



# Биомедицинска електроника

Бранислав Геразов

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

Биомедицинска електроника ~ Предавања и вежби v0.75

© Copyright by Branislav Gerazov, 2016 – 2019

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  
<http://www.openhandproject.org/>

# Содржина

<b>1 Вовед</b>	<b>4</b>
1.1 Историја на биомедицинското инженерство . . . . .	4
<b>2 Основи на човековата анатомија и физиологија</b>	<b>8</b>
2.1 Структура на клетка . . . . .	8
2.2 Деполаризација на клетка . . . . .	11
2.3 Нервни импусли . . . . .	13
2.4 Градба и работа на мускулите . . . . .	17
2.5 Градба и работа на човековото срце . . . . .	22
<b>3 Електронски кола за мерење на биопотенцијалите</b>	<b>25</b>
3.1 Електромиографија . . . . .	25
3.2 Електрокардиограм . . . . .	26
<b>4 Обработка на биомедицинските сигнали</b>	<b>31</b>
4.1 Основи на Фурьеовата анализа . . . . .	31
4.2 Фурьеова трансформација на временски отсечоци ФТВО . . . . .	38
4.3 Вовед во филтри . . . . .	40
4.4 Основи на дигиталните филтри . . . . .	40
4.5 Процесирање на ЕМГ сигнали . . . . .	45
4.6 Процесирање на ЕКГ сигнали . . . . .	53
<b>5 Примена на машинското учење на биомедицинските сигнали</b>	<b>60</b>
5.1 Основни парадигми во машинското учење . . . . .	60
5.2 Вообичаени чекори во примената на машинското учење . . . . .	62
5.3 Тренирање на моделите за машинско учење . . . . .	62
5.4 Невронски мрежи . . . . .	64
5.5 Препознавање на човечка активност со акселерометри . . . . .	71
<b>Додаток А Слободен и отворен софтвер за инженерска и научна работа</b>	<b>78</b>
A.1 Слободен софтвер . . . . .	78
A.2 Четири слободи . . . . .	78
A.3 Предности на слободниот софтвер . . . . .	79
A.4 Одржливост . . . . .	80
A.5 Слободен софтвер за инженерска и научна работа . . . . .	81
<b>Додаток Б Питон за процесирање на биомедицинските сигнали</b>	<b>82</b>
B.1 Основи поставки во ГНУ/Линукс . . . . .	82
B.2 Основи на работата со Питон . . . . .	83
B.3 Основи на Нумпай и Матплотлиб . . . . .	86

# Поглавје 1

## Вовед

Биомедицинската електроника е дел од областа биомедицинско инженерство која се занимава со примената на инженерските принципи и концепти на дизајнирање 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 здравството, 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 лубето. 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.

### 1.1 Историја на биомедицинското инженерство

Биомедицинското инженерство постоело со векови, дури и милениуми. Првата вештачка протеза е вештачкиот палец на ногата направен од дрво и кожа најден на мумија од средовечна жена близу античкиот град Теба во Египет која датира од периодот меѓу 1070 – 670 г. п.н.е, Сл. 1.1. 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. 1.2.

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. 1.3.

Во 1848 германскиот лекар и физичар Херман фон Хелмхолц<sup>3</sup> првпат ги употребил инженерските принципи на физиолошки проблем идентификувајќи 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 некој вид на зраци кои ги нарекол „X“ зраци. По ова откритие, научната заедница започнува

<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/René%C3%A9\\_Laennec](https://en.wikipedia.org/wiki/René%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)

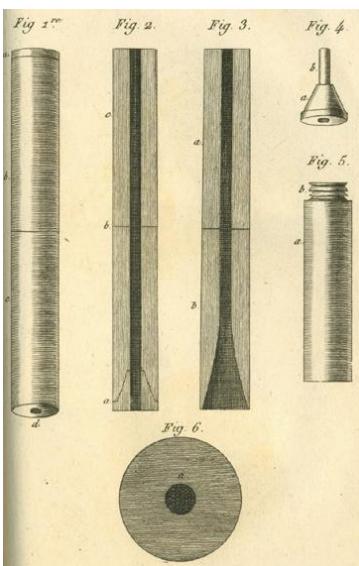
<sup>4</sup>Wikipedia: Wilhelm Röntgen [https://en.wikipedia.org/wiki/Wilhelm\\_Röntgen](https://en.wikipedia.org/wiki/Wilhelm_Röntgen)



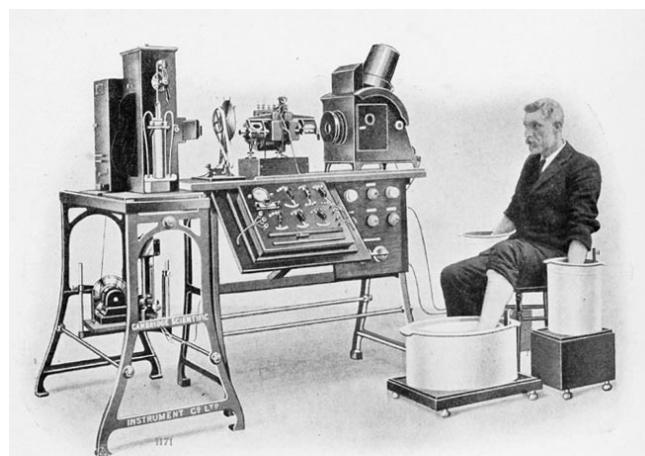
**Сл. 1.1:** Првата протеза – вештачки палец најден на мумија од стариот Египет.<sup>5</sup>



**Сл. 1.2:** Копија од ногата од Капуа од 300 г. п.н.е. пронајдена во Италија.<sup>6</sup>



**Сл. 1.3:** 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. 1.4:** Еден од најраните ЕКГ апарати направен во 1911 г. компанијата за научни инструменти од Кембриџ – *Cambridge Scientific Instrument Company*.<sup>8</sup>

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 одлука 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 ги сними отчукувањата на срцето, првиот практично употреблив електрокардиограм (ЕКГ) го

<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>

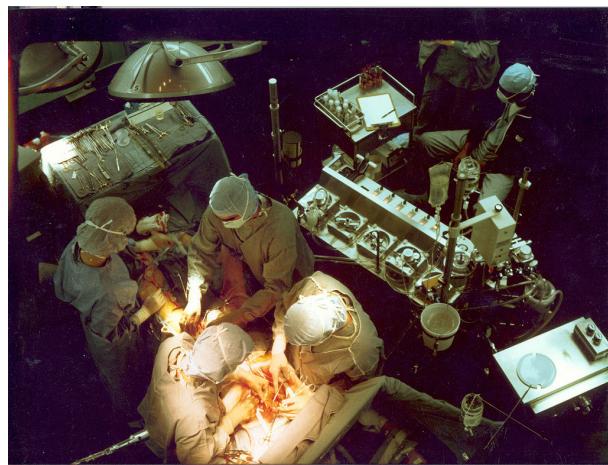
<sup>6</sup>By [http://wellcomeimages.org/indexplus/ofb\\_images/9a/e9/fd5080571b64c69ab0dc7e574b82.jpg](http://wellcomeimages.org/indexplus/ofb_images/9a/e9/fd5080571b64c69ab0dc7e574b82.jpg) Gallery: <http://wellcomeimages.org/indexplus/image/M0012307.html>, CC BY 4.0, <https://commons.wikimedia.org/w/index.php?curid=36352216>

<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. 1.5:** Оддел од респиратори за луѓе со детска парализа во 1953 г.<sup>11</sup>



**Сл. 1.6:** Употреба на машина за кардиопулмонарен бајпас во коронарна хирургија.<sup>12</sup>

измислува во 1903 холандскиот лекар **Вилијам Ајндховен**<sup>9</sup> за што ја добива Нобеловата награда за медицина во 1924 г. Еден таков уред е прикажан на Сл. 1.4. 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 ден 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 **Освалт Институтот за физика и медицина** 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, Охајо, 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 нормалниот и абнормалниот мозок, 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 Ајндховен, а подоцна и галванометарот за снимање на Сименс, кој му овозможил снимање на напони од редот на 100  $\mu$ 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. 1.5 е прикажан еден цел 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 биле постигнати големи напредоци во **нуклеарната медицина**. Во 1953 била направена машината за **кардиопулмонарен бајпас**<sup>14</sup> која прврмено

<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)



Сл. 1.7: Уред за магнетна резонанса.<sup>19</sup>



Сл. 1.8: Вештачко срце SynCardia.<sup>20</sup>

може да ја превземе работата на срцето и белите дробови пред се 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 (CT)**<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 полиња може да ја опфати дистрибуцијата на молекулите на водата во телото, па со тоа да се набљудува неговата анатомија и физиолошките процеси, Сл. 1.7.

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 активна молекула.

Првото **вештачко срце**<sup>18</sup> успешно вградено во човек било срцето Јарвик-7, чиј наследник е прикажан на Сл. 1.8, 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 620 денови. Најчесто тоа се користи како привремено решение пред да е можна трансплантирања на вистинско срце.

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

<sup>20</sup> By SynCardia Systems, Inc. - Own work, CC BY-SA 3.0, <https://commons.wikimedia.org/w/index.php?curid=22097831>

## Поглавје 2

# 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, односно нивните електрични карактеристики, за да може да се дизајнираат системи за нивно набљудување, како и стимулација. Од физиолошки аспект, пак, најзначајни 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).

### 2.1 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 саморепродукција, бидејќи го содржат целокупниот наследен материјал. Типичната градба на една клетка е дадена на Сл. 2.1.<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. По својот облик клетките исто така можат да бидат топчети, плочести, коцкасти, вртеновидни или пак разгранети. На пример: неврните 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, неврна, мускулна, полова клетка итн.

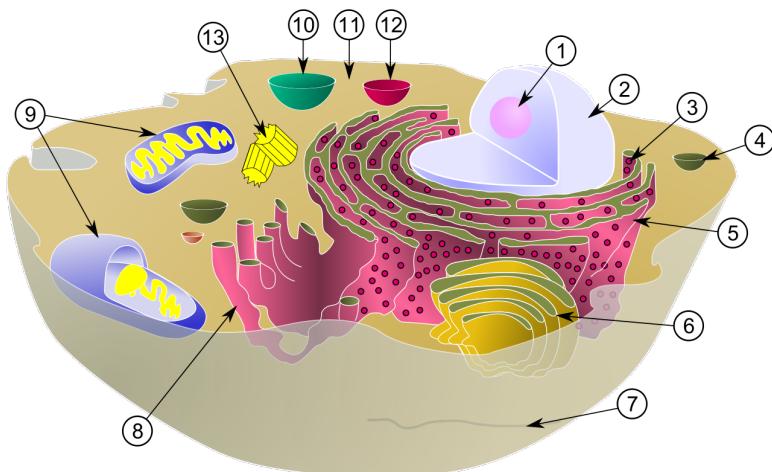
Однадвор клетката е обвиткана со **клеточна мембрана**.<sup>4</sup> Ваквата мембрана е составена од два слоја на масти во кои се вметнати различни белковини и мало количество на шеќери. Клеточната мембрана е многу важна, бидејќи таа одредува што може да влезе, а што да излезе од клетката. Клеточната мембрана 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. 2.3.

<sup>1</sup>Текстот во делот 2.1 користи исечоци 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 овде: Kurzgesagt – In a Nutshell: What Are You?<https://www.youtube.com/watch?v=JQVmDkZT4>

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

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



**Сл. 2.1:** 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) цитоплазма, 13) центриоли.<sup>7</sup>

**Цитоплазмата** е полутечна маса со зрнеста структура, изградена од вода и минерални соли, неоргански и органски материји: шеќери, масти и белковини. Таа содржи цитоскелет кој и го одржува обликот и овозможува транспорт на метии низ неа. Зрнестата структура на цитоплазмата ја сочинуваат посебни делчиња наречени органели во кои се вршат животните процеси. Познати органели се:

- **јадрото** е органела која ја има во секоја клетка. Јадрото е обвиткано со јадрена обивка 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 разградување на материите и се ослободува енергија 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 на резервни материји како вода, масти кај масните клетки и други органски материји. Вакуолите, воедно, служат и за исфрлање на непотребните материји од клетката,
- **гликоген** е сложен шеќер кој се складира како резервен шеќер во цитоплазмата на 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.

<sup>5</sup>Овој процес е претставен преку анимација во видеото: From DNA to protein - 3D <https://www.youtube.com/watch?v=gG7uCskUOrA>

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

<sup>7</sup>Од MesserWoland и Szczepan1990 — сопствено дело (создадена со Inkscape), CC BY-SA 3.0, <https://commons.wikimedia.org/w/index.php?curid=1279365>

**Табела 2.1:** 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

Процентната застапеност на некои од нив во клетките од човековиот организам е дадена во Табелата 2.1. 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 нивната процентна 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.

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

- **Јаглеидратите**, 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, и целулозата кај растенијата. Ошто земено, јаглеидратите се извор на енергија на клетките. Со нивното разложување, како краен производ се добива јаглероден диоксид и вода и се ослободува енергија.<sup>8</sup> Освен тоа, тие влегуваат во составот на мембраниите, 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

<sup>8</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=dBnniuab-oM>

клетките најчесто се јавуваат како резерви. По потреба, особено при подолгото гладување, тие се разложуваат до  $\text{CO}_2$  и  $\text{H}_2\text{O}$ , ослободувајќи притоа големо количство енергија. Освен енергетска, овие соединенија имаат и структурна и заштитна функција, бидејќи учествуваат во изградбата на клеточната мембра и другите мембрани системи во клетката. Разликуваме прости липиди кои во својот состав содржат виши масни киселини кои се делат на незаситени и заситени<sup>9</sup>, и сложени липиди кои содржат и нелипидно соединение. Еден пример за сложени липиди се фосфолипидите кои содржат фосфорната киселнина 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 – главно во мускулните клетки, тоа се протеините актин и миозин,
  - антителата, кои се протеини, имаат заштитна функција,
  - протеините во вид на ензими, ги убрзуваат биохемиските процеси во клетката итн.
- Нуклеинските киселини се соединенија со големо биолошко значење, бидејќи го изградуваат 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).

## 2.2 Деполаризација на клетка

Во основата зад електричните промени кои настануваат во организмот, којшто може да бидат детектирани преку соодветни сензори, стои процесот на деполаризација на една клетка. Деполаризацијата<sup>10</sup> во биолошки смисол, подразбира кога во клетката настанува ненадејна и драматична промена во електричен домен, Сл. 2.2. Во својата стационарна состојба, клетките ја одржуваат својата внатрешност на негативен електричен потенцијал, споредено со нивното опкружување, преку одржувањето на различна концентрација на позитивните и негативните јони.

Во општ случај, без влијание на надворешното електрично поле, непосредно од внатрешната страна на клеточната мембра се натрупува вишок негативни јони, а позитивните јони се натрупуваат на надворешната страна на мембраната. На тој начин во близина на мембраната 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 спроводливост, се двете електроди, а клеточната мембра со својата мала спроводливост ја има улогата на диелектрик кој ги разделува електродите.<sup>12</sup>

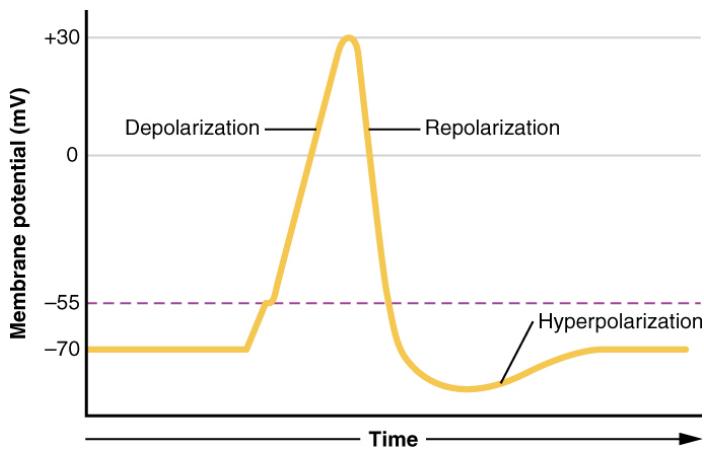
Разликата во потенцијалите помеѓу внатрешноста и надворешноста на клетката се нарекува потенцијал на клеточната мембра или биоелектричен потенцијал во мирување. Сите електрични процеси во човековото тело се должат на промената на овие

<sup>9</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/>.

<sup>10</sup>Wikipedia: Depolarization <https://en.wikipedia.org/wiki/Depolarization>

<sup>11</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>12</sup>Текстот во овој поднаслов користи исчеоци од Wikipedia: Биоелектричен потенцијал [https://mk.wikipedia.org/wiki/Биоелектричен\\_потенцијал](https://mk.wikipedia.org/wiki/Биоелектричен_потенцијал)



Сл. 2.2: Деполаризација на една клетка.<sup>11</sup>

потенцијали (Гершановски). Големината на биоелектричниот потенцијал може да се мери со помош на специјална микро-електрода со која се пробива клеточната мембра 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.

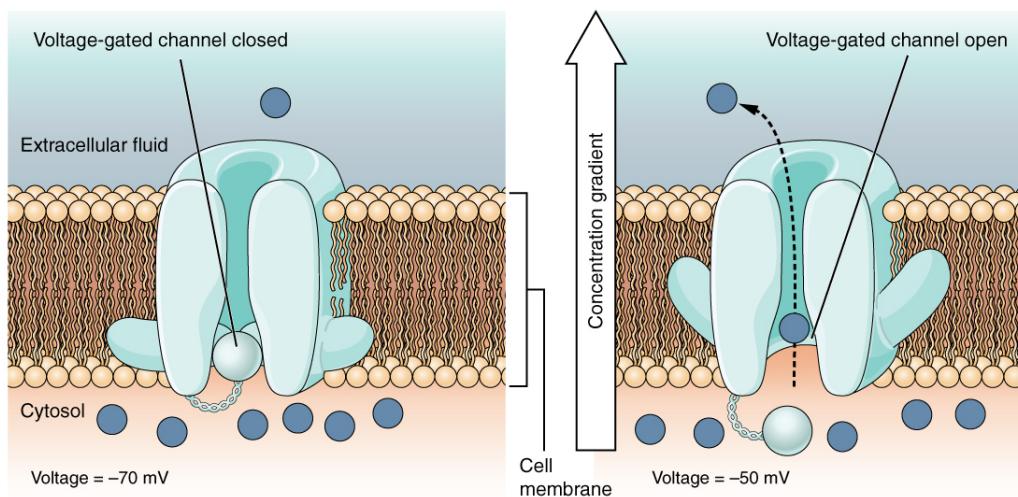
Најважна улога во создавањето на биоелектричниот потенцијал има дифузијата на натриумовите  $\text{Na}^+$  и калиумовите  $\text{K}^+$  јони преку клеточната мебрана, која е резултат на дејството на таканаречените **натриум-калиумови пумпи**. Преку нив за секои два внесени  $\text{K}^+$  јони, од клетката се испуштуваат три  $\text{Na}^+$  јони со употреба на еден молекул на АТФ.<sup>13</sup> На овој начин не само што се намалува електричниот потенцијал на клетката, туку и се создаваат услови за нејзина брза деполаризација. Така, екстрацелуларната течност содржи голема концентрација на  $\text{Na}^+$  јони кои не можат да проникнат во клетката поради тоа што пропустливоста на мембраната на  $\text{Na}^+$  јони е 100 пати поголема од пропустливоста за  $\text{K}^+$  јони. Оваа нерамнотежа ја одржуваат и **напонски регулирани јонски канали** кои биоелектричниот потенцијал ги држи затворени, Сл. 2.3. И покрај високата концентрација на  $\text{K}^+$  во внатрешноста на клетката, таа вообичаено содржи и негативно наелектризирани честички кои ја воспоставуваат негативната поларизација.

