



# Биомедицинска електроника

Бранислав Геразов

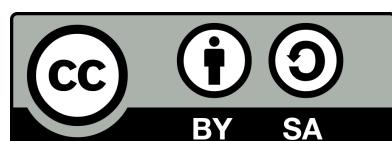
Факултет за електротехника и информациски технологии  
Универзитет Св. Кирил и Методиј во Скопје, Македонија

Биомедицинска електроника ~ Предавања и вежби v0.90

© Copyright by Branislav Gerazov, 2016 – 2023

Скрипта од предавањата и вежбите по предметот Биомедицинска електроника  
Институт за електроника  
Факултет за електротехника и информациски технологии  
Универзитет Св. Кирил и Методиј во Скопје, Македонија

This work is licensed under a Creative Commons Attribution-  
ShareAlike 4.0 International. [https://creativecommons.org/  
licenses/by-sa/4.0/](https://creativecommons.org/licenses/by-sa/4.0/)



Сликата на насловната страна е логото на [Open Hand Project](#), денес [Open Bionics](#) кои нудат слободни инструкции и шеми за самоградба на протеза за рака која може да се контролира со ЕМГ [https://www.instructables.com/  
Ada-Robotic-Hand-Open-Bionics/](https://www.instructables.com/Ada-Robotic-Hand-Open-Bionics/)

# Содржина

<b>0 Вовед</b>	<b>5</b>
0.1 Историја на биомедицинското инженерство . . . . .	5
<b>1 Основи на човековата анатомија и физиологија</b>	<b>10</b>
1.1 Структура на клетка . . . . .	10
1.2 Деполаризација на клетка . . . . .	14
1.3 Нервни импулси . . . . .	18
1.4 Градба и работа на мускулите . . . . .	22
1.5 Градба и работа на срцето . . . . .	27
1.6 Градба и работа на мозокот . . . . .	29
<b>2 Снимање на биомедицинските сигнали</b>	<b>34</b>
2.1 Фотоплетизмографија . . . . .	34
2.2 Електромиографија . . . . .	38
2.3 Електрокардиограм . . . . .	41
2.4 Електроенцефалограм . . . . .	45
<b>3 Процесирање на биомедицинските сигнали</b>	<b>51</b>
3.1 Основи на Фуриеовата анализа . . . . .	51
3.2 Фуриеова трансформација на временски отсекоци ФТВО . . . . .	58
3.3 Вовед во филтри . . . . .	60
3.4 Основи на дигиталните филтри . . . . .	60
3.5 Процесирање на ЕМГ сигнали . . . . .	66
3.6 Процесирање на ЕКГ сигнали . . . . .	75
3.7 Процесирање на ЕЕГ сигнали . . . . .	81
<b>4 Машинското учење со биомедицински сигнали</b>	<b>88</b>
4.1 Основни парадигми во машинското учење . . . . .	88
4.2 Вообичаени чекори во примената на машинското учење . . . . .	90
4.3 Тренирање на моделите за машинско учење . . . . .	90
4.4 Невронски мрежи . . . . .	92
4.5 Препознавање на човечка активност со акселерометри . . . . .	99
<b>Додаток А Слободен и отворен софтвер за инженерска и научна работа</b>	<b>105</b>
A.1 Слободен софтвер . . . . .	105
A.2 Четири слободи . . . . .	105
A.3 Предности на слободниот софтвер . . . . .	106
A.4 Одржливост . . . . .	107
A.5 Слободен софтвер за инженерска и научна работа . . . . .	108
<b>Додаток Б Основи на користење на ГНУ/Линукс</b>	<b>109</b>
B.1 Инсталирање на ГНУ/Линукс . . . . .	109
B.2 Основи поставки во ГНУ/Линукс . . . . .	112

Б.3 Основни наредби за работа во команѓна линија . . . . .	112
<b>Додаток В Пајтон за процесирање на биомедицинските сигнали</b>	<b>119</b>
В.1 Зошто Пајтон? . . . . .	119
В.2 Пајтон интерпретер . . . . .	121
В.3 Виртуелни средини за Пајтон . . . . .	121
В.4 Развојни средини за Пајтон . . . . .	124
<b>Додаток Г Основи на програмирање во Пајтон</b>	<b>127</b>
Г.1 Основи на Пајтон . . . . .	127
Г.2 Контрола на текот на програмата . . . . .	131
Г.3 Креирање на листи во еден ред . . . . .	132
Г.4 Функции . . . . .	132
Г.5 NumPy, SciPy и Matplotlib . . . . .	134
<b>Додаток Д Правење кориснички интерфејс со Пајтон</b>	<b>137</b>
Д.1 Платформи за креирање на ГУИ во Пајтон . . . . .	137

# Поглавје 0

## Вовед

Биомедицинската електроника е дел од областа биомедицинско инженерство која се занимава со примената на инженерските принципи и концепти во дизајнот и реализацијата на електронски системи во медицината и биологијата во здравствени цели. Главната цел е да се употребат познавањата од електротехниката и информациските технологии, како и инженерските вештини за решавање на проблеми, за унапредување на здравствтото, вклучувајќи ја дијагнозата, набљудувањето и терапијата (Enderle et al., 2005). Биомедицинското инженерство е изразено **мултидисциплинарна област**. Инженерите кои работат во оваа област мора да имаат еднакво цврсти познавања од биологијата и медицината и обратно – лекарите и биолозите во оваа област мора да се и добри инженери. Биомедицинското инженерство е и изразено **хумана област** поради тоа што резултатите од работата на ова поле имаат директен удел во подобрувањето на здравјето, а со тоа и животите на луѓето. Така, едни од најизразените достигнувања во биомедицинското инженерство се развојот на биокомпабилни протези за лицата со хендикеп, најразлични уреди за набљудување на човековото тело како што се ултразвукот, ЕКГ (електрокардиограм), ЕЕГ (електроенцефалограм) и МР (магнетната резонанса), пејсмејкери, машини за дијализа, вештачки органи, како вештачкото срце, како и цела низа на медицински уреди за дијагностицирање и терапија. Според статистичките податоци на Националната академија за инженерство, во светот денес работат околу 32.000 биомедицински инженери.

### 0.1 Историја на биомедицинското инженерство

Биомедицинското инженерство постоело со векови, дури и милениуми. Првата пронајдена **протеза** е вештачкиот палец на ногата направен од дрво и кожа најден на мушија од средовечна жена близу античкиот град Теба во Египет која датира од периодот меѓу 1070 – 670 г. п.н.е, Сл. 1. Уште еден важен показател за историјата на биоинженерството е бронзената нога од Капуа<sup>1</sup> најдена во Капуа, Италија, која потекнува од 300 г. п.н.е, чија реплика е прикажана на Сл. 2.

Во модерното време биомедицинското инженерство доживува забрзан развој. Во 1816 г. скромноста го натерала францускиот лекар Рене Лаенек<sup>2</sup> да употреби весник завиткан во фишек да ѝ го слушне срцето на една дама што подоцна го навело да го создаде првиот **стетосткоп**, Сл. 3.

Во 1848 германскиот лекар и физичар Херман фон Хелмхолц<sup>3</sup> првпат ги употребил инженерските принципи на физиолошки проблем, мерејќи ја електричната отпорност на мускулите и нервните ткива. Во 1895 германскиот физичар Вилхелм Рентген<sup>4</sup> случајно открил дека катодна цевка можела да направи хартија покриена со слој од бариум платиноцијанид да свети, дури и кога тие се поставени во различни простории. Тој заклучил дека цевката мора да еmitува

<sup>1</sup>Wikipedia: Capua Leg [https://en.wikipedia.org/wiki/Capua\\_Leg](https://en.wikipedia.org/wiki/Capua_Leg)

<sup>2</sup>Wikipedia: René Laennec [https://en.wikipedia.org/wiki/Ren%C3%A9\\_Laennec](https://en.wikipedia.org/wiki/Ren%C3%A9_Laennec)

<sup>3</sup>Wikipedia: Hermann von Helmholtz [https://en.wikipedia.org/wiki/Hermann\\_von\\_Helmholtz](https://en.wikipedia.org/wiki/Hermann_von_Helmholtz)

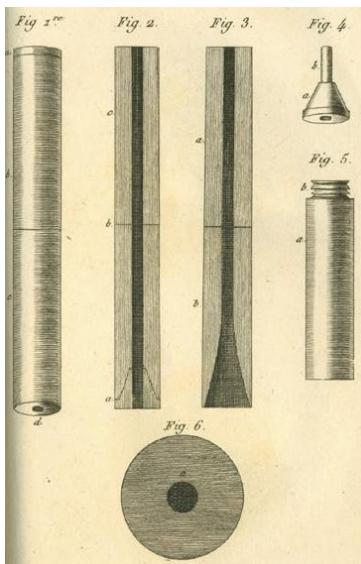
<sup>4</sup>Wikipedia: Wilhelm Röntgen [https://en.wikipedia.org/wiki/Wilhelm\\_R%C3%86ntgen](https://en.wikipedia.org/wiki/Wilhelm_R%C3%86ntgen)



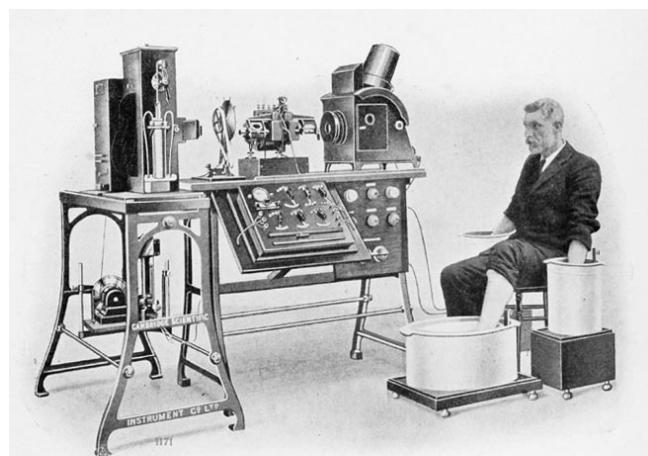
**Сл. 1:** Првата протеза – вештачки палец најден на мумија од стариот Египет.<sup>5</sup>



**Сл. 2:** Копија од ногата од Капуа од 300 г. п.н.е. пронајдена во Италија.<sup>6</sup>



**Сл. 3:** Првиот нацрт на стетоскоп од 1819 г.<sup>7</sup>



PHOTOGRAPH OF A COMPLETE ELECTROCARDIOGRAPH, SHOWING THE MANNER IN WHICH THE ELECTRODES ARE ATTACHED TO THE PATIENT, IN THIS CASE THE HANDS AND ONE FOOT BEING IMMESED IN JARS OF SALT SOLUTION

**Сл. 4:** Еден од најраните ЕКГ апарати направен во 1911 г. компанијата за научни инструменти од Кембриџ – *Cambridge Scientific Instrument Company*.<sup>8</sup>

некој вид на зраци кои ги нарекол „X“ зраци. По ова откритие, научната заедница започнува во голема мера да ги истражува пробивачките, но и уништувачките, својства на X-зраците. Овие истражувања резултираат со развојот на нивната примена за неинвазивно набљудување на човековото тело, речиси во целост елиминирајќи ја потребата од „истражувачка хирургија“. Брзиот напредок бил овозможен и од одлуката на Рентген да не ги патентира своите пронаоѓања. За нив, тој ја добива првата Нобелова награда по физика во 1901 г.

Иако Александар Муирхед за првпат во 1872 поврзал жици на раката на еден пациент за да му

<sup>5</sup>A prosthetic toe in the Cairo Museum by Jacky Finch <http://www.livescience.com/4555-world-prosthetic-egyptian-mummy-fake-toe.html>

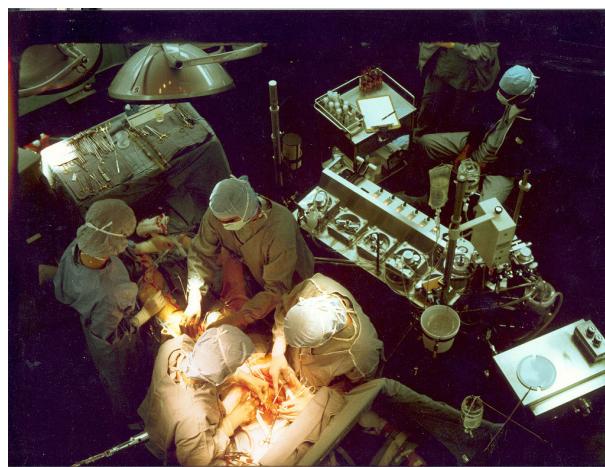
<sup>6</sup>By [http://wellcomeimages.org/indexplus/obf\\_images/9a/e9/fd5080571b64c69ab0dc7e574b82.jpg](http://wellcomeimages.org/indexplus/obf_images/9a/e9/fd5080571b64c69ab0dc7e574b82.jpg) Gallery: <http://wellcomeimages.org/indexplus/image/M0012307.html>, CC BY 4.0, <https://commons.wikimedia.org/w/index.php?curid=36352216>

<sup>7</sup>By René-Théophile-Hyacinthe Laennec - Cropped from Image:Rene-Theophile-Hyacinthe Laennec (1781-1826) Drawings stethoscope and lungs.jpg, Public Domain, <https://commons.wikimedia.org/w/index.php?curid=5443542>

<sup>8</sup>By [http://en.wikipedia.org/wiki/Image:Willem\\_Einthoven\\_ECG.jpg](http://en.wikipedia.org/wiki/Image:Willem_Einthoven_ECG.jpg), Public Domain, <https://commons.wikimedia.org/w/index.php?curid=1153294>



**Сл. 5:** Оддел од респиратори за луѓе со детска парализа во 1953 г.<sup>11</sup>



**Сл. 6:** Употреба на машина за кардиопулмонарен бајпас во коронарна хирургија.<sup>12</sup>

ги сними отчукувањата на срцето, првиот практично употреблив **електрокардиограм** (ЕКГ) го измислува во **1903** холандскиот лекар **Вилијам Ајнховен**<sup>9</sup> за што ја добива Нобеловата награда за медицина во **1924** г. Еден таков уред е прикажан на Сл. 4. Неговиот изум – жичаниот галванометар, му овозможил за првпат да ги детектира и сними многу малите електрични струи произведени од човековото срце. Неговата машина давала точност која и денес е тешко достиглива од модерните ЕКГ апарати (**Barold, 2003**).

Биомедицинското инженерство за првпат се појавило како област во високото образование во **1921** г. при **Институтот за физика и медицина Освалт** во Франкфурт, Германија, а во **1969** г. е направена првата комбинирана **докторска програма** за доктор по медицински науки и филозофија (MD/PhD), при одделот за Биомедицинско инженерство во Универзитетот Кејс Вестерн Резерв во Кливленд, Охajo, САД.

Германскиот психијатар **Ханс Бергер**<sup>10</sup> го измислува **електроенцефалограмот** (ЕЕГ) во **1924**. Тој е првиот што ги изучува различните бранови и ритми присутни кај мозокот кај луѓе кои страдаат од епилепсија, како што се  $\alpha$ -брановите со опсег на фреквенции  $7,812 - 13,28$  Hz. Првин тој вметнувал две електроди, една од предната страна и една од задната, под скалпот, за потоа да ги прицврсти на кожата со помош на гумени ластици. За своите мерења тој го користел жичаниот галванометар на Ајнховен, а подоцна и галванометарот за снимање на Сименс, кој му овозможил снимање на напони од редот на  $100\text{ }\mu\text{V}$ .

Во **1928** г. бил измислен **Дринкер респираторот**, којшто претставува еден тип на вентилатор со негативен притисок наречен и „челичен бел дроб“. Овој уред служи за овозможување на процесот на дишење кога човекот не може тоа да го направи тоа со сопствена сила. Првата примена на Дринкер респираторот била кај 8 годишно девојче заболено од детска парализа во Клиниката за деца во Бостон, САД. На Сл. 5 е прикажан еден цели оддел со вакви машини за луѓе со детска парализа во болницата Ранчо Лос Амигос во Калифорнија, САД во **1953** г.

Во **1950** г. бил измислен **електронскиот микроскоп**<sup>13</sup> којшто можел да постигне многу големи резолуции поради тоа што електроните имаат 100.000 пати пократка бранова должина од видливата светлина. Во периодот **1950 – 1960** бил постигнат голем напредок во **нуклеарната медицина**. Во **1953** била направена машината за **кардиопулмонарен бајпас**<sup>14</sup> која првично

<sup>9</sup>Wikipedia: Willem Einthoven [https://en.wikipedia.org/wiki/Willem\\_Einthoven](https://en.wikipedia.org/wiki/Willem_Einthoven)

<sup>10</sup>Wikipedia: Hans Berger [https://en.wikipedia.org/wiki/Hans\\_Berger](https://en.wikipedia.org/wiki/Hans_Berger)

<sup>11</sup>By Food and Drug Administration - <http://www.fda.gov/cber/summaries/cent092302pp.htm>, Public Domain, <https://commons.wikimedia.org/w/index.php?curid=3227578>

<sup>12</sup>By Jerry Hecht - <http://fmp.cit.nih.gov/hi/> - Image ID: 657 C, Public Domain, <https://commons.wikimedia.org/w/index.php?curid=780261>

<sup>13</sup>Wikipedia: Electron microscope [https://en.wikipedia.org/wiki/Electron\\_microscope](https://en.wikipedia.org/wiki/Electron_microscope)

<sup>14</sup>Wikipedia: Cardiopulmonary bypass [https://en.wikipedia.org/wiki/Cardiopulmonary\\_bypass](https://en.wikipedia.org/wiki/Cardiopulmonary_bypass)

може да ја превземе работата на срцето и белите дробови пред сè за време на операции на срцето, и може да се искористи за воспоставување на контролирана хипотермија.

Во 1970 г. биле воведени техниките на **компјутерска томографија (КТ)**<sup>15</sup> во која компјутерски се обработуваат повеќе Рентген слики за добивање на напречни пресеци на скенираниот дел од телото, како и **магнетна резонанса (MRI)**<sup>16</sup> која преку употреба на силни магнетни полиньа може да ја опфати дистрибуцијата на молекулите на водата во телото, па со тоа да се набљудува анатомијата и физиолошките процеси на меките ткива, Сл. 7. Одредени техники во магнетната резонанса дури овозможуваат и мапирање на поврзаноста на мозокот преку анализа на дифузијата на молекулите на водата долж нервните влакна, Сл. 8.

Техниката на **томографија со позитронска емисија (PET)**<sup>17</sup> била воведена во 1980. Со која може да се снимат физиолошките процеси во телото преку анализа на гама зрачењето кое се внесува во телото преку биолошки активни молекули.

Развојот на биомедицинското инженерство денес овозможува и надминување на нашите биолошки ограничувања и оштетувања. Првото **вештачко срце**<sup>18</sup> успешно вградено во човек било срцето Јарвик-7, чиј наследник е прикажан на Сл. 9, во 1982 на Универзитетот во Јута. Срцето било вградено во 61-годишен човек кој живеел со него 112 денови. Најдолготрајна успешна употреба на вештачко срце е 1700 денови, односно 4,6 години. Најчесто тоа се користи како привремено решение пред да е можна трансплантирања на истинско срце.

Исто така, вештачките протези за раце и нозе денес овозможуваат замена на одреден степен од функционалноста на природните екстремитети. Така, во лабораторијата за медици во групата за биомехатроника на МИТ се развиени протези за нозе кои овозможуваат непречено одење, трчање, дури и танцување, Сл. 10.<sup>19</sup> Интерфејсите мозок-компјутер пак, овозможуваат мисловна контрола на електронските уреди. За првпат таа е реализирана на ФЕИТ, поранешен ЕТФ, од страна на проф. Божиновски во 1988 кога со алфа ЕЕГ брановите е реализирана контрола на мобилен робот (**Bozinovski et al., 1988**). Овие системи исто така овозможуваат комуникација на лутето со парализа со околината преку текст, а во поново време и преку говор.<sup>20</sup>

<sup>15</sup>Wikipedia: Computer Tomography - CT scan [https://en.wikipedia.org/wiki/CT\\_scan](https://en.wikipedia.org/wiki/CT_scan)

<sup>16</sup>Wikipedia: Magnetic resonance imaging [https://en.wikipedia.org/wiki/Magnetic\\_resonance\\_imaging](https://en.wikipedia.org/wiki/Magnetic_resonance_imaging)

<sup>17</sup>Wikipedia: Positron emission tomography [https://en.wikipedia.org/wiki/Positron\\_emission\\_tomography](https://en.wikipedia.org/wiki/Positron_emission_tomography)

<sup>18</sup>Wikipedia: Artificial heart [https://en.wikipedia.org/wiki/Artificial\\_heart](https://en.wikipedia.org/wiki/Artificial_heart)

<sup>19</sup>TED Hugh Herr: New bionics let us run, climb and dance <https://www.youtube.com/watch?v=CDsNZJTWW0w>

<sup>20</sup>Synthetic speech generated from brain recordings <https://www.youtube.com/watch?v=3pv0vT82Cys>

<sup>19</sup>By Jan Ainali - Own work, CC BY 3.0, <https://commons.wikimedia.org/w/index.php?curid=3546051>

<sup>22</sup>By Xavier Gigandet et. al. - Gigandet X, Hagmann P, Kurant M, Cammoun L, Meuli R, et al. (2008) Estimating the Confidence Level of White Matter Connections Obtained with MRI Tractography. PLoS ONE 3(12): e4006. doi:10.1371/journal.pone.0004006, CC BY 2.5, <https://commons.wikimedia.org/w/index.php?curid=8134159>

<sup>23</sup>By SynCardia Systems, Inc. - Own work, CC BY-SA 3.0, <https://commons.wikimedia.org/w/index.php?curid=22097831>



Сл. 7: Уред за магнетна резонанса.<sup>21</sup>



Сл. 8: Мапирање на поврзаноста на мозокот со МР.<sup>22</sup>



Сл. 9: Вештачко срце SynCardia.<sup>23</sup>



Сл. 10: Вештачка протеза за нога со напредни можности.

# Поглавје 1

## Основи на човековата анатомија и физиологија

Разбирањето на човековата анатомија и физиолошките процеси кои се одвиваат во човековиот организам е од големо значење за дизајнирањето на биомедицинските електронски системи. Од анатомски аспект потребно е познавање на структурата на различните ткива, на пр. нивните електрични карактеристики, за да може да се дизајнираат системи за нивно набљудување, моделирање, симулација и стимулација.

Од физиолошки аспект, од значење се процесите кои генерираат промени во електричното поле во телото, а со тоа можат да бидат детектирани преку електроди. Со овие процеси се занимава областа **електрофизиологија**, а нивното познавање е важно за дизајнот на системи како што се електромиограмот (ЕМГ), електрокардиограмот (ЕКГ) и електроенцефалограмот (ЕЕГ).

### 1.1 Структура на клетка

**Клетката** е основната структурна и функционална единица на сите живи организми.<sup>1</sup> Таа е најмал жив биолошки систем.<sup>2</sup> Клетките се способни за размножување, бидејќи го содржат целокупниот наследен материјал.

Типичната градба на една клетка е дадена на Сл. 1.1.<sup>3</sup> Иако основната градба на сите клетки во организмите е иста, сепак, во зависност од местото и функцијата, клетките можат да се разликуваат по својата големина и облик. По големина клетките кај повеќеклеточните организми се движат помеѓу 10 и 100  $\mu\text{m}$ , но можат да бидат и многу поголеми како што се мускулните клетки и невроните. Така, невроните во ‘рбетниот столб можат да достигнат должина од 1 m. По својот облик клетките исто така можат да бидат топчети, плочети, коцкести, вртеновидни или пак разгранети. На пример: нервните клетки со своите долги продолжетоци многу се разликуваат од мускулните клетки или од прозирните клетки во очната леќа. Според местоположбата и функцијата во организмот, разликуваме епителна, нервна, мускулна, полова клетка итн.

Онадвор клетката е обвиткана со **клеточна мембрана**.<sup>4</sup> Ваквата мембрана е составена од два слоја на масти во кои се вметнати различни белковини и мало количество на шеќери. Клеточната мембрана е многу важна, бидејќи таа одредува што може да влезе, а што да излезе од клетката.

<sup>1</sup> Текстот во делот 1.1 користи исчеоци од Wikipedia: Клетка <https://mk.wikipedia.org/wiki/Клетка>

<sup>2</sup> Интересно видео за тоа колку ние сме клетките кои ги содржиме и обратно може да видите тука: Kurzgesagt – In a Nutshell: What Are You? <https://www.youtube.com/watch?v=JQVmKDUkZT4> како и за тоа каде е границата помеѓу живата и нејзина материја тука: In a Nutshell: What Is Life? Is Death Real? <https://www.youtube.com/watch?v=Q0Caac08wus>

<sup>3</sup> Погуично видео за структурата на клетката може да видите овде: Biology: Cell Structure I Nucleus Medical Media <https://www.youtube.com/watch?v=v=URUJD5NEXC8>

<sup>4</sup> Wikipedia: Клеточна мембрана [https://mk.wikipedia.org/wiki/Клеточна\\_мембрана](https://mk.wikipedia.org/wiki/Клеточна_мембрана)

Клеточната мембра на е многу тенка и е составена од фосфолипиди кои имаат еден хидрофобичен и еден хидрофиличен крај. Фосфолипидите се распоредени во бимолекуларен слој описан според течно-мозаичниот модел, а разните протеини се вметнати на различни места во овој бимолекуларен слој, види понатаму Сл. 1.4.

**Цитоплазмата** е полутечна маса со зрнеста структура, изградена од вода и минерални соли, неоргански и органски материји: шеќери, масти и белковини. Гликогенот, на пример, е сложен шеќер кој се складира како резервен шеќер во цитоплазмата на клетката. Таа содржи цитоскелет кој ѝ го одржува обликот и овозможува транспорт на материји низ неа. Зрнестата структура на цитоплазмата ја сочинуваат посебни делчиња наречени органели во кои се вршат животните процеси. Некои од поважните органели се:

- **јадрото** е органела која ја има во секоја клетка. Јадрото е обвите со јадрена обвивка со пори. Во јадрото се сместени хромозомите, а во нив гените кои во дезоксирибонуклеинската киселина (ДНК) ја носат генетската информација. Во гените се сместени и контролните механизми за управување со различните животни процеси кои се одвиваат во клетката,
- **рибозомите** се органелите кои се одговорни за произведување на протеините врз база на генетската информација<sup>5</sup>,
- **митохондриите** се клеточни органели во кои се врши аеробното разложување на материите и се создава енергија складирана во вид на аденоzin трифосфат (АТФ)<sup>6</sup>. Митохондриите содржат свои гени и се способни сами да синтетизираат протеини и да се размножуваат,
- **вакуоли** претставуваат мали торбички што служат за складирање и транспорт на резервни материји како вода, масти кај машините клетки и други органски материји. Вакуолите, воедно, служат и за исфрлање на непотребните материји од клетката<sup>7</sup>,
- **лизозоми** се органели кои ги собираат и отстрануваат отпадните материјали од клетката.

Сите клетки се изградени од хемиски елементи кои се среќаваат и во неживата природа. Од над 100 познати хемиски елементи, околу 60 влегуваат во составот на клетките. При тоа, овие елементи количински не се застапени еднакво, па се разликуваат:

- **Макроелементи** – јаглерод С, кислород О, водород Н и азот N,
- **Микроелементи** – фосфор P, сулфур S, натриум Na, калциум Ca, магнезиум Mg, калиум K и хлор Cl,
- **Елементи во трагови** – јод I, флуор F, железо Fe, цинк Zn, манган Mn и др.

Релативната застапеност на некои од нив во клетките од човековиот организам е дадена во Табелата 1.1. Во составот на клетките, хемиските елементи се јавуваат во форма на соединенија - органски и неоргански. Биолошкото значење на овие хемиски елементи е големо, независно од

---

<sup>5</sup>Овој процес е претставен преку анимација во видеото: From DNA to protein - 3D <https://www.youtube.com/watch?v=gG7uCskU0rA>

Пореалистична претстава е дадена во видеото DNA animations by wehi.tv for Science-Art exhibition <https://www.youtube.com/watch?v=7Hk9jct2oZY>

Реалистични анимации од био-механичките процеси кои се одвиваат во клетките се дадени во видеото The Molecular Basis of Life <https://www.youtube.com/watch?v=fpHaxzroYxg>

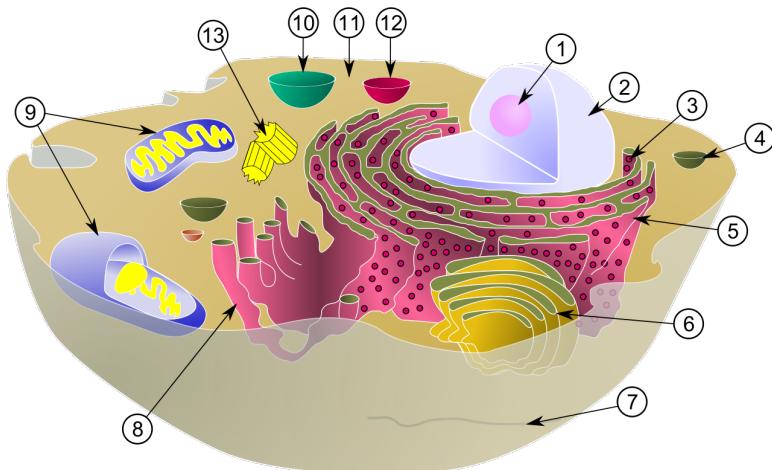
Упите едно реалистично видео за основната градба и физиологија на клетките е Inner Life Of A Cell - Inner Life of the Cell Full Version <https://www.youtube.com/watch?v=yKW4F0Nu-UY>

За јазикот со кој зборуваат клетките преку во кој главната улога ја играат протеините погледнете го видеото Kurzgesagt: The Most Complex Language in the World <https://www.youtube.com/watch?v=TYPFenJQciw>

<sup>6</sup>На англиски Adenosine triphosphate (ATP).

<sup>7</sup>Едно опширно предавање за моториката зад транспортот на материји во клетката се дадени во видеото Ron Vale (UCSF, HHMI) 1: Molecular Motor Proteins <https://www.youtube.com/watch?v=9RUHJhskW00>

<sup>8</sup>Од MesserWoland и Szczepan1990 — сопствено дело (создадена со Inkscape), CC BY-SA 3.0, <https://commons.wikimedia.org/w/index.php?curid=1279365>



**Сл. 1.1:** Градба на клетката: 1) јадренце, 2) клеточно јадро, 3) рибозом, 4) везикула, 5) Груб ендоплазматичен ретикулум (ЕР), 6) Голциев систем, 7) цитоскелет, 8) Мазен ЕР, 9) митохондрија, 10) вакуола, 11) лизозом, 12) лизозом, 13) центриоли.<sup>8</sup>

**Табела 1.1:** Застапеност на елементите во клетките на човекот.

Елемент	%	Елемент	%	Елемент	%
Водород	9,5	Сулфур	0,3	Железо	траги
Јаглерод	18,5	Калциум	1,5	Натриум	0,2
Азот	3,3	Магнезиум	0,1	Калиум	0,4
Кислород	65,0	Манган	траги	Јод	траги
Фосфор	1,0	Цинк	траги	Хлор	0,2

нивната релативна застапеност. Недостигот на кој и да е елемент предизвикува нарушување во нормалното функционирање на клетките и на организмот.

Од неорганските соединенија најзастапена е водата, која е неопходна за животот. Таа во просек, во клетките е застапена со околу 80%, но некои клетки содржат повеќе, на пр. ембрионалните клетки содржат до 95%. Водата за клетката е значајна од неколку причини:

- метаболизмот во клетките е можен само во водна средина,
- таа е дисперзно средство за клеточните колоиди, т.е. мешавините од соединенија со една, две или повеќе фазни состојби,
- учествува во транспортот на материите,
- има голем топлински капацитет.

Други неоргански соединенија во клетката се минералните соли. Тие најчесто се јавуваат растворени, дисоциирани на јони, и тоа:

- катјони – Натриум  $\text{Na}^+$ , Калиум  $\text{K}^+$ , Калциум  $\text{Ca}^{2+}$ , и Магнезиум  $\text{Mg}^{2+}$ ,
- анјони – хлорид  $\text{Cl}^-$ , сулфат  $\text{SO}_4^{2-}$ ,  $\text{HCO}_3^-$ , и фосфат  $\text{PO}_4^{3-}$ .

Јоните имаат важни функции во клетката. Тие учествуваат во одржувањето на осмотската хомеостаза, или рамнотежа, меѓу клетката и нејзината околина, одговорни се за ацидобазичната хомеостаза на клетката, а исто така ја регулираат и активноста на некои ензими. Јоните се и во основата на електрофизиологијата на клетките – тие се носители на електричните струи.

Во клетките се јавуваат четири типа органски соединенија: јаглехидрати, липиди, протеини и нуклеински киселини.

- **Јаглехидратите**, или шеќерите, се органски соединенија изградени од јаглерод, водород и кислород. Од моносахаридите, најпознати соединенија во клетката се пентозите рибоза и деоксирибоза и хексозите глукоза, која клетките ја користат најчесто за добивање енергија, и фруктоза. Од дисахаридите значајна е сахарозата, или трсчен шеќер, додека од полисахаридите најзначајни се гликогенот во животинските клетки и скробот во растителните клетки, како и целулозата кај растенијата. Општо земено, јаглехидратите се извор на енергија на клетките. Со нивното разложување, како краен производ се добива јаглероден диоксид и вода и се ослободува енергија.<sup>9</sup> Освен тоа, тие влегуваат во составот на мембрани, клеточниот сид и други клеточни структури.
- **Липидите**, или масните материји, се група органски соединенија за кои е карактеристично што се хидрофобни, односно нерастворливи во вода, а се раствораат во органски растворувачи, на пр. етер, ацетон, бензин и др. Липидите се извор на енергија и затоа во клетките најчесто се користат како резерви за енергија. По потреба, особено при подолгото гладување, тие се разложуваат до  $\text{CO}_2$  и  $\text{H}_2\text{O}$ , ослободувајќи притоа големо количество енергија. Освен енергетска, овие соединенија имаат и структурна и заштитна функција, бидејќи учествуваат во изградбата на клеточната мембра и другите мембрански системи во клетката. Разликуваме прости липиди кои во својот состав содржат виши масни киселини кои се делат на незаситени и заситени<sup>10</sup>, и сложени липиди кои содржат и нелипидно соединение. Еден пример за сложени липиди се фосфолипидите кои содржат фосфорната киселина која им дава хидрофиличен крај. Попознати липиди во составот на клетката се неутралните масти и липоидите:
- **Протеините**, или белковините, имаат огромно биолошко значење, како за клетката, така и за организмот. По водата, количински се најзастапени во клетката. Тие не можат лесно да бидат искористени за добивање енергија но од нив зависат основните особини на клетката, метаболизам и градба:
  - учество во градбата на клеточните структури: мембрани, органели и др.,
  - учество во различни движења на клетката – во мускулните клетки, тоа се протеините актин и миозин,
  - антителата, кои се протеини, имаат заштитна функција,
  - протеините во вид на ензими, ги забрзуваат биохемиските процеси во клетката итн.
- **Нуклеинските киселини** се соединенија со големо биолошко значење, бидејќи го зачувуваат наследниот материјал и имаат централна улога во синтезата на протеините, а преку нив ги определуваат главните особини и карактеристики на клетката. Има два типа нуклеински киселини: **ДНК** (деоксирибонуклеинска киселина) и **РНК** (рибонуклеинска киселина).

<sup>9</sup>Фруктозата, која заедно со глукозата влегува во составот на сахарозата, не може директно да се метаболизира во човековиот организам, туку мора првин да биде претворена во масти во црниот дроб кои потоа можат да бидат искористени за енергија или наталожени. Robert H. Lustig – Sugar: The Bitter Truth <https://www.youtube.com/watch?v=dBnniuA6-oM>

<sup>10</sup>Заситените масти кои претежно се од животинско потекло се посебно штетни за човековиот кардиоваскуларен систем. Michael Greger M.D. – Low-Carb Diets & Coronary Blood Flow <https://nutritionfacts.org/video/low-carb-diets-and-coronary-blood-flow/>.

## 1.2 Деполаризација на клетка

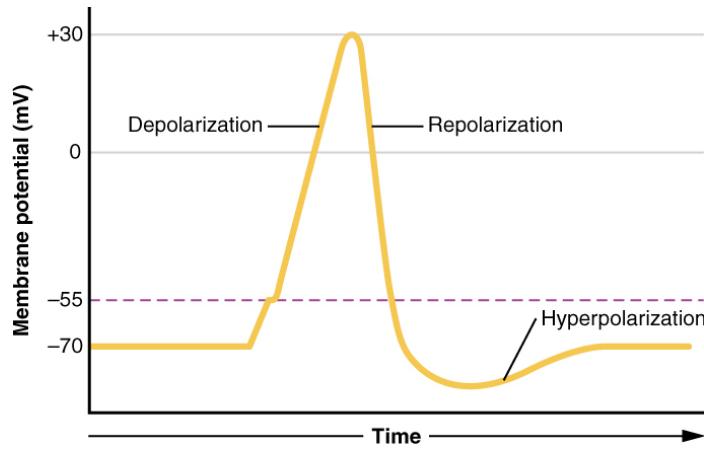
Во основата зад електричните промени кои настапуваат во организмот, којшто може да бидат детектирани преку соодветни сензори, стои процесот на деполаризација на една клетка. **Деполаризацијата<sup>11</sup>** во биолошки смисол, подразбира ненадејна и драматична електрична промена во клетката, Сл. 1.2. Во својата рамнотежна стационарна состојба, клетките ја одржуваат својата внатрешност на негативен електричен потенцијал, споредено со нивното опкружување, преку одржувањето на различна концентрација на позитивни и негативни јони на двете страни од мемраната.

### Биоелектричен потенцијал на мирување

Во ошт случај, без влијание на надворешното електрично поле, поголема концентрација на  $\text{Na}^+$  јони се натрупва на надворешната страна на мемраната, додека помала концентрација на  $\text{K}^+$  јони и други негативни јони се натрупваат на внатрешната страна од мемраната. На тој начин во близина на мемраната доаѓа до формирање на двоен електричен слој. По своите електрични својства клетката и нејзината околина може да се прикажат како кондензатор. Цитоплазмата и течноста што ја опколува, кои имаат висока проводност, се двете електроди, а клеточната мембрана со својата мала проводност, ја има улогата на диелектрик кој ги разделува електродите.<sup>13</sup>

Разликата во потенцијалите помеѓу внатрешноста и надворешноста на клетката се нарекува **потенцијал на клеточната мембрана** или **биоелектричен потенцијал на мирување**. Сите електрични процеси во човековото тело се должат на промената на овие потенцијали (**Гершановски**). Големината на биоелектричниот потенцијал може да се мери со помош на специјална микро-електрода со која се пробива клеточната мембрана и таа се внесува во нејзината внатрешност. Втората електрода се поставува во течноста на меѓуклеточниот простор. Биоелектричните потенцијали се движат околу  $-70 \text{ mV}$ , но немаат секогаш иста големина.

Најважен механизам за создавање на биоелектричниот потенцијал е активното пумпање на натриумовите  $\text{Na}^+$  и калиумовите  $\text{K}^+$  јони низ клеточната мембрана, која е резултат на дејството на **натриум-калиумови пумпи** прикажани на Сл. 1.3.<sup>14</sup> Преку нив за секои два внесени  $\text{K}^+$  јони,



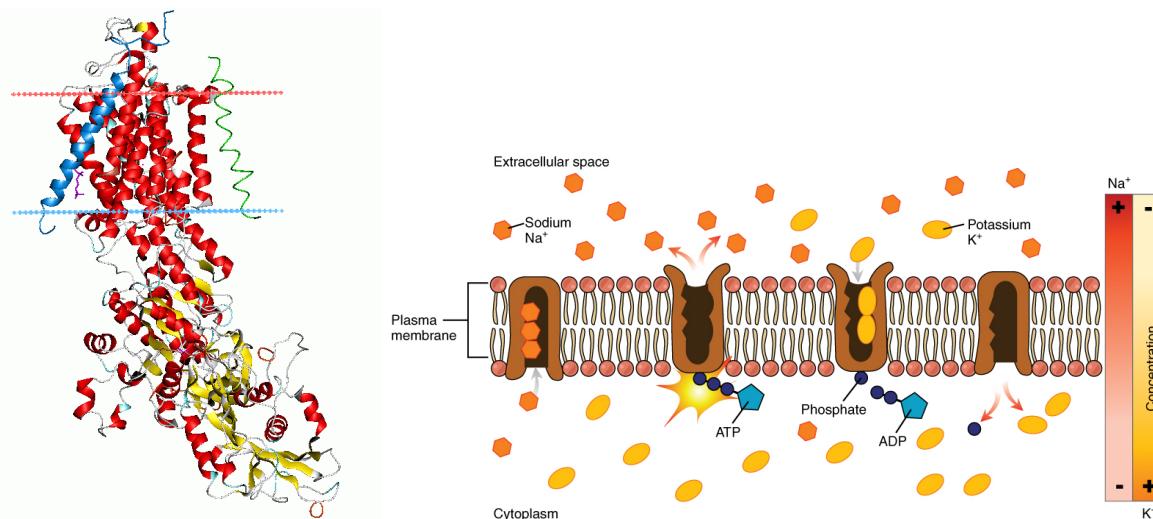
Сл. 1.2: Деполаризација на една клетка.<sup>12</sup>

<sup>11</sup>Wikipedia: Depolarization <https://en.wikipedia.org/wiki/Depolarization>

<sup>12</sup>By OpenStax College - Anatomy & Physiology, Connexions Web site. <http://cnx.org/content/col11496/1.6/>, Jun 19, 2013., CC BY 3.0, <https://commons.wikimedia.org/w/index.php?curid=30147928>

<sup>13</sup>Текстот во овој поднаслов користи исечоци од Wikipedia: Биоелектричен потенцијал [https://mk.wikipedia.org/wiki/Биоелектричен\\_потенцијал](https://mk.wikipedia.org/wiki/Биоелектричен_потенцијал)

<sup>14</sup>Натриум-калиумовите пумпи биле откриени од данскиот научник Јенс Кристиан во 1957, за што добива дел од Нобеловата награда по хемија во 1997.



Сл. 1.3: Изглед на една натриум-калиумова пумпа (лево) и принцип на работа (десно).<sup>16</sup>

од клетката се испуштуваат три Na<sup>+</sup> јони со употреба на еден молекул на АТФ.<sup>15</sup> На овој начин се намалува електричниот потенцијал на клетката и се создаваат услови за нејзина брза деполаризација. Така, екстрацелуларната течност содржи голема концентрација на Na<sup>+</sup> јони кои не можат да проникнат во клетката поради тоа што пропустливоста на мембраната на Na<sup>+</sup> јони е 100 пати помала од пропустливоста за K<sup>+</sup> јони.

И покрај високата концентрација на K<sup>+</sup> во внатрешноста на клетката, таа вообичаено содржи и негативно наелектризиирани честички кои ја воспоставуваат негативната поларизација.

### Процес на деполаризација

Откако клетката ќе го воспостави биоелектричниот потенцијал на мирување, таа може да биде побудена и да оствари деполаризација. При деполаризацијата, за време помало од 1 ms, надворешната страна на клеточната мембрана станува негативна во однос на внатрешната, Сл. 1.2. Промената на потенцијалот на мембраната при побудување на клетките се нарекува акциски потенцијал.

Побудувањето на клетките може да се оствари на повеќе начини и тоа под:

- механичко дејство,
- хемиско дејство,
- електрично дејство и друго.

За да настане деполаризација, дразбата треба да е доволно голема за да го доведе потенцијалот на клетката до праговниот напон од -55 mV. Ова доведува до отворање на **напонски регулирани јонски канали** на Na<sup>+</sup><sup>17</sup> предизвикано од електричната побуда, Сл. 1.4. Со тоа се зголемува пропустливоста на мембраната на јоните на натриумот Na<sup>+</sup> па тие во голем број навлегуваат во клетката поларизирајќи ја позитивно нејзината внатрешност.

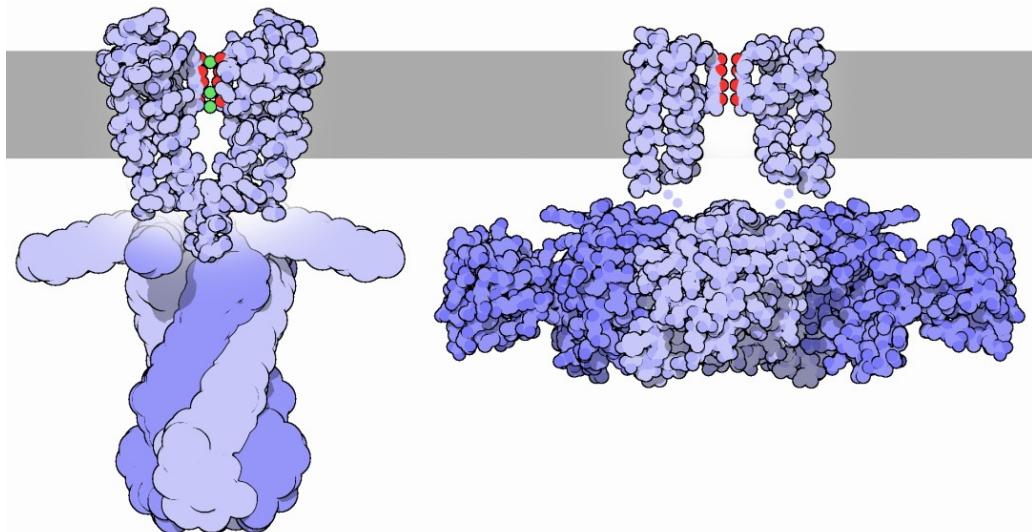
Wikipedia: Sodium–potassium pump [https://en.wikipedia.org/wiki/Sodium–potassium\\_pump](https://en.wikipedia.org/wiki/Sodium–potassium_pump)

<sup>15</sup>Овој процес, како и целиот процес на деполаризација кај неврните импулси се претставени како анимација во видеото Nerve Impulse Molecular Mechanism [3D Animation] <https://www.youtube.com/watch?v=fHRC8SLLcH0> а реалистичен приказ на работата на јонските пумпи може да се види во Neuronal Signaling and Sodium-Potassium Pump (from PDB-101) <https://www.youtube.com/watch?v=ZKE8qK9UCrU>

<sup>16</sup>By OpenStax - <https://cnx.org/contents/FPtK1zmh@8.25:fEI3C80t@10/Preface>, CC BY 4.0, <https://commons.wikimedia.org/w/index.php?curid=30131190>

By Andrei Lomize - Own work. Protein image from OPM database, CC BY-SA 3.0, <https://commons.wikimedia.org/w/index.php?curid=34170807>

<sup>17</sup>Wikipedia: Sodium channel [https://en.wikipedia.org/wiki/Sodium\\_channel](https://en.wikipedia.org/wiki/Sodium_channel)



**Сл. 1.4:** Поедноставен напонски регулиран калиумов канал во затворена (лево) и отворена состојба (десно).<sup>18</sup>

Деполаризацијата завршува кога внатрешноста на клетката ќе постигне позитивна поларизација. Ваквата прераспределба на електричните полнежи трае се додека трае и побудената состојба на клетките, после која повторно се воспоставува мембранныот потенцијал на мирување.

Брзата дифузија на натриумовите јони трае од 0,1 до 1 ms. Создадениот позитивен полнеж во внатрешната страна на мембраната го забавува натамошното навлегување на натриумовите јони. Тој исто така ги активира и каналите за K<sup>+</sup><sup>19</sup> со што калиумовите јони започнуваат да ја напуштаат клетката.<sup>20</sup>

Како последица од ова повторно се воспоставува биоелектричниот потенцијал на мирување. Овој процес се нарекува **реполаризација**. Низ целиот овој процес на деполаризација и реполаризација натриум-калиумовите пумпи постојано работат.<sup>21</sup>

Процесот на реполаризација предизвикува натфрлање, всушност потфрлање, бидејќи калиумовите јони продолжуваат да излегуваат од клетката снижувајќи го нејзиниот потенцијал под оној во мирување. Ова потфрлање се нарекува **хиперполаризација**. Хиперполаризацијата престанува со затворањето на сите напонски контролирани јонски пумпи, со што на натриум-калиумовите пумпи, кои работат целото време, им се дава простор да го воспостават потенцијалот на мирување.

Веднаш по хиперполаризацијата напонски-контролираните натриумови пумпи се во неактивна состојба во која не можат да бидат побудени, дури и при доаѓање на нова надворешна побуда, како на пример надворешен акциски потенцијал. Ова време се нарекува **време на непобудливост** и тоа игра клучна улога во еднонасочното простирање на акционите потенцијали.

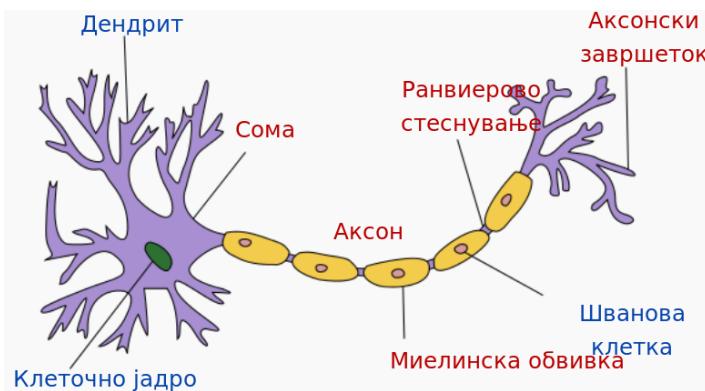
<sup>18</sup>By David Goodsell - "Molecule of the Month: Potassium Channels". RCSB Protein Data Bank. doi: 10.2210/rscb\_pdb/mom\_0032, CC BY 3.0,

<sup>19</sup>Wikipedia: Potassium channel [https://en.wikipedia.org/wiki/Potassium\\_channel](https://en.wikipedia.org/wiki/Potassium_channel)

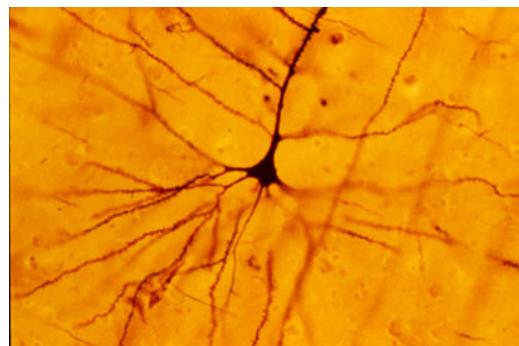
<sup>20</sup>Интересно е да се истакне дека натриумовите и калиумовите канали еволуирале селективни филтри кои пропуштаат исклучиво натриумови, односно калиумови јони иако тие имаат разлика во дијаметар помала од Å (ангстром) = 0,1 nm. Кај натриумовите канали, катјоните се движат кон сè поголемо стеснување на каналот кое на крајот изнесува 0,3 × 0,5 nm, што е премало за поголемите калиумови јони. Кај калиумовите канали, растојанието помеѓу контакт точките на селективниот филтер го охрабрува ослободувањето на калиумовите јони од водата и нивно врзување и поминување низ каналот. Натриумовите јони се помали и не можат да пополнат овој меѓупростор па остануваат врзани за молекулите на водата и не можат да поминат низ каналот.

<sup>21</sup>Уште една убава претстава на целиот процес на деполаризација и реполаризација е дадена во видеото Action Potential Animation [https://www.youtube.com/watch?v=-h\\_kWFm2faQ](https://www.youtube.com/watch?v=-h_kWFm2faQ)

Промената на биоелектричниот потенцијал за време на деполаризацијата го овозможува преносот на електрични импулси во клетките, како кај невроните, но и помеѓу нив. Со тоа деполаризацијата како начин на комуникација е од особено значење за работата на многу видови на клетки, но и на човековиот организам во целина.



Сл. 1.5: Градба на невроните.<sup>23</sup>



Сл. 1.6: Човечки неврон виден под микроскоп.<sup>24</sup>

### 1.3 Нервни импулси

#### Градба на неврон

Нервните клетки или невроните се основните градбени единки на нервниот систем.<sup>22</sup> За разлика од другите клетки во телото, невроните се карактеризираат со надразливост, осетливост и спроводливост. Овие карактеристики се во согласност со нивната специфична функција. Градбата на невроните е прикажана на Сл. 1.5 и содржи:

- тело на нервната клетка или невроцит,
- дендрити (влезови), и
- нервно влакно или аксон (излез).

На Сл. 1.6 е прикажан неокортикален пирамидален неврон обоеан по методот на Голци. Се забележуваат многубројните дендрити кои излегуваат, всушност водат кон невроцитот.

#### Побудување и движење на нервните импулси

Побудите кај невроните можат да бидат физички, електрични и хемиски, но тие освен што можат да извршат побудување на невронот, можат и да го инхибираат истиот. Така, инхибиторската побуда наместо да предизвика промени во невронот во насока на деполаризација, таа врши негова хиперполаризација. Оваа хиперполаризација предизвикува пад на биоелектричниот потенцијал на невронот под овој на мирување. Оваа додатна негативна поларизација мора да се надмине за да настане деполаризација, со што таа е отежната. Позитивните побуди пак, го зголемуваат напонот во невронот, што предизвикува невронот да може полесно да биде однесен во деполаризација во однос на неговата состојба во мирување.

Без разлика од поларитетот на побудите, тие секогаш патуваат од дендритите кон телото на невронот и конвергираат кон коренот на аксонот, Сл. 1.7. Тука се врши нивно сумирање, односно еден вид на биолошка интеграција, за да се одреди одсивот на невронот. Ако сумата на побудите достигне одреден напон наречен **напон на праг**, доаѓа до деполаризација којашто започнува да се простира од коренот на аксонот кон неговиот врв. Овој бран на деполаризација долж аксонот се нарекува **акциски потенцијал** или **нервен импулс**.

