

# Počítačové zpracování signálu

RNDr. Zbyšek Posel, Ph.D.



Ústí nad Labem 2020

**Kurz:** Matematický software  
**Obor:** Aplikovaná informatika

**Klíčová slova:**

**Anotace:** Úvodní kurz Počítačové zpracování signálu je zaměřen na získání znalostí o základních metodách zpracování v časové a frekvenční oblasti a jejich implementace pomocí jazyka Python. Výuková zátěž je rozdělena rovnoměrně mezi teoretickou a implementační část. Studenti v průběhu kurzu vypracovávají ročníkovou (skupinovou) práci, kterou na konci předmětu obhajují formou prezentace a validace vypracovaného programu v Pythonu. Kurz je určen pro studenty, kteří úspěšně prošli úvodním kurzem Pythonu (*Algoritmizace a programování I*) a základním kurzem matematiky (*Matematika I*).

Jazyková korektura nebyla provedena, za jazykovou stránku odpovídá autor.

© Katedra informatiky, PřF, UJEP v Ústí nad Labem, 2020  
Autor: RNDr. Zbyšek posel, Ph.D.

# Obsah

Úvodní slovo	4
1 Počítačové zpracování signálu	6
2 Charakteristiky signálu v časové oblasti	9
3 Konvoluce, kovariance a korelace signálu	12
4 Posílení signálu v šumu - kumulační a korelační techniky	15
5 Metody zpracování signálu ve frekvenční oblasti	18
6 Transformace signálu	21
7 Diskrétní Fourierova transformace	24
8 Zpracování reálných signálů	27

# Úvodní slovo

Opora k předmětu *Počítačové zpracování signálu* je zaměřena zejména na využití nástrojů jazyka Python, kterými jsou knihovny NumPy, SciPy a SymPy, pro implementaci metod zaměřených na získávání informací z jednorozměrného signálu. Vzhledem k širokému záběru této oblasti jsou nejdříve popsány základní principy, které se uplatňují vždy a poté jsou opory rozděleny do tří tematických celků. Nejprve je pozornost zaměřena na metody pracující v časové oblasti. Jde o výpočet základních charakteristik signálu, ale i o velice důležité metody konvoluce, kovariance a korelace, na kterých stojí mnoho pokročilejších metod. Tato část je zakončena představením metod pro posílení signálu v šumu, které vhodně propojují například korelaci a kumulační techniky. V druhé části je pozornost obrácena směrem ke spektru signálu a představení metod pro práci v tomto prostoru. Nejvíce je zohledněn princip Fourierovy transformace a metod, které ji rozšiřují a to zejména vlnkové transformace, kepsrální analýzy nebo Hilbertovy transformace. V rámci dotčených kapitol je představena myšlenka diskrétní a rychlé Fourierovy transformace a aplikace filtrů pracujících ve spektrální oblasti. Poslední celek je zaměřen na zpracování reálných signálů a to zejména z oblasti analýzy fyziologických a biologických dat, kde je pozornost zaměřena na již představenou Hilbertovu nebo vlnkovou transformaci. V průběhu kurzu jsou zadány dvě seminární práce, které následujíc po tematicky ucelených částech a to po metodách zpracování signálu v časové a frekvenční oblasti. Všechny výše zmíněné kapitoly jsou doplněny o implementace v jazyku Python, které lze prostudovat po přihlášení do e-learningového systému MOODLE.

# Příklad seminární práce

V průběhu kurzu budou zadány dvě seminární práce, na kterých budou moci studenti pracovat v průběhu speciálně vyčleněných seminářů. Součástí seminární práce je protokol o vypracování a programové kódy. Při plnění těchto prací mohou studenti spolupracovat na tvorbě programů, ale protokol odevzdává každý sám za sebe. Seminární práce může obsahovat například tato zadání.

## Posílení EKG signálu v šumu

Pomocí korelačních a kumulačních technik získáte jednu periodu EKG signálu zatíženého uměle šumem. Pomocí korelačních technik nejdříve odhadněte jednu periodu tohoto signálu a poté pomocí kumulační techniky (prostá kumulace, exponenciální, s klouzavým oknem aj.) zjistíte jednu periodu EKG signálu bez šumu. Výstupy budou korelační funkce EKG signálu, vliv délky periody na kvalitu průměrovaného signálu, jedna perioda EKG signálu bez šumu, která vykazuje nejlepší kvalitu.

