



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

ÚSTAV TELEKOMUNIKACÍ

DEPARTMENT OF TELECOMMUNICATIONS

TVORBA LABORATORNÍCH ÚLOH DO ANGLICKÉ VERZE PŘEDMĚTU ANALÝZA SIGNÁLŮ A SOUSTAV

DEVELOPMENT OF LABORATORY EXERCISES FOR SUBJECT THE SIGNALS AND SYSTEMS ANALYSIS

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

Josef Doleček

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. Vojtěch Zvončák

BRNO 2017

Bakalářská práce

bakalářský studijní obor **Audio inženýrství**
Ústav telekomunikací

Student: Josef Doleček

ID: 174277

Ročník: 3

Akademický rok: 2016/17

NÁZEV TÉMATU:

Tvorba laboratorních úloh do anglické verze předmětu Analýza signálů a soustav

POKyny PRO VYPRACOVÁNÍ:

Vytvořte laboratorní úlohy anglické verze předmětu Analýza signálů a soustav, které budou naprogramovány formou skriptů pro prostředí MATLAB. Úlohy budou zaměřeny na oblasti diskrétní konvoluce, zobrazení komplexních čísel v komplexní rovině, spektrální analýza, korelace, vzorkovací teorém, z - transformace, použití filtrů DP, HP, PP, PZ. Ke každé oblasti budou vytvořeny minimálně dva rozsáhlé skripty.

DOPORUČENÁ LITERATURA:

[1] SMÉKAL, Zdeněk. Systémy a signály: 1D a 2D diskrétní a číslicové zpracování. Praha: Sdělovací technika, 2013. ISBN 978-80-86645-23-0.

[2] INGLE, Vinay K. a John G. PROAKIS. Digital signal processing using MATLAB. 3rd ed. Stamford, Conn.: Cengage Learning, c2012. ISBN 1111427372.

Termín zadání: 1.2.2017

Termín odevzdání: 8.6.2017

Vedoucí práce: Ing. Vojtěch Zvončák

Konzultant:

doc. Ing. Jiří Mišurec, CSc.
předseda oborové rady

UPOZORNĚNÍ:

Autor bakalářské práce nesmí při vytváření bakalářské práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č.40/2009 Sb.

ABSTRAKT

Bakalářská práce popisuje základní operace se signály a pomocí názorných ukázek v programu Matlab má tyto operace přiblížit studentům předmětu CASS (Signals and Systems Analysis). V teorii jsou popsány poznatky o spojitých signálech (jejich rozdělení, periodické signály, Fourierova řada), komplexních číslech, vzorkování a diskrétních signálech (operace konvoluce a korelace). Praktické aplikace pak objasňují: syntézu harmonických složek obdélníkového signálu, konvoluci a korelaci dvou diskrétních signálů, využití korelace k detekci vzdálenosti objektu od radaru, komplexní model harmonického signálu a vzorkování signálů.

KLÍČOVÁ SLOVA

Signály, Fourierova řada, diskrétní konvoluce, vzájemná korelace, komplexní model harmonického signálu, vzorkování, Matlab.

ABSTRACT

The bachelor thesis deals with basic operation with signals and, using demonstrative applications in Matlab, it should explain these operation to CASS (Signals and Systems Analysis) students. In the theory part continuous-time signals (signals classification, periodic signals, Fourier series), complex numbers, sampling and discrete-time signals (operation of convolution and correlation) are described. Practical applications explain: square signal harmonics synthesis, two discrete-time signals convolution and cross-correlation, object distance detection by radar using cross-correlation, complex model of harmonic signal and signals sampling.

KEYWORDS

Signals, Fourier series, discrete convolution, cross-correlation, complex model of harmonic signal, sampling, Matlab.

DOLEČEK, J. *Tvorba laboratorních úloh do anglické verze předmětu Analýza signálů a soustav*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací, 2017. 34 s. Vedoucí bakalářské práce Ing. Vojtěch Zvončák.

PROHLÁŠENÍ

Prohlašuji, že svoji bakalářskou práci na téma Tvorba laboratorních úloh do anglické verze předmětu Analýza signálů a soustav jsem vypracoval samostatně pod vedením vedoucího bakalářské práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené bakalářské práce dále prohlašuji, že v souvislosti s vytvořením této bakalářské práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a/nebo majetkových a jsem si plně vědom následků porušení ustanovení § 11 a následujících zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), ve znění pozdějších předpisů, včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

V Brně dne

.....

(podpis autora)

PODĚKOVÁNÍ

Děkuji vedoucímu této bakalářské práce, panu inženýrovi Vojtěchu Zvončákovi, za vedení, pomoc a rady při zpracování této práce. Děkuji všem svým spolužákům, učitelům a zaměstnancům VUT za obohacení, kterého se mi dostalo tím, že jsem měl možnost se s nimi setkat při studiu na VUT. Děkuji mé manželce, rodičům a celé rodině za lásku, trpělivost a podporu.

V Brně dne

.....

(podpis autora)

OBSAH

Seznam obrázků	viii
Seznam tabulek	viii
Úvod	1
1 Signály	2
1.1 Úvod.....	2
1.2 Dělení signálů	2
2 Komplexní čísla	4
3 Periodické signály a jejich spektrum	6
3.1 Úvod.....	6
3.2 Fourierova řada	6
3.3 Komplexní model harmonického signálu	9
4 Vzorkování	10
4.1 Úvod.....	10
4.2 Ideální vzorkování	10
5 Signály s diskrétním časem	12
5.1 Úvod.....	12
5.2 Operace s diskrétními signály	13
6 Realizace názorných aplikací v programu Matlab	16
6.1 Sčítání harmonických složek obdélníkového signálu	16
6.2 Konvoluce diskrétních signálů.....	18
6.3 Korelace diskrétních signálů.....	20
6.4 Simulace radaru	21
6.5 Komplexní model harmonického signálu	22
6.6 Vzorkování.....	23
7 Závěr	25
Literatura	26

SEZNAM OBRÁZKŮ

Obrázek 1.1: Harmonický signál $u(t)$ daný rovnicí 1.2.	3
Obrázek 2.1: Zobrazení komplexního čísla v Gaussově rovině.	5
Obrázek 3.1: a) Pilovitý periodický signál se základní periodou T_1 ; b) - d) jedna perioda součtu 2, 3 a 4 harmonických složek pilovitého signálu.	7
Obrázek 5.1: Příklad grafického zobrazení obecného diskrétního signálu.	13

SEZNAM TABULEK

Tabulka 6.1: Tabulková metoda výpočtu diskrétní konvoluce.	19
Tabulka 6.2: Příklad výpočtu diskrétní konvoluce pomocí aplikace GUI1.	20

ÚVOD

Bakalářská práce se zabývá základními operacemi se spojitými a diskrétními signály a jejich realizací pomocí aplikací v programu Matlab, které mají být využity ve výuce předmětu Signals and systems analysis (CASS).

U aplikací by měl být kladen důraz především na názornost postupu výpočtu. Názornosti a pochopení postupu studenty můžeme dosáhnout především pomocí odkrývání jednotlivých kroků výpočtu a jejich grafického znázornění. Právě proto jsou vždy součástí aplikací grafy a – pokud nám to okolnosti dovolí – dostatečně podrobné výpisy dílčích výpočtů.

První kapitoly popisují teorii o signálech – nejprve jejich dělení, komplexní čísla, dále spektrum periodických signálů, vzorkování, diskrétní signály a operace s diskrétními signály. V poslední kapitole 6 pak nalezneme popisy aplikací programu Matlab s grafickým uživatelským rozhraním.

1 SIGNÁLY

1.1 Úvod

Každá zpráva, sdělení nebo údaj, který nás zajímá, který má pro nás nějaký význam, obecně nazýváme informací. Informace sama o sobě je nehmotná. Fyzikální veličinu, která nám dovolí přenášet nebo uchovávat informace, označujeme obecně jako signál. Signál je hmotným nositelem informace. Může to být například elektrické napětí, elektrický proud nebo výkon, tlak, teplota apod. Pro přenos informace se převážně používají elektrické a optické signály [1].

1.2 Dělení signálů

Signály můžeme dělit podle jejich vlastností. Základní dělení signálů podle náhodnosti jejich průběhu je na:

- signály deterministické,
- signály náhodné.

Deterministické (determinované, regulární, určené) signály jsou takové, které mají známou hodnotu závislé proměnné pro každou hodnotu nezávislé proměnné. Jejich průběh je definován známou funkcí nebo posloupností, např. $\cos x$, $\ln x$ apod. Dále se deterministické signály dělí na:

- deterministické signály periodické,
- deterministické signály neperiodické.

