robust controller is to be designed such that the closed-loop characteristic polynomial is placed in the stable polytope (linear cover) of reflection vectors. It means that the following problems have to be solved:

1. choice of initial polynomial C(z-1) for generating the polytope V(C),
2. choice of k + 1 most suitable vertices of V(C) to build a target simplex S,
3. choice of a target polynomial E(z-1).

In the following section some "thumb rules" are given for choosing a stable target simplex S.

To choose *k + 1* vertices of the target simplex *S* we use the well known fact that poles with positive real parts are preferred to those with negative ones [1]. The positive reflection vectors $v_i^+(C)$ with *i* odd and negative reflection vectors $v_i^-(C)$ with *i* even are chosen yielding *k* vertices. The *(k+1)*th vertex of the target simplex *S* is chosen as the mean of the remaining reflection vectors.

The target polynomial $E(z^{-1})$ of order *k* is reasonable to be chosen inside the stable polytope of reflection vectors *V(C)*. A common choice is $E(z^{-1})=C(z^{-1})$.

For higher-order polynomials the size of the target simplex S is considerably less than the volume of the polytope of reflection vectors V. That is why the above quadratic programming method with a preselected target simplex S works only if uncertainties are sufficiently small. Otherwise it is reasonable to use some search procedure to find a robust controller such that the polytope of closed-loop characteristic polynomial is placed inside the stable polytope of reflection vectors V(C).

2 Illustrative Examples

Consider a system described by the second order transfer function

$$G_P(s) = \frac{B(s)}{A(s)} e^{-Ds} = \frac{b_0}{s^2 + a_1 s + a_0} e^{-Ds} \tag{29}$$

with individual coefficients varying within uncertainty intervals

$$b_0 \in \langle 3;4 \rangle, a_1 \in \langle -0.8;-0.4 \rangle, a_0 \in \langle 0.6;1 \rangle, D = 0.6 \tag{30}$$

After some modification of (29) we obtain

$$G_P(s) = \frac{1}{\frac{1}{b_0}s^2 + \frac{a_1}{b_0}s + \frac{a_0}{b_0}} e^{-Ds} = \frac{1}{a_2' s^2 + a_1' s + a_0'} e^{-Ds} \tag{31}$$

with individual coefficients varying within uncertainty intervals

$$a_2' \in \left\langle \frac{1}{4};\frac{1}{3}\right\rangle, a_1' \in \left\langle -\frac{4}{15};-\frac{1}{10}\right\rangle, a_0' \in \left\langle \frac{3}{20};\frac{1}{3}\right\rangle, D=0.6 \quad (32)$$

To assess stability, four continuous-time Charitonov transfer functions are consideredThey have been converted to the discrete region with sampling period *T=0.6s*:

$$G_{+-}\left(z^{-1}\right)=\frac{0.5661z^{-2}+0.6013z^{-3}}{1-2.022z^{-1}+1.197z^{-2}} \quad G_{++}\left(z^{-1}\right)=\frac{0.629z^{-2}+0.7386z^{-3}}{1-2.411z^{-1}+1.616z^{-2}}$$

$$G_{-+}\left(z^{-1}\right)=\frac{0.8648z^{-2}+1.073z^{-3}}{1-2.25z^{-1}+1.896z^{-2}} \quad G_{--}\left(z^{-1}\right)=\frac{0.75z^{-2}+0.8135z^{-3}}{1-1.75z^{-1}+1.271z^{-2}} \quad (33)$$

2.1 Robust pole-placement algorithm

Consider the nominal continuous-time transfer function (29) with $b_0=3.5, a_1=0.6, a_0=0.8$ converted to the discrete-time with the sampling period *T=0.6s*:

$$G_{nom}\left(z^{-1}\right)=\frac{0.6956z^{-2}+0.7851z^{-3}}{1-2.095z^{-1}+1.433z^{-2}} \quad (34)$$

One possible way how to design a stable controller is to design it for the nominal plant model and apply it in all other plant models.

Based on the solution of the Diophantine equation the following controller was designed for the continuous-time transfer function $G_{nom}(s)$:

$$G_{FB}\left(z^{-1}\right)=\frac{Q(z^{-1})}{P(z^{-1})}=\frac{2.1-2.7z^{-1}+0.012z^{-2}}{1+2.3z^{-1}+1.5z^{-2}} \quad (35)$$

with the corresponding control law:

$$u_2(k)=-2.3u_2(k-1)-1.5u_2(k-2)+$$
$$+2.1y(k)-2.7y(k-1)+0.012y(k-2) \quad (36)$$

The target closed-loop characteristic polynomial according to previous consideration is:

$$C(z)=(1+0.37z^{-1})(1+0.35z^{-1})(1+0.35z^{-1})$$
$$(1-0.45z^{-1})(1-0.45z^{-1}) \quad (37)$$

Closed-loop step responses under the discrete-time feedback controller (35) and feed-forward controllers $G_{FF}\left(z^{-1}\right)=S(z^{-1})/P(z^{-1})=\frac{1}{36.4}$ in case of $G_{+-}(s)$ and $G_{++}(s)$ transfer functions, respectively, and $G_{FF}\left(z^{-1}\right)=\frac{1}{4.75}$ in case of $G_{-+}(s)$ and $G_{--}(s)$ transfer functions are illustrated in (Fig.1).

2.2 Controller Design via Reflection Coefficients

Consider the nominal continuous-time transfer function (29) with $b_0=3.5, a_1=0.6, a_0=0.8$ converted to the discrete-time with the sampling period *T=0.6s* (34). The task is to design a discrete-time controller (3), $\upsilon=\mu=2$.

From the transfer function (34) and matrix form of (9) results:

$$C=\begin{bmatrix} 0 & 0 & 0 & 0.7851 & 0 & 0 \\ 1.433 & 0 & 0 & 0.6956 & 0.7851 & 0 \\ -2.095 & 1.433 & 0 & 0 & 0.6956 & 0.7851 \\ 1 & -2.095 & 1.433 & 0 & 0 & 0.6956 \\ 0 & 1 & -2.095 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}\begin{bmatrix} p_2 \\ p_1 \\ 1 \\ q_2 \\ q_1 \\ q_0 \end{bmatrix}$$

Let us choose the initial polynomial $C(z^{-1})$ for generating the polytope $V(C)$ as follows

$$C(z)=[1-(0.3\pm 0.2i)z^{-1}][1-0.2z^{-1}][1-0.3z^{-1}][1-0.4z^{-1}] \tag{38}$$

with reflection coefficients $r_1=1.5,\ r_2=-0.93,\ r_3=0.297,\ r_4=-0.0482,\ r_4=0.00312.$

Now we can find the reflection vectors $v_i\,(C)$ of the initial polynomial $C(z^{-1})$ leading to the matrix form of the target simplex S (vertex polynomial coefficients)

$$S=\begin{bmatrix} 0 & 0 & 0 & -0.3 & 0.5 & 0.6 \\ 0 & 0 & -0.3 & 0.5 & 0 & 0 \\ 0 & -0.3 & 0.5 & 0 & 0 & 0 \\ -0.3 & 0.5 & 0 & 0 & 0 & 0 \\ 0.5 & 1 & 0 & 0 & -0.3 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

The discrete-time controller design task for the nominal transfer function (34) has been solved via quadratic programming taking $\alpha=0.1$ in the cost function J (16).

For the selected target simplex S we have obtained the following discrete-time feedback controller

$$G_{FB}\left(z^{-1}\right)=\frac{Q(z^{-1})}{P(z^{-1})}=\frac{1.926-2.737z^{-1}+0.3513z^{-2}}{1+2.032z^{-1}+1.223z^{-2}} \tag{39}$$

with the control law

$$\begin{aligned} u_2(k) &= -2.032u_2(k-1)-1.223u_2(k-2)+ \\ &+1.926y(k)-2.727y(k-1)+0.3513y(k-2) \end{aligned} \tag{40}$$

Corresponding closed-loop step responses under the feedback controller (39) and feed-forward controller $G_{FF}\left(z^{-1}\right)=S(z^{-1})/P(z^{-1})=\dfrac{1}{23.9}$ in case of $G_{+-}(s)$ and $G_{++}(s)$ transfer functions, respectively, and $G_{FF}\left(z^{-1}\right)=\dfrac{1}{4.44}$ in case of $G_{-+}(s)$ and $G_{--}(s)$ transfer functions are in (Fig.2).

Conclusion

The paper deals with the development of robust methods based on reflection vectors methodology for computation of control law coefficients guaranteeing stability, robustness and high performance with respect to parameter uncertainties. Theoretical results were verified on the examples. Proposed methods were tested for both stable and unstable processes.

The paper proposes theoretical principles and design methodology of robust discrete-time controllers for systems with parametric uncertainties.

Two illustrative examples were solved using quadratic programming for suitably defined cost function. Simulation results prove applicability of the proposed robust controller design theory for systems with parametric uncertainty.

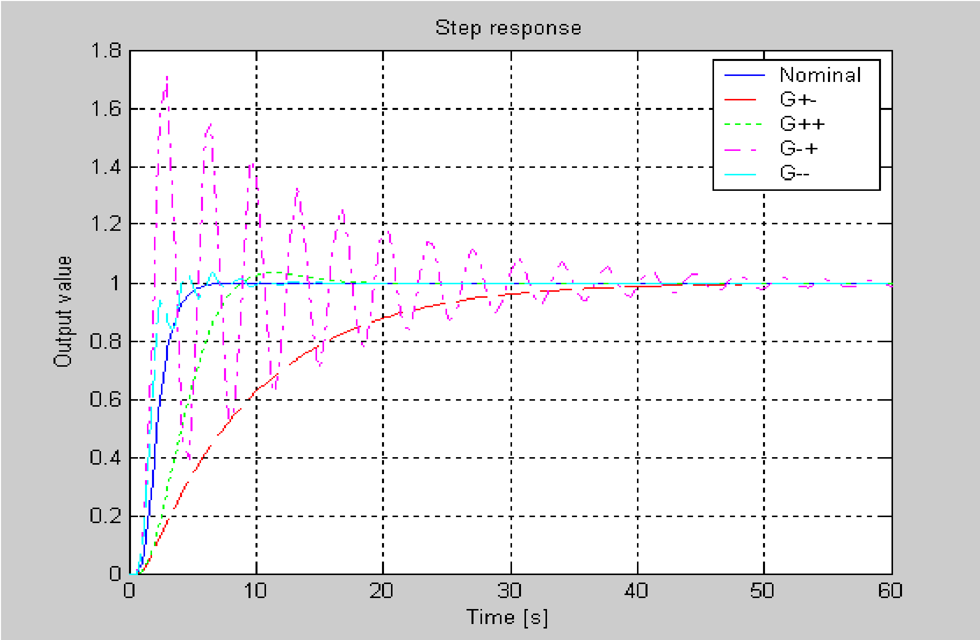


Fig.1. Closed-loop step responses under robust controller.