## Detekce vzorů v EKG signálu pomocí statistických ukazatelů a korelačních technik

Z databáze MIT-BIH Arrhythmia získáte deset nezávislých signálů EKG. Pomocí korelace, kovariance a statistických ukazatelů (Vašeho výběru) zjistíte podobnost signálů, a to jak lokální, tak globální. Podobnost přehledně prezentujte graficky, pomocí kovarianční matice, korelačních funkcí aj. Při analýze nezapomeňte odstranit ty artefakty, které by Vám mohly zabránit nalézt nejlepší shodu včetně odstranění trendu, šumu a vlivu počátku na výsledky korelace. Výstupy budou kovarianční matice, mapa míry shody jednotlivých signálů se signálem testovaným, jednotlivé korelační funkce a vybrané statistické ukazatele.

## Návrh filtru EKG signálu ve spektru s využitím Fourierovy transformace

Pomocí Fourierovy transformace EKG signálů získáte jeho „reprezentativní“ spektrum. Pomocí tohoto spektra navrhnete filtr, který odstraní šum z EKG signálu, umístěném v souboru, který Vám bude dodán. Pomocí (diskrétní) rychlé Fourierovy transformace získáte spektrum filtrovaného signálu a po aplikaci filtru převedte pomocí zpětné transformace signál zpět do časové oblasti. Prezentujte, jaký typ šumu původní signál obsahoval (nízkofrekvenční, vysokofrekvenční aj.). Graficky zobrazte, jak byla zlepšena kvalita signálu po aplikaci filtru. Dále bude výstupem reprezentativní spektrum signálu, spektrum zašumněného signálu, spektrum signálu po filtraci a jeho časový průběh.

# 1 Počítačové zpracování signálu



## CÍLE KAPITOLY

Cílem této úvodní kapitoly je definovat základní pojmy z oblasti počítačového zpracování signálu, poskytnout přehled jejich dělení, reprezentace signálu v digitální podobě, problematiku vzorkování včetně aliasingu, dělení šumu, který v signálu může nastat a algoritmů pro generování signálu na počítači. V neposlední řadě je cílem poskytnout přehled o aktuálních trendech ve zpracování signálu a oblastech jejich použití, např. v IOTA senzorech (Edge computing), snímání aktivity těla (biologické a fyziologické signály) nebo vyhodnocení experimentálních dat z ostatních měřících zařízení.

V této kapitole se dozvíte

- kde všude se vyhodnocuje signál a proč je užitečný,
- jak se převádí analogový signál v počítači (vzorkování, kvantování) a co je aliasing,
- jaký může nastat v signálu šum,
- jaké jsou základní přístupy a metody v analýze signálu a kde nachází uplatnění.



## KLÍČOVA SLOVA

dělení signálů, šum v signálu, počítačový model signálu, A/D převodník, amplituda, frekvence, Nyquistův-Shannonův teorém, aliasing

**Náplň látky je popsána v následujících zdrojích**

- Současné trendy ve zpracování signálu lze najít v online zdrojích např. zde [1] nebo zde [2].
- Počítačový popis signálu lze najít například zde [3] nebo zde [4].
- Vztahy mezi signálem a systémem, který ho generuje lze nalézt například zde [5].
- Převod analogového signálu do digitální podoby a problematiku vzorkování lze nalézt například zde [5] nebo zde [3].

Jednoduché příklady generování periodických signálů a vliv parametrů na jejich tvar nebo vliv vzorkování na kvalitu podobu signálu lze dále nalézt v e-learningovém systému MOODLE.



## SHRNUTÍ

Po prostudování byste měli mít následující dovednosti a znalosti.

- Mít povědomí o problematice vzorkování signálu.
- Mít povědomí o šumu, který může signál zkreslit.
- Umět popsat periodický signál pomocí matematického modelu.
- Umět pracovat s funkcemi knihoven NumPy, a SciPy, pro generování různých typů signálů.
- Mít povědomí o tom, kde všude se signály používají pro vyhodnocení stavu systému.