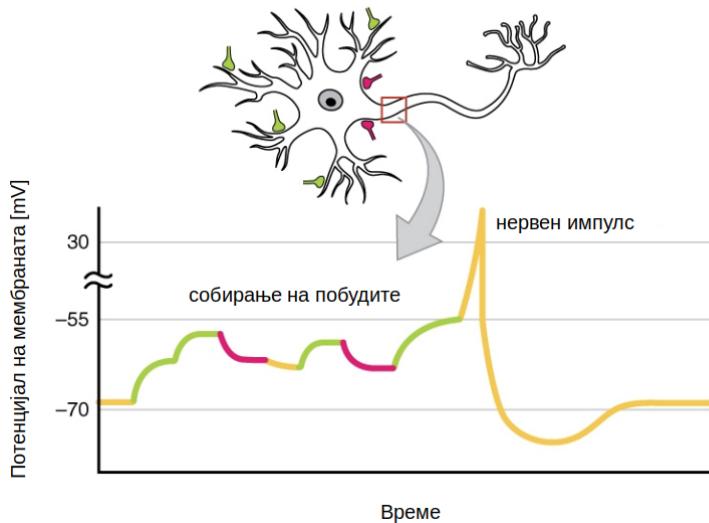
<sup>22</sup>Текстот во овој параграф користи исечоци од Wikipedia: Неврон <https://mk.wikipedia.org/wiki/Неврон>

<sup>23</sup>Wikipedia: Неврон <https://mk.wikipedia.org/wiki/Неврон>, оригинал од Quasar Jarosz на Википедија на английски, CC BY-SA 3.0, <https://commons.wikimedia.org/w/index.php?curid=7616130>

<sup>24</sup>Original uploader was Cahass at en.wikipedia — Transferred from en.wikipedia, CC BY-SA 3.0, <https://commons.wikimedia.org/w/index.php?curid=651365>

<sup>25</sup>By OpenStax College - Anatomy & Physiology, Connexions Web site. <http://cnx.org/content/col11496/1.6/>, Jun 19, 2013., CC BY 3.0, <https://commons.wikimedia.org/w/index.php?curid=30147931>

<sup>26</sup>By Laurentaylorj - Own work, CC BY-SA 3.0, <https://commons.wikimedia.org/w/index.php?curid=26311114>



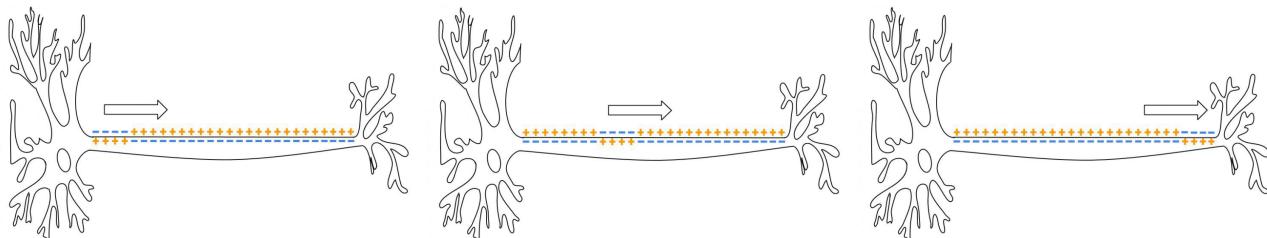
**Сл. 1.7:** Интегрирање на побудите во телото на невронот.<sup>25</sup>

Простирањето на деполаризацијата долж невронот е прикажана на сл. 1.8 и 1.9.<sup>27</sup> Таа се одликува со попречна струја насочена навнатре кон клетката на местото на простирање.

По пристигнувањето на нервните импулси на крајот од аксонот, тие предизвикуваат ослободување на неуротрансмитери од невронот. Овие неуротрансмитери претставуваат побуда за дендрити на други неврони или пак за мускулни клетки.<sup>30</sup> По простирањето на нервниот импулс низ аксонот на невронот, невронот мора повторно да го воспостави биоелектричниот потенцијал во мирување. Пред невронот да може да проследи нов нервен импулс, треба да помине времето на непобудливост.<sup>31,32</sup>

### Глиа клетки

Швановите клетки претставуваат еден вид на **глиа клетки** кои се задолжени за создавање на миелинската обвивка, види подолу. Глиа клетките се од суштинско значење за правилната работа на невроните.<sup>33</sup>



**Сл. 1.8:** Фази од простирањето на нервниот импулс.<sup>26</sup>

<sup>27</sup>Wikipedia: Action potential [https://en.wikipedia.org/wiki/Action\\_potential](https://en.wikipedia.org/wiki/Action_potential)

<sup>29</sup>Blausen.com staff. "Blausen gallery 2014". Wikiversity Journal of Medicine. DOI:10.15347/wjm/2014.010. ISSN 20018762. - Own work, CC BY 3.0, <https://commons.wikimedia.org/w/index.php?curid=29452220>

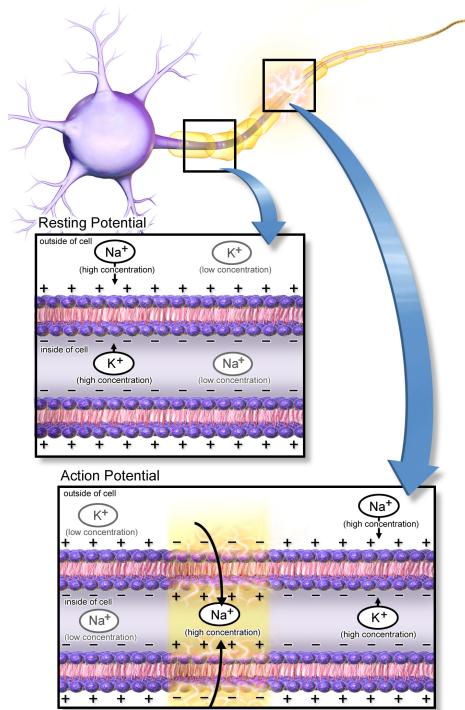
<sup>29</sup>By Artwork by Holly Fischer - <http://open.umich.edu/education/med/resources/second-look-series/materials> - CNS Slide 4, CC BY 3.0, <https://commons.wikimedia.org/w/index.php?curid=24367125>

<sup>30</sup>Постојат различни видови на неуротрансмитери и тие го насочуваат нашето однесување на различни начини. Интересно предавање поврзано со оваа тема е Simon Sinek: Why Leaders Eat Last <https://www.youtube.com/watch?v=ReRcHdeUG9Y>

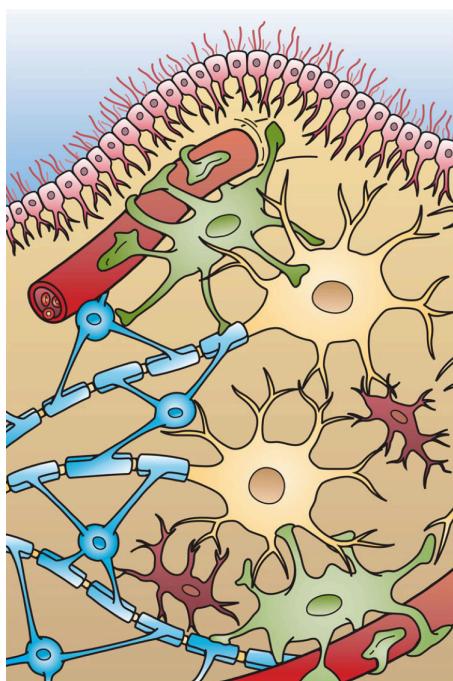
<sup>31</sup>Добра илустрација на целиот процес на простирање на нервните импулси е дадена во видеото The Nervous System, Part 2 - Action! Potential! Crash Course AP 9 [https://www.youtube.com/watch?v=0ZG8M\\_ldA1M](https://www.youtube.com/watch?v=0ZG8M_ldA1M)

<sup>32</sup>Интересно е да се спомене дека иако немаат нервен систем, растенијата исто така користат акциски потенцијали за сигнализација и движење TED: Greg Gage – Electrical experiments with plants that count and communicate <https://www.youtube.com/watch?v=pvBlSFVmaww>

<sup>33</sup>Wikipedia: Glia <https://en.wikipedia.org/wiki/Glia>



**Сл. 1.9:** Простирање на нервниот импулс.<sup>28</sup>



**Сл. 1.10:** Глиа клетки во централниот нервен систем (од горе надолу): епендимални клетки (розово), астроцити (зелено), микроглиални клетки (темно црвено) и олигодендроцити (сино).<sup>29</sup>

Уште повеќе, според некои истражувања нивното моделирање во вештачките невронски мрежи носи подобрени перформанси (Porto-Pazos et al., 2011). Некои од поважните типови на глиа клетки, претставени на Сл. 1.10, се:

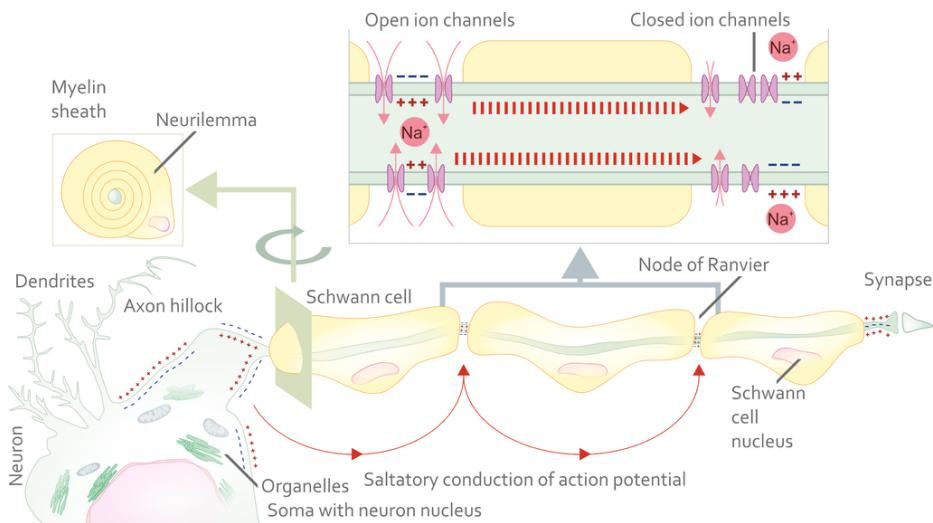
- епендималните клетки – одговорни за излачување и придвижување на мозочно-спиналната течност,
- астроцитите – задолжени за пренос на хранливите материји од капиларите до невроните и отпадните материји од невроните во спротивна насока тие се чуварите на бариерата мозок-крв,
- микроглиалните клетки – одговорни за инфраструктурно позиционирање на невроните, и
- олигодендроцитите (во периферниот нервен систем тоа се Швановите клетки) – одговорни за создавање на миелинската обвивка.

## Миелинска обвивка

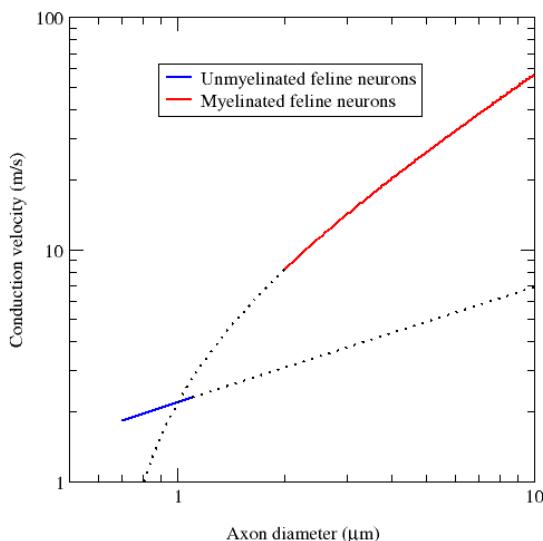
Миелинската обвивка се состои од низа на сегменти одделени со стеснувања наречени **јазли** на Ранвиер. Нејзината задача е да ја намали капацитетноста на мемраната и да ја зголеми нејзината отпорност во деловите помеѓу два соседни јазли преку попречување на движењето на јоните низ мемраната на аксонот. На тој начин, единствената размена на јони се случува во немиелизираните јазли, Сл. 1.11. Пропагацијата на акцискиот потенцијал долж деловите покриени со миелин се одвива преку струја во цитоплазмата која е доволно силна да предизвика деполаризацијата во соседниот јазол на Ранвиер. На овој начин деполаризацијата се простира со потскокнување од јазол до јазол.<sup>35</sup> Миелинот со тоа овозможува забрзан пренос на нервните импулси, како и поголема енергетска ефикасност на целиот процес. Последново е особено важно ако се има во предвид дека нервниот систем користи околу 20% од енергијата на човековиот организам.

<sup>34</sup>By Helixitta - Own work, CC BY-SA 4.0, <https://commons.wikimedia.org/w/index.php?curid=44043108>

<sup>35</sup>Wikipedia: Saltatory conduction [https://en.wikipedia.org/wiki/Saltatory\\_conduction](https://en.wikipedia.org/wiki/Saltatory_conduction)



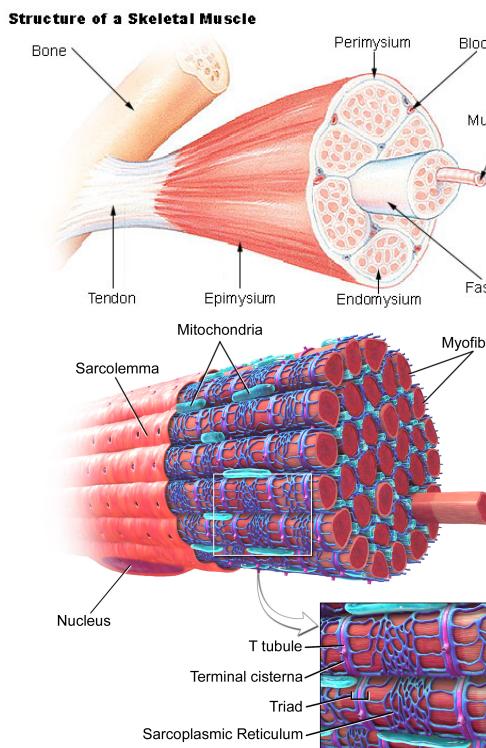
**Сл. 1.11:** Простирање на нервниот импулс во миелизиран неврон со скокови.<sup>34</sup>



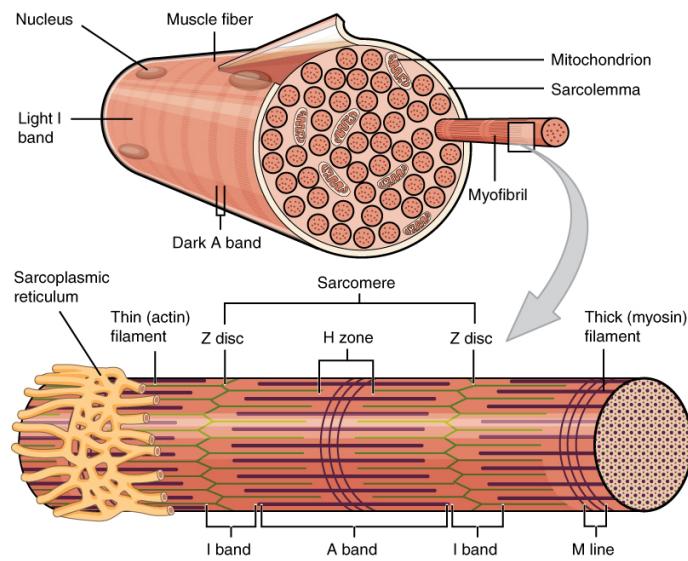
**Сл. 1.12:** Влијание на миелинската обвивка врз брзината на простирање на нервниот импулс.<sup>36</sup>

Миелинизацијата ја зголемува средната брзина на простирање на нервните импулси која се движи од 1 m/s до преку 100 m/s и расте со зголемување на дијаметарот на аксонот. Брзината на простирање во зависност од дијаметарот на аксонот во  $\mu\text{m}$  и миелинизацијата е дадена на Сл. 1.12. Брзината  $v$  на миелинираните неврони зависи линеарно од дијаметарот на аксонот  $d$ , т.е.  $v \propto d$ , додека брзината на немиелинизираните неврони зависи од квадратниот корен на дијаметарот  $v \propto \sqrt{d}$ . На графиконот екстраполираните вредности се прикажани со испрекината линија.

<sup>36</sup>By WillowW - Own work, CC BY 3.0, <https://commons.wikimedia.org/w/index.php?curid=3844705>



**Сл. 1.13:** Структура и состав на скелеталните мускули.<sup>37</sup>



**Сл. 1.14:** Структура и состав на скелеталните мускулни влакна.<sup>38</sup>

## 1.4 Градба и работа на мускулите

Мускулите претставуваат ткива составени од мноштво спонови на **мускулни влакна**, кои ја имаат способноста да генерираат сила и движење преку контракција. Тие го сочинуваат активниот дел на локомоторниот апарат.<sup>39</sup> Постојат три видови на мускулни ткива во човековиот организам:

- **напречно-пругасто** ткиво од кое се изградени скелеталните мускули кои преку **тетиви** се врзани за скелетот на човекот, имаат голема брзина на контракција и овозможуваат движење преку свесна контрола,
- **мазно** ткиво кое ги формира сидовите на внатрешните органи, а највеќе се среќаваат во сидовите на крвните садови и гастроинтестиналниот тракт; тие имаат бавна брзина на контракција и работат без наша волја, и
- **срцево** ткиво кое го формира сидовите на срцевите празнини, т.е. комори. По градбата е слично на напречно-пругастото, а по функција на мазното мускулно ткиво, со средна брзина на контракција и вон наша свесна контрола.

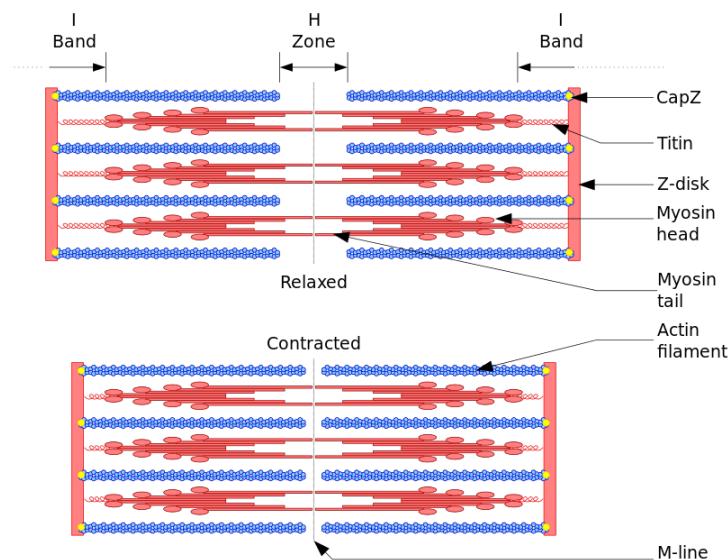
Структурата и составот на мускулните влакна од напречно-пругастото ткиво е прикажан на Сл. 1.14. Може да се види дека мускулните влакна се клетки со повеќе клеточни јадра. Нивната основна градбена единка се протеинските нишки наречени **миофибрили**. Миофибрилите се обвиткани од **саркоплазмичен ретикулум (СР)** и се составени од наизменична низа на напречни структури изградени од тенки нишки на **актин** и дебели нишки на **миозин**. Елементарниот дел на оваа повторувачка структура се нарекува **саркомер**. Поради внатрешната структура на

<sup>37</sup> Illu muscle structure – Public Domain, <https://commons.wikimedia.org/w/index.php?curid=1197565>

Skeletal Muscle Fiber – By BruceBlaus. Blausen.com staff (2014). "Medical gallery of Blausen Medical 2014". WikiJournal of Medicine 1 (2). DOI:10.15347/wjm/2014.010. ISSN 2002-4436. - Own work, CC BY 3.0, <https://commons.wikimedia.org/w/index.php?curid=29452230>

<sup>38</sup> Organization of skeletal muscle – By OpenStax - <https://cnx.org/contents/FPtK1zmh@8.25:fEI3C80t@10/Preface>, CC BY 4.0, <https://commons.wikimedia.org/w/index.php?curid=30015051>

<sup>39</sup> Википедија – Мускул <https://mk.wikipedia.org/wiki/Мускул>



**Сл. 1.15:** Меѓусебната положба на протеинските нишки при релаксација, односно контракција.<sup>41</sup>

миофибрилите, скелеталните, а и срцевите мускулни влакна имаат пругаст, т.е. стриест изглед.

Контракциите на скелетните мускули се **неврогенични**, односно тие се предизвикани од нервни импулси на моторните неврони, наречени мотоневрони, кои предизвикуваат деполаризација на мускулните влакна, а со тоа и нивна **контракција**.<sup>40</sup> Притоа, мотоневроните се разграничуваат и инервираат повеќе мускулни влакна, активирајќи ги истовремено. Збирот на мотоневрон со мускулните влакна кои ги инервира се нарекува **моторна единица**. Останатите два типа на мускули се **миогенични** што значи дека тие самите се одговорни за предизвикување на сопствената контракција, иако нивната работа може да биде модулирана од автономниот нервен систем.

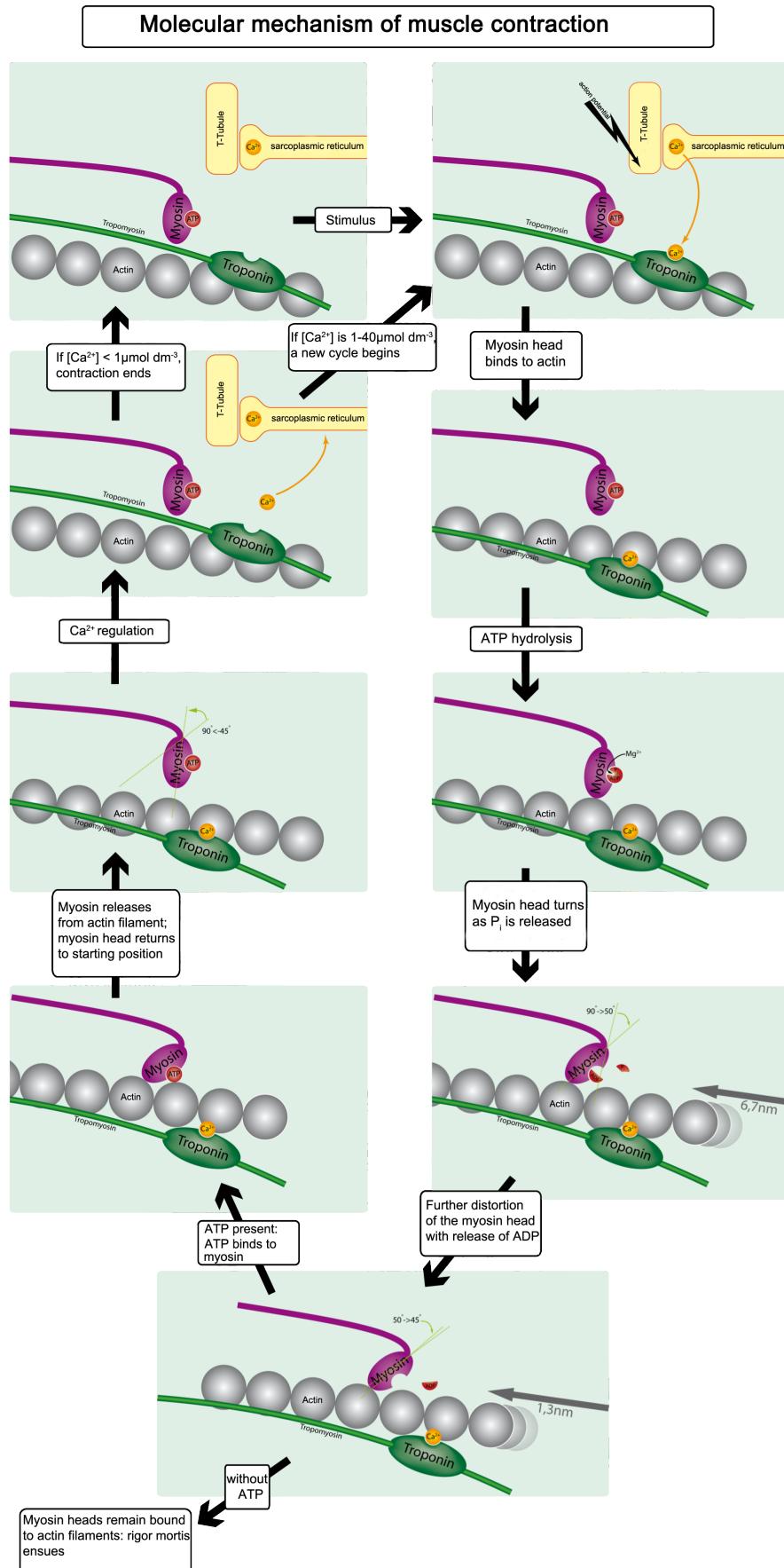
Покрај различниот тип на влезна побуда на која реагираат, контракцијата на сите три типа на мускулно ткиво се должи на истиот физиолошки механизам. Имено, при активација протеинските нишки на **актин** и **миозин** во мускулните влакна според **теорија на лизгање на нишките**, тие лизгаат едни покрај други генерирајќи сила и придвижување, т.е. контракција. На Сл. 1.15 е прикажана меѓусебната положба на протеинските нишки при релаксација, односно контракција на мускулните влакна.

Физиолошкиот процес којшто го овозможува лизгањето на нишките, односно контракцијата на мускулите, е прикажан чекор по чекор на Сл. 1.16.<sup>42</sup> Деполаризацијата на мускулното влакно ја предизвикува надворешниот акционен потенцијал на мотоневронот. Таа се пренесува низ Т тубулите и ги активира напонски контролирани калциумови помпи во саркоплазмичниот ретикулум, кој е одговорен за чување на калциумот во мускулните влакна. Ова предизвикува ослободување на калциумови  $\text{Ca}^{2+}$  јони од СР. Синхронизираното ослободување на калциум од илјадниците СР предизвикува зголемување на калциумот во целата мускулна клетка. Ако концентрацијата на калциум се зголеми во опсег  $1 - 40 \mu\text{mol}/\text{dm}^3$ , тој ја започнува контракцијата преку врзување за тропонинот. Оваа врска предизвикува промена во положбата на тромиозинот со што се ослободуваат локациите на актинот на којшто може да се врзе миозинот.

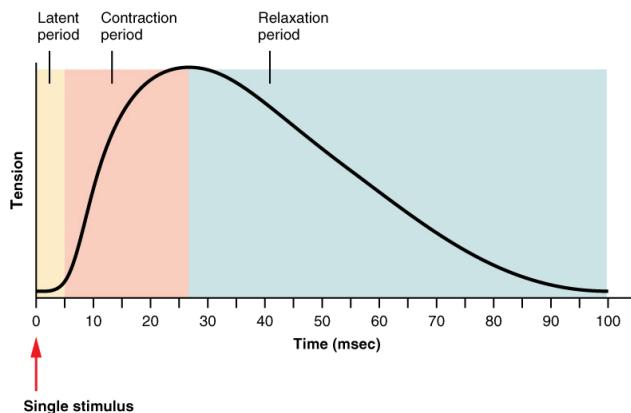
<sup>40</sup>Wikipedia: Muscle Contraction. [https://en.wikipedia.org/wiki/Muscle\\_contraction](https://en.wikipedia.org/wiki/Muscle_contraction)

<sup>41</sup>By Slashme at English Wikipedia. Richfield, David. "Medical gallery of David Richfield 2014". Wikiversity Journal of Medicine 1 (2). DOI:10.15347/wjm/2014.009. ISSN 2001-8762. - Own work, CC BY-SA 3.0, <https://commons.wikimedia.org/w/index.php?curid=2264027>

<sup>42</sup>Анимирана илустрација на целиот процес е дадена во: McGraw-Hill Animations – Muscle Contraction Process [HD Animation] <https://www.youtube.com/watch?v=ousflr0zQHc>



Сл. 1.16: Чекори во контракцијата на мускулните влакна.<sup>43</sup>



Сл. 1.17: Миограм на елементарна контракција на мускулно влакно.<sup>47</sup>

Во тој момент, миозинот, кој е во високо-енергетската состојба со врзана молекула на аденоzin дифосфат (АДФ) и неоргански фосфор Р, резултат на хидролизата на **аденоzin трифосфат (АТФ)** молекулата<sup>44</sup>, се врзува за актинот. При врзувањето се ослободуваат АДФ и Р молекулите во два чекори – во првиот се ослободува Р а потоа се ослободува останатиот АДФ. Ова предизвикува виткање на миозинот кој го повлекува актинот, генерирајќи сила, како и движење, кое изнесува околу 7 nm. Со ова миозинот ја добива својата ниско-енергетска состојба.

Во следниот чекор нова АТФ молекула се врзува за миозинот, што го ослободува од актинот.<sup>45</sup> Неговата хидролиза на АДФ и Р го враќа во почетната високо-енергетска состојба. Ако тропомиозинот се уште не го покрива актинот, целиот процес се повторува. Во целиот процес калциумот  $\text{Ca}^{2+}$  активно се впушчува во саркоплазмичниот ретикулум, па при недостиг на надворешен акционен потенцијал за активирање на неговото ослободување, неговата концентрација паѓа со што тропомиозинот се враќа во првичната состојба и мускулот преминува во состојба на релаксација.

Деполаризацијата на скелеталните мускули е слична со онаа на невроните.<sup>46</sup> Таа настапува поради активацијата на натриумовите канали, што предизвикува навлегување на натриумовите јони  $\text{Na}^+$  во внатрешноста на клетката. Овој процес престанува со затворањето на овие канали, а до реполаризација на клетката доаѓа поради истекувањето на калиумовите јони  $\text{K}^+$  надвор. Напонот на мирување е поголем во однос на овој на невроните и изнесува околу -90 mV, а при деполаризација може да достигне и до +75 mV. Акциониот потенцијал трае околу 2 – 4 ms, максималниот период на непобудливост околу 1 – 3 ms, а брзината на пренос на потенцијалот долж мускулот изнесува околу 5 m/s. Елементарна контракција на мускулно влакно предизвикана од единечен нервен импулс е прикажана на Сл. 1.17. Може да се види дека потребно е да помине одредено време, кое е од редот на 10 ms, за побудата да пропагира во мускулното влакно пред тоа да ја започне неговата контракцијата.

Доколку пристигне нов нервен импулс пред мускулното влакно да се врати во неговата релаксирана состојба доаѓа до таканаречена **сумација**. Ова е прикажано на Сл. 1.18. Постојат два типа на сумација и тоа:

- **фреквенциска сумација** – пристигнуваат поголем број на нервни импулси на истото мускулно влакно, и
- **сумација на повеќе влакна** – доаѓа до активирање на нови мускулни влакна, т.е. моторни

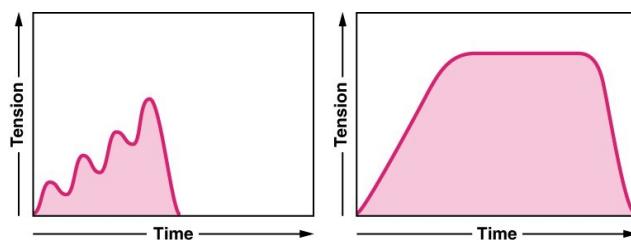
<sup>43</sup>By Muskel-molekular.png: Hank van Helvete derivative work: GravityGilly - Muskel-molekular.png, CC BY-SA 2.5, <https://commons.wikimedia.org/w/index.php?curid=10703870>

<sup>44</sup>Adenosine triphosphate (ATP) [https://en.wikipedia.org/wiki/Adenosine\\_triphosphate](https://en.wikipedia.org/wiki/Adenosine_triphosphate)

<sup>45</sup>При недостиг на АТФ, мускулите остануваат во грч.

<sup>46</sup>Wikipedia: Action potential [https://en.wikipedia.org/wiki/Action\\_potential](https://en.wikipedia.org/wiki/Action_potential)

<sup>47</sup>By OpenStax - <https://cnx.org/contents/FPtK1zmh@8.25:fEI3C80t@10/Preface>, CC BY 4.0, <https://commons.wikimedia.org/w/index.php?curid=30015047>

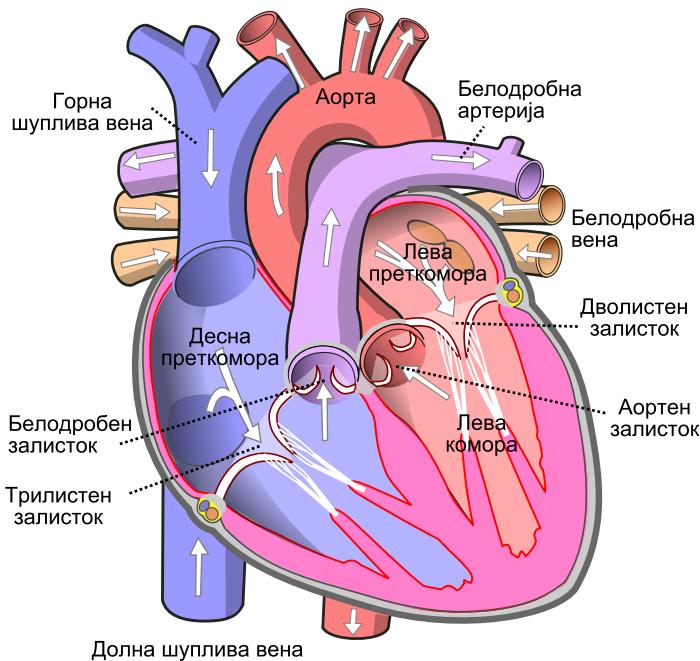


**Сл. 1.18:** Миограм на сумација и тетанус при контракција на мускул.<sup>48</sup>

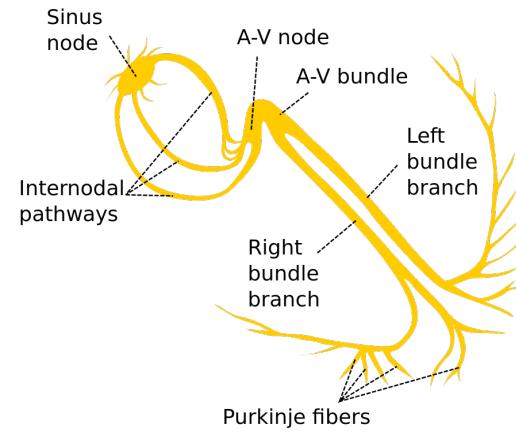
единици.

При максимална активација на мускулот, во фаза на контракција се околу 1/3 од мускулните влакна. На овој начин човековото тело се штити од можни оштетувања до кои може да дојде поради контракција на мускулите. Така, до оштетување на тетивите може да дојде при активација на 95% од мускулните влакна. Ако сумацијата продолжи и се интезивира, мускулот ќе ја достигне својата врвна сила и ќе влезе во заситување. Оваа појава се нарекува **тетанус**, Сл. 1.18.

<sup>48</sup>By OpenStax - <https://cnx.org/contents/FPtK1zmh@8.25:fEI3C80t@10/Preface>, CC BY 4.0, <https://commons.wikimedia.org/w/index.php?curid=30015045>



Сл. 1.19: Анатомија на човечкото срце.<sup>51</sup>



Сл. 1.20: Систем за пренос на деполаријација во срцето.<sup>52</sup>

## 1.5 Градба и работа на срцето

Срцето е празен мускулен орган чија функција е движењето на крвта во организмот.<sup>49</sup> Срцето лежи во срцевото ќесе - перикард, кое има заштитна функција. Срцевиот сид пак, е изграден од три слоеви: надворешен (епикард), среден (миокард) и внатрешен (ендокард). Во самото срце постојат крвни садови преку кои се снабдува со хранливи материји и кислород, а ги оддава екскретите и јаглерод диоксидот. Срцето има форма на тристрани пирамида, со врвот завртена на надолу и налево, со тежина од 230 до 340 g.

Срцето содржи четири комори: две **преткомори** и две главни **комори**. Мускулните сидови на преткоморите се потенки од тие на коморите, поради тоа што значаен удел во придвижување на крвта од преткоморите во коморите зема гравитационата сила. Иако волуменот на двете комори е ист, мускулниот сид на левата комора е подебел од оној на десната, бидејќи тука крвта се испумпува со голем притисок во аортата за да се разнесе низ целото тело преку големиот крвоток. Од друга страна, десната комора ја испумпува крвта во малиот крвоток кој има задача да ја збогати со кислород низ белите дробови и да ја врати во левата преткомора на срцето.

Меѓу преткоморите и коморите се наоѓаат **атриовентрикуларните (AV)** мембрани, и тоа: левата преткомора е одделена од левата комора со дволисни залистоци, додека меѓу десната преткомора и комора има тролисни залистоци. Меѓу коморите и излезните крвни садови постојат **полумесечеви (PM)** мембрани, обете тролисни.<sup>50</sup>

Во срцевиот циклус постојат 5 фази, во кои мускулите од преткоморите и коморите вршат контракција и релаксација, а атриовентрикуларните и полумесечевите мембрани се отвораат и затвораат. Тие се:

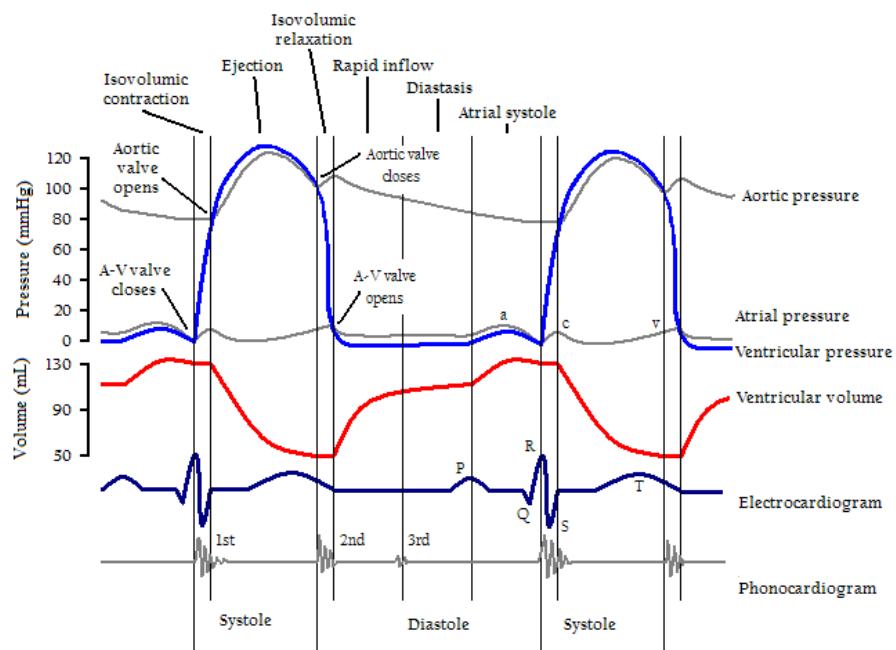
1. Контракција на преткомори – AV отворени, PM затворени, преткоморите ги надополнуваат коморите со околу 30% од волуменот на крвта при мирување од 130 ml.

<sup>49</sup> Википедија – Срце <https://mk.wikipedia.org/wiki/Срце>

<sup>50</sup> Wikipedia – Heart <https://en.wikipedia.org/wiki/Heart>

<sup>43</sup> Од original by Yaddah after Wapcapletderivative by B. Jankuloski — File:Diagram of the human heart (cropped).svg, CC BY-SA 4.0, <https://commons.wikimedia.org/w/index.php?curid=53624221>

<sup>52</sup> By Madhero88 - Own work, CC BY-SA 3.0, <https://commons.wikimedia.org/w/index.php?curid=6887499>.



Сл. 1.21: Срцев циклус.<sup>55</sup>

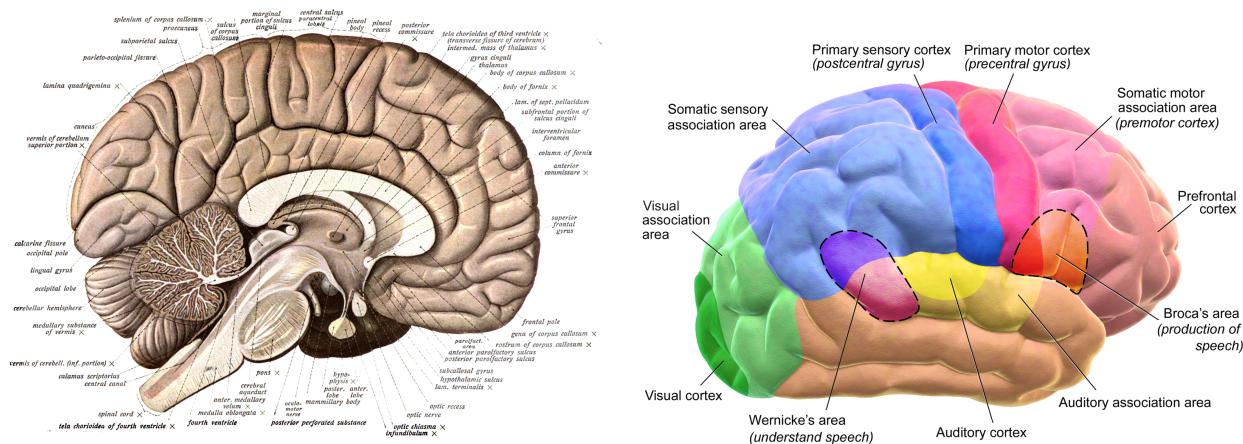
2. Изоволумена контракција на комори – АВ затворени, ПМ затворени, коморите започнуваат со контракција, нивниот волумен не се менува но притисокот расте.
3. Контракција на комори со испумпување – АВ затворени, ПМ отворени, притисокот во коморите го надминува оној во аортата (80 mmHg) и пулмонарната артерија (10 mmHg), исфрлањето е ненадејно и истиснува 70 – 90 ml од секоја комора, што остава околу 50 ml за следната контракција.
4. Изоволумена релаксација на комори – АВ затворени, ПМ затворени, притисокот во коморите паѓа од 120 mmHg што ги затвора ПМ мембрани, АВ отвораат штом овој притисок падне под оној во преткоморите.
5. Полнење на комори – АВ отворени, ПМ затворени, коморите пасивно се полнат до 70% од волуменот. Истото важи и за преткоморите.

За време на секое отчукување срцето минува редоследно низ овие фази од срцевиот циклус<sup>53</sup> прикажан на Сл. 1.21, што е овозможено од прогресивната деполаризација на срцевиот мускул. Оваа деполаризација започнува од **пејсмејкер** клетките во **синоатриалниот јазол**, сешири низ преткоморите, поминува низ **атриовентрикуларниот јазол** надолу кон **чопот на Хис**, за на крајот да се прошири преку **нишките на Пуркинџе** низ коморите. Овие ткива го сочинуваат **електричниот систем за пренос на срцето**<sup>54</sup>, кој е одговорен за пренесување на електричните импулси на деполаризацијата низ срцевиот мускул прикажан на Сл. 1.20. АВ јазолот го внесува потребното задоцнување во простирањето на деполаризацијата за контракцијата коморите да дојде дури по постигнување на максималната контракција на преткоморите. Нишките на Пуркинџе пак се шират од дното на срцето па нагоре за контракцијата на коморите да започне од затворениот крај кон отворите со што се овозможува поефикасно испумпување на крвта.

<sup>53</sup>Wikipedia: Cardiac cycle [https://en.wikipedia.org/wiki/Cardiac\\_cycle](https://en.wikipedia.org/wiki/Cardiac_cycle)

<sup>54</sup>Wikipedia: Electrical conduction system of the heart [https://en.wikipedia.org/wiki/Electrical\\_conduction\\_system\\_of\\_the\\_heart](https://en.wikipedia.org/wiki/Electrical_conduction_system_of_the_heart)

<sup>55</sup>By DanielChangMD revised original work of DestinyQx - Wikimedia Commons, File:Cardiac Cycle Left Ventricle.PNG. Source file has a serious mistake which was corrected in the new image., CC BY-SA 2.5, <https://commons.wikimedia.org/w/index.php?curid=17610757>



Сл. 1.22: Анатомска и функционална структура на човечкиот мозок.<sup>60</sup>

## 1.6 Градба и работа на мозокот

### Нервен систем кај човекот

Нервниот систем е органски систем кој претставува целина од нервните елементи: невроните, нервите и нервните ткива.<sup>56</sup> Основната морфолошка - физиолошка единица на нервниот систем е невронот, односно нервната клетка. Нервниот систем влијае и ја контролира работата на сите останати органски системи со помош на нервите и нивните разграноци; тој навлегува во сите делови на телото и е во врска со сите ткива и нивните клетки.

Каде 'рбетниците нервниот систем е составен од централен и периферен нервен систем. Черепниот и 'рбетниот мозок го сочинува централниот нервен систем. Од друга страна, периферниот нервен систем го сочинуваат сите нервни клетки надвор од централниот нервен систем, во кои влегуваат телесниот и автономниот нервен систем.<sup>57</sup> Автономниот, односно вегетативниот, нервен систем пак се дели на симпатичен и парасимпатичен нервен систем.<sup>58</sup>

### Черепен мозок

Черепниот мозок е виш нервен центар и е дел од централниот нервен систем.<sup>59</sup> Неговата структура и функционални единици се прикажани на Сл. 1.22.

Черепниот мозок е составен од неколку делови, и тоа:

- голем или преден,
- меѓумозок,
- среден,
- мал или заден, и
- продолжен мозок.

<sup>56</sup> Википедија – Нервен систем [https://mk.wikipedia.org/wiki/Нервен\\_систем](https://mk.wikipedia.org/wiki/Нервен_систем)

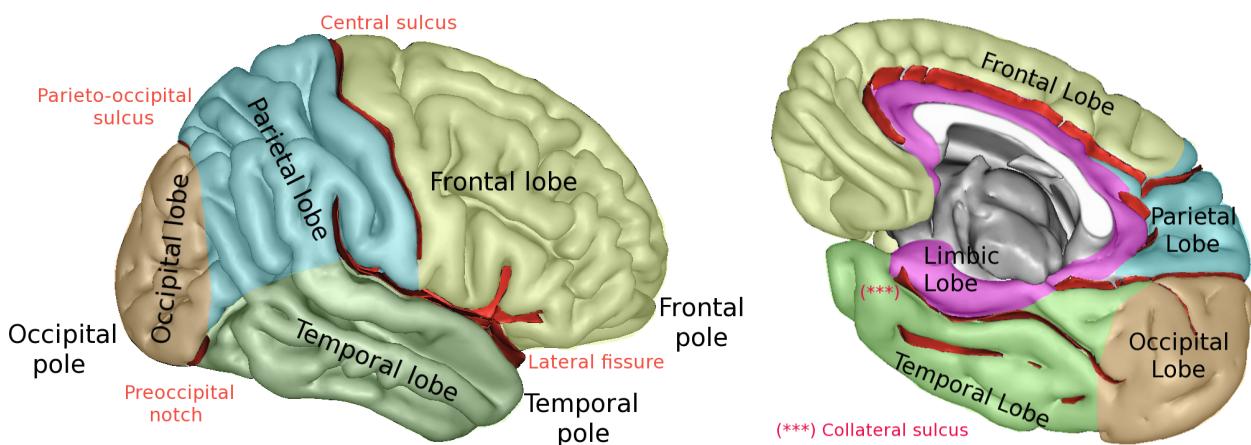
<sup>57</sup> Иако растенијата немаат нервен систем, тие некогаш содржат ткива кои ја играат неговата улога за координација на движењето на одредени нивни делови. TED – Greg Gage – Electrical experiments with plants that count and communicate <https://www.youtube.com/watch?v=ppvBLSFVmaw>

<sup>58</sup> Википедија – Автономен нервен систем [https://mk.wikipedia.org/wiki/Автономен\\_нервен\\_систем](https://mk.wikipedia.org/wiki/Автономен_нервен_систем)

<sup>59</sup> Википедија – Мозок <https://mk.wikipedia.org/wiki/Мозок>

<sup>60</sup> By Dr. Johannes Sobotta - Atlas and Text-book of Human Anatomy Volume III Vascular System, Lymphatic system, Nervous system and Sense Organs, Public Domain, <https://commons.wikimedia.org/w/index.php?curid=29135452>

Blausen.com staff (2014). "Medical gallery of Blausen Medical 2014". WikiJournal of Medicine 1 (2). DOI:10.15347/wjm/2014.010. ISSN 2002-4436. - Own work, CC BY 3.0, <https://commons.wikimedia.org/w/index.php?curid=60100749>



Сл. 1.23: Распоред на лобусите во мозокот.<sup>61</sup>

Кај големиот и малиот мозок има мозочна кора која е набрчкана, со многу бразди или виуги.

Постојат два типа на нервното ткиво кај черепниот мозок:

- сива маса – составена од невроцити и дендрити и
- бела маса – составена од аксони.

Белата маса е распределена во централниот дел на черепниот мозок, додека пак сивата маса е распоредена во периферниот дел, каде ја оформува мозочната кора и субкортикалните јадра.

### Голем мозок

Големиот мозок<sup>62</sup> е поделен на две хемисфери, лева и десна, поврзани меѓусебно со корпус калосумот<sup>63</sup>. Двете хемисфери макроскопски претставуваат огледални слики една на друга со мали разлики.<sup>64</sup> На микроскопско ниво постојат значителни разлики во функциите на клетките и количините на невротрансмитери и ним соодветните рецептори. Така, постојат повисоки нивоа на невротрансмитерот норепинефрин, т.е. норадреналин, во десната хемисфера. Норадреналинот во мозокот е одговорен за зголемена побуда и внимание, го подобрува запишувањето и исчитувањето на сеќавања, и го фокусира вниманието.<sup>65</sup> Од друга страна, во левата хемисфера постојат повисоки нивоа на допамин, кој е клучен за мотивација во однесувања базирани на потрага по награда.<sup>66</sup> Исто така, има повеќе бела маса во десната хемисфера и повеќе сива маса во левата хемисфера.

Латерализација во функцијата помеѓу хемисферите е посебно изразена за функции од ниско ниво како контролата на деловите на телото или приемот на сетилните информации од спротивната страна на телото. Најдобриот пример за воспоставена латерализација на функции од високо ниво се регионите на Брока и Вернике кои се одговорни за јазикот и зборувањето, кои често се наоѓаат исклучиво во левата хемисфера. Всушност, тие се поврзани со доминантната рака, па кај леваците знаат да бидат и во десната хемисфера. Освен ова, популарните тврдења за латерализација на логичките и креативните функции се неосновани.

<sup>61</sup>By Sebastian023, CC BY-SA 3.0, <https://commons.wikimedia.org/w/index.php?curid=21020857>

By Sebastian023, CC BY-SA 3.0, <https://commons.wikimedia.org/w/index.php?curid=21020886>

<sup>62</sup>Големината на големиот мозок е поврзана со големината на телото и следи закон на степен со експонент од 0,75. На пример, најголем мозок имаа китовите тежок 8 kg, додека слоновите имаат мозок тежок 5 kg. Отстапувањата од ова правило се најзначајни кај мајмуните, кои имаат 5 – 10 пати поголем мозок од предвиденото. Ова, како и густината на нервните клетки, начинот и брзината на нивната поврзаност се физички корелати на интелигенцијата.

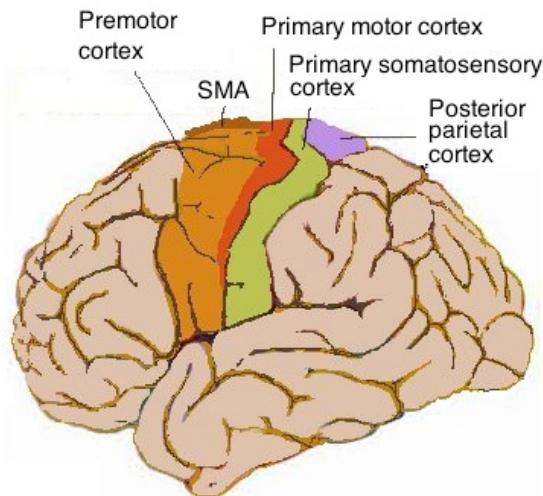
Wikipedia – Brain size [https://en.wikipedia.org/wiki/Brain\\_size](https://en.wikipedia.org/wiki/Brain_size)

<sup>63</sup>лат. *corpus callosum*

<sup>64</sup>Wikipedia – Cerebral hemisphere [https://en.wikipedia.org/wiki/Cerebral\\_hemisphere](https://en.wikipedia.org/wiki/Cerebral_hemisphere)

<sup>65</sup>Wikipedia – Norepinephrine <https://en.wikipedia.org/wiki/Norepinephrine>

<sup>66</sup>Wikipedia – Dopamine <https://en.wikipedia.org/wiki/Dopamine>



Сл. 1.24: Поставеност на моторниот и соматосензорниот кортекс.<sup>68</sup>

### Лобуси во големиот мозок

Лобусите на мозокот се анатомски разграничени делови на мозокот кои се задолжени за различни мозочни функции.<sup>67</sup> Големиот мозок е поделен на 6 лобуси:

- преден или фронтален лобус,
- париетален лобус,
- окципитален лобус,
- слепоочен или темпорален лобус,
- лимбичен лобус, и
- инсуларен кортекс.

Нивната поставеност во големиот мозок е прикажана на Сл. 1.23.

Фронталниот лобус се наоѓа на предниот крај на секоја од хемисферите. Фронталниот лобус ги содржи повеќето од невроните осетливи на допамин во мозокот, кои се асоцираат со награда, внимание, кратко-трајно памтење, планирање и мотивација. Допаминот, исто така, ги ограничува и селектира сензорните информации кои доаѓаат од таламусот. На предниот крај од фронталниот лобус е поместен префронталниот кортекс кој е поврзан со желбата за живот, карактеристиките на личноста, планирање на комплексни дејствија, носењето одлуки и прилагодување на социјалното однесување.<sup>69</sup> На задниот крај од фронталниот лобус се наоѓа моторниот кортекс, кој е одговорен за волева, т.е. свесна, контрола на скелеталните мускули.

Париеталниот лобус служи за интегрирање на сетилните информации помеѓу различните модалности, вклучувајќи го чувството за простор и навигација. Тој го содржи и дорсалниот поток на визуелниот систем. На предниот крај се наоѓа соматосензорниот кортекс до кој доаѓаат најголемите сетилни информации од кожата како допир, температура и болка.