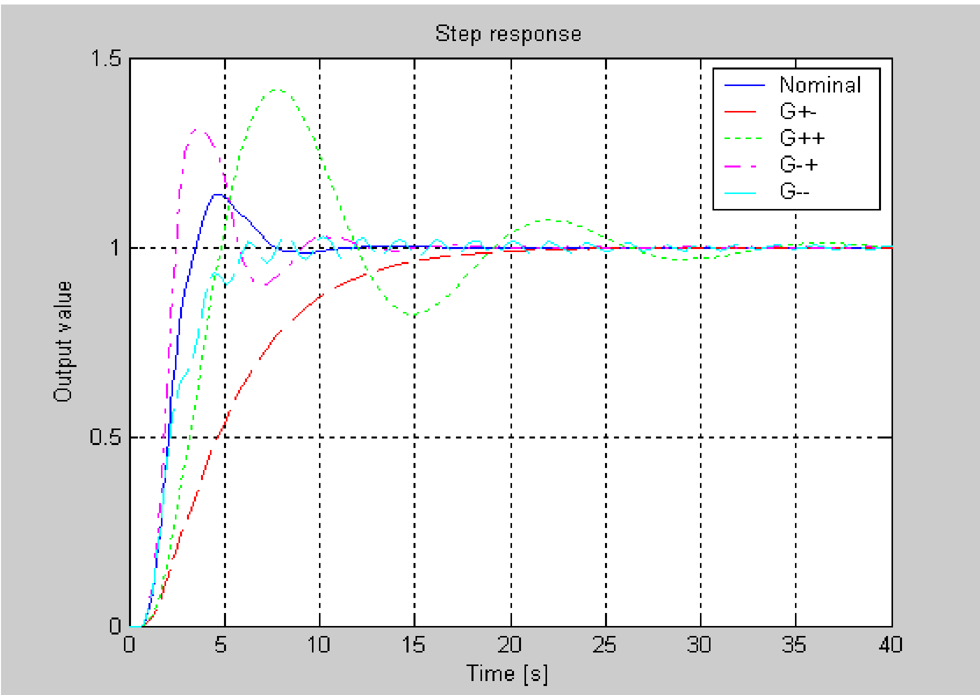


Fig.2. Closed-loop step responses under robust controller.

Acknowledgement

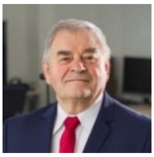
This paper was supported by Slovak Scientific Grant Agency -VEGA No 1/0822/08/4.

References

[1] Š. Kozák, "Development of control engineering methods and their applications in industry". In 5th Int. Scientific-Technical Conference Process Control 2002. Kouty nad Desnou, Czech Republic, 2002.
[2] M. Kocúr, "HW realizácia PID algoritmov na báze FPGA štruktúr". Slovak University of Technology in Bratislava, Bratislava.
[3] V. Viswanathan, "Embedded Control Using FPGA". Indian Institute of Technology, Bombay, 2005
[4] L. H. Keel, S. P. Bhattachayya, "A linear programming approach to controller design". Automatica, vol. 35, pp. 1717 – 1724, 1999.
[5] J. L. Diaz-Barrero, J. J. Egozcue, "Characterization of polynomials using reflection coefficients". Appl. Math. E-Notes, vol. 4, 114 – 121, 2004.
[6] B. Picinbono, M. Bendir, "Some properties of lattice autoregressive filters". IEEE Trans. Acoust. Speech Signal Process, 34, 342 – 349, 1986.
[7] A. M. Oppenheim, R. W. Schaffer, "Discrete-Time Signal Processing". Prentice-Hall, Englewood Cliffs, 1989.
[8] S. M. Kay, "Modern Spectral Estimation". Prentice Hall, New Jersey, 1988.
[9] J. Ackermann, "Robust Control" Systems with Uncertain Physical Parameters. Springer-Verlag. London, UK, 1993.
[10] D. Bishop, "Fixed point package user's guide". http://www.vhdl.org
[11] Xilinx, "SP601 Hardware User Guide".
[12] J. Cigánek, Š. Kozák, "Robust controller design techniques for unstable systems". In Int. conf. Cybernetics and informatics, Vyšná Boca, Slovak Rep. 2010.
[13] Ü. Nurges, "Robust pole assignment via reflection coefficients of polynomials". Automatica, 42(7), 1223 – 1230, 2006.
[14] Ü. Nurges, "New stability conditions via reflection coefficients of polynomials". IEEE Transactions on Automatic Control, 50(9), 1354 – 1360, 2005.

Authors

Ing. Ján Cigánek, PhD.
Faculty of Electrical Engineering and Information Technology,
Slovak University of Technology in Bratislava, Slovakia
jan.ciganek@stuba.sk
graduated from the Faculty of Electrical Engineering and Information Technology, Slovak University of Technology in Bratislava and obtained his MSc in Cybernetics. Currently, he is a Assistant Professor at the Institute of Automotive Mechatronics FEI STU In Bratislava. His research interests include robust control design, computational tools and Scada systems.



prof. Ing. Štefan Kozák, PhD.
Faculty of Informatics,
Pan-European University in Bratislava, Slovakia
stefan.kozak@paneurouni.com
obtained MSc. in Electrical Engineering from the Slovak Technical University in Bratislava in 1970 and the Ph.D. in Technical Cybernetics from the Slovak Academy of Sciences in 1978. Currently he is with the Institute of Applied Informatics of Pan-European University in Bratislava. His research interests include system theory, linear and nonlinear control methods, numerical methods and software for modeling, control, signal processing and embedded intelligent systems.

International Journal
Information Technology Applications (ITA)
Volume 13, Number 1, September 2024

USING MOTION CAPTURE SYSTEMS TO CREATE VIRTUAL REALITY SCENES

Csongor Mészáros

Abstract:

Motion capture systems have become essential in the evolving field of virtual reality experience design, providing users with an interactive journey. This paper delves into the world of motion capture, focusing on the Rokoko motion capture system and tools. Known for its precision and flexibility, Rokoko technology allows creators to meticulously capture real-world motion and seamlessly bring virtual worlds to life. When integrated with Unreal Engine, Rokoko's motion capture technology provides developers with a convenient approach to incorporating human motion into virtual environments. This collaboration goes beyond aspects to include procedural elements that unlock the full capabilities of the system. By using Rokoko's accessories, creators can enhance the capture of body gestures and subtle facial expressions. By mastering this technology, creators can create emotionally charged scenes that enhance the quality of their digital worlds. Our goal is to demonstrate methods for creating VR environments using Rokoko Motion Capture in a variety of scenarios.

Keywords:

Rokoko, Motion capture, Unreal Engine, virtual reality.

Introduction

In the field of virtual reality, where the boundaries between the digital and physical worlds are blurring seamlessly, the combination of motion capture technology and artistic ingenuity has enormous potential to create experiences that push the boundaries of conventional reality.

This paper explores these exciting areas of motion capture technology. We look at the Rokoko motion capture software and its accessories and delve into Rokoko's live stream into the Unreal Engine environment, where it incorporates human movement into realistic VR scenarios.

We take a deep dive into the unique features and details of the system. Our goal is to unravel the mechanisms that allow the system to capture the essence of human movement. We then import this into the virtual scenery via live streaming or pre-recorded captures.

In our work with the Rokoko Motion Capture System we will construct different VR projects. Each one uses motion capture for different purposes. This way we can capture the essence of Rokoko's use in many scenarios.

1 Motion Captures Past

Motion capture is a technology that converts human movement into digital data to bring characters to life with greater realism. The process involves tracking and analysing movement using a variety of techniques, such as reflective markers or sensors that record the position and orientation of specific points on the body.

The evolution of animation has moved from hand-drawn 2D animation to digital 3D animation. The first animations were created using "stop motion", where objects or characters were moved manually, and each movement was captured on a separate frame. [1]

Later came the rotoscope, in which live-action shots were changed frame by frame to mimic the movement of the actors. The technique was laborious but resulted in smooth and realistic animation. [2]

In the 1980s, optical motion capture emerged, using a network of cameras to track the reflections of markers placed on the body. This technique made it possible to create a 3D digital movement of a person based on the recorded movements of the marks. [3]

2 Types of Motion Capture Systems

This section goes through the main types of motion capture systems. Starting with the big trio of OMC, IMC, MMC, we then move on to hybrid motion capture systems and finally some of the emerging motion capture systems.

2.1 Optical Motion Capture

Optical motion capture systems use cameras to track reflective markings on the wearer's body. The markings reflect infrared light, and the cameras use this reflected light to determine the position of the markings in three-dimensional space. Optical motion capture systems are the most accurate type of motion capture and can be used to capture a wide range of movements, including fine motor movements. [4]

Active OMC is when the markers emit light, which increases accuracy but also increases cost and complexity.

Passive OMC means the markers reflect light from external sources, they are cheaper but less accurate.

2.2 Inertial Motion Capture

Inertial motion capture is a technology that uses inertial sensors to track the movement of objects, people or animals. It is a relatively new technology that has already found widespread application in a variety of fields, including sports, entertainment, healthcare and robotics.

IMUs are essential components of IMC systems. They are small, lightweight devices that contain three main sensors: gyroscopes, accelerometers and sometimes magnetometers. Gyroscopes measure angular velocity, accelerometers measure linear acceleration and magnetometers measure magnetic field strength. [5]

By combining data from these sensors, IMUs can track the position, orientation and velocity of an object in 3D space. The collected data can then be used to plot the motion of the object over time. [5]

2.3 IMC vs OMC

There are many comparisons in the IMC vs. IMC or OMC vs. OMC category, but it's hard to find a test that focuses on their strengths rather than their weaknesses. One such test was a comparison of patient recovery from stroke. In terms of results, IMC is a more appropriate motion capture tool for monitoring recovering patients as it provides more consistent data on a scale of 0.08 m/s, which is clinically significant in some cases. [6].

2.4 Markerless Motion Capture

All previous mocap systems have relied on markers to capture motion. However, MMC is an alternative that uses various techniques based on computer vision and machine learning algorithms to capture motion without these markers. [7]

It works through motion sensing. An actor or object performs a movement in a pre-prepared space. Several cameras or depth sensors are strategically placed in this space. The cameras record data. This data is fed into algorithms designed to capture the movement of the person/object. The algorithms analyze the data and based on the identification of key body parts, estimate both 3D position and pose orientation in each frame. Finally, the data is formatted to be compatible with the chosen software, where it can be animated and further edited. [7]

2.5 Hybrid Motion Capture

Hybrid motion capture systems combine two major technologies to overcome the limitations of each and provide a solution for capturing complex and realistic motion. The combination of these technologies offers several benefits. IMUs can compensate for occlusion and provide data in low light environments, increasing overall accuracy. Fewer markers are required compared to purely optical systems. This makes the process less intrusive and reduces the need for post-processing clean-up. Hybrid systems can be used for motion capture in environments unsuitable for purely optical systems, such as outdoor filming or live stage performances. With fewer cameras and markers, setup time can be significantly reduced, making the process more efficient.

2.6 Exploring Emerging Motion Capture Technologies

The big three - IMC, OMC and MMC - seem to dominate the current market. However, it is always beneficial to look for new or under-utilized technologies within Mocap.