## OTÁZKY

1. Jaké druhy periodických signálů znáte a kde se tyto signály vyskytují?
2. Jaké matematické modely jsou pro popis těchto signálů využity?
3. Jak lze periodické signály vygenerovat pomocí funkcí knihoven NumPy a SciPy?
4. Jak můžete zjistit, že Vaše vzorkování je správné?

## OTÁZKY K ZAMYŠLENÍ

1. V jaké části spektra zpracováváme signály nejčastěji?
2. Jaké jsou výhody a nevýhody reprezentace signálů matematickým modelem?
3. Jak lze odhadnout, který šum signál obsahuje?

## ÚKOLY

1. Pomocí jazyka Python nagenertejete sadu různých periodických signálů (sinusový, obdélníkový, pilový aj.).
2. Pro jejich generování využijte následně i knihovny NumPy a SciPy.
3. Zjistěte, jaké jsou hardwarové nároky (paměť, procesor, čas výpočtu aj.) pro nagenrování signálu s danou periodou a s celkovou dobou záznamu v rozsahu 10 až 30 minut.
4. Při generování periodických signálů měňte vzorkování a popište jeho vliv na zaznamenaný signál.



## 2 Charakteristiky signálu v časové oblasti



### CÍLE KAPITOLY

Cílem této kapitoly je procvičit si základní operace se signály a spočítat základní charakteristiky signálu v časové oblasti a to včetně modulace a vysvětlení demodulace signálu.

V této kapitole se dozvíte

- jaké jsou základní charakteristiky signálů v časové oblasti,
- jak lze tyto charakteristiky efektivně počítat,
- jak jsou definovány modulace signálů (amplitudová, frekvenční a fázová),
- jak probíhá jejich demodulace,
- jaké nástroje jazyka Python lze pro tyto úlohy využít.



### KLÍČOVÁ SLOVA

amplituda, frekvence, fáze, střední hodnota, práce a energie signálu, modulace, frekvenční zdvih

**Náplň látky je popsána v následujících zdrojích**

- Základní charakteristiky signálů v časové oblasti a jejich výpočet lze nalézt například zde [3] nebo zde [5].
- Princip modulace signálu lze nalézt například zde [17].
- Příklad frekvenční modulace pomocí Pythonu lze nalézt například zde [18].

Jednoduché skripty na výpočet základních charakteristik signálů v časové oblasti a příklady modulace a demodulace signálů lze dále najít v e-learningovém systému MOODLE.



### SHRNUTÍ

Po prostudování byste měli mít následující dovednosti a znalosti.

- Umět vypočítat jednoduché charakteristiky signálů v časové oblasti.
- Umět definovat různé druhy modulace signálu a ukázat vliv jednotlivých parametrů modulace na modulovaný signál.

- Využít pro výše zmíněné úlohy prostředí Python.

## OTÁZKY

1. Jaké jsou základní charakteristiky signálu v časové oblasti?
2. Kde se tyto charakteristiky používají?
3. Jaké druhy modulace znáte a kde se používají?

## OTÁZKY K ZAMYŠLENÍ

1. Jak probíhá demodulace signálu?

## ÚKOLY

1. Pomocí vlastní nebo vestavěné funkce Pythonu nagenerte různé druhy periodických signálů.
2. Spočítejte jejich základní charakteristiky včetně, práce a energie signálu a porovnejte je mezi sebou.
3. Postupně proveďte amplitudovou, frekvenční a fázovou modulaci dříve nagenovaných signálů.
4. Pomocí knihovny `matplotlib` vizualizujte výsledky.



# 3 Konvoluce, kovariance a korelace signálu



## CÍLE KAPITOLY

Cílem této kapitoly je poskytnout přehled nejvíce využívaných metod pro zpracování signálu v časové oblasti a to zejména konvoluce, kovariance a korelace. Důraz je vedle definice a pochopení těchto metod kladen dále na jejich implementaci na různé druhy periodických signálů pomocí vlastních skriptů nebo funkcí knihoven jazyka Python.

V této kapitole se dozvíte

- princip konvoluce, kovariance a korelace a jejich vzájemný vztah,
- využití těchto metod pro získání informací z periodických signálů,
- implementaci těchto metod v knihovnách NumPy a SciPy jazyka Python.