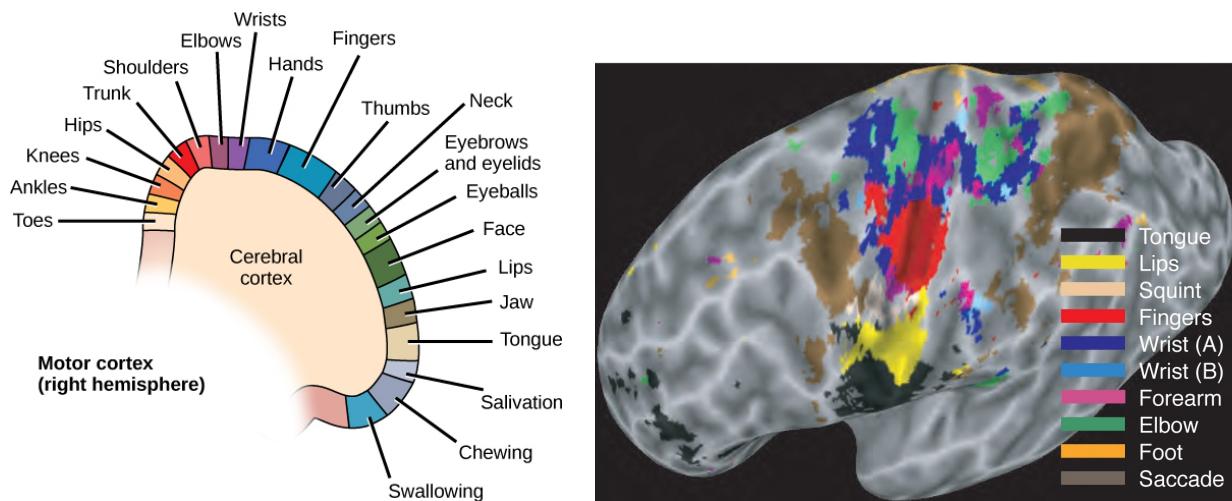
Окципиталниот лобус е центарот за процесирање на визуелни информации и го содржи речиси целиот визуелен кортекс, наречен и V1. Тој вклучува региони специјализирани за визуелопросторно процесирање, разликување на бои, и перцепција на движење.

Темпоралниот лобус е вклучен во процесирањето на сетилни информации за добивање на

<sup>67</sup> Wikipedia – Lobes of the brain [https://en.wikipedia.org/wiki/Lobes\\_of\\_the\\_brain](https://en.wikipedia.org/wiki/Lobes_of_the_brain)

<sup>68</sup> By Cortex sensorimoteur1.jpg: Pancratderivative work: Iamozy - Own work, This file was derived from: Cortex sensorimoteur1.jpg:, CC BY-SA 3.0, <https://commons.wikimedia.org/w/index.php?curid=33981076>

<sup>69</sup> Wikipedia – Prefrontal cortex [https://en.wikipedia.org/wiki/Prefrontal\\_cortex](https://en.wikipedia.org/wiki/Prefrontal_cortex)



**Сл. 1.25:** Распределба на контролата на скелеталните мускули долж моторниот кортекс теоретски и практично.<sup>71</sup>

изведени значења потребни за зачувување на визуелни сеќавања, разбирање на јазик и асоцијација на емоции. Тој исто така го содржи **аудиторниот кортекс**, познат и како A1.

**Лимбичниот лобус** ја сочинува средната површина на хемисферите и содржи делови од фронталниот, париеталниот и темпоралниот лобус.

**Инсуларниот кортекс** се наоѓа длабоко во јазот помеѓу фронталниот и темпоралниот лобус.<sup>70</sup> Тој е клучен за свеста и е поврзан со емоциите и регулирањето на телесната хомеостаза. ја сочинува средната површина на хемисферите и содржи делови од фронталниот, париеталниот и темпоралниот лобус.

## Моторен и соматосензорен кортекс

Положбата на моторниот и соматосензорниот кортекс се дадени на Сл. 1.24.

**Моторниот кортекс** се наоѓа на задниот крај од фронталниот лобус и е вклучен во планирањето, контролата и извршувањето на волеви, или свесни, движења. Тој е поделен на три дела:

- **примарен моторен кортекс** – ги генерира нервните импулси кои низ ’рбетниот мозок се спроведуваат до скелеталните мускули,
- **предмоторен кортекс** – одговорен за моторна контрола, вклучувајќи подготвка за движење, сетилно управување со движењето, просторно водење при посегнувањето, како и директна контрола на некои мускули,
- **додатна моторна област** – одговорна за планирање на секвенца од движења, како и координирања на двете страни на телото.

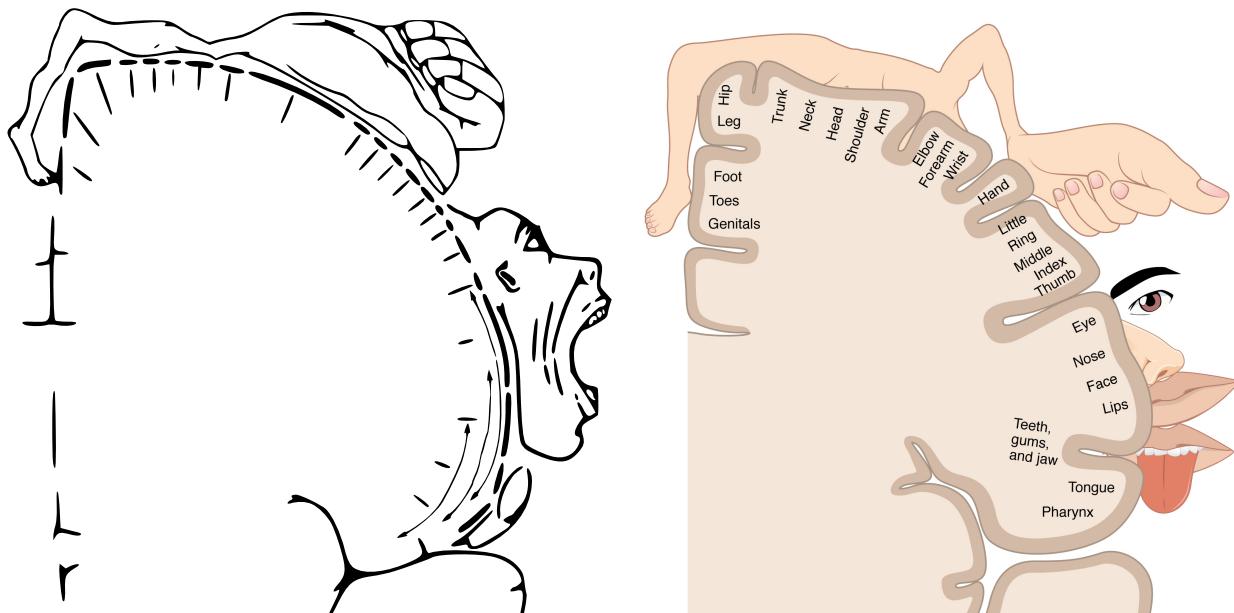
Поедноставената теорија за поврзувањето на моторниот кортекс со скелеталните мускули постулира дека секој неврон е директно поврзан со мускулот кој го контролира, како и дека секоја точка во моторниот кортекс контролира еден или мала група на поврзани мускули. При тоа, деловите на телото се непропорционално распределени долж моторниот кортекс во зависност од потребата за прецизна контрола, како што е прикажано на Сл. 1.25.

Оваа распределба често се претставува преку „кортикално човече“ или хомункулус. Човечето има

<sup>70</sup>Wikipedia – Insular cortex [https://en.wikipedia.org/wiki/Insular\\_cortex](https://en.wikipedia.org/wiki/Insular_cortex)

<sup>71</sup>By CNX OpenStax - [http://cnx.org/contents/GFy\\_h8cu@10.53:rZudN6XP@2/Introduction](http://cnx.org/contents/GFy_h8cu@10.53:rZudN6XP@2/Introduction), CC BY 4.0, <https://commons.wikimedia.org/w/index.php?curid=49934991>

By Michael Graziano - emailed personally, CC BY-SA 1.0, <https://commons.wikimedia.org/w/index.php?curid=52490617>



**Сл. 1.26:** Моторно (лево) и соматосензорно човече (десно).<sup>73</sup>

несразмерна големина на деловите на телото која одговара на нивната застапеност на мозочната кора. Моторното кортикално човече е прикажано на Сл. 1.26.

Во реалноста ова не е сосема точно. Повеќето неврони од моторниот кортекс се само поврзани со синапсата на 'рбетниот мозок, а не директно со моторните неврони. Исклучок од ова се директната кортико-моторна врска за контрола на прстите, која претставува еволуциска специјализација. Уште повеќе, експериментите покажуваат дека секоја точка од моторниот кортекс делува врз низа на мускули и зглобови, а притоа постои големо ниво на преклопување, Сл. 1.25.<sup>72</sup>

Соматосензорниот кортекс се наоѓа на предниот крај од париеталниот лобус, веднаш до моторниот кортекс, и е собирен центар за најголемите сетилни информации од кожата: допир, температура и болка. Слично како и моторниот кортекс, деловите од телото се нерамномерно застапени, според бројот на сетилни неврони кој го содржат. Според оваа распределба се дефинира и „соматосензорното човече“ прикажано на Сл. 1.25.

<sup>72</sup>Објаснување за ова нудат експериментите со мајмуни во кои при стимулација на различни центри на моторниот кортекс доаѓа до извршување на комплексни движења, на пр. затворање на дланката и нејзино поместување до устата која се отвора, или отворање на дланката и посегнување со раката напред. Навистина, компјутациски модели покажале дека кога нормалниот репертоар на движења на мајмуните би се мапирал според сличноста, би се добил распоред кој одговара на нивната распределба во моторниот кортекс.

<sup>73</sup>By mailto:ralf@ark.in-berlin.de - File:Homunculus-ja.pngTranslated from Image:Homunculus-de.svg by Was a bee., CC BY-SA 4.0, <https://commons.wikimedia.org/w/index.php?curid=49377875>

By OpenStax College - Anatomy Physiology, Connexions Web site. <http://cnx.org/content/col11496/1.6/>, Jun 19, 2013., CC BY 3.0, <https://commons.wikimedia.org/w/index.php?curid=30148008>

## Поглавје 2

# Снимање на биомедицинските сигнали

### 2.1 Фотоплетизмографија

Фотоплетизмографијата (ФПГ)<sup>1</sup> е оптичка метода за мерење на промените во волуменот на крвта во капиларите на кожата и поткожното ткиво.<sup>2</sup> Уредот за ФПГ се нарекува **фотоплетизмограф**, а снимката **фотоплетизмограм**. ФПГ најчесто се добива со уред наречен импулсен оксиметар<sup>3</sup>, прикажан на сл. 2.1. Еден типичен ФПГ сигнал е прикажан на сл. 2.2.

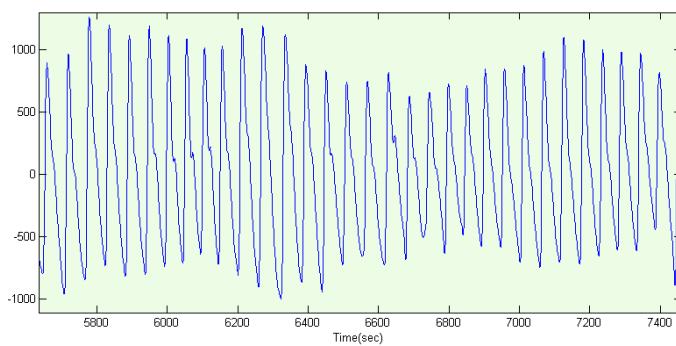
#### Принцип на работа

ФПГ се добива со осветлување на кожата со ЛЕД диода и мерење на промените во апсорпцијата на светлина на кожата со помош на фотодиода, односно фототранзистор. Притоа мерењето најчесто се врши на светлината која продира низ кожата, а може да се направи и на светлината која се рефлектира од неа.

Промените во апсорпцијата на светлина се должат на промената на волуменот на крвта предизвикани пред сè од испумпувањето на крвта од срцето. Со секоја контракција на коморите, нов бран на притисок се шире низ крвните садови. Иако бранот слабее при простирањето низ кардиоваскуларниот систем, сепак е доволен да доведе до проширување на артериите и



Сл. 2.1: Импулсен оксиметар.<sup>4</sup>



Сл. 2.2: Фотоплетизмограм добиен со импулсен оксиметар поставен на увото.<sup>5</sup>

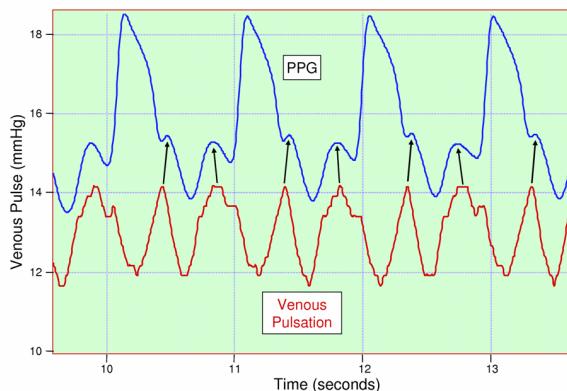
<sup>1</sup>англ. *Photoplethysmography (PPG)*

<sup>2</sup>Wikipedia – Photoplethysmogram <https://en.wikipedia.org/wiki/Photoplethysmogram>

<sup>3</sup>Wikipedia – Pulse oximetry [https://en.wikipedia.org/wiki/Pulse\\_oximetry](https://en.wikipedia.org/wiki/Pulse_oximetry)

<sup>4</sup>By User:Spl4 - Own work, Public Domain, <https://commons.wikimedia.org/w/index.php?curid=17870721>

<sup>5</sup>By TomTrottier.com - Own work, CC BY-SA 4.0, <https://commons.wikimedia.org/w/index.php?curid=81502253>



**Сл. 2.3:** ФПГ со видливо суперпониран сигнал од пулсирањето во вените.<sup>6</sup>

капиларите во поткожното ткиво. Секое проширување на крвните садови се регистрира како импулс во брановиот облик на ФПГ сигналот. Ако оксиметарот не ја притиска премногу кожата, во ФПГ може да се види и пулсот на проширување на вените, прикажан на сл. 2.3.

### Мерење на ФПГ

Вообичаено импулсниот оксиметар се поставува на прстот. Меѓутоа, во случаи на шок, хипотермија и сл. кога периферниот крвоток е намален, импулсниот оксиметар се поставува на главата и тоа на ушите, носот или челото. Исто така, ФПГ кај бебиња или мали деца кај кои не може да се добие добар сигнал од прстите, оксиметарот може да биде поставен на раката или ногата. Голема пречка во снимањето на добар ФПГ сигнал е појавата на артефакти предизвикани од движење.

ФПГ може да биде мерен и на повеќе локации на телото, на пр. двете уши, показалците и големите прсти на ногата. Ова се нарекува мулти-локацијска фотоплетизмографија (МФПГ)<sup>7</sup>

Дополнително, оксиметрите преку употреба на две LED диоди – едната со црвена а другата со инфрацрвена светлина, можат да ја одредат и сатурацијата на крвта со кислород ( $\text{SpO}_2$ )<sup>8</sup>, прикажано на сл. 2.4. Методата се заснова на различната апсорпција на светлина на овие две бранови должини од хемоглобинот во зависност од тоа дали носи кислород (оксихемоглобин) или не (деоксихемоглобин) прикажана на сл. 2.5. Оксиметрите можат да ја одредат сатурацијата со точност во рамките на 2% споредено со поточното но инвазивно одредување на истата. Нормалната сатурација треба да биде во опсегот 95 – 99%.

### Примена

Најчестата примена на ФПГ е да се одреди брзината на пулсот на срцето. Затоа вообичаено оксиметрите најчесто не го прикажуваат измерениот ФПГ, ами само го прикажуваат моменталниот пулс изразен во отчукувања во минута (bpm)<sup>13</sup>. Оксиметрите исто така ја прикажуваат и сатурацијата на крвта со кислород.

Висината на наизменичната компонента на ФПГ е пропорционална на разликата на систолниот и дијастолниот притисокот. Постојат методи кои со помош на калибрација можат да ја одредат

<sup>6</sup>CC BY-SA 3.0, <https://commons.wikimedia.org/w/index.php?curid=77541535>

<sup>7</sup>англ. *multi-site photoplethysmography (MPPG)*.

<sup>8</sup>англ. *saturation of peripheral oxygen*.

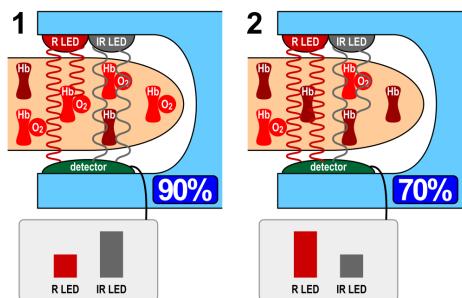
<sup>10</sup>By Adrian Curtin - Own work, CC BY-SA 3.0, <https://commons.wikimedia.org/w/index.php?curid=20510064>

<sup>10</sup>By Cmglee - Own work, CC BY-SA 4.0, <https://commons.wikimedia.org/w/index.php?curid=129109035>

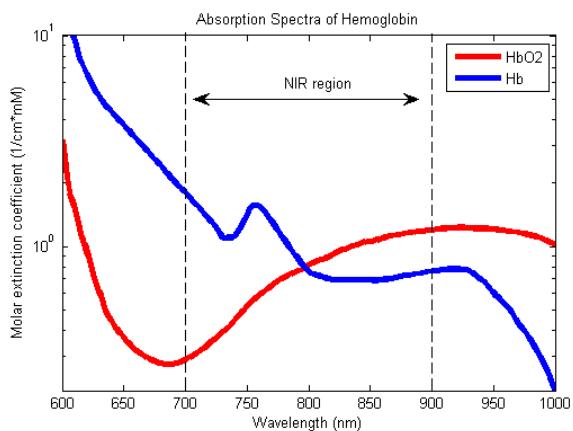
<sup>12</sup>CC BY-SA 3.0, <https://commons.wikimedia.org/w/index.php?curid=77541527>

<sup>12</sup>CC BY-SA 3.0, <https://commons.wikimedia.org/w/index.php?curid=77541546>

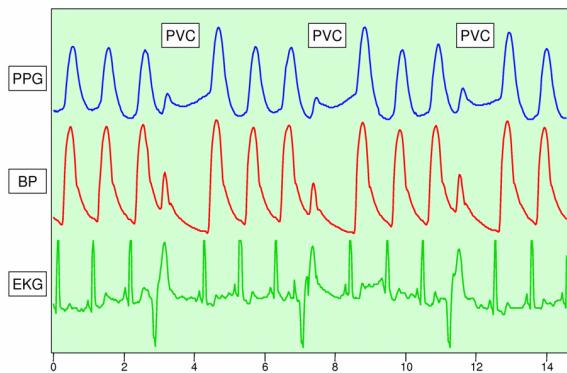
<sup>13</sup>англ. *beats per minute*.



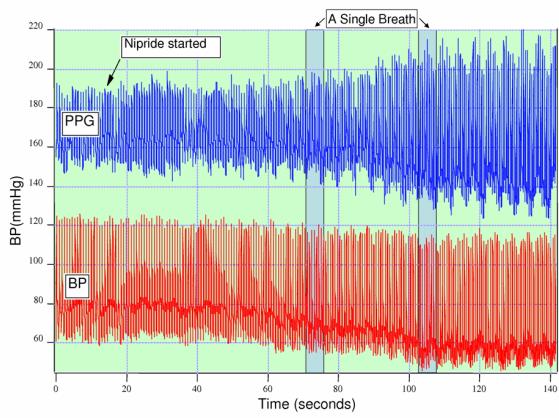
**Сл. 2.4:** Поедноставен принцип на работа на импулсниот оксиметар.<sup>9</sup>



**Сл. 2.5:** Апсорбициони спектри на оксихемоглобин ( $\text{HbO}_2$ ) и деоксихемоглобин (Hb) за првена и инфра првена бранова должина.<sup>10</sup>



**Сл. 2.6:** ФПГ со прерана вентрикуларна контракција.<sup>11</sup>



**Сл. 2.7:** Циклусот на вдишување може да се забележи во амплитудната анвелопа на ФПГ. На оваа слика се гледа и зголемувањето на амплитудата на ФПГ како последица на анестезија.<sup>12</sup>

апсолутната вредност на крвниот притисок врз база на ФПГ. Ова е поудобна алтернатива на класичното мерење на притисок со сфингломанометар, а овозможува и континуирано следење на промените на притисокот.

Од обликот на ФПГ можат да се видат атипични отчукувања на срцето. На сл. 2.6 е прикажан брановиот облик на предвремена вентрикуларна контракција. Други нарушувања во работата на срцето кои можат да се видат од ФПГ се вентрикуларна тахикардија и вентрикуларна фибрилација.

Освен работата на срцето, протокот на крвта во кожата може да биде модулиран и од повеќе други физиолошки системи, пред сè дишењето. При вдишување се намалува интраплеуралниот притисок помеѓу градниот кош и белите дробови. Ова го намалува притисокот врз шупливатата вена и десната преткомора, со што се зголемува протокот, но се намалува испумпаниот волумен, па срцето забрзува. При издишувањето, се случува спротивното па имаме зголемен испумпан волумен и намалување на пулсот. Поради тоа брзината на дишење може да се одреди од амплитудните промени на ФПГ, како што може да се види на сл. 2.7.

Со помош на ФПГ може да се следи и длабочината на анестезијата. Така, ефектите на анестезијата можат да се видат преку зголемување на динамиката на ФПГ, сл. 2.7. Дополнително, за време на операција, анестезиологот мора да врши постојана проценка дали

пациентот е доволно под анестезија. Ова може да се направи преку увидување на промени во ФПГ сигналот при нанесување на рез на пациентот.

Конечно, со помош на мулти-локациската МФПГ може да се оцени состојбата на периферната циркулација на испитаникот. На овој начин можат да се детектираат болести на периферната циркулација, дисфункција на автономниот нервен систем, некои типови на коронарни артериски болести и вкочанетост на артериите.

## 2.2 Електромиографија

Електромиографијата е електродијагностичка медицинска техника за евалуација и снимање на електричната активност на скелеталните мускули, [Поглавје 1.4.](#)<sup>14</sup> Уредот за ЕМГ се нарекува електромиограф, а снимката електромиограм. Електромиографот го детектира електричниот потенцијал генериран со деполаризација на мускулните клетки. Анализата на ЕМГ сигналот има повеќекратно значење во медицината: за анализа на абнормалности во мускулната функција, односно за идентификација на невромускулни болести, како и за детектирање на нарушувања во моторната контрола. Во биомедицинското инженерство ЕМГ сигналот може да се употреби за анализа на нивото на активација односно регрутација на мускулните влакна, или за анализа на биомеханиката на човековите или животинските движења. Преку него може да се изврши ефикасна контрола на простетичките уреди како простетички дланки, раце и нозе. Не постојат познати контраиндикации од правењето на ЕМГ.

Постојат два вида на ЕМГ:

- површинска ЕМГ – која неинвазивно ја проценува мускулната активност преку нејзино снимање од површината на кожата, и
- интрамускулна ЕМГ - која инвазивно ја снима мускулната активност преку забодување на иглести електроди во мускулното ткиво.

ЕМГ може да биде снимена од еден пар електроди, но и од покомплексна низа составена од повеќе електроди. Притоа мора да бидат искористени најмалку две електроди поради тоа што ЕМГ ја регистрира разликата во потенцијали помеѓу две точки на телото.

### Површинска ЕМГ

Ограничувањата во површинската ЕМГ се тоа што таа е ограничена на снимање на мускули кои се наоѓаат близу до површината на кожата, меѓусебната интерференција меѓу соседните мускули, како и негативното влијание на дебелината на поткожното ткиво на локацијата за снимање која варира со тежината на човекот. Поради тоа, површинската ЕМГ може да послужи само за ограничена проценка на мускулната активност.

Површинската ЕМГ се употребува во клиниките за физиотерапија во скlop на системите за [биофидбек](#) кои им даваат визуелна или звучна информација за нивото на активација на мускулот на лукето кои доаѓаат за рехабилитација. Површинската ЕМГ исто така може да се употреби за детектирање на невромускулни болести, но таа не е доволно прецизна за да биде искористена за разликување помеѓу невропатични и миопатични (мускулни) растројства, како и за утврдување на точната невромускулна болест.

### Интрамускулна ЕМГ

Најдноставниот начин за снимање на интрамускулната ЕМГ е со употреба на монополарни иглести електроди. Ова може да биде тенка жица забодена во мускулот, а како референтна електрода може да се искористи и неинвазивна површинска електрода. Електродите кои се употребуваат во диагностиката се доволно цврсти за пробивање на кожата и се изолирани во целост, освен на врвот. Постојат и концентрични електроди кај кои жицата заедно со изолацијата се наоѓа внатре во металната игла која пак служи како референтна електрода. Поради меѓусебната близост на двета проводника, концентричните електроди даваат послаби сигнали, но се порезистентни на електрични артефакти од соседните ткива. Најфините иглести електроди дозволуваат регистрација на активноста на индивидуални мускулни влакна.

<sup>14</sup>Wikipedia – Electromyography <https://en.wikipedia.org/wiki/Electromyography>

## Поставување на електродите

Пред поставувањето на електродите за ЕМГ, кожата треба соодветно да се подготви, што вообичаено вклучува нејзино чистење со алкохол. Во некои случаи потребно е да бидат отстранети и влакната од локацијата на снимање. Поставувањето на електродите треба да биде на локација помеѓу средината на мускулот и точката во која за него е прицврстена тетивата. Се избегнува поставување на електродите на средината на мускулот затоа што таа е точката со максимална инервација, односно на таа локација се наоѓаат најголем дел од приклучоците на мотоневроните со мускулот. Така, при инервација на мускулот, електричните потенцијали кои во оваа точка пропагираат во мноштво насоки, меѓусебно ќе бидат поништени, со тоа намалувајќи ја амплитудата на ЕМГ сигналот.

## ЕМГ сигнал

Електромиографијата ја мери сумата на акциски потенцијали од моторните единици под електродата. Секој акциски потенцијал на моторна единица (MUAP<sup>15</sup>) во себе ја содржи деполаризацијата на сите мускулни влакна кои се во составот на таа моторна единица, односно се инервирани од еден мотоневрон. Притоа, обликот на акцискиот потенцијал кој се регистрира во ЕМГ зависи од составот на моторната единица, односно бројот на мускулни влакна, типот на мускулните влакна итн.

Изворот на ЕМГ сигналите е биоелектричниот потенцијал на мирување на мускулните клетки кој изнесува околу  $-90\text{ mV}$ . Измерениот сигнал се движи во опсегот од  $50\text{ }\mu\text{V}$ , па сè до  $20$  и  $30\text{ mV}$ , во зависност од мускулот кој се мери. Основната фреквенција на активација на моторните единици изнесува  $7$  -  $20\text{ Hz}$ , во зависност од големината на мускулот.

Основното процесирање на ЕМГ сигналите се состои од насочување на наизменичниот сигнал со употреба на целобранов насочувач, односно пресметувка на неговата апсолутна вредност. Ова овозможува наоѓање на средната мускулна активација преку пресметка на средната вредност на насочениот сигнал, која би била еднаква на  $0$  кај сировиот ЕМГ сигнал.

Со посложено процесирање може да се изврши декомпозиција на ЕМГ сигналот составните акциски потенцијали на моторните единици. MUAP сигналите имаат карактеристични облици кои се разликуваат еден од друг, а зависат од составот на моторната единица и локацијата на електродите. Иако ова е тежок проблем, предложени се низа на методи за негово решавање.

## Мерење на мускулната активација

Во клиничката пракса, една од основните функции на ЕМГ е да се одреди степенот на мускулна активација. Ова вообичаено се прави при максимална волева контракција (MVC<sup>16</sup>) на мускулот кој се тестира. Силата која ја генерира мускулот е високо корелирана со измерената ЕМГ активација. Постојат два начини да се одреди степенот на активација:

- **средна активација** – средната вредност на насочениот ЕМГ сигнал добиен на излез од целобранов насочувач,
- **врвна активација** – максимална вредност на насочениот ЕМГ сигнал,

Притоа, средната апсолутна вредност (ARV<sup>17</sup>) има помал степен на варијација при мерењето на мускулната активност на главните скелетални мускули, споредена со врвната еднонасочна вредност (PRV<sup>18</sup>).

ЕМГ исто така може да се искористи за оценка на измореноста на еден мускул. За таа цел се следи зголемувањето на средната апсолутна вредност на ЕМГ сигналот, зголемувањето на

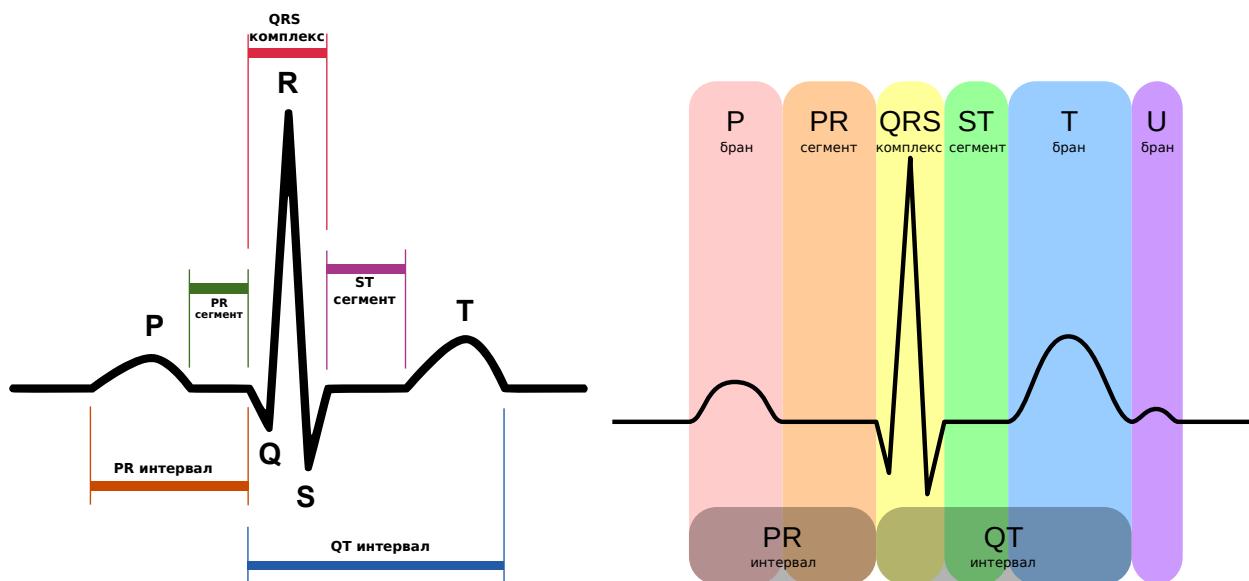
<sup>15</sup>анг. motor unit action potential

<sup>16</sup>анг. Maximal Voluntary Contraction

<sup>17</sup>анг. average rectified value

<sup>18</sup>анг. peak rectified value

траењето на акцискиот потенцијал на мускулот како и поместувањето на целиот сигнал кон пониските фреквенции.



Сл. 2.8: Карактеристични точки и интервали на ЕКГ сигналот.<sup>21</sup>

## 2.3 Електрокардиограм

Електрокардиограмот (ЕКГ)<sup>19</sup> е запис на електричната активност на срцето, Поглавје 1.5., во даден временски период со помош на електроди поставени на телото на човекот. Овие електроди служат за детектирање на малите електрични промени на кожата кои настануваат поради деполаризацијата на срцевиот мускул за време на секое отчукување. Вообично за снимање на ЕКГ се употребуваат 10 електроди поставени на екстремитетите и на градите од човекот. Со нивна употреба се врши мерење на електричниот биопотенцијал на срцето од 12 различни агли и негово снимање во период од вообично 10 s. На овој начин, се забележува вкупната амплитуда и насоченост на деполаризацијата во секој момент од срцевиот циклус<sup>20</sup> прикажан на Сл. 1.21.

### Облик на ЕКГ сигналот

Прогресијата на деполаризација на срцето го дава карактеристичниот облик на ЕКГ сигналот прикажан на Сл. 2.8. За лекарите, обликот на ЕКГ носи големо количство на информации за структурата на срцето и функционирањето на неговиот електричен систем за пренос. Меѓу другото, ЕКГ може да се искористи за мерење на брзината на чукање на срцето, големината и поставеноста на коморите и преткоморите, присуството на општетувања во срцевиот мускул или неговиот електричен систем за пренос, ефектот од лекарствата за срце, како и функцијата на вградените пејсмејкери.

Карактеристични точки во ЕКГ записот се:

- Р бранот – деполаризацијата на преткоморите,
- QRS комплексот – деполаризација на коморите,
- Т бранот – реполаризација на коморите.

Притоа, бранот на реполаризација на преткоморите се совпаѓа со QRS-комплексот па не може да се воочи во ЕКГ сигналот. Дополнително на овие, по завршетокот на Т бранот, може да се појави мал У бран кој одговара на задоцнетата реполаризација на нишките на Пуркинџе, но тој

<sup>19</sup>Wikipedia: Electrocardiography <https://en.wikipedia.org/wiki/Electrocardiography>

Википедија – Електрокардиографија <https://mk.wikipedia.org/wiki/Електрокардиографија>

<sup>20</sup>Wikipedia: Cardiac cycle [https://en.wikipedia.org/wiki/Cardiac\\_cycle](https://en.wikipedia.org/wiki/Cardiac_cycle)

**Табела 2.1:** Типични времетраења на карактеристичните точки, сегменти и интервалите во ЕКГ сигналот.

Регион	Времетраење [ms]
P бран	<80 ms
PR интервал	120 – 200 ms
QRS комплекс	80 – 100 ms
T бран	160 ms
QTc интервал	<440 ms

не е секогаш таму.

Помеѓу карактеристичните точки се дефинирани следните интервали и сегменти:

- **PR интервал** – од почетокот на P бранот до почетокот на QRS комплексот, го рефлектира времето потребно за електричниот импулс да пристигне од синусниот јазол до атриовентрикуларниот (AV) јазол,
- **PR сегмент** – од крајот на P бранот до почетокот на QRS комплексот, го рефлектира времето кога преткоморите се деполаризирани,
- **ST сегмент** – го поврзува QRS комплексот со T бранот и го претставува периодот во кој коморите се деполаризирани,
- **QT интервал** – од почетокот на QRS комплексот до крајот на T бранот, бидејќи варира со брзината на работа на срцето, вообичаено се изразува како QTc корегирано со деление со коренот од RR интервалот.

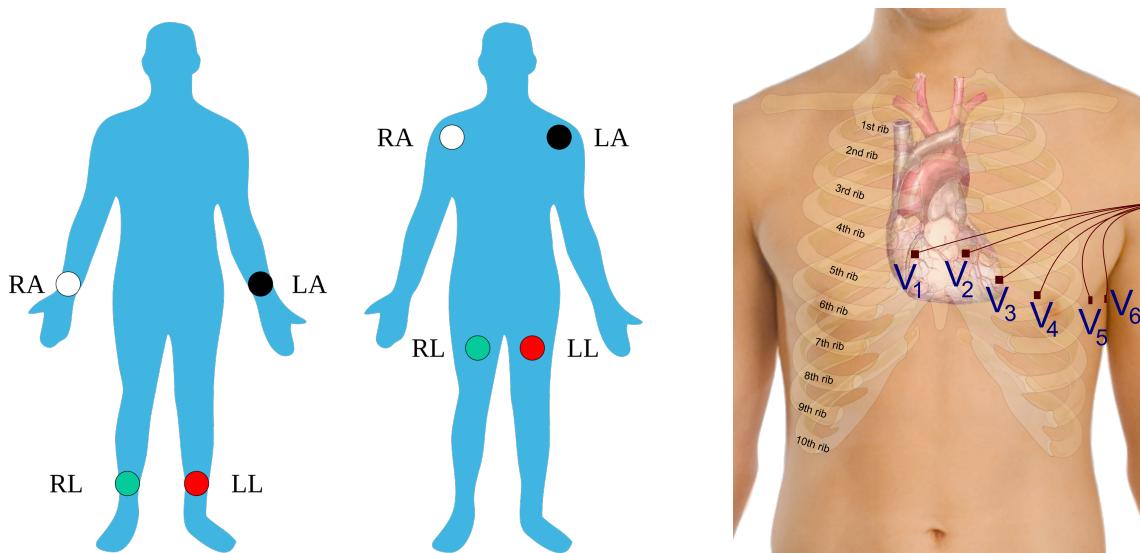
Секоја од карактеристичните точки и интервалите помеѓу нив имаат предвидливо времетраење, опсег на дозволиви амплитуди, и типичен облик. Типичните времетраења се дадени во Табела 2.1. Секоја девијација од нормалата посочува кон можна патологија и поради тоа е од клиничко значење. За поедноставно мерење на ЕКГ сигналот тој се печати на милиметарска хартија на која секој 1 mm одговара на време од 40 ms на x-оската, и 0,1 mV на y-оската.

## Електроди и нивно поставување

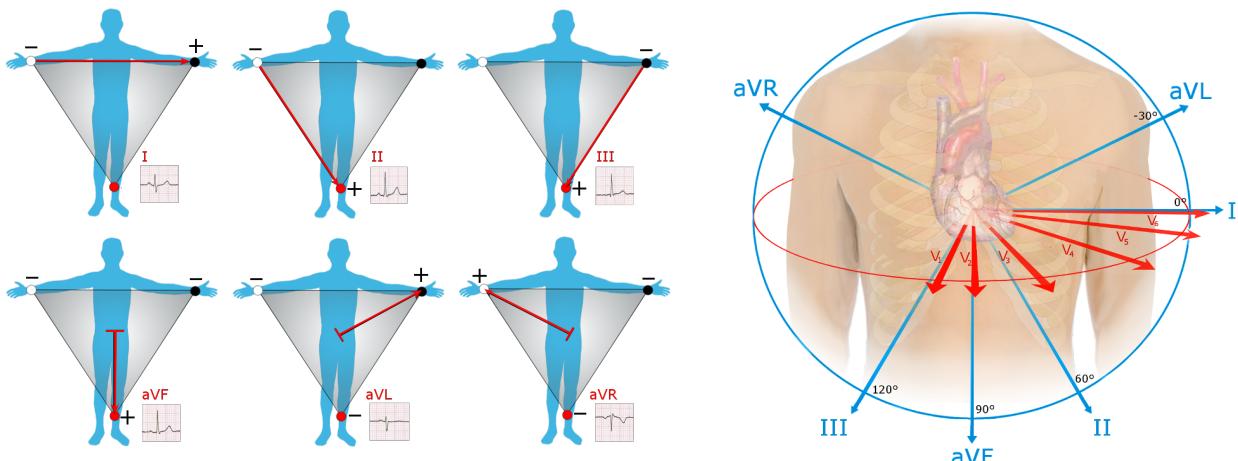
Во професионалните уреди со 12 канали ЕКГ сигналот се мери со 10 електроди. За регистрирање на ЕКГ сигналот се доволни две електроди, меѓутоа во професионалните уреди се прават 12 канали со помош на 9 од 10-те електроди за претставување на активноста на срцето долж 6 вертикални и 6 хоризонтални оски. Притоа, како негативна електрода се употребува или една од 9-те електроди или пак напонот се пресметува во однос на виртуелна електрода добиена како средна вредност на потенцијалот измерен во трите електроди поставени на екстремитетите на човекот. Оваа виртуелна електрода се нарекува и **централен потенцијал на Вилсон**. Десетте електроди се:

- **RA** – на десната рака, не треба да се постави на мускулите,
- **LA** – на истата локација на поставување на RA, но на левата рака,
- **RL** – на левата нога, на долниот крај од листот, треба да се избегнат коските,
- **LL** – на истата локација на RL, но на левата нога,
- **V1** – во четвртиот меѓуребрен простор (помеѓу ребрата 4 и 5) веднаш десно од градната коска (стернумот),

<sup>21</sup>Преведено од Agateller (Anthony Atkielski), converted to svg by atom. - SinusRhythmLabels.png, Public Domain, <https://commons.wikimedia.org/w/index.php?curid=1560893> и од Derivative: Hazmat2Original: Hank van Helvete - This file was derived from: EKG Komplex.svg, CC BY-SA 3.0, <https://commons.wikimedia.org/w/index.php?curid=31447187>



Сл. 2.9: Поставување на електродите за снимање на ЕКГ сигналот.<sup>22</sup>



Сл. 2.10: Ориентација на напоните кои ги регистрира 12-каналниот ЕКГ уред.<sup>23</sup>

- **V2** – како V1 но веднаш лево од градната коска,
- **V3** – помеѓу V2 и V4,
- **V4** – во петиот меѓуребрен простор (помеѓу ребрата 5 и 6) долж средната линија на клучната коска,
- **V5** – хоризонтално до V4, долж линијата која поминува на половина од растојанието помеѓу средината и крајот на клучната коска,
- **V6** – хоризонтално до V4 и V5, долж страничната линија на телото.

Сите овие се користат за добивање на ЕКГ сигналот долж една оска, освен LL која се употребува за заземјување на уредот за мерење. Ова е неопходно за поништување на индуцираните електромагнетни шумови од напонот на градската мрежа. Поставувањето на десетте електроди на телото е прикажано на Сл. 2.9.

<sup>22</sup>Преведено од Original:MoodyGrooveVector:TwispThis SVG diagram includes elements that have been taken or adapted from this diagram: Wikicouple.svg. - Own work based on: Limb Leads.jpg by MoodyGroove., Public Domain, <https://commons.wikimedia.org/w/index.php?curid=3405434> и од Mikael Häggström - Own work, CC0, <https://commons.wikimedia.org/w/index.php?curid=20064293>

<sup>23</sup>Превземено од Npatchett - Own work, CC BY-SA 4.0, <https://commons.wikimedia.org/w/index.php?curid=39235282> и од Npatchett - Own work, CC BY-SA 4.0, <https://commons.wikimedia.org/w/index.php?curid=39235260>

## Снимање на ЕКГ сигналот

Дванаесетте канали снимени од професионалните ЕКГ уреди се поделени во три типови: 3 напони на екстремитетите, 3 аугментирани напони на екстремитетите, и 6 градни напони. Шестте канали поврзани со екстремитетите се распоредени во вертикалната рамнина, а шестте градни канали се распоредени во хоризонталната рамнина.

Добивањето на дванаесетте напони е прикажано на Сл. 2.10 Трите напони на екстремитетите се означени како I, II и III. Тие се пресметуваат како:

$$I = LA - RA, \quad (2.1)$$

$$II = LL - RA, \quad (2.2)$$

$$III = LL - LA. \quad (2.3)$$

Аугментираните напони на екстремитетите се аналогни на напоните на екстремитетите, со таа разлика што како негативна електрода се зема централниот потенцијал на Голдберг. Голдберговиот централен потенцијал се добива како средна вредност од потенцијалите на две од електродите на екстремитетите. Сите аугментирани напони можат да се изразат и преку Вилсоновиот потенцијал  $V_W$  според:

$$aVR = RA - \frac{1}{2}(LA + LL) = \frac{3}{2}(RA - V_W), \quad (2.4)$$

$$aVL = LA - \frac{1}{2}(RA + LL) = \frac{3}{2}(LA - V_W), \quad (2.5)$$

$$aVF = LL - \frac{1}{2}(RA + LA) = \frac{3}{2}(LL - V_W), \quad (2.6)$$

каде:

$$V_W = \frac{1}{3}(RA + LA + LL). \quad (2.7)$$

Шесте градни напони лежат во хоризонталната рамнина која е нормална на вертикалната рамнина дефинирана од напоните на екстремитетите и аугментираните напони на екстремитетите. Тие се означени како и градните електроди, односно  $V_1 - V_6$ , а се пресметуваат во однос на  $V_W$ .



Сл. 2.11: Нормална мозочна активност наспроти епилептичен напад.<sup>25</sup>

## 2.4 Електроенцефалограм

Електроенцефалографијата (ЕЕГ)<sup>24</sup> е електрофизиолошка метода за снимање на електричната активност на мозокот, Поглавје 1.6. Вообичаено се изведува неинвазивно со поставување на електроди на скалпот, но постои и инвазивна верзија во која електродите се поставуваат директно на мозочната кора – **електрокортикографија (ЕКоГ)**. ЕЕГ ги мери промените во биоелектричниот потенцијал предизвикани од електричната активност на невроните, односно од јонските струи предизвикани од акциските потенцијали, види Поглавје 1.3.

### Клиничка апликација

Во клиничка смисла, ЕЕГ претставува континуирано мониторирање на спонтаната мозочна активност во одреден временски период, вообичаено 20 – 30 минути. Дијагнозата потоа се базира на спектралната содржина на електричните бранови, односно од типот на бранови кои можат да се забележат во сигналот.

ЕЕГ најчесто се користи за дијагностицирање на **епилепсија**, која предизвикува абнормални флуктуации во мозочните бранови, Сл. 2.11. Станува збор за нарушенa функција на инхибирачките неврони во мозокот па настанува позитивна повратна врска која доведува до неконтролирана деполаризација на голем број на неврони. Оваа масовна деполаризација се нарекува епилептичен напад или иктал. Во ЕЕГ анализата вообичаено се следи интерикалниот период, во кој можат да се забележат врвови во сигналот кои не се доволни за започнување на напад.

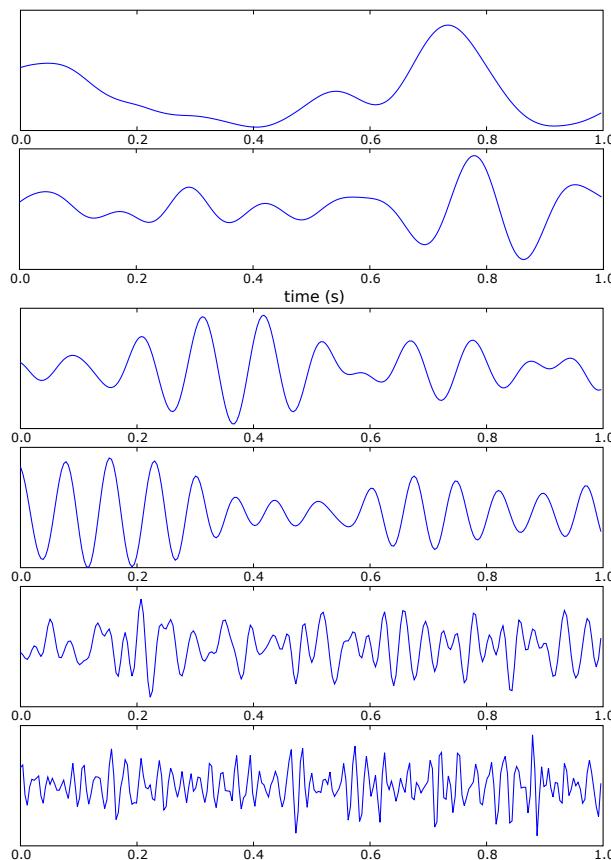
Во клиничката пракса, ЕЕГ исто така се употребува за дијагноза на пореметувања во спиењето, кома и мозочна смрт, како и за следење на степенот на анестезија. Во поново време, ЕЕГ се употребуваат за утврдување на пореметувања со недостаток на внимание и хиперактивност (АДХД)<sup>26</sup>. Порано, ЕЕГ се употребувала и за првична дијагноза на тумори и мозочни удари, но денес таа е заменета со методите со повисока резолуција како магнетната резонанса (МР) и компјутерската томографија (КТ).

Во терапевтски цели, ЕЕГ се употребува како повратна врска во **биофидбекот**. Со нејзина помош,

<sup>24</sup>Wikipedia – Electroencephalography <https://en.wikipedia.org/wiki/Electroencephalography>

<sup>25</sup>CC BY-SA 2.0, <https://commons.wikimedia.org/w/index.php?curid=845554>

<sup>26</sup>анг. *Attention deficit hyperactivity disorder (ADHD)*.



**Сл. 2.12:** Ритмичка активност на мозокот снимена со ЕЕГ во петте опсези (од горе надолу): делта, тета, алфа, му, бета и гама.<sup>27</sup>

може да се вежба достигнување на состојба на смиреност и фокусираност. На овој начин може да се третираат пореметувања како АДХД, но и да се поттикнат врвни перформанси кај спортисти и музичари.

### ЕЕГ опсези

ЕЕГ активноста вообичаено се опишува во однос на ритмичка активност и транзиенти. Традиционално ритмичната активност се дели во фреквенциски подопсези прикажани на Сл. 3.23. Иако до одреден степен овие опсези се предмет на договор, тие опишуваат мозочна активност која има одредена дистрибуција во мозокот или пак е поврзана со одредена активност. Поголемиот дел од мозочната активност во клиничките испитувања се наоѓа во опсегот 1 – 20 Hz. ЕЕГ активноста под и над овој опсег има значење во научните истражувања.

**Делта бранови** се дефинирани во опсегот до 4 Hz и претставуваат бранови со висока амплитуда кои се јавуваат фронтално кај возрасните. Типични се за длабок сон, кај бебиња и некои задачи кои бараат континуирано внимание.

**Тета бранови** во опсегот 4 – 8 Hz се бранови кои се јавуваат при поспаност, неактивност и при потиснување на одговор или акција.

<sup>27</sup>By Hugo Gamboa - Own work, CC BY-SA 3.0, <https://commons.wikimedia.org/w/index.php?curid=473245>

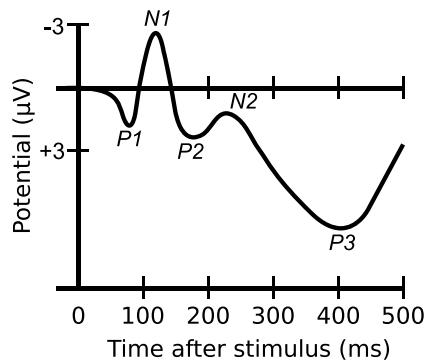
By Hugo Gamboa - This is raw eeg. The signal was acquired in the Oz position processed with scipy and saved with matplotlib., CC BY-SA 3.0, <https://commons.wikimedia.org/w/index.php?curid=473247>

By Hugo Gamboa - Own work, CC BY-SA 3.0, <https://commons.wikimedia.org/w/index.php?curid=473243>

CC BY-SA 3.0, <https://commons.wikimedia.org/w/index.php?curid=473209>

By Hugo Gamboa - Own work, CC BY-SA 3.0, <https://commons.wikimedia.org/w/index.php?curid=473244>

By Hugo Gambo - Own work, CC BY-SA 3.0, <https://commons.wikimedia.org/w/index.php?curid=473246>



**Сл. 2.13:** Типичен облик на ЕРП кој вклучува N1 и P3 компоненти. Забележете дека  $y$ -оската е превртена што е неретка практика во истражувањата на ЕРП.<sup>32</sup>

**Алфа бранови** се јавуваат во опсегот 8 – 14 Hz. Тие се најизразени во задните делови на мозокот и на доминантната хемисфера и го отсликуваат затворањето на очите, опуштеност, како и инхибиција на активност. Во патологијата се индикативни за кома. Историски биле откриени први па затоа и се наречени алфа бранови.

**Му бранови** со опсег од 8 – 12 Hz се бранови слични на алфа брановите, но се изразени во сензоромоторниот кортекс. Затоа се наречени и сензоромоторен ритам. Тие се индикативни за неактивност на моторните неврони и поради тоа се значајни во **интерфејсите мозок-компјутер (БКИ)**.<sup>28</sup>

**Бета бранови** во опсегот 14 – 31 Hz се бранови со мала амплитуда со симетрична фронтална дистрибуција. Тие се индикативни за активно размислување, фокусирање, висока растревоженост, стрес и анксиозност.

**Гама бранови** се бранови над 32 Hz кои се најизразени во соматосензорниот кортекс и го отсликуваат меѓусетилното процесирање, на пр. вид и слух, како и при препознавање на објекти од кратко-трајната меморија. Индикативни се за „аха“ моментот.

### Евоцирани потенцијали

Еден начин на испитување на мозочната активност со помош на ЕЕГ е преку анализа на **евоцираните потенцијали (ЕП)**.<sup>29</sup> Таа се заснова на усреднување на мозочните бранови врзано со презентирање на некој вид на побуда на сетилата, како визуелна, аудиторна или соматосензорна.<sup>30</sup> Кога се употребува покомплексен стимул, кој бара поголемо ниво на процесирање од мозокот, станува збор за **потенцијали поврзани со настани (ЕРП)**.<sup>31</sup> Усреднувањето е нужно поради малиот однос на сигнал-шум кој го имаат ЕЕГ сигналите, поради што е нужна и повеќекратна презентација на стимулот.

Обликот на ЕРП, прикажан на Сл. 2.13. се состои од низа на позитивни и негативни напонски врвови кои се нарекуваат и **компоненти**. Вообичаено различните ЕРП компоненти се именуваат според поларитетот на напонскиот врв и времето на неговото појавување кое некогаш е изразено и во стотици милисекунди. Така, **N1 (N100)** означува негативна компонента која се јавува на 100 ms од стимулот, а **P3 (P300)** означува позитивна компонента на 300 ms. Времето на појавување не е егзактно и вообичаено се однесува на опсег на вредности, на пр. P3 може да се јави помеѓу 250 – 700 ms од побудата.

<sup>28</sup>анг. *Brain-computer interface*.

<sup>29</sup>анг. *Evoked potentials (EPs)*.

<sup>30</sup>Wikipedia – Event-related potential [https://en.wikipedia.org/wiki/Event-related\\_potential](https://en.wikipedia.org/wiki/Event-related_potential).

<sup>31</sup>анг. *Event related potentials (ERPs)*.

<sup>32</sup>By Original:ChomsVector:Mononomic - Own work based on: Constudevent.gif, CC BY-SA 3.0, <https://commons.wikimedia.org/w/index.php?curid=5543904>

Во истражувањата вообичаено се следат неколку добро востановени компоненти. Така, N1 компонентата, која вообичаено е дел од N1-P2 комплекс, се јавува при непредвиден аудиторен, визуелен или соматосензорен стимул.<sup>33</sup> На пр., при презентација на регуларен звучен такт, N1 ќе се јави ако еден од ударите се отстрани од секвенцата. Со тоа може да се утврди дека мозокот има способност за следење на ритам.

P3 компонентата, која се дели на P3a и P3b компоненти, вообичаено се јавува при носење на одлуки, мисловна реакција, неовообичаени побуди, како и побуди со апстрактно значење за испитаникот.<sup>34</sup> Така, P3 се јавува при приказ на бројки, но не и светлина.

ЕРП се користат во когнитивните науки, когнитивната психология и психофизиолошките истражувања. ЕРП се употребуваат кај некои видови на ЕЕГ за БКИ.

## Интерфејс мозок-машина (БКИ)

ЕЕГ сигналите можат да се искористат за остварување на БКИ. Притоа, вообичаени се два пристапа кои се базираат или на детекција на му ритамот или на анализата на ЕРП.