Откако клетката ќе го воспостави биолектричниот потенцијал на мирување, таа може да биде побудена и да оствари деполаризација. Побудувањето на клетките може да се оствари на повеќе начини и тоа под:

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

При деполаризацијата, за време помало од 1 ms, надворешната страна на клеточната мембра станува негативна во однос на внатрешната, Сл. 2.2. Промената на мембранныот потенцијал при побудување на клетките се нарекува **акционен потенцијал**. Во почетниот стадиум на деполаризација, преку отворање на напонски регулирани јонски канали предизвикано од електричната побуда, Сл. 2.3, се зголемува пропустливоста на мембраната на јоните на натриумот  $\text{Na}^+$  па тие во голем број навлегуваат во клетката поларизирајќи ја позитивно нејзината внатрешност. Ова се случува поради отворање на напонски регулирани јонски канали предизвикано од побудата на клетката, Сл. 2.3. 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.

<sup>13</sup>Овој процес, како и целиот процес на деполаризација кај нервните импулси се претставени како анимација во видеото Nerve Impulse Molecular Mechanism [3D Animation] <https://www.youtube.com/watch?v=fHRC8S1LcHO>



Сл. 2.3: Напонски регулирани јонски канали.<sup>14</sup>

Брзата дифузија на натриумовите јони трае од 0,1 до 1 ms. Создадениот позитивен полнеж во внатрешната страна на мемраната го забавува натамошното навлегување на натриумовите јони. Тој исто така ги активира и каналите за  $K^+$  со што калиумовите јони започнуваат да ја напуштаат клетката. 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>15</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 состојба во која не можат да бидат побудени, дури и при доаѓање на нов надворешна побуда, како на пример надворешен акционен потенцијал. Ова време се нарекува **време на непобудливост** и тоа игра клучна улога во еднонасочното простирање на акционите потенцијали.

Промената на биоелектричниот потенцијал за време на деполаризацијата го овозможува преносот на електрични импусли 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.

## 2.3 Нервни импусли

### Градба на неврон

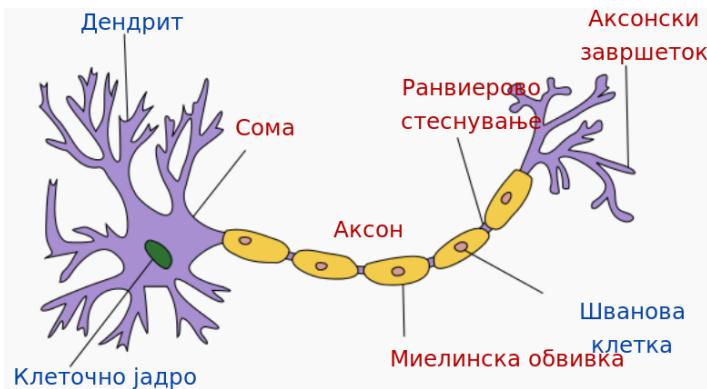
Нервните клетки или **невроните** се основните градбени единки на нервниот систем.<sup>16</sup> За разлика од другите клетки во телото, невроните се карактеризираат со надразливост, осетливост и спроводливост. Овие карактеристики се во согласност со нивната специфична функција. Градбата на невроните е прикажана на Сл. 2.4 и содржи:

- тело на нервната клетка или невроцит,

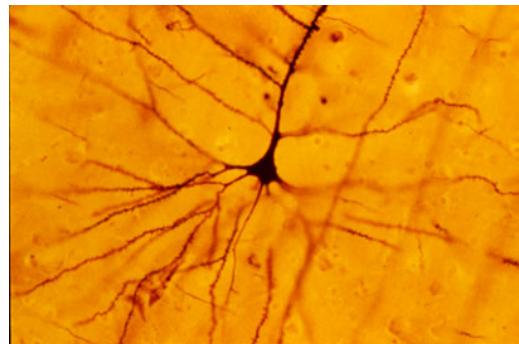
<sup>14</sup>By OpenStax College - Anatomy & Physiology, Connexions Web site. <http://cnx.org/content/col11496/1.6/>, Jun 19, 2013., CC BY 3.0, <https://commons.wikimedia.org/w/index.php?curid=30147924>

<sup>15</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)

<sup>16</sup>Текстот во овој параграф користи исечоци од Wikipedia: Неврон <https://mk.wikipedia.org/wiki/Неврон>



Сл. 2.4: Градба на невроните.<sup>17</sup>



Сл. 2.5: Човечки неврон виден под микроскоп.<sup>18</sup>

- дендрити (влезови), и
- нервно влакно или аксон (излез).

На Сл. 2.5 е прикажан неокортikalен 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. 2.6. 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**.

Простирањето на деполаризацијата долж невронот е прикажана на Сл. 2.7, а фазите во простирањето на Сл. 2.8.<sup>20</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. По простирањето на нервниот импулс низ аксонот на невронот, невронот мора повторно да го воспостави биолектричниот потенцијал

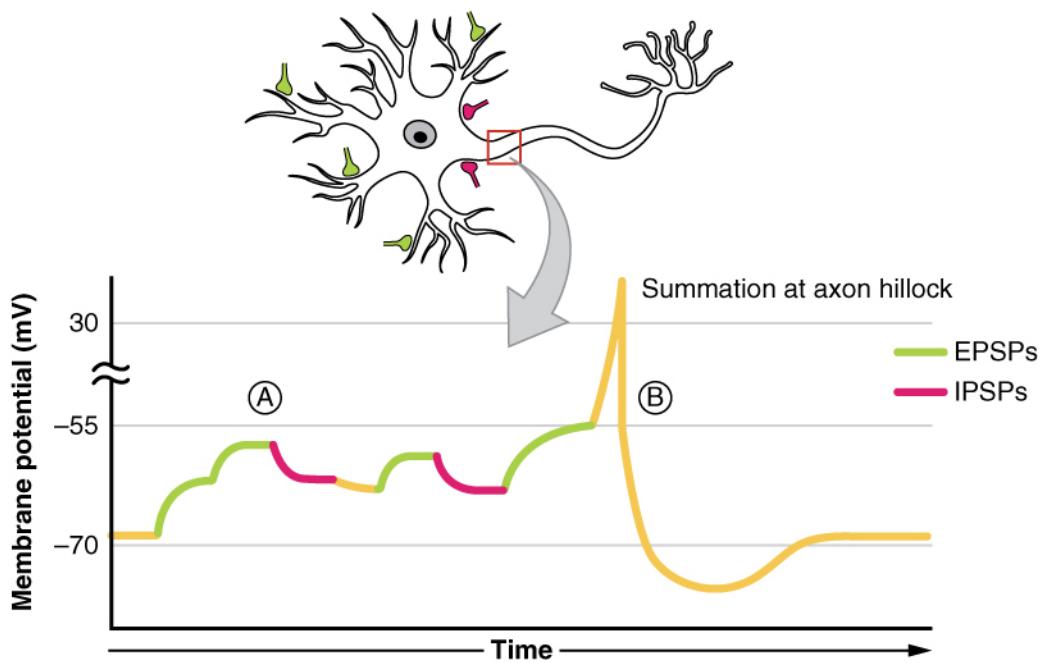
<sup>17</sup> Wikipedia: Неврон <https://mk.wikipedia.org/wiki/Неврон>, оригинал од Quasar Jarosz на Википедија на англиски, CC BY-SA 3.0, <https://commons.wikimedia.org/w/index.php?curid=7616130>

<sup>18</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>

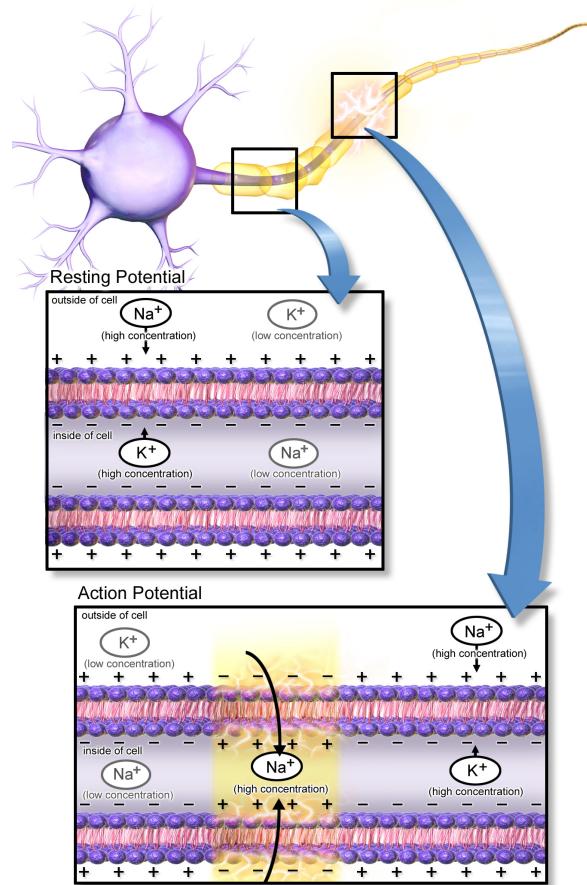
<sup>19</sup> By OpenStax College - Anatomy & Physiology, Connexions Web site. <http://cnx.org/content/col11496/1.6/>, Jun 19, 2013., CC BY 3.0, <https://commons.wikimedia.org/w/index.php?curid=30147931>

<sup>20</sup> Wikipedia: Action potential [https://en.wikipedia.org/wiki/Action\\_potential](https://en.wikipedia.org/wiki/Action_potential)

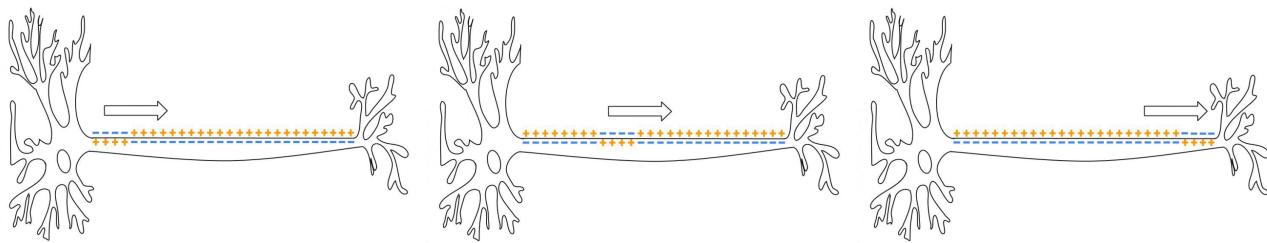
<sup>21</sup> Blausen.com staff. "Blausen gallery 2014". Wikiversity Journal of Medicine. DOI:10.15347/wjm/2014.010. ISSN 20018762. - Own work, CC BY 3.0, <https://commons.wikimedia.org/w/index.php?curid=29452220>



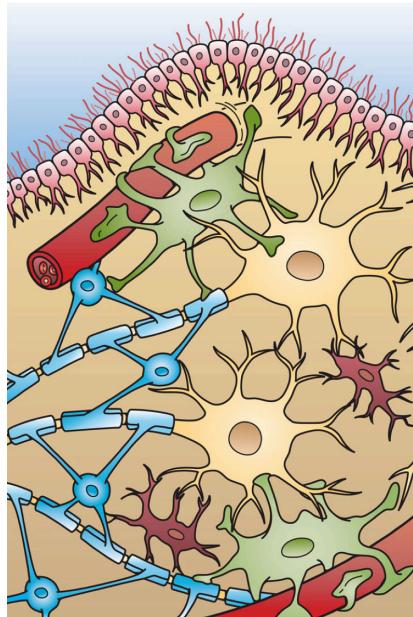
Сл. 2.6: Интегрирање на побудите во телото на невронот.<sup>19</sup>



Сл. 2.7: Простирање на нервниот импулс.<sup>21</sup>



**Сл. 2.8:** Фази од простирањето на нервниот импулс.<sup>22</sup>



**Сл. 2.9:** Глиа клетки во централниот нервен систем (од горе надолу): епендимални клетки (розово), астроцити (зелено), микроглиални клетки (темно црвено) и олигодендроцити (сино).<sup>25</sup>

во мирување. Пред невронот да може да проследи нов нервен импулс, треба да помине времето на непобудливост.<sup>23</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>24</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. 2.9, се:

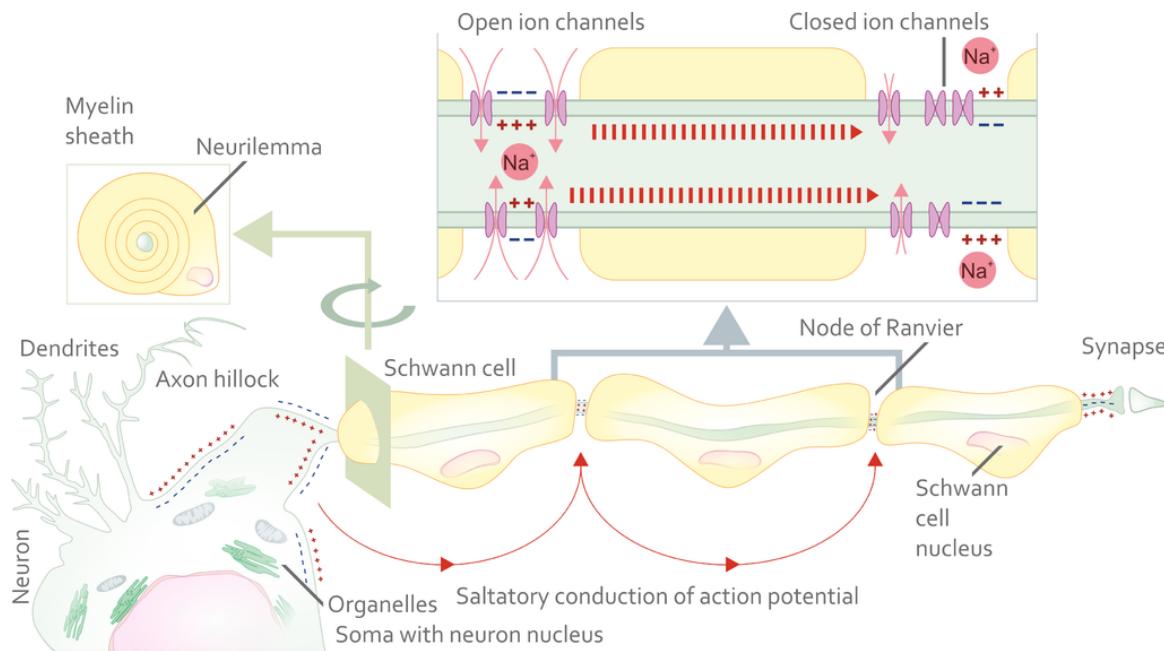
- епендималните клетки – одговорни за излачување и придвижување на мозочно-спиналната течност,
- астроцитите – задолжени за пренос на хранливите материји од капиларите до невроните и отпадните материји од невроните во спротивна насока тие се чуварите на мозочно-крвната бариера,

<sup>23</sup>Како што кажавме погоре, добра илустрација на целиот процес на простирање на нервните импулси е дадена во видеото Nerve Impulse Molecular Mechanism [3D Animation] <https://www.youtube.com/watch?v=fHRC8S1LcH0>

<sup>24</sup>By Laurentaylorj - Own work, CC BY-SA 3.0, <https://commons.wikimedia.org/w/index.php?curid=26311114>

<sup>25</sup>Wikipedia: Glia <https://en.wikipedia.org/wiki/Glia>

<sup>25</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>



Сл. 2.10: Простирање на нервниот импулс во миелизиран неврон со скокови.<sup>26</sup>

- 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. 2.10. 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>27</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.

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. 2.11. 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 графикот екстраполираните вредности се прикажани со испрекинати линии.

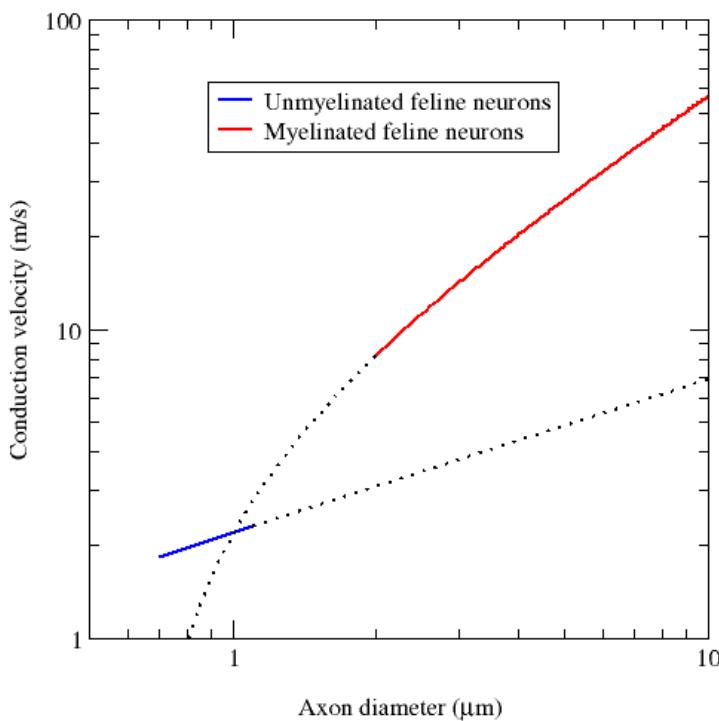
## 2.4 Градба и работа на мускулите

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

<sup>26</sup>By Helixitta - Own work, CC BY-SA 4.0, <https://commons.wikimedia.org/w/index.php?curid=44043108>

<sup>27</sup>Wikipedia: Saltatory conduction [https://en.wikipedia.org/wiki/Saltatory\\_conduction](https://en.wikipedia.org/wiki/Saltatory_conduction)

<sup>28</sup>By WillowW - Own work, CC BY 3.0, <https://commons.wikimedia.org/w/index.php?curid=3844705>



**Сл. 2.11:** Влијание на миелинската обвивка врз брзината на простирање на нервниот импулс.<sup>28</sup>

на локомоторниот апарат.<sup>29</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, на пр. гастроинтестиналниот тракт и уринарниот систем; 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. 2.13. 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 миофибрилите, скелеталните, а и срцевите мускулни влакна имаат пругаст, т.е. стриест изглед.