## KLÍČOVÁ SLOVA

kovariance, konvoluce, korelace, auto-korelace, kovarianční matice

**Náplň látky je popsána v následujících zdrojích**

- Matematické definice lze nalézt například zde [13] nebo zde [3].
- Numerický výpočet těchto metod lze nalézt například zde [14] nebo zde [19].
- Implementace metod v knihovnách NumPy a SciPy jazyku Python lze nalézt zde [6] nebo zde [7].

Implementace konvoluce, kovariance a korelace lze najít dále v e-learningovém systému MOODLE.



## SHRNUTÍ

Po prostudování byste měli mít následující dovednosti a znalosti.

- Znat definice a podstatu konvoluce, kovariance a korelace signálu a znát význam jednotlivých parametrů metod.
- Umět aplikovat tyto metody pomocí vlastních skriptů nebo funkcí knihoven NumPy a SciPy.

- Umět interpretovat výsledky těchto metod.

## OTÁZKY

1. Jak se od sebe liší konvoluce, kovariance a korelace?
2. Jaké je naopak propojení těchto metod?
3. Jak se od sebe liší korelace a auto-korelace?
4. Jakou křivku můžeme čekat, pokud aplikujeme auto-korelaci na různé druhy periodických signálů?

## OTÁZKY K ZAMYŠLENÍ

1. Kde se v analýze signálů využívají výstupy těchto metod?

## ÚKOLY

1. Nagenerte různé druhy periodických signálů.
2. Postupně aplikujte konvoluci, kovarianci a korelaci na nagenované signály. Výsledky vizualizujte pomocí knihovny `matplotlib`.
3. Porovnejte výstupy auto-korelace na různé druhy periodických signálů.



## 4 Posílení signálu v šumu - kumulační a korelační techniky



### CÍLE KAPITOLY

Cílem této kapitoly je poskytnout přehled metod pro posílení signálu v šumu založených na korelačních a kumulačních technikách a to pro periodické signály s aditivní složkou šumu. Budou zmíněny metody od prosté kumulace až po exponenciální.

V této kapitole se dozvíte

- jaké jsou předpoklady pro použití korelačních a kumulačních technik pro posílení signálu v šumu,
- jaké jsou metody kumulace od prosté kumulace, až po exponenciální kumulaci,
- jak lze tyto metody implementovat do jazyka Python s využitím knihoven NumPy a SciPy.



### KLÍČOVÁ SLOVA

korelace, prostá kumulace, exponenciální kumulace, pevné okno, poměrné zlepšení signálu k šumu,

**Náplň látky je popsána v následujících zdrojích.**

- Přehled základních kumulačních a korelačních metod pro posílení signálu v šumu lze nalézt například zde [3] nebo zde [4].
- Implementace vybraných metod v knihovnách jazyka Python NumPy a SciPy lze nalézt zde [6] a zde [7].

Příklady implementace jednotlivých kumulačních technik lze dále nalézt v e-learningovém systému MOODLE.



### SHRNUTÍ

Po prostudování byste měli mít následující dovednosti a znalosti.

- Mít povědomí o podmínkách, kdy lze kumulační a korelační techniky pro posílení signálu v šumu použít.
- Umět implementovat základní kumulační a korelační techniky pomocí jazyka Python.

- Při implementaci využít nástroje knihoven NumPy a SciPy.

## OTÁZKY

1. Za jakých podmínek, lze použít kumulační techniku?
2. Jak budete postupovat, pokud nebude perioda signálu předem známa?
3. Jak vypadá jedna iterace korelační metody pro posílení signálu v šumu?

## OTÁZKY K ZAMYŠLENÍ

1. Jaké jsou nevýhody těchto metod?
2. Lze tyto metody využít v real-time aplikacích? Pokud ano, jaký je podle Vás nejlepší postup z hlediska rychlosti a managementu s daty?

## ÚKOLY

1. Pomocí Vašeho skriptu nebo funkcí knihoven NumPy a SciPy nagenerezujte periodický signál s aditivním šumem.
2. Pomocí kumulační techniky s pevným a klouzavým oknem provedte posílení signálu v šumu. Stejnou implementaci proveďte i pro exponenciální pokles.
3. Vypočítejte výkonové parametry (poměr zlepšení signálu k šumu aj.) a výsledky porovnejte.
4. Pomocí funkcí knihoven NumPy a SciPy implementujte iterativní postup pro posílení signálu k šumu založený na korelační technice.
5. Výsledky porovnejte s kumulačními technikami aplikovanými dříve.