Во БКИ базирани на му ритамот се анализира неговата распределба долж моторниот кортекс. Од неа може да се заклучи кои центри моторниот кортекс се неактивни, а кои се активни. Преку тоа може да се заклучи која активност ја замислува испитаникот, на пр. затворање на тупаница на левата рака, и преку тоа да се изврши кодирање на контролата на еден електронски уред. Важно е да се спомене дека поради постојењето на огледалните неврони, доволно е човек да замисли некоја акција за да го активира соодветниот центар во моторниот кортекс – вистинско движење не е потребно.

Кај БКИ базирано на ЕРП, вообичаено се работи со визуелни стимули. На пример, со анализа на P3 ЕРП компонентата при трепкања на буквите од азбуката може да се одреди на која буква се фокусира испитаникот. На тој начин е возможно да се оствари комуникација со луѓе парализи, кај кои не постои контрола врз скелеталните мускули. Точноста на ваквиот БКИ изнесува 95% а протокот кој може да се оствари е околу 4 знаци во минута.

## Предности и недостатоци на ЕЕГ

Споредена со останатите техники за набљудување на работата на мозокот како функционалната магнетна резонанса (ФМР) или томографијата со емисија на позитрони (ПЕТ)<sup>35</sup>, ЕЕГ има низа на предности:

- **ниската цена** – цената на хардверот потребен за ЕЕГ е од редот на 5.000 – 10.000 € е значително пониска од онаа за ФМР на пример која зависи од силата на електромагнетот и се движи во опсег од 1,5 – 7 милиони €,
- **преносливост** – поради малите димензии и малата тежина, ЕЕГ може да се употреби на терен или во амбулантските возила, наспроти ФМР чиј магнет тежи 1 t,
- **висока временска резолуција** – ЕЕГ нуди набљудување на мозочната активност со резолуција од милисекунди, додека ФМР се базиран на контрастот кој зависи од нивото на кислород во крвта (БОЛД)<sup>36</sup> сигналот кој има доцнење од 2 – 3 s од мозочната активност,
- **толерантност на поместување на испитаникот** – за разлика од останатите техники за снимање на мозочната активност, ова е особено важно кај снимањето на децата,
- **безшумна** – дозволува работа со аудиторни стимули,
- **не предизвикува клаустрофобија**,

<sup>33</sup>Wikipedia – N100 <https://en.wikipedia.org/wiki/N100>.

<sup>34</sup>Wikipedia – P300 (neuroscience) [https://en.wikipedia.org/wiki/P300\\_\(neuroscience\)](https://en.wikipedia.org/wiki/P300_(neuroscience)).

<sup>35</sup>анг. *Positron emission tomography*.

<sup>36</sup>анг. *Blood-oxygen-level dependent (BOLD)*.

- не употребува високи магнетни полиња – што е важно кај луѓето со метални импланти,
- не употребува високи магнетни полиња – за разлика од фМР, што е важно кај луѓето со метални импланти,
- не употребува радијација – за разлика од ПЕТ,
- **неинвазивност.**

Дополнително ЕЕГ може да се употреби за:

- **детекција на скриено процесирање** – активност на мозокот која не е поврзана со физичка активност, ова може да се употреби на пример за детекција на лажење,
- работа и комуникација со параплегичари,
- следење на чекорите на процесирање во мозокот – а не само крајниот резултат,
- следење на развојот на мозокот низ животот.

Недостатоците на ЕЕГ вклучуваат:

- **мала просторна резолуција** – површинската резолуција зависи од бројот на електроди, но длабинската резолуција е неодредена; таа не е споредлива со волуменската резолуција на фМР на пример,
- **мала осетливост на длабинска активност,**
- **мал однос сигнал-шум** – малата амплитуда на мозочните бранови изискува употреба на напредни методи за постпроцесирање, како и голем број на повторувања на мерењата,
- **појава на артефакти** – предизвикани од мускулна активност, на пр. трепкање, но и работата на срцето,
- **долго време за поставување.**

## Хардвер и поставување

Вообичаената амплитуда на ЕЕГ сигналите изнесува од  $10 \mu\text{V}$  до  $100 \mu\text{V}$ . Со инвазивно поставување на електродите може да се добијат амплитуди во опсегот  $10 - 20 \text{ mV}$  во ЕКоГ. Поради ова, нужна е употребата на диференцијални засилувачи со вообичаено засилување од 60 до  $100 \text{ dB}$ , односно  $1.000\times - 100.000\times$ .

Фреквенцијата на семплирање кај клиничките уреди е во опсегот  $256 - 512 \text{ Hz}$ , но за истражувачки цели се употребуваат и фреквенции на семплирање од  $20 \text{ kHz}$ .

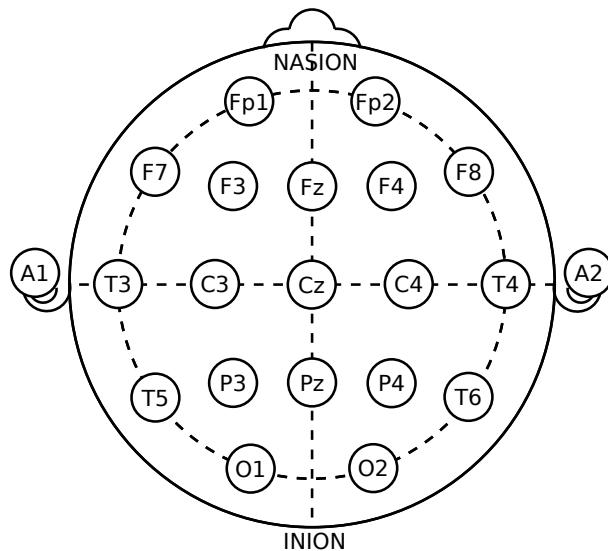
Бројот на електроди во клиничките системи изнесува 21 од кои 19 се за снимање на мозочните бранови, а 2 се референтни. ЕЕГ системите за биофидбек вообичаено имаат помалку електроди, а некогаш имаат само 2 електроди кои се доволни за снимање на 1 канал. Од друга страна, за истражување се користат ЕЕГ системи со дури 256 електроди.

Распоредот на електродите на скалпот е дефиниран со меѓународниот  $10 - 20$  систем<sup>38</sup>, прикажан на Сл. 2.14. Тој го добил името од таму што растојанието помеѓу електродите изнесува 10 или 20 % од целовкупната должина на черепот во двете оски. Секоја од локациите е означена со буква и бројка. Буквата го означува делот од мозокот, **Поглавје. 1.6**, над кој се наоѓа електродата: префронтален (Fp), фронтален (F), темпорален (T), париетален (P), окципитален (O), и централен (C). Додека пак, бројката ја означува хемисферата – непарните броеви се однесуваат на левата хемисфера, а парните броеви се однесуваат на десната хемисфера; и растојанието од средината. Алтернативно, со “z” се означува средната линија на главата.

Според изборот на референтните електроди разликуваме неколку типови на **монтажи** во ЕЕГ:

<sup>37</sup>By ████124 (talk) - Own work, Public Domain, <https://commons.wikimedia.org/w/index.php?curid=10489987>

<sup>38</sup>Wikipedia – 10 – 20 system (EEG) [https://en.wikipedia.org/wiki/10-20\\_system\\_\(EEG\)](https://en.wikipedia.org/wiki/10-20_system_(EEG))



**Сл. 2.14:** Поставеност на 21 електрод според меѓународниот 10 – 20 стандард.<sup>37</sup>

- **секвенцијална монтажа** – секој канал се добива како разлика од две соседни електроди,
- **референтна монтажа** – сите канали се пресметуваат спрема референтната електрода, таа може да биде една од централните електроди ако е пожелано да се набљудува разликата во мозочната активност помеѓу хемисферите, но најчесто се пресметува како средна вредност од електродите поставени на ушите – А1 и А2 во 10 – 20 системот,
- **средна референтна монтажа** – како референтна електрода се зема средната вредност од сите електроди,
- **Лапласова монтажа** – како референтна за секоја електрода се зема тежинската средна вредност од околните електроди.

## Поглавје 3

# Процесирање на биомедицинските сигнали

Со преминот на биомедицинските сигнали во [дигитален домен](#), се отвораат вратите кон примена на различните техники на дигиталната обработка на сигналите за нивната обработка.

Поради философијата на отвореност и соработка која е основа зад движењето за слободен софтвер, денес сè повеќе инженери и научници ја засноваат својата работа на платформи базирани на слободен софтвер. Повеќе за историјата, предностите и продорот на слободниот софтвер во инженерската и научно истражувачката практика може да прочитате во [Додатокот А](#).

Во духот на философијата зад слободниот софтвер, а следејќи ги светските инженерски и научни трендови, вежбите во предметот [Биомедицинска електроника](#) во целост ќе бидат изработени со слободен и отворен софтвер, поточно во слободниот програмскиот јазик [Пајтон](#). За вовед во екосистемот на Питон направен за инженерска и научна работа и за основните системски поставувања погледнете во [Додатокот В](#).

### 3.1 Основи на Фурьеовата анализа

Фреквенцискиот спектар на биомедицинските сигнали е клучен во нивната анализа и обработка во биомедицинските електронски системи. Со негова помош можеме да ги видиме карактеристиките на сигналот кои не се јасно видливи во неговиот временски облик. За пресметка на спектарот на биомедицинските сигнали ќе се послужиме со [Фурьеовата трансформација](#) и нејзината временски определена форма [Фурьеова трансформација на временски отсечоци \(ФТВО\)](#)<sup>1</sup>. Ќе видиме како ФТВО води до една прегледна форма за претставување на временско/спектрални особини на сигналите – [спектрограмот](#).

За потребите на овој курс, во ова поглавје ќе биде направен краток преглед на основите на Фурьеовата анализа. Притоа ќе се задржиме само до особините и деталите кои се релевантни за дигиталното процесирање на биомедицинските сигнали. За подетален преглед погледнете ја книгата [Богданов \(1997\)](#), по која се работи на предметот [Основи на дигитално процесирање на сигнали](#) на додипломски студии на [ФЕИТ](#).

<sup>1</sup>Short-Time Fourier Transform (STFT).

<sup>2</sup>Wikipedia – Joseph Fourier [https://en.wikipedia.org/wiki/Joseph\\_Fourier](https://en.wikipedia.org/wiki/Joseph_Fourier). Слика од Louis-Léopold Boilly - <https://www.gettyimages.com.au/license/169251384>, Public Domain, <https://commons.wikimedia.org/w/index.php?curid=3308441>



§ Дополнително. Францускиот математичар и физичар **Жан-Батис Жозеф Фурие**<sup>2</sup> (1768–1830) анализирајќи го ширењето на топлината, во 1822 за првпат постулирал дека која било функција може да биде претставена преку серија на синусоиди. Тој бил син на кројач и останал без својот татко на 9 годишна возраст. За време на француската револуција бил член на локалниот револуционерен комитет, за што и бил во затвор. Го придржувал Наполеон Бонапарта во неговиот поход во Египет во 1798, а по враќањето во Франција во 1801 станува претседател на општината Изер во Гренобл, каде започнува да прави експерименти за топлината.

## Фуриев интеграл

Фуриевата трансформација (**ФТ**) на сигналот  $x(t)$  е дефинирана со интегралот:

$$X(\omega) = \int_{-\infty}^{\infty} x(t)e^{-j\omega t} dt, \quad \omega \in (-\infty, \infty), \quad (3.1)$$

каде со  $\omega = 2\pi f$  е означена кружната фреквенција.

Инверзната Фуриева трансформација (**ИФТ**) е дефинирана со интегралот:

$$x(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} X(\omega)e^{j\omega t} d\omega, \quad t \in (-\infty, \infty). \quad (3.2)$$

$X(\omega)$  се нарекува **фреквенциски спектар** на сигналот  $x(t)$  и претставува комплексна функција од  $\omega$  која може да се запише како:

$$X(\omega) = |X(\omega)|e^{j\angle X(\omega)} = A(\omega)e^{j\varphi(\omega)}, \quad (3.3)$$

каде  $A(\omega)$  претставува **амплитуден спектар**, а  $\varphi(\omega)$  **фазен спектар** на сигналот  $x(t)$ . Притоа, ако сигналот е реална функција од  $t$  тогаш амплитудниот спектар е парна функција од  $\omega$ , а фазниот е непарна функција од  $t$ . Бидејќи интеграл од непарна функција во интервал симетричен околу координатниот почеток е 0, тогаш од (3.2), користејќи ја **Ојлеровата формула**, добиваме:

$$x(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} A(\omega)e^{j\phi(\omega)}e^{j\omega t} d\omega, \quad (3.4)$$

$$= \frac{1}{2\pi} \int_{-\infty}^{\infty} A(\omega) [\cos(\omega t + \phi(\omega)) + j \sin(\omega t + \phi(\omega))] d\omega, \quad (3.5)$$

$$= \frac{1}{\pi} \int_0^{\infty} A(\omega) \cos(\omega t + \phi(\omega)) d\omega. \quad (3.6)$$

Основните својства на Фуриевата трансформација се дадени во **Табелата 3.1**.

Сигналот  $x(t)$  и неговата Фуриева трансформација  $X(\omega)$  чинат **Фуриев пар** што се означува како  $x(t) \leftrightarrow X(\omega)$ . За Фуриевата трансформација се користи и симболот  $\mathcal{F}$ , односно пишуваме  $\mathcal{F}\{x(t)\} = X(\omega)$ , или за ИФТ  $\mathcal{F}^{-1}\{X(\omega)\} = x(t)$ .

Во **Табелата 3.2** се дадени некои позначајни Фуриеви парови.

**Табела 3.1:** Позначајни својства на Фуриеовата трансформација.

Својство	Временски домен	Фуриеов домен
Линеарност	$ax(t) + bg(t)$	$aX(\omega) + bG(\omega)$
Поместување	$x(t - t_0)$	$e^{-j t_0 \omega} X(\omega)$
Модулација	$e^{-j \omega_0 t} x(t)$	$X(\omega - \omega_0)$
Конволуција	$x(t) * g(t) = \int_{-\infty}^{\infty} x(\tau)g(t - \tau) d\tau$	$X(\omega)G(\omega)$
Мултипликација	$x(n)g(n)$	$\frac{1}{2\pi} X(\omega) * G(\omega) = \frac{1}{2\pi} \int_{-\infty}^{\infty} X(\Omega)G(\omega - \Omega) d\Omega$

**Табела 3.2:** Позначајни Фуриеви парови.

$x(t)$	$X(\omega)$
$\delta(t)$	1
1	$2\pi\delta(\omega)$
$\Pi(t) = \begin{cases} 1 &  t  < T \\ 0 &  t  > T \end{cases}$	$2T \frac{\sin(T\omega)}{T\omega}$
$\frac{\sin(\Omega t)}{\pi t}$	$\Pi(\omega) = \begin{cases} 1 &  \omega  < \Omega \\ 0 &  \omega  > \Omega \end{cases}$
$\cos(\omega_0 t)$	$\pi [\delta(\omega + \omega_0) + \delta(\omega - \omega_0)]$
$\sin(\omega_0 t)$	$j\pi [\delta(\omega + \omega_0) - \delta(\omega - \omega_0)]$

Енергијата на сигналот може да се пресмета преку неговиот спектар со употреба на теоремата на Парсевал<sup>3</sup>:

$$E = \int_{-\infty}^{\infty} |x(t)|^2 dt = \frac{1}{2\pi} \int_{-\infty}^{\infty} |X(\omega)|^2 d\omega = \int_{-\infty}^{\infty} |X(2\pi ft)|^2 df \quad (3.7)$$

\* Важно! Во Табелата 3.2 забележете дека Фуриеовата трансформација на 1 е  $2\pi\delta(\omega)$ , додека на  $\cos(\omega_0 t)$  е  $\pi [\delta(\omega + \omega_0) + \delta(\omega - \omega_0)]$ . Ова е во склад со теоремата на Парсевал дека енергијата во спектарот на косинусот помножена со коефициент  $1/2\pi$  треба да е еднаква со онаа во временскиот сигнал. Поради тоа двата огледални Диракови импулси имаат двојно помала амплитуда, што не е случај кај константниот сигнал за кој имаме единствен Дираков импулс на фреквенција 0.

## Фуриев ред

Ако функцијата  $x(t)$  е периодична со периода  $T$ , односно основна фреквенција  $f_0 = 1/T$ , тогаш важи:

$$x(t) \equiv x(t + T), \quad \forall t. \quad (3.8)$$

Ако со  $\hat{x}(t)$  го означиме сегментот на  $x(t)$  во основниот интервал  $t \in [-T/2, T/2]$ :

$$\hat{x}(t) = x(t) \cdot \Pi(t), \quad (3.9)$$

<sup>3</sup>Wikipedia – Parseval's Theorem [https://en.wikipedia.org/wiki/Parseval%27s\\_theorem](https://en.wikipedia.org/wiki/Parseval%27s_theorem)

каде:

$$\Pi(t) = \begin{cases} 1, & |t| \leq \frac{T}{2}, \\ 0, & |t| > \frac{T}{2}. \end{cases}, \quad (3.10)$$

тогаш можеме да ја изразиме  $x(t)$  како сума од вакви функции  $\hat{x}(t)$  поместени за мултили од  $T$ :

$$x(t) = \sum_{r=-\infty}^{\infty} \hat{x}(t - rT). \quad (3.11)$$

Во тој случај во Фуриев домен, функцијата наместо да биде континуирана по кружната фреквенција  $\omega$ , таа ќе биде еднаква на сума од комплексни синусоиди која се нарекува и **Фуриев ред**:

$$x(t) = \sum_{k=-\infty}^{\infty} c_k e^{jk\frac{2\pi}{T}t} = \sum_{k=-\infty}^{\infty} c_k e^{jk\omega_0 t}, \quad (3.12)$$

каде  $\omega_0$  е основната кружна фреквенција на сигналот, дадена со:

$$\omega_0 = 2\pi f_0 = \frac{2\pi}{T}. \quad (3.13)$$

Ова значи дека фреквенциите на комплексните синусоиди претставуваат мултили од основната кружна фреквенција на сигналот  $\omega_0$ . Овие мултили уште се нарекуваат и **хармоници** или **виши хармоници**, а  $f_0$  и  $\omega_0$  се нарекува **основен хармоник**. Притоа коефициентите  $c_k$  кои претставуваат амплитуди на комплексните синусоиди можат да се добијат преку:

$$c_k = \frac{1}{T} \int_{-T/2}^{T/2} x(t) e^{-jk\frac{2\pi}{T}t} dt. \quad (3.14)$$

Фреквенцискиот спектар на ваков периодичен сигнал  $x(t)$  претставува сума од поместени еквидистантни Диракови импулси  $\delta(\omega)$ :

$$X(\omega) = 2\pi \sum_{k=-\infty}^{\infty} c_k \delta(\omega - k\omega_0). \quad (3.15)$$

### $\mathcal{Z}$ трансформација

За да видиме како Фуриевата анализа може да се примени врз дискретни сигнали ќе ја воведеме  $\mathcal{Z}$  трансформацијата (Rabiner and Schafer, 1978) дефинирана со:

$$X(z) = \sum_{n=-\infty}^{\infty} x(n) z^{-n}, \quad (3.16)$$

каде  $x(n)$  е дискретна верзија од континуираната функција  $x(t)$ :

$$x(n) = x(nT_s), \quad (3.17)$$

$$T_s = \frac{1}{f_s}. \quad (3.18)$$

Тука,  $f_s$  е фреквенцијата на семплирање, а  $T_s$  е периодата на семплирање.

Инверзната  $\mathcal{Z}$  трансформација е дефинирана со:

$$x(n) = \frac{1}{2\pi j} \oint_C X(z) z^{n-1} dz, \quad (3.19)$$

каде  $C$  е затворена контура во  $\mathcal{Z}$  рамнината која го вклучува центарот, а се наоѓа во пределот на конвергенција на трансформацијата  $z \in (R_1, R_2)$ . Позначајни особини на  $\mathcal{Z}$  трансформацијата се дадени во **Табелата 3.3**, а позначајни  $\mathcal{Z}$  трансформации во **Табелата 3.4**.

**Табела 3.3:** Позначајни својства на  $\mathcal{Z}$  трансформација.

Својство	Временски домен	$\mathcal{Z}$ домен
Линеарност	$ax_1(t) + bx_2(t)$	$aX_1(z) + bX_2(z)$
Поместување	$x(n - n_0)$	$z^{n_0}X(z)$
Конволуција	$x(n) * h(n) = \sum_{k=-\infty}^{\infty} x(k)h(n-k)$	$X(z)H(z)$
Мултипликација	$x(n)\Pi(n)$	$\frac{1}{2\pi j} \oint_C X(\nu)W(z/\nu)\nu^{-1}d\nu$

**Табела 3.4:** Позначајни  $\mathcal{Z}$  парови.

$x(n)$	$X(z)$
$\delta(n - n_0)$	$z^{-n_0}$
$\Pi(n) = \begin{cases} 1, & 0 \geq n < N, \\ 0, & \text{поинаку.} \end{cases}$	$\sum_{n=0}^{N-1} z^{-n} = \frac{1 - z^{-N}}{1 - z^{-1}}$
$a^n u(n)$	$\frac{1}{1 - z^{-1}}, \quad  a  <  z $

### Фурьеова трансформација на дискретен сигнал

Фурьеовата трансформација на дискретен сигнал може да се добие од равенката за  $\mathcal{Z}$  трансформацијата (3.16) со замената на јадрото  $z = e^{-j\omega T_s}$ , што се поедноставува на  $z = e^{-j\omega}$  за  $T_s = 1$ :

$$X(\omega) = \sum_{n=-\infty}^{\infty} x(n)e^{-j\omega n}, \quad (3.20)$$

$$x(n) = \frac{1}{2\pi} \int_{-\pi}^{\pi} X(\omega)e^{j\omega n} d\omega. \quad (3.21)$$

Во овој случај, ја ограничуваме  $\mathcal{Z}$  трансформацијата на единечната кружница во  $z$ -рамнината опишана со  $e^{-j\omega} = \cos(\omega) + j\sin(\omega)$ , а дигиталната кружна фреквенција  $\omega$  има интерпретација на агол во оваа рамнина. Со други зборови Фурьеовата трансформација на дискретниот сигнал претставува специјален случај на  $\mathcal{Z}$  трансформацијата.

Бидејќи новото јадро  $e^{-j\omega}$  е периодична функција со периода  $2\pi$ , следува дека фреквенцискиот спектар на дискретните сигнали е исто така периодичен со  $2\pi$  во однос на кружната фреквенција  $\omega$ . За произвилно  $T_s$ , спектарот има периода  $2\pi/T_s$  во однос на кружната фреквенција  $\omega$ , односно периодата е  $1/T_s = f_s$  во однос на фреквенцијата  $f$ .

\* Важно! Фурьеовата трансформација  $X(\omega)$  на дискретниот сигнал  $x(n)$  е сеуште континуирана функција од  $\omega$ !

### Дискретна Фурьеова трансформација

Како што беше случај во аналоген домен, ако еден дискретен сигнал е периодичен со периода  $N$ , односно:

$$\tilde{x}(n) = \tilde{x}(n + N), \quad -\infty < n < \infty, \quad (3.22)$$

тогаш  $\tilde{x}(n)$  во Фурьеов домен ќе биде претставена како дискретна сума од синусоиди:

$$\tilde{X}(k) = \sum_{n=0}^{N-1} \tilde{x}(n) e^{-jk\frac{2\pi}{N}n}, \quad (3.23)$$

$$\tilde{x}(n) = \frac{1}{N} \sum_{k=0}^{N-1} \tilde{X}(k) e^{jk\frac{2\pi}{N}n}. \quad (3.24)$$

Ако сега ја земеме Фурьеовата трансформација на дискретниот сигнал  $x(n)$  дадена во (3.20), која претставува  $\mathcal{Z}$  трансформација на сигналот  $x(n)$  долж единечната кружница, и истата ја семплираме на еквидистантни фреквенции  $\omega_k$  дадени со:

$$\omega_k = k \frac{2\pi}{N}, \quad (3.25)$$

тогаш ќе го добијеме изразот за **дискретната Фурьеова трансформација ДФТ**:

$$X(k) = \sum_{n=0}^{N-1} x(n) e^{-jk\frac{2\pi}{N}n}, \quad (3.26)$$

кој е еквивалентен со (3.23)! Ова значи дека семплириралиот спектар што го добиваме со (3.34) соодветствува на сигнал  $\tilde{x}(n)$  која претставува периодична верзија од сигналот  $x(n)$  со период  $N$ , односно важи:

$$x(n) = \tilde{x}(n) \cdot \Pi(n), \quad (3.27)$$

каде

$$\Pi(n) = \begin{cases} 1, & 0 \leq n < N, \\ 0, & \text{поинаку.} \end{cases} \quad (3.28)$$

Со други зборови, дискретниот спектар на еден дискретен сигнал  $x(n)$  добиен со ДФТ претставува всушност спектар на периодичната верзија на сигналот во која тој се повторува бесконечно со периода еднаква на неговата должина  $N$ .

\* **Пример.** Да земеме дека должината на дискретниот сигнал е  $N = 4$ , тогаш спектарот на сигналот добиен со ДФТ ќе биде пресметан за следните вредности на кружната фреквенција  $\omega$ :  $\omega_0 = 0$ ,  $\omega_1 = \pi/2$ ,  $\omega_2 = \pi$  и  $\omega_3 = 3\pi/2$ . Поради симетричноста на амплитудниот спектар за реални сигнали, што се сведува на симетричност на единечната кружница во однос на реалната оска во  $z$  рамнината, примероците  $\omega_1$  и  $\omega_3$  се огледални слики и се идентични. Поради тоа што тие не носат дополнителна информација, вообичаена практика е огледалните слики да се исфрлат, при што мора да се удвојат соодветните примероци кои остануваат за да се зачува валидноста на теоремата на Парсевал. По исфрлањето, последниот примерок од спектарот ќе биде за кружна фреквенција  $\omega_3 = \pi$ , односно, за произволно  $T_s$ , за фреквенција  $f = f_s/2$ . Така, конечната скратена верзија на ДФТ спектарот би се состоела од:  $X(0)$ ,  $2X(\omega_1)$  и  $X(\pi)$ .

За произволно  $T_s$  примероците ги земаме во кружните фреквенции:

$$\omega_k = k \frac{2\pi}{NT_s}, \quad (3.29)$$

односно:

$$f_k = k \frac{1}{NT_s} = k \frac{f_s}{N}. \quad (3.30)$$

Резолуцијата на спектарот добиен со ДФТ:

$$\Delta\omega = \frac{2\pi}{N}, \quad (3.31)$$

е одредена од должината на сигналот  $N$ . За произволно  $T_s$  таа ќе биде:

$$\Delta\omega = \frac{2\pi}{NT_s}, \quad (3.32)$$

односно:

$$\Delta f = \frac{f_s}{N}. \quad (3.33)$$

За зголемување на резолуцијата на спектарот добиен со ДФТ постојат две методи: *i*) да се земат повеќе примероци на сигналот  $N$ , и *ii*) да се додадат  $N_0$  нули на крајот од сигналот. Последново ќе ја зголеми резолуцијата на семплирање во јадрото во (3.34), без да ги смени границите на сумата:

$$X(k) = \sum_{n=0}^{N-1} x(n)e^{-jk\frac{2\pi}{N+N_0}n}. \quad (3.34)$$

Сепак, добиената резолуција нема да донесе нова информација во ДФТ спектарот, туку само ќе генерира интерполирана верзија на оригиналниот спектар со  $N$  точки.

Соодветно **инверзната дискретна Фуриеова трансформација (ИДФТ)** е дадена со изразот:

$$x(n) = \frac{1}{N} \sum_{k=0}^{N-1} X(k)e^{jk\frac{2\pi}{N}n}. \quad (3.35)$$

За пресметка на ДФТ во праксата се користат алгоритми за нејзино брзо пресметување чија пресметковна сложеност е пропорционална со  $N \log N$ , кои се нарекуваат **брза Фуриеова трансформација (БФТ)** (Богданов, 1997).<sup>4</sup> Бидејќи БФТ работи брзо само за должини на сигналот од  $N = 2^M$ , вообичаена пракса е додавањето нули на сигналот за да се задоволи овој услов.

### Проценка на амплитудата на сигналот од неговиот спектар

За да видиме колкава е амплитудата на ФТ спектарот на еден дискретен сигнал ќе го пресметаме спектарот на една простопериодична низа  $x(n)$  со кружна фреквенција  $\omega_0$ :

$$x(n) = A \cos(\omega_0 n) = A \cos(2\pi f_0 n). \quad (3.36)$$

Спектарот на овој сигнал е даден со равенството:

$$X(\omega) = \frac{A}{2} 2\pi (\delta(\omega + \omega_0) + \delta(\omega - \omega_0)). \quad (3.37)$$

Ако земеме отсечок на овој сигнал со должина од  $N$  примероци и ја пресметаме континуираната ФТ, тоа е еквивалентно на пресметка на ФТ врз производот на  $x(n)$  со правоаголен прозорец  $w(n) = \Pi(n)$  со траење  $-N/2 \leq n < N/2$ . Според својството на Фуриеовата трансформација според кое множењето на два сигнали во временски домен претставува нивна конволуција во Фуриеов домен, имаме:

$$\hat{X}(\omega) = \frac{1}{2\pi} \int_{-\infty}^{\infty} X(\Omega) W(\omega - \Omega) d\Omega \quad (3.38)$$

$$= \frac{1}{2\pi} \int_{-\infty}^{\infty} \frac{A}{2} 2\pi (\delta(\Omega + \omega_0) + \delta(\Omega - \omega_0)) W(\omega - \Omega) d\Omega \quad (3.39)$$

$$= \frac{A}{2} (W(\omega + \omega_0) + W(\omega - \omega_0)), \quad (3.40)$$

<sup>4</sup>Wikipedia – Fast Fourier transform (FFT) [https://en.wikipedia.org/wiki/Fast\\_Fourier\\_transform](https://en.wikipedia.org/wiki/Fast_Fourier_transform)

каде  $W(\omega)$  е ФТ на прозорецот дадена со:

$$W(\omega) = \frac{\sin(N\frac{\omega}{2})}{\sin(\frac{\omega}{2})}. \quad (3.41)$$

Според Лопиталовото правило<sup>5</sup>, кое вели дека во случаи кога за гранична вредност на функција се добива неопределен израз како  $0/0$  или  $\infty/\infty$ , тогаш гранична вредност постои и таа се наоѓа преку изводите на двете функции, односно:

$$W(0) = \lim_{\omega \rightarrow 0} \frac{\sin(N\frac{\omega}{2})}{\sin\frac{\omega}{2}} = \lim_{\omega \rightarrow 0} \frac{\sin'(N\frac{\omega}{2})}{\sin'\frac{\omega}{2}} = \lim_{\omega \rightarrow 0} \frac{N \cos(N\frac{\omega}{2})}{\cos\frac{\omega}{2}} = N \quad (3.42)$$

Бидејќи амплитудата на спектарот на правоаголниот прозорец за  $\omega \neq 0$  е мала во споредба со  $N$  за  $\omega = 0$ , спектарот на  $\hat{x}(n)$  за фреквенција  $\omega_0$  ќе биде:

$$\hat{X}(\omega_0) = \frac{A}{2} (W(2\omega_0) + W(0)) \approx \frac{A}{2} N \quad (3.43)$$

## 3.2 Фуриевата трансформација на временски отсекоци ФТВО

Главниот недостаток на Фуриевата трансформација е тоа што таа не ни дава никаква временска информација за сигналот кој што го анализираме. Поради нестационарната природа на биомедицинските сигнали, Фуриевата анализа на целиот сигнал може да ни каже некои општи карактеристики за сигналот, но не може да ни ги покаже деталите. На пример, вкупниот спектар на еден музички сигнал може да ни каже дали во него има изразен бас преку анализата на енергијата на сигналот во ниските фреквенции, но ваквиот вкупен спектар не може да ни каже колку е брз ритамот на музичкиот сигнал. Оттаму потребата за временска локализација на спектралната информација што ја нуди **Фуриевата трансформација на временски отсекоци ФТВО**.



**§ Дополнително.** Унгарскиот електроинженер **Денеш Габор**<sup>6</sup> (1900–1979) е овој кој во 1946 ја адаптира Фуриевата трансформација за временска анализа и ја добива ФТВО. Тој е и изумителот на холографијата за што добива Нобелова награда по Физика во 1971. Една од неговите најпознати изјави е: „Најдобриот начин да се предвиди иднината е таа да се создаде.“

ФТВО го анализира спектарот во низа од кратки временски отсекоци наречени **рамки**, земени од биомедицинскиот сигнал со помош на функција за селекција  $w(n)$  која се нарекува **прозорец**, според равенството (3.44). Тука со  $i$  е означен редниот број на рамката,  $N$  е големината на прозорецот, а  $H$  е големината на скокот кој го прави прозорецот долж сигналот од една рамка до друга. Оваа метода на анализа на нестационарните сигнали преку земање на отсекоци со лизгање на прозорец по нивната должина се нарекува и **метода на прозорци**.

$$X_i(k) = \sum_{n=-N/2}^{N/2-1} w(n)x(n+i \cdot H)e^{-jk\frac{2\pi}{N}n}, \quad i = 0, 1, 2, \dots \quad (3.44)$$

На овој начин наместо еден вкупен спектар, се добива низа од спектри од сигналот пресметани за различни временски моменти во избрана нивна околина.

<sup>5</sup> Википедија – Лопиталово правило [https://mk.wikipedia.org/wiki/Лопиталово\\_правило](https://mk.wikipedia.org/wiki/Лопиталово_правило)

<sup>6</sup> Wikipedia: Dennis Gabor. [https://en.wikipedia.org/wiki/Dennis\\_Gabor](https://en.wikipedia.org/wiki/Dennis_Gabor)

## Видови на прозорци

Претходната анализа исто така го илустрира фактот дека спектарот на прозорецот има критично влијание за точната претстава на спектарот на биомедицинските сигнали. Поради оваа причина дизајнирани се низа од различни видови на прозорци, секој со различни спектрални карактеристики, а со тоа и со различна примена.<sup>7</sup> Сите тие во временски домен можат да се претстават како сума од косинуси, а во спектрален домен како сума од  $\text{sinc}(n)$  функции.

Во ФТВО анализата на биомедицинските сигнали најупотребувани прозорци (Serra and O Smith III, 2014) се:

- Правоаголен прозорец

$$\text{rect}(n) = 1, \quad -N/2 \leq n \leq N/2 \quad (3.45)$$

$$\text{Rect}(k) = \text{sinc}(k) \quad (3.46)$$

- Хан прозорец

$$\text{hann}(n) = 0,5 + 0,5 \cos\left(2\pi \frac{n}{N}\right), \quad -N/2 \leq n \leq N/2 \quad (3.47)$$

$$\text{Hann}(k) = 0,5 \text{sinc}(k) + 0,25 (\text{sinc}(k-1) + \text{sinc}(k+1)) \quad (3.48)$$

- Хаминг прозорец

$$\text{hamm}(n) = 0,54 + 0,46 \cos\left(2\pi \frac{n}{N}\right), \quad -N/2 \leq n \leq N/2 \quad (3.49)$$

- Блекман прозорец

$$\text{black}(n) = 0,42 - 0,5 \cos\left(2\pi \frac{n}{N}\right) + 0,080,5 \cos\left(4\pi \frac{n}{N}\right), \quad -N/2 \leq n \leq N/2 \quad (3.50)$$

- Блекман-Харис прозорец

$$\text{blackharr}(n) = \frac{1}{N} \sum_{i=0}^3 a_i \cos\left(2i\pi \frac{n}{N}\right), \quad -N/2 \leq n \leq N/2 \quad (3.51)$$

каде

$$a_0 = 0,35876, a_1 = 0,48829, a_2 = 0,14128, a_3 = 0,01168 \quad (3.52)$$

## Карактеристики на прозорците

Најважните спектрални карактеристики на прозорците се **ширината на главното крило** и **амплитудата на најголемото споредно крило**. Идеалниот прозорец има бескрајно тесно главно крило и нема споредни крила. Во реалноста станува збор за компромис помеѓу овие два параметри. Различните видови на прозорци можеме да ги генерираме со функцијата `get_window` од модулот `scipy.signal`. Во следниот код тоа е направено за правоаголниот прозорец `boxcar`. Притоа, амплитудните спектри се нормализираат до 0 dB.

```
import numpy as np
from matplotlib import pyplot as plt
from scipy import fftpack as fft
from scipy import signal as sig

m = 512 # околина за анализа
n = 64 # должина на прозорец
m_half = m / 2
n_half = n / 2
w = np.zeros(m)
```

<sup>7</sup>Wikipedia: Windowing function [https://en.wikipedia.org/wiki/Window\\_function](https://en.wikipedia.org/wiki/Window_function)

```
w[m_half-n_half : m_half+n_half] = sig.get_window('boxcar', n)
w_spec = fft.fft(w, m)
w_amp = np.abs(w_spec) / n
eps = 1e-9
w_amp[w_amp < eps] = eps
w_log = 20 * np.log10(w_amp)
w_log = w_log - np.max(w_log)
w_shift = fft.fftshift(w_log)

plt.figure(figsize=(12, 5))
plt.subplot(121)
plt.plot(w)
plt.axis([0, m, 0, 1]) # [xmin, xmax, ymin, ymax]
plt.grid()
plt.subplot(122)
plt.plot(w_shift)
plt.grid()
plt.axis([0, m, -100, 0])
```

Притоа за `eps` е земено произволно мала вредност со цел да ја избегнеме пресметката на логаритамот за вредности на спектарот блиски до нулата. Вистинската нумеричка прецизност за `float64` може да се добие со Нумпај наредбата `np.finfo(float).eps` и таа изнесува  $2,220446049250313 \cdot 10^{-16}$ .

Различните прозорци и нивните амплитудни спектри генериирани со овој код се прикажани на Сл. 3.1. Може да се види дека правоаголниот прозорец иако е добар за процесирање во временски домен има најлоши спектрални карактеристики во поглед на амплитудата на страничните крила. Од друга страна, најмало влијание на страничните крила имаме кај Блекман-Харис прозорецот, но тој за сметка на тоа има најшироко главно крило. Хановиот и Хаминговиот прозорец имаат иста широчина на главното крило на различно распоредена енергија во страничните крила.

### 3.3 Вовед во филтри

### 3.4 Основи на дигиталните филтри

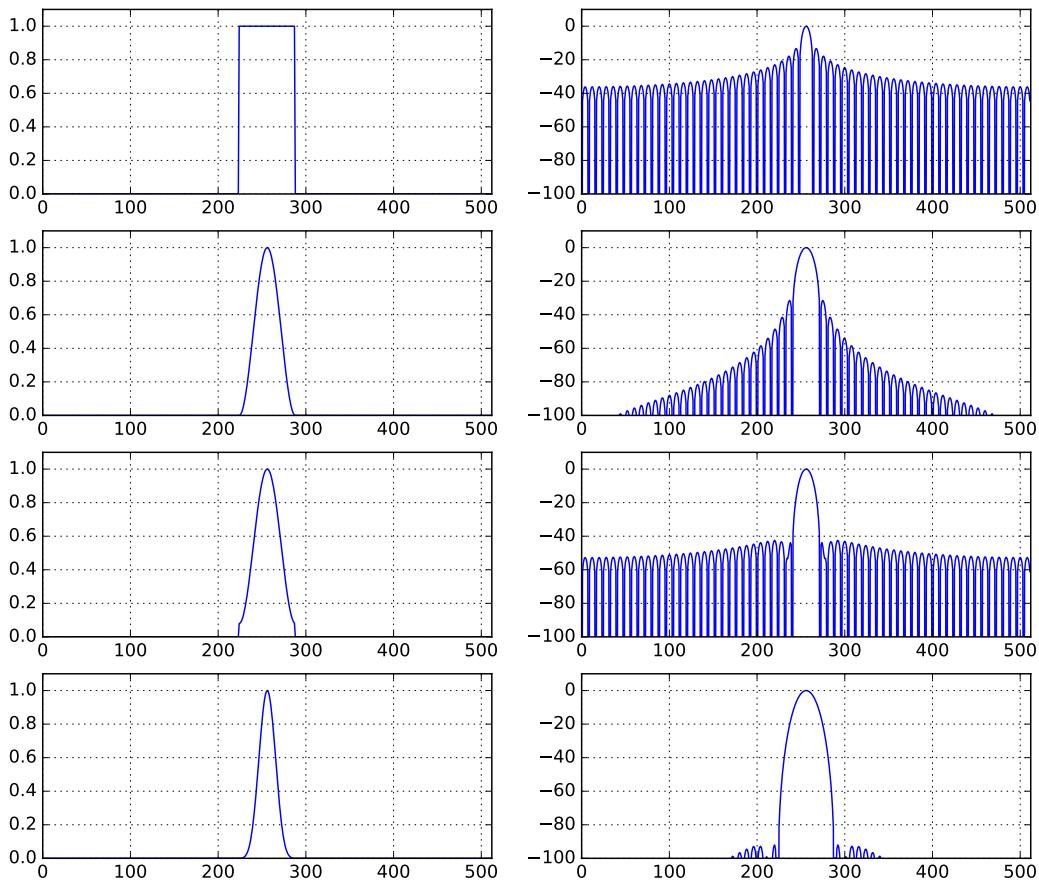
**Филтрите** претставуваат системи чија примарна намена е обликувањето на спектарот на биомедицинските сигнали. Аналогни и дигитални филтри се потребни за ограничување на спектарот во АД конверзијата, а играат клучна улога во отстранувањето на несаканите електромагнетни шумови. Последново е особено критично кај биомедицинските сигнали поради нивната мала амплитуда. Аналогните филтри се реализираат со помош на збир на пасивни и активни електронски елементи, чија нагоденост е неопходна, но скапа. Од друга страна, реализацијата сложени филтерски структури во дигитален домен е многу поекономична, а нумеричките операции на кои се базираат дигиталните филтри не се подложни на временски и температурни влијанија.

Дигиталните филтри претставуваат линеарни и временски инваријантни (**ЛВИ**), односно линеарни и инваријантни на поместување дискретни системи<sup>8</sup>. Овие две особини изискуваат ако за даден влезен сигнал  $x(n)$  системот го дава излезниот сигнал  $y(n) = H\{x(n)\}$ , тогаш мора да важи (Богданов, 1997):

$$H\left\{\sum_{m=1}^M a_m x_m(n)\right\} = \sum_{m=1}^M a_m y_m(n), \quad \forall a_m \text{ -- линеарност и} \quad (3.53)$$

$$H\{x(n-k)\} = y(n-k), \quad \forall k \text{ -- инваријантност на поместување.} \quad (3.54)$$

<sup>8</sup>Linear time-invariant (LTI), односно linear shift-invariant (LSI) системи.



**Сл. 3.1:** Споредба на временскиот и спектрален облик на четири најчесто користени прозорци (од горе надолу): правоаголен, Ханов, Хамингов и Блекман-Харисов.

Секој ЛВИ систем е во потполност описан од неговиот **импулсен одсив**  $h(n)$  односно неговата **преносна функција** во  $z$ -домен  $H(z)$ , или пак конечно од неговата **фреквенциска преносна функција** во Фурьеов домен  $H(\omega)$ . Како и за останатите ЛВИ системи, излезниот сигнал на дигиталните филтри  $y(n)$  за даден влезен сигнал  $x(n)$  може да се добие во трите домени со следните релации (Rabiner and Schafer, 1978):

$$y(n) = x(n) * h(n) = \sum_{m=-\infty}^{\infty} x(m)h(n-m), \quad (3.55)$$

$$Y(z) = H(z)X(z), \quad (3.56)$$

$$Y(\omega) = H(\omega)X(\omega). \quad (3.57)$$

За системот да биде остварлив, нужно е тој да биде **каузален**, односно да важи:

$$h(n) \equiv 0, \quad n < 0. \quad (3.58)$$

За системот пак да биде стабилен потребен и доволен услов е да важи:

$$\sum_{n=-\infty}^{\infty} |h(n)| < \infty. \quad (3.59)$$

### Типови филтри според должината на импулсниот одсив

Според импулсниот одсив филтрите ги делиме на две основни групи и тоа филтри со:

- **конечен импулсен одсив (ФИР<sup>9</sup>)** и

<sup>9</sup>Од англиското Finite impulse response.

- бесконечен импулсен одсив (ИИР<sup>10</sup>).

ФИР филтрите имаат одредени предности над ИИР филтрите, а тоа се пред сè нивната стабилност и нивната линеарна фазна карактеристика. ИИР филтрите пак можат да постигнат подобра амплитудна фреквенциска карактеристика за помал ред на филтерот.

Сите ЛВИ системи кои се практично применливи како дигитални филтри можат да се описат со **диференцната равенка**:

$$y(n) = \sum_{i=0}^p b_i x(n-i) - \sum_{i=1}^q a_i y(n-i). \quad (3.60)$$

Според ова равенство секој одбирок на излезниот сигнал  $y(n)$  зависи од  $p$  претходни примероци на влезниот сигнал и  $q$  минати примероци од излезниот, односно во системот има повратна врска, па велиме дека тој е **рекурзивен**. ИИР филтрите секогаш се реализираат со повратна врска, од каде произлегуваат и проблемите со нивната стабилност. ФИР филтрите можат да бидат реализирани и со повратна врска, но најчесто се без неа. Преносната функција на системот описан од диференцната равенка можеме да ја добиеме од (3.60):

$$y(n) + \sum_{i=1}^q a_i y(n-i) = \sum_{i=0}^p b_i x(n-i), \quad (3.61)$$

$$\left( 1 + \sum_{i=1}^q a_i z^{-i} \right) Y(z) = \sum_{i=0}^p b_i z^{-i} X(z), \quad (3.62)$$

$$H(z) = \frac{Y(z)}{X(z)} = \frac{\sum_{i=0}^p b_i z^{-i}}{1 + \sum_{i=1}^q a_i z^{-i}}. \quad (3.63)$$

Тука имплицитно претпоставуваме дека  $a_0 = 1$ . Во праксата овој услов секогаш се обезбедува преку нормализирање на останатите  $a$  коефициенти. Бројот на коефициенти во повратната врска на ИИР филтрите вообичаено се зема да е еднаков со бројот на коефициенти во директната врска. Па, важи:

$$q = p = N, \quad (3.64)$$

каде со  $N$  се означува **редот на филтерот**. Преносната карактеристика  $H(z)$  исто така може да биде претставена преку нејзините **полови и нули** во  $z$ -рамнината:

$$H(z) = \frac{A \prod_{i=1}^p (1 - c_i z^{-1})}{\prod_{i=1}^q (1 - d_i z^{-1})}. \quad (3.65)$$

### Типови филтри според амплитудната карактеристика

Фреквенциската карактеристика на филтрите ја добиваме директно од преносната карактеристика со замената  $z = e^{j\omega}$ :

$$H(\omega) = \frac{\sum_{i=0}^p b_i e^{-j i \omega}}{1 + \sum_{i=1}^q a_i e^{-j i \omega}}. \quad (3.66)$$

Таа може дополнително да се изрази преку нејзината амплитуда и фаза како:

$$H(\omega) = |H(\omega)| e^{j \angle H(\omega)} = A(\omega) e^{j \phi(\omega)}, \quad (3.67)$$

каде  $A(\omega)$  е амплитудната **фреквенциска карактеристика** на филтерот, додека  $\phi(\omega)$  е неговата **фазна карактеристика**. Според амплитудната карактеристика разликуваме пет типови на филтри и тоа:

- **нископропусни (НП)**,

---

<sup>10</sup>Од англиското Infinite impulse response.

- високопропусни (ВП),
- пропусни на опсег (ПО),
- непропусни на опсег (НО) и
- сепропусни (СП).<sup>11</sup>

Кога фазната карактеристика на филтерот  $\phi(\omega)$  е линеарна тогаш групното доцнење  $\tau(\omega)$  е константно:

$$\tau(\omega) = -\frac{d\phi(\omega)}{d\omega} = \text{const.} \quad (3.68)$$

Ова значи дека филтерот нема да внесе фазни изобличувања. Со тоа, компонентите на сигналот на различни фреквенции нема да бидат различно задоцнети па ќе биде задржана нивната компактност во излезниот сигнал, односно нема да дојде до нивно расејување. Ова е посебно важно во обработката на биомедицинските сигнали, па затоа за нивно филтрирање вообичаено се употребуваат ФИР филтри.

### Типови ФИР филтри со линеарна фазна карактеристика

Строго линеарната фазна карактеристика не може да биде постигнато со ИИР или со аналогните филтри (Богданов, 1997), туку само со ФИР филтри чив импулсен одсив е симетричен во однос на средниот примерок  $h(\frac{N-1}{2})$ :

$$h(n) = h(N - 1 - n), \quad (3.69)$$

или пак кога е антисиметричен во однос на него:

$$h(n) = -h(N - 1 - n). \quad (3.70)$$

Тука редот на филтерот  $N$  ја дава и должината на импулсниот одсив, што не е случај кај ИИР филтрите.

Постојат четири типови на ФИР филтри со линеарна фазна карактеристика:

#### Тип I – Симетричен импулсен одсив, N непарен

Импулсниот одсив на овој тип на филтри можеме да го запишеме како:

$$h(n) = h(0)\delta(n) + h(1)\delta(n - 1) + \cdots + h(\frac{N-1}{2})\delta(n - \frac{N-1}{2}) + \cdots \quad (3.71)$$

$$\cdots + h(1)\delta(n - (N-2)) + h(0)\delta(n - (N-1)), \quad (3.72)$$

па за преносната функција имаме:

$$H(z) = h(\frac{N-1}{2})z^{-\frac{N-1}{2}} + \sum_{i=0}^{\frac{N-3}{2}} h(i) \left( z^{-i} + z^{-(N-1-i)} \right) \quad (3.73)$$

$$= z^{-\frac{N-1}{2}} \left( h(\frac{N-1}{2}) + \sum_{i=0}^{\frac{N-3}{2}} h(i) \left( z^{-i+\frac{N-1}{2}} + z^{i-\frac{N-1}{2}} \right) \right) \quad (3.74)$$

$$= z^{-\frac{N-1}{2}} \sum_{i=0}^{\frac{N-1}{2}} a(n) \frac{z^n + z^{-n}}{2}, \quad (3.75)$$

каде:

$$a(n) = \begin{cases} h(\frac{N-1}{2}), & n = 0 \\ 2h(-n + \frac{N-1}{2}) & n = 1, 2, \dots, \frac{N-3}{2} \end{cases}. \quad (3.76)$$

<sup>11</sup>Овој тип на филтри наоѓа примена кај системите за синтеза на дигитално ехо и реверберација.

Од (3.75) може да се пресмета фреквенциската карактеристика на филтерот:

$$H(\omega) = e^{-j\omega \frac{N-1}{2}} \sum_{n=0}^{\frac{N-1}{2}} a(n) \cos(n\omega). \quad (3.77)$$

#### Тип II – Симетричен импулсен одсив, N парен

Следејќи ја истата постапка како за ФИР филтерот од тип I можеме да дојдеме до фреквенциската карактеристика на типот II:

$$H(\omega) = e^{-j\omega \frac{N-1}{2}} \sum_{n=1}^{\frac{N-1}{2}} b(n) \cos((n - \frac{1}{2})\omega), \quad (3.78)$$

каде:

$$b(n) = 2 h(-n + \frac{N}{2}), \quad n = 1, 2, \dots, \frac{N}{2}. \quad (3.79)$$

Од (3.78) може да се види дека без оглед на вредноста на коефициентите на филтерот  $b(n)$  неговата фреквенциска карактеристика ќе биде 0 за  $\omega = \pm\pi$ . Поради тоа, овој тип на филтер не може да се употреби како високопропусен или непропусник на опсег.

#### Тип III – Антисиметричен импулсен одсив, N непарен

Фреквенциската карактеристика на типот III е:

$$H(\omega) = e^{-j(\omega \frac{N-1}{2} - \frac{\pi}{2})} \sum_{n=1}^{\frac{N-1}{2}} c(n) \sin(n\omega), \quad (3.80)$$

каде:

$$c(n) = 2 h(-n + \frac{N-1}{2}), \quad n = 1, 2, \dots, \frac{N-1}{2}. \quad (3.81)$$

Од (3.80) следи дека без оглед на вредноста на  $c(n)$  неговата фреквенциска карактеристика ќе биде 0 за  $\omega = 0$  и за  $\omega = \pm\pi$ . Поради тоа, овој тип на филтер може да се употреби само како пропусник на опсег.

#### Тип IV – Антисиметричен импулсен одсив, N парен

Фреквенциската карактеристика на овој филтер е:

$$H(\omega) = e^{-j\omega \frac{N-1}{2}} \sum_{n=1}^{\frac{N}{2}} d(n) \sin((n - \frac{1}{2})\omega), \quad (3.82)$$

каде:

$$d(n) = 2 h(-n + \frac{N}{2}), \quad n = 1, 2, \dots, \frac{N}{2}. \quad (3.83)$$

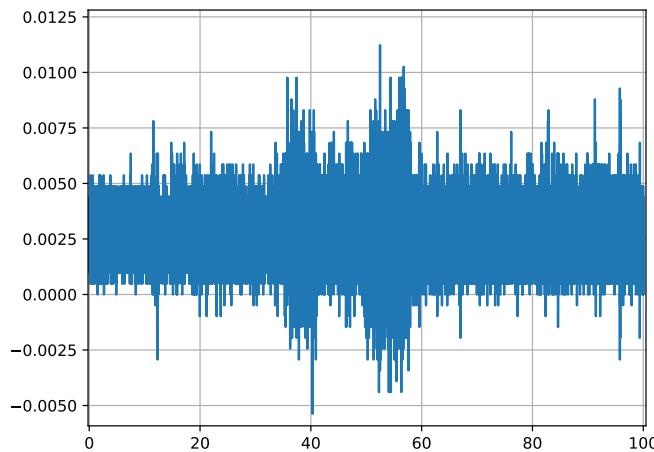
Од (3.82) следи дека фреквенциска карактеристика ќе биде 0 за  $\omega = 0$ , па овој тип на филтер не може да се употреби како нископропусен или пропусник на опсег.

Постојат различни пристапи за [дизајн на дигитални филтри](#). Трите најпознати методи за дизајн на ФИР филтри се:

- метода на прозорци,
- дизајн заснован на DFT,
- оптимален дизајн на еднаквобранести филтри.

Од друга страна за дизајн на ИИР филтри најпознати методи се:

- Батерворт,
- Бесел,
- Чебишев,
- Елиптичен.