Electromyography-Capture: The electrical activity of the muscles is measured using electrodes on the surface of the skin. This activity is then applied to the actor's predicted movement. The use of EMG is specific and therefore only applicable in certain areas: rehabilitation, prosthetics and sports science. [8]

Acoustic motion capture: Motion is captured using arrays of microphones placed around the recording area itself. The system analyses the sounds made by the actor as they move: footsteps, rustling of clothing, and these sounds are analyzed and transformed by a machine learning algorithm into the predicted movement of the subject. [9]

3 Software and Kit

Motion capture exists in a symbiotic world of software and hardware. It is where our digital creations come to life. The brain is software that processes and interprets the data collected by special rigs with the appropriate hardware. These rigs, sometimes using cameras, sensors or suits, convert the physical movements of people or things into digital images. The use of both software and hardware can work well together to create animations that are convincing and realistic. These range from blockbuster movies to innovative virtual reality experiences.

3.1 Inertial Suits

Rokoko is a leading provider of motion capture systems and software that enable filmmakers to capture and animate human motion easily and accurately.

Their innovative solutions have revolutionized the way motion capture is used in industries ranging from film and television production to game development and virtual reality applications. The advantage of this technology is that it can be used for shorter shots in game development, but the disadvantage is that it is not suitable for long streaming or so-called "V-tubes" [10].

Perception Neuron Smartsuit Pro is another full body motion capture suit designed for a wide range of users. It uses a combination of inertial sensors and magnetic tracking devices to capture accurate motion data. The suit is highly versatile and offers support for a variety of sensing environments, including indoor and outdoor environments. [11]

Xsens MVN Awinda is a full body motion capture system that uses inertial sensors and magnetic tracking devices to capture high fidelity data. It is known for its exceptional accuracy and reliability, making it a popular choice for professional applications. Its compact and lightweight design makes it portable and suitable for a variety of environments. [12]

Table 1. Inertial suit comparison.

Feature	Rokoko Smartsuit Pro	Perception Neuron Smartsuit Pro	Xsens MVN Awinda
Tracking technology	Inertial sensors	Inertial sensors + magnetic trackers	Inertial sensors + magnetic trackers
Full-body tracking	Yes	Yes	Yes
Hand and finger tracking	Yes	Yes	Yes
Data accuracy	Good	Excellent	Excellent
Portability	Yes	Yes	No
Price estimate	$2,999	$3,999	$5,999

4 Setting Up

After evaluating the resources and manpower are required for this project. We chose the Rokoko Smartsuit Pro, which is a comprehensive MC system with several key components. It requires calibration and careful setup for our work. In this section we venture into each of these instruments one by one.

The Smartsuit Pro is embedded with 9 DOF sensors for accurate 3D motion tracking. The suit is compatible with smart gloves for capturing hand and finger movements. The smart gloves improve motion capture accuracy by tracking complex hand and finger gestures. The suit typically requires a minimum of three amps through a single port to operate efficiently, so we chose a pair of AlzaPower Onys 20000mAh.

We set up a dedicated router, an AX3000 4-stream Wi-Fi 6 router. It's not necessary, but it's highly recommended because it makes it easier to connect as it's, always in the same room as the equipment. It can be moved around with the whole set, so the overall set-up time is significantly reduced if we want to use the Mocap in other areas.

We installed the Rokoko software and linked it to our accounts. We then plugged in the Smartsuit Pro and updated it, the same with the gloves one at a time, but this connection was only via the plugin cable, so we needed to set up the Wi-Fi connection. The Rokoko software's UI helps us with this, we needed to log in to our previously set up router, set the receiver IP from CMD-Ipconfig, select a receiver port, we chose 14041, and with WPA2 security selected, we are ready to make our set remote from the PC.

Now our setup is missing one key feature, Face Capture. Fortunately, the Rokoko Face Capture application is available for iPhone iOS 13.0 or later versions, so it is easy to download. We connect the selected phone to our router and activate the capture. We accept when Rokoko asks for permission to connect to our phone, and now our mocap setup is complete.

To achieve our goal and create scenes, we needed other software to do this. We chose Unreal Engine because it is compatible with Rokoko Live Stream and MetaHumans, which we used to create realistic digital humans to use in our scenes.

5 Preparing Unreal Engine

The project has all assets, but they need customisation and synchronisation to work in Unreal Engine. We will go through the steps of integrating the Metahuman characters with motion capture data into the Unreal Engine scene, as this will serve as a starting point for all our future scenarios.

First, we created a blank Unreal Engine (UE) project into which we imported the third person and VR content packs. Opened the Epic Game Market and installed the "Rokoko Studio Live" and "MetaHuman" plugins into our project, then downloaded the "Rokoko resources" file from their website and imported it into our project. [10]

In order to import pre-created or our own metahuman we use Quixel Bridge, which is integrated in UE, and with our plugin already installed it can easily download and transfer the metahuman into our project. Before adding our metahuman to the scene, we made sure that the Force LOD value was 0, which made our metahuman always render with maximum detail.

Now we can export our pre-recorded motion capture to UE, but it needs to be exported with the UE4 settings. We then select the Retargeter tool, which is part of the third person pack, to convert the UE4 animation to UE5 for compatibility with the Metahuman.

When all of this is done, we configure the Metahuman blueprint to recognise the UE5 skeleton, for a seamless integration.

6 Scenes

We've chosen a three-scene approach to demonstrate the practicality and capabilities of motion capture technology in VR environments. We focus on showing how motion capture technology enhances immersion and simplifies animation workflows. Each scene focuses on individual strengths of motion capture development.

6.1 First scene - Cinematic

In the first scene, we wanted to highlight motion capture's ability to convey subtle emotion and nuanced movement for cinematic storytelling in VR.

In the scene, a single actor wearing our Rokoko MC setup begins the scene seated, then the actor slowly rises from the chair, their posture reflecting a sense of weariness or emotional weight. Actor walks towards the window with a deliberate but hesitant gait. Actor reaches the window, pause, and lowers their head, conveying a sense of sadness.

This scene was tried at many frame rates, but after much trial and error we found that the 30-fps cap worked best for the scene, as it stayed within the traditional cinematic language.

6.1.1 Scene Setup

Firstly, we performed the scene so we had a mocap, and could export it into the scene, then we imported the metahuman and adjusted the animation as in the was mentioned in the previous sections.

The second part was to create an environment that felt dark and gloomy. To achieve this type of environment, we used assets from Quixel Bridge and Epic Game Marketplace and RealityCapture. Once we were happy with the basic look of our house, we started to add some ominous effects such as adjusted lighting, ExpontentialHeightFog and many other features from UE to make our scene fit our taste.

With the environment ready, we needed to set up our Metahuman. We add the UE Metahuman Riggs to him, this must be done in the Sequencer. These riggs were used to fine-tune the movement of our characters to the environment. For example, to prevent the Metahuman's hands from sinking into the sofa when he stands up from the chair. Then we would polish the scene and move the furniture and the metahuman around to get closer to what we envisioned.

Once the animation looked good, we started to adapt our project to a VR environment. In this scene there should be no movement from the user, so we decided to make the VR pawn possess the camera that follows the scene, so that it has a more cinematic perspective.

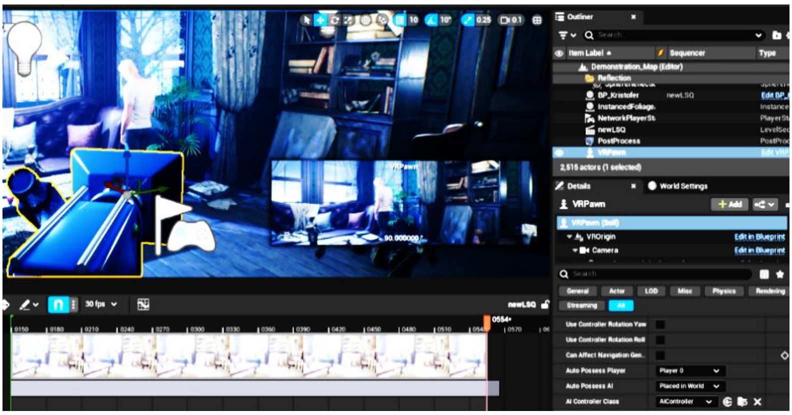


Fig.1. First scene.

6.2 Second Scene - Movement

In the second scene, we wanted to demonstrate and highlight MOCAP's potential to make exercise and rehabilitation more engaging and interactive.

The scene begins with an actress who is performing a pre-arranged set of exercises in a bright and airy virtual gym environment, with upbeat music playing in the background.

6.2.1 Scene Setup

The actress begins by turning towards the VR user with an energetic wave of her right hand. The actress then transitions into the exercises, performing three sets of exercises, after which she gives the user a thumbs up as positive reinforcement. The movement ends with the actress mirroring the initial wave with her left hand. This animation is played in a loop, so the user can watch the exercises as many times as they like, so it's important that the actress returns relatively to the same point from which she started, so that the animation loop switch is unseen.

In the same way as before, we created a simple gym environment with afternoon lighting, adding floor, walls, roof, lighting and exercise equipment.

The VR implementation in this scene is vastly different from the first scene as the player needs to be able to move around. We chose to use teleportation-based locomotion. We added the VR pawn to the scene and set it to autopossess for the user, then we needed to mark the space where our player could move. We used the NavMeshBoundsVolume for the boundaries and its counterpart the NavModifierVolume to create restricted areas for the user.

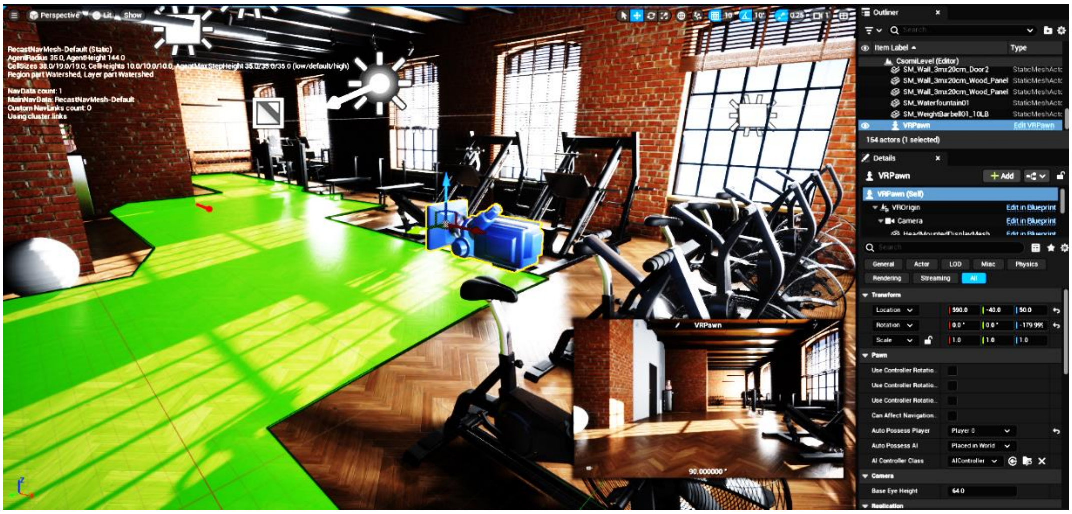


Fig.2. Second scene.

6.3 Third Scene

This scene focuses on livestreaming mocap to metahumans to provide a better real-time experience for learning or social activities in a VR environment for multiple users.

6.3.1 Scene Setup

The first thing we did was to set up the livestream. We started by opening the Metahumans blueprint and going to AnimGraph where we added two variables. The first is a retarget asset, so it should be a Rokoko body map data with class "reference", the other is a simple name variable. After that we renamed the UE blueprints actor name to match the Rokoko actor name that will be livestreamed. When this was done, we applied the predestined Metahuman_Bonemap to the Rokoko body pose and compiled it.