# 5 Metody zpracování signálu ve frekvenční oblasti



## CÍLE KAPITOLY

Cílem této kapitoly je poskytnout přehled metod zpracování signálu ve frekvenční oblasti založených na znalosti spektra signálu, včetně návrhu a implementace jednoduchých filtrů (lineární, propust' nebo adaptivní).

V této kapitole se dozvíte

- co je spektrum signálu a proč je užitečné ho znát,
- jaké metody zpracování signálu jsou založené na znalosti spektra,
- jak probíhá návrh jednoduchých filtrů ve frekvenční oblasti,
- jaké metody jsou implementovány v prostředí Pythonu.



## KLÍČOVÁ SLOVA

spektrum signálu, Fourierova transformace, Hilbertova transformace, horním dolní a pásmová propust', lineární filtr, adaptivní filtr

**Náplň látky je popsána v následujících zdrojích.**

- Popis spektra signálu lze nalézt například zde [20] nebo zde [21].
- Metody zpracování signálu založené na znalosti spektra lze nalézt například zde [22], zde [23] nebo zde [3].
- Přehled filtrů a jejich aplikace v jazyku Python lze nalézt zde [20] a zde [21].
- Přehled jednotlivých funkcí knihoven NumPy a SciPy pro návrh a implementaci filtrů lze dále nalézt zde [6] a zde [7].

Implementace vybraných filtrů na periodický signál se šumem lze dále nalézt v e-learningovém systému MOODLE.



## SHRNUTÍ

Po prostudování byste měli mít následující dovednosti a znalosti.

- Vědět, co je spektrum a proč potřebujeme jeho znalost. Jaké vlastnosti lze ze spektra signálu zjistit.
- Mít přehled o filtrech, které jsou založené na znalosti spektra signálu a o jejich použití.
- Umět implementovat jednoduché filtry na bázi propustí, nebo na bázi lineárních a adaptivních metod.
- Umět implementovat výše zmíněné filtry v Pythonu za pomoci NumPy a SciPy.

## OTÁZKY

1. Co je spektrum a proč ho chceme znát?
2. Jaké vlastnosti signálu lze ze spektra odhadnout pouhým pohledem?
3. Jaké metody využívají znalosti spektra signálu?
4. Na čem jsou založeny lineární filtry a propusti?

## OTÁZKY K ZAMYŠLENÍ

1. Jaká jsou omezení při analýze vlastností signálu ze spektra?

## ÚKOLY

1. Pomocí Vašeho skriptu nebo funkcí knihoven NumPy a SciPy nagenertejte periodický signál s různými druhy šumu (aditivní, náhodný). Využijte k tomu dobrý i špatný generátor náhodných čísel.
2. Pomocí funkcí knihoven NumPy a SciPy získejte spektra těchto signálů a porovnejte je.
3. Na signály uvedené výše aplikujte postupně lineární filtr, propusti a adaptivní filtry. Měňte parametry filtrů a porovnejte výsledky pomocí knihovny matplotlib.



## 6 Transformace signálu



### CÍLE KAPITOLY

Cílem této kapitoly je poskytnout přehled o základních transformacích signálu, která se při zpracování využívají, od klasické Fourierovy transformace, přes vlnkovou transformaci až k Hottelingově nebo Z-transformaci. Dále je cílem představit některé další speciální transformace, využívané ve specializovaných oblastech zpracování signálu, např. Hilbertova transformace při zpracování biologických a fyziologických signálů.

V této kapitole se dozvíte

- jaké jsou transformace signálů,
- jak jsou tyto transformace využívány při zpracování signálů.



### KLÍČOVÁ SLOVA

Fourierova transformace, vlnková transformace, škálogram, kepstrum, Hottelingova transformace, Walshova transformace, Z-transformace, Hilbertova transformace

**Náplň látky je popsána v následujících zdrojích.**

- Fourierovu transformaci lze nalézt například zde [22] nebo zde [3].
- Popis vlnkové transformace lze nalézt například zde [23].
- Z-transformaci lze nalézt například zde [3] nebo zde [4].
- Ostatní typy transformací lze nalézt například zde [21] nebo zde [3].
- Hilbertovu transformaci a ostatní transformace používané na fyziologický signál lze nalézt například zde [29], zde [25] nebo zde [26].