Signál (funkce) $s(t)$ je periodický, když existuje kladné číslo T_1 z oboru reálných čísel, tj. $T_1 \in \mathbb{R}$ a $T_1 > 0$, takové, že pro všechna reálná $t \in \mathbb{R}$ platí:

$$s(t + T_1) = s(t). \quad (1.1)$$

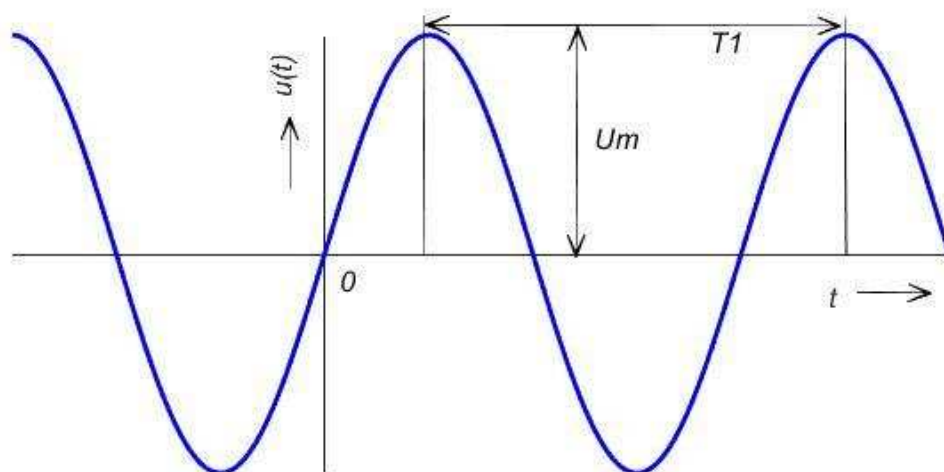
Tento periodický signál pak značíme jako $s_p(t)$. Nejmenší hodnota T_1 , pro kterou je podmínka splněna, se nazývá základní perioda. Periodický signál $s_p(t)$ lze dále dělit na:

- harmonický signál,
- neharmonický signál.

Harmonický signál je definován pomocí funkce kosinus nebo sinus:

$$u(t) = U_m \cdot \cos\left(\frac{2\pi}{T_1} t + \varphi_1\right). \quad (1.2)$$

Maximální hodnota U_m se nazývá amplituda, fázový rozdíl mezi počátkem funkce $\cos x$ a počátkem souřadnic je počáteční fáze φ_1 a kmitočet je roven $f_1 = 1/T_1$. Úhlový kmitočet $\omega_1 = 2\pi f_1 = \frac{2\pi}{T_1}$ [1]. Na obrázku 1.1 vidíme časový průběh obecného harmonického signálu.



Obrázek 1.1: Harmonický signál $u(t)$ daný rovnicí 1.2.

2 KOMPLEXNÍ ČÍSLA

I mezi signály se setkáváme s komplexními čísly. Komplexní číslo můžeme uvádět v algebraickém tvaru:

$$a = a_1 + ja_2, \quad (2.1)$$

kde $j = \sqrt{-1}$ je imaginární jednotka, a_1 je reálná složka a a_2 imaginární složka komplexního čísla.

Absolutní hodnota neboli modul má hodnotu:

$$|a| = \sqrt{a_1^2 + a_2^2}. \quad (2.2)$$

Argument komplexního čísla $\varphi = \arctg \frac{a_1}{a_2}$ a modul komplexního čísla můžeme využít k zápisu v goniometrickém tvaru:

$$a = |a|(\cos \varphi + j \sin \varphi). \quad (2.3)$$

Eulerův vztah říká, že:

$$e^{j\varphi} = \cos \varphi + j \sin \varphi, \quad (2.4)$$

což znamená, že:

$$a = |a|e^{j\varphi}. \quad (2.5)$$

Komplexní čísla znázorňujeme v Gaussově rovině, která je určena reálnou a imaginární osou [2] (viz obrázek 1.5).

V praxi se s komplexními čísly setkáme například u exponenciálního diskretního signálu vyjádřeného vztahem:

$$s[n] = a^n. \quad (2.6)$$

Ve vztahu popisujícím signál, mohou nastat dva případy podle toho, zda základ a je reálné nebo komplexní číslo. V případě, že je základ a reálné číslo, tvar diskretního signálu bude závislý na jeho hodnotě.

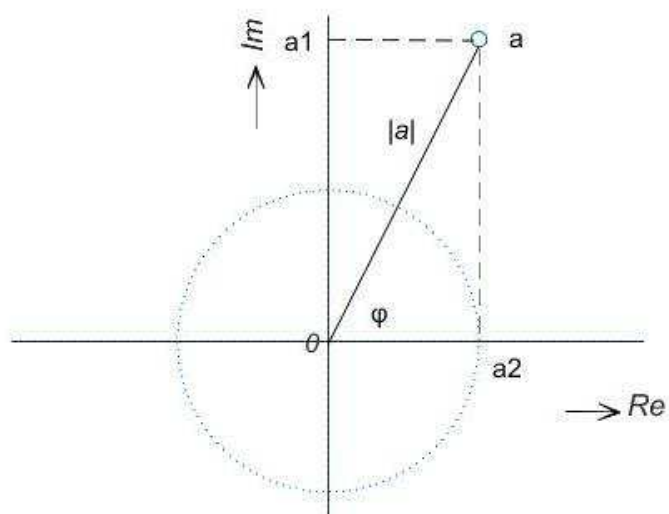
Je-li základ a komplexní číslo, lze jej vyjádřit jako

$$a = r \cdot e^{j\varphi} = r \cdot \cos \varphi + jr \cdot \sin \varphi. \quad (2.7)$$

a po dosazení vztahu (2.7) do definičního vztahu (2.6) získáme vztah

$$\begin{aligned} s[n] &= a^n = (r \cdot e^{j\varphi})^n = r^n \cdot e^{jn\varphi} = r^n (\cos \varphi n + jr \cdot \sin \varphi n) = \\ &= r^n \cos \varphi n + jr^n \cdot \sin \varphi n \end{aligned} \quad (2.8)$$

což nám umožňuje nahlížet zvlášť na reálnou a imaginární složku tohoto signálu [3].



Obrázek 2.1: Zobrazení komplexního čísla v Gaussově rovině.

Komplexní čísla jsou ve zpracování signálů široce využívána. Dalším příkladem je komplexní model harmonického signálu, který je popsán v kapitole 3.2.

3 PERIODICKÉ SIGNÁLY A JEJICH SPEKTRUM

3.1 Úvod

Každý periodický signál je tvořen součtem harmonických signálů, jejichž souhrn se nazývá spektrum. Proto spektrální analýze říkáme též harmonická analýza. Harmonická analýza nebo syntéza je operace, při níž je periodický signál rozkládán na dílčí harmonické složky (nebo skládán z harmonických složek),

- jejichž kmitočet je dán celistvým násobkem základního kmitočtu $f_1 = 1/T_1$, $\omega_1 = 2\pi f_1$. Kmitočet harmonických složek je tedy $1 \cdot f_1, 2 \cdot f_1, 3 \cdot f_1, 4 \cdot f_1, \dots$,
- amplituda dílčích harmonických složek je dána spektrem amplitud (modulů),
- počáteční fáze dílčích harmonických složek je dána spektrem fází.

Harmonická analýza pomocí rozkladu do složek Fourierovy řady a vyjádření signálu v kmitočtové oblasti nám přiblíží odhad očekávaných vlastností signálu lépe než pozorování jeho časových změn. U lineárních systémů navíc platí princip superpozice a můžeme sledovat vliv jednotlivých harmonických složek na celkovou odezvu [1].

Na obrázku 3.1 a) je vidět příklad periodického pilovitého průběhu. V bodech b) - d) pak vidíme, jak jsou postupně přidávány jednotlivé n -té harmonické složky s průběhem podle funkce sinus s rostoucí frekvencí $f_n = n f_1$ a klesající amplitudou $U_{mn} = \frac{U_m}{n}$. Průběh se blíží pilovitému signálu, úplný pilovitý signál je pak získán při nekonečném počtu harmonických složek.