After compilation, our metahumans' body positions remained the same, which was a problem because every mesh that is livestreamed by Rokoko needs to have a T-pose starting position to receive the data correctly. So, we used the pose asset we'd already downloaded with the Rokoko resources. We dragged it into the metahuman blueprint and set our details to it, then set the animation mode to use our animation blueprint, which made our metahuman ready for livestreaming.

We set up our Rokoko studio and enabled livestreaming. We selected the 14043 port, then opened the live link in UE where the 14043 studio port appeared. With the livestream working, it gave us two assets, the face and the body MC. We added them to the metahuman, which allowed our metahuman to move in the livestream.

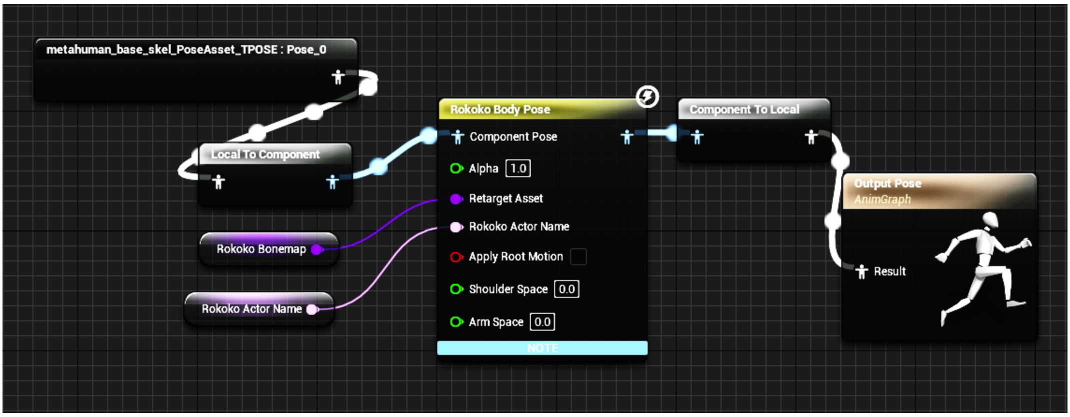


Fig.3. Live Stream Blueprint.

There are many ways to add multiplayer to a game, so we decided to use the Steam online subsystem. First, we converted our UE project into a C++ project. Then we added the Steam plugin, we used a slightly modified version called Rissnsy/steam [13]. This we added directly into our project files, after which we opened our sln file in VS, and built our project to catch errors. Once it was successfully built, we needed to modify the DefaultEngine.ini file to allow the usage of the steam online subsystem for networking features.

```
[/Script/Engine.GameEngine]
+NetDriverDefinitions=(DefName="GameNetDriver",DriverClassName="OnlineSubsystemSteam.SteamNetDriver",DriverClassNameFallback="OnlineSubsystemUtils.IpNetDriver")

[OnlineSubsystem]
DefaultPlatformService=Steam

[OnlineSubsystemSteam]
bEnabled=true
SteamDevAppId=480

[/Script/OnlineSubsystemSteam.SteamNetDriver]
NetConnectionClassName="OnlineSubsystemSteam.SteamNetConnection"
```

Fig.4. DefaultEngine.ini [14]

The third part was to create a main menu where users would start, and they could choose to create a server, join a server, or simply quit. The first step was to create a simple main menu level that would load before the playable levels. After using the UE widgets to create the layout for the buttons, we created four widgets and a player controller blueprint and a steam session blueprint. Once the layout was done, we needed to make these widgets work, so we used the UE blueprints to program these buttons.

The first step was to make them visible, so we opened the Player Control blueprint and created a simple function to add our widgets to the viewport. Now that the buttons were visible, we needed to create a Main Menu blueprint, which consists of three simple functions, as this widget only takes us to our chosen widget, which are: Create Server, Browse Server, Quit. The blueprint here on the player control removes the main menu from the viewport and adds the preferred widget in its place.

To create the Create Server function, we needed to adapt the host session event in the Steam session blueprint. In the Create Server function, our main job is to take the ServerName and the AmountOfSlots variable, convert them into a usable form, and then use the on_clicked create button function to pass it to the host session, which leads into the Steam Session Blueprint. Here we import the CreateAdvancedSessionSteam function that comes with the Steam plugin and pass it our session information, if everything checks out it will create a server level.

The last one is the server browser, which consists of two parts. The first one is a ServerCardBP which needs data from the host "Server Name, Current Players, MaxCurrentPlayers, Ping" and fills the button with this information and it also has a button function which on click joins the session where the data came from. The second is the ServerlistBP, which consists of a simple back button and a clickable refresh button, which clears the previously found session and calls the ServerCardBP to fill in the newly found sessions.

Once everything was prepared, we went back to the main map and created a class environment and added possessable VR pawns to prevent all the users from being spawned at the same point. We created the movement area using NavMeshBoundsVolume, just like in the second scenario.

Then everything in the scenario looked just as we imagined, we exported our projects with UE and joined from other PCs.

Conclusion

As part of this work we have deepened our understanding in working with motion capture and the process of creating scenes with it. In conclusion, the project was successful in achiveing the goal set at the beginning of the project.

Acknowledgement

I would like to thank my thesis supervisor, Mr. RNDr. Ján Lacko, PhD., for his professional guidance, consultation, valuable advice, and comments that helped me in the preparation of this work.

References

[1] Greenberg, R. (2018). How animation won over the Lightning Sketch: Re-Evaluating humorous phases of funny faces. Animation, 13(2), 162–174. https://doi.org/10.1177/1746847718783641

[2] Cartwright, L. (2012). The hands of the animator: rotoscopic projection, condensation, and repetition automatism in the Fleischer apparatus. Body & Society, 18(1), 47–78. https://doi.org/10.1177/1357034x11432562

[3] Zhou, H., & Hu, H. (2008). Human motion tracking for rehabilitation—A survey. Biomedical Signal Processing and Control, 3(1), 1–18. https://doi.org/10.1016/j.bspc.2007.09.001

[4] Chatzitofis, A., Zarpalas, D., Kollias, S., & Daras, P. (2019). DeepMoCaP: Deep optical motion capture using multiple depth sensors and Retro-Reflectors. Sensors, 19(2), 282. https://doi.org/10.3390/s19020282

[5] Lee, C. J., & Lee, J. K. (2022). Inertial Motion Capture-Based Wearable Systems for Estimation of Joint Kinetics: A Systematic Review. Sensors, 22(7), 2507. https://doi.org/10.3390/s22072507

[6] Lee, J., Shin, S. Y., Ghorpade, G., Akbas, T., & Sulzer, J. (2019). Sensitivity comparison of inertial to optical motion capture during gait: implications for tracking recovery. ICORR. https://doi.org/10.1109/icorr.2019.8779411

[7] Desmarais, Y., Mottet, D., Slangen, P., & Montesinos, P. (2021). A review of 3D human pose estimation algorithms for markerless motion capture. Computer Vision and Image Understanding, 212, 103275. https://doi.org/10.1016/j.cviu.2021.103275

[8] Gohel, V., & Mehendale, N. (2020). Review on electromyography signal acquisition and processing. Biophysical Reviews, 12(6), 1361–1367. https://doi.org/10.1007/s12551-020-00770-w

[9] Field, M., Stirling, D., Naghdy, F., & Pan, Z. (2009). Motion capture in robotics review. International Conference on Control and Automation. https://doi.org/10.1109/icca.2009.5410185

[10] Full body motion capture integrating body, finger and face capture in one intuitive mocap system. (n.d.). Retrieved online, February 12, 2024 https://www.rokoko.com/products/full-performance-capture

[11] NeuronMocap. (n.d.). Perception Neuron Studio - Professional and precise MOCAP suit. Retrieved online, February 18, 2024 https://neuronmocap.com/pages/perception-neuron-studio-system

[12] XSENS MVN Link: High-Performance motion Capture suit for professionals. (n.d.). Retrieved online, March 1, 2024, https://www.movella.com/products/motion-capture/xsens-mvn-link

[13] Risensy. (n.d.). GitHub - Risensy/Steam. GitHub. Retrieved online, March 5, 2024, https://github.com/Risensy/Steam

[14] Online subsystem Steam. (n.d.). Retrieved online, March 12, 2024, https://docs.unrealengine.com/4.26/en-US/ProgrammingAndScripting/Online/Steam/

Authors

Mgr. Csongor Mészáros
Student at the Faculty of Informatics,
Pan-European University in Bratislava, Slovakia
Csongoralma0@gmail.com
The focus of his studies is oriented to front-end programming, digitalization, and the utilization of possibilities provided by Virtual and Augmented reality.

International Journal
Information Technology Applications (ITA)

Volume 13, Number 1, September 2024

INTERACTIVE WEB GAME

Michal Heban

Abstract:

This thesis deals with the development of a computer game, specifically a turn-based strategy adventure game, in which the player controls his character in a web-based environment. By harnessing the potential of web technologies, we seek to overcome the traditional barriers of computer games by enabling play anywhere. Thanks to the accessibility of the Internet, players can play the game on their favorite devices, regardless of location or time. The key concepts of our game will be uncovering maps, collecting a variety of gear, creating unique items, and engaging in battles using spells and various tactical elements. These elements are designed to provide a deep and immersive experience, encouraging players to strategize and plan their moves carefully. By designing comprehensive functionality and attractive visual design enriched with 2D graphical elements, we aim to create a visually appealing environment that enhances the overall player experience. The game's art style and user interface have been carefully crafted to ensure they are both engaging and intuitive, allowing players to easily navigate and enjoy the game.

Keywords:

React, game, pixel art, adventure, Next.js

Introduction

In the realm of computer gaming, we are witnessing a continual rise in popularity, driven by increased interest in interactive forms of leisure, often revolving around digital entertainment. With the advancement of the internet and its widespread availability, the use of web platforms for gaming is becoming increasingly common. The advantages of these platforms include the opportunity to play games without the need for downloading or installing necessary software.

Therefore, in our work, we will focus on the creation of an interactive RPG turn-based strategy implemented in a web environment. We will cover multiple domains starting with theoretical foundations essential for game development.

Following the theoretical groundwork, our next step involves determining the genre and style, refining the identity and aesthetics of the game. Subsequent to this stage, we move on to the practical process of prototyping various screens and user interfaces to materialize conceptual ideas.

After completing the prototyping phase our focus will shift towards the implementation stage, where we translate design into functional components, bringing our interactive web game to life within the web environment. Lastly we will conduct thorough testing to ensure the smooth functioning of all components, verifying that each element operates seamlessly withing the game.

1 Web Development

With the rise of web applications, the way people interact with software has undergone a significant transformation. Unlike traditional native applications that require installation and operate locally on specific operating systems, web applications only necessitate a functional web browser and internet access, regardless of the operating system, as they function on remote servers. This connectivity enables access from various devices, ranging from smartwatches and smartphones to computers and smart TVs.

The web application operates on the client-server principle and consists of two main parts: frontend and backend. The frontend corresponds to the client-side, which focuses on the user and is responsible for the visual content of the application and its interaction with the user. The primary element is the graphical user interface (GUI), which includes buttons, multimedia content, and forms.