Implementace vybraných transformací na signál a vizualizace výstupů lze dále nalézt v e-learningovém systému MOODLE.



### SHRNUTÍ

Po prostudování byste měli mít následující dovednosti a znalosti.

- Mít přehled o transformacích používaných při zpracování signálů.

- Mít povědomí o transformacích používaných při zpracování biologických a fyziologických signálů.
- Umět implementovat vybrané transformace na periodický signál se šumem.
- Umět využít funkce knihoven NumPy a SciPy pro transformaci signálu.

## OTÁZKY

1. Proč se transformace při zpracování signálu používají?
2. Jaký je rozdíl mezi škálovaným vlnkovým spektrem a spektrem Fourierovy transformace?
3. Co je kepsstrum a kde se využívá?
4. Jaký je princip Hilbertovy transformace?

## OTÁZKY K ZAMYŠLENÍ

1. Proč je mnoho výše uvedených transformací založeno na Fourierově?

## ÚKOLY

1. Pomocí Vašeho skriptu nebo funkcí knihoven NumPy a SciPy nagenertejte periodické signál se šumem.
2. Postupně aplikujte vybrané transformace (Fourierova, vlnková, kosínová aj.) a vizualizujte jejich výstupy. Výsledky porovnejte z hlediska množství informací obsažených ve výstupech a z hlediska časové a hardwarové náročnosti metod.



# 7 Diskrétní Fourierova transformace



## CÍLE KAPITOLY

Cílem této kapitoly je popsat princip diskrétní Fourierovy transformace (DFT) a její implementace v jazyku Python pomocí vlastních skriptů nebo pomocí knihoven NumPy a SciPy. Dále je cílem této kapitoly aplikovat DFT širokou řadu signálů a porovnat výstupy.

V této kapitole se dozvíte

- jak je definována Fourierova řada a na ní navázána Fourierova transformace,
- jaký je vliv koeficientů řady na aproximaci,
- jak je definována diskrétní Fourierova transformace,
- jak lze DFT implementovat do prostředí Python a jaké jsou chyby těchto implementací,
- jak je DFT implementována v knihovnách NumPy a SciPy.



## KLÍČOVÁ SLOVA

Fourierova řada, aproximace, DFT, posun ve spektru, změna měřítka, rychlá Fourierova transformace, Cooley-Tukey algoritmus

**Náplň látky je popsána v následujících zdrojích.**

- Definice Fourierovy řady lze nalézt například zde [12] a Fourierovy transformace například zde [12].
- Výpočet DFT pomocí algoritmu rychlé Fourierovy transformace lze nalézt například zde [27].
- Popis Cooley-Tukey algoritmu lze nalézt například zde [28].

Implementace vybraných numerických metod integrace funkce jedné proměnné lze dále nalézt v e-learningovém systému MOODLE.



## SHRNUTÍ

Po prostudování byste mít následující dovednosti a znalosti.

- Umět aproximovat periodickou funkci Fourierovou řadou.

- Znat využití DFT a její implementace v knihovnách jazyka Python.
- Umět popsat výpočet spektra pomocí DFT a mít povědomí o numerických chybách, které mohou vzniknout.
- Umět získat spektrum pomocí DFT s využitím vlastních skriptů nebo pomocí funkcí knihoven NumPy a SciPy.

## OTÁZKY

1. Na jakém principu je založena aproximace funkce pomocí Fourierovy řady?
2. Jaké chyby vznikají při aproximaci funkce v případě sinové a pilové funkce?
3. Jaký je princip DFT a její implementace pomocí rychlé Fourierovy transformace?
4. Na čem je závislý rozsah frekvencí, které lze pomocí DFT získat?

## OTÁZKY K ZAMYŠLENÍ

1. Jak lze z frekvencí získat zpět původní signál?
2. Jak bude vypadat signál, pokud při zpětné transformaci budete uvažovat pouze reálnou nebo imaginární složku DFT?