3.2 Fourierova řada

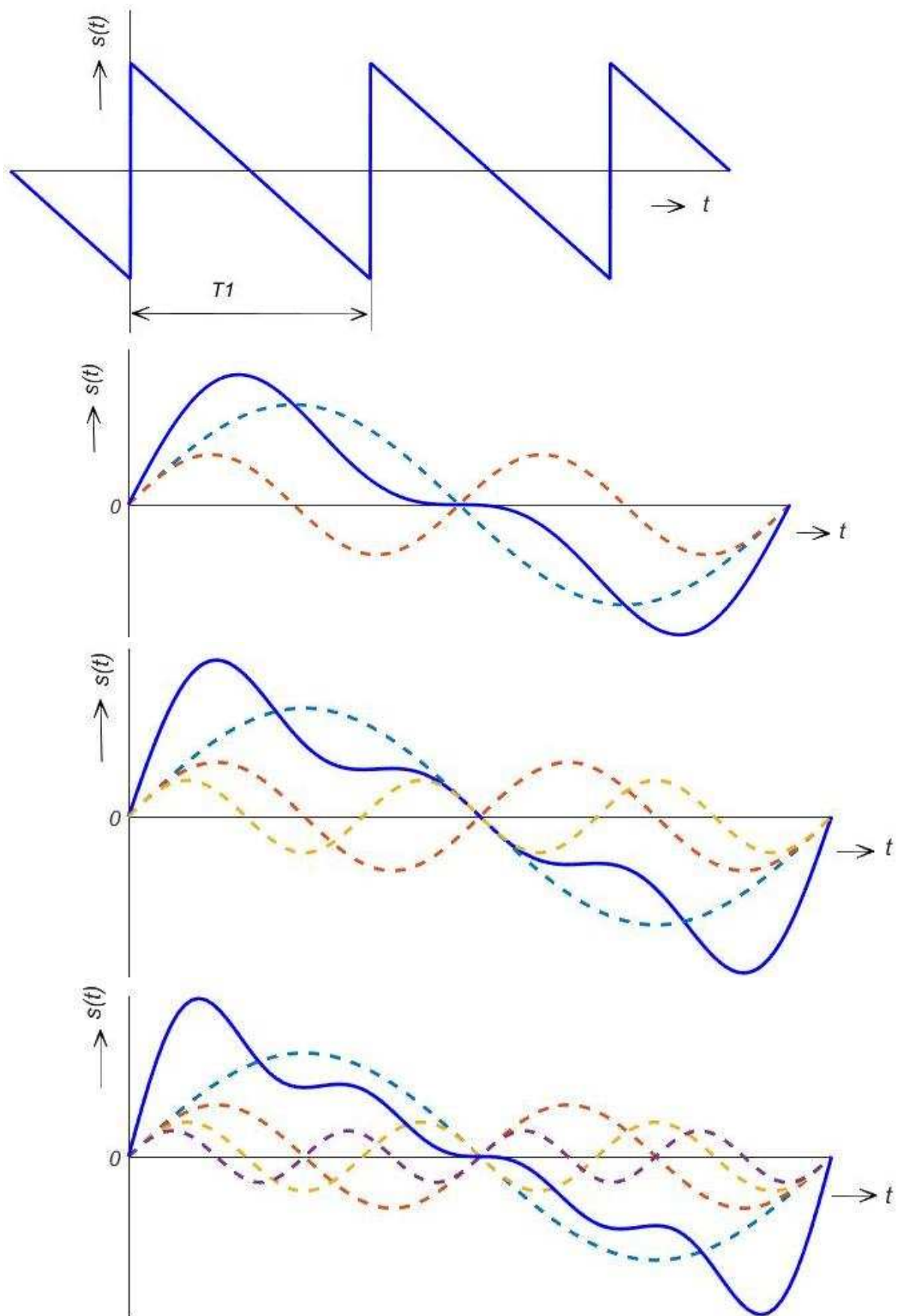
Jean Baptiste Fourier v roce 1823 uvedl metodu, kterou nazval harmonickou analýzou.

Matematická definice Fourierovy řady zní:

Nechť $f(x)$ je periodická funkce s periodou 2π , tj. $f(x + 2\pi) = f(x)$ pro všechna x , a necht' $f(x)$ a $f'(x)$ jsou po částech spojitě funkce v intervalu $\langle -\pi, \pi \rangle$. Označme:

$$a_n = \frac{1}{\pi} \int_{-\pi}^{\pi} f(x) \cos nxdx \quad (n = 0, 1, 2, 3, \dots), \quad (3.1)$$

$$b_n = \frac{1}{\pi} \int_{-\pi}^{\pi} f(x) \sin nxdx \quad (n = 0, 1, 2, 3, \dots). \quad (3.2)$$



Obrázek 2.2: a) Pilovitý periodický signál se základní periodou T_1 ; b) - d) jedna perioda součtu 2, 3 a 4 harmonických složek pilovitého signálu.

Pak v každém bodě x , v němž $f(x)$ je spojitá, platí:

$$\frac{a_0}{2} + \sum_{n=1}^{\infty} (a_n \cos nx + b_n \sin nx) = f(x), \quad (3.3)$$

a v každém bodě x , v němž $f(x)$ není spojitá, platí:

$$\frac{a_0}{2} + \sum_{n=1}^{\infty} (a_n \cos nx + b_n \sin nx) = \frac{f(x+0) + f(x-0)}{2}, \quad (3.4)$$

kde $f(x+0) = f(x_+)$ je limita zprava v bodě x , a $f(x-0) = f(x_-)$ je limita zleva v bodě x . Protože chceme využít i jinou periodu než 2π , zavedeme úhlový kmitočet $\omega_1 = \frac{2\pi}{T_1}$, kde T_1 je nová perioda periodického signálu $s(t)$. Dostaneme tak první tvar Fourierovy řady:

$$s(t) = \frac{a_0}{2} + \sum_{k=1}^{\infty} a_k \cos(k\omega_1 t) + b_k \sin(k\omega_1 t), \quad (3.5)$$

$$a_0 = \frac{2}{T_1} \int_0^{T_1} s(t) dt, \quad (3.6)$$

$$a_k = \frac{2}{T_1} \int_0^{T_1} s(t) \cos(k\omega_1 t) dt, \quad (3.7)$$

$$b_k = \frac{2}{T_1} \int_0^{T_1} s(t) \sin(k\omega_1 t) dt, \quad (3.8)$$

$$k = 1, 2, 3, 4, \dots$$

Použitím vztahu:

$$\cos(\alpha + \beta) = \cos \alpha \cos \beta - \sin \alpha \sin \beta,$$

získáme druhý tvar Fourierovy řady [1]:

$$s(t) = \frac{C_0}{2} + \sum_{k=1}^{\infty} C_k \cos(k\omega_1 t + \varphi_k), \quad (3.9)$$

kde

$$C_k \cos(k\omega_1 t + \varphi_k) = C_k \cos \varphi_k \cos k\omega_1 t - C_k \sin \varphi_k \sin k\omega_1 t, \quad (3.10)$$

$$a_k = C_k \cos \varphi_k, \quad -b_k = C_k \sin \varphi_k,$$

$$C_k = \sqrt{a_k^2 + b_k^2}, \quad \varphi_k = \arctan \frac{-b_k}{a_k}. \quad (3.11)$$

3.3 Komplexní model harmonického signálu

Pro harmonický signál je možné vytvořit model s komplexními čísly:

$$\begin{aligned} s(t) &= C_1 \cos(\omega_1 t + \varphi_1) = C_1 \frac{e^{j(\omega_1 t + \varphi_1)} + e^{-j(\omega_1 t + \varphi_1)}}{2} \\ &= \frac{C_1}{2} e^{j\varphi_1} e^{j\omega_1 t} + \frac{C_1}{2} e^{-j\varphi_1} e^{-j\omega_1 t} \\ &= c_1 e^{j\omega_1 t} + c_{-1} e^{-j\omega_1 t}. \end{aligned} \quad (3.12)$$

Při výpočtu byly použity Eulerovy vztahy:

$$e^{jx} = \cos x + j \sin x, \quad e^{-jx} = \cos x - j \sin x,$$

$$\cos x = \frac{e^{jx} + e^{-jx}}{2}, \quad \sin x = \frac{e^{jx} - e^{-jx}}{2}. \quad (3.13)$$

Koeficienty c_1 a c_{-1} jsou komplexní čísla [1]:

$$\begin{aligned} c_1 &= \frac{C_1}{2} e^{j\varphi_1} = |c_1| e^{j \arg c_1}, & c_{-1} &= \frac{C_1}{2} e^{-j\varphi_1} = |c_{-1}| e^{-j \arg c_{-1}}, \\ |c_1| &= \frac{C_1}{2}, \quad \arg c_1 = \varphi_1, & |c_{-1}| &= \frac{C_1}{2}, \quad \arg c_{-1} = -\varphi_1. \end{aligned} \quad (3.14)$$

Koeficienty c_1 a c_{-1} zobrazené v komplexní rovině jsou koncovými body fázorů, které s měnícím se časem t opisují kružnici o poloměru $\frac{C_1}{2}$. Komplexní model harmonické funkce je prakticky zobrazován v aplikaci popsané v kapitole 6.5.

4 VZORKOVÁNÍ

4.1 Úvod

Převod signálu se spojitým časem na číslicový signál (analogově - číslicový převod) sestává ze tří kroků:

- vzorkování – převod signálu se spojitým časem na signál s diskretním časem (diskretní signál).
- kvantování – cílem je vyjádřit diskretní vzorky pomocí konečné množiny čísel (čísel o konečné délce). Vzniká při něm kvantovací šum.
- kódování – převod kvantovaných vzorků na binární čísla v určitém kódu.

Vzorkování můžeme rozdělit do tří skupin:

- ideální vzorkování,
- vzorkování 1. druhu a
- vzorkování 2. druhu.

Jestliže chceme signál se spojitým časem (analogový signál) zpracovat diskretním nebo číslicovým systémem, musíme použít A/Č (analogově – číslicový) a Č/A (číslícově – analogový) převodník [1].

4.2 Ideální vzorkování

Signál se spojitým časem $s(t)$ je přeměněn na posloupnost veličin, vyjádřených okamžitými hodnotami signálu $s(t)$. Těto posloupnosti říkáme diskretní signál $s[n]$ a postup nazýváme ideální vzorkování nebo diskretizace.

Operaci ideálního vzorkování si lze představit jako součin signálu $s(t)$ s nekonečnou posloupností Diracových impulzů, které jsou od sebe vzdáleny rovnoměrně o vzorkovací interval T :

$$s_{\text{id}}(t) = s(t)s_{\delta}(t) = s(t) \sum_{n=-\infty}^{\infty} \delta(t - nT) = \sum_{n=-\infty}^{\infty} s(nT)\delta(t - nT). \quad (4.1)$$

Spektrální funkce diskretního signálu má podle [1] tvar:

$$S_{\text{id}}(\omega) = \frac{1}{T} \sum_{n=-\infty}^{\infty} S(\omega - k\omega_{\text{vz}}), \quad (4.2)$$

kde $\omega_{\text{vz}} = 2\pi f_{\text{vz}}$ je vzorkovací úhlový kmitočet a $f_{\text{vz}} = \frac{1}{T}$ je vzorkovací frekvence.