The backend, the server-side component of the application, is responsible for the application logic, data storage, and all associated calculations. It commonly communicates with the front end through HTTP requests, which are processed on the server side and returned to the client side as a response [1]. In some cases, the server may further connect to data sources or other services necessary for request processing.

In today's web application development, various frameworks are commonly used to assist programmers in building applications. These frameworks offer numerous advantages throughout the development process. They provide consistency, efficiency, maintainability, and scalability for the project.

Most popular frontend frameworks are React, Angular and Vue.js, where React is most popular among them according to trends on StackOverflow. In terms of performance, React also emerged as the winner. The comparison was conducted on three identical pages, where each was created in corresponding framework [2,3,4]. The comparison was utilizing Core Web Vital metrics obtained from the PageSpeed Insight tool. This includes First Contentful Paint, Largest Contentful Paint, Total Blocking Time, Cumulative Layout Shift, and Speed Index. The result can be seen in the table below (Tab.1).

Table 1. Comparison of Core Web Vitals for frontend frameworks.

	React	Angular	Vue.js
First Contentful Paint (s)	0.3	0.4	0.4
Largest Contentful Paint (s)	0.9	1.1	0.4
Total Blocking Time (ms)	0	0	170
Cumulative Layout Shift	0.045	0.045	0
Speed Index (s)	0.7	0.9	0.9
Overall performance	99	98	95

2 Main Concepts of the Game

In game development, genre and style are pivotal choices. We've opted for an adventurous RPG set in a fictional world dominated by magic. As for style, we're embracing a text-based approach with pixel graphics for a retro vibe, fostering player imagination. With a dark theme consisting of black background and white text, the game will be known as "Adventurer Chronicles".

Whole gameplay will be centered around player’s character, where player can choose from three classes, with each class offering three specializations. First class will be Warrior which is divided into these specializations: Guardian knight, Berserker and Shield Bearer.

Second class will be Mage, with specializations for being Elemental mage, Demonomancer or Druid. Lastly there will be Master of Weapons which offer these specializations: Sword Master, Sharpshooter, Assassin.

The game world will be expansive, comprising regions with towns which will be used for player to rest, get new gear, train new spells or taking quests. On other hand for battle and adventure there will be multiple locations, each with its own set of monsters specific to that area.

Attributes such as strength, agility, intelligence, endurance, and vitality further define the player's capabilities. Alongside these attributes, a player's strength will be determined by their level, equipment, and spells, which can vary in level and quality. Higher quality items provide greater strength. Players can advance their level by accumulating the necessary experience for leveling up. Attributes can be enhanced either through leveling up or by consuming specific items tailored for this purpose. Equipment can be acquired by defeating monsters, unlocking treasure chests, or purchasing them from in-game shops. Additionally, players will have the option to craft their own equipment and items using professions like alchemy or blacksmithing, further enriching the gameplay experience.

In the game, there are two types of currency: basic and premium. Basic currency is used for all aspects of the game, from buying items to training spells or abilities. It can be obtained through missions, looting monsters, or selling unwanted gear or items. Basic currency includes bronze, silver and gold coins. Premium currency, acquired from specific powerful monsters, is used for purchasing superior bonus items or enhancing player abilities. Premium currency includes diamonds for exclusive items and magic stones for exchanging or upgrading items.

Materials are items used in crafting equipment or consumable items. Players can acquire them through various means, similar to spells or equipment. They can be obtained from shops, missions, or as loot from monsters. Collecting materials from specific locations will also be a significant method of acquisition.

Each player will have access to three basic resources. Experience directly influences the player's level, serving as the main indicator of the player's success and progress in the game. Among the active resources crucial for the player's survival and success will be health and mana. Health represents the player's physical condition and determines how much damage the player can withstand before being defeated. Players will lose health during battles with monsters and will have the opportunity to replenish it using various items or actions. Mana is a mysterious energy serving as the fuel for all spells. Different levels of spells require different amounts of mana. Similar to health, mana can be replenished using items or actions.

Spells will serve as the primary resource for combat and will play a significant role in the game. Each spell will have its own level, which can be further enhanced to increase its potency. Additionally, spells can be upgraded to unlock higher tiers, making them even more powerful. However, casting spells will consume mana.

3 Prototypes and Data Model

Creating prototypes for game screens and defining a data model are vital steps in game development. They enable developers to gather feedback before final implementation, ensuring the final product meets expectations. Additionally, prototypes help maintain workflow continuity by providing clear direction for developers, reducing uncertainty about the next steps in the development process.

3.1 Prototypes

To create prototypes for our game, we utilized the MockFlow tool. We designed prototypes for all important sections of our game starting with home screen.

The home screen will serve as an introduction to the game, providing users with information about the game and options to either register or log in. Both the registration and login pages will feature simple forms.

Upon logging in, users will be directed to the dashboard, where they'll find comprehensive details about their character, including name, class, specialization, level, experience, health, and mana. Additionally, the dashboard will feature sections dedicated to character attributes, active spells, and in-game currencies (Fig.1). For new users without a created character, an option to do so will be provided.

Fig.1. Prototype for dashboard together with header.

The next screen is the map screen. The map is divided into two main parts: crossroads and specific towns or locations. At the crossroads, players are located if they have not entered any town or location yet. On crossroad the player has the option to choose new location or town to travel to or to enter the current place. If the player is in a town, a brief description of the town along with its buildings will be displayed. Each building has its own screen, including its name, description, and actions based on the type of building. Buildings include, for example, the town hall for accepting and handing out quests, the training hall for learning and practicing spells, shops, or professional halls. The location contains a list of enemies along with the option to battle or gather resources.

The inventory will be divided into four main sections, which look almost identical, all having a filter along with a table of items or equipment that the player owns. These sections are for equipment, items, spells and materials.

On the spells screen, the player will switch between two sections: active spells and all learned spells. Active spells show how many spells the player can still activate, their level, name along with the number of experience points, and information about the spell. It also allows the player to upgrade the spell if all conditions are met. All learned spells will include a filter and the option to set some spells as active.

The attributes tab will serve to provide a more detailed view of all attributes, both primary and secondary, along with all character bonuses.

The appearance for each profession will be the same, differing only in text. Within each profession, there will be further categorization into research and crafting. Crafting will display a list of recipes with their levels, materials needed for their production, and crafting time. Below the recipes, there will be a current production list with completion times. In the research section, players will have the option to select materials and attempt to discover new recipes by clicking the research button.

3.2 Data Model

The data model for our game is expected to be quite robust, with an initial estimate of around eighty tables, some of which may contain over ten columns. We used the QuickDBD tool for creating the data model. The data model can be divided into several components.

The first set revolves around the player's character and the player themselves. It includes the table for the player's character, containing fields such as character name, class, specialization, level, experience, current health or mana, number of active spells, and location. This table also connects to several others. Among these tables is one for the player's currency status. There's also an attributes table, recording their values along with the number of points the character has available for allocation. Following that is the spells table, which holds information about the player's spells, including their level, experience, grade, spell status, and availability. While this table doesn't include other spell data, it references another table for that information. Similarly, a table for the player's equipment, materials, or items will be associated with them.

The subsequent set of tables is related to the game world. This may include records of cities, regions, locations, as well as buildings in the city.

Next in line are tables dealing with professions. The first one will contain data about players and their professions. The second will store information about the player's learned recipes or guides. The last set will comprise the list of products the player has entered for crafting. Auxiliary tables in this case will include the recipes themselves and the materials needed for their production.

4 Implementation

Before diving into implementation, it's crucial to choose some fundamental elements such as development environments, technologies, libraries, and programs we'll use for development. We'll start with the development environment. Given our experience with JetBrains development environments, we've decided to use WebStorm for web application development and DataGrip for database management. As a framework for building the application, we've opted for Next.js with TypeScript. Next.js is a library built upon React and provides many methods for server-side rendering, which can significantly boost the performance of our game. For styling, we've chosen the Tailwind CSS library, which allows us to quickly incorporate CSS classes for any element and achieve the desired appearance.

For code cleanliness and adherence to set rules, we relied on the ESLint library. For authentication and authorization, we utilized the NextAuth library, which seamlessly integrates into a Next.js project. Input field and form validation were handled using the zod library. For data handling we chose a PostgreSQL database and utilized the Prisma ORM library. Graphic elements were created using the Aseprite program.

The server-side portion was implemented using the serverless architecture provided by Next.js. Finally the whole application will be deployed to the Vercel platform for its simplicity in deploying Next.js projects, as it's a framework developed by Vercel.

All the screens were implemented based on visuals from prototyping, with minimal to no changes.

Data handling from the server was managed using the fetch function along with getServerSideProps from the Next.js library, facilitating server-side rendering when needed. Some data which were used throughout the entire application was stored using the context functionality, allowing access from anywhere within our game. This was particularly beneficial for managing character information such as experience, health, mana, spells or equipment. Additionally, it ensured a consistent and efficient data flow, enhancing overall performance and user experience.

For registration and user sessions, we utilized the authentication function from the NextAuth library. This library not only stores data from the registration process but also manages user sessions, ensuring that players do not have to log in every time they return to the game. By leveraging the credential-based method for login, we provided a secure and seamless experience. This approach simplifies user access and enhances the overall usability of the game.

Overall performance was improved by utilizing memorization techniques, which helped to reduce redundant calculations and enhance efficiency. This optimization was particularly beneficial for handling frequently accessed data and rendering complex components. In our game it was used for data table as material or equipment list in inventory or some data calculations.

Navigation throughout the application was achieved using the routing capabilities provided by Next.js. This system utilizes the folder structure in the project, where every folder within the main app directory that contains a page.tsx file is recognized as a route. This approach simplifies the routing process and helps maintain an organized and easily navigable codebase. A similar approach is employed for the serverless architecture, where each folder within the api directory serves as an endpoint for API calls. Within these folders are files named route.ts, containing the logic executed when an API call to this endpoint is made.

We developed several reusable components for various aspects of the application, such as filters or custom bars for resource management. Additionally, each spell, material, equipment or building had a corresponding pixel art icon, enhancing the visual depth of our text-based game. Not only icon but also some bigger elements had its own art, for example scroll which shows spell information (Fig.2).

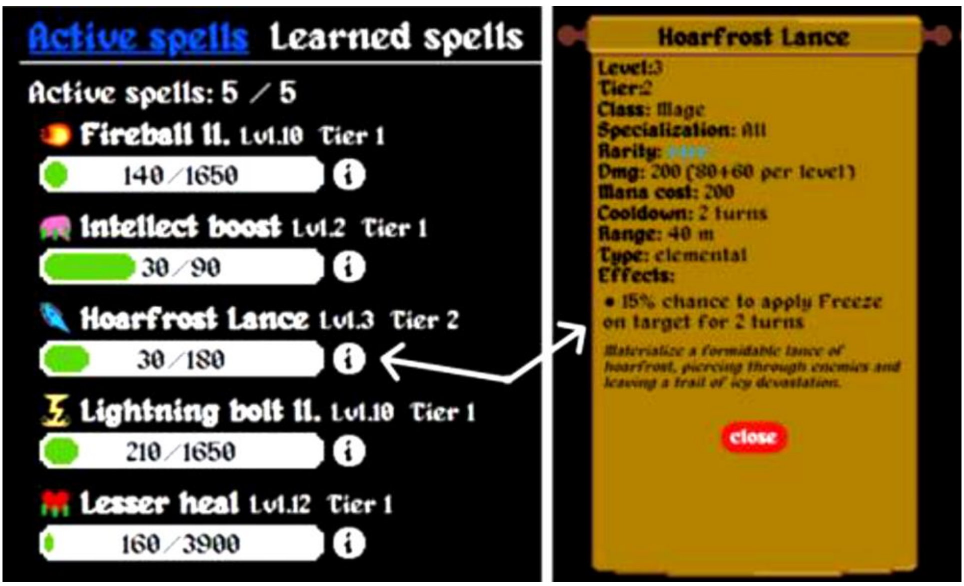


Fig.2. Art style example of scroll with spell information and spell icons.