## ÚKOLY

1. Pomocí Fourierovy řady aproximujte jednoduchou lineární funkci a obdélníkový signál. Parametry řady odvoďte.
2. Pomocí vlastního skriptu proveďte DFT jednoduchého signálu složeného z konstantní složky a dvou periodických signálů a omezte počet vzorků na rozumné množství ( $< 100$ ).
3. Proveďte to samé pomocí funkcí knihoven NumPy a SciPy.
4. Pomocí zpětné transformace získajte ze spektra původní signál. Vizualizujte, jak vypadá zpětně transformovaný signál při zohlednění pouze reálné nebo imaginární složky transformace.



## 8 Zpracování reálných signálů



**CÍLE KAPITOLY** Cílem této kapitoly je poskytnout přehled o základních metodách pro zpracování reálných signálů. Pozornost je zde zaměřena na zpracování biologických a fyziologických signálů pomocí vybraných transformací (Fourierova, vlnková a Hilbertova) a na získání informací z transformovaného signálu, jako je například poloha QRS komplexu.

V této kapitole se dozvíte

- jaké metody a transformace se využívají pro zpracování reálných biologických a fyziologických signálů,
- kde lze tyto signály získat a co jsou anotace,
- jak se ohodnocuje kvalita navržených algoritmů,
- kde se tyto algoritmy využívají a jaké jsou moderní trendy.



### KLÍČOVÁ SLOVA

EKG, EMG, QRS komplex, MIT-BIH Arrhythmia databáze, Hilbertova transformace, vlnková transformace, pozitivní prediktivita

**Náplň látky je popsána v následujících zdrojích.**

- Význam fyziologických a biologických signálů a informace v nich obsažené lze nalézt například zde [29].
- Databáze s volně dostupnými fyziologickými signály lze nalézt například zde [30].
- Přehled metod pro zpracování EKG signálu lze nalézt například zde [31].
- Ukazatele kvality navržených algoritmů lze nalézt například zde [25].
- Implementace Hilbertovy transformace v jazyku Python lze nalézt například zde [6] a zde [7] nebo zde [32].

Ukázku detekce QRS komplexu v signálu EKG nebo výpočet charakteristik pro signál EMG lze dále nalézt v e-learningovém systému MOODLE.



### SHRNUTÍ

Po prostudování byste mít následující dovednosti a znalosti.

- Mít povědomí o počítačových metodách pro analýzu biologických a fyziologických signálů.
- Umět získat volně dostupný fyziologický signál i s anotacemi.
- Umět implementovat vybrané algoritmy a modely pro analýzu EKG nebo EMG signálů.
- Umět pro analýzu využít funkce knihoven NumPy a SciPy.
- Umět ve zjednodušené míře implementovat dosažené výsledky.

## OTÁZKY

1. Jaké jsou metody pro analýzu biologických a fyziologických signálů?
2. Jaké jsou moderní trendy při analýze EKG a EMG signálů?
3. Jak lze ohodnotit úspěšnost navrženého algoritmu?

## OTÁZKY K ZAMYŠLENÍ

1. Jaké chyby a šum je nutné před samotnou analýzou signálu vyloučit?

## ÚKOLY

1. Z veřejně dostupné databáze si stáhněte EKG nebo EMG signál i s anotacemi.
2. Pomocí vybrané metody detekujte QRS komplex v EKG signálu nebo spočítejte celkovou práci svalu v EMG signálu.
3. Úspěšnost algoritmu porovnejte pomocí statistických ukazatelů s poskytnutými anotacemi.