**Сл. 3.2:** Изглед на вчитаниот ЕМГ сигнал.

### 3.5 Процесирање на ЕМГ сигнали

Во овој дел ќе направиме низа основни процесирања на еден ЕМГ сигнал. Како примерок ќе го искористиме сигналот вклучен во Биоспј<sup>12</sup> `emg.txt`. Во заглавието на овој текстуален фајл се дадени основните параметри на ЕМГ сигналот кој е семплиран со фреквенција  $f_s$  од 1000 Hz со резолуција од 12 битови по одбирок. Да ја вчитаме и да ја прикажеме на графикон.

```
import numpy as np
from matplotlib import pyplot as plt

# %% вчитај го фајлот
file_name = 'data/emg.txt'
emg = np.loadtxt(file_name)
fs = 1000
n_bit = 12

# %% направи го наизменичен и скалирај го од -1 до 1
emg = emg - 2**n_bit-1
emg = emg / 2**n_bit-1
t = np.arange(0, emg.size/fs, 1/fs)

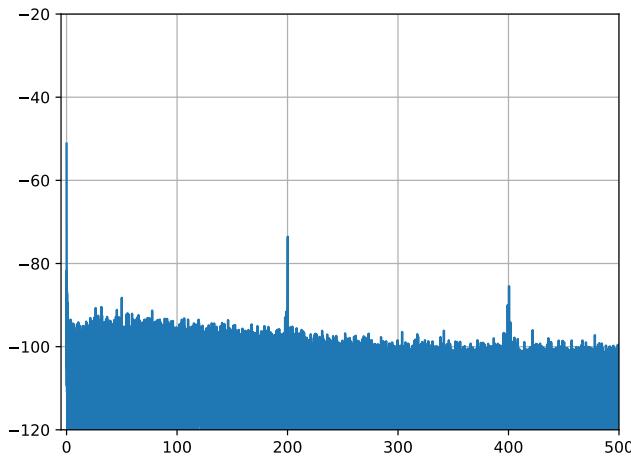
# %% plot
plt.figure()
plt.plot(t, emg)
plt.grid()
```

При што, за вчитување од текст фајлот сме ја искористиле Нумпај функцијата `loadtxt`. Бидејќи во 12 битниот запис нема кодирање на негативни амплитуди со двоен комплемент, треба првин да го направиме сигналот наизменичен а потоа и да го скалираме во опсег од  $-1$  до  $1$ . Добиениот графикон е прикажан на Сл. 3.2

#### Фреквенциски спектар на ЕМГ сигналот

За пресметка на спектарот на ЕМГ сигналот ќе се послужиме со дискретната Фурьеова трансформација (ДФТ), воведена во Поглавјето 3, поточно алгоритамот за брза Фурьеова трансформација (БФТ) кој е имплементиран во модулот `fftpack` во Сајпај пакетот.

<sup>12</sup>BioSPPy – Biosignal Processing in Python. <https://github.com/PIA-Group/BioSPPy>



**Сл. 3.3:** Амплитуден спектар на ЕМГ сигналот.

```
from scipy import fftpack as fft
# пресметка на бројот на точки за БФТ
x = np.ceil(np.log2(emg.size))
n_fft = np.int(2**x)

emg_fft = fft.fft(emg, n_fft)
emg_amp = np.abs(emg_fft)
n_keep = np.int(n_fft/2 + 1)
emg_amp = emg_amp[:n_keep]
emg_amp[1:-1] = 2 * emg_amp[1:-1]
emg_amp = emg_amp / emg.size
emg_db = 20 * np.log10(emg_amp)

emg_pha = np.angle(emg_fft)
emg_pha = np.unwrap(emg_pha)

f = np.linspace(0, fs/2, n_keep)

# плотирање
plt.figure()
plt.plot(f, emg_db)
plt.grid()
```

Притоа за пресметка на амплитудниот спектар најпрвин го пресметавме бројот на точки за пресметка на БФТ кој треба да биде  $2^x$  а потоа ги исфрливме примероците од спектарот кои претставуваат копии и се редундантни. Добиениот амплитуден спектар на ЕМГ сигналот е прикажан на Сл. 3.3. Може да видиме дека во сигналот има изразена еднонасочна (DC) компонента и хармониски шум со фреквенција од 200 и 400 Hz. Исто така, гледаме дека на сигналот не е суперпониран шум од градската мрежа на 50 Hz.

За отстранување на еднонасочната компонента ќе го направиме следното:

```
import bme
emg = emg - emg.mean()
```

За пресметка на амплитудниот спектар на биомедицинските сигнали згодно е да направиме функција која ќе ја поместиме во нов Питон фајл `bme.py`, којшто всушност ќе претставува модул за сите функции во предметов. Еве како треба да изгледа неговата содржина:

```

#!/usr/bin/env python3
"""
Функции за обработка на биомедицинските сигнали.
"""

import numpy as np
from scipy import fftpack as fft
from matplotlib import pyplot as plt

def get_spectrum(fs, x, n_fft=None, log=True):
    """
    Пресметај спектар со БФТ.
    """

    n = x.size
    if n_fft is None:
        n_fft = np.int(2**np.ceil(np.log2(n)))
    x_spec = fft.fft(x, n_fft)
    x_spec = np.abs(x_spec)
    x_spec = x_spec / n
    n_keep = np.int(n_fft/2) + 1
    x_spec = x_spec[0:n_keep]
    x_spec[1 : -1] = 2 * x_spec[1 : -1]
    f = np.linspace(0, fs/2, n_keep)
    if log:
        x_spec = 20 * np.log10(x_spec)
    return f, x_spec

```

Сега спектрарот на кој бил сигнал може да го пресметаме повикувајќи ја нашата функција `get_spectrum` на следниот начин:

```

import bme
f, emg_spec = bme.get_spectrum(fs, emg, n_fft)

```

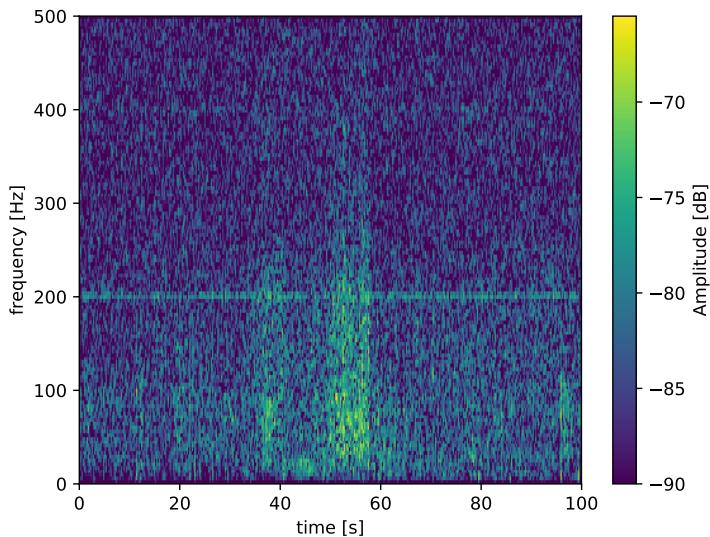
## Спектрограм на ЕМГ сигналот

Директно од амплитудниот спектар на ЕМГ сигналот, не можеме да видиме во кој дел доаѓа до израз мускулната активација. За да дојдеме до оваа информација ни треба темпорална, односно временска, информација заедно со спектралната. За таа цел ќе се послужиме со **спектрограмот** – еден од најзначајните прикази на биомедицинските сигнали. На него во исто време може да се набљудуваат карактеристиките на сигналот и во временски и во спектрален домен. За да го добиеме ќе се послужиме со Фурьеовата трансформација на временски отсекоци (ФТВО), воведена во [Поглавјето 3](#).

```

n_win = 256 # 2**x
n_half = np.int(n_win / 2)
n_hop = n_half # Hann, Hamming
# додади нули за прозорецот во првата рамка да биде центриран на првиот примерок
pad = np.zeros(n_half)
emg_pad = np.concatenate((pad, emg, pad))
pos = 0
frames = None
while pos <= emg_pad.size - n_win:
    frame = emg_pad[pos : pos+n_win]
    f, frame_spec = bme.get_spectrum(fs, frame, n_win)
    frame_2d = frame_spec[:, np.newaxis]
    if frames is None:
        frames = frame_2d
    else:
        frames = np.concatenate((frames, frame_2d), axis=1)

```



**Сл. 3.4:** Спектрограм на ЕМГ сигналот.

```

pos += n_hop

plt.figure()
plt.imshow(frames,
           extent=[0, t[-1], 0, f[-1]],
           aspect='auto',
           origin='lower',
           vmin=-90)
cbar = plt.colorbar()
plt.xlabel('time [s]')
plt.ylabel('frequency [Hz]')
cbar.ax.set_ylabel('Amplitude [dB]')

```

Приказот е даден на Сл. 3.4. Можеме јасно да видиме дека ЕМГ сигналот регистрира активација на мускулот во два наврати, еднаш околу 37 s со кратко времетраење, и втор пат околу 49 s со поголемо траење и интензитет. Исто така може да видиме дека главниот дел од спектралната моќност на ЕМГ сигналот се наоѓа во опсегот од 20 до 280 Hz. Оваа информација ќе ни послужи за дизајнирање на филтер за добивање на почиста претстава на ЕМГ сигналот.

Бидејќи и понатаму ќе треба да го пресметуваме спектрограмот на биомедицинските сигнали, во нашиот модул `bme.py` ќе додадеме функции за негова пресметка и исцртување.

```

def get_spectrogram(fs, x, n_win=None, log=True, win_type='hann'):
    """
    Пресметај го спектрограмот.
    """
    if n_win is None:
        n_win = 256
    win = sig.get_window(win_type, n_win)
    n_half = np.int(n_win / 2)
    n_hop = n_half # Hann, Hamming
    pad = np.zeros(n_half)
    x_pad = np.concatenate((pad, x, pad))
    pos = 0
    frames = None
    while pos <= x_pad.size - n_win:
        frame = x_pad[pos : pos+n_win]

```

**§ Дополнително.** За прикажување на амплитудата на спектрограмот постојат најразлични низи на бои, наречени мали на боја. Долго време во стандардна употреба е мапата `jet`, но таа има низа на недостатоци, поради кои денес се исфрла од употреба. Мапата искористена за приказ на спектрограмите во ова поглавје е новата `viridis` која е претставена на конференцијата Сајнай 2015<sup>13</sup>.

```
frame = frame * win
f, frame_spec = get_spectrum(fs, frame, n_win, log=log)
frame_2d = frame_spec[:, np.newaxis]
if frames is None:
    frames = frame_2d
else:
    frames = np.concatenate((frames, frame_2d), axis=1)
pos += n_hop
t = np.arange(0, frames.shape[1]*n_hop/fs, n_hop/fs)
return t, f, frames

def show_spectrogram(t, f, spectrogram, vmin=-90, vmax=0):
    """
    Прикажи го спектрограмот.
    """
    plt.figure()
    plt.imshow(spectrogram, extent=[0, t[-1], 0, f[-1]], aspect='auto',
               origin='lower', vmin=vmin, vmax=vmax)
    plt.colorbar()
```

\* Важно! Најважните два параметри во пресметување на спектрограмот се:

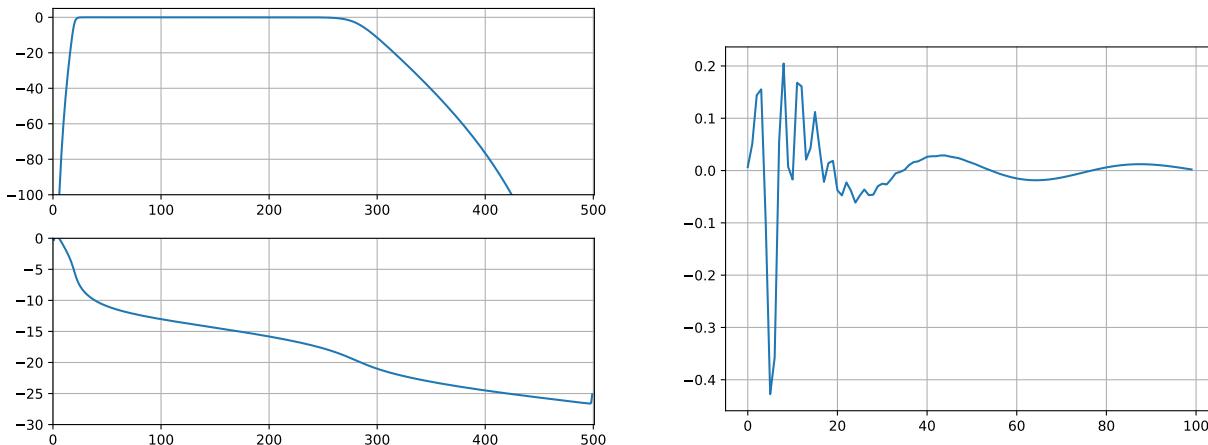
1. **Типот на прозорец.** Разликите во спектрограмот кои се добиваат со различните прозорци зависат од нивните спектрални карактеристики прикажани на Сл. 3.1.
2. **Големината на прозорецот.** Колку е поголем прозорецот толку повеќе точки, односно поголема резолуција, во спектарот на рамката земена од сигналот. Но исто така, колку поголем прозорец толку погруба временска резолуција на ФТВО анализата. Имено, временските моменти на промена во спектарот биваат размачкани. Така, за добра временска резолуција ни требаат пократки прозорци (рамки од сигналот), кои пак повлекуваат лоша фреквенциска резолуција. Потребата од компромис меѓу време и фреквенција е главниот недостаток кај спектрограмите, односно ФТВО анализата.

## Филтрирање на ЕМГ сигналот

За филтрирање на спектарот на ЕМГ сигналот ќе се послужиме со филтер со бесконечен импулсен одсив (ИИР) поради неговите добри фреквенциски карактеристики, коишто беа изложени во Поглавјето 3.3. Каде ЕМГ сигналот обликот на сигналот не е од големо значење, па може да употребиме и филтри без строго линеарна фазна карактеристика. Ќе дизајнираме филтер пропусник на опсег со кој сакаме да добиеме почиста верзија од нашиот ЕМГ сигнал.

```
from scipy import signal as sig
f_l = 20
f_h = 280
order = 9
b, a = sig.iirfilter(order, [f_l/(fs/2), f_h/(fs/2)], btype='bandpass', ftype='butter')
```

<sup>13</sup> Особено интересното претставување на новата мапа може да го погледнете на следниот линк: Nathaniel Smith and Stéfan van der Walt – A Better Default Colormap for Matplotlib <https://www.youtube.com/watch?v=xAoljeRJ3LU>



**Сл. 3.5:** Амплитудна и фазна карактеристика (лево) и дел од импулсниот одсив (десно) на дизајнираниот ИИР филтер пропусник на опсег.

```
f, h_f = sig.freqz(b, a, fs=fs)

# %% фреквенциска карактеристика на филтерот
plt.figure()
plt.subplot(2, 1, 1)
plt.plot(f, 20*np.log10(np.abs(h_f)))
plt.grid()
plt.subplot(2, 1, 2)
plt.plot(f, np.unwrap(np.angle(h_f)))
plt.grid()
plt.tight_layout()

# %% импулсен одсив
excite = np.zeros(100)
excite[0] = 1 # Дираков импулс
h_n = sig.lfilter(b, a, excite)
plt.figure()
plt.plot(h_n)
plt.grid()
```

Амплитудната и фазната карактеристика на дизајнираниот филтер, како и дел од неговиот импулсен одсив се дадени на Сл. 3.5. Може да се види дека амплитудната карактеристика е строго монотона, што е и карактеристика на Батерворт ИИР филтите.

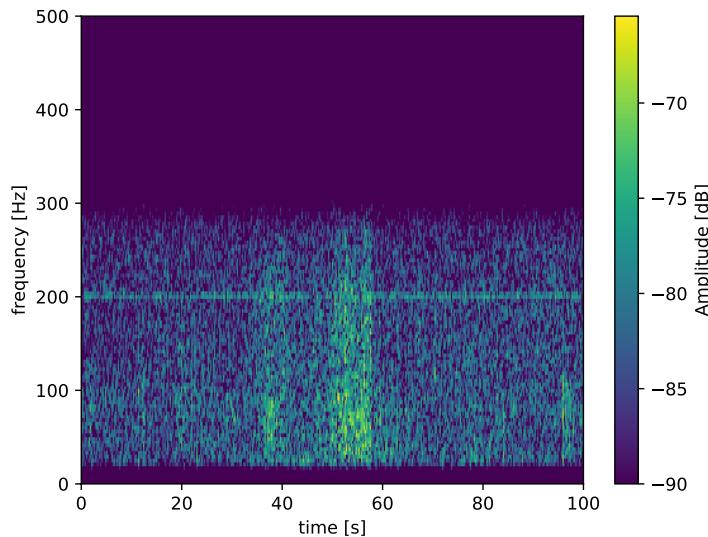
Следно ќе го исфилтрираме ЕМГ сигналот со дизајнираниот ИИР филтер и ќе го прикажеме неговиот спектрограм, даден на Сл. 3.6.

```
emg_filt = sig.lfilter(b, a, emg_noDC)
t, f, emg_filt_specgram = bme.get_spectrogram(fs, emg_filt, n_win)
bme.show_spectrogram(t, f, emg_filt_specgram, vmin=-90, vmax=None)
```

### Детекција на мускулна активација

За детекција на мускулна активација, ЕМГ сигналот вообичаено се поминува низ целобранов насочувач.<sup>14</sup> Таквиот сигнал се процесира со поместувачки усреднувач, што всушност претставува нископропусен филтер реализиран со методата на прозорци. Почетокот на активацијата на мускулите се детектира во усреднетиот сигнал со наогање на временските моменти кога сигналот поминува зададен prag. Притоа, pragот се пресметува според

<sup>14</sup>Процесирањето на ЕМГ сигналот во овој дел се базира на обработката интегрирана во програмскиот пакет Биоспај <https://github.com/PIA-Group/BioSPPy>.



**Сл. 3.6:** Спектрограм на филтриралиот ЕМГ сигнал.

амплитудната содржина на усреднетиот ЕМГ сигнал. Конечно, детектираните моменти на активација, повторно се усреднуваат, овој пат со поголема временска константа, односно со прозорец со поголемо времетраење, за добивање на конечниот сигнал на мускулна активација.

```
# %% целобранов насочувач
emg_rect = np.abs(emg_filt)

# поместувачки усреднувач
t_win = .05 # 50 ms
n_win = np.int(t_win * fs)
win = sig.get_window('parzen', n_win)
win = win / np.sum(win)

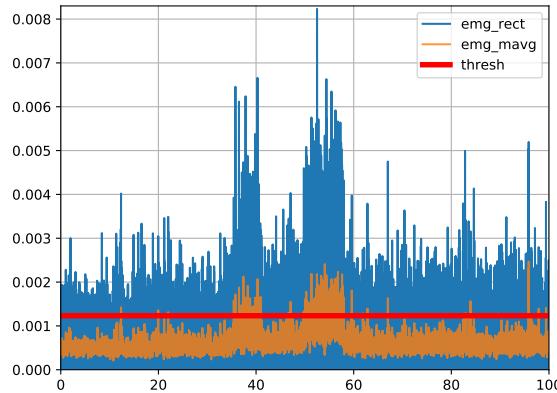
# симетрично падирање и усреднување на сигналот
n_pad = np.int(n_win / 2)
emg_pad = np.concatenate((emg_rect[n_pad:0:-1], emg_rect, emg_rect[-1:-n_pad:-1]))
emg_mavg = sig.convolve(emg_pad, win, 'same')
emg_mavg = emg_mavg[n_pad:-n_pad+1]

# %% пресметка на прагот
thresh = 1.2*emg_mavg.mean() + 2*emg_mavg.std()

plt.figure()
plt.plot(t, emg_rect)
plt.plot(t, emg_mavg, alpha=.8)
plt.plot([t[0], t[-1]], [thresh, thresh], c='r', lw=4)
plt.grid()

# %% детекција на премини преку прагот
over = np.where(emg_mavg > thresh)[0]
under = np.where(emg_mavg <= thresh)[0]
onsets = np.intersect1d(over[1], under)
onsets += 1

plt.figure()
plt.plot(t, emg_rect)
plt.plot(t, emg_mavg, alpha=.8)
```



**Сл. 3.7:** Добиени сигнали со целобрановото насочување и усреднувањето на ЕМГ сигналот, како и пресметаниот праг за детекција на мускулна активност.

```

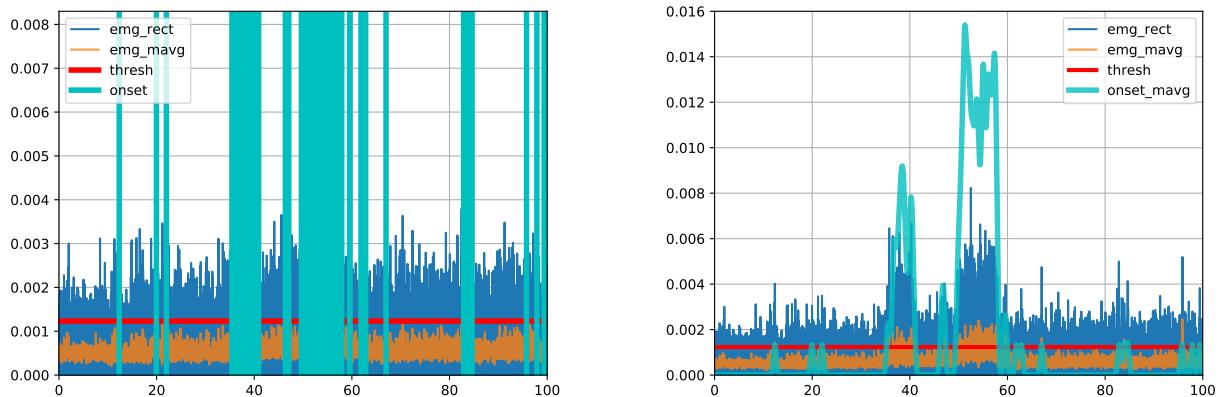
plt.plot([t[0], t[-1]], [thresh, thresh], c='r', lw=4)
for onset in onsets:
    plt.plot([t[onset], t[onset]], [0, emg_rect.max()], c='c', lw=4)
plt.grid()

# %% усреднување со поголема временска константа
t_win = 2 # 50 ms
n_win = np.int(t_win * fs)
win = sig.get_window('parzen', n_win)
win = win / np.sum(win)
onset_sig = np.zeros(emg_rect.size)
onset_sig[onsets] = 1
onset_mavg = sig.convolve(onset_sig, win, 'same')

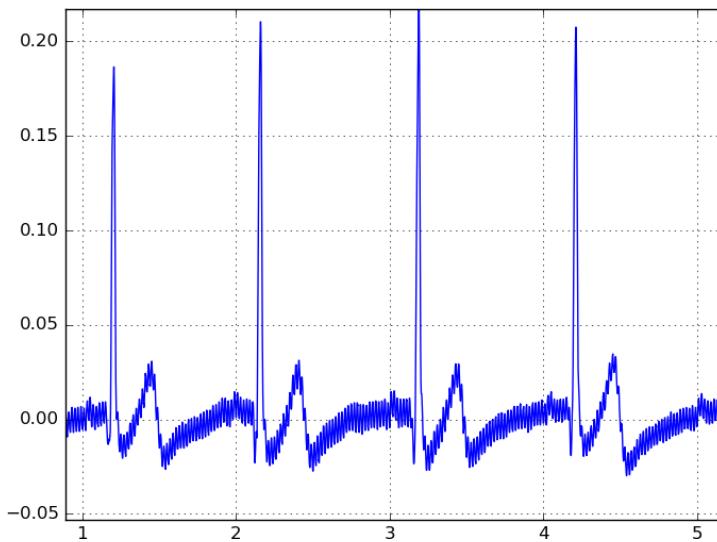
plt.figure()
plt.plot(t, emg_rect)
plt.plot(t, emg_mavg, alpha=.8)
plt.plot([t[0], t[-1]], [thresh, thresh], c='r', lw=3)
plt.plot(t, onset_mavg, c='c', lw=4, alpha=.8)
plt.grid()

```

Добиените сигнали по целобрановото насочување, усреднувањето и пресметаниот праг се прикажани на Сл. 3.7. На Сл. 3.8 пак се прикажани детектираниите премини низ прагот и нивната усредната верзија. Можеме да видиме дека конечно добиениот сигнал ни дава добра претстава за временските моменти и интензитетот на мускулната активација регистрирани во ЕМГ сигналот.



**Сл. 3.8:** Детектирани премини на усреднетиот ЕМГ сигнал низ прагот (лево) и нивната усредната верзија (десно).



Сл. 3.9: Изглед на измерениот ЕКГ сигнал.

## 3.6 Процесирање на ЕКГ сигнали

Во овој дел ќе направиме низа основни процесирања на еден ЕКГ сигнал. Како примерок ќе го искористиме пример сигналот вклучен во Биоспај<sup>15</sup> `ecg.txt`. Во овој текстуален фајл е запишан ЕКГ сигнал со времетраење од 15 s семплиран со фреквенција  $f_s$  од 1000 Hz со резолуција од 12 битови по одбирок. Истата ќе ја вчитаме со следниот код.

```
import numpy as np
import matplotlib.pyplot as plt
import scipy.fftpack as ff

ecg = np.array([])
count = 0

with open('ecg.txt') as fh:
    for line in fh:
        if count > 3:
            val = float(line[:-3])
            ecg = np.append(ecg, val)
        count += 1

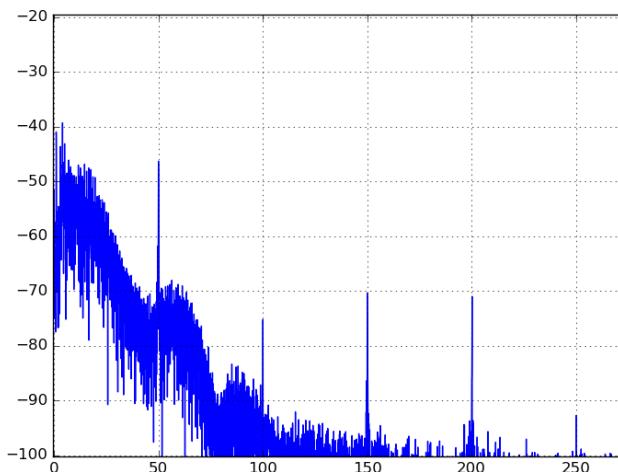
ecg = (ecg - 2**11) / 2**11
Fs = 1000
t = np.arange(0, ecg.size/Fs, 1/Fs)
plt.plot(t, ecg)
plt.grid()
```

На почетокт од кодот се увезува деленето со автоматско каствување во `float` од Python 3.5 (1), како и потребните модули (2–4). Потоа се иницијализира `ecg` NumPy векторот (6) во кој во `for` јамката (10) се вчитуваат сите примероци од ЕКГ сигналот конвертирани во `float` (12–13). Потоа сигналот се сведува во опсег од –1 до 1 (16) и се врши негово испртување со помош на функцијата `plot` од `matplotlib` (17–20). Добиениот график е прикажан на Сл. 3.9

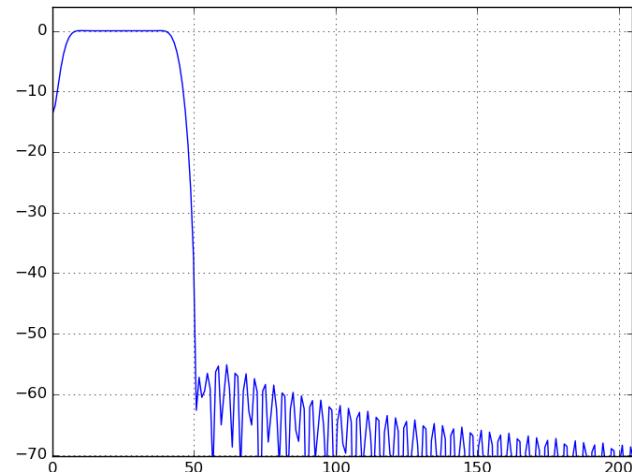
### Спектар на ЕКГ сигналот

Од графикот може да се видат отчукувањата на срцето преку QRS комплексот на ЕКГ сигналот. Исто така, можат да се забележат два вообичаени проблеми со ЕКГ сигналите – дрифт на DC

<sup>15</sup>BioSPPy – Biosignal Processing in Python. <https://github.com/PIA-Group/BioSPPy>



**Сл. 3.10:** Амплитуден спектар на измерениот ЕКГ сигнал.



**Сл. 3.11:** Преносна функција на дизајнираниот FIR филтер пропусник на опсег.

нивото и шум од напонот од градска мрежа на 50 Hz. Овие два проблеми можат наједноставно да се елиминираат преку употреба на филтер пропусник на опсег. За да видиме во кој опсег се движи ЕКГ сигналот најпрвин ќе направиме анализа на неговиот амплитуден спектар.

```
N = ecg.size
Nfft = 2**np.ceil(np.log(N)/np.log(2))

ecg_spec = ff.fft(ecg, Nfft)
ecg_spec = np.abs(ecg_spec)
ecg_spec = ecg_spec / N
ecg_spec = ecg_spec[0:Nfft/2+1]
ecg_spec[2:-1] = ecg_spec[2:-1] * 2
ecg_spec = 20*np.log10(ecg_spec)

f = np.linspace(0,Fs/2,Nfft/2+1)
plt.figure()
plt.plot(f, ecg_spec)
plt.grid()
```

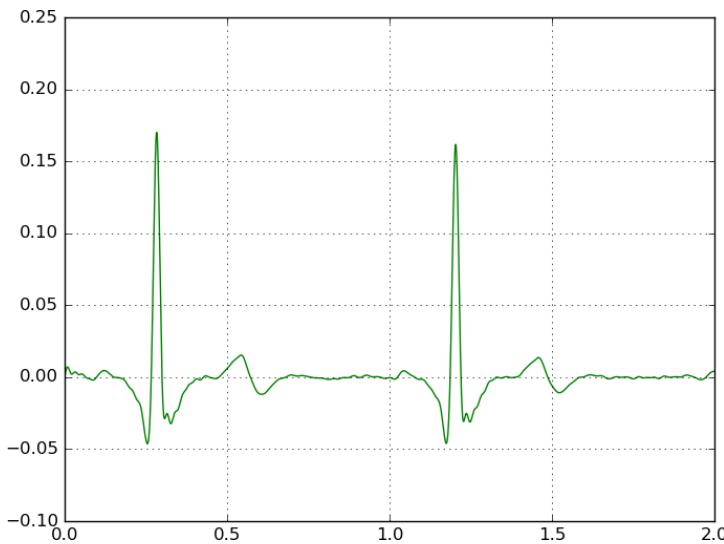
Во овој код, најпрвин се пресметува следниот поголем број  $2^n N_{FFT}$  од должината на сигналот за оптимално пресметување на FFT (22). Потоа се користи `fft` функцијата од модулот `scipy.fftpack` за пресметка на FFT (24), од која се пресметува амплитудниот спектар (25), кој се скалира со должината на сигналот (26). Поради тоа што половина од коефициентите се пресликуваат, дел од нив може да се отфрлат (28), а нивните слики треба да се удвојат (29). На крајот се пресметува логаритам од амплитудниот спектар (29) и фреквенциите на кој се поставени секој од коефициентите (31). Добиениот амплитуден спектар е прикажан на Сл. 3.10.

## Филтрирање на ЕКГ сигналот

Може да се види дека ЕКГ сигналот нема значајни компоненти над 45 Hz па можеме да дизајнираме FIR<sup>16</sup> филтер пропусник на опсег со гранични фреквенции 3 и 45 Hz. За тоа ќе ја искористиме методата за дизајн со прозорци ([Богданов, 1997](#)) кој е имплементиран во модулот `scipy.signal` со функцијата `firwin()`. Карактеристиките на филтерот може да ги визуелизирате користејќи ја функцијата `freqz()` со која го добиваме графикот на Сл. 3.11.

```
order = int(0.3 * Fs) # непарен за тип 1
if order % 2 == 0: # ако е парен
```

<sup>16</sup>Wikipedia: Finite impulse response. [https://en.wikipedia.org/wiki/Finite\\_impulse\\_response](https://en.wikipedia.org/wiki/Finite_impulse_response)



Сл. 3.12: Исфилтрирана верзија од ЕКГ сигналот.

```

order = order + 1

import scipy.signal as sig

f_l = 3
f_h = 45
b = sig.firwin(order,[f_l, f_h],pass_zero=False,nyq=Fs/2)
w, h = sig.freqz(b)
f = w / np.pi *Fs/2

plt.figure()
plt.plot(f, 20*np.log10(np.abs(h)))
plt.grid()

ecg_filt = sigfiltfilt(b, 1, ecg)
plt.figure()
plt.plot(t,ecg_filt,'g')
plt.grid()

```

Филтрирањето ќе го извршиме со функцијата `filtfilt()` која го филтрира сигналот во двете насоки со што се постигнува филтриран сигнал кој нема фазно доцнење во однос на оригиналниот сигнал. Исфилтрираниот сигнал може да се види на Сл. 3.13. Може да се види дека елиминиран е проблемот со непостојаноста на DC нивото, како и шумот од градската мрежа. Сепак, може да се види и дека QRS комплексот има помала остринा.

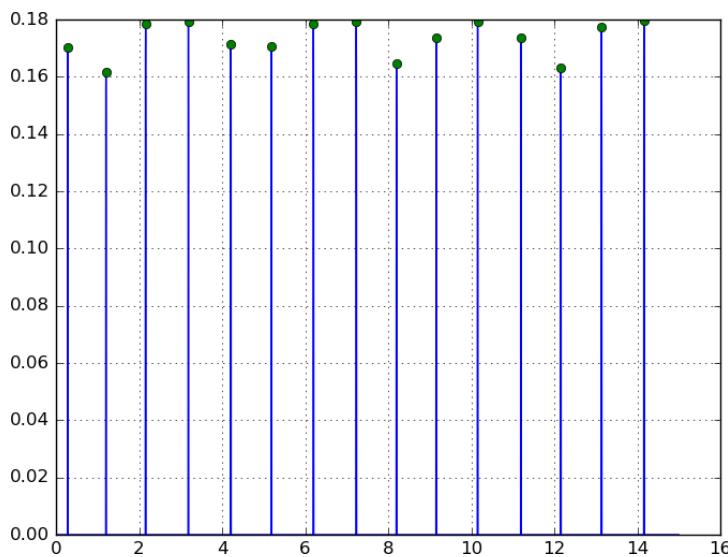
### Наоѓање на R врвовите во ЕКГ сигналот

Основната информација за брзината на отчукување на срцето може да се добие со детекција на R врвовите во ЕКГ сигналот. Наједноставен начин за тоа да се направи е да се најдат сите максимуми на филтрираниот ЕКГ сигнал кои имаат амплитуди над избран праг. Тоа е реализирано во следниот код, кој ги наоѓа врвовите како што е прикажано на Сл. ??

```

ecg_max = np.max(ecg)
threshold = ecg_max * 0.6 # 60% од max
ecg_th = np.zeros(ecg.size)
ecg_th[ecg_filt > threshold] = ecg_filt[ecg_filt > threshold]
R = np.array([], dtype='int')
for i in range(1,ecg.size):

```



Сл. 3.13: Најдени врвови во ЕКГ сигналот.

```

if ecg_th[i] > ecg_th[i-1] and \
    ecg_th[i] > ecg_th[i+1]:
    R = np.append(R,i)

plt.figure()
plt.plot(t,ecg_th)
plt.plot(t[R],ecg_th[R],'o')
plt.grid()

```

### Пресметка на бројот на отчукувања во минута

Откако ги имаме детектирано R врвовите во ЕКГ сигналот, од нив може да се пресмета моментната брзина на отчукување на срцето која вообичаено се изразува во bpm (beats per minute). За пресметување на измазната верзија од оваа брзина можеме да искористиме лизгачи прозорец за усреднување. Пресметаните моментни и усреднети вредности за bpm се прикажани на Сл. 3.14.

```

R_diff = np.diff(R)
R_diff = R_diff/Fs # во s
bpm = 60 / R_diff

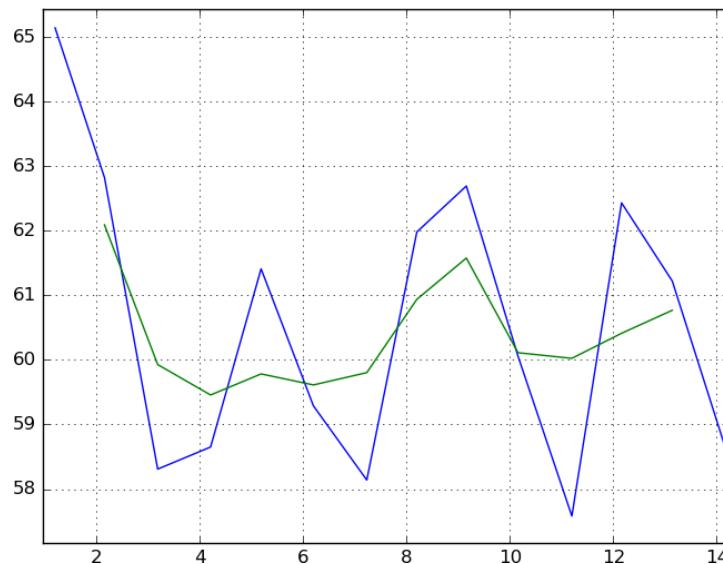
# smoothing - running average
smooth_ord = 3
win = np.ones(smooth_ord)
win = win / np.sum(win)
bpm_smooth = signal.convolve(bpm, win, mode='same' )
bpm_smooth = bpm_smooth[1:-1]

plt.figure()
plt.plot(t[R[1:]],bpm)
plt.plot(t[R[2:-1]],bpm_smooth)
plt.grid()

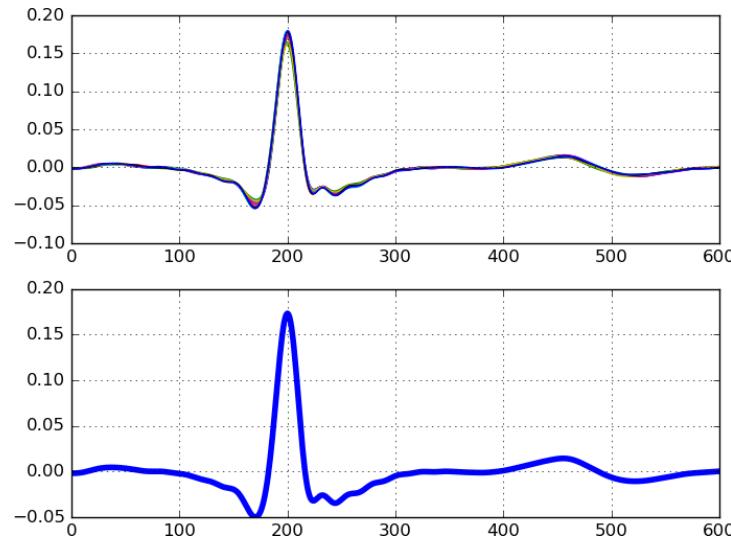
```

### Усредната репрезентација на ЕКГ сигналот

Со помош на детектираниите R врвови можеме од целиот ЕКГ сигнал да направиме усредната репрезентација на сите отчукувања на срцето. Потоа, оваа репрезентација може да се искористи за детекција на отчукувањата кои отстапуваат од неа. Оваа репрезентација е прикажана на Сл. 3.15



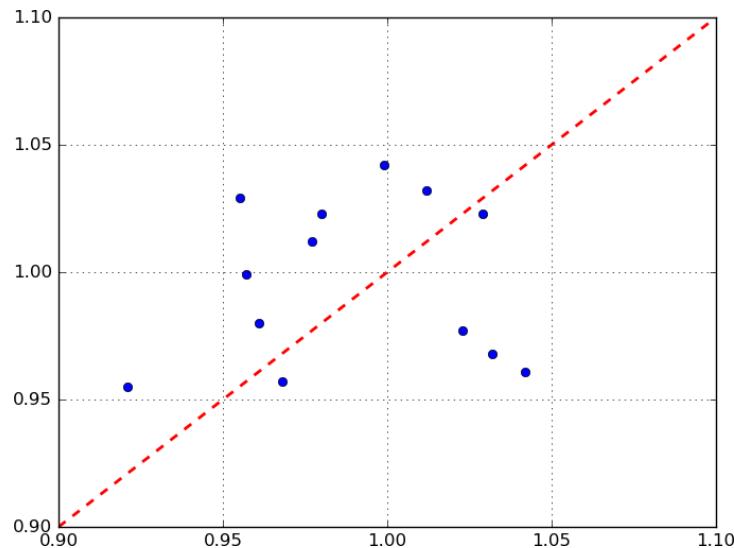
**Сл. 3.14:** Број на отчукувања на срцето во минута пресметани од ЕКГ сигналот.



**Сл. 3.15:** Усредната репрезентација на отчукувањата на срцето во ЕКГ сигналот.

```
R_left = 0.200 * Fs
R_right = 0.400 * Fs
templates = np.zeros(R_left+R_right)
for r in R:
    ecg_temp = ecg_filt[r-R_left : r+R_right]
    templates = np.vstack((templates, ecg_temp))

ecg_template = np.mean(templates[1:], 0)
plt.figure()
plt.subplot(211)
plt.plot(templates[1:].T)
plt.grid()
plt.subplot(212)
plt.plot(ecg_template, linewidth=4)
plt.grid()
```



Сл. 3.16: RR дијаграм на отчукувањата на срцето во ЕКГ сигналот.

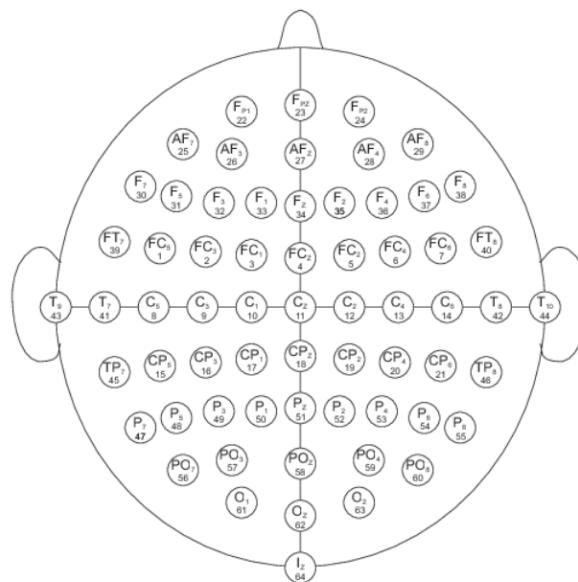
### Пресметување на RR дијаграмот од ЕКГ сигналот

Еден начин да се прикаже работата на срцето е преку RR дијаграмот. Во него времетраењето од последното отчукување се прикажува во однос на времетраењето на претпоследното отчукување, Сл. 3.16.

```
RRs = np.zeros(2)
for i in range(1, len(R_diff)):
    RR = np.array([R_diff[i-1], R_diff[i]])
    RRs = np.vstack((RRs, RR))

RRs = RRs[1:,:]

plt.figure()
plt.plot(RRs[:,0], RRs[:,1], 'o')
plt.plot([0.9, 1.1], [0.9,1.1], 'r--', linewidth=2)
plt.grid()
```



**Сл. 3.17:** Распоред на електродите според меѓународниот 10 – 10 систем.

### 3.7 Процесирање на ЕЕГ сигнали

Во овој дел ќе ги процесираме сигналите добиени со ЕЕГ. За таа цел ќе искористиме снимки објавени во рамките на ЕЕГ датасетот на реални и замислени моторни движења (Schalk et al., 2004), наменет за развој на БКИ, а достапен на страната на Физионет (Goldberger et al., 2000).<sup>17</sup>

Датасетот се состои од 1500 ЕЕГ снимки од 109 доброволци, секоја со траење од 1 до 2 минути. Секој испитаник завршил 14 експерименти - 2 едноминутни контролни снимки во мирување со очи отворени и затворени, и 3 двоминутни снимки правејќи една од 4 задачи:

- отворање и затворање на лева и десна шака,
- замислено отворање и затворање на лева и десна шака,
- отворање и затворање на двете дланки или стапала, и
- замислено отворање и затворање на двете дланки или стапала.

Снимките се направени со 64-канален ЕЕГ уред и фреквенција на семплирање од 160 Hz. Распоредот на електродите е направен според меѓународниот 10 – 10 систем прикажан на Сл. 3.17. Снимките вклучуваат и анотации на дадените наредбите:

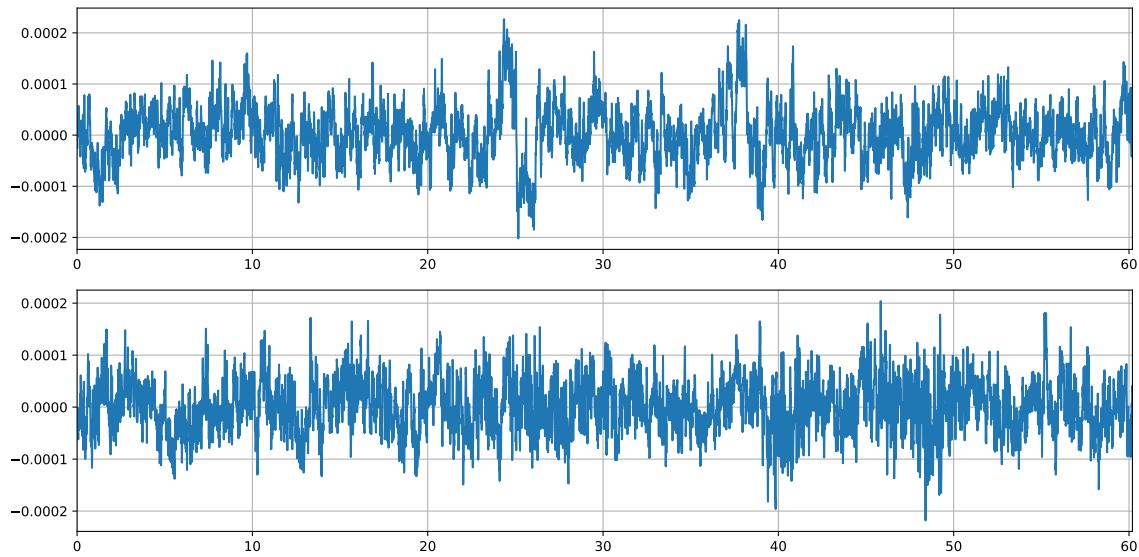
- T0 – означува неактивност,
- T1 – означува затворање или замислено затворање, и
- T2 – означува отворање или замислено отворање.

#### Вчитување на ЕЕГ сигналите

Во овој дел од вежбата ќе вчитаме две контролни снимки од базата и ќе пробаме да потврдиме преку присуството на изразени алфа бранови дали испитаникот ги има очите отворени или затворени. ЕЕГ снимките се зачувани во европскиот дата формат (ЕДФ+)<sup>18</sup> кој е направен за размена и зачувување на медицински сигнали објавен првпат во 1992 г. ЕДФ+ е надградба која овозможува зачувување на испрекинати сигнали, како и додатни информации во видот на анотации, стимули или настани.

<sup>17</sup>EEG Motor Movement/Imagery Dataset <http://www.physionet.org/pn4/eegmmidb/>

<sup>18</sup>анг. European Data Format (EDF).



**Сл. 3.18:** Временски облик на ЕЕГ сигналите за очи отворени (горе) и очи затворени (долу).

За вчитување на ЕДФ+ фајловите ќе го искористиме моќниот и слободен пакет за обработка, визуелизација и анализа на неврофизиолошки податоци MNE<sup>19</sup> кој е направен за работа со ЕЕГ сигнали. Читаните сигнали ќе ги зачуваме во **пикл**<sup>20</sup> фајл кој може потоа лесно да го вчитаме за анализа.

```

import numpy as np
import mne
import pickle

# %% очи отворени
file_name = 'S001R01.edf'
data = mne.io.read_raw_edf(file_name, preload=True)
eegs = data.get_data()
info = data.info
fs = info['sfreq']
channels = data.ch_names
channels = [x.replace('.', '') for x in channels]
t = np.arange(eegs.shape[1])/fs
eeg_eyes_open = eegs[channels.index('Cz'), :]

# %% очи затворени
file_name = 'S001R02.edf'
data = mne.io.read_raw_edf(file_name, preload=True)
eegs = data.get_data()
eeg_eyes_closed = eegs[channels.index('Cz'), :]

# %% save as pickle
with open('eeg_sample.pkl', 'wb') as f:
    data = fs, eeg_eyes_open, eeg_eyes_closed
    pickle.dump(data, f, -1)

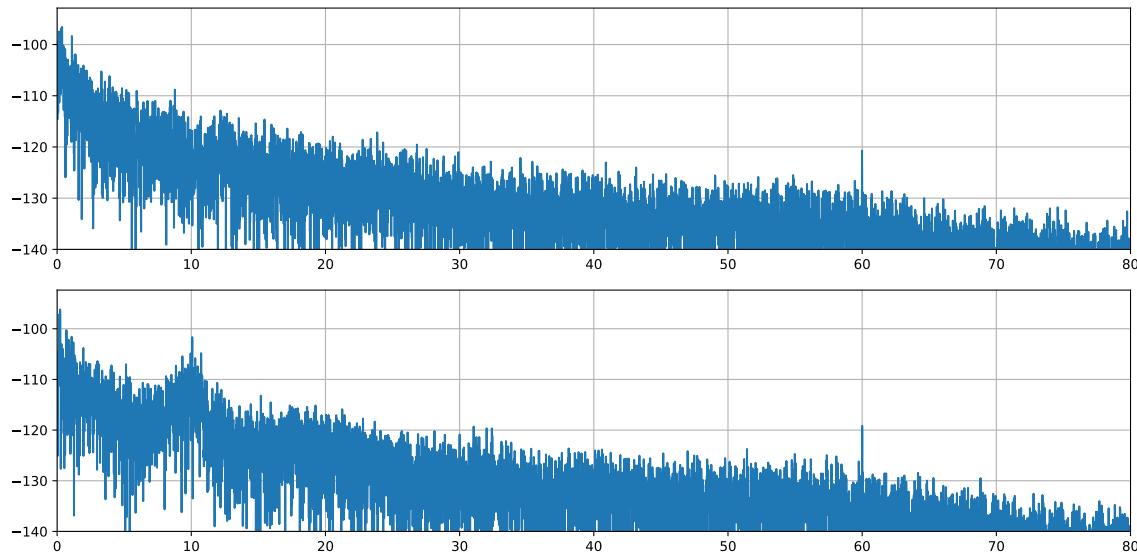
```

### Временски и спектрален облик на ЕЕГ сигналите

Да ги прикажеме двета ЕЕГ сигнали во временски и во спектрален домен, Сл. 3.18 – 3.20.

<sup>19</sup>MNE – MEG + EEG Analysis & Visualisation <https://martinos.org/mne>.

<sup>20</sup>анг. *pickle*.



**Сл. 3.19:** Спектрална содржина на ЕЕГ сигналите за очи отворени (горе) и очи затворени (долу).

```

import numpy as np
from matplotlib import pyplot as plt
from scipy import signal as sig
import pickle
import bme

# %% вчитај ги ЕЕГ сигналите
with open('eeg_sample.pkl', 'rb') as f:
    data = pickle.load(f)

fs, eeg_open, eeg_closed = data
t = np.arange(eeg_open.size) / fs

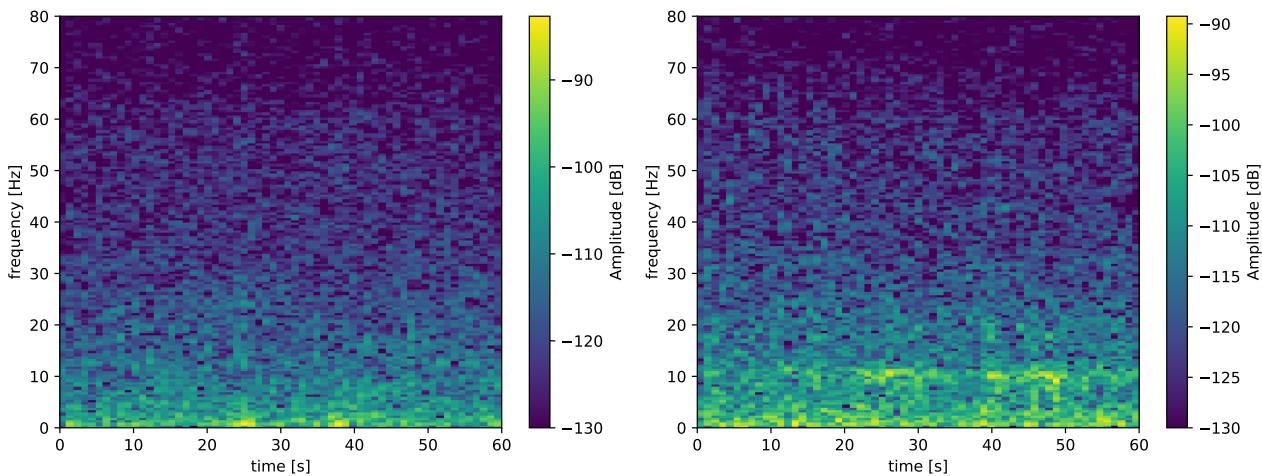
# %% прикажи во време
plt.figure()
plt.subplot(211)
plt.plot(t, eeg_open)
plt.grid()
plt.subplot(212)
plt.plot(t, eeg_closed)
plt.grid()

# %% прикажи спектар
f, eeg_closed_amp = bme.get_spectrum(fs, eeg_closed)
f, eeg_open_amp = bme.get_spectrum(fs, eeg_open)

plt.figure()
plt.subplot(211)
plt.plot(f, eeg_open_amp)
plt.grid()
plt.subplot(212)
plt.plot(f, eeg_closed_amp)
plt.grid()

# %% прикажи спектрограм
t_frame, f_frame, eeg_open_spec = bme.get_spectrogram(fs, eeg_open, n_win=320)
t_frame, f_frame, eeg_closed_spec = bme.get_spectrogram(fs, eeg_closed, n_win=320)

```



Сл. 3.20: Спектрограм на ЕЕГ сигналите за очи отворени (лево) и очи затворени (десно).

```
bme.show_spectrogram(t_frame, f_frame, eeg_open_spec)
bme.show_spectrogram(t_frame, f_frame, eeg_closed_spec)
```

Од сликите може да се забележи дека навистина има повеќе енергија во алфа опсегот 8 – 13 Hz во ЕЕГ сигналот добиен со очи затворени, отколку во оној добиен со очи отворени. Ова го потврдува очекуваното. Од спектарот иста така може да се види дека во сигналите е останата компонента на 60 Hz која соодветствува на напонот на градска мрежа во САД.