The main combat mechanics were implemented based on the principles of a turn-based game. Each side, whether player or monster, would take turns using spells, abilities, or items. Monster abilities were chosen dynamically based on an algorithm that selected the most suitable spell for the situation at that moment. The outcome of the fight was also influenced by buffs and debuffs that could be applied using spells or items. The battle concluded when the health of one side reached zero or below.

5 Testing

The entire application underwent manual testing, where we tested the complete functionality and addressed any errors. Additionally, we conducted usability testing with ten respondents playing the game on a test server. Two of them had no experience with gaming, three were occasional gamers and five were gaming enthusiasts.

Based on their feedback, we fine-tuned aspects of the game where issues or inconsistencies were found. The most significant concern among players, especially those with less experience, was the lack of a robust tutorial. Since the game featured numerous sections with actions, not all were thoroughly explained. Some players expected a confirmation email after registration, but this feature was not part of the implementation and was not added at the time. Four out of ten players expressed dissatisfaction with the limited number of items, spells, and equipment in the game, so additional content was developed. Crafting these items was time-consuming as all graphical assets were created manually. While this aspect did not affect gameplay, it enhanced the overall player experience.

By adhering closely to our concepts and prototypes without deviation, we encountered minimal errors throughout the development process.

Conclusion

The browser-based game Adventurer Chronicles combines elements of a text-based game enriched with 2D graphics, allowing players to control their character, complete tasks, collect spells, equipment, items and engage in turn-based combat with enemies. All project objectives, such as the game's appearance, functionality, and theme, were successfully achieved, as defined during the conceptualization process, including the design of individual screens and corresponding user interfaces. The resulting game is fully playable on all devices, from mobile phones to computers and smart TVs with internet access, without the need for additional downloads or installations.

Acknowledgement

When working on this project, I would like to express my gratitude to Mr. RNDr. Ján Lacko, PhD. who provided me with valuable advice and consultations throughout the process.

References

[1] Kornienko, D.V. et al. The Single Page Application architecture when developing secure Web services. In Journal of Physics: Conference Series 2021. Vol. 2091, no. 1, s. 012065.

[2] Vue Landing Page. Retrieved online, May 18, 2024 https://logrocket-vue-landing-page.ver-cel.app

[3] React App. Retrieved online, May 18, 2024 https://logrocket-react-landing-page.vercel.app

[4] LogrocketAngular. Retrieved online, May 18, 2024 https://logrocket-angular-landing-page.vercel.app

Authors

Bc. Michal Heban
Faculty of Informatics,
Pan-European University in Bratislava, Slovakia
michal.heban52@gmail.com
Student of Faculty of Informatics on Pan-European University, with extensive knowledge in modern web development.

International Journal
Information Technology Applications (ITA)

Volume 13, Number 1, September 2024

SYNCHRONIZATION OF SPEECH FOR SYNTHETIC AVATARS IN VIRTUAL REALITY

Lóránt Boráros, Ján Lacko

Abstract:

The goal of this research paper is to review existing methods of voice synchronization for avatars in virtual reality environment and to implement one of the existing methods or to design a new method. To create voices for avatars we will use AI powered automatic text-to-speech generators. The output of this research paper will be the design and implementation of an interactive application in virtual reality.

Keywords:

Virtual reality, generative AI, animation, synchronization, avatar.

Introduction

Virtual Reality encompasses the interaction between participants and creators within a simulated environment, relying on components such as virtual environments, immersion, and interactivity. This medium is shaped by the unique perspectives of its participants and the inventive design efforts of the creator, resulting in diverse and dynamic VR experiences. Avatars are fundamental parts of this process, as they serve as the digital representation of the user, enabling the interaction with the objects within the virtual space. Recent advancements in generative AI models, such as ChatGPT, have revolutionized text generation and language processing by enhancing the sophistication and realism of human-machine interactions. This paper explores the integration of these technologies by developing an interactive VR application that utilizes AI-powered communication, synthesized speech, and synchronized animations to create a comprehensive and immersive user experience. The proposed application leverages Unreal Engine, ReadSpeaker.ai and NVIDIA Audio2Face to achieve seamless interaction and engagement, showcasing the synergy between VR and AI-driven innovations.

1 Virtual Reality

The fundamental components essential for experiencing virtual reality, or any reality, encompass the virtual environment, immersion, interactivity, and the individuals involved in both creating and experiencing the medium.

Participants and Creators in VR: The essence of any virtual reality (VR) experience lies in the interaction between participants and creators. Participants, as the primary actors within the virtual realm, bring their individual perspectives, backgrounds, and cognitive abilities to shape their unique experiences. This diversity underscores the dynamic nature of VR encounters, wherein each participant engages with the virtual world in a distinct manner.

While human participants are the focus of this discussion, VR experiences can extend to non-human entities, as evidenced by experiments involving diverse subjects such as fish, cockroaches, and praying mantises. On the other hand, creators play a crucial role in designing and implementing VR applications and systems. Referred to as composers or developers, these individuals or teams craft the virtual environments that participants engage with, facilitating immersive and interactive experiences.

Virtual World as Content: Central to the VR experience is the concept of the virtual world, which encompasses the content presented through the medium. Whether existing solely within the creator's imagination or shared with others, a virtual world can be described as a collection of objects, rules, and relationships governing interactions within a simulated space. Unlike physical reality, a virtual world can take various forms, from imaginary settings to representations of real places. The transition from a conceptual description to a tangible experience occurs through the utilization of VR systems, integrating hardware, software and content to immerse participants in virtual environments.

Immersion in Alternate Realities: Immersion serves as a cornerstone of the VR experience, allowing participants to engage with alternate realities or viewpoints distinct from their physical surroundings. Within the context of VR, immersion entails both mental and physical engagement, wherein participants perceive and interact with synthetic stimuli presented by the VR system. Mental immersion, often referred to as a sense of presence, denotes a deep level of engagement and suspension of disbelief within the virtual environment. Physical immersion, on the other hand, involves bodily interaction with VR technology, enabling participants to navigate and manipulate the virtual space through sensory feedback.

Interactivity and Participant Agency: Interactivity forms a vital component of VR, enabling participants to influence and shape their virtual experiences through actions and choices. Whether through direct manipulation of virtual objects or changes in viewpoint within the virtual world, interactivity empowers participants to actively engage with the VR environment. This interactive paradigm distinguishes VR from traditional media forms, facilitating dynamic and participatory storytelling. While computer graphics often enhance interactivity, text-based interactive fiction exemplifies an early form of interactive storytelling that predates graphical VR experiences.

Dynamic and Static Worlds in VR: VR environments vary in their degree of interactivity, with some presenting static worlds resistant to participant modifications, while others offer dynamic environments that permit alterations. The emergence of 360-degree movies blurs the distinction between VR and surround media, challenging traditional definitions of immersive storytelling. While some consider interactive viewing of 360-degree movies as a form of VR, others debate the extent to which it constitutes true virtual reality. This ambiguity underscores the evolving nature of VR and the ongoing discourse surrounding its defining characteristics and boundaries. [1]

2 Avatars

Avatars serve as digital representations or virtual selves within a virtual environment, enabling users to interact and engage in activities. According to Bartle (2010), an avatar is the entity through which users channel their in-world activity, allowing them to identify uniquely within the virtual space. Bell and Robbins-Bell (2008) expand this definition, describing avatars as digital representations with agency and real-time control by human agents. These representations, whether graphical or textual, facilitate user interaction and presence within the virtual environment. Bartle's definition emphasizes the importance of avatars as entities enabling interaction and presence detection, rather than merely graphical or textual representations.

In multiuser virtual environments, each user typically adopts an avatar as their representation, allowing other users to detect their presence and engage with them. This representation ensures that users can interact within the shared virtual space, regardless of whether the environment is single or multiuser.

While uniqueness was once considered a criterion for avatars, it has been dismissed in favor of facilitating interaction and engagement among users. Avatars, as defined by Bartle, serve as virtual selves within the virtual environment, providing a medium for interaction and engagement between human agents and the virtual space.

The concept of avatars extends beyond mere graphical or textual representations; they emcompass the user's presence and agency within the virtual environment. Avatars enable users to navigate and interact within the digital space, enhancing immersion and facilitating communication. Ultimately, avatars play a crucial role as human-computer communication interfaces, bridging the gap between users and virtual environments, and enabling meaningful interaction and engagement. [2]

2.1 MetaHuman

The process of generating photorealistic human characters for Unreal Engine 5 (UE5) can be achieved through the utilization of MetaHuman Creator, a cloud-based application developed by Epic Games. MetaHumans, characterized as intricately detailed 3D models of human subjects, are tailored and refined using this platform. Within the MetaHuman Creator interface, users are afforded extensive customization options for facial attributes, including features such as facial textures, hairstyles, body shapes, age and gender. Additionally, a diverse array of clothing options and accessories are available for selection, further enhancing the realism and individuality of the digital human representations. [3]

3 Generative AI Models

Generative AI models, like text generators and language processors, have revolutionized natural language processing by enabling machines to produce text that closely resembles human writing. These models, such as OpenAIs ChatGPT, utilize advanced deep learning techniques to perform tasks like text generation, translation, summarization, and question answering with remarkable accuracy and coherence.

3.1 Large Language Models

Large Language Models (LLMs) stand as a significant leap forward in the realm of natural language processing and artificial intelligence (AI) research. They have revolutionized machine capabilities, enabling them to comprehend and generate language in a manner akin to humans.

Leveraging deep learning methodologies and extensive datasets, LLMs have showcased remarkable proficiency across diverse language-oriented tasks, including but not limited to text generation, translation, summarization, question answering, and sentiment analysis.

The history of LLMs traces back to the early days of language model development and neural network exploration. Initial endeavors relied on statistical techniques and n-gram models, albeit limited by their inability to capture extensive contextual dependencies in language. The advent of neural networks, coupled with the availability of vast datasets, spurred the exploration of more sophisticated approaches. [4]

3.2 ChatGPT

OpenAI's creation, ChatGPT, represents a modern language model built on the GPT-3.5 architecture, trained extensively on a vast corpus of internet-derived text data spanning books, articles, wikis, and websites.

Renowned for its adeptness in generating human-like responses and engaging in meaningful conversations with users, ChatGPT stands as a testament to the remarkable capabilities of contemporary language models. In the field of computer vision, researchers are developing vision-language models inspired by the capabilities of ChatGPT. These models aim to enrich multimodal dialogues by seamlessly integrating both visual and textual information, thereby enhancing the model's understanding and response generation in diverse contexts.