# Literatura

- [1] Signal Processing [online]. Elsevier B.V. ©2020 [cit. 04.06.2020], Dostupné z: <https://www.journals.elsevier.com/signal-processing>
- [2] IEEE Signal Processing Society [online]. IEEE©2020 [cit. 04.06.2020], Dostupné z: <https://signalprocessingsociety.org/>
- [3] JAN, Jiří. *Číslicová filtrace, analýza a restaurace signálů*. VUTIUM Brno, 2002. ISBN 80-214-2911-9.
- [4] UHLÍŘ, Jan; SOVKA, Pavel. *Číslicové zpracování signálů*. ČVUT Praha, 2002, ISBN 80-01-02613-2.
- [5] ŠEBESTA, Vladimír. *Systémy, procesy a signály*. VUT Brno, 2001, ISBN 80-214-1925-3.
- [6] NumPy [online]. NumPy developers: ©2020 [cit. 04.06.2020], Dostupné z: <https://numpy.org/>
- [7] SciPy [online]. SciPy developers: ©2020 [cit. 04.06.2020], Dostupné z: <https://www.scipy.org/>
- [8] matplotlib [online]. The Matplotlib development team ©2012-2018 [cit. 04.06.2020], Dostupné z: <https://matplotlib.org/>
- [9] CHARVÁT, Jura; BUDÍNSKÝ, Bruno. *Matematika I-část 1.*, ČVUT Praha, 2000, ISBN 80-01-02175-0.
- [10] CHARVÁT, Jura; BUDÍNSKÝ, Bruno. *Matematika I-část 2.*, ČVUT Praha, 2000, ISBN 80-01-02174-2.
- [11] CHARVÁT, Jura; BUDÍNSKÝ, Bruno. *Matematika II*, ČVUT Praha, 1996, ISBN 80-01-01092-9.
- [12] REKTORYS, Karel a kol. *Přehled užití matematiky-svazek 1*. SNTL Praha, 1988. ISBN 04-022-88-01.
- [13] REKTORYS, Karel a kol. *Přehled užití matematiky-svazek 2*. SNTL Praha, 1988. ISBN 04-022-88-02.
- [14] VICHER, Miroslav. *Numerická matematika*. UJEP Ústí nad Labem, 2003. ISBN 80-7044-516-5.
- [15] RALSTON, Anthony. *Základy numerické matematiky*. Academia, 1978 ISBN 80-105-1256-4.
- [16] HRACH, Rudolf. *Počítačová fyzika I*. UJEP, 2003. ISBN 80-7044-521-1

- [17] ŽALUD, Václav. *Moderní radioelektronika*. BEN, 2000. ISBN 80-86056-47-3
- [18] Simple Python FM Synthesis [online]. Doug Olson ©2020 [cit. 04.06.2020], Dostupné z: <http://www.mrcolson.com/2016/04/21/Simple-Python-FM-Synthesis.html>
- [19] KNUTH, Donald. *The Art of Computer Programming, Volumes 1-4A Boxed Set*. Addison-Wesley Professional, 2011. ISBN 978-0321751041.
- [20] SHENOI, B. A. *Introductin to Digital Signal Processing and Filter Design*. John Wiley&Sons, 2005. ISBN 0471656380.
- [21] UNPINGCO, José. *Python for Signal Processing*. Springer, 2014. ISBN 978-3-319-01341-1.
- [22] HLAVÁČ, Václav. *Vlnková transformace* [online].[cit. 19.03.2020]. Dostupné z: <http://people.ciirc.cvut.cz/~hlavac/TeachPresCz/11DigZprObr/12FourierTxCz.pdf>.
- [23] ŠVEC, Martin. *Waveletové transformace*. UJEP, 2008. ISBN 978-80-7044-987-5.
- [24] SVATOŠ, Josef *Biologické signály I: geneze, zpracování a analýza*. ČVUT Praha, 1998. ISBN 80-01-01822-9.
- [25] BENITEZ, D. et al. The use of the Hilbert transform in ECG signal analysis. *Computers in Biology and Medicine*. 2001, **31**(5), 399-406. ISSN
- [26] RANGAYYAN, Rangaraj M. *Biomedical signal analysis*. IEEE Press, 2015. ISBN 978-0-470-0-91139-6.
- [27] RAO, Kamisetty R.; KIM, Do N.; HWANG, Jae J. *ast Fourier Transform - Algorithms and Applications*. Springer, 2010. ISBN 978-1402066283.
- [28] COOLEY, DJ. W. et al. An algorithm for the machine calculation of complex Fourier series. *Math. Comput.* 2001, **19**(90), 297-301. ISSN 0025-5718
- [29] SVATOŠ, Josef *Biologické signály I: geneze, zpracování a analýza*. ČVUT Praha, 1998. ISBN 80-01-01822-9.
- [30] PhysioNet - The Research Resource for Complex Physiologic Signals [online]. MIT Laboratory for Computational Physiology [cit. 19.03.2020], Dostupné z: <https://physionet.org/>
- [31] GACEK, Adam; PEDRYCZ, Witold. *ECG Signal Processing, Classification and Interpretation*. Springer, 2012. ISBN 978-0-85729-867-6.
- [32] The WFDB Software Package [online]. [cit. 19.03.2020], Dostupné z: <https://archive.physionet.org/physiotools/wfdb.shtml>