Spektrální funkce ideálně vzorkovaného signálu $s(t)$ s periodou T je rovna superpozici rovnoměrně posunutých spekter původního signálu, násobených konstantou $1/T$. Spektrum diskretního signálu je vždy periodické, s periodou rovnou vzorkovacímu kmitočtu.

Aby se nepřekrývala sousední spektra ideálně vzorkovaného signálu, tak musí platit nerovnost:

$$\frac{\pi}{T} > \omega_{\max}, \quad \text{neboli} \quad f_{\text{vz}} > 2f_{\max}, \quad (4.3)$$

Kde f_{\max} je kmitočet složky spektra $S(\omega)$, která má nejvyšší hodnotu kmitočtu. Vztah (4.3) je nazýván vzorkovací poučka (neboli Nyquistův teorém). Pokud vzorkovací poučka neplatí, dojde k překrývání sousedních spekter, a pak už nelze bez chyb obnovit původní spektrální funkci $S(\omega)$, takže také nelze obnovit původní časový průběh signálu $s(t)$.

Nesplnění vzorkovací poučky (4.3) má za následek překrývání sousedních spekter ideálně vzorkovaného signálu $S_{\text{id}}(\omega)$. Tento jev se nazývá aliasing. Pokud chceme překrývání zabránit, tak spojitý signál $s(t)$ musí mít omezeno spektrum $S(\omega)$ antialiasingovým kmitočtovým filtrem typu dolní propusti [1].

5 SIGNÁLY S DISKRÉTNÍM ČASEM

5.1 Úvod

Časová osa signálů s diskretním časem (diskretních signálů) je tvořena konečnou nebo spočetnou množinou okamžiků. V technické praxi je to nejčastěji $\{nT, n \in \mathbb{Z}\}$, nebo její podmnožina, kde \mathbb{Z} je množina celých čísel. Můžeme zavést relativní čas $\frac{nT}{T} = n$, kde T je např. perioda vzorkování. Normovaný čas n je potom bezrozměrná veličina. Diskretní signály budou podle potřeby značeny jako $s[n]$ nebo $s[nT]$. Nebude-li řečeno jinak, budeme dále předpokládat, že diskretní signály mají konečnou absolutní hodnotu pro všechna přípustná n [4, 1].

Grafické zobrazení diskretního signálu je zobrazeno na obrázku 5.1.

V zápise můžeme diskretní signály vyjádřit těmito způsoby:

$$s[n] = \{1,2; 0,7; -1,5; 1; 0,4; -0,3; 1,2; 0,8\}. \quad (5.1)$$

↑

Šipka v zápisu (5.1) značí hodnotu pro $n = 0$. Další možné způsoby zápisu:

$$s[n] = \begin{cases} 1,2 & \text{pro } n = -3, 3 \\ 0,7 & \text{pro } n = -2 \\ -1,5 & \text{pro } n = -1 \\ 1 & \text{pro } n = 0 \\ 0,4 & \text{pro } n = 1 \\ -0,3 & \text{pro } n = 2 \\ 0,8 & \text{pro } n = 4 \\ 0 & \text{pro ostatní } n \end{cases} \quad (5.2)$$

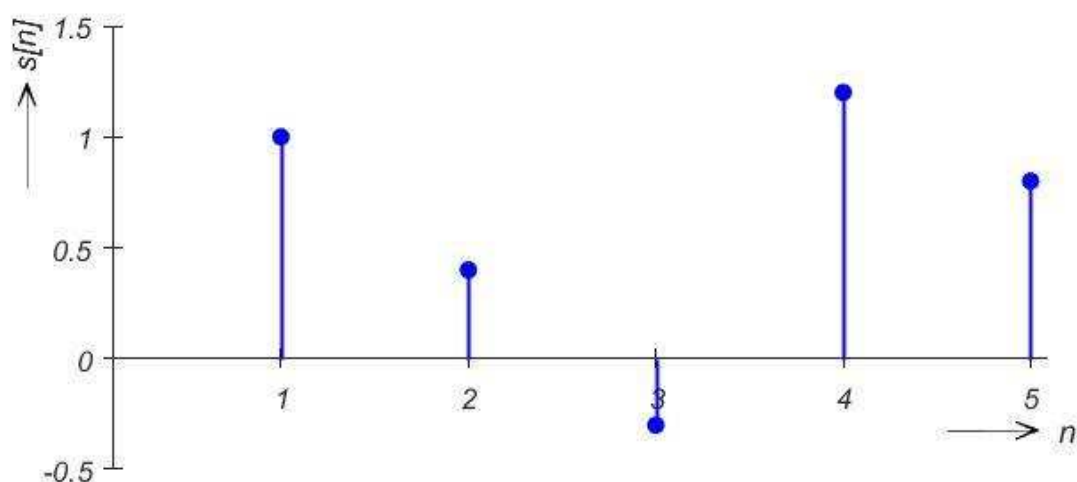
n	-3	-2	-1	0	1	2	3	4
$s[n]$	1,2	0,7	-1,5	1	0,4	-0,3	1,2	0,8

Libovolnou posloupnost $s[n]$ můžeme vyjádřit jako:

$$s[n] = \sum_{m=-\infty}^{\infty} s[m]\delta[n - m], \quad (5.4)$$

kde δ je jednotkový impuls [1]:

$$\begin{aligned} \delta[n] &= 1 && \text{pro } n = 0, \\ \delta[n] &= 0 && \text{pro } n \neq 0. \end{aligned} \quad (5.5)$$



Obrázek 5.1: Příklad grafického zobrazení obecného diskretního signálu.

5.2 Operace s diskretními signály

5.2.1 Součet a součin

Mějme definovány diskretní signály

$$x[n] = \{\dots; x[-2]; x[-1]; x[0]; x[1]; x[2]; \dots\},$$

$$y[n] = \{\dots; y[-2]; y[-1]; y[0]; y[1]; y[2]; \dots\},$$

pak jejich součet je definován jako:

$$x[n] + y[n] = \{\dots; x[-2] + y[-2]; x[-1] + y[-1]; x[0] + y[0]; x[1] + y[1]; x[2] + y[2]; \dots\} \quad (5.6)$$

a jejich součin takto [3]:

$$x[n]y[n] = \{\dots; x[-2]y[-2]; x[-1]y[-1]; x[0]y[0]; x[1]y[1]; x[2]y[2]; \dots\}. \quad (5.7)$$

5.2.2 Modulo

Hodnota $\text{mod}_N(x)$ je zbytek po celočíselném dělení čísla x číslem N , tedy číslo v intervalu $\langle 0, N - 1 \rangle$. Periodickou posloupnost $s_p[n]$ s periodou N pak můžeme k posloupnosti $s[n]$ délky N přiřadit vztahem:

$$s_p[n] = s[\text{mod}_N(n)], \quad (5.8)$$

kde $s[n]$ je konečná posloupnost o délce N [1].

5.2.3 Pravoúhlá okénková posloupnost

Pravoúhlá okénková posloupnost $R_N[n]$ je definována vztahem:

$$\begin{aligned} R_N[n] &= 1 & \text{pro } n \in \langle 0, N-1 \rangle, \\ R_N[n] &= 0 & \text{pro } n \notin \langle 0, N-1 \rangle, \end{aligned} \quad (5.9)$$

Pomocí této posloupnosti můžeme libovolnou posloupnost převést na posloupnost délky N . Můžeme například udělat opačnou operaci k operaci modulo, tj. převedeme periodickou posloupnost $s_p[n]$ s periodou N na jednorázovou posloupnost $s[n]$ délky N tím, že posloupnost $s_p[n]$ vynásobíme pravoúhlo okénkovou posloupností [1]:

$$s[n] = s_p[n] \cdot R_N[n]. \quad (5.10)$$

5.2.4 Konvoluce

Lineární diskrétní konvoluce

Lineární diskrétní konvolucí posloupností $s_1[n]$ a $s_2[n]$ rozumíme posloupnost $z[n]$ danou vztahem:

$$z[n] = s_1[n] * s_2[n] = s_2[n] * s_1[n] = \sum_{m=0}^{N-1} s_1[m]s_2[n-m]. \quad (5.11)$$

Jestliže posloupnost $s_1[n]$ má délku N_1 a posloupnost $s_2[n]$ má délku N_2 , pak výsledná konvoluce $z[n]$ má délku $N = N_1 + N_2 - 1$ [1].

Periodická diskrétní konvoluce

Periodická konvoluce je operace, která se děje mezi dvěma periodickými signály $s_{p1}[n]$ a $s_{p2}[n]$:

$$z_p[n] = \sum_{m=0}^{N-1} s_{p1}[m]s_{p2}[n-m] = s_{p1}[n] * s_{p2}[n] \quad (5.12)$$

Jestliže perioda posloupnosti $s_{p1}[n]$ je N_1 a perioda posloupnosti $s_{p2}[n]$ je N_2 , tak aby výsledek měl nějaký smysl a nenastávalo překrytí, měly by mít obě posloupnosti periodu $N = N_1 + N_2 - 1$. Toho dosáhneme tak, že periodu N_1 a N_2 rozšíříme vzorky s hodnotou 0 na periodu N . Periodické signály jsou získávány pomocí operace modulu N [1].