### Предфилтрирање на ЕЕГ сигналот

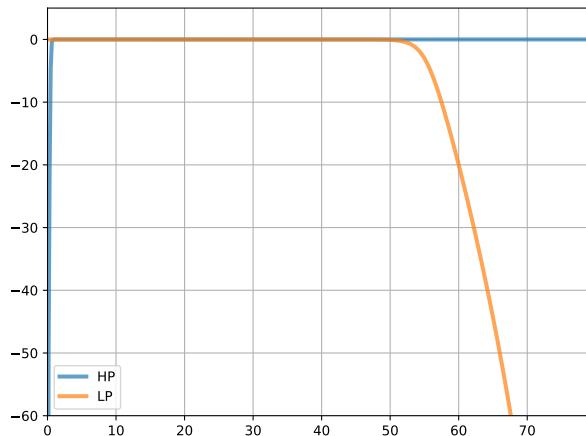
Во овој дел ќе направиме предфилтрирање на ЕЕГ сигналите со цел на отстранување на деловите кои се надвор од интерес за анализата. Така, најпрвин ќе употребиме високопропусен филтер со гранична фреквенција од 0,5 Hz со што ќе ја исфрлимиме еднонасочната компонента, т.е. DC офсетот, и нискофреквенциските компоненти кои соодветствуваат на поместување, односно дрифт на DC компонентата.

```
f_l = 0.5
order = 5
b, a = sig.iirfilter(order, f_l/(fs/2), btype='high', ftype='butter')
f, h = sig.freqz(b, a, fs=fs)
plt.figure()
plt.plot(f, 20*np.log10(np.abs(h)))
eeg_open_hp = sig.lfilter(b, a, eeg_open)
eeg_closed_hp = sig.lfilter(b, a, eeg_closed)
```

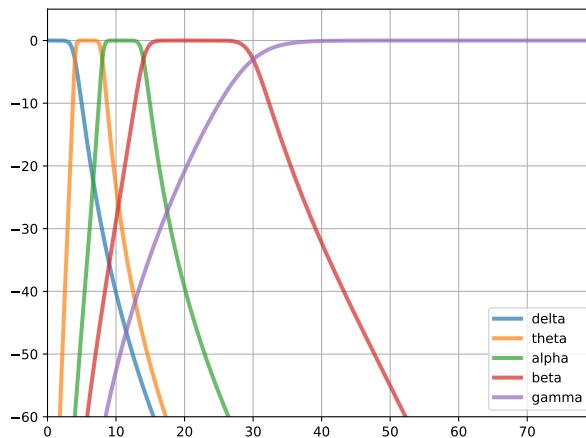
Следно ќе дизајнираме нископропусен филтер со кој ќе ги исфрлимиме компонентите над 60 Hz вклучувајќи го тута и шумот од напонот на градската мрежа.

```
r = 9
b, a = sig.iirfilter(order, f_h/(fs/2), btype='low', ftype='butter')
f, h = sig.freqz(b, a, fs=fs)
plt.figure()
plt.plot(f, 20*np.log10(np.abs(h)))
eeg_open_filt = sig.lfilter(b, a, eeg_open_hp)
eeg_closed_filt = sig.lfilter(b, a, eeg_closed_hp)
```

Амплитудната карактеристика на двата филтри е прикажана на Сл. 3.21.



**Сл. 3.21:** Амплитудната карактеристика на високопропусниот и нископропусниот филтер за претпроцесирање на ЕЕГ сигналите.



**Сл. 3.22:** Амплитудни карактеристики на банката на филтри за делење на ЕЕГ сигналот на подопсези.

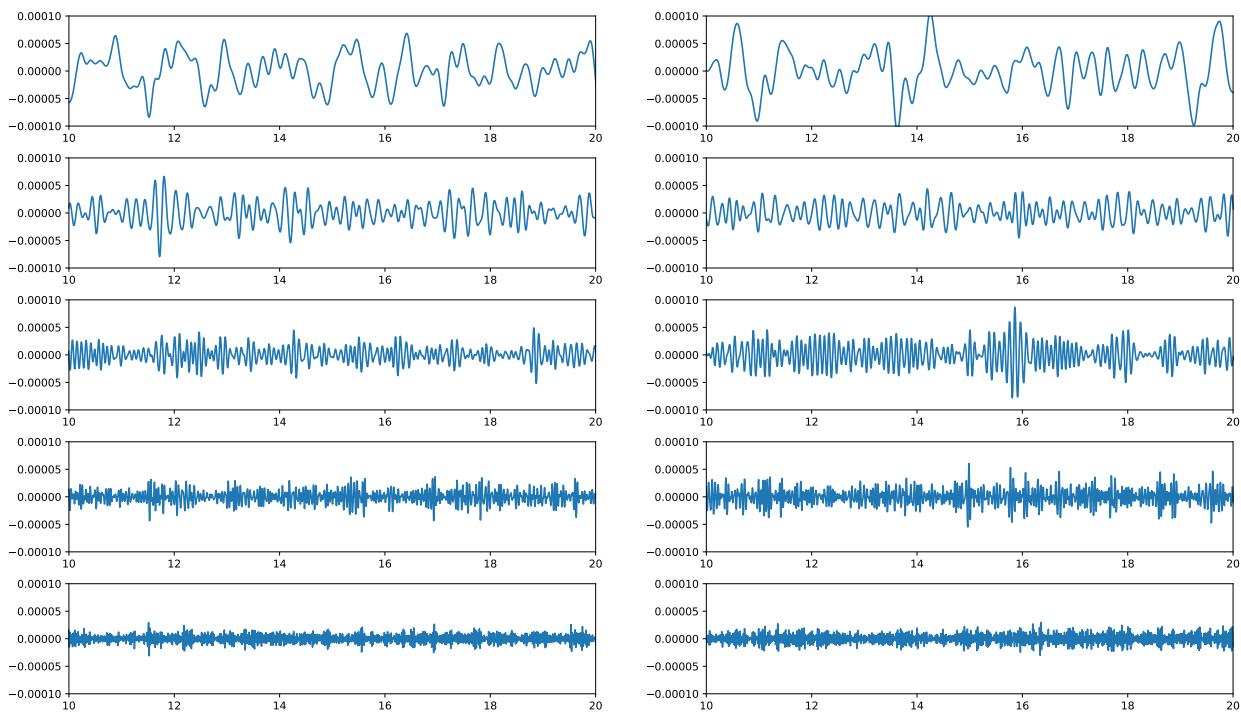
### Поделба на ЕЕГ сигналот во подопсези

Во овој дел ќе дизајнираме банка на филтри која ќе ги подели ЕЕГ сигналите на стандардните ЕЕГ подопсези: делта, тета, алфа, бета и гама. Амплитудните карактеристики на филтрите се прикажани на Сл. 3.22.

```
b_delta, a_delta = sig.iirfilter(5, 4, btype='low', fs=fs)
b_theta, a_theta = sig.iirfilter(5, [4, 8], btype='band', fs=fs)
b_alpha, a_alpha = sig.iirfilter(5, [8, 13], btype='band', fs=fs)
b_beta, a_beta = sig.iirfilter(5, [13, 30], btype='band', fs=fs)
b_gamma, a_gamma = sig.iirfilter(5, 30, btype='high', fs=fs)

filter_bank = [
    (b_delta, a_delta),
    (b_theta, a_theta),
    (b_alpha, a_alpha),
    (b_beta, a_beta),
    (b_gamma, a_gamma),
]
bands = 'delta theta alpha beta gamma'.split()
```

```
# %% исцртaj ја амплитудната карактеристика на банката
plt.figure()
for (b, a), band in zip(filter_bank, bands):
```



**Сл. 3.23:** Подопсези на ЕЕГ сигналот (од горе надолу): делта, тета, алфа, бета и гама, за очи отворени (лево) и очи затворени (десно).

```
f, h = sig.freqz(b, a, fs=fs)
plt.plot(f, 20*np.log10(np.abs(h)), lw=3, alpha=.7, label=band)

plt.grid()
plt.legend()
plt.axis([0, f[-1], -60, 5])
plt.tight_layout()
```

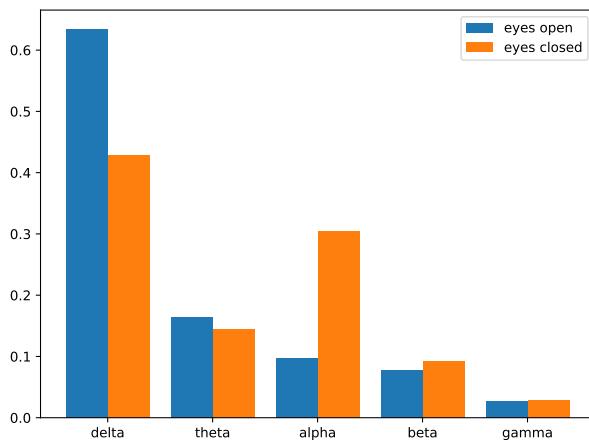
Да ги исфилтрираме ЕЕГ сигналите со банката на филтри и да ги прикажеме добиените сигнали.

```
eeg_open_bands = []
eeg_closed_bands = []
for b, a in filter_bank:
    eeg_open_bands.append(sig.lfilter(b, a, eeg_open_filt))
    eeg_closed_bands.append(sig.lfilter(b, a, eeg_closed_filt))
```

```
# %% плотирање на опсези за очи отворени
plt.figure(figsize=(10, 10))
for i, eeg in enumerate(eeg_open_bands):
    plt.subplot(5, 1, i+1)
    plt.plot(t, eeg)
    plt.axis([10, 20, -1e-4, 1e-4])
plt.tight_layout()
```

```
# %% плотирање на опсези за очи затворени
plt.figure(figsize=(10, 10))
for i, eeg in enumerate(eeg_closed_bands):
    plt.subplot(5, 1, i+1)
    plt.plot(t, eeg)
    plt.axis([10, 20, -1e-4, 1e-4])
plt.tight_layout()
```

Може да видиме од сигналите прикажани на Сл. ??, дека навистина имаме поизразени алфа



**Сл. 3.24:** Релативна моќност во подопсезите на ЕЕГ сигналите.

бранови во случајот кога очите на испитаниот се затворени.

### Пресметка на моќноста во подопсезите на ЕЕГ сигналот

За да ја квантфицираме разликата во подопсезите на ЕЕГ сигналите во двата случаји ќе ја пресметаме моќноста на сигналот во секој подопсег. Истите ќе ги нормализираме се вкупната моќност на ЕЕГ сигналот за да се добие релативната моќност за секој од подопсезите. Пресметаните моќности потоа ќе ги прикажеме на графикон за споредба.

```
eeg_open_bands = np.array(eeg_open_bands)
eeg_closed_bands = np.array(eeg_closed_bands)

eeegs_open_pow = np.sum(eeg_open_bands**2, axis=1)
eeegs_closed_pow = np.sum(eeg_closed_bands**2, axis=1)

# %% нормализирај ја моќноста
eeegs_closed_pow_norm = eeegs_closed_pow / np.sum(eeegs_closed_pow)
eeegs_open_pow_norm = eeegs_open_pow / np.sum(eeegs_open_pow)

# %% прикажи ја графички
plt.figure()
plt.bar(np.arange(5) + 0.2, eeegs_open_pow_norm, width=0.4)
plt.bar(np.arange(5) + 0.6, eeegs_closed_pow_norm, width=0.4)
plt.legend(['eyes open', 'eyes closed'])
plt.xticks(np.arange(5) + 0.4, bands)
```

Како што може да видиме од графиконот прикажан на Сл. 3.24, дека моќноста во алфа подопсегот при очи затворени изнесува дури 30% од вкупната моќност на сигналот, споредена со 10% во случајот кога очите на испитаниот се отворени.

✓ **Задача за дома.** Имплементирај алгоритам за БКИ кој ќе работи врз база на промените во ЕЕГ сигналот предизвикани од затворањето на очите на корисникот.

## Поглавје 4

# Машинското учење со биомедицински сигнали

Машинското учење, а особено длабокото учење, е област која започнува да доминира во голем број на инженерски и научни области. Во биомедицинското инженерство, машинското учење не само што донело решенија за проблемите за кои немало решение или за кои решението било прекомплексно и прескало, на пр. системи базирани на експертски правила, тоа во некои области понудило и перформанси близку до човечките, особено во анализата на биомедицински слики како КТ или МР.

### 4.1 Основни парадигми во машинското учење

Машинското учење е подобласт од областа вештачка интелигенција. Тоа опфаќа модели чии параметри се адаптираат на множество на податоци низ процес наречен на тренирање, односно учење (Bishop, 2006; Müller et al., 2016). На највисоко ниво разликуваме три парадигми на машинско учење, според кои ги делимите моделите од оваа област на три категории:

- учење со надзор<sup>1</sup>,
- учење без надзор<sup>2</sup>, и
- насочено учење<sup>3</sup>.

#### Учење со надзор

Кај учењето со надзор моделите се тренираат со познати целни излезни секвенци, уште наречени основна вистина<sup>4</sup>. Според типот на податоци во излезните секвенци разликуваме два генерални типови на модели и тоа модели за:

- **класификација** – ако излезните секвенци се дискретни па може да се каже дека одговараат на затворено множество на класи или категории, и
- **регресија** – ако излезните секвенци се континуирани.

Овие два типа на модели вообичаено ја имаат истата структура; клучната разлика е во типот на излезот кој го даваат.

Примери за употреба на модели за класификација во биомедицинското инженерство се алгоритмите за: одредување на постоењето на тумор во даден биомедицински сигнал или слика,

<sup>1</sup>Анг. *supervised learning*.

<sup>2</sup>Анг. *unsupervised learning*.

<sup>3</sup>Анг. *reinforcement learning*.

<sup>4</sup>Анг. *ground truth*.

на пр. КТ или МР, малигноста на туморот, одредување на медицинска дијагноза базирана на биомедицински параметри и сигнали, контрола на уреди преку анализа на ЕЕГ сигнали, детекција на епилептични напади во ЕЕГ сигналот, препознавање на атипични отчукувања на срцето во ЕКГ сигналот, одредување на типот на човековата активност, квалитетот на сон и предвидување на пад според сигнали од акселерометри, итн. Од друга страна, примери за модели за регресија се алгоритмите за: одредување на должината на престој на пациентите во болниците, контрола на протези преку интрамускулна ЕМГ, предвидување на ризикот од различни типови на рак според бројот на купени цигари, одредување на веројатноста пациентот да преживее даден хируршки зафат, итн.

## Учење без надзор

Алгоритмите за учење без надзор се од посебно значење денес, поради широката достапност на големи множества на податоци за кои нема ознаки, односно целни секвенци. Ова е така поради тоа што процесот на анотација вообичаено се базира на рачно внесување на ознаки од луѓе експерти, што целиот процес го прави временски захтевен и скап. Најпознатите претставници на алгоритмите за учење без надзор се алгоритмите за:

- **кластерирање** – кои вршат групирање на влезните податоци во одреден број на класи кој може да биде внесен од корисникот или одреден автоматски. Најпознат алгоритам за оваа намена е алгоритамот на **K-средни вредности**<sup>5</sup> кој го дели множеството податоци во  $K$  кластери описани со средната вредност на примероците во секој кластер, наречени и центроиди, и
- **намалување на димензионалноста** – служат за визуелизација на распределбата на едно множество на повеќедимензионални податоци во две или три димензии преку нивна трансформација. Анализата на принципиелни (главни) компоненти (PCA)<sup>6</sup> е еден пример за егзактна математичка метода за оваа намена, која е и реверзибилна. Тука спаѓаат стохастични алгоритми базирани на PCA, а најпознат алгоритам за оваа намена е алгоритамот за **t-дистрибуираното стохастичко врамување на соседи** (t-SNE)<sup>7</sup>.

## Насочено учење

Во оваа категорија спаѓаат алгоритмите кои вообичаено се поистоветуваат со областа вештачка интелигенција. Тие имаат крајна цел, но таа не е доволна за директно тренирање на моделот. Тука спаѓаат алгоритмите за играње на компјутерски игри кои неодамна ги надиграа најдобрите човечки играчи<sup>8</sup>, но и алгоритмите за контрола на агенти во виртуелни симулации кои треба да научат некоја задача, на пример како оптимално да го придвижуваат човечкото тело за да стигнат од точка А до точка Б<sup>9</sup>.

Крајната цел на овие алгоритми е да победат, но таа е предалеку во иднината за директно да може да се искористи во учењето. За надминување на овој проблем, се прават стратегии за наградување на алгоритамот долж играта, со цел учењето да се насочи во правата насока. На пример, влезот на алгоритамот може да биде моменталната состојба на полето за игра, а тој треба на излез да го даде следниот потег кој би го довел во подобра позиција, односно со поголеми шанси да победи. Овие алгоритми вообичаено треба да балансираат помеѓу **истражување** и **искористување**, или експлорација и експлоатација. Тренирањето на моделите со насочено учење се одвива преку методата на залудни обиди, односно алгоритамот се пушта да игра самиот против себе милиони пати се додека не научи да победува.

<sup>5</sup>Анг. *K-means*.

<sup>6</sup>Анг. *Principal Component Analysis (PCA)*.

<sup>7</sup>Анг. *t-distributed Stochastic Neighbour Embedding*.

<sup>8</sup>Google DeepMind: Ground-breaking AlphaGo masters the game of Go

<https://www.youtube.com/watch?v=SUbqykXVx0A>

<sup>9</sup>Google's DeepMind AI Just Taught Itself To Walk <https://www.youtube.com/watch?v=gn4nRCC9TwQ>

## 4.2 Вообичаени чекори во примената на машинското учење

Постојат низа од чекори кои вообичаено се остваруваат во употребата на алгоритмите за машинско учење при решавањето на даден проблем:

- анализа на влезните податоци / сигнали / слики – во оваа фаза се анализира типот на влезните податоци, се одредува дали постојат некомплетни податоци, се прави визуелизација на нивната распределба, и статистиката на нивните карактеристики,
- поделба на множества за тренирање, валидација и тестирање – дел од податоците се одвојува за конечна оценка на истренираниот модел, додека дел од остатокот се одвојува за континуирана валидација во фазата на тренирање,
- претпроцесирање и екстракција на обележја – во оваа фаза се прави скалирање или нормализација на влезните податоци, притоа треба да се води особена сметка на поделбата на податоците на множества за тренирање/валидација/тестирање. Во случај на употреба на влезни податоци со голем број на корелирани димензии, како и во случај кога се работи со биомедицински сигнали, вообичаено во оваа фаза се врши екстракција на обележја<sup>10</sup>. За ова може да се применат методи како PCA, пресметување на спектар, спектрограми, итн.
- тренирање на моделот – оптимизација на параметрите на моделот со употреба на множеството за тренирање и валидација, види Поглавје 4.3,
- конечна евалуација на моделот – ова се прави со одвоеното множество за тестирање со цел да се оцени како работи истренираниот модел за множество од нови, не видени влезни податоци.

## 4.3 Тренирање на моделите за машинско учење

Тренирањето на моделот, односно нагодувањето на неговите параметри се врши со помош на множеството на тренирање. Овој процес се изведува **итеративно**. Овде ќе претпоставиме дека се работи за алгоритам за учење со надзор, за кој се познати точните вредности кои сакаме да ги добијеме на неговиот излез. Во секоја итерација на учењето, се пресметува грешката која моделот ја прави врз база на познатите излези и неговите параметри се адаптираат во насока на намалување на оваа грешка. Чекорот со кој се врши адаптацијата на параметрите се нарекува и **чекор на учење**<sup>11</sup>.

Тренирањето вообичаено завршува кога: *i*) грешката на моделот не се подобрува за одредено ниво на толеранција последователно во избран број на епохи или *ii*) ако е постигнат максималниот дозволен број на итерации. Ако тренирањето запре поради постигнување на максималниот број на итерации, а не дојде до заситување на опаѓањето на грешката на моделот, тогаш велиме дека настанало **подтренирање**<sup>12</sup>, односно дека алгоритамот има потенцијал да се дотренира за подобрување на неговите перформанси.

Проблемот со овој начин на тренирање е тоа што параметрите на моделот може да се оптимизираат толку добро на множеството за тренирање што тој би ја изгубил својата способност за **генерализација**, односно точност за нови, до тогаш не видени влезни податоци. Оваа појава се нарекува **надтренирање**<sup>13</sup> и претставува посебен проблем во тренирањето на моделите за машинско учење. Постојат три главни начини тоа да се избегне:

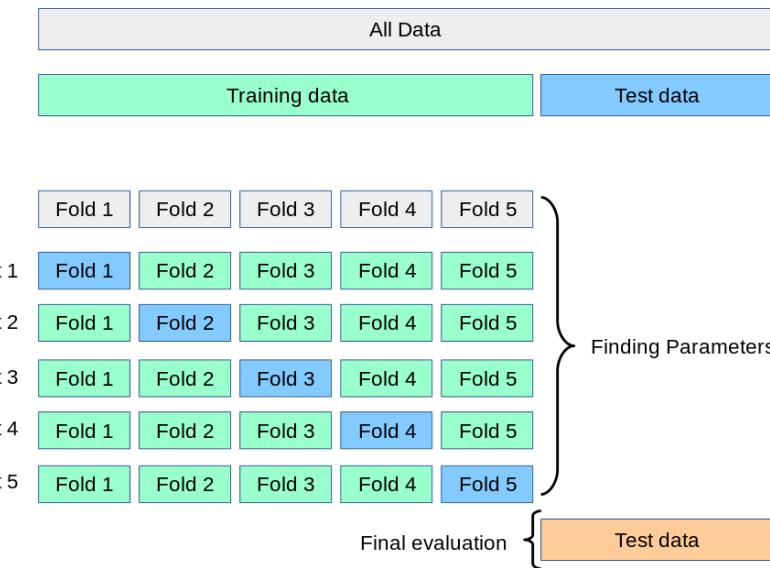
- ограничување на **моќта на моделот** – поедноставни модели не можат совршено да описват покомплексни влезни податоци и сигнали,

<sup>10</sup>Анг. *feature extraction*.

<sup>11</sup>Анг. *learning rate*.

<sup>12</sup>Анг. *underfitting*.

<sup>13</sup>Анг. *overfitting*.



Сл. 4.1:  $K$ -кратна меѓувалидација за  $K = 5$ .<sup>17</sup>

- **регуларизација** – низа на методи кои ја ограничуваат способноста на моделите да опишат комплексни податоци без намалување на нивната моќ. Степенот на регуларизација се нагодува преку **кофициент на регуларизација**,
- **рано запирање**<sup>14</sup> – постапка која ја евалуира грешката на моделот на множеството за валидација секоја итерација и го запира тренирањето кога таа ќе започне да расте.

Освен за рано запирање, множеството за валидација служи и за избор и оптимизација на глобалните параметри на моделот и процесот на учење, како на пример: архитектурата на моделот, т.е. бројот на неговите параметри, чекорот на учење, кофициентот на регуларизација итн. Овие глобални, суштински параметри се нарекуваат **хиперпараметри**.

Изборот на хиперпараметри се прави врз основа на оптимизирање на перформансите на моделот на множеството за валидација. Притоа, за да се избегне влијанието на изборот на подмножество од влезните податоци на овој процес, што е од посебно значење кога располагаме со мало количество на влезни податоци, вообичаено се прави оптимизација на хиперпараметрите во јамка за **меѓувалидација**<sup>15</sup>. Тоа се изведува во два чекора:

- се дели влезното множество на податоци на множества за тренирање и тестирање,
- множеството на тренирање се дели на  $K$  подмножества од кои секое се зема за множество за валидација во една итерација на јамката за меѓувалидација, а останатите се употребуваат за тренирање, како што е прикажано на Сл. 4.1.

Ова се нарекува  **$K$ -кратна меѓувалидација**<sup>16</sup>.

Постојат низа на проблеми кои можат да настанат при употреба на овој едноставен пристап за меѓувалидација. Секој од нив се решава со адаптација на изборот на примероци во подмножествата за меѓувалидација:<sup>18</sup>

- ако датасетот е подреден по класите кои треба да се препознаат тогаш може да се случи некоја од класите да ја има само во тест множеството; поради ова вообичаено се употребува

<sup>14</sup>Анг. *early stopping*.

<sup>15</sup>Анг. *cross-validation*.

<sup>16</sup>Анг. *K-fold cross-validation*.

<sup>17</sup>Превземено од сајтот scikit-learn – 3.1. Cross-validation: evaluating estimator performance [https://scikit-learn.org/stable/modules/cross\\_validation.html#cross-validation](https://scikit-learn.org/stable/modules/cross_validation.html#cross-validation)

<sup>18</sup>scikit-learn – 3.1. Cross-validation: evaluating estimator performance [https://scikit-learn.org/stable/modules/cross\\_validation.html#cross-validation](https://scikit-learn.org/stable/modules/cross_validation.html#cross-validation)

мешање<sup>19</sup> на датасетот пред неговата поделба,

- ако датасетот има групи, на пример испитаници, тогаш може да се случи истиот испитаник да биде и во множеството за тренирање и во тоа за тестирање; ова се надминува со одредувањето на подмножествата врз база на групите податоци, т.н. ***K*-кратна меѓувалидација по групи**<sup>20</sup> на датасетот,
- ако датасетот има нерамномерна распределба на класите, тогаш може во некое од подмножествата таа класа и воопшто да не се појави, па да не фигурира во евалуацијата; ова се надминува со **стратификација** на класите по подмножествата, т.н. ***K*-кратна меѓувалидација со стратификација**<sup>21</sup>, и
- во проблеми со временски серии во кои треба да се предвиди иднината врз база на минатите податоци, едноставната поделба на подмножества може да доведе алгоритамот да искористи информации од иднината за да направи подобра предикција; во тој случај, при оформување на подмножествата треба да се внимава на временската последователност на податоците.

Овие различни пристапи во меѓувалидацијата, како и нивната комбинација, се илустрирани на [Сл. 4.2](#).

По завршување на меѓувалидацијата, моделот со избраните хиперпараметри се тренира повторно врз целото множество на влезни податоци, без множеството за тестирање. Вака тренираниот модел се евалуира на тест множеството. И овде, за избегнување на влијанието на изборот на тест множество врз оценката на моделот, што е од посебно значење кога имаме мало множество на податоци, може да се направи меѓуоценка на моделот преку јамка на **вгнездена меѓувалидација**. Во овој пристап, целото множество на податоци најпрвин се дели на  $N$  подмножества од кои секое се користи како тест множество во надворешната јамка, а остатокот се дели на  $M$  подмножества во стандардна јамка за меѓувалидација.

## 4.4 Невронски мрежи

Еден од најпознатите модели за машинско учење, којшто лежи во основата на длабокото учење се вештачките **невронски мрежи (NN)**. Основната градбена единка на невронските мрежи се софтверските неврони чиј начин на работа е инспириран од физиологијата на невроните описана во [Поглавјето 1.3](#).

### Вештачки неврон

Секој неврон има  $K$  влезови на кои се носи влезниот вектор  $\mathbf{x} = [x_0, x_1, \dots x_{K-1}]$ . Влезовите на невронот се скалирани со тежински коефициенти, или тежини<sup>23</sup>  $w_k$ , кои можат да бидат и негативни. Активацијата на невронот  $a$  се добива преку сумирање на скалираните влезови. Конечно, излезот на невронот  $y$  се одредува преку излезната нелинеарност  $f(a)$ . Притоа, вообичаено се додава и **коефициент на поместување**<sup>24</sup> кој одговара на прагот на излезната нелинеарност.

$$y = f(a) = f\left(\sum_{k=0}^{K-1} w_k x_k + b\right) = f(\mathbf{w}\mathbf{x}^T + b) \quad (4.1)$$

<sup>19</sup>Анг. *shuffle*.

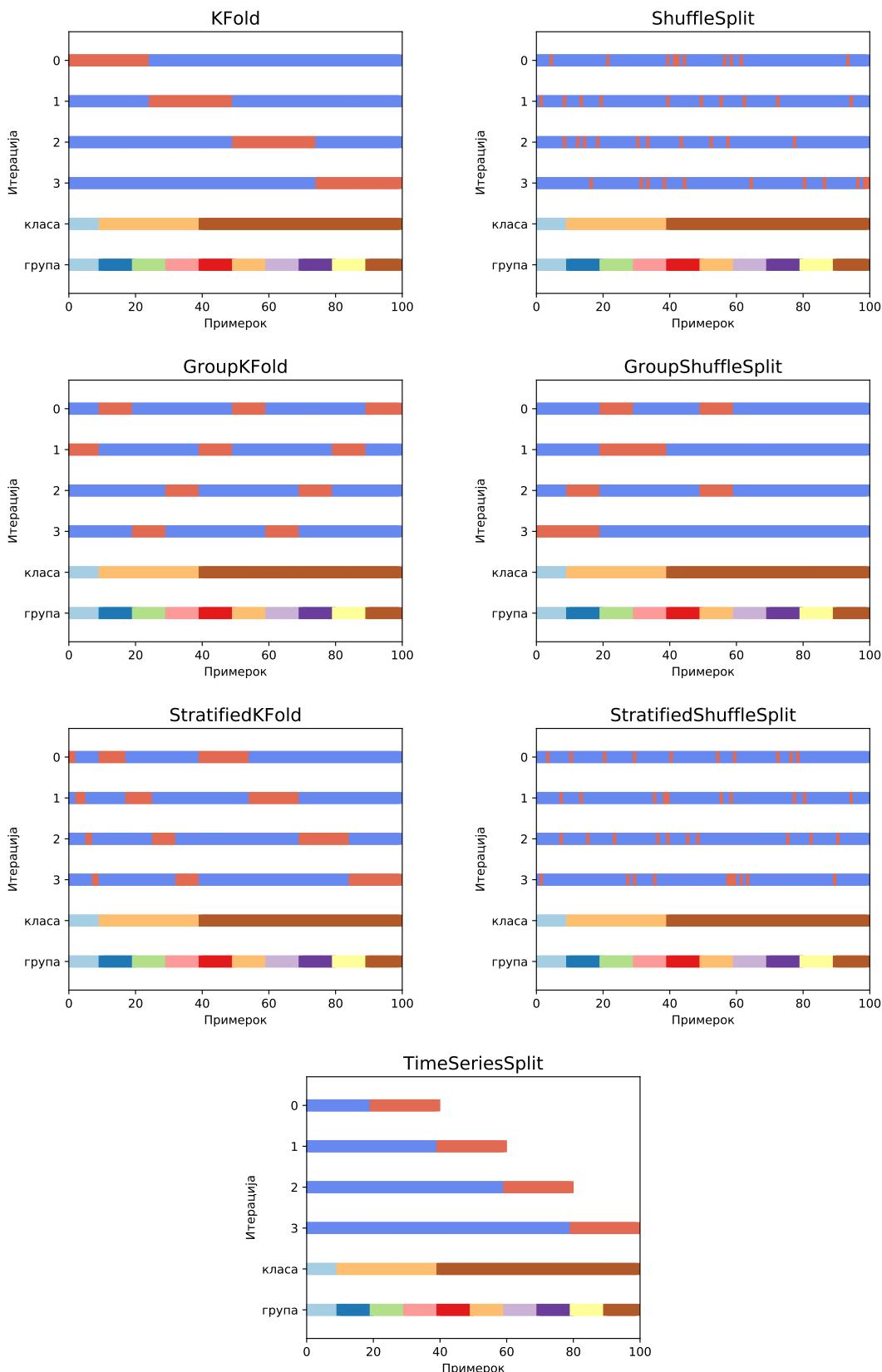
<sup>20</sup>Анг. *group k-fold cross-validation*.

<sup>21</sup>Анг. *stratified k-fold cross-validation*.

<sup>22</sup>Генерирано со прилагоден код од овој на сајтот scikit-learn – Visualizing cross-validation behavior in scikit-learn [https://scikit-learn.org/stable/auto\\_examples/model\\_selection/plot\\_cv\\_indices.html](https://scikit-learn.org/stable/auto_examples/model_selection/plot_cv_indices.html)

<sup>23</sup>Анг. *weights*.

<sup>24</sup>Анг. *bias*.

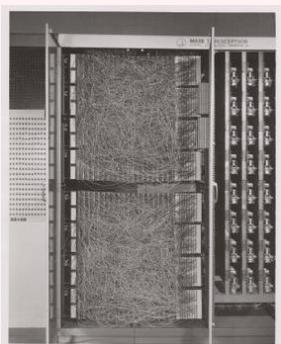


**Сл. 4.2:** Напредни методи за  $K$ -кратна меѓувалидација за  $K = 4$ . Прикажани се множеството за тренирање (сино) и множеството за тренирање (првено) за секоја итерација од јамката за меѓувалидација.<sup>22</sup>

Тука,  $\mathbf{w} = [w_0, w_1, \dots, w_{K-1}]$  е векторот на тежини на овој неврон. Ако на влез се донесат низа на  $N$  примероци од влезните податоци  $\mathbf{x}_n$ , (4.1) преминува во:

$$y_n = f(a_n) = f(\mathbf{w}\mathbf{x}_n^T + b) \quad \text{за } n = 0, 1, 2, \dots, N-1, \quad (4.2)$$

$$\mathbf{y} = f(\mathbf{a}) = f(\mathbf{w}\mathbf{X}^T + b), \quad (4.3)$$



**§ Дополнително.** Првата имплементација на вештачки неврон, наречена **перцептрон** била изработена од американскиот психолог **Франк Розенблат<sup>25</sup>** во 1958 во Лабораторијата за аеронаутика во Корнел, САД. Тој бил најпрвин изведен во софтвер на ИБМ 704, а потоа и во хардвер. Хардверската изведба била направена за препознавање на слики регистрирани со 400 фотокелии, тежините на влезовите биле реализирани со потенциометри, а нивното учење со вртење на потенциометрите со електрични мотори

каде со **X** е означена матрицата по чии редици се поставени влезните примероци  $\mathbf{x}_n$ .

<sup>25</sup>Wikipedia – Perceptron <https://en.wikipedia.org/wiki/Perceptron>

Од анонимен извор - [http://www.peoples.ru/science/psychology/frank\\_rosenblatt/](http://www.peoples.ru/science/psychology/frank_rosenblatt/), CC BY-SA 4.0, <https://commons.wikimedia.org/w/index.php?curid=64998425>

Од Source (WP:NFCC#4), Fair use, <https://en.wikipedia.org/w/index.php?curid=47541432>

Моделите за регресија базирани на употреба на еден неврон се нарекуваат **линеарна регресија**, додека оние за класификација се нарекуваат **логистичка регресија**.

## Плитки невронски мрежи

Во наједноставниот случај, невронските мрежи имаат еден скриен слој и еден излезен слој на неврони, како што е прикажано на Сл. 4.3. Ваквите модели се нарекуваат **плитки** невронски мрежи. Секој неврон од скриениот слој е поврзан со секој од коефициентите на влезниот вектор  $\mathbf{x}$ . Секој неврон од излезниот слој пак е поврзан со секој неврон од скриениот слој. Поради ова поврзување, овој вид на слоеви се нарекуваат и **целосно поврзани** односно **гости**<sup>26</sup>. Оваа едноставна архитектура сепак им овозможува на плитките невронски мрежи да моделираат било која нелинеарна функција, па тие се уште познати и како **универзални апроксиматори**.

Излезот на една плитка невронска мрежа за даден влезен вектор на податоци  $\mathbf{x}$  може да ја пресметаме преку:

$$\mathbf{y}_h = f_h(\mathbf{a}_h) = f_h(\mathbf{W}_h \mathbf{x}^T + \mathbf{b}_h), \quad (4.4)$$

$$\mathbf{y} = f_o(\mathbf{a}_o) = f_o(\mathbf{W}_o \mathbf{y}_h + \mathbf{b}_o) = f_o(\mathbf{W}_o \cdot f_h(\mathbf{W}_h \mathbf{x}^T + \mathbf{b}_h) + \mathbf{b}_o), \quad (4.5)$$

каде со  $h$  се означени параметрите и излезите добиени од скриениот слој, а со  $o$  оние од излезниот слој. Овојпат, бидејќи во секој слој може да имаме повеќе неврони, нивните тежини се распоредени долж редиците на матриците за тежини  $\mathbf{W}$  а нивните коефициенти на поместување во векторите колони  $\mathbf{b}$ .

На тој начин, излезот на мрежата се добива со процесирање на влезните податоци слој по слој се додека не се дојде до излезниот слој на мрежата.<sup>27</sup> Овој процес се нарекува и **пропагација нанапред**.<sup>28</sup>

Повторно, ако на влез се донесат низа на  $N$  примероци од влезните податоци  $\mathbf{x}_n$  добиваме:

$$\mathbf{Y}_h = f_h(\mathbf{A}_h) = f_h(\mathbf{W}_h \mathbf{X}^T + \mathbf{b}_h), \quad (4.6)$$

$$\mathbf{Y} = f_o(\mathbf{A}_o) = f_o(\mathbf{W}_o \mathbf{Y}_h + \mathbf{b}_o) = f_o(\mathbf{W}_o \cdot f_h(\mathbf{W}_h \mathbf{X}^T + \mathbf{b}_h) + \mathbf{b}_o), \quad (4.7)$$

Во рамките на една невронска мрежа, излезните нелинеарности на невроните во скриените и излезните слоеви вообичаено се разликуваат. Типичен избор за нелинеарности во скриениот слој се:

- **сигмоида**  $\sigma(a) = \frac{1}{1+e^{-a}}$  – дава излез во опсег  $0 - 1$ ,
- **тангенс хиперболикум**  $\tanh(a) = \frac{e^{2a}-1}{e^{2a}+1}$  – дава излез во опсег од  $-1$  до  $1$ , и
- **полубранов насочувач**  $ReLU(a) = \begin{cases} a & \text{ако } a > 0 \\ 0 & \text{поинаку} \end{cases}$  – предноста на употреба на полубрановиот насочувач е поедноставната пресметка на излезот на невроните, подобрата пропагација на градиентот во процесот на тренирање, како и реткоста на активација на невроните – при случајна иницијализација на тежините половина од невроните ќе дадат  $0$ . Уште една мотивација за употреба на оваа нелинеарност е биолошката аналогија која се должи на нејзината асиметричност.

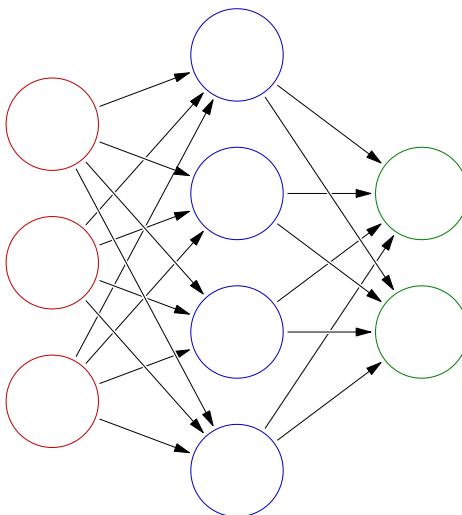
За невроните во излезниот слој вообичаено се користат:

- **сигмоида** – за класификација,

<sup>26</sup>Анг. *fully connected* и *dense*.

<sup>27</sup>Длабоките невронски мрежи имаат повеќе скриени слоеви, види понатаму во [Поглавјето](#)

<sup>28</sup>Анг. *forward pass* или *feed forward*.



**Сл. 4.3:** Плитка невронска мрежа со еден скриен слој составен од четири неврони (сино), и еден излезен слој составен од 2 неврони (зелено). Влезните податоци вообичаено се пртаат како влезен слој (црвено).<sup>29</sup>

- софтмакс  $f(a_j) = \frac{e^{a_j}}{\sum_{j=0}^{J-1} e^{a_j}}$  – за класификација со повеќе излезни класи  $J$ , каде  $a_j$  е активацијата на невронот кој соодветствува на класата  $j$ ; софтмакс функцијата го нормализира збирот на излезот на сите излезни неврони, па може да се каже дека ни дава апроксимација на веројатноста на секоја од класите  $f(a_j) \approx P(y = j | \mathbf{a})$ ,
- линеарна  $f(a) = a$  – кај моделите за регресија.

## Длабоко учење

Со додавање на повеќе скриени слоеви во невронската мрежа се добиваат **длабоки невронски мрежи (ДНН)**<sup>30</sup>. Длабокото учење е подобласт во машинското учење која ги опфаќа моделите базирани на повеќеслојни **вештачки невронски мрежи** (Goodfellow et al., 2016). Иако плитките невронски мрежи претставуваат универзални апроксиматори, тоа важи само кога моќта на мрежата, одредена од бројот на неврони во скриениот слој, е доволно голема. Се покажува дека додавање на неврони во скриениот слој, односно додавање на широчина на мрежата не е толку ефикасно како додавање на длабочина на мрежата.

Невронските мрежи кои постигнуваат денес надчовечки перформанси имаат и до 1000 скриени слоеви (He et al., 2016). Поради комплексноста на тренирањето на милионите параметри на длабоките невронски мрежи, тие доживуваат процут дури на почетокот од XXI век, по општата достапност на графичките процесори кои овозможуваат паралелна имплементација на алгоритмите за тренирање.

Постојат три основни подвидови на длабоките невронски мрежи:

- обични **длабоки невронски мрежи** – овие се нарекуваат и **повеќениковски перцептрони (МЛП)**<sup>31</sup> во чест на првиот перцептрон,
- конволуциски **невронски мрежи (КНМ)** – базирани на конволуција со филтри се употребуваат за процесирање на 1Д, 2Д и 3Д сигнали, а речиси секогаш за обработка на слики,
- рекурентни **невронски мрежи (РНМ)** – базирани на повратна врска на невроните од скриените слоеви што им дава еден вид на меморија за претходните примероци, се

<sup>29</sup>Модифицирано од Glosser.ca - Own work, Derivative of File:Artificial neural network.svg, CC BY-SA 3.0, <https://commons.wikimedia.org/w/index.php?curid=24913461>

<sup>30</sup>Анг. Deep Neural Networks (DNNs).

<sup>31</sup>Анг. multi layer perceptron.

употребуваат за процесирање на 1Д сигнали,

- рекурентни конволуциски невронски мрежи (РКНМ) – комбинација од РНМ која го употребува излезот од КНМ, се употребуваат за процесирање на 4Д сигнали, пред сè видео.

## Тренирање на невронските мрежи

Проблемот на тренирање на невронските мрежи се сведува на промена на тежините и коефициентите на поместување на секој од невроните во насока на намалување на грешката која ја прави мрежата (Nielsen, 2015). Еден начин тоа да се направи е со примена на алгоритамот спуштање по градиентот (ГД)<sup>32</sup>, кој е итеративен алгоритам за пронаоѓање на минимумот на дадена функција. Конкретно, за да дојдеме до минимумот на функцијата  $f(x)$  од моментната позиција  $x_i$ , треба да направиме чекор во насока спротивна на градиентот за таа позиција:

$$x^{i+1} = x^i - \frac{df}{dx} \cdot \Delta x, \quad (4.8)$$

каде со  $\Delta x$  е означен големината на чекорот кој го земаме.

За примена на ГД при тренирањето на невронските мрежи дефинираме **функција на грешка**  $\mathcal{L}(y, \tilde{y}) = \mathcal{L}(y, g(\theta, \mathbf{x}))$  која зависи од точниот, т.е. целниот излез  $y$  и излезот добиен од мрежата  $\tilde{y} = g(\theta, \mathbf{x})$ , кој пак зависи од параметрите на невронската мрежа  $\theta$  и влезниот вектор  $\mathbf{x}$ . Притоа,  $\theta = [\theta_0, \theta_1, \dots, \theta_{L-1}] = [\mathbf{W}_0, \mathbf{b}_0, \mathbf{W}_1, \mathbf{b}_1, \dots, \mathbf{W}_{L-1}, \mathbf{b}_{L-1}]$  каде  $L$  е бројот на слоеви на мрежата. Тогаш имаме:

$$\theta_l^{i+1} = \theta_l^i - \frac{\partial \mathcal{L}}{\partial \theta_l} \cdot \eta \quad \text{за } \theta_l \in \theta, \quad (4.9)$$

каде  $\frac{\partial \mathcal{L}}{\partial \theta_l}$  е парцијалниот извод на функцијата на грешка во однос на параметарот  $\theta_l$ , а  $\eta$  е чекорот на учење. Притоа, за пресметување на парцијалниот извод се користи правилото за пресметување на **извод на сложена функција**<sup>33</sup>, односно:

$$\frac{\partial \mathcal{L}}{\partial \theta_l} = \frac{\partial \mathcal{L}}{\partial \tilde{y}} \cdot \frac{\partial \tilde{y}}{\partial y_{L-2}} \cdots \frac{\partial y_{l+1}}{\partial \theta_{l+1}} \cdot \frac{\partial y_l}{\partial \theta_l}, \quad (4.10)$$

Каде  $y_l$  е излезот на  $l$ -от слој на мрежата. На пример, за модел составен од еден неврон, градиентот во однос на тежината  $w_0$ , ќе биде:

$$\frac{\partial \mathcal{L}}{\partial w_0} = \frac{\partial \mathcal{L}}{\partial \tilde{y}} \cdot \frac{\partial \tilde{y}}{\partial w_0}. \quad (4.11)$$

Користејќи го правилото за извод на сложена функција, пресметување на градиентот за апдејтирање на сите параметри  $\theta$  на невронската мрежа започнува со пресметка на градиентите за излезниот слој, па за последниот скриен слој и оди наназад до почетокот на мрежата. Овој процес се нарекува **пропагација наназад**.<sup>34</sup>

За ГД да може да се употреби за тренирање на невронски мрежи, мора функцијата на грешка, како и сите излезни нелинеарности на невроните во мрежата да бидат диференцијабилни. Инаку не би можел да се пресмета градиентот за секој од параметрите. Ова е причината за употреба на сигмоидата наместо пресекување на активацијата со остат праг, како што било направено во оригиналниот перцептрон, кај кого важело:

$$y = f(a) = \begin{cases} 1 & \text{ако } a > 0,5 \\ 0 & \text{поинаку} \end{cases}. \quad (4.12)$$

Денес најчесто се употребуваат следните функции за грешка:

<sup>32</sup>Анг. *gradient descent (GD)*.

<sup>33</sup>Анг. *chain-rule*.

<sup>34</sup>Анг. *backpropagation*.

- средна квадратна грешка  $MSE = \frac{1}{N} \sum_{n=0}^{N-1} (y - \tilde{y})^2$  – основна функција на грешка за регресија и бинарна класификација,
- меѓу-ентропија  $CE = -\frac{1}{N} \sum_{n=0}^{N-1} y \ln \tilde{y} + (1 - y) \ln(1 - \tilde{y})$  – кај модели за класификација со излезна нелинеарност сигмоида нејзиниот извод има подобри карактеристики,
- логаритамска веројатност<sup>35</sup>  $LL = -\ln \tilde{y}$  – кај моделите со софтмакс функција на излез.

Во рамките на функцијата на грешка се вклучува и дел за регуларизација. Најчесто тоа е регуларизација на  $L2$  нормата на параметрите на моделот  $\theta$ . На пример, при употреба на средната квадратна грешка, би имале:

$$L(y, \tilde{y}, \theta) = L(y, \tilde{y}) + \lambda \sum_{l=0}^{L-1} (\mathbf{W}_l^T \mathbf{W}_l + \mathbf{b}_l^T \mathbf{b}_l), \quad (4.13)$$

каде  $\lambda$  е коефициентот на регуларизација, а со  $\mathbf{W}_l^T \mathbf{W}_l$  и  $\mathbf{b}_l^T \mathbf{b}_l$  се добиваат сумите од квадратите за сите параметри на невроните од слојот  $l$ .

При тренирањето на невронски мрежи, поради тоа што вообичаено се работи со големи множества на влезни податоци, станува неисплатливо градиентот да се пресметува за сите влезни примероци. Другиот екстрем е адаптацијата на параметрите да се прави со градиентот пресметан за секој од примероците земен по случаен избор. Оваа варијанта на алгоритамот се нарекува **стохастично спуштање по градиентот (СГД)**<sup>36</sup>. Изминувањето на целото множество за тренирање се нарекува и **епоха**. Случајниот избор на влезните примероци технички се изведува преку случајно мешање на датасетот пред секоја епоха, по што следи секвенцијално земање на примероците.

Сепак, пресметката на градиентот по примерок не дава добра естимацијата на вистинскиот градиент на функцијата на грешка. Поради тоа, најчесто се употребува компромисно решение во кое се зема подмножество, или **купче**<sup>37</sup> примероци од множеството за тренирање по случаен избор, и се врши адаптација на параметрите за пресметаниот градиент.

Оваа верзија на ГД алгоритамот се нарекува **спуштање по градиентот со купчиња (МБГД)**<sup>38</sup>. Технички постои разлика помеѓу МБГД и БГД – кај МБГД се врши апдејтирање на параметрите за секое купче, додека кај БГД тоа се прави по поминување низ целото множество за тренирање. Скоро секогаш за тренирање на невронските мрежи, а и други алгоритми за машинско учење, се употребува МБГД алгоритамот но под името СГД.

При употреба на СГД, чекорот на учење е еден од најважните параметри во тренирањето на невронските мрежи. Ако е преголем оптимизацијата може да го натфрли минимумот на функцијата на грешка, додека пак, ако е премал, на алгоритамот ќе му бидат потребни многу итерации за да заврши тренирањето. Затоа вообичаено се употребуваат **стратегии на промена** на чекорот на учење<sup>39</sup>. Една едноставна, а често употребувана, стратегија е чекорот да се намалува во тек на тренирањето. Постојат и понапредни алгоритми, каков што е **Адам** (Kingma and Ba, 2014), кои в' предвид ги земаат првиот и вториот момент, односно брзината и забрзувањето, на промената на градиентот во однос на претходните итерации за адаптација на чекорот на учење.

Еден голем проблем со употребата на СГД алгоритамот е можноста алгоритамот да заглави во локален минимум, по цена на промашување на глобалниот минимум. Овој ризик е поголем кај помалите НМ, додека кај длабоките НМ, веројатностите за постоење на локален минимум се минорни. Кај нив проблемот се сведува на заглавување на СГД во рамни региони на функцијата на грешка. Двата проблеми можат да се решат преку стратегии за промена на чекорот на учење, кои се базираат на негова реиницијализација на голема вредност по одреден број на итерации.

<sup>35</sup>Анг. *log-likelihood*.

<sup>36</sup>Анг. *stochastic gradient descent (SGD)*.

<sup>37</sup>Анг. *batch* или *mini-batch*.

<sup>38</sup>Анг. *mini batch gradient descent (MBGD)*.

<sup>39</sup>Анг. *learning rate scheduling*.

## 4.5 Препознавање на човечка активност со акселерометри

Во овој дел ќе примениме невронска мрежа за препознавање на човечка активност<sup>40</sup> врз база на сигналите добиени од акселерометарот кој денес вообичаено е вклучен во смартфоните. За таа цел ќе го искористиме датасетот направен од Лабораторијата за податочно рударење на безжични сензори на Универзитетот Фордхам, Њу Јорк, САД<sup>41</sup> описан во трудот Kwapisz et al. (2011).

Во датасетот се содржани 1.098.207 мерења од акселерометар за три типови на активности: одење, трчање, качување по скали, симнување по скали, седење и стоење. Акселерометарот мери три оски  $x$ ,  $y$  и  $z$ , и го регистрира и земјиното забрзување, при што сигнал со вредност  $10 = 1 \text{ g} = 9,81 \text{ m/s}^2$ .<sup>42</sup> Фреквенцијата на семплирање на сигналите  $f_s$  е 20 Hz. Во датасетот се поместени сигнални снимени со 30 испитаници, што ќе биде важно за неговата поделба на множества за тренирање и тестирање.

Сигналите се зачувани во текстуалниот фајл `WISDM_ar_v1.1_raw.txt` во формат на податоци одвоени со запирка (ЦСВ)<sup>43</sup> па за негово вчитување и анализа ќе го искористиме модулот Пандас<sup>44</sup>. Притоа, Пандас ќе ги прочита сигналите во податочна рамка<sup>45</sup>.

```
import numpy as np
from matplotlib import pyplot as plt
import pandas as pd

file_name = 'data/har_wisdm/WISDM_ar_v1.1_raw.txt'
columns = 'id act time x y z'.split()
data = pd.read_csv(file_name,
                   header=None,
                   names=columns,
                   lineterminator=';')

fs = 20
data = data.dropna()
```

### Анализа на влезните податоци

За запознавање со сигналите ќе прикажеме временски отсекоци од 10 s од истите за секоја од типовите на активност.

```
activities = data.act.unique().tolist()
for axis in 'x y z'.split():
    plt.figure(figsize=(7, 10))
    for i, activity in enumerate(activities):
        mask_act = data.act == activity
        y = data.loc[mask_act, axis].values
        plt.plot(np.arange(200)/fs, y[200:400] + i*20, label=activity)

    plt.grid()
    plt.legend(loc='upper right')
    plt.title(f'{axis}-оска')
    plt.tight_layout()
```

Добиениот графикон е прикажан на Сл. 4.4. Може да видиме дека сигналите од акселерометарот за  $y$  оската, која е всушност вертикалната оска на човекот, има најголеми вредности за

<sup>40</sup>Анг. *human activity recognition (HAR)*.