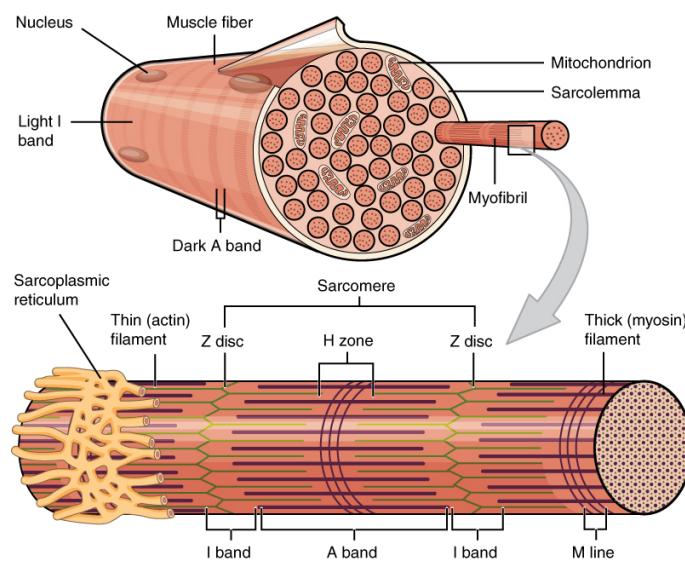
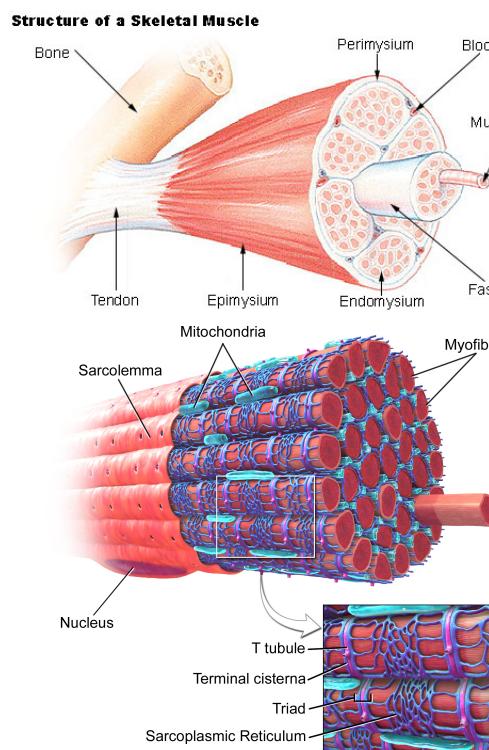
Контракциите на скелетните мускули се **неврогенични**, односно тие се предизвикани од нервни импулси на моторните неврони, наречени мотоневрони, кои предизвикуваат деполаризација на

<sup>29</sup> Википедија – Мускул <https://mk.wikipedia.org/wiki/Мускул>

<sup>30</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>31</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>



**Сл. 2.13:** Структура и состав на скелеталните мускулни влакна.<sup>31</sup>

**Сл. 2.12:** Структура и состав на скелеталните мускули.<sup>30</sup>

мускулните влакна, а со тоа и нивна **контракција**.<sup>32</sup> Притоа, мотоневроните се разгрануваат 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 самите се одговорни за предизвикување на сопствената контракција, иако нивната работа може да биде модулирана од автономниот нервен систем.

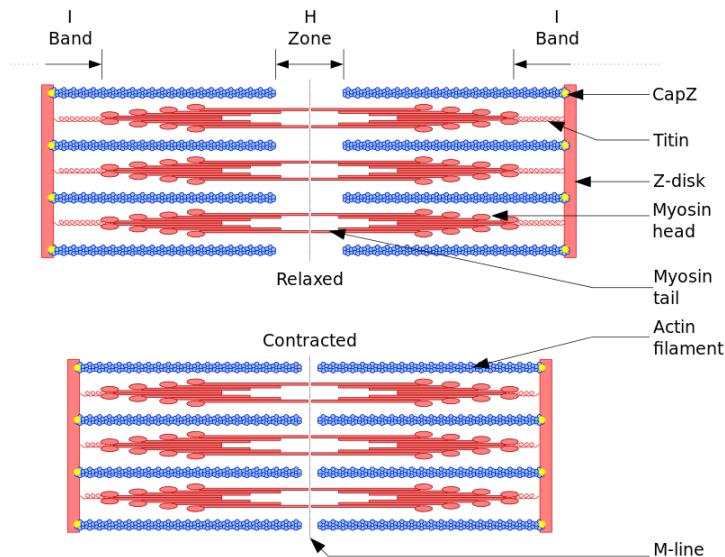
Покрај различниот тип на влезна побуда на која реагираат, контракцијата на сите три типа на мускулно ткиво се должи на истиот физиолошки механизам. Имено, при активација протеинските нишки на **актин** и **миозин** во мускулните влакна според **теорија на лизгање на нишките**, тие лизгаат едни покрај други генерирајќи сила и придвижување, т.е. контракција. На Сл. 2.14 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 по чекор на Сл. 2.15.<sup>34</sup> Деполаризацијата на мускулното влакно ја предизвикува надворешниот акционен потенцијал на мотоневронот. Таа се пренесува низ Т тубулите и ги активира напонски контролирани калциумови пумпи 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

<sup>32</sup>Wikipedia: Muscle Contraction. [https://en.wikipedia.org/wiki/Muscle\\_contraction](https://en.wikipedia.org/wiki/Muscle_contraction)

<sup>33</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>34</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. 2.14:** Меѓусебната положба на протеинските нишки при релаксација, односно контракција.<sup>33</sup>

миозинот.

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>36</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-енергетската состојба.

Во следниот чекор нова АТФ молекула се врзува за миозинот, што го ослободува од актинот.<sup>37</sup> Неговата хидролиза на АДФ и Р го враќа во почетната високо-енергетска состојба. Ако тропомиозинот сеуште не го покрива актинот, целиот процес се повторува. Во целиот процес калциумот  $\text{Ca}^{2+}$  активно се впумпува 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>38</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  $\text{K}^+$  нанадвор. 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 **миограмот** на Сл. 2.16. 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

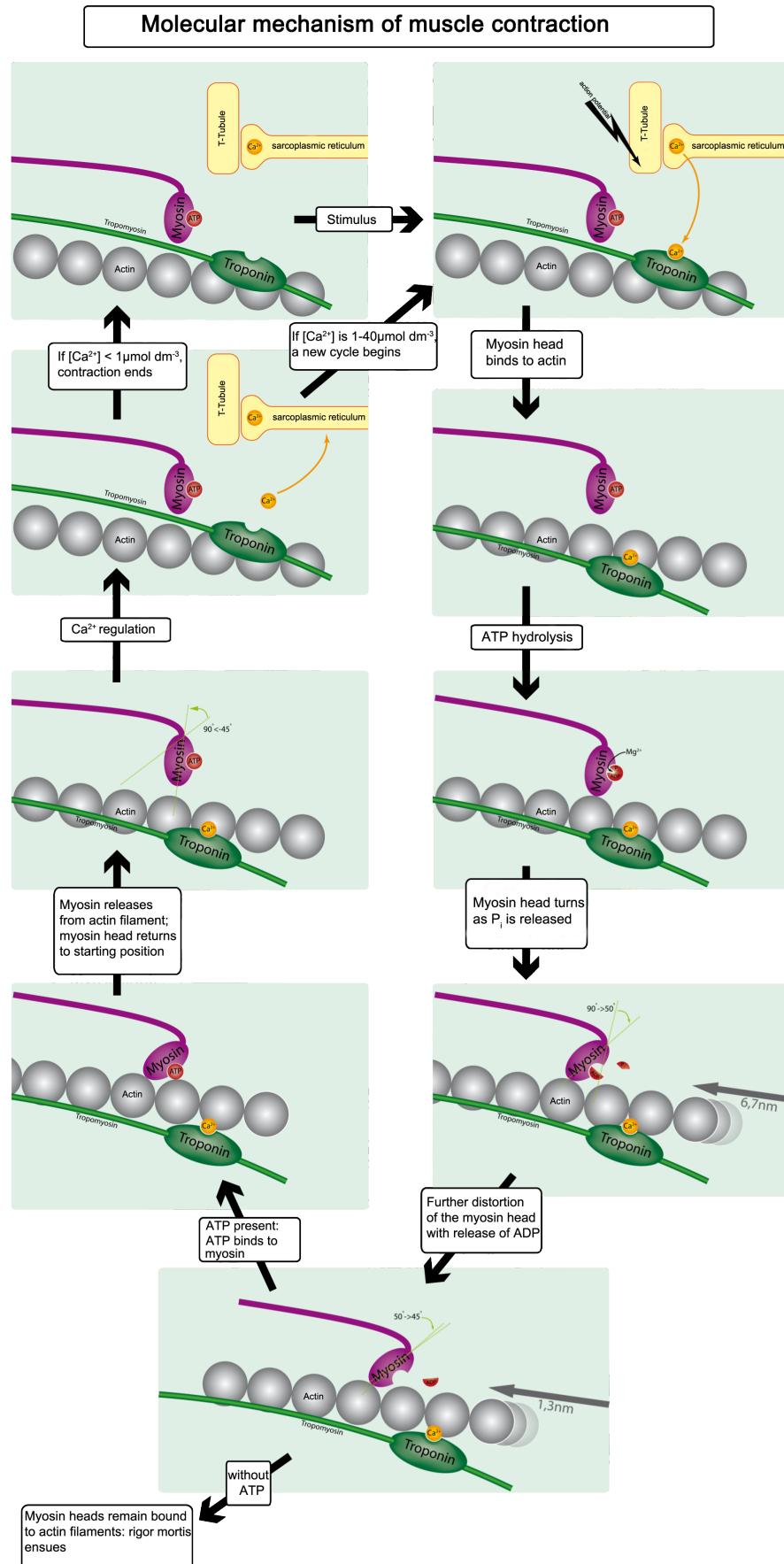
<sup>35</sup> By Muskel-molekular.png: Hank van Helvete derivative work: GravityGilly - Muskel-molekular.png, CC BY-SA 2.5, <https://commons.wikimedia.org/w/index.php?curid=10703870>

<sup>36</sup> Adenosine triphosphate (ATP) [https://en.wikipedia.org/wiki/Adenosine\\_triphosphate](https://en.wikipedia.org/wiki/Adenosine_triphosphate)

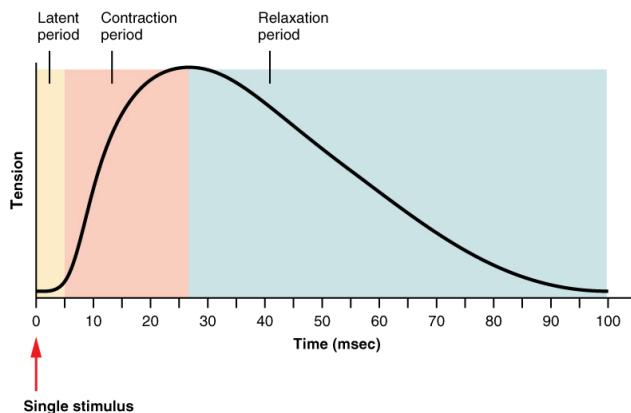
<sup>37</sup> При недостиг на АТФ, мускулите остануваат во грч.

<sup>38</sup> Wikipedia: Action potential [https://en.wikipedia.org/wiki/Action\\_potential](https://en.wikipedia.org/wiki/Action_potential)

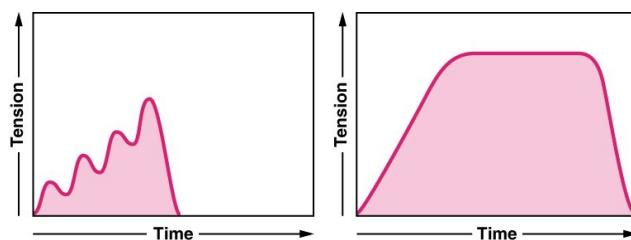
<sup>39</sup> By OpenStax - <https://cnx.org/contents/FPtK1zmh@8.25:fEI3C80t@10/Preface>, CC BY 4.0, <https://commons.wikimedia.org/w/index.php?curid=30015047>



Сл. 2.15: Чекори во контракцијата на мускулните влакна.<sup>35</sup>



Сл. 2.16: Миограм на елементарна контракција на мускулно влакно.<sup>39</sup>



Сл. 2.17: Миограм на сумација и тетанус при контракција на мускул.<sup>40</sup>

релаксирана состојба доаѓа до таканаречена **сумација**. Ова е прикажано на Сл. 2.17. 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 активација на нови мускулни влакна, т.е. моторни 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. 2.17.

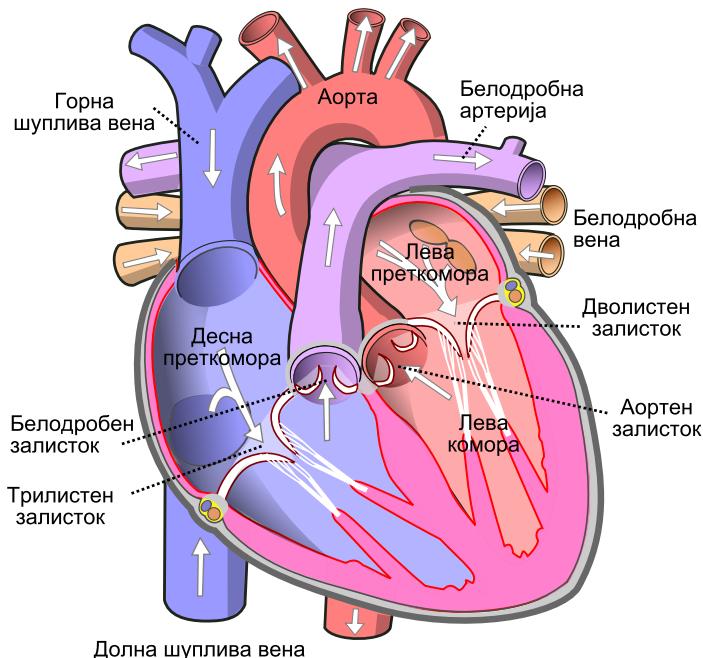
## 2.5 Градба и работа на човековото срце

Срцето е празен мускулен орган чија функција е движењето на кrvта во организмот.<sup>41</sup> Срцевиот сид 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 тристрана пирамида, со врвот завртена на надолу и налево, со тежина од 230 до 340 g. Срцето лежи во срцевото ќесе - перикард, кое има заштитна функција.

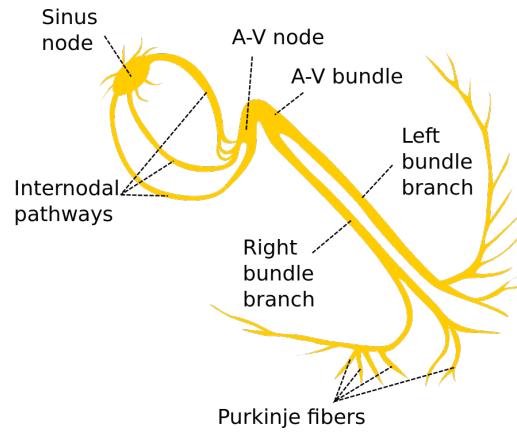
Срцето содржи четири комори: две **преткомори** и две главни **комори**. Мускулните сидови на преткоморите се потенки од тие на коморите, поради тоа што значаен удел во придвижување на кrvта од преткоморите во коморите зема гравитационата сила. Мускулниот сид на левата комора е подебел од оној на десната, бидејќи тука кrvта се испуштува со голем притисок во

<sup>40</sup>By OpenStax - <https://cnx.org/contents/FPtKizmh@8.25:fEI3C80t@10/Preface>, CC BY 4.0, <https://commons.wikimedia.org/w/index.php?curid=30015045>

<sup>41</sup>Википедија – Срце <https://mk.wikipedia.org/wiki/Срце>



Сл. 2.18: Анатомија на човечкото срце.<sup>43</sup>



Сл. 2.19: Систем за пренос на деполаризација во срцето.<sup>44</sup>

аортата за да се разнесе низ целото тело преку големиот крвоток. Од друга страна, десната комора ја испуштува 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 (АВ) 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 (ПМ) мембрани, обете тролисни.<sup>42</sup>

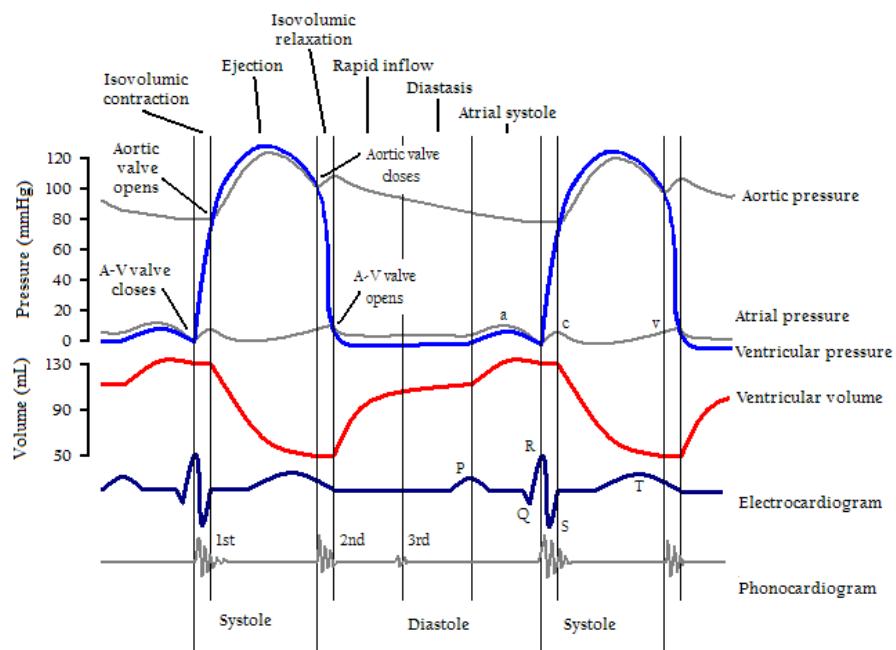
Во срцевиот циклус постојат 5 фази, во кои мускулите од преткоморите и коморите вршат контракција и релаксација, а атриовентрикуларните и полумесечевите мембрани се отвораат и затвораат. Тие се:

1. Контракција на преткомори – АВ затворени, ПМ отворени, преткоморите ги надополнуваат коморите со околу 30% од волуменот на крвта при мирување од 130 ml.
2. Изоловумена контракција на комори – АВ затворени, ПМ затворени, коморите започнуваат со контракција, нивниот волумен не се менува но притисокот расте.
3. Контракција на комори со испуштување – АВ затворени, ПМ отворени, притисокот во коморите го надминува оној во аортата (80 mmHg) и пулмонарната артерија (10 mmHg), исфирањето е ненадејно и истиснува 70 – 90 ml од секоја комора, што остава околу 50 ml за следната контракција.
4. Изоловумена 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.

<sup>42</sup>Wikipedia – Heart <https://en.wikipedia.org/wiki/Heart>

<sup>43</sup>Od 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>44</sup>By Madhero88 - Own work, CC BY-SA 3.0, <https://commons.wikimedia.org/w/index.php?curid=6887499>.

Сл. 2.20: Срцев циклус.<sup>47</sup>

За време на секое отчукување срцето минува редоследно низ овие фази од срцевиот циклус<sup>45</sup> прикажан на Сл. 2.20, 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>46</sup>, кој е одговорен за пренесување на електричните импулси на деполаризацијата низ срцевиот мускул прикажан на Сл. 2.19. АВ јазолот го внесува потребното задоцнување во простирањето на деполаризацијата за контракцијата коморите да дојде дури по постигнување на максималната контракција на преткоморите. Нишките не 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>45</sup> Wikipedia: Cardiac cycle [https://en.wikipedia.org/wiki/Cardiac\\_cycle](https://en.wikipedia.org/wiki/Cardiac_cycle)

<sup>46</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>47</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>

## Поглавје 3

# Електронски кола за мерење на биопотенцијалите

### 3.1 Електромиографија

Електромиографијата е електродијагностичка медицинска техника за евалуација и снимање на електричната активност на скелеталните мускули.<sup>1</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.

#### Површинска ЕМГ

Површинската 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. 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 потковожното 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 луѓето 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

<sup>1</sup>Wikipedia – Electromyography <https://en.wikipedia.org/wiki/Electromyography>

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 близокост на двата 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.