The ongoing advancements in this field have led to the emergence of GPT-4, representing a significant milestone in language model evolution. GPT-4 builds upon its predecessors by further augmenting the model's capabilities through the seamless integration of visual data as an integral part of the input. This pivotal integration of visual cues empowers the model to comprehend and generate responses that reflect a nuanced understanding of both textual and visual information. Consequently, GPT-4 facilitates more contextually rich and sophisticated conversations in multimodal settings, heralding a new era in AI-driven communication interfaces. [4]

4 Text-to-Speech Generation

As the demand for text-to-speech generation tools grows at an unprecedented rate, new methods are developed involving exciting new approaches and solutions. We present two models using Mel-spectograms to synthetize speech from text inputs.

4.1 Tacotron 2

Mel-spectrograms are a type of spectrogram that represents the frequency content of audio signals in a way that mimics the human auditory system's response. They are derived from the short-time Fourier transform (STFT) magnitude, with the frequency axis transformed using a nonlinear scale based on human perception. Mel spectrograms emphasize lower frequencies critical for speech intelligibility while de-emphasizing higher frequencies dominated by noise bursts. This makes them a preferred choice in speech processing tasks like speech recognition and text-to-speech synthesis for their ability to capture essential acoustic features effectively.

Tacotron 2 is an innovative neural network architecture designed for text-to-speech synthesis. At its core, Tacotron 2 consists of a recurrent sequence-to-sequence feature prediction network coupled with a modified WaveNet model acting as a vocoder. The feature prediction network translates character embeddings into mel-scale spectrograms, while the WaveNet vocoder generates time-domain waveforms from these spectrograms. By employing mel spectrograms as conditioning inputs instead of traditional linguistic and acoustic features, Tacotron 2 streamlines the speech synthesis process and achieves remarkable results.

One of the key contributions of Tacotron 2 is its ability to produce speech that closely resembles human speech quality. With a mean opinion score (MOS) of 4.53, Tacotron 2 achieves comparable results to professionally recorded speech, showcasing its effectiveness in generating natural-sounding audio. Moreover, the study presents ablation studies that validate the design choices behind Tacotron 2, demonstrating the impact of using mel spectrograms as conditioning inputs to WaveNet. This compact acoustic representation not only simplifies the speech synthesis pipeline but also leads to a significant reduction in the size of the WaveNet architecture.

Overall, Tacotron 2 represents a significant advancement in text-to-speech synthesis technology. By leveraging neural network architectures and mel-spectrograms, Tacotron 2 offers a streamlined and efficient approach to generating high-quality speech directly from text inputs. With its ability to produce natural-sounding audio comparable to human speech, Tacotron 2 opens up new possibilities for applications requiring speech synthesis, ranging from virtual assistants to accessibility tools and beyond. [5]

4.2 WaveGlow

The WaveGlow model presents a novel approach to high-quality speech synthesis from mel-spectrograms, integrating principles from Glow and WaveNet. Unlike traditional auto-regressive models, WaveGlow employs a flow-based architecture, simplifying both implementation and training by utilizing a single network trained solely on maximizing the likelihood of training data.

In speech synthesis, both real-time performance and quality are crucial. WaveGlow addresses these challenges by eliminating auto-regression, which tends to limit synthesis speed and quality. By directly modeling the distribution of audio samples conditioned on mel-spectrograms, WaveGlow sidesteps the need for complex sequential processing, achieving rapid synthesis rates exceeding 16 kHz without sacrificing quality. This is particularly crucial for applications requiring fast and interactive speech synthesis, where WaveGlow's efficiency and effectiveness shine.

The core of WaveGlow's architecture lies in its flow-based network, ensuring invertibility to facilitate direct calculation of likelihood. The WaveGlow model transforms simple Gaussian distributions into audio distributions, while conditioning on mel-spectrograms for context-aware synthesis. By strategically incorporating mel-spectrograms throughout the network and utilizing early outputs to facilitate multi-scale information integration, WaveGlow achieves both high-quality synthesis and efficient inference, offering a promising solution for real-world deployment in various speech synthesis applications. [6]

5 Animation Generation and Synchronization

Synchronization of speech and facial animation is a fundamental property of immersive and realistic virtual content. Ensuring this feature is crucial and necessary to achieve positive user experiences. The demand for more content grows together with the expectation of achieving higher and higher quality. Chasing excellence unavoidably leads to exploring new ways and developing new methods to the increase the involvement of automatization of these processes.

5.1 Methods to Generate Emotions

We present an innovative solution for real-time 3D facial animation driven by audio input. This model employs deep neural networks to map input waveforms to 3D facial coordinates. Addressing the challenge of generating expressive facial animations directly from vocal audio, this approach concurrently deciphers nuanced facial expression variations not explicitly captured by the audio signal. By training on a modest dataset of 3–5 minutes of high-quality animation data, this model demonstrates robustness across speakers of varying genders, accents, and languages.

In contrast to conventional vision-based performance capture methods, this model offers a cost-effective and logistically simpler alternative. While vision systems remain the industry standard, they entail elaborate setups and significant processing overhead. Moreover, they encounter difficulties in accommodating diverse user inputs in real-time applications. This audio-based approach complements these systems by providing a streamlined solution for generating large volumes of dialogue-driven content, particularly suited for in-game dialogue, virtual reality avatars, and telepresence applications.

This model addresses the challenges of generating natural facial animations solely from audio input. By adopting a data-driven approach and training the neural network end-to-end, we enable the model to learn intricate facial expression patterns and nuances from the training data. Key innovations include a specialized convolutional network architecture tailored for processing human speech, a mechanism to infer emotional states from audio cues, and a sophisticated loss function ensuring temporal stability during animation.

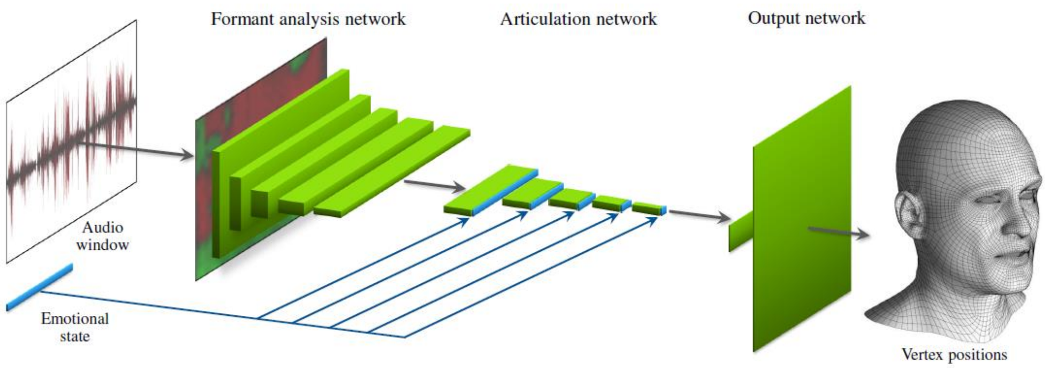


Fig.1. NN model structure to extract facial features and emotions during speech [7].

The resulting model shown on (Fig.1) achieves real-time generation of expressive 3D facial motion, providing per-frame positions of control vertices of a fixed-topology facial mesh. This output format ensures compatibility and flexibility for downstream animation systems while minimizing reliance on specific details. With applications spanning in-game dialogue, low-cost localization, virtual reality, and telepresence, our model presents a versatile and scalable solution for creating compelling facial animations driven solely by vocal audio inputs. [7]

5.2 The Lip-Sync Process

Using the phoneme chart to translate what is heard in the dialog into their audible counterparts, then translation of those phonemes into a predetermined set of visual phonemes are the fundamental steps of lip-sync process. The steps of animating lip syncing are summarized in following points.

Determine the Speech Pattern: The first step involves identifying the speech pattern of the dialog, accounting for dialects and abbreviations to support the accurate assignments of phonemes.

Analyze the Audible Dialog to Determine Phonemes: Audio files are loaded to pinpoint sound occurrences and their corresponding frames, with a scrub tool aiding precision. Real spoken words, including contractions, are transcribed rather than phonemes themselves, facilitating algorithmic translation.

Use Timing Table to Set Frames: Once phonemes are translated and their corresponding frames identified, morphing techniques are applied to transition between visual phonemes. Straight morphing linearly progresses between objects, while weighted morphing blends multiple objects for added expressions.

Getting the Finished Animation: Lip-syncing requires refinement, potentially adjusting poses or timing for realism. Fine-tuning ensures mouth movements align with auditory processing, enhancing believability. Certain letters may be dropped according to guidelines, preserving visual phonetic pronunciation.

Synchronization Time and Artistic Freedom: Setting phoneme sequences at the right time within word pronunciation range is crucial, although precise algorithms may not capture the nuances of human speech entirely. This step balances synchronization and artistic liberty, allowing for efficient results without compromising creative expression. [8]

6 Application Design and Implementation

Our goal is to develop an interactive VR application that generates answers for given text input, synthesizes speech from it, and presents it via an avatar with synchronized lip movement. To achieve our goal, we designed our application using the Unreal Engine and its ecosystem, that provides tools, which incorporate the technologies we discussed in the previous sections.

The proposed structure of our program consist of three major components. The first one is processing the user input gained in the form of text and generates a response. The second component syntethizes the computer generated response into audio format. The third and final component takes the synthetized audio as input, generates the corresponding facial expressions for our avatar and synchronizes it with the played audio in real time.

6.1 Response Generation Using ChatGPT

Generative language models revolutionized the field of text creation and the leading tool is ChatGPT developed by OpenAI. The standard way of communication with ChatGPT involves a chat like interface, as the name suggests. However, using this method third party programs are unable to communicate with it. Fortunately another method has been developed to expand the usability of ChatGPT in the form on API interface. [9]

To use the API interface we need to aquire an API-key from OpenAI for authentication purposes. For this we first must register an account on the OpenAI website and buy some credits as using the API interface is a paid feature. To simplify the internal structure of our communication we will utilize a free plugin called **Complete OpenAI API plugin** accesible from the Unreal Engine Marketplace or from its own GitHub repository. [10]

The purpose of using ChatGPT as our text generation tool is in its ability to realistically replicate humanlike interactions and to utilize its conversational capabilities. Unfortunately, communication through API calls does not maintain information gained from previous interactions, unlike through the chat-based interface, and treats every subsequent API call as the start of a new conversation. This property prevents us from achieving our goals.

To mitigate the negative effects of the API call-based communication we created a small memory in which we save the last three request-response interaction between the user and ChatGPT and append it to every subsequent request sent by the user. This way we provide enough information to maintain the illusion of a sentient conversational partner while limiting the growth of traffic costs induced by the increased request size.

6.2 Speech Synthetization by ReadSpeaker.ai

The demand for accessible speech generation tools surged a wide variety of commercial text-to-speech (TTS) services provided by the top IT corporations. Most notable tools are **Google Cloud TTS** service, **Amazon Polly**, **Microsoft Azure Speech Service** and **IBM Watson TTS** service. All of them provide similar pay-as-you-go service models with comparable prices. Besides these services there are numerous smaller providers like ReadSpeaker.ai, who offers a unlimited free trial for developers of their services, which includes a plugin for Unreal Engine. We ought to mention a notable open-source tool developed by NVIDIA called Tacotron 2, which provides pre-trained models and also a online tool to train custom models with voices provided by the user.