Kruhová diskrétní konvoluce

Mějme dva diskrétní signály konečné délky: $s_1[n]$ má délku N_1 a $s_2[n]$ má délku N_2 . Kruhová cyklická diskrétní konvoluce je definována vztahem:

$$z[n] = s_1[n] * s_2[n] = R_N[n] \sum_{m=0}^{N-1} s_{p1}[m]s_{p2}[n-m]. \quad (5.13)$$

Posloupnosti $s_{p1}[n]$ a $s_{p2}[n]$ jsou periodické posloupnosti, vytvořené opakováním signálů $s_1[n]$ a $s_2[n]$ s periodou $N = N_1 + N_2 - 1$. Pokud bychom neprovedli prodloužení period popsané výše, došlo by k výrazným chybám překrýváním opakujících se posloupností (časový aliasing). Kruhovým posunutím o m vzorků rozumíme operaci,

která posloupnosti délky N přiřazuje posloupnost [1]:

$$R_N[n]s[\text{mod}_N(n - m)]. \quad (5.14)$$

5.2.5 Korelace

Operace korelace se podobá operaci konvoluce. Korelace ukazuje míru podobnosti průběhu dvou signálů. Korelace $\gamma_{x,y}$ dvou spojitých signálů $x(t)$ a $y(t)$ je rovna [1]:

$$\gamma_{x,y}(\tau) = \int_{-\infty}^{\infty} x(t)y(t - \tau) dt, \quad \tau \in \mathbb{R}. \quad (5.15)$$

Diskrétní korelace se používá v radarové a sonarové technice, v geologii a číslicových komunikacích. Pro jednoduchost uvažujme jednoduchý pulzní (vysílající jednotlivé impulzy) radar s jedinou anténou, která se střídavě připojuje k vysílači a k přijímači. Nejprve se vyšle krátký diskrétní signál a po přepnutí se přijímá signál odražený. Tímto radarem se dá určit například směr, vzdálenost a výška objektu, např. letadla.

Máme tedy dva diskrétní signály $x[n]$ a $y[n]$, které chceme porovnat. Necht' např. u radaru je $x[n]$ vzorkovaný signál vysílače a $y[n]$ je vzorkovaný přijímaný signál, který je zpožděn odrazem od cíle a znehodnocen aditivním šumem, což můžeme popsat vztahem:

$$y[n] = \alpha \cdot x[n - D] + w[n], \quad (5.16)$$

kde α je útlum signálu na trase, D je zpoždění signálu $x[n]$ a $w[n]$ je aditivní šum. Na základě zpoždění D se snažíme určit vzdálenost cíle od radaru. Právě tento parametr můžeme vypočítat pomocí korelace.

Vzájemná (křížová) korelace diskrétních signálů $x[n]$ a $y[n]$ je definována vztahem:

$$r_{xy}[m] = \sum_{n=-\infty}^{\infty} x[n] \cdot y[n - m] = \sum_{n=-\infty}^{\infty} x[n + m] \cdot y[n],$$

$$m = 0, \pm 1, \pm 2, \dots \quad (5.17)$$

Proměnná m vyjadřuje časové posunutí a index xy označuje signály, mezi nimiž je prováděna korelace. Podobně lze definovat:

$$r_{yx}[m] = \sum_{n=-\infty}^{\infty} y[n] \cdot x[n - m] = \sum_{n=-\infty}^{\infty} y[n + m] \cdot x[n],$$

$$m = 0, \pm 1, \pm 2, \dots \quad (5.17)$$

Porovnáním definic (2.42) a (2.43) vidíme, že platí [3]:

$$r_{xy}[m] = r_{yx}[-m]. \quad (5.17)$$

6 REALIZACE NÁZORNÝCH APLIKACÍ V PROGRAMU MATLAB

Následující aplikace byly vytvořeny pomocí programu Matlab 2016a v prostředí Matlab GUIDE a obsahují grafické uživatelské rozhraní (GUI), které poskytuje větší pohodlí a přehlednost při užívání. Rozložení objektů (tabulky, text, grafy) v GUI by mělo studentům poskytnout dostatečnou názornost prováděných operací. Další výhodou těchto aplikací je, že není nutné, aby uživatel znal příkazy pro zadávání příkazů do příkazového okna programu Matlab.

Aplikace je nutné spouštět otevřením souboru s příponou .m a příkazem Run (klávesa F5). Aplikace jsou obsaženy v příloze elektronické verze této práce a na disku, který je přiložen na konci tištěné verze práce.

6.1 Sčítání harmonických složek obdélníkového signálu

6.1.1 Užívání aplikace GUI2squareSig

Aplikace GUI2squareSig slouží ke krokovanému sčítání zadaného počtu harmonických složek obdélníkového signálu. Aplikace takto dokazuje, že se i obdélníkový signál, jako každý periodický signál, skládá z určitého počtu dílčích harmonických složek. Pro přesné vykreslení obdélníkového signálu by bylo nutné sečíst nekonečný počet harmonických, ale cílem této aplikace je, aby z grafu bylo zřejmé, že se s každou přidanou složkou průběh součtu blíží průběhu obdélníkového signálu.

Po spuštění vidíme v horní části okna dva prázdné grafy, ve spodní části ovládání funkce. Uživatel zadá (vepsáním čísla od 1 do 21 nebo tažením myši na prvním posuvníku) počet harmonických složek n , které mají být sečteny a dále čas, po který má funkce čekat mezi vykreslením harmonické a jejím přičtením k výslednému signálu. Kliknutím na tlačítko „Sum in steps“ uživatel spustí postupné vykreslování a sčítání harmonických složek od $k=1$ po $k=2*n-1$, kde n je zadaný počet (lichých) harmonických. V levém grafu se vždy zobrazí k -tá harmonická, která je po čase zvoleném uživatelem zobrazena i v pravém grafu, kde je ihned přičtena ke všem doposud zobrazeným harmonickým. Levý graf zároveň slouží k zobrazení amplitudového spektra. Po každém přičtení další harmonické je uživatel vyzván k potvrzení dalšího kroku kliknutím myši kamkoliv do prostoru okna aplikace. Po sečtení všech harmonických dojde k zobrazení grafu výsledného signálu a jeho modulového spektra.

Je možné zvolit okamžitý výpočet (tlačítko „Sum now“), který vykreslí všechny harmonické s výsledným signálem ihned a poté zobrazí graf signálu a jeho spektra. Tlačítko „Add one harmonic“ slouží k výpočtu a přičtení další harmonické k výslednému signálu. Obě tato tlačítka by měl uživatel použít až při proběhnutí předchozí funkce. V opačném případě sice proběhne zvolená funkce, ale poté pokračuje funkce, která nebyla dokončena.

6.1.2 Vnitřní struktura aplikace GUI2squareSig

Při spuštění aplikace je nejprve odečtena hodnota n z pole `editNHarm` a poté deklarovány proměnné:

- A – počáteční amplituda rovna 100 [jednotek],
- T_s – vzorkovací perioda určuje, po jakém diskretním čase je spočítán další vzorek signálu,
- `xAxisSamples` – „časová osa“, neboli délka intervalu, ve kterém jsou počítány vzorky signálu a vykreslen graf.

Při kroku $T_s = 0.0025$ a délce intervalu `xAxisSamples = 0:Ts:1` dojde k výpočtu 401 vzorku, tj. jedné periody první harmonické složky, která je rovna periodě obdélníkového signálu. Tyto úvodní parametry jsou uloženy do dvou souborů typu `.mat`, které jsou pak načítány následujícími funkcemi.

První posuvník `sliderN` je nastaven na hodnoty od 1 do 21 a pole `editN`, které je s posuvníkem propojeno, je ošetřeno tím, že při editaci projde zapsaná řada znaků podmínkami a jestliže nejsou splněny, je uživatel vyzván k zadání čísel od 1 do 21.

Důležitými částmi aplikace jsou tlačítka, jejichž aktivováním jsou spuštěny samotné funkce pro postupný nebo okamžitý výpočet součtu n harmonických. Funkce jsou aktivovány příkazem `Callback`, tedy při kliknutí na příslušný objekt.

Funkce spuštěná příkazem `Callback` tlačítka „Sum in steps“ (`buttonSumStep_Callback`, jak je uvedeno v kódu funkce) nejdříve načte soubory s údaji o počtu a parametrech harmonických, které zadal uživatel. Dále jsou předem deklarovány prozatím nulové hodnoty posloupností amplitud `amplitudes` a frekvencí `frequencies`, do kterých budou zaznamenávány hodnoty frekvencí a amplitud harmonických, a `harmonics` společně se `sqrSig`, jejichž účelem je ukládání vzorků harmonických, resp. jejich součtu.

Na začátku prvního cyklu `for`, který probíhá n -krát, dochází ke změně $k=2*i-1$ a uložení hodnoty amplitudy a frekvence k -té harmonické. Uvnitř prvního je vložen druhý `for` cyklus, který probíhá 401krát a dochází v něm k výpočtům jednotlivých vzorků n -té složky výsledného signálu.