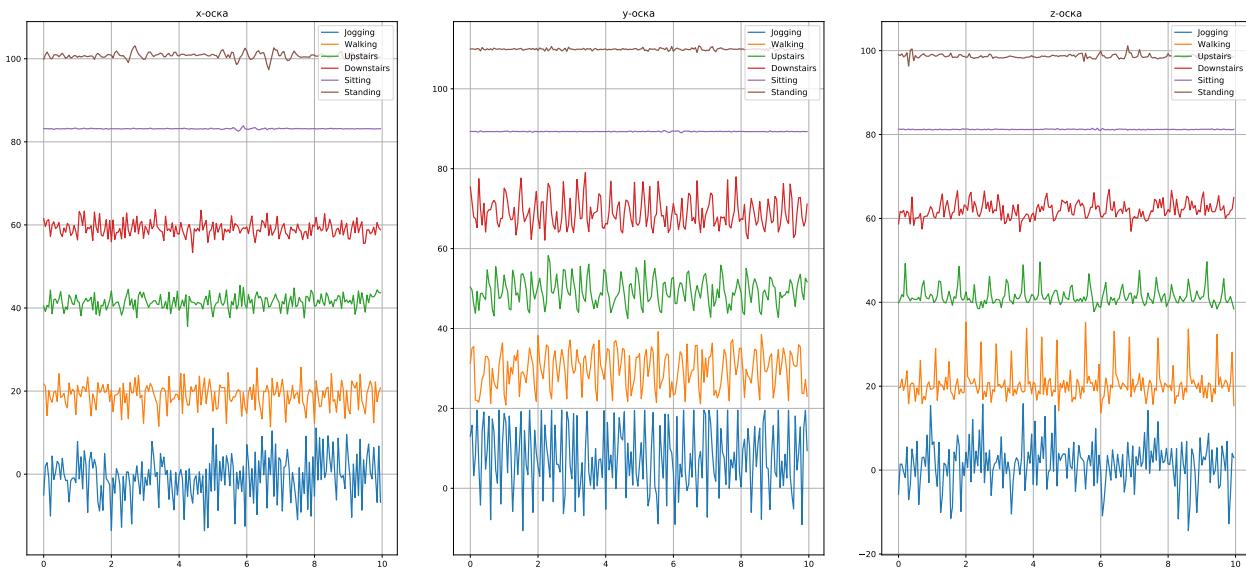
<sup>41</sup>Wireless Sensor Data Mining (WISDM) Lab, Fordham University, New York, USA <http://www.cis.fordham.edu/wisdm/>

<sup>42</sup>Токму мерењето на земјиното забрзување е она што ни овозможува да разликуваме помеѓу седење и стоење.

<sup>43</sup>Анг. *comma separated value (CSV)*.

<sup>44</sup>Pandas <http://pandas.pydata.org/>

<sup>45</sup>Анг. *dataframe*.



**Сл. 4.4:** Временски облик на сигналите добиени од акселерометарот за 10 s на вчитаниот ЕМГ сигнал долж трите осци.

забрзувањето.

### Поделба на податоците на множества за тренирање и тестирање

Во оваа демонстрација на процесот на примена на машинското учење ќе направиме само едноставна поделба на податоците на множества за тренирање и тестирање. За таа цел ќе го искористиме групирањето на податоците по субјект во датасетот. Ќе одвоиме 20% од учесниците за креирање на множеството за тест.

```
data.id = data.id.astype(int)
subjects = data.id.unique().tolist()
# измешај ги
np.random.seed(42)
subjects = np.random.shuffle(subjects)
# подели ги
test_sub_perc = 0.2
test_sub_nr = int(len(subjects) * test_sub_perc)
train_subjects = subjects[:-test_sub_nr]
test_subjects = subjects[-test_sub_nr:]
```

Притоа за случајно мешање на испитаниците го искористивме `random` модулот на Нумпај кој содржи **генератор на псевдо-случајни броеви**<sup>46</sup>. Тоа е машина со конечен број на состојби кој генерира секвенца на броеви чија статистика е како на случајна низа на броеви, иако начинот на кој се генерираат е детерминистички. Така, со поставување на состојбата на генераторот со наредбата `np.random.seed`, при секое извршување на кодот ќе ги добиеме истите резултати. Во конкретниот случај тоа ќе биде истата поделба на датасетот. Ова е вообичаена практика во областа машинско учење во која случајноста игра голема улога, на пр. при иницијализација на параметрите на моделот, со што се обезбедува репродуцибилност на резултатите.

### Екстракција на карактеристики

Датасетот кој го користиме веќе доаѓа со пресметани карактеристики за сигналите од акселерометарот поместени во фајлот `WISDM_ar_v1.1_transformed.arff`. Тие се пресметани за прозорци од 10 s од сигналите на акселерометарот и претставуваат воглавно статистички

<sup>46</sup>Анг. *pseudo-random number generator (PRNG)*.

податоци како средна вредност, средна апсолутна девијација, стандардна девијација, хистограм, и растојание помеѓу врвови. По 15 статистички податоци се пресметани за секоја оска и дополнително е додадена нивната квадратна средина. Така, вкупниот број на карактеристики за секои 10 s од сигналот е 46.

Во нашата анализа, наместо пресметаните карактеристики, ќе го пресметаме спектарот на сигналите за  $y$  оската со прозорци од 10 s, користејќи го нашиот код за пресметка на спектрограми.

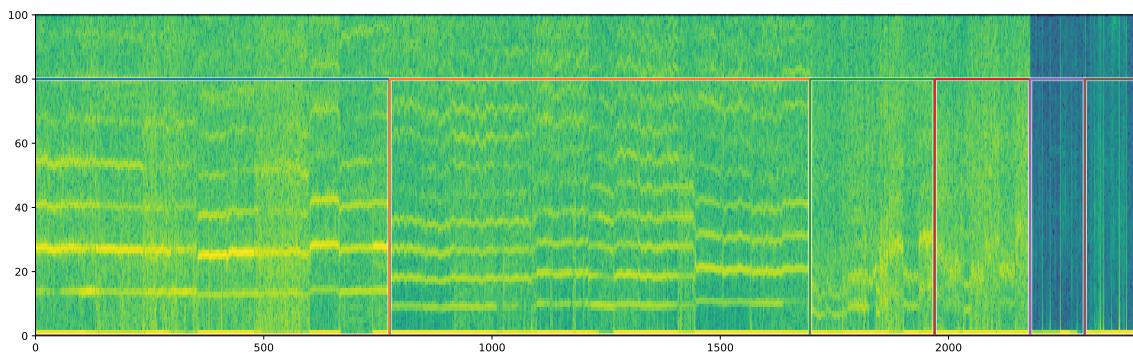
```
import bme

x_train = None
y_train = None
x_test = None
y_test = None

for i_act, act in enumerate(activities):
    mask_act = data.act == act
    for sub in train_subjects:
        mask_sub = data.id == sub
        mask = mask_act & mask_sub
        # провери дали субјектот ја има снимено оваа активност
        if sum(mask) == 0:
            print(f'No {act} data for sub {sub}')
            continue
        x = data.loc[mask, 'y'].values
        _, __, x_feat = bme.get_spectrogram(fs, x, n_win=200)
        x_feat = x_feat.T
        # провери дали има сигнал во секоја од рамките
        mask_thresh = np.all(x_feat > -100, axis=1)
        x_feat = x_feat[mask_thresh, :]
        # претстави ја класата како бинарна низа
        y = np.zeros((x_feat.shape[0], 6))
        y[:, i_act] = 1
        if x_train is None:
            x_train = x_feat
            y_train = y
        else:
            x_train = np.concatenate((x_train, x_feat), axis=0)
            y_train = np.concatenate((y_train, y), axis=0)

    for sub in test_subjects:
        mask_sub = data.id == sub
        mask = mask_act & mask_sub
        if sum(mask) == 0:
            print(f'No {act} data for sub {sub}')
            continue
        x = data.loc[mask, 'y'].values
        _, __, x_feat = bme.get_spectrogram(fs, x, n_win=200)
        x_feat = x_feat.T
        mask_thresh = np.all(x_feat > -100, axis=1)
        x_feat = x_feat[mask_thresh, :]
        y = np.zeros((x_feat.shape[0], 6))
        y[:, i_act] = 1
        if x_test is None:
            x_test = x_feat
            y_test = y
        else:
            x_test = np.concatenate((x_test, x_feat), axis=0)
            y_test = np.concatenate((y_test, y), axis=0)
```

При генерирање на карактеристиките може да видиме дека не секој од испитаниците ги има



**Сл. 4.5:** Карактеристики екстрагирани од множеството за тестирање со пресметка на спектрарот за рамки од 10 s од сигналот за  $y$  оската. Врз нив се исцртани шесте класи: одење, трчање, качување по скали, симнување по скали, седење и стојење.

снимено сите активности. За визуелизација на спектрограмите за различните класи и потврда дека тие можат да се искористат за нивно разликување ќе ги прикажеме на графикон преклопени со класата на активност. Може да видиме на Сл. 4.5 дека постојат јасни разлики во спектрите на сигналите за различните активности па може да очекуваме да имаме добра предикција од нашиот класификатор.

```
x_axis = np.arange(x_test.shape[0])
y_axis = np.arange(x_test.shape[1])
bme.show_spectrogram(x_axis, y_axis, x_test.T)
plt.plot(y_test*80, lw=2)
```

## Тренирање на моделот

За оваа анализа ќе ја искористиме имплементацијата на невронска мрежа за класификација во класата `MLPClassifier` во модулот `neural_network` на пакетот за машинско учење **Сајкит-лерн**.<sup>47</sup>

```
from sklearn import neural_network
from sklearn import metrics

mlp = neural_network.MLPClassifier(
    hidden_layer_sizes=(100, 20, 10), # број на неврони во скриените слоеви
    activation='relu', # нелинеарност на невроните во скриените слоеви
    learning_rate_init=0.01, # чекор на учење
    alpha=1e-4, # коефициент на L2 регуларизација
    solver='adam', # алгоритам за тренирање
    max_iter=3000, # максимален број на итерации
    tol=1e-9, # толеранција за заситување на тренирањето
    n_iter_no_change=20, # запирање на тренирањето заради заситување
    early_stopping=False, # рано запирање со валидацијски сет
    verbose=1) # побогат испис при тренирањето
mlp.fit(x_train, y_train) # тренирање на моделот
```

```
Iteration 1, loss = 4.23441581
Iteration 2, loss = 1.70667296
Iteration 3, loss = 1.47485129
Iteration 4, loss = 1.46390913
Iteration 5, loss = 1.37562193
Iteration 6, loss = 1.30116524
Iteration 7, loss = 1.25296934
Iteration 8, loss = 1.18017693
```

<sup>47</sup>sciKit-Learn <http://scikit-learn.org/stable/>

```
Iteration 9, loss = 1.08401595
...
Iteration 141, loss = 0.29614614
Training loss did not improve more than tol for 20 consecutive epochs. Stopping.
```

Може да видиме дека тренирањето на моделот завршува за 141 итерација при постигната грешка од 0,2961. Во Сајкитлерн во невронските мрежи за класификација на повеќе класи се употребува софтмакс како излезен слој, а меѓу-ентропијата како функција на грешка.

## Евалуација на моделот

За да направиме предвидување врз база на множеството за тестирање може да се послужиме со `mlp.predict_proba` функцијата која ќе ни го даде излезот на невронската мрежа, односно `mlp.predict` функцијата која ќе ни ја даде предвидената излезна класа како бинарен вектор користејќи праг од 0,5.

За евалуација на мрежата, дополнително е имплементирана функцијата `mlp.score` која ќе ја искористиме за да видиме која е точноста на предвидувањата на нашата мрежа.

```
accuracy = mlp.score(x_test, y_test)
print(accuracy)
```

```
0.8168664737494833
```

Може да видиме дека нашиот модел постигнува 81,69% точност што е далеку над точноста на случаен избор од 16,67%! Дополнителни подобрувања може да оствариме со земање в'предвид на сигналите од акселерометарот за другите оски, како и оптимизација на хиперпараметрите на невронската мрежа.

За дополнително да уочиме каде најмногу греши невронската мрежа може да ја пресметаме **матрицата на грешка**.<sup>48</sup> За да ја пресметаме, ќе направиме предвидување на класите со тест множеството.

```
y_pred_prob = mlp.predict_proba(x_test)
cm = metrics.confusion_matrix(
    y_test.argmax(axis=1),
    y_pred_prob.argmax(axis=1))
print(cm)
```

```
[[741  0  30   4   0   1]
 [ 3 792  84  41   1   0]
 [ 55  12 142  59   0   5]
 [ 19  11  41 138   0   0]
 [  0  0  0  0  99  21]
 [  0  0  3  0   6 111]]
```

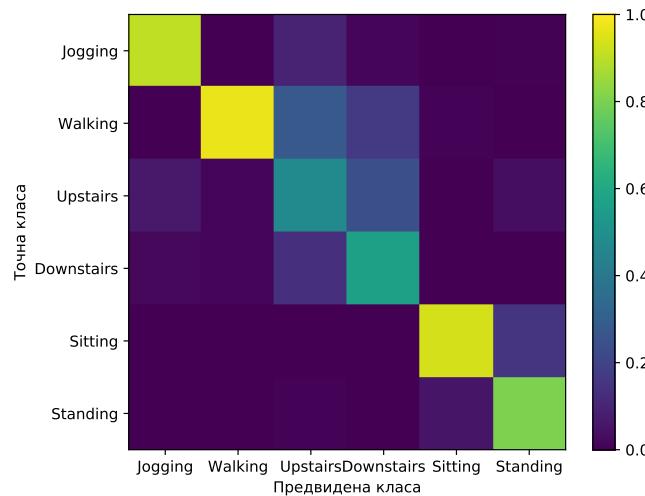
Редиците претставуваат точните класи, а колоните класите предвидени од невронската мрежа. Истата можеме да ја нормализираме за да ја добиеме процентуалната грешка која ја прави мрежата по предвидена класа. Исто така, за појасен приказ ќе ја прикажеме матрицата на грешка преку **топлинска мапа**<sup>49</sup> на Сл. 4.6

```
cm = cm / cm.sum(axis=0)
print(cm)

fig, ax = plt.subplots()
im = ax.imshow(cm, aspect='auto', interpolation='nearest', vmax=1, vmin=0)
```

<sup>48</sup>Анг. *confusion matrix*.

<sup>49</sup>Анг. *heat map*.



**Сл. 4.6:** Матрица на грешка за истренираната невронска мрежа.

```
ax.set(xticks=np.arange(cm.shape[1]),
       yticks=np.arange(cm.shape[0]),
       xticklabels=activities,
       yticklabels=activities,
       ylabel='Точна класа',
       xlabel='Предвидена класа')
fig.colorbar(im)
```

```
[[0.90586797 0.         0.1        0.01652893 0.         0.00724638]
 [0.00366748 0.97177914 0.28     0.16942149 0.00943396 0.        ]
 [0.06723716 0.01472393 0.47333333 0.24380165 0.         0.03623188]
 [0.02322738 0.01349693 0.13666667 0.57024793 0.         0.        ]
 [0.         0.         0.         0.         0.93396226 0.15217391]
 [0.         0.         0.01      0.         0.05660377 0.80434783]]
```

Може да се види дека највеќе грешки невронската мрежа прави при одлучување помеѓу класите одење по скали нагоре и надолу. Причините за ова може да ги видиме во сличноста на нивните временски сигнали на Сл. 4.4 и нивните спектри на Сл. 4.5. Поради, сличноста на овие сигнали, како и поради сличниот вид на активност, не е невообичаено при детекција на човечка активност овие две класи да се состават во една класа „одење по скали“.

✓ **Задача за дома.**

1. Дополни го кодот за екстракција на карактеристики и од останатите оски и види колку ќе се подобрят перформансите на невронската мрежа.
2. Искористи ја функцијата за меѓувалидација по групи `GroupShuffleSplit` имплементирана во Сајкитлерн и оптимизирај го изборот на хиперпараметри на невронската мрежа. Која е најдобрата точност што може да се постигне?
3. Спореди ги резултатите со тренирање и оптимизација на невронска мрежа со карактеристиките понудени заедно со овој датасет.

## Додаток А

# Слободен и отворен софтвер за инженерска и научна работа

Еден од најпрочуените софтверски пакети за нумериичка обработка е програмскиот пакет **Матлаб<sup>1</sup>**. Матлаб, преку својата синтакса на високо ниво дозволува: лесна манипулација на матрици, исцртување на функции и податоци, имплементација на алгоритми, создавање на кориснички интерфејси, итн. Тој може да се употреби во најразлични области од инженерската практика, меѓу кои и во дигиталната обработка на звук, слика и видео. За првпат бил издаден во 1984 г., а во 2004 г. имал 1 милион корисници инженери, научници и економисти.

Сепак Матлаб, како комерцијален софтвер носи и низа од недостатоци, пред сè високата цена која го става вон дофат на студентите, истражувачите, малите компании, како и на научно истражувачките и образовните установи во поголем дел од светот. Други недостатоци на Матлаб се ограничната преносливост на кодот, како и неговата затвореност.

### A.1 Слободен софтвер

Денес сè повеќе инженери и научници ја напуштаат употребата на комерцијалниот затворен софтвер и својата работа ја засноваат на платформи базирани на **слободен софтвер<sup>2</sup>**. Ова пред сè се должи на философијата на движењето за слободен софтвер започнато од **Ричард Сталман<sup>3</sup>** во 1983 г. со креирањето на ГНУ оперативниот систем, а подоцна со воспоставување на Фондацијата за слободен софтвер<sup>4</sup> во 1985 г., како и поширокото **движење за отвореност<sup>5</sup>**, а тоа е заедништво во создавањето и напредувањето на технологијата и човештвото.

### A.2 Четири слободи

Слободниот софтвер е дефиниран со четирите слободи:<sup>6</sup>

- **Слобода 0.** Слобода да ја користите програмата за било која намена.

---

<sup>1</sup>MATLAB®Matrix Laboratory, The MathWorks, Inc., Natick, Massachusetts, United States. <http://www.mathworks.com/products/matlab/>

<sup>2</sup>Wikipedia – Free software movement [https://en.wikipedia.org/wiki/Free\\_software\\_movement](https://en.wikipedia.org/wiki/Free_software_movement)

<sup>3</sup>Wikipedia – Richard Stallman [https://en.wikipedia.org/wiki/Richard\\_Stallman](https://en.wikipedia.org/wiki/Richard_Stallman)

Предавање на Ричард Сталман за философијата на движењето за слободен софтвер – Richard Stallman – Free software, free society, TEDxGeneva 2014 [https://www.youtube.com/watch?v=Ag1AKl\\_2GM](https://www.youtube.com/watch?v=Ag1AKl_2GM)

<sup>4</sup>Wikipedia – Free Software Foundation [https://en.wikipedia.org/wiki/Free\\_Software\\_Foundation](https://en.wikipedia.org/wiki/Free_Software_Foundation)

<sup>5</sup>Wikipedia – Open-source model [https://en.wikipedia.org/wiki/Open-source\\_model](https://en.wikipedia.org/wiki/Open-source_model)

Nathan Seidle – How Open Hardware will Take Over the World, TEDxBoulder [https://www.youtube.com/watch?v=xGhj\\_llNtd0](https://www.youtube.com/watch?v=xGhj_llNtd0)

<sup>6</sup>Превземено од вебстраницата на организацијата Слободен софтвер Македонија <https://slobodensoftver.org.mk/shto>

Добавањето рестрикции за користење на слободен софтвер, како што се временските рестрикции („Пробен период од 30 дена“, „Лиценцата истекува на 1 јануари 2005“), рестрикции на целта („Дозволена е употреба за истражувачки и некомерцијални цели“) или рестрикции на географската област („Мора да се користи во земјата А“), ја прават програмата неслободна.

- **Слобода 1.** Слобода да проучите како работи програмата и како истата да ја адаптирате на сопствените потреби.

Добавањето легални или практични рестрикции на разбирањето или менувањето на програмата, како што се задолжително купување на специјални лиценци, потпишување на спогодба за неоткривање (Non-Disclosure-Agreement) или правењето изворниот код да биде недостапен, исто така ја прават програмата неслободна. Без слободата да се менува програмата, луѓето ќе останат на милост на единствен снабдувач.

- **Слобода 2.** Слобода да редистрибуирате копии за да му помогнете на вашиот сосед.

Софтверот може да се копира/дистрибуира скоро без никакви трошоци. Ако не смеете да му дадете некоја програма на некој човек кому таа му треба, тоа ја прави програмата неслободна. Се разбира, доколку сакате, за ваквите активности можете да наплатите.

- **Слобода 3** Слобода да ја подобрувате програмата и да ги издадете вашите подобрувања во јавноста, од што корист ќе има целата заедница.

Сите луѓе не се подеднакво добри програмери. Некои луѓе пак воопшто не знаат да програмираат. Оваа слобода им дозволува на оние луѓе кои немаат време или знаење да решат некој проблем индиректно да пристапат до слободата за менување на програмата. Се разбира, доколку сакате, за ваквите активности можете да наплатите.

Доколку софтверот не ги исполнува сите горни услови, тогаш тој не е слободен софтвер.

### A.3 Предности на слободниот софтвер

Од практичен аспект, отворениот софтвер има низа предности над затворениот софтвер и тоа:

- **достапноста** – поради основната премиса на давање на изворниот код, со цел да се овозможи неговиот развој од заедницата, отворениот софтвер е *de facto* и бесплатен софтвер. Така, повеќето производители на слободниот софтвер живеат од донацији, но и од продавање поддршка за нивниот производ.
- **безбедноста** – поради достапноста на изворниот код, не постои начин производителот на софтверот да прави нешто скриено од вас, а секој спорен дел од кодот е подложен на промена од заедницата. Кај затворениот софтвер тоа не е случај.<sup>7,8</sup>
- **слободата од производителот** – како корисници на отворениот софтвер, вие не сте затворени во екосистемот на производителот.<sup>9</sup> Истиот тој софтвер може да биде превземен од друга заедница на програмери и да продолжи неговото одржување и развој во друга насока.

<sup>7</sup> Во Windows 10 производителот го задржува правото да ги чува вашите приватни податоци како што вели во изјавата за приватност: “Finally, we will access, disclose and preserve personal data, including your content (such as the content of your emails, other private communications or files in private folders), when we have a good faith belief that doing so is necessary ...”

Истите механизми се додадени во претходните верзии на Windows преку автоматските надградби.

Zach Epstein, Windows 10 is spying on almost everything you do – here's how to opt out, Jul 31, 2015, <http://bgr.com/2015/07/31/windows-10-upgrade-spying-how-to-opt-out/>

Ashley Allen, How to Stop Windows 7 and 8 From Spying on You <http://www.etechnix.com/stop-windows-7-8-spying/>

<sup>8</sup> Епл и Самсунг ги забавија телефоните на корисниците преку нивното редовно ажурирање <https://www.cnet.com/news/apple-and-samsung-fined-for-slowing-down-phones-with-updates/>

<sup>9</sup> Don Reisinger – Steve Jobs wanted to ‘further lock customers’ into Apple’s ‘ecosystem’ <https://www.cnet.com/news/steve-jobs-wanted-to-further-lock-customers-into-apples-ecosystem/>

- подобар **квалитет** – при воспоставување на критична големина на заедницата околу еден отворен софтвер, развојот не може да се спореди со ресурсите кои ги поседува било која корпорација во светот. Така, развојот на **Линукс јадрото**<sup>10</sup>, кое е во основата оперативниот систем **ГНУ/Линукс** познат и само како **Линукс**<sup>11</sup> и повеќе од 600-те **ГНУ/Линукс дистрибуции**<sup>12</sup>, првично напишано од **Линус Торвалдс**<sup>13</sup>, денес претставува најголемиот здружен проект во историјата на човештвото со околу 6000 активни развивачи, над 20 милиони редови на код, и со проценета развојна вредност од над 2 милијарди евра.<sup>14</sup>

Сите овие придобивки заедно придонесуваат за широка распространетост на слободниот софтвер денес.<sup>15</sup> Така, ГНУ/Линукс и ФриБСД<sup>16</sup> оперативните системи се во употреба во 98,27% од серверите на интернет (споредено со 1,73% со Виндоус), 71,9% од паметните телефони (Андроид оперативниот систем), и 99% од суперкомпјутерите<sup>17</sup>. Сепак, неговиот пробив во персоналните компјутери засега е мал – 2,5% (наспроти 75,5% на Виндоус и 14,9% на МекОС).

## A.4 Одржливост

Постојат различни начини на кои се реализира финансиската поддршка на слободниот софтвер и покрај бесплатноста и тоа:

- финансиска поддршка од компанији – зад многу пакети слободен софтвер стојат компанији од чиј интерес е неговиот развојот, Најдобар пример за тоа е можеби самото Линукс јадро на кое работат инженери од многу компании од целиот свет, а најголемиот придонес го има компанијата Интел. Тука се и низа на ГНУ/Линукс дистрибуции меѓу кои Убунту<sup>18</sup>, Федора<sup>19</sup>, и ОпенСусе<sup>20</sup>, како и пакетите за длабоко учење Тензорфлоу<sup>21</sup> и Пајторч<sup>22</sup>, исто така развивани од компанији,
- финансиска поддршка од јавно финансирање и грантови – голем број на слободни софтвери се плод на работата на инженери и научници финансиирани од државите низ светот или од приватни фондации. Таков е на пример КиКАД софтверот за електронски дизајн и изработка на печатени плочи развибан во ЦЕРН<sup>23</sup>, софтверот за обработка на аудио Аудасити започнат во Универзитетот Карнеги Мелон<sup>24</sup>, или пак пакетот за машинско учење Сајкитлрн започнат во Ирија<sup>25</sup>,
- бизнис модел базиран на поддршка – најголемата компанија која денес работи исклучиво со слободен софтвер е Ред Хет чиј ГНУ/Линукс оперативен систем е еден од најзастапените на интернет серверите.<sup>26</sup> Ред Хет заработува преку продажба на поддршка за овој оперативен систем и во моментов е проценета на вредност од 38 милијарди УСД,
- финансиска поддршка од донацији – многу слободни софтвери егзистираат благодарејќи на донацији направени од нивните корисници. Тука спаѓаат најголем број од ГНУ/Линукс

<sup>10</sup>Wikipedia – Linux kernel [https://en.wikipedia.org/wiki/Linux\\_kernel](https://en.wikipedia.org/wiki/Linux_kernel)

<sup>11</sup>Wikipedia – Linux <https://en.wikipedia.org/wiki/Linux>

<sup>12</sup>Wikipedia – List of Linux distributions [https://en.wikipedia.org/wiki/List\\_of\\_Linux\\_distributions](https://en.wikipedia.org/wiki/List_of_Linux_distributions)

<sup>13</sup>Wikipedia – Linus Torvalds [https://en.wikipedia.org/wiki/Linus\\_Torvalds](https://en.wikipedia.org/wiki/Linus_Torvalds)

<sup>14</sup>Добар документарен филм за раѓањето и развојот на ГНУ/Линукс оперативниот систем е Revolution OS - 2001 <https://www.youtube.com/watch?v=Eluzi700-P4>

<sup>15</sup>Wikipedia: Linux - Market share and uptake [https://en.wikipedia.org/wiki/Linux#Market\\_share\\_and\\_uptake](https://en.wikipedia.org/wiki/Linux#Market_share_and_uptake)

<sup>16</sup>FreeBSD <https://www.freebsd.org/>

<sup>17</sup>Linux is Running on Almost All of the Top 500 Supercomputers <https://itsfoss.com/linux-supercomputers-2017/>

<sup>18</sup>Ubuntu <https://www.ubuntu.com/>

<sup>19</sup>Fedora <https://getfedora.org/>

<sup>20</sup>OpenSUSE <https://www.opensuse.org/>

<sup>21</sup>TensorFlow – An end-to-end open source machine learning platform <https://www.tensorflow.org/>

<sup>22</sup>PyTorch – from research to production <https://pytorch.org/>

<sup>23</sup>KiCad EDA – A Cross Platform and Open Source Electronics Design Automation Suite <http://kicad-pcb.org/>

<sup>24</sup>Audacity – Free, open source, cross-platform audio software <https://www.audacityteam.org/>

<sup>25</sup>scikit-learn – Machine Learning in Python <https://scikit-learn.org/stable/index.html>

<sup>26</sup>Red Hat – The world's leading provider of open source solutions <https://www.redhat.com>

дистрибуциите како на пример Манџаро<sup>27</sup> или Минт<sup>28</sup>, а исто така Спајдер<sup>29</sup> развојната средина за Питон која ќе ја користиме во предметов,

- ентузијазам – мотивот нешто да се создаде или подобри и да се сподели со целиот свет понекогаш е доволен мотив за развој на слободниот софтвер. Постојат низа пакети со заедници на развивачи кои немаат финансиски придобивки од нивната работа на проектот, но сепак продолжуваат да работат на него водени од сопствените убедувања и стремеж кон повисоки вредности.

## A.5 Слободен софтвер за инженерска и научна работа

Постојат низа на слободни софтвери кои можат да бидат искористени за обработка на нумерички податоци.

- ГНУ Октејв<sup>30</sup> има синтакса направена да биде во голема мера компатибилна со онаа на Матлаб. Во Октејв се реализирани голем број на пакети кои можат да се искористат за обработка на најразлични типови на сигнали. Проблемот со Октејв е во неговата мала брзина на извршување, поради што највеќе се употребува во образоването како замена за Матлаб.
- Сајлаб<sup>31</sup> е слободен софтвер за нумеричка обработка наменет за инженери и научници, во употреба од 1994 г. Сајлаб во себе вклучува и слободна замена за Симуллинк пакетот на Матлаб, наречена Икскос<sup>32</sup>.
- Питон<sup>33</sup> е широко распространет, повеќенаменски, интерпретиран и динамичен програмски јазик на високо ниво направен од Гуидо ван Росум<sup>34</sup> во 1989 г. Иако не е наменет строго за нумеричка анализа, елегантната и едноставна синтакса која овозможува лесна читливост, како и неговата широка распространетост во најразлични области, го прават Питон идеална основа за слободната работа и соработка на научната и образовната заедница широк светот.
- Џулиа<sup>35</sup> е јазик за нумеричко процесирање со компајлирање направен на МИТ, кој иако има синтакса на високо ниво како онаа на Матлаб, работи речиси еднакво брзо со код напишан во С. И покрај големиот потенцијал на Џулија, за сега неговата примена останува ограничена во области во кои е неопходна голема процесирачка моќ.

<sup>27</sup>Manjaro – Professional Linux at its best <https://manjaro.org/>

<sup>28</sup>Linux Mint – From freedom came elegance <https://linuxmint.com/>

<sup>29</sup>Spyder – The Scientific Python Development Environment <https://manjaro.org/>

<sup>30</sup>GNU Octave – Scientific Programming Language <https://www.gnu.org/software/octave/>

<sup>31</sup>Scilab – Open source software for numerical computation <https://www.scilab.org/>

<sup>32</sup>Xcos <https://www.scilab.org/software/xcos>

<sup>33</sup>Python <https://www.python.org/>

<sup>34</sup>Wikipedia – Guido van Rossum [https://en.wikipedia.org/wiki/Guido\\_van\\_Rossum](https://en.wikipedia.org/wiki/Guido_van_Rossum)

<sup>35</sup>The Julia Programming Language <https://julialang.org/>

## Додаток Б

# Основи на користење на ГНУ/Линукс

### Б.1 Инсталирање на ГНУ/Линукс

Иако користењето на Пајтон не е врзано со ГНУ/Линукс оперативниот систем, вежбите во овој предмет ќе се базираат на работа во ГНУ/Линукс.

#### Зошто да инсталираам ГНУ/Линукс?

Линукс има низа предности кои ги споменавме во Додатокот А меѓу кои:

- безбедност – тој е побезбеден од другите оперативни системи поради неговите вградени безбедносни карактеристики и затоа што е помалку подложен на штетен софтвер и вируси,
- стабилност – Линукс е познат по својата стабилност и доверливост, што е една од главните причини што го прави популарен избор за сервери и други критични системи,
- слободен софтвер – нема лиценци кои треба да се платат.

Уште повеќе, користењето на ГНУ/Линукс ќе ве направи подобри идни инженери:

- употреба во компании и институции – Линукс има широка употреба во технолошките компании и во научно-истражувачките установи. Со тоа, искуството што ќе го стекнете со негова употреба ќе ви овозможи едноставно да се вклопите во екосистемот во кој ќе се најдете во вашата инженерска кариера,
- пристап до алатки за развој – многу алатки за развој на софтвер се достапни некогаш и исклучиво за Линукс. Често се случува, дури и ако постои напатство за инсталирање на некои пакети на Виндоус, тоа да биде изразено постара верзија од пакетот, или пак да има многу ограничена поддршка, поради тоа што развиваците користат Линукс,
- контрола и приспособување – кај Линукс корисникот има целосна контрола врз оперативниот систем и хардверот на кој тој е инсталiran преку едитирање на текстуални датотеки. Дополнително, Линукс овозможува висок степен на прилагодување, како на изгледот на десктоп средината, така и на начинот на функционирање на оперативниот систем. Овој степен на контрола ви ги одврзува рацете и ви нуди можност за зголемување на вашите познавања за тоа како работи вашиот хардвер и цел еден оперативен систем, вклучувајќи ги и апликациите,
- командна линија – Линукс често се користи преку командната линија, што ќе ви помогне да ги развиете своите вештини за нејзина употреба. Тоа ќе ви ја олесни работата со системите кои можат да се пристапат преку командна линија, како на пример работа со интернет сервери,

- сеприсутност на различни платформи – освен десктоп компјутерите, речиси сè друго работи на Линукс – од вгнездените платформи како Распбери Пај, преку сите сервери на интернет, па сè до најголемите светски суперкомпјутери. Со тоа искуството што ќе го стекнете со работа во Линукс на вашиот персонален компјутер е директно применливо на цела низа платформи, од кои со дел сигурно ќе се сртнете во вашата професионална кариера. Всушност вашиот Линукс лаптоп е де факто сервер на кој можат да се приклучите од друг компјутер, или пак да стартувате веб апликација до која можат да пристапат корисници преку интернет. Конечно, софтверот кој сте го напишале и сте го тестирале на вашиот лаптоп, ќе работи и на вгнездена платформа (во рамки на можностите) и на интернет сервер, без потреба од големи модификации;
- слободен софтвер – работата со слободен софтвер ќе ви овозможи да станете дел од заедницата која го развива. Со тоа не само што ќе се стекнете со увид во начинот на развој на различни софтвери, туку ќе можете и самите да придонесете во нивниот развој, унапредувајќи ги вашите вештини да работите развој во рамките на тим.

Ова е причината зошто и покрај малата застапеност на ГНУ/Линукс на персоналните компјутери од 2,5% кај општата популација, кај инженерите и програмерите оваа бројка изнесува 40% како за професионална така и за лична употреба, споредено со 49% за употреба на Виндоус за професионална употреба.<sup>1</sup> Ако земеме в'обзир дека 15% од корисниците на Виндоус го користат емулататорот за Линукс WSL, тогаш излегува дека мнозинството (55%) користат Линукс за развој на софтвер.

## Начин на инсталирање

Доколку веќе немате ГНУ/Линукс, истиот се препорачува да го инсталирате паралелно на постоечкиот оперативен систем во dual boot режим. Во најмала рака може да инсталирате ГНУ/Линукс во виртуелна машина, но ова може да ги ограничи постоечките ресурси за процесирање на сигналите.

## Избор на дистрибуција

Иако постојат повеќе од 600 ГНУ/Линукс дистрибуции<sup>2</sup>, вистинскиот избор за инсталирање се сведува на дузина од најпопуларните дистрибуции, затоа што популарноста е некој вид на гарант за нивната стабилност како и леснотијата на наоѓање на поддршка. Добар преглед на популарноста на различните Линукс дистрибуции, како и повеќе информации за истите може да најдете на вебстраницата *Distrowatch*<sup>3</sup>. Во моментов таму топ 5 се: MX Linux, EndeavourOS, Mint, Manjaro и Pop!\_OS. Вреди да ги споменеме и дистрибуциите: Ubuntu, Fedora, openSUSE, и Arch.

Едни од главните разлики помеѓу дистрибуциите се:

- инсталирањето на пакети - различните дистрибуции имаат различни менаџери на пакети кои служат за инсталирање и ажурирање на софтвер;
- режимот на ажурирање на оперативниот систем – некои дистрибуции имаат периодични помали надградби и поголеми надградби на подолг рок, така Убунту има нова верзија со долготрајна поддршка (LTS) на секои две години; некои дистрибуции во периодичните надградби го ажурираат системот на најновите верзии на сите софтвери (rolling release) со што се заобиколува потребата од поголеми надградби, на пример Арч има нова надградба секој месец и нема потреба од други надградби.

---

<sup>1</sup>Stack Overflow Developer Survey 2022: Operating system <https://survey.stackoverflow.co/2022/#section-most-popular-technologies-operating-system>

<sup>2</sup>Wikipedia – List of Linux distributions [https://en.wikipedia.org/wiki/List\\_of\\_Linux\\_distributions](https://en.wikipedia.org/wiki/List_of_Linux_distributions)

<sup>3</sup><http://distrowatch.com/>

Овие карактеристики се заеднички за цели фамилии на дистрибуции, на пример Кубунту и Убунту користат `apt` за инсталација на `.deb` пакети како и Дебиан – оригиналната дистрибуција на која се базирани. Дополнително, сите тие се базираат на поголеми надградби на подолг рок.

## Избор на десктоп средина

Важно е да се направи разлика помеѓу ГНУ/Линукс дистрибуцијата и изгледот на нејзиниот кориснички интерфејс. Интерфејсот на дистрибуцијата се базира на **десктоп средината** која ја користи. Кај ГНУ/Линукс оперативните системи постојат различни десктоп средини како GNOME и KDE Plasma, Xfce, LXQt, Cinnamon, Mate, i3wm, Pantheon, итн. Дистрибуциите вообичаено вклучуваат и одредено ниво на прилагодба на стандардниот изглед на десктоп средината. На пример, Убунту додава панел со икони за стартирање и контрола на програмите од левата страна во GNOME, додека Манџаро додава докер со икони на долниот раб од десктопот.

GNOME<sup>4</sup> е модерна десктоп средина со елегантен интерфејс која се фокусира на едноставност и функционалност. Со самото тоа, таа не нуди голема конфигурабилност. Од друга страна, KDE Plasma<sup>5</sup> важи за најмоќна и најконфигурабилна десктоп средина за Линукс системите. Постојат и поедноставни средини кои се наменети за помала потрошувачка на ресурси како Xfce, LXQt итн.

## Некои препораки

Во Лабораторијата за дигитално процесирање на сигнали ќе работиме со Убунту<sup>6</sup> кој ја користи десктоп средината GNOME. Убунту е една од најпопуларните дистрибуции и вообичаено е првиот избор при преминување на Линукс. Во поново време, дистрибуции кои се базирани на Убунту ја надминаа по популарност. Тука се Линукс Минт со Cinnamon<sup>7</sup> и Pop!\_OS со GNOME<sup>8</sup>. Двете може да се добар прв избор за преминување на Линукс.

Напуштајќи го овој традиционален пристап, една алтернативна препорака е да инсталirate Арч базирана дистрибуција како Манџаро со KDE Plasma.<sup>9</sup> Арч базираните дистрибуции имаат подобар систем на репозиторија за инсталација на пакети, наспроти системот кој го користи Убунту во којшто често за да инсталирате некој софтвер ќе треба да додавате нови репозиторија кои треба да ги најдете на Интернет. Така, во Арч базираните дистрибуции секој софтвер можете да го инсталирате со пребарување на 5те репозиторија на системот во терминалот, без воопшто да треба да отворате веб прелистувач. Уште повеќе, во Арч базираните системи, оперативниот систем и сите инсталирани софтвери постојано се ажурираат на најновата верзија, елиминирајќи ја потребата од периодични големи надградби на системот.

На крајот, конкретниот избор на дистрибуција можеби и не е толку важен, тоа е само првиот чекор во светот на ГНУ/Линукс. За да ја почувствувате вистинската слобода на избор, пробајте дузина различни дистрибуции и најдете ја таа која вас највеќе ви одговара.<sup>10</sup>

<sup>4</sup><https://www.gnome.org/>

<sup>5</sup><https://kde.org/>

<sup>6</sup><https://ubuntu.com/>

<sup>7</sup><https://linu xmint.com/>

<sup>8</sup><https://system76.com/pop>

<sup>9</sup><https://manjaro.org/download/#Official>

<sup>10</sup>Ова истражување неретко се прави, посебно од страна на нови корисници на Линукс, и се нарекува *distro hopping*.

## Б.2 Основи поставки во ГНУ/Линукс

За работа со ГНУ/Линукс ќе го искористиме стандардниот БАШ терминал.<sup>11</sup> Вреди да се напомене дека постојат понапредни шел средини како на пример Z shell или Zsh.<sup>12</sup> Во поново време, ГНУ/Линукс дистрибуциите доаѓаат со инсталлиран и конфигуриран Zsh шел.

Вообичаена кратенка за отворање на нов терминал е `Ctrl-Alt-t`, или ако дистрибуцијата доаѓа со терминал на спуштање може да го отворите со копчето `F12`.

За почеток треба во вашата корисничка папка, односно домашен директориум, да отворите нова папка со името на предметот. Во ГНУ/Линукс основен фолдер на секој корисник е `/home/user_name/`, а кратенка за него е `~`. Тоа може да го направите преку фајл експлорерот, или преку терминалот:

```
~ $ mkdir bme
```

Следно, од Гитлаб страната на предметот Дигитално процесирање на аудио<sup>13</sup> превземете го фолдерот со звучни сегменти кои ќе ги користите во овој вовед во работа во терминал. Тоа можете да го направите на следниот начин:

```
~ $ cd bme
~/bme $ git clone https://gitlab.com/feeit-freecourseware/biomedical-electronics.git
~/bme $ cp -r biomedical-electronics/code/data .
```

## Б.3 Основни на работа во командна линија

“Graphical user interfaces make the easy stuff easy, command line interfaces make the difficult stuff possible.”

### Совети за пишување

- `↑ / ↓` – врти низ командите од историјата (`zsh` нуди понапредно вртење според буквите кои веќе се напишани)
- `Ctrl + ← / Ctrl + →` – го придвижува курсорот еден збор лево / десно
- `Ctrl + w` – го брише последниот збор
- `Ctrl + u` – брише до почетокот на линијата
- `Ctrl + Shift + c / Ctrl + Shift + v` – copy / paste
- `Ctrl + l` – го чисти екранот

### Промпт

- стандарден промпт: user, host, wd (working directory), `$`
- `whoami`, `hostname`
- `echo $USER`, `echo $HOSTNAME`
- `users`

---

<sup>11</sup>Bourne-Again Shell (BASH) средината е стандардна за сите дистрибуции на ГНУ/Линукс и МекОС. Добро напатство за нејзина употреба претставува: Software Carpentry – The Unix Shell <http://swcarpentry.github.io/shell-novice/>

<sup>12</sup>Oh-My-Zsh! A Work of CLI Magic—Tutorial for Ubuntu <https://medium.com/wearetheledger/oh-my-zsh-made-for-cli-lovers-installation-guide-3131ca5491fb>

<sup>13</sup><https://gitlab.com/feeit-freecourseware/biomedical-electronics>

- прилагодување: `export PS1="[\t \u@\h \w]\$ "`
- повторно вчитување на шелот `source .bashrc`

## Структура на директориуми

- коренот на дрвото на сите директориуми во Линукс е root `/`
- `cat /etc/fstab`, `df -h`
- корисничка папка / дома `~`
- `pwd`
- релативни и апсолутни патеки
- `cd`, `bashmarks`, `autojump`
- `cd -`
- `mkdir`
- `ls` + опции и аргументи
- наоѓање помош: `--help`, `man`, `tldr`
- `tree`, `tree -d`
- големина на папка: `du -sh .`
- празно место: `df -h .`

## Вградени функции во шелот и програми

- `type`
  - `type cd`
  - `type mkdir`
  - `type ls`
- `echo $PATH`
- `ls /usr/bin`

## Датотеки

- детален листинг `ls -l`
- пермисии
- `touch` за креирање на датотеки
- премести `mv`
- избриши `rm`
- `alias rm=trash`

## Сè е датотека!

- `stdin`, `stdout`, `stderr`
- редирекција

- `echo hello word`
- `echo hello word >&1`
- `echo hello word > greeter`
- `ls /usr/bin > programs`
- `cat programs`
- `less programs` – во терминалот `less > more`
- `head programs`
- `tails programs`
- `bat programs`
- pipes
  - `ls /usr/bin | less`
  - `ls /usr/bin | wc`
  - `ls /usr/bin | grep zip` – името на оваа програма доаѓа од ед наредбата g/re/p = global search for RegEx and print
  - `ls /usr/bin | grep zip > zips`
  - `ls /usr/bin | grep zip | wc -l`
  - `ls /usr/bin | grep zip | wc -l > zips_total`
  - `ls --help | less`
  - `ls --help | grep newest`

## Подесување на кернелот

- Сите ГНУ/Линукс подесувања и контроли може да се пристапат преку текст фајлови
- `ls /sys/class/`
- `cd /sys/class/leds/input4::scrolllock`
- `echo 1 > brightness`
- `echo 1 | sudo tee brightness`
- `cd /sys/class/backlight/intel_backlight`
- `cat brightness`
- `cat max_brightness`
- `echo 600 | sudo tee brightness`

## Инсталирање на софтвер

- `sudo su`
- менаџери на пакети:
  - Debian, Ubuntu, Mint, Raspbian: `apt-get`, `apt`, `aptitude`
  - Fedora, Red Hat, CentOS: `dnf`

- Arch, Manjaro: `pacman`, `yay`
- OpenSUSE: `zypper`
- нема потреба да се пребарува на Интернет:
  - Ubuntu:
    - \* `sudo apt search`
    - \* `sudo apt install`
  - Manjaro/Arch:
    - \* `yay -Ss kdenlive`
    - \* `yay -S kdenlive`

## History

- `history`
- `history | less`
- `history | tail`
- `Ctrl + r` – пребарување наназад
- `history | fzf`

## Работа со текст и код

- разгледување на фајлови:
  - `cat bme.py`
  - `less bme.py`
  - `bat bme.py`
- пребројување на линии: `cat *.py | wc -l` или само `wc -l *.py`
- пребарување:
  - `grep fft *.py`
  - `grep fft *.py -C 5`
  - `pdfgrep biomedicinska_elektronika_v*.pdf -C 5`
- едитирање:
  - `nano < micro`
  - `vi < vim < neovim` – веројатно најдобриот едитор/развојна средина на сите времиња,
  - LunarVim<sup>14</sup> е развојна средина базирана на `neovim` со вклучени поставки и додатоци,
  - `emacs` е оперативен систем без добар едитор.

Иако `vi` доаѓа прединсталiran на сите Линукс дистрибуции, `neovim` нуди многу повеќе можности и до денес останува во топ 5 развојни средини<sup>15</sup>. Тоа е поради низа предности кои ги има `neovim`, а кои вклучуваат:

<sup>14</sup><https://www.lunarvim.org/>

<sup>15</sup><https://survey.stackoverflow.co/2022/#section-most-popular-technologies-integrated-development-environment>

- неограничена конфигурабилност – може да се конфигурира со креирање сопствени кратенки и команди,
- минимално користење на ресурси – со време на стартивање од 20 ms, висока респонсивност, и мала потрошувачка на РАМ,
- интеграција во терминал – `neovim` работи директно во терминал што овозможува негово користење при пристапување на машини преку Интернет; ова е и најчестата примена на постариот `vi` едитор,
- скриптирање и плагини – `neovim` има разработена архитектура за развој на додатоци (плагини) која им овозможува на корисниците да ја прошират неговата функционалност со помош на едноставниот јазик за скриптирање `lua`<sup>16</sup>,
- центриран околу тастатурата – им овозможува на корисниците да ги извршуваат сите задачи без потреба да користат глушец, ниту стрелките од тастатурата. Ова го прави одличен за слепо типкање бидејќи прстите воопшто не треба да го напуштат главниот ред од тастатурата. Уште повеќе, бидејќи `neovim` претставува модален едитор, секое копче од тастатурата претставува и команда за навигација низ текстот односно негово едитирање. Овие работи во комбинација со моќта на скриптирањето го прават `neovim` најефикасниот и најбрз текст едитор. Употребата на тастатурата исто така доведува до подобрена ергономија при работата што носи и здравствени придобивки.

Поради низата на предности, `neovim` е бр. 1 најсакан едитор од инженерите и програмерите.<sup>17</sup>

## Работа со аудио

Овој дел е превземен од скриптата за Електроакустика.<sup>18</sup>

- `cd electroacoustics/code/audio`
- `ls`
- `play viluska.wav` – треба да се инсталира `sox` за ова да работи
- `soxi viluska.wav`
- `soxi -D viluska.wav`
- пресметка на вкупна должина на аудио со `awk`
  - `for f in *.wav; do echo $f ${soxi -D $f}; done >> wav_lens`
  - `cat wav_lens | awk '{sum += $2} END {printf sum}'`
  - `soxi -D *.wav | awk '{sum += $1} END {printf sum}'`
  - `soxi -D *.wav | awk '{sum += $1} END {printf sum/60}'`
- со Пајтон преку `pyp`<sup>19</sup> кој може да се инсталира со `pip install pyp`
  - `soxi -D *.wav | pyp -b 'sum = 0' 'sum += float(x)' -a 'sum'`
  - `pyp --explain -b 'sum = 0' 'sum += float(x)' -a 'sum'`
- конверзија на фајлови во mp3
  - `sox viluska.wav viluska.mp3` or `sox viluska.{wav,mp3}`

<sup>16</sup>TJ DeVries – Neovim as a Personal Development Environment [https://www.youtube.com/watch?v=IK\\_-C0GXfjo](https://www.youtube.com/watch?v=IK_-C0GXfjo)

<sup>17</sup><https://survey.stackoverflow.co/2022/#section-most-loved-dreaded-and-wanted-integrated-development-environment>

<sup>18</sup><https://gitlab.com/feeit-freecourseware/electroacoustics>

<sup>19</sup><https://github.com/hauntsaninja/pyp>

- `for w in *.wav; do sox $w $(basename $w).mp3; done`
- симнување на YouTube видео и извлекување на аудиото
  - `yt-dlp https://www.youtube.com/watch?v=Ag1AKIL_2GM -o video.mp4`
  - `ffmpeg -i video.mp4 -vn -acodec copy audio.aac`
  - `for f in *.mp4; do echo $f; ffmpeg -i $f -vn -acodec copy "$(basename $f).aac"; done`
  - или може само `yt-dlp --extract-audio`
  - `tldr yt-dlp`

## Работа со процеси

- `xlogo` vs `xlogo &` – треба да се инсталира `xorg-xlogo`
- Контрола на извршување:
  - `Ctrl + c` - прекини го извршувањето (праќа `INT` сигнал на процесот)
  - `Ctrl + z` - паузира го извршувањето (праќа `TSTP` сигнал на процесот)
  - `kill` - праќа `TERM` сигнал на процесот
  - `kill KILL` - праќа `STP` сигнал на кернелот
  - `killall`
- Набљудување на системот:
  - `top < htop`
  - со графикони: `gtop` and `bashtop`
  - за GPU `nvtop`
  - за запишување на диск `iotop`
- Гасење на системот:
  - `poweroff`
  - `shutdown`
  - `reboot`

## Корисно

- `zsh`<sup>20</sup> - понапреден шел<sup>21</sup>
- `tmux`<sup>22</sup> – мултиплексер за терминали
- фајл експлорери: `ranger`, `vifm`
- покажувачи на големина на папки: `gdu`, `ncdu`
- напреден калкулатор: `qalc`
  - `343 m/s to km/h`
  - `100 GBP to MKD`

<sup>20</sup><https://ohmyz.sh/>

<sup>21</sup><https://medium.com/wearetheledger/oh-my-zsh-made-for-cli-lovers-installation-guide-3131ca5491fb>

<sup>22</sup><https://github.com/tmux/tmux/wiki>

- мејл клиент: `neomutt`<sup>23</sup>,
- календар: `cal`
- временска прогноза: `curl wttr.in/skopje`

## Забава

- `asciiquarium`
- `cmatrix`
- `lolcat - ls | lolcat` or `cmatrix | lolcat`
- `sl`
- `cowsay`, `fortune`, `cowfortune`, `fortune-mod-chuck`
- `cointop`
- `telnet towel.blinkenlights.nl`
- `http://artscene.textfiles.com/vt100/`
- `curl -s http://artscene.textfiles.com/vt100/movglobe.vt | pv -q -L 9600`
- `curl -s http://artscene.textfiles.com/vt100/bambi_godzilla | pv -q -L 9600`

## Корисни линкови

- Terminals, shells and SSH<sup>24</sup>
- MIT Missing Semester<sup>25</sup>
- Primeagen developer workflow<sup>26</sup>

<sup>23</sup><https://github.com/LukeSmithxyz/mutt-wizard>

<sup>24</sup><https://j11g.com/2023/01/14/i-dont-understand-terminals-shells-and-ssh/>

<sup>25</sup><https://youtu.be/Z56Jmr9Z34Q>

<sup>26</sup><https://youtu.be/bdumjiHabHQ>

## Додаток В

# Пајтон за процесирање на биомедицинските сигнали

За процесирањето на дигиталните биомедицински сигнали ќе го користиме програмскиот јазик **Пајтон** и тоа неговата понова верзија **3**, заедно со библиотеките:

- Нумпј – за работа со вектори и матрици,<sup>1</sup>
- Сајпај – за дигитално процесирање на сигнали,<sup>2</sup>
- Матплотлиб – за визуелизација.<sup>3</sup>

Освен овие постојат мноштво библиотеки за Пајтон кои се користат во научните истражувања како на пример **Пандас**<sup>4</sup> за статистички анализи, **Симпај**<sup>5</sup> за симболичка математика, **Сајкит-лern**<sup>6</sup> за машинско учење итн.

Како интерфејс кон Пајтон ќе ја користиме интерактивната конзола **ИПајтон**<sup>7</sup> и научната развојна средина за Пајтон **Спајдер**<sup>8</sup>.

### B.1 Зошто Пајтон?

Пајтон е програмски јазик на високо ниво, кој е еден од најпопуларните јазици во софтверската индустрија.<sup>9</sup> Во анализите Пајтон постојано котира како еден од најдобрите програмски јазици во однос на популарноста и употребата. Овој раст на популарноста на Пајтон се должи на низа на предности:

- Лесен за учење и читање: Пајтон има едноставна и јасна синтакса, што им олеснува на почетниците брзо да го научат јазикот, а на корисниците да можат полесно да одржуваат код напишан од тимот,
- Сестран: Пајтон може да се користи за широк опсег на задачи како што се развој на веб апликации, научни пресметки, анализа на податоци, вештачка интелигенција и многу повеќе.

---

<sup>1</sup>NumPy <http://www.numpy.org/>

<sup>2</sup>SciPy <http://www.scipy.org/>

<sup>3</sup>MatPlotLib <http://matplotlib.org/>

<sup>4</sup>Pandas <http://pandas.pydata.org/>

<sup>5</sup>SymPy <http://www.sympy.org/en/index.html>

<sup>6</sup>SciKit-Learn <http://scikit-learn.org/stable/>

<sup>7</sup>IPython Interactive Computing <http://ipython.org/>

<sup>8</sup>Spyder – The Scientific PYthon Development EnviRonment <https://github.com/spyder-ide/spyder>

<sup>9</sup>Овој дел е напишан со помош на ChatGPT на Open AI.