We used the ReadSpeaker.ai plugin for our implementation as it offers an unlimited free trial sample for developers for non-commercial use after filling out a form on their website. Due to this tool the implementation of this component of our application is trivial. We have to extract the output text parsed from ChatGPTs API response and provide it as input for their TTS module. The modul generates the audio as output, which we will save in WAV format for later use.

6.3 Animation Generation via NVIDIA Audio2Face

To showcase Audio2Face AI powered generative abilities we need to import a suitable object to serve as our canvas. For this purpose we will import a MetaHuman character created by the MetaHuman Creator tool, which is accesible from the Unreal Editor by a plugin. This technology provides native integration to NVIDIAs Audio2Face application which works with similar models. To import a MetaHuman we simply add it to our project via the plugin. To add the MetaHuman to our scene we simply drag it and drop it to the desired location.

Audio2Face supports two types of communication. The standard version includes a graphical user interface, which offers a wide variety of settings and manipulation with assets of the scene and massive number of features. The other one is a so called „headless connection“, which supports access to application via command line instructions and API calls, although this variant offers limited access to some of its features. Despite the limitations one of the main advantages of the headless connection is the support of remote access when installed to a server. Our application will use the API interface to generate animations and send animation data from Audio2Face.

To create a new stage in Audio2Face we have to run it with the GUI. After creating the project we have to perform some initial settings, which include the import of the correct 3D model paired with preferred speech generator neural network. The current version offers two AI models, Mark and Claire with the older v1.1 and the newer and more sophisticated v2 versions. When importing the 3D model we have to select to correct one named makr_solved_arkit, which includes the ARKit component as well. It is extremely important to use this model, because the ARKit component drives the animations of the MetaHuman character.

Before we move on we have to activate the **LiveLink** attribute via the object properties window of the audio2face object shown in (Fig.2). We need to activate the StreamLiveLink feature and the Audio Stream feature, as this is the technology that enables us to connect to our application in Unreal Engine and receive realtime animation data.

We also need to enable the LiveLink connection in Unreal Engines, which is done via opening the LiveLink window, which is located under **Window** menu item and **Virtual Production** submenu item. Adding a source to the LiveLink interface is shown on (Fig.3).

There are three important variables for which we need to set the correct values. The first one is the port number which receives the animation stream from Audio2Face. The second one is the Audio Port number, which receives the audio stream from Audio2Face. The third and final one is the Audio Sample Rate, which has the default value 16 kHz. We need to change it to 48 kHz because that is the rate of Audio2Faces output stream. Now our connection is live and we are ready to use it.

To use the features provided by the headless connection of Audio2Face, we will run it locally, which means we can access it on localhost. Documentation is available on 127.0.0.1:8011/docs# where we can find detailed description for every API call, the structure of their request and response message and a “try it out” function. [9]

The last step of our implementation involves the creation and connection of API calls via blueprint nodes in Unreal Engine. We utilized a plugin called **Http Blueprint** to better manage HTTP communication by simplifying tasks such as sending and receiving requests and responses. Moreover, it aids in both constructing and parsing messages.

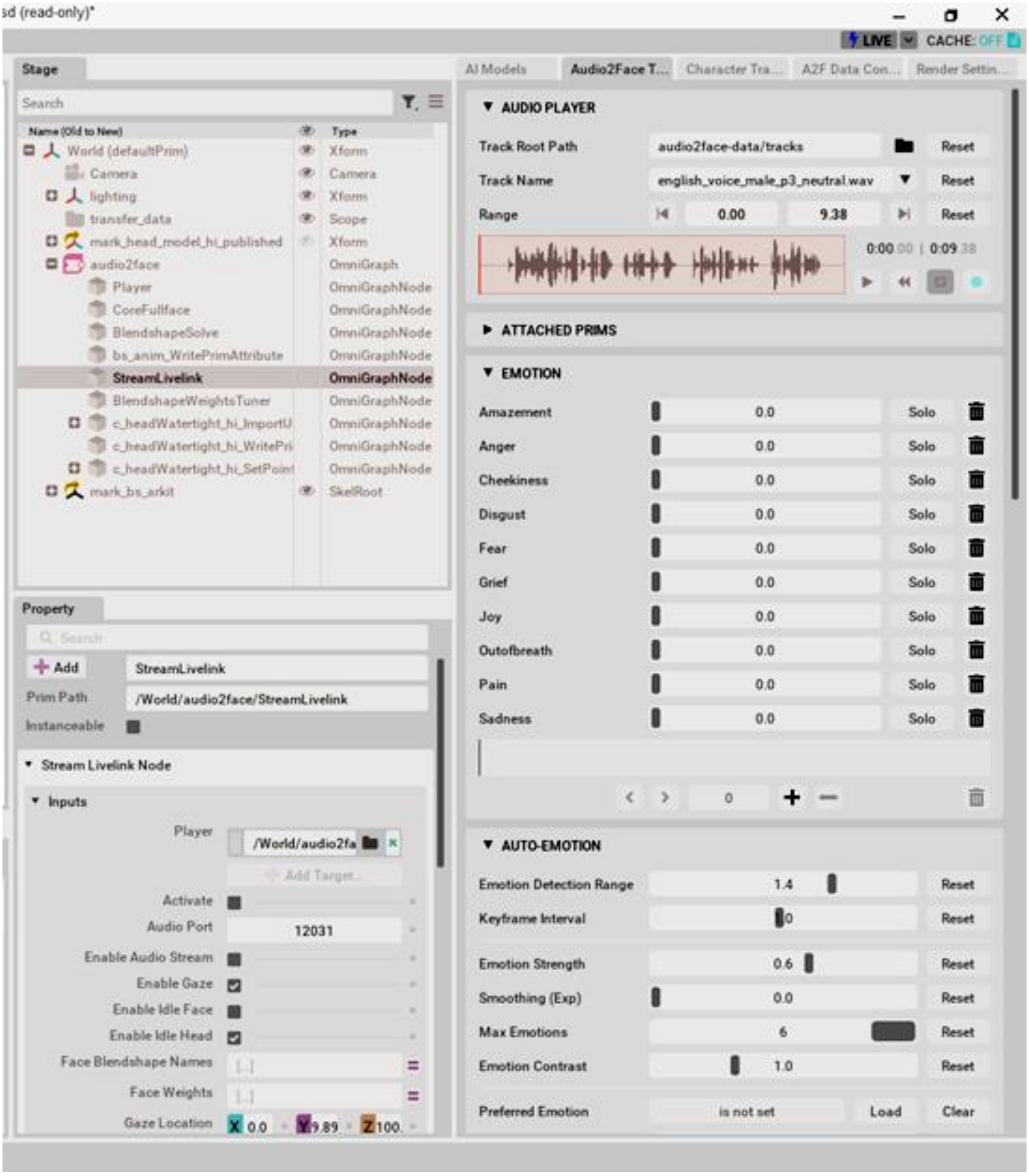


Fig.2. Audio2Face LiveLink settings.

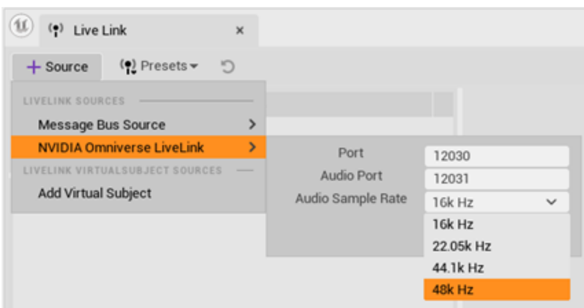


Fig.3. Connecting the NVIDIA Omniverse source to LiveLink.

Conclusion

The goal of this research paper was to present the current state of virtual reality, text-to-speech generation, text and animation generation methods powered by artificial intelligence. Furthermore we managed to design and implement an application which combines all the aforementioned technologies to create a responsive and immersive experience. We succefully achieved our goals.

Acknowledgement

I would like to express my gratitude to Mr. RNDr. Ján Lacko, PhD. my project supervisor, for his invaluable guidance, support, and encouragement throughout the entire duration of this project.

References

[1] W. R. Sherman and A. B. Craig, Understanding Virtual Reality, Cambridge: Morgan Kaufmann, 2018.

[2] K. J. L. Nevelsteen, "Virtual World, a Definition Incorporating Distributed Computing and Instances," *Encyclopedia of Computer Graphics and Games,* pp. 210-220, 2015.

[3] H. Wenter and W. Ogterop, "Using MetaHuman to Create a Photorealistic Human for UE5," in *Unreal Engine 5 Character Creation, Animation, and Cinematics*, Birmingham, Packt Publishing Ltd, 2022, p. 608.

[4] M. Usman Hadi and col., "Large Language Models: A Comprehensive Survey of its Applications, Challenges, Limitations, and Future Prospects," 2023.

[5] J. Shen and col., "Natural TTS Synthesis by conditioning WaveNet on MEL Spectograms," Google Inc., University of California, Berkeley, 2018.

[6] R. Prenger, R. Valle and B. Catanzaro, "WaveGlow: A flow-based generative network for speech synthesis," NVIDIA Corporation, 2018.

[7] T. Karras, T. Aila, S. Laine, A. Herva and J. Lehtinen, "Audio-Driven Facial Animation by Joint End-to-End Learning of Pose," *ACM Transactions on Graphics,* vol. 36, no. 4, pp. 1-12, August 2017.

[8] A. R. Itimad, G. Sulong and H. Kolivand, "Realistic Lip Syncing for Virtual Character Using Common Viseme Set," *Computer and Information Science,* vol. 8, no. 3, August 2015.

[9] "API Reference-OpenAi API," OpenAi, [Online]. Available: https://platform.openai.com/docs/api-reference. [Accessed 10 08 2024].

[10] LifeEXE, 25 02 2023. [Online]. Available: https://github.com/life-exe/UnrealOpenAIPlugin. [Accessed 01 05 2024].

[11] "Audio2Face Overview," NVIDIA, 08 08 2024. [Online]. Available: https://docs.omniverse.nvidia.com/audio2face/latest/. [Accessed 10 08 2024].

Authors

Mgr. Lóránt Boráros
Faculty of Informatics, Pan-European University in Bratislava, Slovakia
xboraros@paneurouni.com
He is currently in the final year of master's degree program
And interested in the fields of virtual and augmented reality,
as well as computer vision.

RNDr. Ján Lacko, PhD.
Faculty of Informatics, Pan-European University in Bratislava, Slovakia
jan.lacko@paneurouni.com
His research interests include digitization of objects from the field
of cultural heritage, healthcare, industry, urban planning and their display
by various techniques, including virtual and augmented reality.

List of Reviewers

Issue 1/2024, in alphabetic order

doc. Ing. Petr Drahoš, PhD. Slovak University of Technology in Bratislava, Slovakia
doc. Ing. Oto Haffner, PhD. Slovak University of Technology in Bratislava, Slovakia
doc. Ing. Erik Kučera, PhD. Slovak University of Technology in Bratislava, Slovakia
prof. Danica Rosinová, PhD. Slovak University of Technology in Bratislava, Slovakia

ISSN: 2453-7497 (online)
ISSN 1338-6468