Tyto hodnoty vzorku jsou uloženy do matice `harmonics`, kde k -tá složka zaujímá n -tý řádek a číslo sloupce vzorku odpovídá počítadlu `t` vloženého cyklu `for`. V rovnici pro výpočet vzorku je pak `t` posunuto o 1 zpět, aby vypočtený vzorek odpovídal průběhu posloupnosti `xAxisSamples`, nikoliv počítadlu `for` cyklu. Pomocí funkce `sin(t)` pak vypočteme požadovaný vzorek. Například pro $i=1$ a $t=1$:

```
harmonics(i,t)=(amplitudes(i)*sin(2*pi*frequencies(i)*Ts*(t-1)))
harmonics(1,1)=(amplitudes(1)*sin(2*pi*(2*1-1)*0.0025*(1-1))
harmonics(1,1)=(100)*sin(0)=0
```

je do matice `harmonics` na pozici `(1,1)` uložena hodnota 0. Tato hodnota pak bude díky posunu `t-1` odpovídat hodnotě vektoru `xAxisSamples(1)=0`, která v grafech určuje souřadnice osy x .

Ihned po vyčíslení je každý vzorek k -té harmonické přičten k výsledné posloupnosti `sqrSig`. Následně je odečtena hodnota druhého posuvníku `wait`, která určuje čas, po který funkce čeká mezi zobrazením grafu průběhu k -té harmonické a zobrazením grafů součtu a amplitudového spektra. Při zobrazení každé harmonické se změní hodnoty jejich parametrů pod grafem. Konkrétně je zde vypsáno pořadí složky k , amplituda A a frekvence f .

Po proběhnutí všech cyklů jsou uloženy hodnoty `harmonics`, `sqrSig`, `frequencies` i `amplitudes`, aby bylo možné je načíst ve funkci `buttonAdd_Callback`, která pak přičte pouze jednu harmonickou a nemusí počítat všechny předchozí znovu.

Funkce s okamžitým výpočtem funguje velmi podobně s tím rozdílem, že průběžně nezobrazuje vypočtené harmonické ani součty, pouze výsledný součet a spektrum.

Funkce přidání jedné harmonické pak provádí pouze jeden cyklus, který probíhá 401krát.

6.2 Konvoluce diskretních signálů

6.2.1 Užívání aplikace GUI1

Aplikace GUI1 provádí lineární diskretní konvoluci dvou vstupních posloupností (signálů) $x[n]$ a $y[n]$ zadaných uživatelem s textovým výpisem a grafickým znázorněním dílčích výpočtů.

Po spuštění aplikace je uživatel vyzván k zadání vstupních posloupností do tabulek. Oba vstupní signály mají délku 5 a n od 0 do 4 a uživatel může zvolit načtení přednastavených hodnot. Tlačítkem „Start convolution in steps“ je zahájen postupný výpočet konvoluce. V úvodu dojde k převrácení signálu $y[n]$ na $y[-n]$ a ve společném grafu také k zobrazení signálů $x[n]$, $y[-n]$ a pomocné (na začátku nulové) posloupnosti, která bude pro každé m výčtem násobků $x[n] \cdot y[m-n]$. Poté uživatel kliknutím (kamkoliv v okně aplikace) zahájí samotný výpočet konvoluce $z[m]$. Průběžně se měnící hodnoty m a n jsou vypisovány pod teoretickým vzorcem výpočtu. Pod nimi najdeme vyčíslení hodnot $x[n] \cdot y[m-n]$ a na závěr dojde k sečtení všech těchto násobků a zapsání výsledku $z[m]$ pro dané m . Po proběhnutí všech m je v novém okně zobrazen graf výsledné konvoluce $z[m]$. Uživatel může zvolit i okamžitý výpočet výsledku pomocí tlačítka „Convolve now“, který je ale vhodné spouštět tehdy, když neprobíhá žádný krokovaný výpočet.

6.2.2 Vnitřní struktura aplikace GUI1

Při spuštění aplikace je volána „úvodní“ funkce (`GUI1_OpeningFcn`) a vykreslen průběh obou (nulových) vstupních posloupností do horních grafů. Totéž se stane, když jsou změněny hodnoty v tabulkách nebo načteny přednastavené hodnoty signálů. Již při změně posloupností jsou k nim přidány nuly, aby bylo možné je lépe zobrazit a poté ve společném grafu vzájemně posouvat a vyznačovat násobky jejich prvků. Takto vznikají vektory `xDisp`, `yDisp`, `nAxis` a později i `zmnDisp`.

Spuštěním funkce `buttonStart_Callback` pomocí stisknutí prvního zmíněného tlačítka dojde k převrácení pořadí prvků signálu z $y[n]$ na $y[-n]$, zobrazení obou

Tabulka 6.1: Tabulková metoda výpočtu diskrétní konvoluce.

m	N	-3	-2	-1	0	1	2	3	4	z[m]
	x[n]				3	1	2			
	y[n]					1	-1			
0	y[-n]		-1	1						0
1	y[1-n]			-1	1					3
2	y[2-n]				-1	1				-2
3	y[3-n]					-1	1			1
4	y[4-n]						-1	1		2
5	y[5-n]							-1	1	0

vstupních signálů, rotaci posloupností xDisp, yDisp a zmnDisp o 90° z řádkového vektoru na sloupcový a k jejich spojení v matici xyizMat. (Tyto kroky jsou zařazeny kvůli zobrazení matice v grafu typu stem, který je schopen zobrazit pouze jeden vektor nebo matici a matici zobrazuje po řádcích.) Po vykreslení prvního, převráceného druhého a poté obou signálů společně přechází funkce k samotnému výpočtu konvoluce.

Postup výpočtu následuje takzvanou tabulkovou metodu diskrétní konvoluce. Ta využívá poznatků z operací periodická a kruhová konvoluce zmíněných v kapitole 5.2.4.

Příklad výpočtu jednoduché konvoluce tabulkovou metodou vidíme v tabulce 6.1. V případě realizace v programu Matlab je jednodušší provést cyklický posun prvků převráceného signálu y[-n] a jejich následný výběr, než je posouvat jako při běžném postupu (viz tabulku 2.1). Posun byl realizován pomocí funkce circshift (zkratka pro „circular shift“ neboli kruhový posun).

Výpočet členů v této aplikaci z[m] je znázorněn v tabulce 6.2, kde yi je převrácený signál y a yics1-5 jsou yi s postupně aplikovanou funkcí circshift. V kódu funkce se proměnné yics1-5 neobjevují, protože operací circshift je při každém průchodu změněn samotný převrácený signál yi:

```
yi=circshift(yi,[1,1]);.
```

První iniciovaný cyklus for se opakuje s m měnícím se od 1 až do výsledné délky konvoluce, tedy 9. V něm je proveden kruhový posun signálu yi a deklarovány proměnné zm a zmnSeq, která souží k výpisu všech násobků $x[n] \cdot y[m-n]$ pro dané m. Do prvního cyklu je vložen druhý, který probíhá od n=1 do n=5. Zde dochází k vymezení diskrétního času n vstupního signálu x a invertovaného posunutého signálu yi, jejichž příslušné hodnoty se budou násobit.

Pokud je například m=1 a n=1, pak signál yi[1] (tedy převrácený signál y který jednou prošel funkcí circshift s posunem 1) je vynásoben hodnotou x[1] s výsledkem $3 \cdot 0 = 0$ (viz tabulka 6.2). Výsledek násobení je uložen jako zmn a přičten k proměnné zm, která je po proběhnutí celého druhého for cyklu uložena pod příslušným m do výsledné

Tabulka 6.2: Příklad výpočtu diskretní konvoluce pomocí aplikace GUI1.

m	n	1	2	3	$z[m]$
	$x[n]$	3	1	2	
	$y[n]$	0	1	-1	
	$y_i[n]$	-1	1	0	
1	$y_{ics1}[n]$	0	-1	1	0
2	$y_{ics2}[n]$	1	0	-1	3
3	$y_{ics3}[n]$	-1	1	0	-2
4	$y_{ics4}[n]$	0	-1	1	1
5	$y_{ics5}[n]$	1	0	-1	-2
6	$y_{ics6}[n]$	-1	1	0	0

konvoluce $z[m]$. Pokud je ale například $m=5$ a $n=2$, nejsou splněny podmínky if a k násobení nedojde. Výsledek násobení zmn je také uložen do posloupností $zmnDisp$ (pro zobrazení v grafu) a $zmnSeq$ (pro textový výpis). Kliknutím pak uživatel přejde k uložení prvku z_m do výsledné konvoluce. Poté je vynulován vektor $zmnDisp$ a signál $y[m-n]$ posunut o $+1$.

6.3 Korelace diskretních signálů

6.3.1 Užívání aplikace GUI3correlation

Aplikace GUI3correlation slouží k výpočtu korelace dvou signálů délky 5. Pracuje na obdobném principu i grafickém rozhraní jako GUI1 (konvoluce).

Po spuštění má uživatel zadat libovolné hodnoty $x[n]$ a $y[n]$. Poté volbou „Start correlation in steps“ zahájí postupný výpočet korelace. Ta začíná od $m=-4$, proto ve společném grafu vidíme signál $y[n]$ nezměněn a signál $x[m+n]$ posunutý o -4 . Následně dochází k postupnému násobení $x[m+n] \cdot y[n]$ a souběžnému vyznačování těchto násobků ve společném grafu s názvem Correlation process. Po proběhnutí všech n , pro která je toto násobení možné, je m zvýšeno o 1 a taktéž i signál $x[m+n]$ posunut o $+1$. Na závěr dojde k zobrazení grafu výsledné korelace. Opět je možné zvolit okamžitý výpočet výsledku tlačítkem „Correlate now“.