## Поставување на електродите

Пред поставувањето на електродите за ЕМГ, кожата треба соодветно да се подготви, што вообичаено вклучува 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.

## 3.2 Електрокардиограм

Електрокардиограмот (ЕКГ)<sup>2</sup> е запис на електричната активност на срцето<sup>3</sup>, Сл. 2.18, 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. Вообичаено 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 вообичаено 10 s. На овој начин, се забележува вкупната амплитуда и насоченост на деполаризацијата во секој момент од срцевиот циклус<sup>4</sup> прикажан на Сл. 2.20.

## Облик на ЕКГ сигналот

Прогресијата на деполаризација на срцето го дава карактеристичниот облик на ЕКГ сигналот прикажан на Сл. 3.1. За лекарите, обликот на ЕКГ носи големо количество 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.

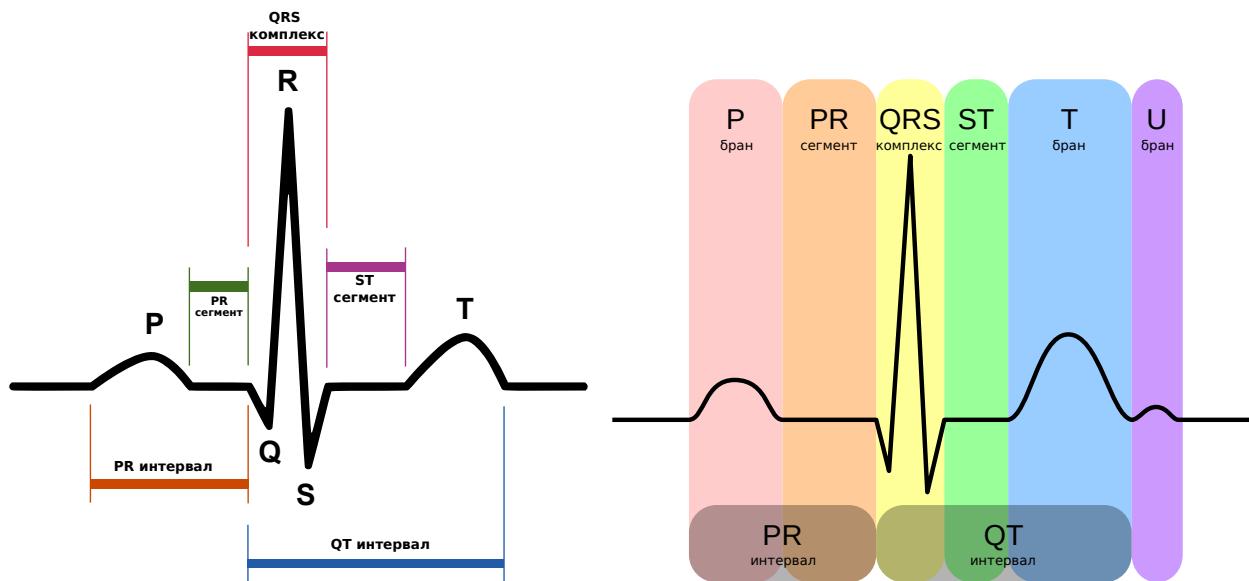
Карактеристични точки во ЕКГ записот се:

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

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

<sup>3</sup>Wikipedia: Heart <https://en.wikipedia.org/wiki/Heart>

<sup>4</sup>Wikipedia: Cardiac cycle [https://en.wikipedia.org/wiki/Cardiac\\_cycle](https://en.wikipedia.org/wiki/Cardiac_cycle)



Сл. 3.1: Карактеристични точки и интервали на ЕКГ сигналот.<sup>5</sup>

- 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 U бран кој одговара на задоцнетата реполаризација на нишките на Пуркинџе, но тој не е секогаш таму.

Помеѓу карактеристичните точки се дефинирани следните интервали и сегменти:

- PR интервал – од почетокот на Р бранот до почетокот на QRS комплексот, го рефлектира времето потребно за електричниот импулс да пристигне од синусниот јазол до атровентрикуларниот (AB) јазол,
- PR сегмент – од крајот на Р бранот до почетокот на QRS комплексот, го рефлектира времето кога преткоморите се деполаризирани,
- ST сегмент – го поврзува QRS комплексот со Т бранот и го претставува периодот во кој коморите се деполаризирани,
- QT интервал – од почетокот на QRS комплексот до крајот на Т бранот, бидејќи варира со брзината на работа на срцето, вообичаено се изразува како QTc корегирано со делење 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 3.1. 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.

<sup>5</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>

**Табела 3.1:** 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

### 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 (стерnumот),
- **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.

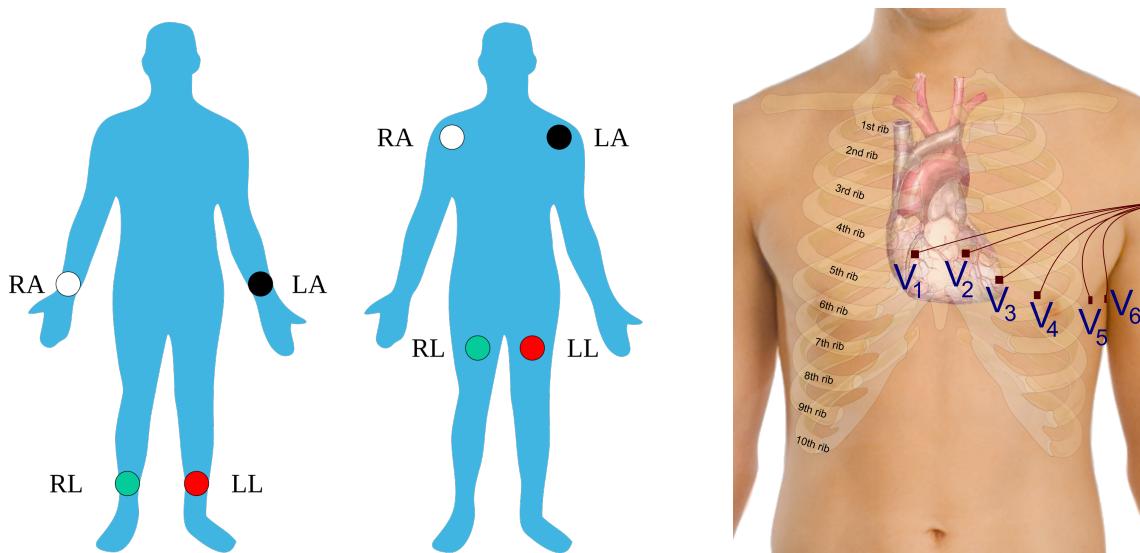
Сите овие се користат за добивање на ЕКГ сигналот долж една оска, освен LL која се употребува за заземјување на уредот за мерење. Ова е неопходно за поништување на индуцираните електромагнетни шумови од напонот на градската мрежа. Поставувањето на десетте електроди на телото е прикажано на Сл. 3.2.

### Снимање на ЕКГ сигналот

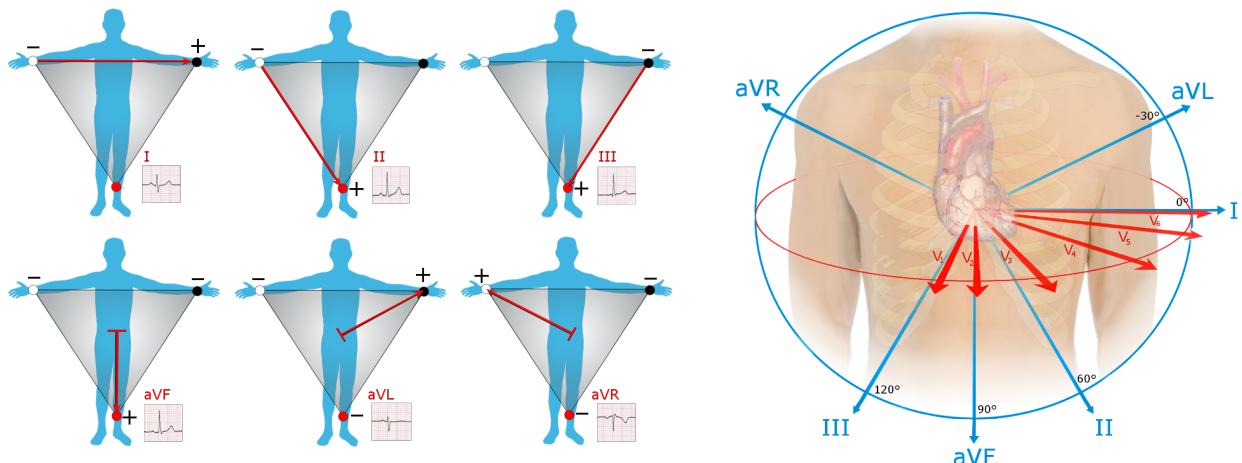
Дванаесетте канали снимени од професионалните ЕКГ уреди се поделени во три типови: 3 напони на екстремитетите, 3 аугментирани напони на екстремитетите, и 6 градни напони. Шестте канали

<sup>6</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>7</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>



Сл. 3.2: Поставување на електродите за снимање на ЕКГ сигналот.<sup>6</sup>



Сл. 3.3: Ориентација на напоните кои ги регистрира 12-каналниот ЕКГ уред.<sup>7</sup>

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. 3.3. Трите напони на екстремитетите се означени како I, II и III. Тие се пресметуваат како:

$$I = LA - RA, \quad (3.1)$$

$$II = LL - RA, \quad (3.2)$$

$$III = LL - LA. \quad (3.3)$$

Аугментираниите 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:

$$aVR = RA - \frac{1}{2}(LA + LL) = \frac{3}{2}(RA - V_W), \quad (3.4)$$

$$aVL = LA - \frac{1}{2}(RA + LL) = \frac{3}{2}(LA - V_W), \quad (3.5)$$

$$aVF = LL - \frac{1}{2}(RA + LA) = \frac{3}{2}(LL - V_W), \quad (3.6)$$

каде:

$$V_W = \frac{1}{3}(RA + LA + LL). \quad (3.7)$$

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

## Поглавје 4

# Обработка 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 [Питон](#). За вовед во екосистемот на Питон направен за инженерска и научна работа и за основните системски поставувања погледнете во [Додатокот Б](#).

### 4.1 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 отсекоци \(ФТВО\)](#)<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 Фурье**<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.

## Фуриеов интеграл

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

$$X(\omega) = \int_{-\infty}^{\infty} x(t) e^{-j\omega t} dt, \quad \omega \in (-\infty, \infty), \quad (4.1)$$

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

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

$$x(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} X(\omega) e^{j\omega t} d\omega, \quad t \in (-\infty, \infty). \quad (4.2)$$

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

$$X(\omega) = |X(\omega)| e^{j\angle X(\omega)} = A(\omega) e^{j\phi(\omega)}, \quad (4.3)$$

каде  $A(\omega)$  претставува **амплитуден спектар**, а  $\phi(\omega)$  **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 (4.2), користејќи ја **Ојлеровата формула**, добиваме:

$$x(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} A(\omega) e^{j\phi(\omega)} e^{j\omega t} d\omega, \quad (4.4)$$

$$= \frac{1}{2\pi} \int_{-\infty}^{\infty} A(\omega) [\cos(\omega t + \phi(\omega)) + j \sin(\omega t + \phi(\omega))] d\omega, \quad (4.5)$$

$$= \frac{1}{\pi} \int_0^{\infty} A(\omega) \cos(\omega t + \phi(\omega)) d\omega. \quad (4.6)$$

Основните својства на Фуриеовата трансформација се дадени во **Табелата 4.1**.

Сигналот  $x(t)$  и неговата Фуриеова трансформација  $X(\omega)$  чинат **Фуриеов пар** што се означува како  $x(t) \leftrightarrow X(\omega)$ . За Фуриеовата 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 4.2** се дадени некои позначајни Фуриеови парови.

**Табела 4.1:** Позначајни својства на Фуриевата трансформација.

Својство	Временски домен	Фуриев домен
Линеарност	$ax(t) + bg(t)$	$aX(\omega) + bG(\omega)$
Поместување	$x(t - t_0)$	$e^{-jt_0\omega} X(\omega)$
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 4.2:** 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>:

$$E = \int_{-\infty}^{\infty} |x(t)|^2 dt = \frac{1}{2\pi} \int_{-\infty}^{\infty} |X(\omega)|^2 d\omega = \int_{-\infty}^{\infty} |X(2\pi ft)|^2 df \quad (4.7)$$

\* Важно! Во Табелата 4.2 забележете дека Фуриевата 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:

$$x(t) \equiv x(t + T), \quad \forall t. \quad (4.8)$$

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

$$\hat{x}(t) = x(t) \cdot \Pi(t), \quad (4.9)$$

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

каде:

$$\Pi(t) = \begin{cases} 1, & |t| \leq \frac{T}{2}, \\ 0, & |t| > \frac{T}{2}. \end{cases}, \quad (4.10)$$

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

$$x(t) = \sum_{r=-\infty}^{\infty} \hat{x}(t - rT). \quad (4.11)$$

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**:

$$x(t) = \sum_{k=-\infty}^{\infty} c_k e^{jk\frac{2\pi}{T}t} = \sum_{k=-\infty}^{\infty} c_k e^{jk\omega_0 t}, \quad (4.12)$$

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

$$\omega_0 = 2\pi f_0 = \frac{2\pi}{T}. \quad (4.13)$$

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:

$$c_k = \frac{1}{T} \int_{-T/2}^{T/2} x(t) e^{-jk\frac{2\pi}{T}t} dt. \quad (4.14)$$

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

$$X(\omega) = 2\pi \sum_{k=-\infty}^{\infty} c_k \delta(\omega - k\omega_0). \quad (4.15)$$

## $\mathcal{Z}$ 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:

$$X(z) = \sum_{n=-\infty}^{\infty} x(n) z^{-n}, \quad (4.16)$$

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

$$x(n) = x(nT_s), \quad (4.17)$$

$$T_s = \frac{1}{f_s}. \quad (4.18)$$

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

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

$$x(n) = \frac{1}{2\pi j} \oint_C X(z) z^{n-1} dz, \quad (4.19)$$

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 4.3](#), а позначајни  $\mathcal{Z}$  трансформации во [Табелата 4.4](#).

**Табела 4.3:** 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 4.4:** 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 $

### Фуриеова трансформација на дискретен сигнал

Фуриеовата трансформација на дискретен сигнал може да се добие од равенката за  $\mathcal{Z}$  трансформацијата (4.16) со замената на јадрото  $z = e^{-j\omega T_s}$ , што се поедноставува на  $z = e^{-j\omega}$  за  $T_s = 1$ :

$$X(\omega) = \sum_{n=-\infty}^{\infty} x(n)e^{-j\omega n}, \quad (4.20)$$

$$x(n) = \frac{1}{2\pi} \int_{-\pi}^{\pi} X(\omega)e^{j\omega n} d\omega. \quad (4.21)$$

Во овој случај, ја ограничуваме  $\mathcal{Z}$  трансформацијата на единствната 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 Фуриеовата 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 произволно  $T_s$ , 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  $f$ , како што беше дискутирано во [Поглавјето ??](#).

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

### Дискретна Фуриеова трансформација

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

$$\tilde{x}(n) = \tilde{x}(n + N), \quad -\infty < n < \infty, \quad (4.22)$$

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

$$\tilde{X}(k) = \sum_{n=0}^{N-1} \tilde{x}(n) e^{-jk\frac{2\pi}{N}n}, \quad (4.23)$$

$$\tilde{x}(n) = \frac{1}{N} \sum_{k=0}^{N-1} \tilde{X}(k) e^{jk\frac{2\pi}{N}n}. \quad (4.24)$$

Ако сега ја земеме Фурьеовата трансформација на дискретниот сигнал  $x(n)$  дадена во (4.20), која претставува  $\mathcal{Z}$  трансформација на сигналот  $x(n)$  долж единечната кружница, и истата ја семплираме на еквидистантни фреквенции  $\omega_k$  дадени со:

$$\omega_k = k \frac{2\pi}{N}, \quad (4.25)$$

тогаш ќе го добијеме изразот за [дискретната Фурьеова трансформација ДФТ](#):

$$X(k) = \sum_{n=0}^{N-1} x(n) e^{-jk\frac{2\pi}{N}n}, \quad (4.26)$$

кој е еквивалентен со (4.23)!. Ова значи дека семплириралиот спектар што го добиваме со (4.34) соодветствува на сигнал  $\tilde{x}(n)$  која претставува периодична верзија од сигналот  $x(n)$  со период  $N$ , односно важи:

$$x(n) = \tilde{x}(n) \cdot \Pi(n), \quad (4.27)$$

каде

$$\Pi(n) = \begin{cases} 1, & 0 \leq n < N, \\ 0, & \text{поинаку.} \end{cases} \quad (4.28)$$

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:

$$\omega_k = k \frac{2\pi}{NT_s}, \quad (4.29)$$

односно:

$$f_k = k \frac{1}{NT_s} = k \frac{f_s}{N}. \quad (4.30)$$

Резолуцијата на спектарот добиен со ДФТ:

$$\Delta\omega = \frac{2\pi}{N}, \quad (4.31)$$

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

$$\Delta\omega = \frac{2\pi}{NT_s}, \quad (4.32)$$

односно:

$$\Delta f = \frac{f_s}{N}. \quad (4.33)$$

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 (4.34), без да ги смени граничите на сумата:

$$X(k) = \sum_{n=0}^{N-1} x(n)e^{-jk\frac{2\pi}{N+N_0}n}. \quad (4.34)$$

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:

$$x(n) = \frac{1}{N} \sum_{k=0}^{N-1} X(k)e^{jk\frac{2\pi}{N}n}. \quad (4.35)$$

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 нејзиниот 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  $\omega_0$ :

$$x(n) = A \cos(\omega_0 n) = A \cos(2\pi f_0 n). \quad (4.36)$$

Спектарот на овој сигнал е даден со равенството:

$$X(\omega) = \frac{A}{2} 2\pi (\delta(\omega + \omega_0) + \delta(\omega - \omega_0)). \quad (4.37)$$

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:

$$\hat{X}(\omega) = \frac{1}{2\pi} \int_{-\infty}^{\infty} X(\Omega) W(\omega - \Omega) d\Omega \quad (4.38)$$

$$= \frac{1}{2\pi} \int_{-\infty}^{\infty} \frac{A}{2} 2\pi (\delta(\Omega + \omega_0) + \delta(\Omega - \omega_0)) W(\omega - \Omega) d\Omega \quad (4.39)$$

$$= \frac{A}{2} (W(\omega + \omega_0) + W(\omega - \omega_0)), \quad (4.40)$$

<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:

$$W(\omega) = \frac{\sin(N\frac{\omega}{2})}{\sin(\frac{\omega}{2})}. \quad (4.41)$$

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:

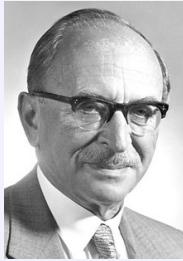
$$W(0) = \lim_{\omega \rightarrow 0} \frac{\sin(N\frac{\omega}{2})}{\sin\frac{\omega}{2}} = \lim_{\omega \rightarrow 0} \frac{\sin'(N\frac{\omega}{2})}{\sin'\frac{\omega}{2}} = \lim_{\omega \rightarrow 0} \frac{N \cos(N\frac{\omega}{2})}{\cos\frac{\omega}{2}} = N \quad (4.42)$$