- Голема заедница и многу библиотеки: Пајтон има голема и активна заедница, што придонесува за огромната и постојано растечка колекција на библиотеки и модули од најразновидни области,
- Компабилност со различни платформи: Пајтон доаѓа инсталiran во дистрибуциите на GNU/Linux, а може да работи на кој било оперативен систем вклучувајќи ги Windows и macOS, што го прави идеален избор за организации кои користат или прават апликации за различни уреди.
- Динамички-тип: Пајтон има динамичко одредување на типот, што значи дека типовите на променливите се одредуваат при извршување, што го олеснува пишувањето, развојот и отстранувањето на грешки во кодот.

Секупно, популарноста на Пајтон најверојатно ќе продолжи да расте во иднина, водена од неговата леснотија на користење, разновидност и зголемената побарувачка за анализа на податоци и апликации за вештачка интелигенција.

### **Предности и недостатоци споредено со С и С++**

- Синтакса: Пајтон има поедноставна и почитлива синтакса во споредба со С и С++, што го олеснува пишувањето и одржувањето на кодот. Од друга страна, С и С++ нудат поголема контрола врз хардверот на ниско ниво и операциите на системско ниво, што ги прави погодни за апликации во кои се критични перформансите.
- Перформанси: С и С++ се компајлирани јазици, што ги прави побрзи од Пајтон, кој е интерпретиран јазик. С и С++ имаат и поголема контрола врз управувањето со меморијата, што може да доведе до нејзино поефикасно користење.
- Развој: Развојот на код во Пајтон е генерално побрз и полесен поради неговата поедноставна синтакса и автоматско управување со меморијата, додека С и С++ бараат повеќе рачно управување со меморијата и може да бидат покомплексни за дебагирање.
- Библиотеки и алатки: Пајтон има голема и активна заедница, што придонесува за постоење на огромна колекција на библиотеки и алатки кои го олеснуваат решавањето на различни проблеми. С и С++ исто така имаат голем број библиотеки и алатки, но достапноста на специфични библиотеки може да варира.
- Употреба: Примената на Пајтон е разновидна и тој вообичаено се користи за развој на веб апликации, научни пресметки, анализа на податоци и вештачка интелигенција, меѓу другото. С и С++ вообичаено се користат за програмирање на системско ниво, развој на игри и други апликации во кои перформансите се критични.

### **Предности и недостатоци споредено со MATLAB**

- Синтакса: Пајтон има поразновидна синтакса во споредба со MATLAB, кој е специјално дизајниран за нумерички пресметки. Пајтон може да се користи за поширок опсег на задачи надвор од нумеричкото пресметување, додека MATLAB е фокусиран конкретно на оваа област.
- Трошоци: MATLAB е комерцијален софтвер со висока цена во споредба со Пајтон, кој е со отворен код и слободно достапен.
- Библиотеки и алатки: И MATLAB и Пајтон имаат голема колекција на библиотеки и алатки, но Пајтон има поголема и поактивна заедница, што придонесе за поширок опсег на библиотеки и алатки, особено во области неповрзани со нумерички пресметки.
- Интероперабилност: MATLAB има богат пакет на функции за нумерички пресметки, што го прави моќна алатка за специфични задачи. Сепак, Пајтон може да работи со поширок

опсег на програмски јазици, што го олеснува интегрирањето во други алатки и сложени системи.

- Употреба: MATLAB е специјално дизајниран за нумеричко пресметување и најчесто се користи во области како што се инженерството, финансите и научните пресметки. Пајтон, од друга страна, е поразновиден јазик и се користи за поширок опсег на задачи, вклучувајќи развој на веб, научни пресметки, анализа на податоци и вештачка интелигенција.

## B.2 Пајтон интерпретер

За работа со Пајтон може да ја искористиме стандардната инсталацијата на Пајтон која доаѓа со секоја ГНУ/Линукс дистрибуција. Пајтон интерпретерот можеме да го повикаме во стандардниот БАШ терминал со:

```
$ python
```

и да ја извршиме архетипната "hello world" програма:

```
Python 3.10.9 (main, Dec 19 2022, 17:35:49) [GCC 12.2.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> print('hello world')
hello world
```

За излегување од Пајтон конзолата треба да ја притиснеме стандардната кратенка `ctrl-d` или да напишеме `exit()`.

## B.3 Виртуелни средини за Пајтон

Стандардната алатка за инсталирање на пакети во Пајтон е `pip`<sup>10</sup>. Инсталирањето на Пајтон пакети со `pip` директно во оперативниот систем не е препорачливо поради можноста за настанување на судир помеѓу пакетите кои системски се инсталирани на ГНУ/Линукс оперативниот систем. За избегнување на овој проблем, како и за изолирање на екосистемот од инсталирани модули на секој засебен проект, правилно е да направиме Пајтон виртуелна средина за нашата работа. Постојат низа пакети за создавање и работа со виртуелните средини во Пајтон.

### venv

Креирањето виртуелни средини со `venv` е поддржано во секоја понова Пајтон верзија. Стандардниот начин за креирање на една виртуелна средина е следниот:

```
~ $ mkdir bme
~ $ cd bme
~/bme $ which pip
/bin/pip
~/bme $ python -m venv venv
~/bme $ ll
drwxr-xr-x - vibe vibe 9 мап 21:58 venv
~/bme $ source venv/bin/activate
```

---

<sup>10</sup>Python Install Package

```
(bme) ~/bme $ which pip
~/bme/venv/bin/pip

(bme) ~/bme $ deactivate

~/bme $ which pip
/bin/pip
```

Гледаме дека сме во виртуелната средина според `(bme)` пред БАШ промптовт. При активација на виртуелната средина патеката до средината се додава на прво место на листата од патеки на кои системот ги бара командите кои ги пишуваме во терминалот. Така, наместо системскиот `pip`, шелот го наоѓа локалниот `pip` во нашата виртуелна средина.

`venv` е наједноставниот, но и најограниченот начин за креирање на виртуелни средини во Пајтон. Дополнителен недостаток на оваа метода е што виртуелната средина ќе биде креирана во рамките на папката на проектот, што може да предизвика проблеми ако папката ја синхронизираме со некој сервис за складирање во облак.

## Pipenv

`Pipenv`<sup>11</sup> е понапреден пакет за менаџирање на виртуелни средини во Пајтон. Тој е и препорачан од Телото за пакување на Пајтон<sup>12</sup>.

`Pipenv` треба да го инсталлираме користејќи го системскиот `pip`<sup>13</sup>:

```
$ sudo pip install pipenv
```

За креирање на виртуелна средина во папката за овој предмет ќе напишеме:

```
~ $ mkdir bme
~ $ cd bme
~/bme $ pipenv install

Creating a virtualenv for this project...
Pipfile: ~/bme/Pipfile
Using /bin/python (3.10.9) to create virtualenv...
[ Creating virtual environment...created virtual environment CPython3.10.9.final.0-64 in
→ 845ms
  creator CPython3Posix(dest=~/.local/share/virtualenvs/bme-P-eb7icK, clear=False,
  → no_vcs_ignore=False, global=False)
  seeder FromAppData(download=False, pip=bundle, setuptools=bundle, wheel=bundle, via=copy,
  → app_data_dir=~/.local/share/virtualenv)
  added seed packages: pip=23.0.1, setuptools=67.4.0, wheel=0.38.4
  activators
  → BashActivator,CShellActivator,FishActivator,NushellActivator,PowerShellActivator,
  PythonActivator

[ Successfully created virtual environment!
Virtualenv location: ~/.local/share/virtualenvs/bme-P-eb7icK
Creating a Pipfile for this project...
Pipfile.lock not found, creating...
Locking [packages] dependencies...
Locking [dev-packages] dependencies...
Updated Pipfile.lock (fedbd2ab7af84cf16f128af0619749267b62277b4cb6989ef16d4bef6e4eef2)!
```

<sup>11</sup>Pipenv – Python Development Workflow for Humans <https://github.com/pypa/pipenv>

<sup>12</sup>Python Packaging Authority PyPA <https://packaging.python.org/en/latest/tutorials/managing-dependencies/>

<sup>13</sup>Python Install Package (pip) <https://pypi.org/project/pip/>

```
Installing dependencies from Pipfile.lock (e4eef2)...
To activate this project's virtualenv, run pipenv shell.
Alternatively, run a command inside the virtualenv with pipenv run.
```

Со ова `pipenv` креира нова виртуелна средина во локална папка во нашата корисничка папка, во неа го копира системскиот Пајтон и `pip`. За активирање на виртуелната средина може да ја искористиме наредбата `pipenv shell` која ја става патеката на виртуелната средина како прва во листата на системски патеки. Така, следниот пат кога ќе сакаме да го повикаме Пајтон интерпретерот или да инсталураме пакет со `pip`, тоа ќе се случува внатре во виртуелната средина:

```
~/bme $ which pip
/usr/bin/pip

~/bme $ echo $PATH
/usr/local/sbin:/usr/local/bin:/usr/bin ...

~/bme $ pipenv shell
Launching subshell in virtual environment...
. ~/.local/share/virtualenvs/bme-aWLbLjVf/bin/activate

(bme) ~/bme $ echo $PATH
~/.local/share/virtualenvs/bme-aWLbLjVf/bin:/usr/local/sbin:/usr/local/bin:...

(bme) ~/bme $ which pip
~/.local/share/virtualenvs/bme-aWLbLjVf/bin/pip
```

За да излеземе од неа повторно може да ја употребиме кратенката `ctrl-d` или да напишеме `exit`.

## Mamba

Постојат и понаредни пакети за креирање и менаџирање со виртуелни средини во Пајтон. Една од најкористените е алатката `conda` која може да се инсталира со Miniconda<sup>14</sup> минималниот пакет од Anaconda<sup>15</sup> дистрибуцијата на Пајтон пакети за работа со нумерички податоци. Иако `conda` е слободен софтвер, зад него стои компанијата Anaconda Inc. која нуди и други платени сервиси.

Во поново време, развиен е слободен софтвер кој се базира на `conda` кој се вика `mamba`<sup>16</sup> кој нуди зголемена брзина и поробусно менаџирање на пакетите во виртуелните средини. Тој може да се инсталира преку Mambaforge<sup>17</sup>.

Да ја инсталураме Mambaforge и да креираме нова виртуелна средина. Забележете дека виртуелните средини креирани со `mamba` не се врзани со папката во која се наоѓаме при нивното креирање, туку тие се инсталираат на ниво на корисник. Ова ја заобиколува потребата за креирање на нова виртуелна средина за секој Пајтон проект кој го работите, односно овозможува да користите една работна средина за произволен број на проекти. Со тоа се олеснува работата со повеќе проекти за кои ни требаат горе-долу истите пакети.

```
$ wget "https://github.com/conda-forge/miniforge/releases/latest/download/Mambaforge-pypy3
-Linux-x86_64.sh"
$ bash Mambaforge-$(uname)-$(uname -m).sh
```

По завршување на инсталацијата треба повторно да ги вчитаме поставките за нашиот шел и можеме да ја креираме нашата виртуелна средина.

<sup>14</sup><https://docs.conda.io/en/latest/miniconda.html>

<sup>15</sup><https://www.anaconda.com/>

<sup>16</sup><https://mamba.readthedocs.io>

<sup>17</sup><https://github.com/conda-forge/miniforge#mambaforge>

```
$ source ~/.bashrc

$ mamba create -n bme python

$ mamba activate bme

(bme) $ which pip
~/mambaforge/envs/bme/bin/pip

(bme) $ mamba deactivate
```

## В.4 Развојни средини за Пајтон

### ИПајтон

Поради ограничните можности на основниот Пајтон интерпретер, вообичаено со Пајтон се работи во интерактивната конзола **ИПајтон** која нуди низа на подобрувања. Нејзe може да ја инсталраме со:

```
$ sudo apt install ipython
```

а по инсталацијата може да ја повикаме со:

```
$ ipython
```

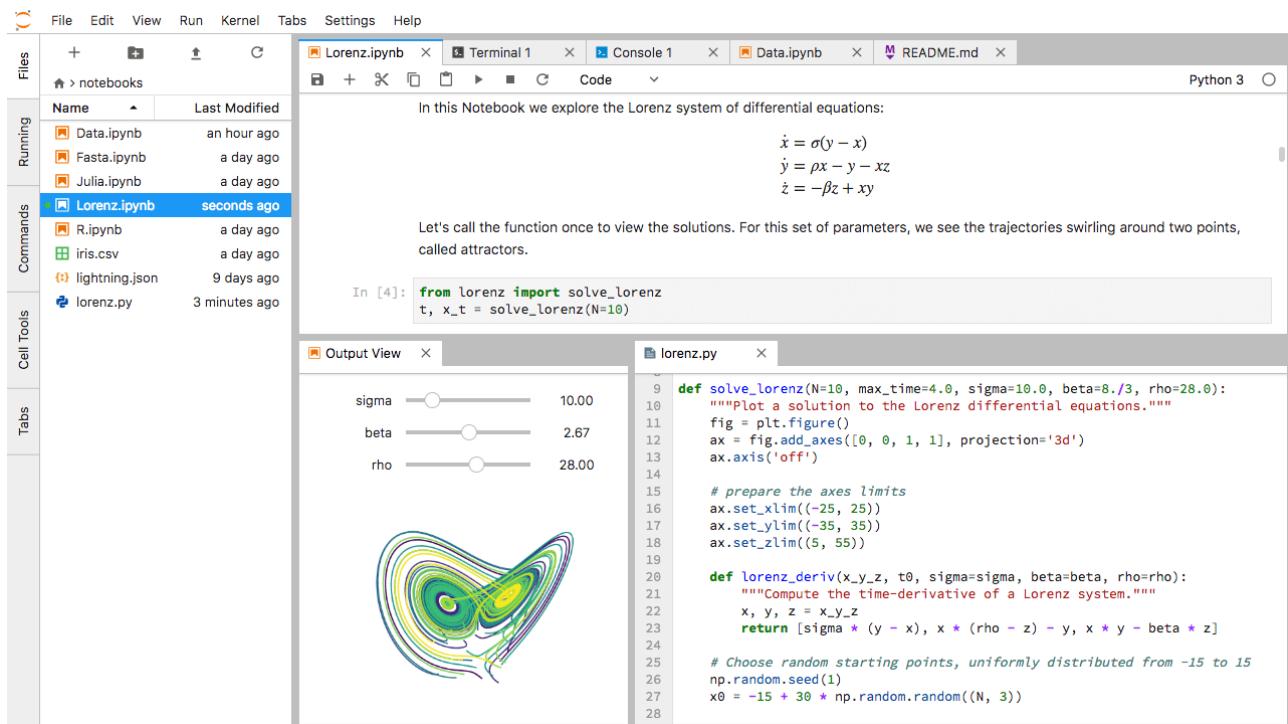
```
Python 3.7.2 (default, Jan 10 2019, 23:51:51)
Type 'copyright', 'credits' or 'license' for more information
IPython 7.3.0 -- An enhanced Interactive Python. Type '?' for help.

In [1]: print('hello world')
hello world
```

Некои од главните придобивки кои ги носи ИПајтон се:

- боење на напишаниот код според синтаксата на Пајтон,
- пристап до стандардната помош во Пајтон, како на пример докстрингови на објекти и напатствието за Пајтон, преку наредбата `help`,
- низа од специјални наредби, наречени и „магии“, како на пример `%timeit` за мерење на времето потребно за извршување на една наредба, `%matplotlib` за овозможување на интерактивно исцртување, или `%history` за испишување, пребарување или запишување на историјата на извршените наредби; повеќе за овие наредби може да се види со наредбата `%magic`,
- информации за секој објект преку употреба на `?` наредбата,
- автоматско комплетирање на имињата на објектите и променливите од локалниот простор на имиња, како и имиња од локалната папка, со употреба на `Tab` копчето,
- пребарување на претходно внесени наредби со стрелките и внесување на првите букви од саканата наредба, а со `ctrl-r` и со пребарување на целата содржина на претходните наредби,
- извршување на шел наредби со помош на `!`.

За да ги видите сите можности кои ги нуди ИПајтон напишете `?` или `%quickref` во интерактивната конзола.



**Сл. В.1:** Јупајтер Лаб развојна средина за работа со ИПајтон тетратки направена како веб апликација.

## Јупајтер Кјутконзола, Јупајтер тетратки и Јупајтер лаб

ИПајтон е основата зад **Јупајтер Кјутконзолата**<sup>18</sup>, која е реализирана во Кјут технологијата и овозможува плотирање во самата конзола кое може да се активира со наредбата `%matplotlib inline`. Таа се стартира во терминалот со наредбата:

```
$ jupyter qtconsole
```

Следниот чекор во развојот на Јупајтер проектот и оној кој му донесе најголем пробив во заедницата се **Јупајтер тетратките** кои овозможуваат **литератно програмирање**<sup>19</sup> – парадигма воведена во 1984 од Доналд Кнут, во која компјутерскиот код е проследен со делови напишани во природен јазик кои можат да содржат слики и математички равенки. Пајтон тетратките се користат често во научните истражувања како медиум за транспарентно споделување на кодот, што е во основата на репродуцибилното истражувањето. Пајтон тетратките се често медиум и за споделување на туторијали или напатствија за користење на одредени пакети.

Во поново време Јупајтер тетратките се вградени во **Јупајтер Лаб**<sup>20</sup> што претставува развојна средина за работа со Пајтон но и со други програмски јазици изработена како веб апликација која може да се користи во рамките на Веб прелистувач. Јупајтер Лаб е прикажан на сл. В.1.

## Спајдер

Во предметот Биомедицинска електроника ќе работиме со развојната средина за Пајтон специјализирана за инженерска и научна работа **Спајдер**<sup>21</sup> прикажана на сл. В.2. Спајдер во себе вклучува:

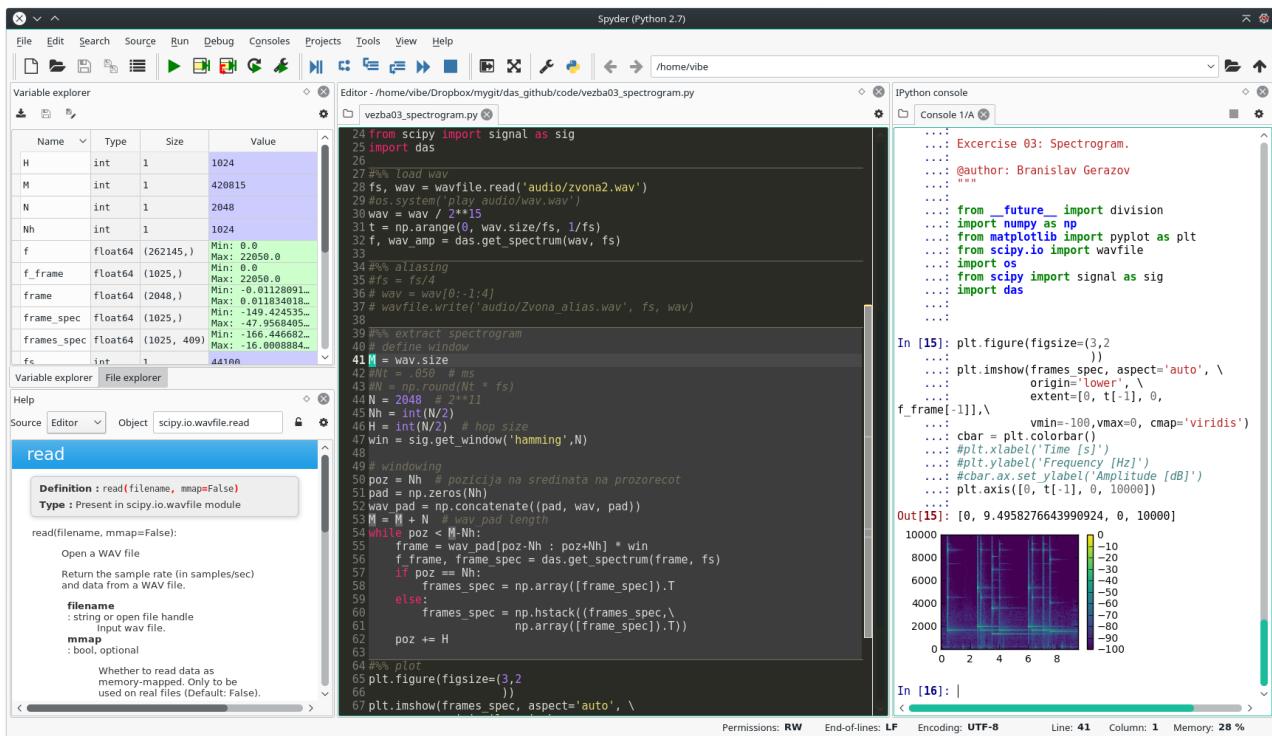
- **Едитор** – со вклучен прелистувач на функции/класи, можности за анализа на код, автоматско завршување на код, и вчитување на дефиниции.

<sup>18</sup>Jupyter QtConsole <https://github.com/jupyter/qtconsole>

<sup>19</sup>Wikipedia – Literate programming [https://en.wikipedia.org/wiki/Literate\\_programming](https://en.wikipedia.org/wiki/Literate_programming)

<sup>20</sup><https://jupyter.org/>

<sup>21</sup>Spyder – The Scientific PYthon Development EnviRonment. <https://www.spyder-ide.org/>



**Сл. В.2:** Спајдер развојната средина за Пајтон специјализирана за инженерска и научна работа.

- **Интерактивна конзола** – интегрирани Пајтон и ИПајтон конзоли со работни простори и поддршка за дебагирање и поддршка за Матплотлиб, овозможуваат инстантна евалуација на кодот напишан во едиторот.
- **Документација** – покажување на документацијата на било која класа или функција повикана во едиторот или конзолата.
- **Приказ на променливи** – овозможува брза анализа на променливите генериирани со некој код.
- **Приказ на датотеки и папки.**
- **Историја на наредби.**

Спајдер можеме да го инсталлираме во виртуелната средина која ја креиравме во работната папка `bme`:

```
(bme) ~/bme $ pip install spyder
```

## Додаток Г

# Основи на програмирање во Пајтон

Добар вовед во програмскиот јазик Пајтон во рамки на екосистемот за научна работа е даден во *Скриптата за Сајпај* (Varoquaux et al., 2015)<sup>1</sup> која е резултат на колективен напор на околу 100 соработници и е достапна со слободна лиценца<sup>2</sup>. Оваа книга претставува отворен проект и во неа, благодарејќи на многуте автори и придонесувачи, се поместени основите за работа не само со Пајтон, Нумпај, Матплотлиб и Сајпај, туку и Пандас, Симпај, Сајкитимиц, Сајкитлрн, па дури и Цајтон<sup>3</sup>.

### Г.1 Основи на Пајтон

Пред да започнеме, мора да нагласиме дека ќе користиме Пајтон 3. Пајтон 2 го достигна својот крај на животот во јануари 2020 година и повеќе не се одржува. Кодот ќе го пишуваме во интерактивната конзола ИПајтон. Алтернативно, може да ја искористиме ИПајтон конзолата вградена во Спајдер.

```
print("Hello world!")
```

```
Hello world!
```

Пред да започнеме, корисно е да ги видиме максимите кои стојат зад дизајнот и филозофијата за пишување на код во Пајтон. Тие се содржани во Зенот на Пајтонот:

```
import this
```

```
The Zen of Python, by Tim Peters
```

```
Beautiful is better than ugly.  
Explicit is better than implicit.  
Simple is better than complex.  
Complex is better than complicated.  
Flat is better than nested.  
Sparse is better than dense.  
Readability counts.  
Special cases aren't special enough to break the rules.  
Although practicality beats purity.  
Errors should never pass silently.  
Unless explicitly silenced.  
In the face of ambiguity, refuse the temptation to guess.  
There should be one-- and preferably only one --obvious way to do it.
```

<sup>1</sup>Scipy Lecture Notes <http://scipy-lectures.org/>

<sup>2</sup>Creative Commons <https://creativecommons.org/>

<sup>3</sup>Cython <https://cython.org/>

```
Although that way may not be obvious at first unless you're Dutch.  

Now is better than never.  

Although never is often better than *right* now.  

If the implementation is hard to explain, it's a bad idea.  

If the implementation is easy to explain, it may be a good idea.  

Namespaces are one honking great idea -- let's do more of those!
```

## Типови на променливи

Пајтон ги поддржува следниве типови на променливи:

- `int` - цел број
- `float` - децимален број
- `complex` - комплексен број
- `bool` - логичка вредност ( `True` или `False` )

Пајтон подржува динамички типови, со други зборови, интерпретерот го одредува типот на променливата при нејзиното декларирање при извршувањето. Типот на променлива може да се менува динамички. Уште една предност е што Пајтон не бара ; на крајот од секоја линија код!

```
a = 3
type(a)
Out: int

c = 0.5
type(c)
Out: float

test = (3 > 4)
test
Out: False

type(test)
Out: bool
```

## Основни математички операции

Пајтон ги поддржува основните математички операции `+`, `-`, `*`, `/`, `**` (за степенување) и `%` (модуло). Па може веднаш да се користи како калкулатор:

```
7 * 3
Out: 21

7/3
Out: 2.3333333333333335

2**10
Out: 1024

8 % 3
Out: 2
```

## Типови контејнери

Пајтон нуди многу ефикасни типови на контејнери, во кои може да се складираат колекции на објекти.

- `str` - низа на знаци или стринг,
- `list` - подредена колекција на објекти кои можат да имаат различни типови,
- `dict` - речници составени од парови „клуч: вредност“ кои поддржуваат брзо пребарување,
- `tuple` - n-торки што не може да се менуваат,
- `set` - неподредени низи на уникатни ставки.

## Стрингови

```
b = 'Hello'
type(b)
Out: str
```

Стринговите поддржуваат некои математички операции. Исто така ' и " се двете OK.

```
b + " World!"
Out: 'Hello World!'

"=" * 80
Out: '====='
```

Сè во Пајтон е објект. Стринговите се исто така објекти и вклучуваат методи за манипулација со нив. Можете да ги видите сите методи ако го напишете името на променлива која е стринг, па . и притиснете на копчето `<Tab>`.

```
b.upper()
Out: 'HELLO'

b.lower()
Out: 'hello'

b.startswith("he")
Out: False

b.endswith(".wav")
Out: False

b.replace("e", "a")
Out: 'Hallo'
```

## Листи

```
colors = ['red', 'blue', 'green', 'black', 'white']
type(colors)
Out: list
```

Индексирањето во Пајтон започнува од `0` (како во C, но различно од MATLAB), а со `-1` е означена последната ставка.

```
colors[2]
Out: 'green'

colors[-1]
Out: 'white'
```

Индексирањето на подлисти од една листа се врши преку `[start:stop]` што ги дава елементите со индекси `i` за кои `start ≤ i < stop` (`i` се движи од `start` до `stop-1`). Можеме да додадеме

и чекор како трет параметар `[start:stop:stride]`. Сите параметри во индексирањето подлисти се опционални.

```
colors[2:4]
Out: ['green', 'black']

colors[3:]
Out: ['black', 'white']

colors[:3]
Out: ['red', 'blue', 'green']

colors[::-2]
Out: ['red', 'green', 'white']

colors[::-1]
Out: ['white', 'black', 'green', 'blue', 'red']
```

Листите се променливи објекти и може да се менуваат:

```
colors[0] = 'yellow'
colors
Out: ['yellow', 'blue', 'green', 'black', 'white']
```

Надоврзување и сортирање:

```
colors + [True, 3.1415]
Out: ['yellow', 'blue', 'green', 'black', 'white', True, 3.1415]

sorted(colors)
Out: ['black', 'blue', 'green', 'white', 'yellow']
```

Листите поддржуваат многу методи за работа со нив. Повторно, можете да ги видите ако го напишете името на променлива која е листа . и притиснете на копчето Tab .

```
colors.append(10)
colors
Out: ['yellow', 'blue', 'green', 'black', 'white', 10]

colors.pop()
Out: 10

colors
Out: ['yellow', 'blue', 'green', 'black', 'white']
```

## Речници

```
tels = {'Urosh': 5752, 'Arpad': 5578}
tels['Igor'] = 5915
tels
Out: {'Urosh': 5752, 'Arpad': 5578, 'Igor': 5915}

tels['Arpad']
Out: 5578

tels.keys()
Out: dict_keys(['Urosh', 'Arpad', 'Igor'])

tels.values()
Out: dict_values([5752, 5578, 5915])
```

```
tels.items()
Out: dict_items([('Urosh', 5752), ('Arpad', 5578), ('Igor', 5915)])

'Igor' in tels
Out: True
```

## Г.2 Контрола на текот на програмата

### if/elif/else

Блоковите код во Пајтон се одредуваат со индентација. Стандардно секое ниво на индентација соодветствува на 4 празни места, кои едиторот автоматски ги вметнува при притискање на `Tab`. Така, во Пајтон нема потреба од употреба на `{ }`.

```
a = 10

if a == 1:
    print(1)
elif a == 2:
    print(2)
else:
    print('A lot')

Out: A lot
```

`if` се оценува како `False` за:

- секој број еднаков на нула (`0`, `0.0`)
- празен контејнер (`list`, `dict`)
- `False`, `None`

а се оценува како `True` за сите останати случаи.

### for

```
for word in ('cool', 'powerful', 'readable'):
    print(f'Python is {word}')

Out: Python is cool
      Python is powerful
      Python is readable
```

Ако имаме потреба од индексирање:

```
for i, item in enumerate(colors):
    print(i, item)

Out: 0 yellow
      1 blue
      2 green
      3 black
      4 white

for name, tel in tels.items():
    print(name, tel)

Out: Urosh 5752
```

```

Arpad 5578
Igor 5915

vowels = 'aeiouy'
for i in 'powerful':
    if i in vowels:
        print(i)

Out: o
     e
     u

a = [1, 0, 2, 4]
for element in a:
    if element == 0:
        continue
    print(1. / element)

Out: 1.0
     0.5
     0.25

```

### while

```

n = 1
while True:
    print(n)
    n += n
    if n > 100:
        break

Out: 1
     2
     4
     8
     16
     32
     64

```

## Г.3 Креирање на листи во еден ред

Пајтон нуди механизам за креирање листа од друга во еден ред. Ова е моќно, но најдобро е да го користите само во случаи кога не ја намалува читливоста на кодот.

```

squares = [i**2 for i in range(10)]
squares

Out: [0, 1, 4, 9, 16, 25, 36, 49, 64, 81]

```

## Г.4 Функции

Функциите се дефинираат со индентација слично со блоковите код при контролата на текот на програмата.

```

def test():
    print('in test function')
test()

```

```
Out: in test function

def disk_area(radius):
    return 3.1415 * radius**2
disk_area(3)

Out: 28.273500000000002
```

Функциите може да се дефинираат со задолжителни (позиционни) аргументи и опционални аргументи (со клучен збор).

```
def speed_of_sound(temp, temp_units="C"):
    """Calculate speed of sound for a given air temperature.

    Parameters
    -----
    temp : float
        Temperature.
    temp_units : str, optional
        Temperature units. "C" for Celsius or "K" for Kelvin.
        Defaults to Celsius.

    Returns
    -----
    float
        Speed in m/s.
    """
    c0 = 331.22 # m/s at 0 C
    temp0 = 273.15 # 0 C
    if temp_units == "C":
        temp += temp0 # convert to Kelvin
    c = c0 * (temp / temp0)**0.5
    return c

speed_of_sound(40)

Out: 354.6436257832055

speed_of_sound(2, "K")

Out: 28.34202966325652
```

Функциите имаат помошни описи наречени `docstring` поместени веднаш под декларацијата. Постојат неколку конвенции за нивно форматирање, на пр. NumPy стандардот. За да пристапиме до помошта за дадена функција, треба да допишеме `?` по името на функцијата.

```
speed_of_sound?

Signature: speed_of_sound(temp, temp_units='C')
Docstring:
Calculate speed of sound for a given air temperature.

Parameters
-----
temp : float
    Temperature.
temp_units : str, optional
    Temperature units. "C" for Celsius or "K" for Kelvin.
```

```
Defaults to Celsius.

Returns
-----
float
Speed in m/s.
File: /tmp/ipykernel_57980/1873094235.py
Type: function
```

## Г.5 NumPy, SciPy и Matplotlib

NumPy, SciPy и Matplotlib се трите библиотеки кои ја сочинуваат основата за користење на Пајтон, кој е јазик за општа намена, за нумеричка обработка на податоци, а со тоа ја отвораат портата за користење на Пајтон за машинско учење и наука за податоци. Во овој дел накратко ќе покажеме некои од нивните функционалности. Ќе навлеземе подлабоко во нив во останатите поглавја од овој материјал. За да ги инсталлираме потребните модули во новата виртуелна средина ќе напишеме:

```
(bme) ~/bme $ pip install numpy scipy matplotlib pyqt5
```

### NumPy низи

Во сржта на NumPy се NumPy податочните низи – мемориски ефикасни контејнери кои овозможуваат брзи нумерички операции. Ајде да видиме како можеме да ги создадеме и да направиме некои основни манипулации со нив.

```
import numpy as np
a = np.array([0, 1, 2, 3])
a
Out: array([0, 1, 2, 3])

a.dtype
Out: dtype('int64')

a.ndim
Out: 1

a.shape
Out: (4,)

a.size
Out: 4
```

Забележете дека го импортираме Нумпай со `import numpy as np` наместо со `from numpy import *`. Ова е препорачана практика за избегнување на оптеретување на постоечките функции во основниот простор на имиња<sup>4</sup>, како и за зачувување на засебен простор со имиња за секој од импортирани модули. Ова овозможува и одлично автоматско надополнување на започнатото име на модул, функција или променлива во Пајтон.<sup>5</sup>

### Креирање на NumPy низи

```
a = np.arange(10) # 0 .. n-1 (!)
b = np.arange(1, 9, 2) # start, end (exclusive), step
```

<sup>4</sup>Добрата структурираност на просторите на имиња (namespaces) е една од важните одлики на Пајтон како што е и наведено во [Зенот на Пајтон](#)

<sup>5</sup>Автоматското надополнување се активира со притискање на `Tab`.

```
c = np.linspace(0, 1, 6) # start, end, num-points
d = np.ones((3, 3)) # (3, 3) is a tuple
e = np.zeros((2, 2))
a, b, c, d, e

Out: (array([0, 1, 2, 3, 4, 5, 6, 7, 8, 9]),
      array([1, 3, 5, 7]),
      array([0., 0.2, 0.4, 0.6, 0.8, 1.]),
      array([[1., 1., 1.],
             [1., 1., 1.],
             [1., 1., 1.]]),
      array([[0., 0.],
             [0., 0.]]))
```

## Повеќе-димензионални низи

```
a = np.array([[0, 1, 2], [3, 4, 5]])
a
Out: array([[0, 1, 2],
            [3, 4, 5]])

a.ndim
Out: 2

a.shape
Out: (2, 3)

a.size
Out: 6
```

## Математички операции со низи

```
lis = range(10)
[i**2 for i in lis]
Out: [0, 1, 4, 9, 16, 25, 36, 49, 64, 81]

arr = np.arange(10)
arr**2
Out: array([ 0,  1,  4,  9, 16, 25, 36, 49, 64, 81])
```

## Matplotlib графикони

Matplotlib е флексибилна библиотека која овозможува лесно и брзо исцртување на променливите, но и креирање на графикони со висок квалитет за употреба во технички извештаи или научни трудови. За да овозможиме интерактивно цртање, треба да ја повикаме магичната функција `%matplotlib` во IPајтон.

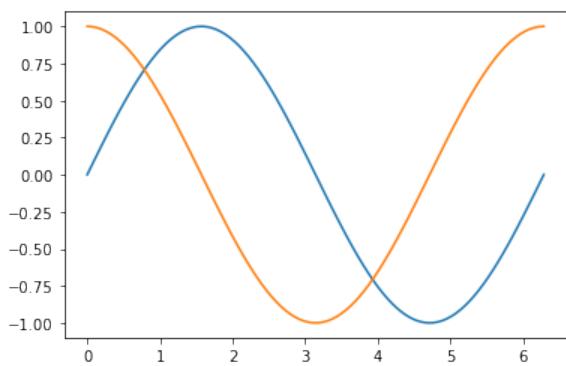
Да ја увеземе библиотеката за исцртување.

```
from matplotlib import pyplot as plt
```

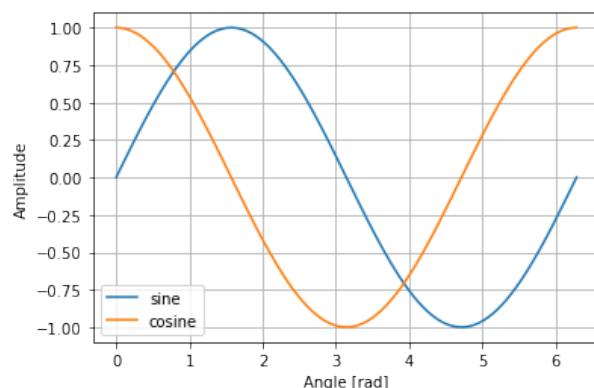
Да направиме низи со синус и косинус.

```
x = np.linspace(0, 2*np.pi, 100)
sin = np.sin(x)
cos = np.cos(x)
```

Конечно, можеме да ги исцртаме низите.



Сл. Г.1



Сл. Г.2

```
plt.plot(x, sin)
plt.plot(x, cos)

Out: [<matplotlib.lines.Line2D at 0x7fd259e8f310>]
```

Графиконот може да го видиме на сл. Г.1.

Можеме да го подобриме графиконот со додавање на информации за оските и мрежа.

```
plt.plot(x, sin)
plt.plot(x, cos)
plt.grid()
plt.xlabel("Angle [rad]")
plt.ylabel("Amplitude")
plt.legend(["sine", "cosine"])

Out: <matplotlib.legend.Legend at 0x7fd2b8094a00>
```

Резултатот е прикажан на сл. Г.2.

## Додаток Д

# Правење кориснички интерфејс со Пајтон

Графичкиот кориснички интерфејс (ГУИ)<sup>1</sup> за вашиот Пајтон код ви овозможува полесно да експериментирате со параметрите на различните алгоритми за процесирање на биомедицинските сигнали. Исто така, ГУИ-то овозможува употреба или претставување на вашите алгоритми интуитивно без навлегување во теоријата, имплементацијата, односно кодот.

### Д.1 Платформи за креирање на ГУИ во Пајтон

Постојат низа на можности за креирање на ГУИ за вашиот Пајтон код. Повеќето од нив овозможуваат креирање на ГУИ-а кои работат на трите главни оперативни системи ГНУ/Линукс, Виндуос и Мекинтош.

#### Tk

Основниот алатник за креирање на ГУИ во Пајтон е **Ткинтер** кој претставува Пајтон врапер на Tk ГУИ алатникот напишан во С.<sup>2</sup> Тој е вклучен во секоја инсталација на Пајтон и претставува слободен софтвер со допустлива лиценца, односно може да се употреби за креирање на комерцијални апликации.

#### wxWidgets

Една алтернатива на Ткинтер е **виксПајтон**<sup>3</sup> кој е врапер за повеќеплатформскиот **виксВицетс**<sup>4</sup> ГУИ алатник, напишан во С++, во кој се креирани многу апликации вклучувајќи ги Аудасити и КиКАД кои ги споменавме во [Поглавјето А](#).

#### GTK

Постојат и помодерни и помоќни платформи за креирање на ГУИ во Пајтон. Една од нив ја сочинуваат **PyGTK**<sup>5</sup> и поновиот **PyGObject**<sup>6</sup> врапер за **GTK**<sup>7</sup> ГУИ алатникот. GTK, напишан во С и CSS, е основниот алатник за десктоп средините на многу Линукс дистрибуции. Во него е направен Гнуум, чии девелопери директно го развиваат и GTK (а се употребува во Дебиан, Убунту, Федора), но и Xfce (Xubuntu, Манџаро, Арч), Цинамон (Минт) и Мате (Убунту Мате).

<sup>1</sup>анг. *Graphical User Interface*.

<sup>2</sup>Tkinter Python wiki <https://wiki.python.org/moin/TkInter>

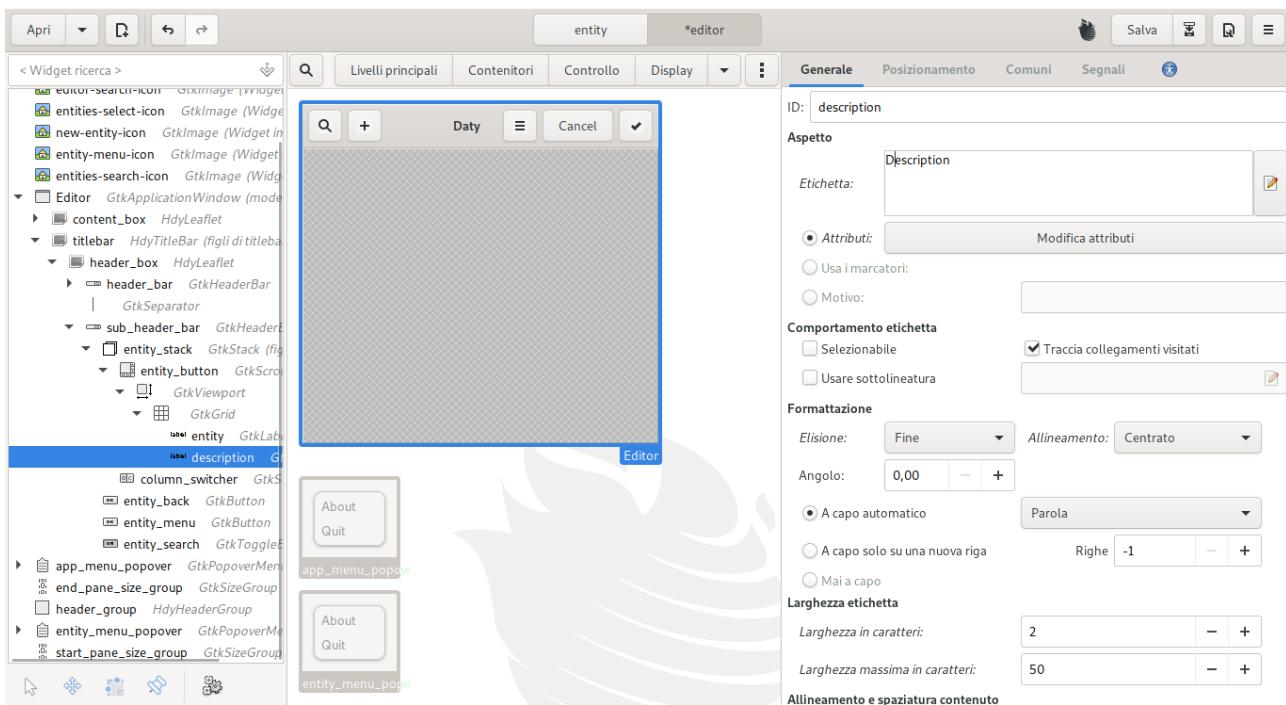
<sup>3</sup>wxPython <https://wxpython.org/>

<sup>4</sup>wxWidgets <https://wxwidgets.org/>

<sup>5</sup>Wikipedia: PyGTK <https://en.wikipedia.org/wiki/PyGTK>

<sup>6</sup>PyGObject <https://pygobject.readthedocs.io>

<sup>7</sup>The GTK Project <https://gtk.org/>



Сл. Д.1: Глејд алатката за дизајн на GTK ГУИ-а.

Иако GTK е главно направен за системите X11 и поновиот Вејланд за Линукс оперативните системи, работи и на други платформи како Виндуос и Мекинтош. GTK е лиценциран под LGPL<sup>8</sup>, што значи дека може да се користи и во комерцијални апликации чиј код не мора да биде со слободна лиценца, под услов тие да не го испорачуваат GTK во својата дистрибуција. PyGTK исто така е достапна под LGPL.

Уште една голема предност која ја нуди GTK е Глејд<sup>9</sup> алатката за брз и лесен развој на ГУИ-а прикажана на Сл. Д.1. Со нејзина помош може едноставно да се состави едно ГУИ, без потреба истото да се генерира преку пишување на код. Вака создаденото ГУИ се зачувува во XML датотека, која може преку GtkBuilder да се употреби во многу програмски јазици вклучувајќи го и Пајтон. Постојат алатки за дизајн на ГУИ-а и за Тк – Пејг<sup>10</sup>, и виксВицетс – виксГлејд<sup>11</sup>, но тие не се толку напредни како Глејд.

## Qt

Втората моќна и модерна платформа за развој на ГУИ-а во Пајтон се базира на Кјут алатникот. Кјут претставува слободен софтвер за креирање на ГУИ-а и повеќеплатформски нативни апликации не само на трите десктоп оперативни системи, туку и на Андроид и на вгнездените микрокомпјутерски системи.<sup>12</sup> Кјут е достапен под комерцијална лиценца, но и допустливата LGPL, како и GPL лиценцата<sup>13</sup>. Кјут стои зад една од најнапредните десктоп средини за ГНУ/Линукс оперативните системи – КДЕ Плазма<sup>14</sup> која ја нудат повеќето популарни дистрибуции вклучувајќи ги: Дебиан, Кубунту, Манџаро, Арч, опенСуз, Федора итн. Исто така, многу апликации ја користат Кјут, меѓу кои и Спајдер.

Кјут компанијата го развива и Пајтон враперот за Кјут наречен ПажСайд.<sup>15</sup> Тој е достапен

<sup>8</sup>Wikipedia: GNU Lesser General Public License [https://en.wikipedia.org/wiki/GNU\\_Lesser\\_General\\_Public\\_License](https://en.wikipedia.org/wiki/GNU_Lesser_General_Public_License)

<sup>9</sup>Glade - A User Interface Designer <https://glade.gnome.org/>

<sup>10</sup>PAGE – Python Automatic GUI Generator <http://page.sourceforge.net/>

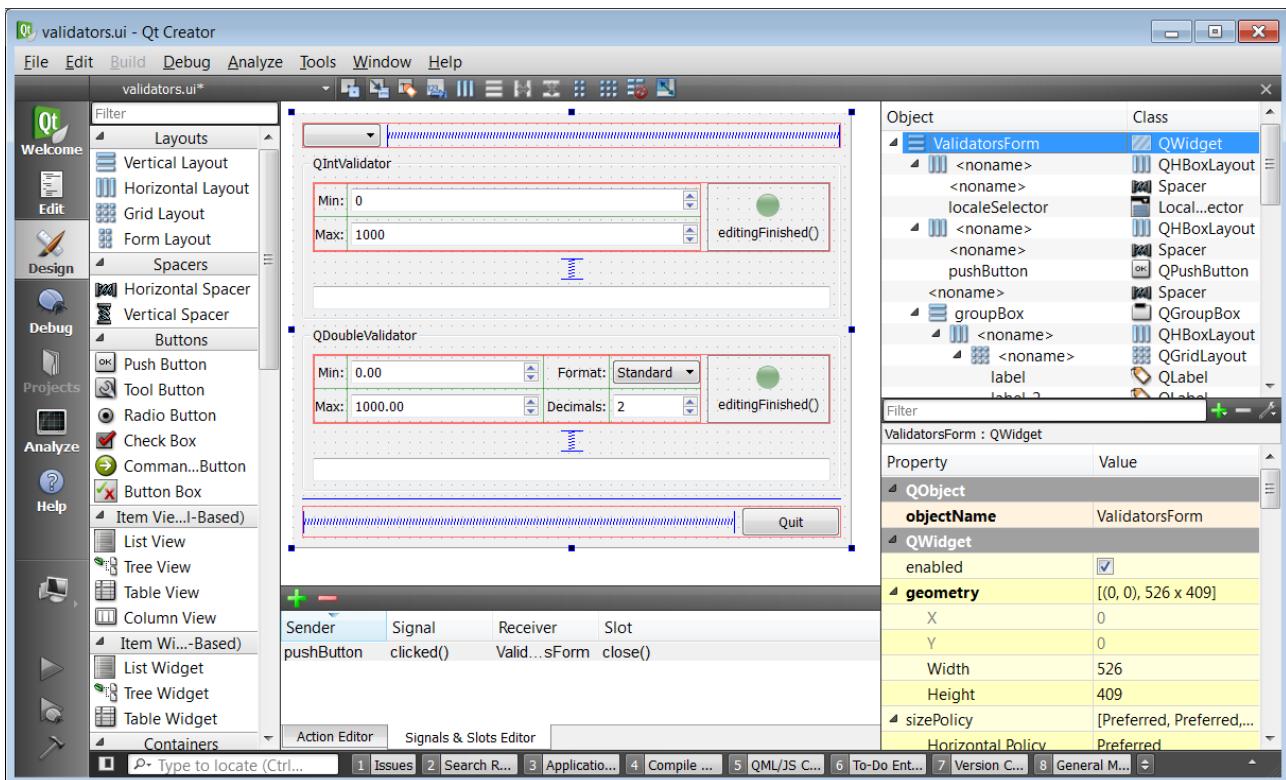
<sup>11</sup>wxGlade – a GUI builder for wxWidgets <http://wxglade.sourceforge.net/>

<sup>12</sup>Wikipedia: Qt [https://en.wikipedia.org/wiki/Qt\\_\(software\)](https://en.wikipedia.org/wiki/Qt_(software))

<sup>13</sup>GNU General Public License [https://en.wikipedia.org/wiki/GNU\\_General\\_Public\\_License](https://en.wikipedia.org/wiki/GNU_General_Public_License)

<sup>14</sup>KDE Plasma <https://kde.org/>

<sup>15</sup>Wikipedia: PySide <https://en.wikipedia.org/wiki/PySide>



Сл. Д.2: КјутКреатор за дизајн на ГУИ во Кјут.

под истите лиценци како и алатникот Кјут, а е развиен како одговор на ПајКјут<sup>16</sup> враперот произведен од друга компанија, кој не вклучува LGPL лиценца. ПајСајд поддржува креирање на апликации за ГНУ/Линукс, Виндоус и Мекинтош, а се работи и на поддршка за Андроид апликации.

За дизајн на ГУИ-а во Кјут постои КјутКреатор<sup>17</sup> прикажан на Сл. Д.2. Тој претставува развојна средина за креирање на Кјут апликации, но може да се искористи и за визуелно составување на ГУИ-а без потреба од пишување код. Слично како Глејд, и КјутКреаторот ги зачувува дизајнираните ГУИ-а во XML файл кој може потоа да се искористи за правење на апликацијата во различни програмски јазици.

## Kivy

Конечно, **Киви**<sup>18</sup> е модерна слободна Пајтон библиотека за развој на мобилни апликации и друг софтвер базиран природни интерфејси, како на пр. преку допир. Таа е достапна под допустливата MIT лиценца и работи на Андроид, иОС, ГНУ/Линукс, Виндоус и Мекинтош. За сега нема визуелна алатка за дизајн на ГУИ-а.

<sup>16</sup>Wikipedia: PyQt <https://en.wikipedia.org/wiki/PyQt>

<sup>17</sup>Wikipedia: QtCreator [https://en.wikipedia.org/wiki/Qt\\_Creator](https://en.wikipedia.org/wiki/Qt_Creator)

<sup>18</sup>Kivy <https://kivy.org/>

# Литература

- S. Serge Barold. Willem einthoven and the birth of clinical electrocardiography a hundred years ago. *Cardiac Electrophysiology Review*, 7(1):99–104, 2003.
- Christopher M Bishop. *Pattern recognition and machine learning*. Springer, 2006. URL <https://www.microsoft.com/en-us/research/uploads/prod/2006/01/Bishop-Pattern-Recognition-and-Machine-Learning-2006.pdf>.
- Stevo Bozinovski, Mihail Sestakov, and Liljana Bozinovska. Using eeg alpha rhythm to control a mobile robot. In *Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pages 1515–1516. IEEE, 1988.
- J.D. Enderle, J.D. Bronzino, and S.M. Blanchard. *Introduction to Biomedical Engineering*. Academic Press series in biomedical engineering. Elsevier Academic Press, 2005. ISBN 9780122386626. URL [https://books.google.mk/books?id=\\_yV3DqIU-tkC](https://books.google.mk/books?id=_yV3DqIU-tkC).
- Ary L Goldberger, Luis AN Amaral, Leon Glass, Jeffrey M Hausdorff, Plamen Ch Ivanov, Roger G Mark, Joseph E Mietus, George B Moody, Chung-Kang Peng, and H Eugene Stanley. Physiobank, physiotoolkit, and physionet: components of a new research resource for complex physiologic signals. *Circulation*, 101(23):e215–e220, 2000.
- Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning*. MIT press, 2016.
- Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Identity mappings in deep residual networks. In *European conference on computer vision*, pages 630–645. Springer, 2016.
- Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.
- Jennifer R Kwapisz, Gary M Weiss, and Samuel A Moore. Activity recognition using cell phone accelerometers. *ACM SigKDD Explorations Newsletter*, 12(2):74–82, 2011.
- Andreas C Müller, Sarah Guido, et al. *Introduction to machine learning with Python: a guide for data scientists*. ” O'Reilly Media, Inc.”, 2016.
- Michael Nielsen. *Neural Networks and Deep Learning*. Determination Press, 2015. URL <http://neuralnetworksanddeeplearning.com>.
- Ana B Porto-Pazos, Noha Veiguela, Pablo Mesejo, Marta Navarrete, Alberto Alvarellos, Oscar Ibáñez, Alejandro Pazos, and Alfonso Araque. Artificial astrocytes improve neural network performance. *PloS one*, 6(4):e19109, 2011.
- Lawrence R. Rabiner and Ronald W. Schafer. *Digital Processing of Speech Signals*. Prentice-Hall signal processing series. Prentice-Hall, 1978. ISBN 9780132136037.
- Gerwin Schalk, Dennis J McFarland, Thilo Hinterberger, Niels Birbaumer, and Jonathan R Wolpaw. Bci2000: a general-purpose brain-computer interface (bci) system. *IEEE Transactions on biomedical engineering*, 51(6):1034–1043, 2004.
- Xavier Serra and Julius O Smith III. Audio signal processing for music applications, 2014. URL <https://www.coursera.org/course/audio>.

Gael Varoquaux, Valentin Haenel, Emmanuelle Gouillart, Zbigniew Jędrzejewski-Szmek, Ralf Gommers, Fabian Pedregosa, Olav Vahtras, Pierre de Buyl, Gert-Ludwig Ingold, Nicolas P. Rougier, and et al. *scipy-lecture-notes*: Release 2015.1 beta, 2015. URL <http://www.scipy-lectures.org/>.

Софija Богданов, Момчило и Богданова. *Дигитално процесирање на сигнали*. Електротехнички факултет, Скопје, 1997. ISBN 9989-630-15-1.

Доне Гершановски. *Предавања по биофизика за општа медицина*.