6.3.2 Vnitřní struktura aplikace GUI3correlation

Při postupném výpočtu probíhá první for cyklus m -krát a druhý n -krát. Při každém vnitřním cyklu je výsledek násobení $rmn=x[m+n] \cdot y[n]$ přičten k součtu rm a přidán k vektorům $rmnDisp$ a $rmnSeq$. Na konci vnitřního cyklu je součet rm přidán k výsledné korelaci $correlation[m]$.

6.4 Simulace radaru

6.4.1 Užívání aplikace GUI4corrRadar

Aplikace využívá operace korelace k výpočtu vzdálenosti imaginárního objektu (například letadla). Vyslaný signál se odráží od objektu a vrací zpět k přijímači. Poté je provedena korelace vyslaného a přijatého signálu a z ní určeno zpoždění, ze kterého určíme vzdálenost objektu.

Po spuštění je uživatel vyzván k zadání vysílaného signálu délky 5 (může zvolit možnost načtení přednastavených hodnot „Load pre-set signal“), vzdálenosti objektu (v rozpětí od 7,5 do 150 km), od kterého se bude signál odrážet a vracet zpět, a odstupu signálu od šumu (signal-to-noise ratio, SNR), který je při zadaných hodnotách nad 200 dB velmi vysoký, proto je hodnota změněna na „Inf.“, tedy nekonečno a není k němu dále přidáván žádný šum. Všechny parametry mají své přednastavené hodnoty, takže uživatel nemusí žádný z nich měnit. Vysílaný signál je zobrazen v levém horním grafu a při každé změně je automaticky aktualizován a zobrazen jeho průběh v tomto grafu.

Poté je nutné zvolený signál vyslat tlačítkem „Transmit and receive signal“. Signál je po malé chvíli se zpožděním a přidaným šumem opět přijat a zobrazen v pravém horním grafu a uživatel může provést korelaci těchto signálů, kterou spustí tlačítkem „Correlate signals“. Poté je ihned provedena korelace, jejíž část obsahující očekávané maximum je zobrazena ve spodním velkém grafu.

Ve spodní části okna pak vidíme zpětný výpočet vzdálenosti zaznamenaného objektu. Nejprve funkce vypíše nalezené maximum korelace $r[n]$ včetně n , kde se toto maximum nachází. Přijatý signál je zpožděn právě o tolik vzorků, o kolik je posunuto n maxima korelace od $n=0$. (Toto je možné ověřit jak výpočtem, tak pomocí předchozí aplikace GUI3correlation pomocí zadání např. jednotkového impulzu v rozdílných časech n .) Na druhém řádku se nachází výpočet zpoždění na základě vzorkovací periody, která je zvolena na 1 μ s, tedy $f_{vz}=1$ MHz. Na posledním řádku je vypsán výpočet vzdálenosti d . V případě, že se vypočtená vzdálenost rovná vzdálenosti zadané (s tolerancí 50 m kvůli zaokrouhlování), je objekt detekován správně. V opačném případě je šum příliš vysoký a odražený vyslaný signál nebylo možné v přijatém signálu rozpoznat.

6.4.2 Vnitřní struktura aplikace GUI4corrRadar

Při spuštění aplikace jsou odečteny přednastavené hodnoty signálu s , vzdálenost objektu $obDist$ a SNR, v kódu uložený jako $sigNoiseRat$. Signál s je jako $sSent$ zobrazen v prvním grafu.

Všechny tři hodnoty získané z objektů GUI jsou uloženy do pomocných souborů, které jsou při každé změně hodnot provedené uživatelem automaticky aktualizovány.

Po kliknutí na tlačítko „Transmit and receive signal“ je spuštěna funkce `buttonSendRec_Callback`, která nejdříve načte soubory s uloženými parametry, dále vypočítá zpoždění ve vzorcích `delay` podle zadané vzdálenosti `obDist` a vytvoří přijímaný signál `sRec` s počtem nul odpovídajícím zpoždění `delay` zařazeným před signál s a zbytkem nul do 1000 (kvůli rozsahu zpoždění) přidaným na konec. K tomuto signálu je pak přidán šum se zadaným odstupem `sigNoiseRat`. Jsou stanoveny minima a maxima

osy y a v grafech zobrazen nejdříve první signál sSent a po vteřinové pauze druhý sRec. Nakonec jsou zpoždění, vyslaný i přijatý signál a uloženy do dalších pomocných souborů.

Funkce buttonCorr_Callback spuštěná tlačítkem „Transmit and receive signal“ nejprve načte soubory s proměnnými sSent, sRec, delay a sigNoiseRat. Poté provede korelaci vyslaného a přijatého signálu sCorr, nalezne čas jejího maxima nMax a provede její posun zpět o dvojnásobek hodnoty zpoždění. Matlab sice nepracuje se záporným časem, ale přesto je možné v grafu zobrazit záporné hodnoty na ose x a to díky vytvoření pomocné posloupnosti nAxis, jejíž záporné hodnoty se budou zobrazovat na ose x. Kladné hodnoty obsahuje pouze pro doplnění na shodnou délku s sCorr, která je sice korelací signálů o délce 5 a 1005, ale Matlab doplnil první signál na délku druhého, a proto má výsledek délku 2009 vzorků. Následuje zobrazení výsledné korelace sCorr v grafu a výpočet vzdálenosti objektu d. Hodnota vzdálenosti d je vypočtena z času nMax, ve kterém se nachází maximum sCorr, vyjde tedy nesprávně, nachází-li se maximum v nesprávném čase vlivem vysokého zašumění signálu. Po porovnání hodnoty teoretického (zadaného) s hodnotou vypočteného zpoždění je vypsána hláška, která informuje uživatele o správnosti či nesprávnosti výsledku.

6.5 Komplexní model harmonického signálu

6.5.1 Užívání aplikace GUI6complexModel

Aplikace slouží k výpočtu komplexního modelu harmonického signálu, který je popsán v kapitole 3.3, včetně vykreslení průběhu signálu a zobrazení spektra modulů a fází koeficientů c_1 a c_{-1} komplexního modelu.

Po spuštění aplikace je uživatel vyzván k zadání vlastních požadovaných vstupních parametrů – amplitudy C , frekvence f a počáteční fáze ϕ , může ale ponechat přednastavené hodnoty. Výpočet komplexního modelu je zahájen tlačítkem „Calculate complex model“, které v levém grafu zobrazí v komplexní rovině c_1 a c_{-1} a jejich součet, tedy výsledný harmonický signál v nulovém čase $s(0)$. Zároveň jsou vypsány hodnoty c_1 a c_{-1} , jejich argumenty a moduly. Poté může uživatel zvolit možnost „Animation of harmonic signal“, která spustí animaci rotace fázorů po kružnici o poloměru $|c_1| = |c_{-1}| = \frac{C}{2}$, jejichž součet znázorňující okamžitou hodnotu harmonického signálu kmitá po reálné ose podle funkce $s(t) = \cos(\omega t + \phi)$. Další možností pro uživatele je volba zobrazení grafu spektra argumentů či modulů c_1 a c_{-1} .

6.5.2 Vnitřní struktura aplikace GUI6complexModel

Při spuštění funkce je deklarována posloupnost 101 vzorku časové osy t a odečteny hodnoty editovatelných polí, které jsou po úpravě uloženy jako $C1$, $w1$ a $p1$.

Při každé změně hodnot v těchto polích jsou uloženy hodnoty automaticky aktualizovány díky funkcím jako editC1_Callback ad., které při každé změně textu odečtou jeho hodnotu a uloží do pomocného souboru.

Funkce buttonCalc_Callback nejprve načte uložené hodnoty vektoru času t , amplitudy $C1$, úhlového kmitočtu $w1$ a počáteční fáze $p1$. Z těchto parametrů jsou pak počítány koeficienty c_1 a c_{-1} , které jsou pak zaokrouhleny na 3 desetinná místa

z důvodu větší přehlednosti při výpisu. Dále jsou vypočítány body kružnice `circle`, na které leží a po které se v animaci budou posouvat koncové body fázorů, a vypsány hodnoty koeficientů, jejich argumenty a moduly. Je také deklarována úsečka, která na reálné ose levého grafu znázorňuje kladnou a zápornou amplitudu kmitání součtu obou fázorů. Pomocné hodnoty $s(1,1)$ a $s(1,2)$, které zahrnují koeficienty násobeny $e^{j\omega_1 t}$ a $e^{-j\omega_1 t}$ (podle vzorce 3.12), je pak možné jednoduše zachytit a znázornit v čase a prostým součtem získat okamžitou hodnotu harmonického signálu. Součet, tedy harmonický signál, je uložen do třetího sloupce matice $s(1,3)$ a čtvrtý sloupec je vyhrazen pro čas, který je kvůli správnému vykreslení při animaci počítán pro každý vzorek znovu. V prvním grafu jsou zobrazeny koeficienty `c1` a `c_1`, jejich součet, kružnice `circle`, fázory, spojnice mezi koncovými body fázorů a jejich součtem a úsečka, po které kmitá součet $s(:,3)$.