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:

$$\hat{X}(\omega_0) = \frac{A}{2} (W(2\omega_0) + W(0)) \approx \frac{A}{2} N \quad (4.43)$$

## 4.2 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 (4.44). 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**.

$$X_i(k) = \sum_{n=-N/2}^{N/2-1} w(n)x(n+i \cdot H)e^{-jk\frac{2\pi}{N}n}, \quad i = 0, 1, 2, \dots \quad (4.44)$$

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

$$\text{rect}(n) = 1, \quad -N/2 \leq n \leq N/2 \quad (4.45)$$

$$\text{Rect}(k) = \text{sinc}(k) \quad (4.46)$$

- Хан прозорец

$$\text{hann}(n) = 0,5 + 0,5 \cos\left(2\pi \frac{n}{N}\right), \quad -N/2 \leq n \leq N/2 \quad (4.47)$$

$$\text{Hann}(k) = 0,5 \text{sinc}(k) + 0,25 (\text{sinc}(k-1) + \text{sinc}(k+1)) \quad (4.48)$$

- Хаминг прозорец

$$\text{hamm}(n) = 0,54 + 0,46 \cos\left(2\pi \frac{n}{N}\right), \quad -N/2 \leq n \leq N/2 \quad (4.49)$$

- Блекман прозорец

$$\text{black}(n) = 0,42 - 0,5 \cos\left(2\pi \frac{n}{N}\right) + 0,080,5 \cos\left(4\pi \frac{n}{N}\right), \quad -N/2 \leq n \leq N/2 \quad (4.50)$$

- Блекман-Харис прозорец

$$\text{blackharr}(n) = \frac{1}{N} \sum_{i=0}^3 a_i \cos\left(2i\pi \frac{n}{N}\right), \quad -N/2 \leq n \leq N/2 \quad (4.51)$$

каде

$$a_0 = 0,35876, a_1 = 0,48829, a_2 = 0,14128, a_3 = 0,01168 \quad (4.52)$$

## 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. 4.1. 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.

### 4.3 Вовед во филтри

### 4.4 Основи на дигиталните филтри

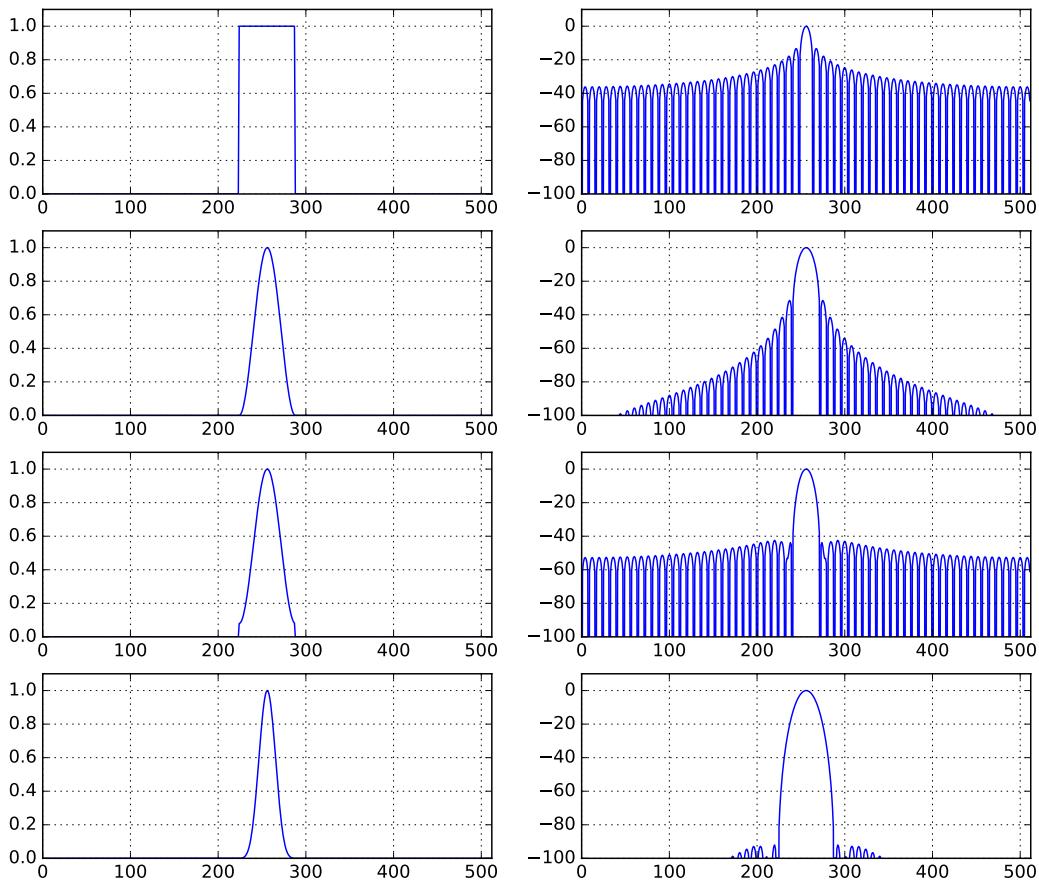
**Филтрите** претставуваат системи чија примарна намена е обликувањето на спектарот на биомедицинските сигнали. Аналогни и дигитални филтри се потребни за ограничување на спектарот во АД конверзијата, а играат клучна улога во отстранувањето на несаканите електромагнетни шумови. Последново е особено критично кај биомедицинските сигнали поради нивната мала амплитуда. Аналогните филтри се реализираат со помош на 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):

$$H\left\{\sum_{m=1}^M a_m x_m(n)\right\} = \sum_{m=1}^M a_m y_m(n), \quad \forall a_m \text{ -- линеарност и} \quad (4.53)$$

$$H\{x(n-k)\} = y(n-k), \quad \forall k \text{ -- инваријантност на поместување.} \quad (4.54)$$

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



**Сл. 4.1:** 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):

$$y(n) = x(n) * h(n) = \sum_{m=-\infty}^{\infty} x(m)h(n-m), \quad (4.55)$$

$$Y(z) = H(z)X(z), \quad (4.56)$$

$$Y(\omega) = H(\omega)X(\omega). \quad (4.57)$$

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

$$h(n) \equiv 0, \quad n < 0. \quad (4.58)$$

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

$$\sum_{n=-\infty}^{\infty} |h(n)| < \infty. \quad (4.59)$$

### 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](#):

$$y(n) = \sum_{i=0}^p b_i x(n-i) - \sum_{i=1}^q a_i y(n-i). \quad (4.60)$$

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 (4.60):

$$y(n) + \sum_{i=1}^q a_i y(n-i) = \sum_{i=0}^p b_i x(n-i), \quad (4.61)$$

$$\left( 1 + \sum_{i=1}^q a_i z^{-i} \right) Y(z) = \sum_{i=0}^p b_i z^{-i} X(z), \quad (4.62)$$

$$H(z) = \frac{Y(z)}{X(z)} = \frac{\sum_{i=0}^p b_i z^{-i}}{1 + \sum_{i=1}^q a_i z^{-i}}. \quad (4.63)$$

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:

$$q = p = N, \quad (4.64)$$

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:

$$H(z) = \frac{A \prod_{i=1}^p (1 - c_i z^{-1})}{\prod_{i=1}^q (1 - d_i z^{-1})}. \quad (4.65)$$

### Типови филтри според амплитудната карактеристика

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

$$H(\omega) = \frac{\sum_{i=0}^p b_i e^{-j i \omega}}{1 + \sum_{i=1}^q a_i e^{-j i \omega}}. \quad (4.66)$$

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

$$H(\omega) = |H(\omega)| e^{j \angle H(\omega)} = A(\omega) e^{j \phi(\omega)}, \quad (4.67)$$

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:

$$\tau(\omega) = -\frac{d\phi(\omega)}{d\omega} = \text{const.} \quad (4.68)$$

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  $h(\frac{N-1}{2})$ :

$$h(n) = h(N - 1 - n), \quad (4.69)$$

или пак кога е антисиметричен во однос на него:

$$h(n) = -h(N - 1 - n). \quad (4.70)$$

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:

$$h(n) = h(0)\delta(n) + h(1)\delta(n - 1) + \cdots + h(\frac{N-1}{2})\delta(n - \frac{N-1}{2}) + \cdots \quad (4.71)$$

$$\cdots + h(1)\delta(n - (N-2)) + h(0)\delta(n - (N-1)), \quad (4.72)$$

па за преносната функција имаме:

$$H(z) = h(\frac{N-1}{2})z^{-\frac{N-1}{2}} + \sum_{i=0}^{\frac{N-3}{2}} h(i) \left( z^{-i} + z^{-(N-1-i)} \right) \quad (4.73)$$

$$= z^{-\frac{N-1}{2}} \left( h(\frac{N-1}{2}) + \sum_{i=0}^{\frac{N-3}{2}} h(i) \left( z^{-i+\frac{N-1}{2}} + z^{i-\frac{N-1}{2}} \right) \right) \quad (4.74)$$

$$= z^{-\frac{N-1}{2}} \sum_{i=0}^{\frac{N-1}{2}} a(n) \frac{z^n + z^{-n}}{2}, \quad (4.75)$$

каде:

$$a(n) = \begin{cases} h(\frac{N-1}{2}), & n = 0 \\ 2h(-n + \frac{N-1}{2}) & n = 1, 2, \dots, \frac{N-3}{2} \end{cases}. \quad (4.76)$$

---

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

Од (4.75) може да се пресмета фреквенциската карактеристика на филтерот:

$$H(\omega) = e^{-j\omega \frac{N-1}{2}} \sum_{n=0}^{\frac{N-1}{2}} a(n) \cos(n\omega). \quad (4.77)$$

#### 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:

$$H(\omega) = e^{-j\omega \frac{N-1}{2}} \sum_{n=1}^{\frac{N-1}{2}} b(n) \cos((n - \frac{1}{2})\omega), \quad (4.78)$$

каде:

$$b(n) = 2 h(-n + \frac{N}{2}), \quad n = 1, 2, \dots, \frac{N}{2}. \quad (4.79)$$

Од (4.78) 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:

$$H(\omega) = e^{-j(\omega \frac{N-1}{2} - \frac{\pi}{2})} \sum_{n=1}^{\frac{N-1}{2}} c(n) \sin(n\omega), \quad (4.80)$$

каде:

$$c(n) = 2 h(-n + \frac{N-1}{2}), \quad n = 1, 2, \dots, \frac{N-1}{2}. \quad (4.81)$$

Од (4.80) 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:

$$H(\omega) = e^{-j\omega \frac{N-1}{2}} \sum_{n=1}^{\frac{N}{2}} d(n) \sin((n - \frac{1}{2})\omega), \quad (4.82)$$

каде:

$$d(n) = 2 h(-n + \frac{N}{2}), \quad n = 1, 2, \dots, \frac{N}{2}. \quad (4.83)$$

Од (4.82) 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.

## 4.5 Процесирање на ЕМГ сигнали

Во овој дел ќе направиме низа основни процесирања на еден ЕМГ сигнал. Како примерок ќе го искористиме сигналот вклучен во Биоспај<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. 4.2

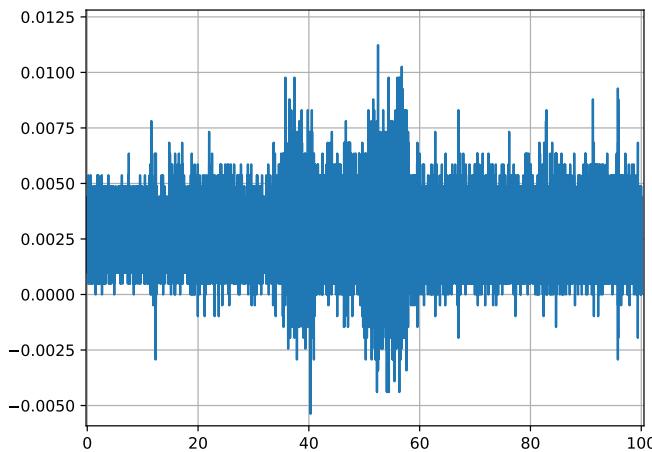
### Фреквенциски спектар на ЕМГ сигналот

За пресметка на спектарот на ЕМГ сигналот ќе се послужиме со дискретната Фуриеова трансформација (ДФТ), воведена во Поглавјето 4, поточно алгоритамот за брза Фуриеова трансформација (БФТ) кој е имплементиран во модулот `fftpack` во Сајпај пакетот.

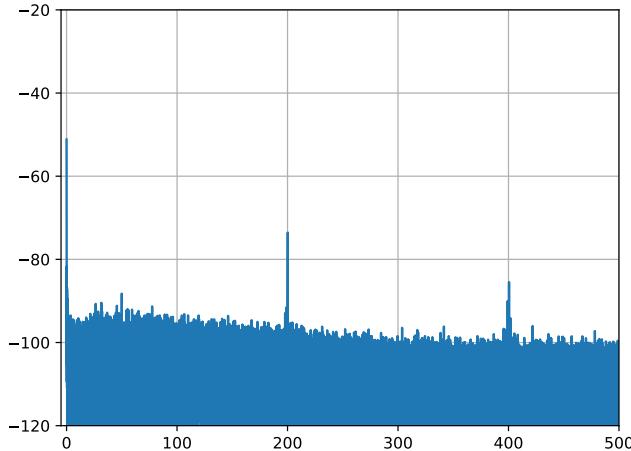
```
from scipy import fftpack as fft
# пресметка на бројот на точки за БФТ
x = np.ceil(np.log2(emg.size))
n_fft = np.int(2**x)

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
```

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



**Сл. 4.2:** Изглед на вчитаниот ЕМГ сигнал.



**Сл. 4.3:** Амплитуден спектар на ЕМГ сигналот.

```
emg_db = 20 * np.log10(emg_amp)

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

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. 4.3. 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 вкупност 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
```

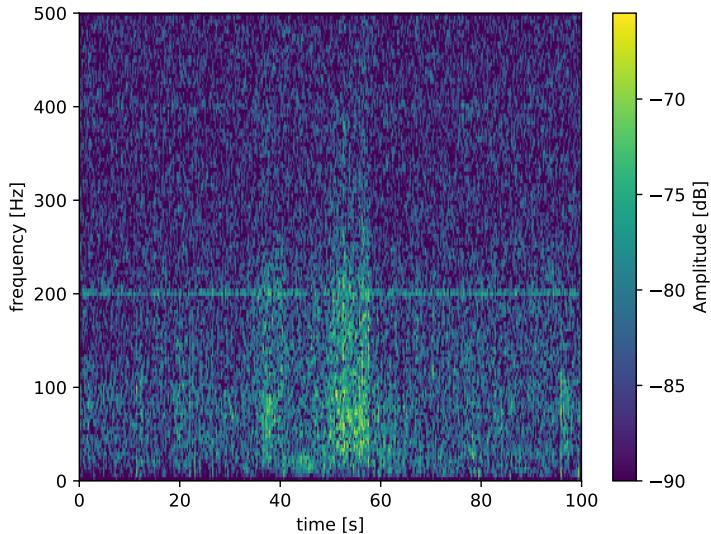
Сега спектарот на кој било сигнал може да го пресметаме повикувајќи ја нашата функција `get_spectrum` на следниот начин:

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

## Спектограм 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 4](#).

```
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:
```



**Сл. 4.4:** Спектрограм на ЕМГ сигналот.

```

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)
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. 4.4. Можеме јасно да видиме дека ЕМГ 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 испртување.

```

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)

```

**§ Дополнително.** За прикажување на амплитудата на спектrogramот 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 спектrogramите во ова поглавје е новата `viridis` која е претставена на конференцијата Сајпај 2015<sup>13</sup>.

```

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]
    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 спектrogramот.
    """
    plt.figure()
    plt.imshow(spectrogram, extent=[0, t[-1], 0, f[-1]], aspect='auto',
               origin='lower', vmin=vmin, vmax=vmax)
    plt.colorbar()

```

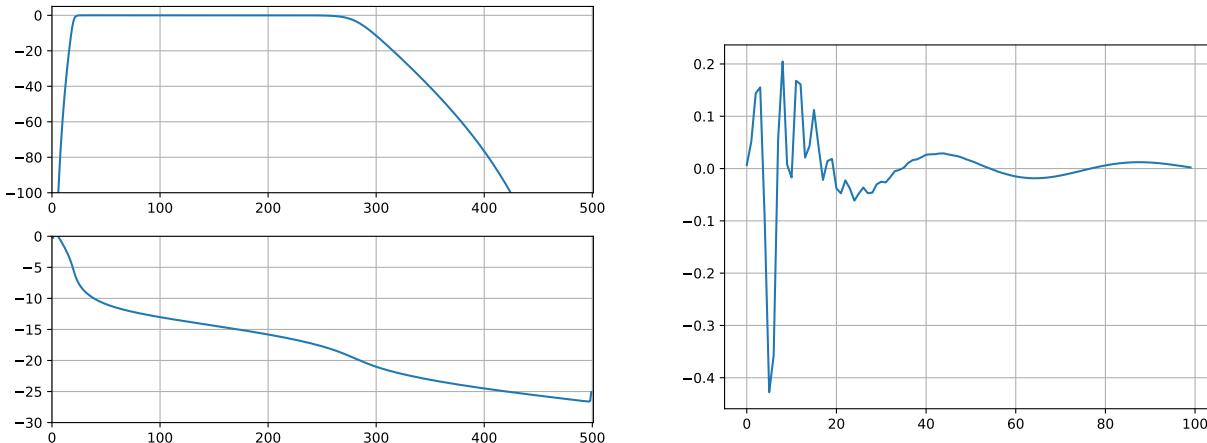
\* Важно! Најважните два параметри во пресметување на спектrogramот се:

1. **Типот на прозорец.** Разликите во спектrogramот кои се добиваат со различните прозорци зависат од нивните спектрални карактеристики прикажани на Сл. 4.1.
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 спектrogramите, 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 4.3. Каде ЕМГ сигналот обликовт 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.

<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=xAoljeRJ3IU>



**Сл. 4.5:** Амплитудна и фазна карактеристика (лево) и дел од импулсниот одсив (десно) на дизајнираниот ИИР филтер пропусник на опсег.

```

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')
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()

```

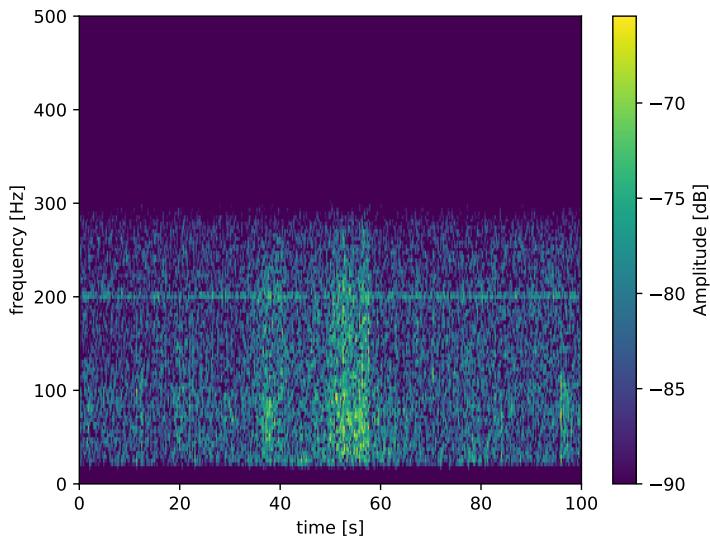
Амплитудната и фазната карактеристика на дизајнираниот филтер, како и дел од неговиот импулсен одсив се дадени на Сл. 4.5. Може да се види дека амплитудната карактеристика е строго монотона, што е и карактеристика на Батерворт ИИР филтрите.

Следно ќе го исфилтрираме ЕМГ сигналот со дизајнираниот ИИР филтер и ќе го прикажеме неговиот спектрограм, даден на Сл. 4.6.

```

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)

```



Сл. 4.6: Спектрограм на филтриралиот ЕМГ сигнал.

### 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 наоѓање на временските моменти кога сигналот поминува зададен праг. Притоа, прагот се пресметува според амплитудната содржина на усреднетиот ЕМГ сигнал. Конечно, детектираниите 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)

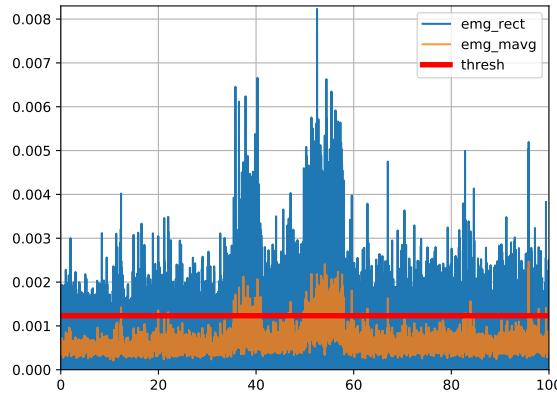
# поместувачки усреднувач
t_win = .05 # 50 ms
n_win = np.int(t_win * fs)
win = sig.get_window('parzen', n_win)
win = win / np.sum(win)

# 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()
```

<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. 4.7:** Добиени сигнали со целобрановото насочување и усреднувањето на ЕМГ сигналот, како и пресметаниот праг за детекција на мускулна активност.

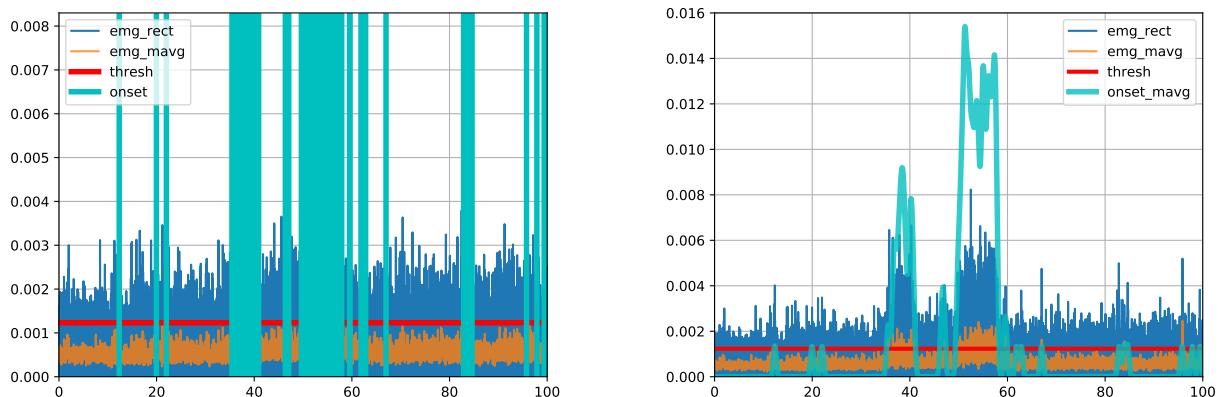
```
# %% детекција на премини преку прагот
over = np.where(emg_mavg > thresh)[0]
under = np.where(emg_mavg <= thresh)[0]
onsets = np.intersect1d(over-1, under)
onsets += 1

plt.figure()
plt.plot(t, emg_rect)
plt.plot(t, emg_mavg, alpha=.8)
plt.plot([t[0], t[-1]], [thresh, thresh], c='r', lw=4)
for onset in onsets:
    plt.plot([t[onset], t[onset+1]], [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. 4.7. На Сл. 4.8 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. 4.8: Детектирани премини на усреднетиот ЕМГ сигнал низ прагот (лево) и нивната усредната верзија (десно).

## 4.6 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 Биоспјај<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.

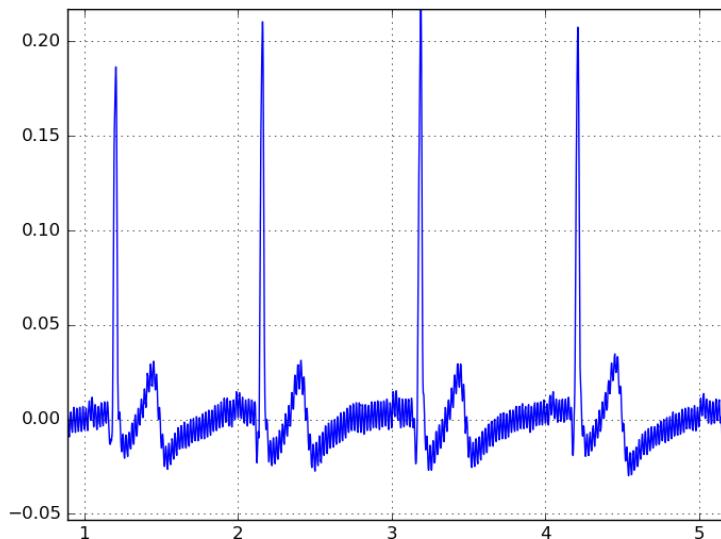
```

1  from __future__ import division
2  import numpy as np
3  import matplotlib.pyplot as plt
4  import scipy.fftpack as ff
5
6  ecg = np.array([])
7  count = 0
8
9  with open('ecg.txt') as fh:
10     for line in fh:
11         if count > 3:
12             val = float(line[:-3])
13             ecg = np.append(ecg, val)
14         count += 1
15
16  ecg = (ecg - 2**11) / 2**11
17  Fs = 1000
18  t = np.arange(0, ecg.size/Fs, 1/Fs)
19  plt.plot(t, ecg)
20  plt.grid()

```

На почетокт од кодот се увезува делењето со автоматско кастување 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 ис纯粹ување со помош на функцијата `plot` од `matplotlib` (17–20). Добиениот график е прикажан на Сл. 4.9

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



Сл. 4.9: Изглед на измерениот ЕКГ сигнал.

### 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 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.

```

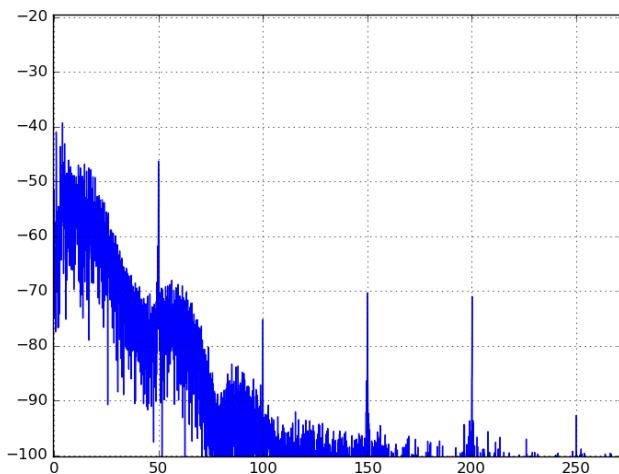
21 N = ecg.size
22 Nfft = 2**np.ceil(np.log(N)/np.log(2))
23
24 ecg_spec = ff.fft(ecg, Nfft)
25 ecg_spec = np.abs(ecg_spec)
26 ecg_spec = ecg_spec / N
27 ecg_spec = ecg_spec[0:Nfft/2+1]
28 ecg_spec[2:-1] = ecg_spec[2:-1] * 2
29 ecg_spec = 20*np.log10(ecg_spec)
30
31 f = np.linspace(0,Fs/2,Nfft/2+1)
32 plt.figure()
33 plt.plot(f, ecg_spec)
34 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. 4.10.

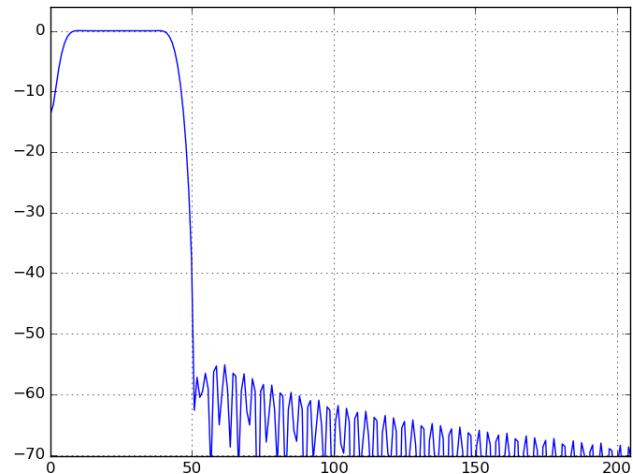
### Филтрирање на ЕКГ сигналот

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

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



**Сл. 4.10:** Амплитуден спектар на измерениот ЕКГ сигнал.



**Сл. 4.11:** Преносна функција на дизајнираниот FIR филтер пропусник на опсег.

модулот `scipy.signal` со функцијата `firwin()`. Карактеристиките на филтерот може да ги визуелизирајме користејќи ја функцијата `freqz()` со која го добиваме графикот на Сл. 4.11.

```

35 order = int(0.3 * Fs) # neparen za tip 1
36 if order % 2 == 0: # ako e paren
37     order = order + 1
38
39 from math import pi
40 import scipy.signal as sg
41 f_l = 3
42 f_h = 45
43 b = sg.firwin(order,[f_l, f_h],pass_zero=False,nyq=Fs/2)
44 w, h = sg.freqz(b)
45 f = w / pi *Fs/2
46
47 plt.figure()
48 plt.plot(f, 20*np.log10(np.abs(h)))
49 plt.grid()

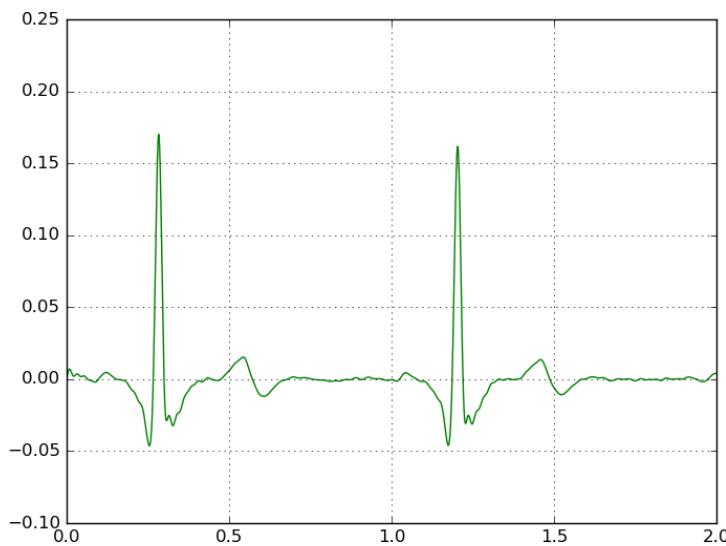
50 ecg_filt = sgfiltfilt(b, 1, ecg)
51 plt.figure()
52 plt.plot(t,ecg_filt,'g')
53 plt.grid()

```

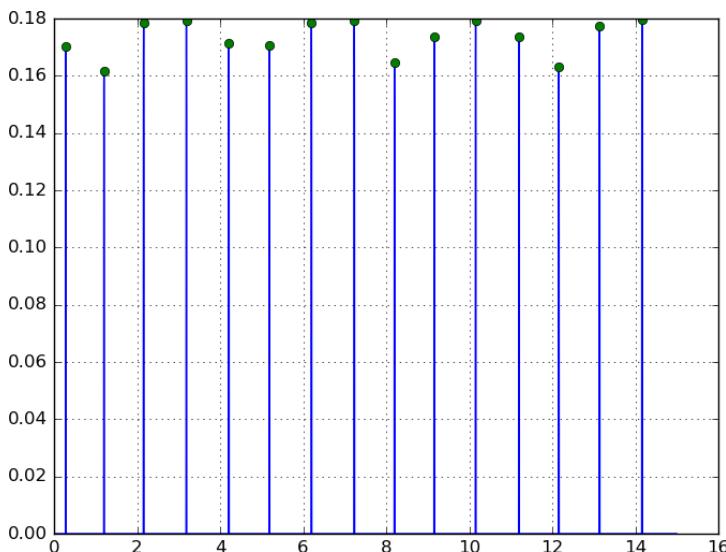
Филтрирањето ќе го извршиме со функцијата `filtfilt()` која го филтрира сигналот во двете насоки со што се постигнува филтриран сигнал кој нема фазно доцнење во однос на оригиналниот сигнал. Исфилтрираниот сигнал може да се види на Сл. 4.13. Може да се види дека елиминиран е проблемот со непостојаноста 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. ??



Сл. 4.12: Исфилтрирана верзија од ЕКГ сигналот.

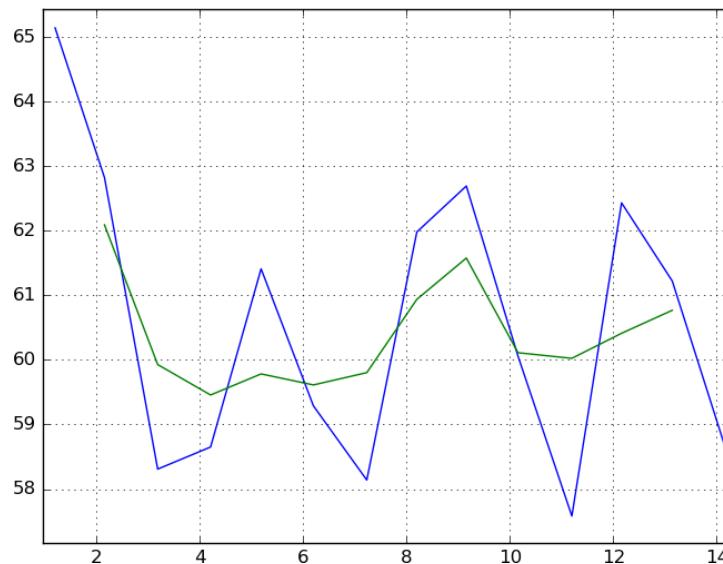


Сл. 4.13: Најдени врвови во ЕКГ сигналот.

```

54     ecg_max = np.max(ecg)
55     threshold = ecg_max * 0.6 # 60% od max
56     ecg_th = np.zeros(ecg.size)
57     ecg_th[ecg_filt > threshold] = ecg_filt[ecg_filt > threshold]
58     R = np.array([], dtype='int')
59     for i in range(1,ecg.size):
60         if ecg_th[i] > ecg_th[i-1] and \
61             ecg_th[i] > ecg_th[i+1]:
62             R = np.append(R,i)
63
64     plt.figure()
65     plt.plot(t,ecg_th)
66     plt.plot(t[R],ecg_th[R], 'o')
67     plt.grid()

```



Сл. 4.14: Број на отчукувања на срцето во минута пресметани од ЕКГ сигналот.

### Пресметка на бројот на отчукувања во минута

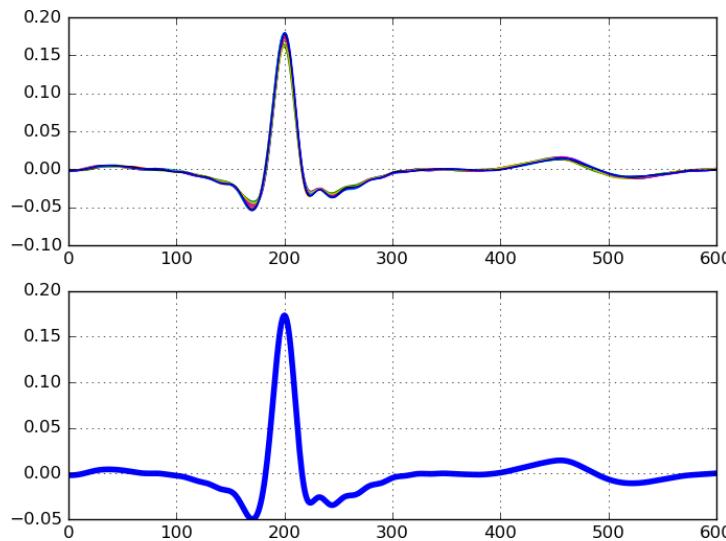
Откако ги имаме детектирано R врвовите во ЕКГ сигналот, од нив може да се пресмета моментната брзина на отчукување на срцето која вообичаено се изразува во bpm (beats per minute). За пресметување на измазната верзија од оваа брзина можеме да искористиме лизгачи прозорец за усреднување. Пресметаните моментни и усреднети вредности за bpm се прикажани на Сл. 4.14.

```

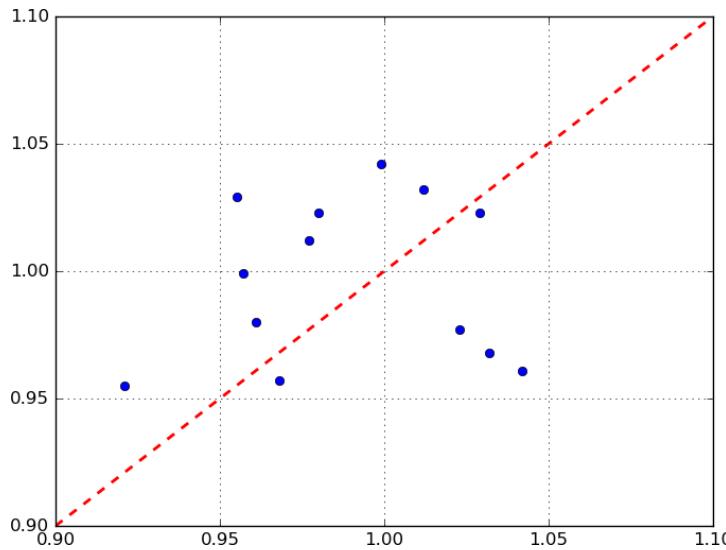
68 R_diff = np.diff(R)
69 R_diff = R_diff/Fs # вo s
70 bpm = 60 / R_diff
71
72 # smoothing - running average
73 smooth_ord = 3
74 win = np.ones(smooth_ord)
75 win = win / np.sum(win)
76 bpm_smooth = signal.convolve(bpm, win, mode='same')
77 bpm_smooth = bpm_smooth[1:-1]
78
79 plt.figure()
80 plt.plot(t[R[1:]], bpm)
81 plt.plot(t[R[2:-1]], bpm_smooth)
82 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. 4.15



Сл. 4.15: Усредната репрезентација на отчукувањата на срцето во ЕКГ сигналот.



Сл. 4.16: RR дијаграм на отчукувањата на срцето во ЕКГ сигналот.

```

83 R_left = 0.200 * Fs
84 R_right = 0.400 * Fs
85 templates = np.zeros(R_left+R_right)
86 for r in R:
87     ecg_temp = ecg_filt[r-R_left : r+R_right]
88     templates = np.vstack((templates, ecg_temp))

89
90 ecg_template = np.mean(templates[1:],0)
91 plt.figure()
92 plt.subplot(211)
93 plt.plot(templates[1:].T)
94 plt.grid()
95 plt.subplot(212)
96 plt.plot(ecg_template, linewidth=4)
97 plt.grid()

```

## 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. 4.16.

```
98  RRs = np.zeros(2)
99  for i in range(1,len(R_diff)):
100     RR = np.array([R_diff[i-1], R_diff[i]])
101     RRs = np.vstack((RRs, RR))
102
103 RRs = RRs[1:,:]
104
105 plt.figure()
106 plt.plot(RRs[:,0], RRs[:,1], 'o')
107 plt.plot([0.9, 1.1], [0.9,1.1], 'r--', linewidth=2)
108 plt.grid()
```

## Поглавје 5

# Примена на машинското учење на биомедицинските 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.

### 5.1 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 тима 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 (ПЦА)<sup>6</sup> е еден пример за егзактна математичка метода за оваа намена, која е и реверзибилна. Тука спаѓаат стохастични алгоритми базирани на ПЦА, 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 скоро 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 придвижат 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 одвивањето на 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 (t-SNE)*.

<sup>8</sup>Google DeepMind: Ground-breaking AlphaGo masters the game of Go <https://www.youtube.com/watch?v=SUBqykXVxOA>

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

## 5.2 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 тренинг,
- **претпроцесирање и екстракција на карактеристики** – 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 најзначајните карактеристики<sup>10</sup>. За ова може да се применат методи како ПЦА, пресметување на спектар, спектрограми, итн.
- **тренирање на моделот** – оптимизација на параметрите на моделот со употреба на множеството за тренирање и валидација, види [Поглавје 5.3](#),
- **конечна евалуација на моделот** – ова се прави со одвоеното множество за тестирање со цел да се оцени како работи истренираниот модел за множество од невидени влезни податоци.

## 5.3 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 добиеме 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 грешката на моделот не се подобрува за одредено ниво на толеранција последователно во избран број на епохи или 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 тренинг 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 невидени 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:

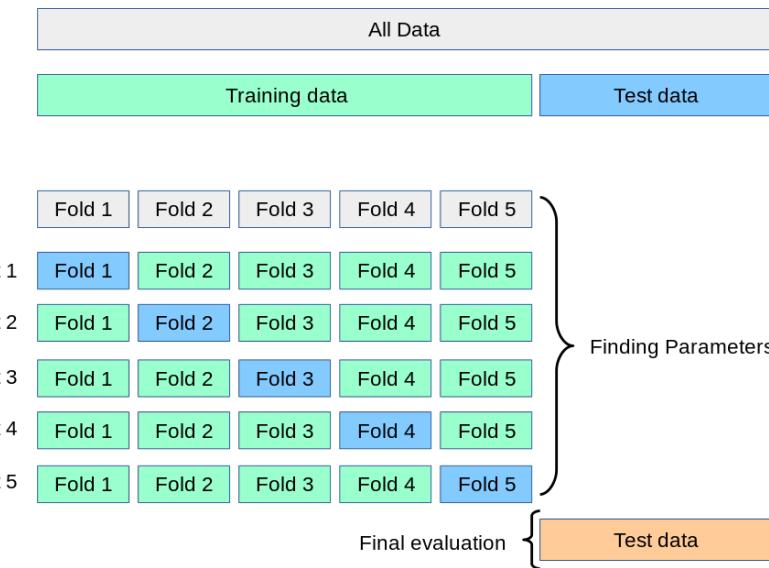
- **ограничување на мокта на моделот** – поедноставни модели не можат апсолутно да описват покомплексни влезни податоци и сигнали,

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

<sup>11</sup>Анг. *learning rate*.

<sup>12</sup>Анг. *underfitting*.

<sup>13</sup>Анг. *overfitting*.



Сл. 5.1:  $K$ -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 на следниот начин, се дели влезното множество на податоци на множества за тренирање и тестирање. Множеството на тренирање потоа 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. 5.1. 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 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 **случајно мешање**<sup>19</sup> на датасетот пред неговото тренирање,
- ако датасетот има групи, на пример субјекти, тогаш може да се случи истиот субјект да биде

<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)

<sup>19</sup>Анг. *shuffle*.

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 пред неговото тренирање,

- 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 оценката; 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 претходните 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. 5.2.

По завршување на меѓувалидацијата, моделот со избрани 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 недворешната јамка, а остатокот се дели на  $M$  подмножества во стандардна јамка за меѓувалидација.

## 5.4 Невронски мрежи

Еден од најпознатите модели за машинско учење, којшто лежи во основата на длабокот учење се вештачките *невронски мрежи (НМ)*. Основната градбена единка на невронските мрежи се софтверските неврони чиј начин на работа е инспириран од физиологијата на невроните описана во Поглавјето 2.3.

### 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.

$$y = f(a) = f\left(\sum_{k=0}^{K-1} w_k x_k + b\right) = f(\mathbf{w}\mathbf{x}^T + b) \quad (5.1)$$

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  $\mathbf{x}_n$ , (5.1) преминува во:

$$y_n = f(a_n) = f(\mathbf{w}\mathbf{x}_n^T + b) \quad \text{за } n = 0, 1, 2, \dots, N-1, \quad (5.2)$$

$$\mathbf{y} = f(\mathbf{a}) = f(\mathbf{w}\mathbf{X}^T + b), \quad (5.3)$$

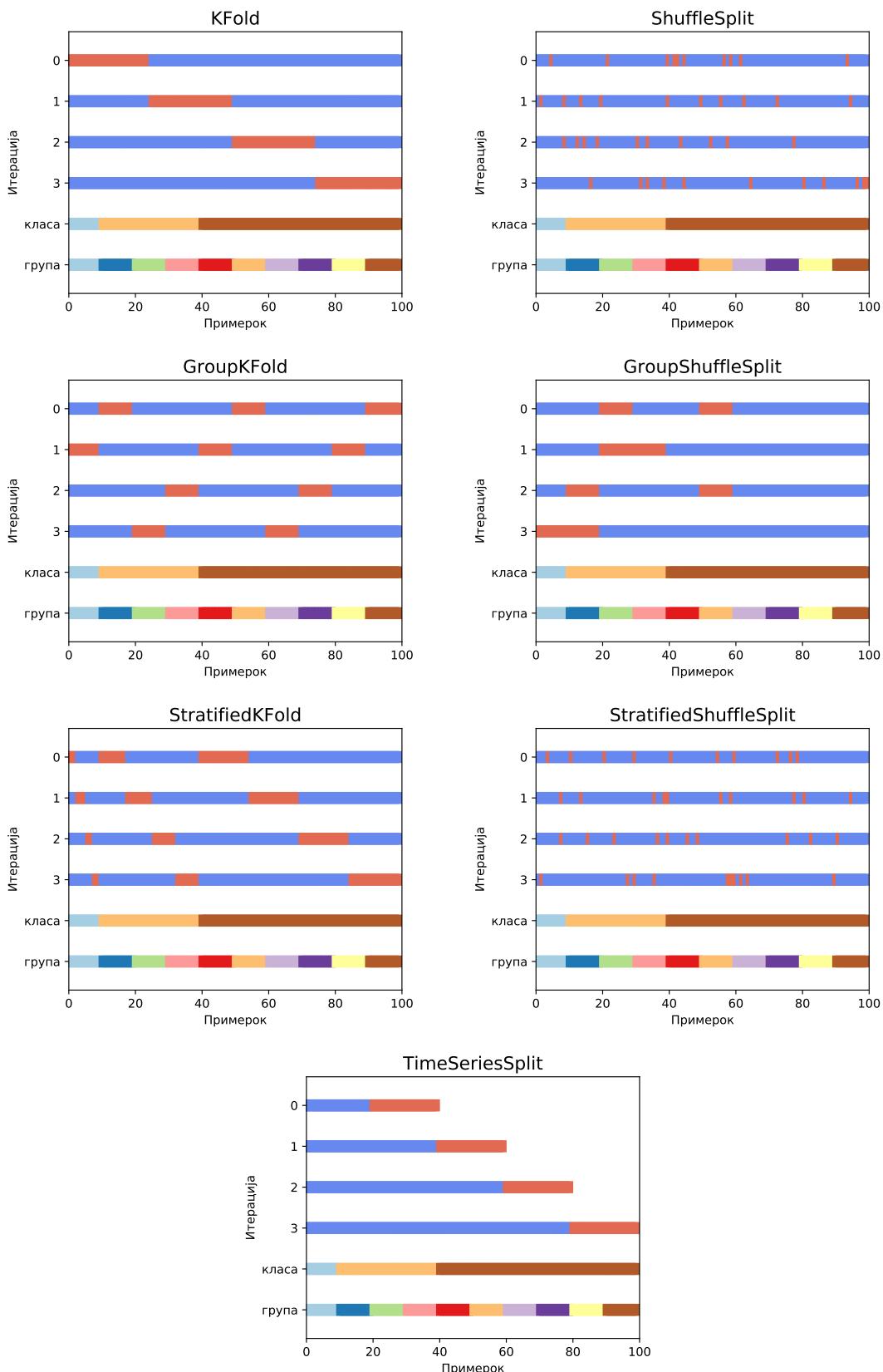
<sup>20</sup>Анг. *group k-fold cross-validation*.

<sup>21</sup>Анг. *stratified k-fold cross-validation*.

<sup>22</sup>Генерирано со адаптиран 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*.

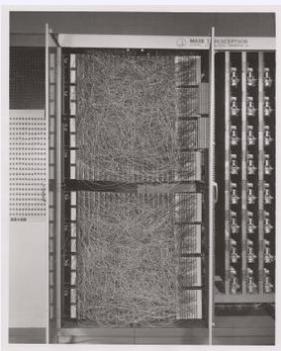
<sup>24</sup>Анг. *bias*.



**Сл. 5.2:** 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>

каде со  $\mathbf{X}$  е означена матрицата по чии редици се поставени влезните примероци  $\mathbf{x}_n$ .

<sup>22</sup> Wikipedia – Perceptron <https://en.wikipedia.org/wiki/Perceptron>



**§ 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

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. 5.3](#). 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 **густи<sup>26</sup>**. Ова едноставна архитектура сепак им овозможува на плитките невронски мрежи да моделаираат 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:

$$\mathbf{y}_h = f_h(\mathbf{a}_h) = f_h(\mathbf{W}_h \mathbf{x}^T + \mathbf{b}_h), \quad (5.4)$$

$$\mathbf{y} = f_o(\mathbf{a}_o) = f_o(\mathbf{W}_o \mathbf{y}_h + \mathbf{b}_o) = f_o(\mathbf{W}_o \cdot f_h(\mathbf{W}_h \mathbf{x}^T + \mathbf{b}_h) + \mathbf{b}_o), \quad (5.5)$$

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:

$$\mathbf{Y}_h = f_h(\mathbf{A}_h) = f_h(\mathbf{W}_h \mathbf{X}^T + \mathbf{b}_h), \quad (5.6)$$

$$\mathbf{Y} = f_o(\mathbf{A}_o) = f_o(\mathbf{W}_o \mathbf{Y}_h + \mathbf{b}_o) = f_o(\mathbf{W}_o \cdot f_h(\mathbf{W}_h \mathbf{X}^T + \mathbf{b}_h) + \mathbf{b}_o), \quad (5.7)$$

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:

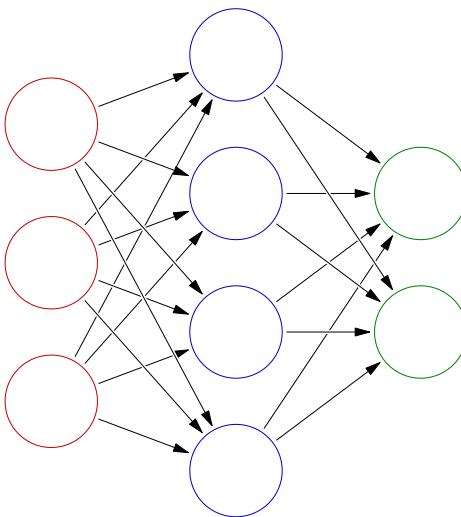
Од анонимен извор - [http://www.peoples.ru/science/psychology/frank\\_rosenblatt/](http://www.peoples.ru/science/psychology/frank_rosenblatt/), CC BY-SA 4.0, <https://commons.wikimedia.org/w/index.php?curid=64998425>

Од Source (WP:NFCC#4), Fair use, <https://en.wikipedia.org/w/index.php?curid=47541432>

<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. 5.3:** 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>

- **сигмоида**  $\sigma(a) = \frac{1}{1+e^{-a}}$  – дава излез во опсег  $0 - 1$ ,
- **тангент 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 полубрановскиот 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,
- **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 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 мокта 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

<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).

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 тренинг.

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 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:

$$x^{i+1} = x^i - \frac{df}{dx} \cdot \Delta x, \quad (5.8)$$

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:

$$\theta_l^{i+1} = \theta_l^i - \frac{\partial \mathcal{L}}{\partial \theta_l} \cdot \eta \quad \text{за } \theta_l \in \theta, \quad (5.9)$$

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:

$$\frac{\partial \mathcal{L}}{\partial \theta_l} = \frac{\partial \mathcal{L}}{\partial \tilde{y}} \cdot \frac{\partial \tilde{y}}{\partial y_{L-2}} \cdots \frac{\partial y_{l+1}}{\partial \theta_{l+1}} \cdot \frac{\partial y_l}{\partial \theta_l}, \quad (5.10)$$

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:

$$\frac{\partial \mathcal{L}}{\partial w_0} = \frac{\partial \mathcal{L}}{\partial \tilde{y}} \cdot \frac{\partial \tilde{y}}{\partial w_0}. \quad (5.11)$$

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

<sup>31</sup>Анг. *multi layer perceptron*.

<sup>32</sup>Анг. *gradient descent (GD)*.

<sup>33</sup>Анг. *chain-rule*.

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 остатар праг, како што било направено во оригиналниот перцентрон, кај кого важело:

$$y = f(a) = \begin{cases} 1 & \text{ако } a > 0,5 \\ 0 & \text{поинаку} \end{cases}. \quad (5.12)$$

Денес најчесто се употребуваат следните функции за грешка:

- [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  $\boldsymbol{\theta}$ . На пример, при употреба на средната квадратна грешка, би имале:

$$L(y, \tilde{y}, \boldsymbol{\theta}) = L(y, \tilde{y}) + \lambda \sum_{l=0}^{L-1} (\mathbf{W}_l^T \mathbf{W}_l + \mathbf{b}_l^T \mathbf{b}_l), \quad (5.13)$$

каде  $\lambda$  е [кофициентот 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 тренинг множество 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 проголем оптимизацијата може да го натфрли минимумот на

<sup>34</sup>Анг. *backpropagation*.

<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)*.

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 каде 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>39</sup>Анг. *learning rate scheduling*.

## 5.5 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 ?.

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 Pandas<sup>44</sup>. Притоа, Pandas 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. 5.4. Може да видиме дека сигналите од акселерометарот за  $y$  оската, која е всушност вертикалната оска на човекот, има најголеми вредности за забрзувањето.

<sup>40</sup>Анг. *human activity recognition (HAR)*.

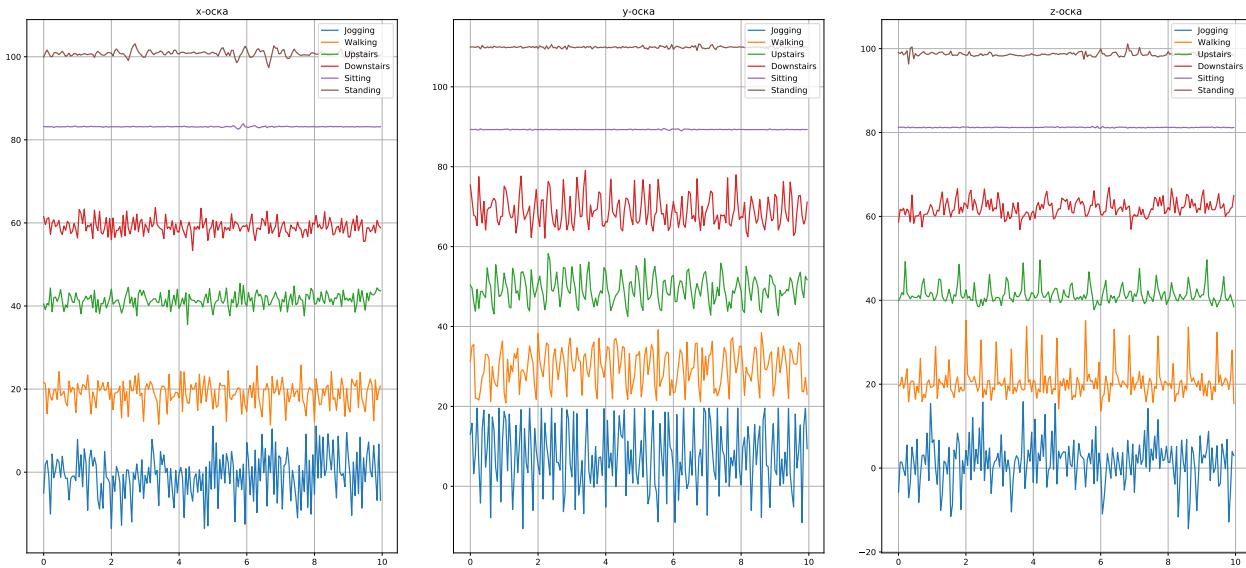
<sup>41</sup>Wireless Sensor Data Mining (WISDM) Lab, Fordham University, New York, USA <http://www.cis.fordham.edu/wisdm/>

<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. 5.4:** Временски облик на сигналите добиени од акселерометарот за 10 s на вчитаниот ЕМГ сигнал долж трите оски.

### 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 состјбата на генераторот со наредбата `np.random.seed`, при секое извршување на кодот ќе ги добиеме истите резултати. Во конкретниот случај тоа ќе бида 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 податоци како средна вредност, средна апсолутна девијација, стандардна девијација, хистограм, и растојание помеѓу врвови. По 15 статистички податоци се пресметани за секоја оска и

<sup>46</sup>Анг. *pseudo-random number generator (PRNG)*.

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 спектограми.

```
import bme

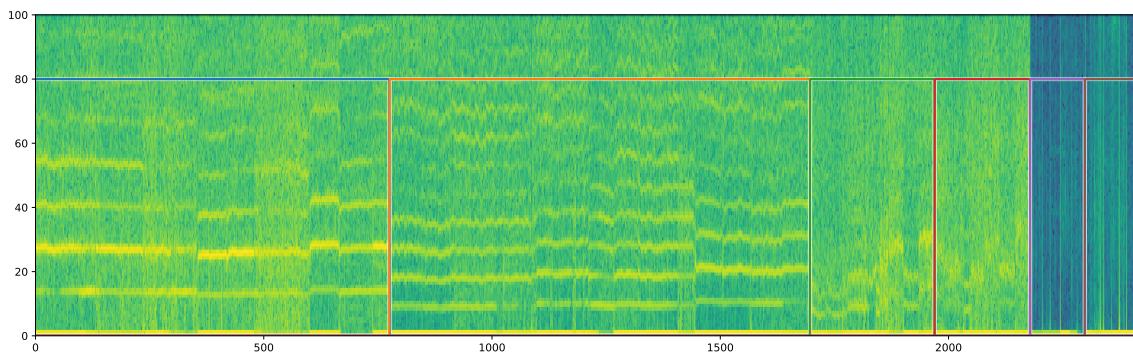
x_train = None
y_train = None
x_test = None
y_test = None
for i_act, act in enumerate(activities):
    mask_act = data.act == act

    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)
```



**Сл. 5.5:** 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.

```
y_test = np.concatenate((y_test, y), axis=0)
```

При генерирање на карактеристиките може да видиме дека не секој од испитаниците ги има 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 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 валидациски 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
```

<sup>47</sup>sciKit-Learn <http://scikit-learn.org/stable/>

```

Iteration 6, loss = 1.30116524
Iteration 7, loss = 1.25296934
Iteration 8, loss = 1.18017693
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 на меѓу-ентропија од 0,2961.

## Евалуација на моделот

За да видиме која е точноста на истренираната невронска мрежа ќе го искористиме следниот код.

```

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 грешки невронската мрежа може да ја пресметаме матрицата на грешка.<sup>48</sup> За да ја пресметаме, ќе направиме предвидување на класите со тест множеството.

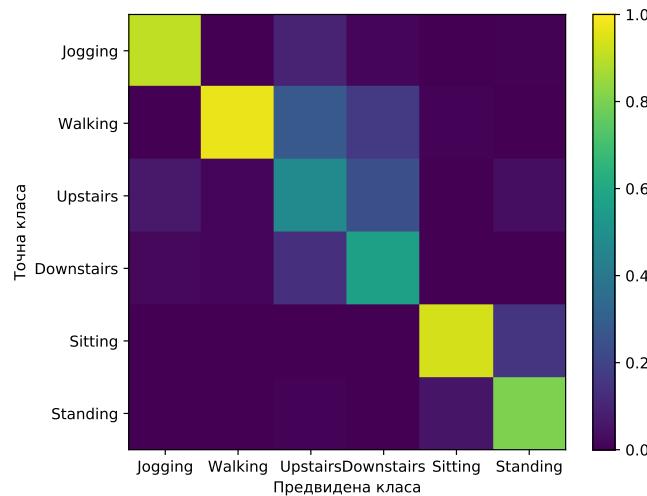
```

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)

```

---

<sup>48</sup>Анг. *confusion matrix*.



Сл. 5.6: Матрица на грешка за истренираната невронска мрежа.

```
[[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.

```
cm = cm / cm.sum(axis=0)
print(cm)
```

```
[[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]]
```

За појасен приказ ќе ја прикажеме матрицата на грешка преку топлинска мапа<sup>49</sup> на Сл. 5.6

```
fig, ax = plt.subplots()
im = ax.imshow(cm, aspect='auto', interpolation='nearest', vmax=1, vmin=0)
ax.xlabel = 'Prediction'
ax.ylabel = 'Ground truth'
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)
```

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

<sup>49</sup>Анг. *heat map*.

временски сигнали на Сл. 5.4 и нивните спектри на Сл. 5.5. 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 нумеришка 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=Ag1AKI1\\_2GM](https://www.youtube.com/watch?v=Ag1AKI1_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\\_1LNtd0](https://www.youtube.com/watch?v=xGhj_1LNtd0)

<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</sup><sup>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>Bo 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.eteknix.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 рас пространетост на слободниот софтвер денес. Така, ГНУ/Линукс и ФриБСД<sup>15</sup> оперативните системи се во употреба во 98,27 % од серверите на интернет (споредено со 1,73 % со Виндоус), 79,3 % од паметните телефони (Андроид оперативниот систем)<sup>16</sup>, и 99 % од суперкомпјутерите<sup>17</sup>. Сепак, неговиот пробив во персоналните компјутери засега е незначителен – 2,1 % (наспроти 87 % на Виндоус и 9,7 % 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 развиени 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 развиен 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 USD,
- 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> FreeBSD <https://www.freebsd.org/>

<sup>16</sup> Оваа бројка е речиси 91 % ако се има в' предвид дека и iOS е базиран на Линукс јадрото.

<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 СпајдерSpyder – The Scientific Python Development Environment <https://manjaro.org/> 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>29</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>30</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 Симулинк пакетот на Матлаб, наречена Икскос<sup>31</sup>.
- Питон<sup>32</sup> е широко распространет, повеќенаменски, интерпретиран и динамичен програмски јазик на високо ниво направен од Гуидо ван Росум<sup>33</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 ширум светот.
- Џулиа<sup>34</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>GNU Octave – Scientific Programming Language <https://www.gnu.org/software/octave/>

<sup>30</sup>Scilab – Open source software for numerical computation <https://www.scilab.org/>

<sup>31</sup>Xcos <https://www.scilab.org/software/xcos>

<sup>32</sup>Python <https://www.python.org/>

<sup>33</sup>Wikipedia – Guido van Rossum [https://en.wikipedia.org/wiki/Guido\\_van\\_Rossum](https://en.wikipedia.org/wiki/Guido_van_Rossum)

<sup>34</sup>The Julia Programming Language <https://julialang.org/>

## Додаток Б

# Питон за процесирање на биомедицинските сигнали

За процесирањето на дигиталните биомедицински сигнали ќе биде искористен програмскиот јазик Питон и тоа неговата нова верзија [3](#), заедно со библиотеките:

- Нумпай – 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 Питон кои се користат во научните истражувања како на пример [Пандас](#)<sup>[4](#)</sup> за статистички анализи, [Симпај](#)<sup>[5](#)</sup> за симболичка математика, [Сајкит-лerner](#)<sup>[6](#)</sup> за машинско учење итн.

Како интерфејс кон Питон ќе ја користиме интерактивната конзола [ИПитон](#)<sup>[7](#)</sup> и научната развојна средина за Питон [Спајдер](#)<sup>[8](#)</sup>.

### Б.1 Основи поставки во ГНУ/Линукс

Иако користењето на Питон не е врзано со ГНУ/Линукс оперативниот систем, вежбите во овој предмет ќе се базираат на работа под ГНУ/Линукс. Доколку веќе немате ГНУ/Линукс, истиот се препорачува да го инсталirate паралелно на постоечкиот оперативен систем. Во најмала рака може да инсталirate ГНУ/Линукс во виртуелна машина, но ова може да ги ограничи постоечките ресурси за процесирање на сигналите.<sup>[9](#)</sup> Во Лабораторијата за дигитално процесирање на сигнали ќе работиме со [Убунту Мате](#)<sup>[10](#)</sup> кој е базиран на [Убунту](#), а ја користи десктоп средината [Мате](#) чиј изглед е базиран на [Гном 2](#), а е имплементиран во [Гном 3](#).<sup>[11](#)</sup>

---

<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>Добар преглед на популарноста на различните Линукс дистрибуции, како и повеќе информации за истите може да најдете на вебстраницата [Distrowatch](#). <http://distrowatch.com/>

<sup>10</sup>Ubuntu MATE. <https://ubuntu-mate.org/>

<sup>11</sup>Кај ГНУ/Линукс оперативните системи можат да се користат различни десктоп средини како GNOME и MATE, но уште и KDE Plasma, Xfce, LXQt, Cinnamon, Pantheon, итн.

За работа со ГНУ/Линукс можеме да го искористиме стандардниот БАШ терминал.<sup>12</sup> Вообичаена кратенка за отворање на нов терминал е `ctrl-alt-t`, или ако дистрибуцијата доаѓа со терминал на спуштање копчето `F12`.

За почеток треба во вашиот основен фолдер<sup>13</sup> да отворите нова папка со името на предметот. Тоа може да го направите преку фајл експлорерот, или преку терминалот:

```
~ $ mkdir bme
```

Следно, од Гитхаб страната на предметот Биомедицинска електроника<sup>14</sup> превземете го фолдерот со скриптата, кодот и примероци од биомедицински сигнали кои ќе ги користиме во вежбите. Тоа можете да го направите на следниот начин:

```
~ $ cd bme
~/bme $ git clone https://github.com/FEEIT-FreeCourseWare/Biomedical_Electronics
```

## Б.2 Основи на работата со Питон

### Питон интерпретер

За работа со Питон може да ја искористиме стандардната инсталацијата на Питон која доаѓа со секоја ГНУ/Линукс дистрибуција. Питон интерпретерот можеме да го повикаме во стандардниот БАШ терминал со:

```
$ python
```

```
Python 3.7.2 (default, Jan 10 2019, 23:51:51)
[GCC 8.2.1 20181127] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> print('hello world')
hello world
```

За излегување од Питон конзолата треба да ја притиснеме стандардната кратенка `ctrl-d` или да напишеме `exit()`.

### ИПитон

Поради ограничениите можности на основниот Питон интерпретер, вообичаено со Питон се работи во интерактивната конзола **ИПитон** која нуди низа на подобрувања. Нејзие може да ја инсталлираме 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.
```

<sup>12</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/>

Постојат и понапредни шел средини како на пример Z shell – скратено Zsh. Еве еден туторијал за нејзина инсталација: 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>Бо ГНУ/Линукс основен фолдер на секој корисник е `/home/user_name/`, а кратенка за него е `~`.

<sup>14</sup>[https://github.com/FEEIT-FreeCourseWare/Biomedical\\_Electronics](https://github.com/FEEIT-FreeCourseWare/Biomedical_Electronics)

```
In [1]: print('hello world')
hello world
```

Некои од главните придобивки кои ги носи ИПитон се:

- пристап до стандардната помош во Питон, како на пример докстрингови на објекти и напатствието за Питон, 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 локалниот фолдер, 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 ИПитон напишете `?`  или `%quickref` во интерактивната конзола.

Дополнително, ИПитон е основата зад [Јупјтер Кјутконзолата](#)<sup>15</sup> прикажана на Сл. Б.1, 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
```

## Виртуелни средини за Питон

Инсталирањето на Питон пакети, како што направивме со ИПитон козолата, директно во оперативниот систем не е препорачливо. За избегнување на судир помеѓу системската инсталација на Питон на ГНУ/Линукс оперативниот систем, како и за изолирање на екосистемот од инсталирани модули на секој засебен проект, правилно е да направиме Питон виртуелна средина за процесирање на дигиталните биомедицински сигнали. Постојат низа пакети за создавање и раководење со виртуелните средини во Питон, од кои `pipenv`<sup>16</sup> е оној кој е препорачан од Телото за пакување на Питон<sup>17</sup>.

За инсталирање на `pipenv` треба да го инсталерираме користејќи го системскиот `pip`<sup>18</sup>. Поради можноста на употреба на Питон 2 како стандарден во инсталираната ГНУ/Линукс дистрибуција, најдобро е да напишеме:

```
$ sudo pip3 install pipenv
```

А доколку системот го нема `pip3`, истиот може да се инсталира како:

<sup>15</sup> Jupyter QtConsole <https://github.com/jupyter/qtconsole>

<sup>16</sup> Pipenv – Python Development Workflow for Humans <https://github.com/pypa/pipenv>

<sup>17</sup> Python Packaging Authority PyPA <https://www.pypa.io/en/latest/>

<sup>18</sup> Python Install Package (pip) <https://pypi.org/project/pip/>

```

File Edit View Kernel Window Help
Jupyter QtConsole 4.1.1
Python 2.7.10 (default, Oct 14 2015, 16:09:02)
Type "copyright", "credits" or "license" for more information.

IPython 4.0.3 -- An enhanced Interactive Python.
?           -> Introduction and overview of IPython's features.
%quickref -> Quick reference.
help        -> Python's own help system.
object?    -> Details about 'object', use 'object??' for extra details.
%uiref     -> A brief reference about the graphical user interface.

In [1]: from __future__ import division
...: import numpy as np
...: from scipy.io import wavfile
...: from matplotlib import pyplot as plt
...:

In [2]: %matplotlib inline

In [3]: plt.plot(np.arange(10))
Out[3]: []

```

In [3]:

**Сл. Б.1:** Јупајтер Кјут конзолата нуди напредна интерактивност.

```
$ sudo apt install python3-pip
```

За креирање на виртуелна средина во фолдерот за овој предмет ќе напишеме:

```
$ mkdir bme
$ pipenv --python 3
```

```

Creating a virtualenv for this project...
Pipfile: /tmp/bme/Pipfile
Using /usr/bin/python3 (3.7.2) to create virtualenv...
Creating virtual environment...Using base prefix '/usr'
New python executable in ~/.local/share/virtualenvs/bme-aWLbLjVf/bin/python3
Also creating executable in ~/.local/share/virtualenvs/bme-aWLbLjVf/bin/python
Installing setuptools, pip, wheel...
done.
Running virtualenv with interpreter /usr/bin/python3

Successfully created virtual environment!
Virtualenv location: ~/.local/share/virtualenvs/bme-aWLbLjVf
Creating a Pipfile for this project...

```

Со ова `pipenv` креира нова виртуелна средина во локален фолдер во нашиот кориснички фолдер, во неа го копира системскиот Питон 3 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 Питон интерпретерот или да инсталлираме пакет со `pip`, тоа ќе се случува внатре во виртуелната

средина:

```
$ which pip
/usr/bin/pip

$ echo $PATH
/usr/local/sbin:/usr/local/bin:/usr/bin ...

$ pipenv shell
Launching subshell in virtual environment...
. ~/local/share/virtualenvs/bme-aWLbLjVf/bin/activate

(bme) $ echo $PATH
~/local/share/virtualenvs/bme-aWLbLjVf/bin:/usr/local/sbin:/usr/local/bin:...

(bme) $ which pip
~/local/share/virtualenvs/bme-aWLbLjVf/bin/pip
```

Гледаме дека `pipenv` ни дозначува дека сме во виртуелната средина со `(bme)` пред БАШ промптот. За да излеземе од неа повторно може да ја употребиме кратенката `ctrl-d` или да напишеме `exit`.

## Спајдер

Наместо да работиме со ИПитон конзолата во терминалот, во предметот биомедицинска електроника ќе ја употребиме развојната средина за Питон специјализирана за инженерска и научна работа **Спајдер**<sup>19</sup> прикажана на Сл. Б.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.
- **Интерактивна конзола** – интегрирани Питон и иПитон 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 фајлови и фолдери.**
- **Историја на наредби.**

Спајдер можеме да го инсталлираме во виртуелната средина која ја креирајме во работниот фолдер `bme`:

```
(bme) $ pip install spyder
```

## Б.3 Основи на Нумпај и Матплотлиб

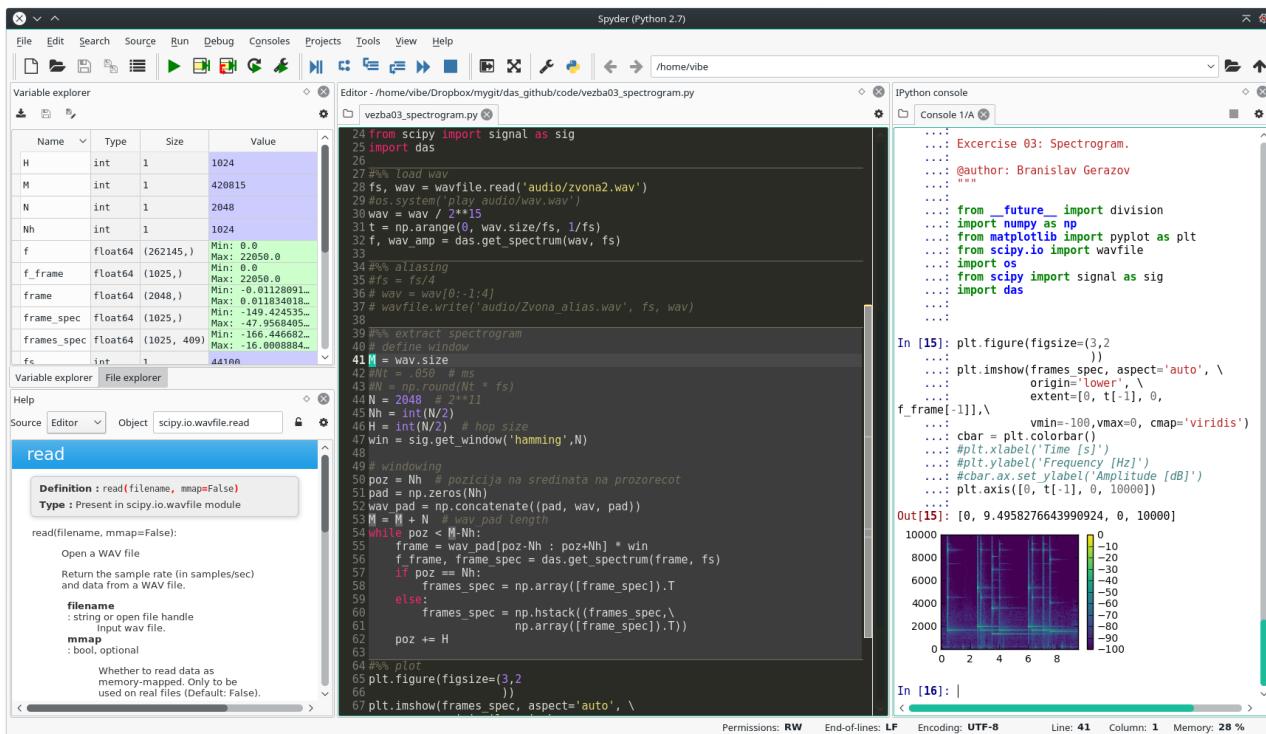
За да ги инсталлираме потребните модули во новата виртуелна средина ќе напишеме:

```
(bme) $ pip install numpy matplotlib scipy
```

Сега може да ги повикаме во ИПитон:

---

<sup>19</sup>Spyder - The Scientific PYthon Development EnviRonment. <https://github.com/spyder-ide/spyder>



**Сл. Б.2:** Спајдер развојната средина за Питон специјализирана за инженерска и научна работа.

```
In [1]: import numpy as np
In [2]: from matplotlib import pyplot as plt
In [3]: x = np.linspace(0, 2*np.pi, 100)
In [4]: y = np.sin(x)
In [5]: %matplotlib
Using matplotlib backend: Qt5Agg

In [6]: plt.plot(x, y)
Out[6]: [<matplotlib.lines.Line2D at 0x7fcbb448ffd0>]
```

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>20</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 Питон.<sup>21</sup> Со наредбата `%matplotlib` се овозможува интерактивното плотирање во ИПитон конзолата.

Добар вовед во програмскиот јазик Питон во рамки на екосистемот за научна работа е даден во [Скриптата за Сајпaj \(Varoquaux et al., 2015\)](#)<sup>22</sup> која е достапна под слободна лиценца<sup>23</sup>. Оваа книга претставува отворен проект и во неа, благодарејќи на многуте автори и придонесувачи, се поместени основите за работа не само со Питон, Нумпай, Матплотлиб и Сајпaj, туку и Пандас, Симпaj, Сајкитимиц, Сајкитлрн, па дури и Цајтон<sup>24</sup>.

За запознавање со основите на програмскиот јазик Питон можат да послужат мноштво на материјали, каков што е [официјалниот туторијал за Питон](#) на вебстраницата на Питон<sup>25</sup> и

<sup>20</sup>Добрата структурираност на просторите на имиња (namespaces) е една од важните одлики на Питон како што е и наведено во стиховите на [Зенот на Питон](#) кој може да го прочитате ако напишете `import this`.

<sup>21</sup>Автоматското надополнување се активира со притискање на `Tab`.

<sup>22</sup>Scipy Lecture Notes <http://scipy-lectures.org/>

<sup>23</sup>Creative Commons <https://creativecommons.org/>

<sup>24</sup>Cython <https://cython.org/>

<sup>25</sup>The Python Tutorial. <https://docs.python.org/3/tutorial/index.html>

Викикнигата Програмирање во Питон<sup>26</sup>, обете во употребува на МИТ.

---

<sup>26</sup>Python Programming [https://en.wikibooks.org/wiki/Python\\_Programming](https://en.wikibooks.org/wiki/Python_Programming)

# 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>.
- 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).
- 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.
- 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.
- 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 Vahteras, 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/>.
- Софија 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*.