Funkce `buttonAnim_Callback` také začíná načtením vstupních hodnot, výpočtem `c1`, `c_1`, kružnice `circle` a výpisem hodnot pro uživatele. Poté ale začíná probíhat cyklus `for` s počítadlem `k` od 1 do 101, ve kterém jsou nejprve vypočteny hodnoty příslušných pozic matice `s`, z nich fázory a spojnice bodů a poté v levém grafu zobrazeny stejné hodnoty jako v předchozí funkci s tím rozdílem, že se v každém cyklu mění. Navíc se v pravém grafu postupně vykresluje okamžitá hodnota koncového bodu součtu obou fázorů v závislosti na čase, tedy harmonická funkce. Na konci cyklu je zařazena pauza délky 0,05 sekundy, která vytváří plynulou postupnou animaci.

Dalšími funkcemi je zobrazení modulů a argumentů koeficientů `c1` a `c_1`. Zde dochází k deklaraci bodů na ose `x` a jejich příslušných hodnot a následně k jejich zobrazení v grafech s rozměry přizpůsobenými hodnotám zmíněných parametrů.

6.6 Vzorkování

6.6.1 Užívání aplikace GUI8sampling

Tato aplikace slouží k demonstraci vzorkování spojitého signálu. Číslicový systém a tedy ani program Matlab nedokáže generovat spojitý signál. Program tedy v podstatě provádí pouze převzorkování signálu, nám ale pro názornost postačí zvolit dostatečně vysokou vzorkovací frekvenci při generování signálu, který při zobrazení v grafu pro reprezentaci spojitého signálu postačí. V této aplikaci se bude jednat o vzorkování signálů o frekvenci od 0 do cca 20 Hz.

Po spuštění aplikace je uživatel vyzván k zadání frekvence a tvaru generovaného „spojitého“ signálu. Může vybírat ze signálu tvaru sinusoidy, trojúhelníkového, pilovitého nebo obdélníkového. Uživatel může ponechat přednastavené hodnoty (signál tvaru funkce `sinus` o frekvenci 2 Hz) a poté vygeneruje signál kliknutím na tlačítko „Generate signal“. V prvním grafu dojde k zobrazení sekundového úseku zvoleného signálu.

Poté uživatel zadá (nebo ponechá přednastavenou hodnotu) vzorkovací frekvenci a spustí vzorkování tlačítkem „Sample signal“. Vzorkování je okamžitě provedeno a výsledný diskrétní signál je zobrazen v druhém grafu.

Zbývá poslední krok – obnovení vzorkovaného signálu. To je spuštěno tlačítkem „Reconstruct signal“ a zobrazeno ve třetím grafu.

Protože je signál pouze převzorkován a nikoliv plně obnoven do spojitého tvaru, není započítán vliv jeho spektra. Můžeme ale jednoznačně sledovat vliv vzorkovací frekvence na míru podobnosti původního a obnoveného signálu. Kromě funkce sinus jsou totiž všechny signály složeny z nekonečného množství harmonických složek, a tedy nebude nikdy možné je obnovit do přesné původní podoby. Zvláště u signálů s ostrými hranami (pilovitý a obdélníkový) můžeme pozorovat překmity (tzv. Gibbsův jev), které jsou způsobeny omezeným počtem harmonických složek.

Zajímavostí je i to, že i když signál tvaru sinus vzorkujeme podle poučky $f_{vz} > 2 \cdot f_{max}$ (například se zadanou základní frekvencí $f_1 = 2$ [Hz] a vzorkovací frekvencí $f_s = 5$ [Hz]), nedochází k příliš přesnému obnovení právě kvůli absenci vlivu spektra do výpočtu obnovovaného signálu.

6.6.2 Vnitřní struktura aplikace GUI8sampling

Při spuštění je vytvořen vektor časové osy $tAxis = 0:Ts0:1-Ts0$ (kde $Ts0 = 0,0001$ je vzorkovací perioda původního signálu) a odečteny hodnoty polí pro zadávání frekvence signálu f_1 a vzorkovací frekvence f_s . Tyto hodnoty jsou uloženy do pomocných souborů, které budou načítány příslušnými funkcemi. Kdykoliv jsou uživatelem změněny hodnoty frekvencí, automaticky jsou aktualizovány v pomocných souborech.

Kliknutím na první tlačítko je spuštěna funkce `buttonGen_Callback`. Ta nejdříve načte soubory obsahující vektor $tAxis$ a frekvenci generovaného signálu f_1 . Poté odečte hodnotu rozbalovacího menu `menuSig` a podle ní je proveden výpočet signálu `sig` daného tvaru. Signál je zobrazen ve spojitém grafu a uložen do dalšího souboru.

Druhé tlačítko spustí funkci `buttonSample_Callback`, která načte soubory s časovou osou $tAxis$, zadanou vzorkovací frekvencí f_s a signálem `sig` vygenerovaným v předchozí funkci. Dále provede deklaraci nulových vektorů `sigSam` a `nAxis` o délkách shodných s hodnotou f_s . Ve for cyklu, který probíhá pro n o 1 do f_s je pak vypočten čas t (pořadové číslo hodnoty z vektoru `sig`), ve kterém bude odebrán vzorek signálu `sig` a do `sigSam(n)` je uložena hodnota signálu `sig(t)`. Vzorkovaný signál `sigSam` je zobrazen v bodovém grafu a uložen.

V poslední funkci `buttonReconstruct_Callback` je provedeno převzorkování signálu `sigSam` z frekvence f_s na původní vzorkovací frekvenci $1/Ts0$, tedy 10000 Hz.

7 ZÁVĚR

V bakalářské práci byl proveden výtah vybraných kapitol teorie spojitéch a diskrétních signálů a poznatky z nich byly aplikovány na vytvoření aplikací v prostředí GUIDE programu Matlab. Tyto aplikace s grafickým uživatelským rozhraním je možné využít ve výuce předmětu CASS i jeho českých ekvivalentů (BASS, KASS).

Teorii periodických signálů, konkrétně obdélníkového signálu, demonstruje aplikace GUI2squareSig, která v postupných krocích sčítá zadaný počet harmonických složek a současně zobrazuje v grafech aktuální přičítanou harmonickou, součet všech vygenerovaných harmonických a jejich spektrum.

Aplikace GUI1 provádí diskrétní lineární konvoluci dvou zadaných signálů o délce 5, zatímco probíhá výpis dílčích kroků výpočtu, který následuje tzv. tabulkovou metodu a využívá poznatky periodické a kruhové konvoluce. Zároveň jsou všechny kroky i výsledná posloupnost zobrazovány v grafech.

V aplikaci GUI3correlation sledujeme výpočet korelace dvou signálů délky 5. Při výpočtu dochází k výpisu dílčích kroků a jejich zobrazení v grafech, stejně jako k zobrazení grafu výsledku.

Aplikace GUI4corrRadar je simulací jednoduchého radaru, který vysílá a přijímá signál odražený od objektu, jehož vzdálenost je možné měnit. Aplikace pak provede korelaci vyslaného a přijatého signálu a z jejího maxima výpočtem ověří vzdálenost daného objektu.

GUI6complexModel slouží k výpočtu a zobrazení komplexního modelu harmonického signálu s parametry zadanými uživatelem. Aplikace pak nejen vypočte komplexní koeficienty a jejich moduly a argumenty, ale tyto hodnoty zobrazí v grafech a animací rotace fázorů v komplexní rovině vykreslí danou harmonickou funkci v čase.

Poslední z vytvořených aplikací GUI8sampling pak ukazuje vzorkování a obnovení signálu tvaru sinusoidy, signálu trojúhelníkového, pilovitého nebo obdélníkového.

Vytvořené aplikace vysvětlují většinou základní operace se signály. Pochopení těchto hlavních operací je ale potřebným základem nejen předmětu CASS. Čím rychleji si je tedy studenti osvojí, tím dříve mohou postupovat k pokročilejším tématům.

LITERATURA

- [1] SMĚKAL, Zdeněk. *Analýza signálů a soustav - BASS: elektronická skripta pro bakalářská studia* [online]. Ústav telekomunikací, FEEC, VUT Brno, 2012 [cit. 2017-04-01]. Dostupné z: <https://moodle.vutbr.cz/>
- [2] KOKTAVÝ, Pavel. *Úvod do studia fyziky: přednáška předmětu Fyzika pro audio inženýrství* [online]. FEKT, VUT Brno, 2014 [cit. 2017-04-13]. Dostupné z: <https://moodle.vutbr.cz/>
- [3] MIŠUREC, Jiří a Zdeněk SMĚKAL. *Číslicové zpracování signálů (BCZS): elektronická skripta pro bakalářská studia* [online]. Ústav telekomunikací, FEEC, VUT Brno, 2011 [cit. 2017-04-03]. Dostupné z: <https://moodle.vutbr.cz/>
- [4] SMĚKAL, Zdeněk. *Číslicové zpracování řeči (MZPR): elektronická skripta pro magisterská studia* [online]. Ústav telekomunikací, FEEC, VUT Brno, 2010 [cit. 2017-04-07]. Dostupné z: <https://moodle.vutbr.cz/>
- [5] SMĚKAL, Zdeněk. *Číslicové zpracování signálů (MCSI): elektronická skripta pro magisterská studia* [online]. Ústav telekomunikací, FEEC, VUT Brno, 2012 [cit. 2017-04-03]. Dostupné z: <https://moodle.vutbr.cz/>
- [6] GRIFFITHS, David F. *An Introduction to Matlab Version 3.1* [online]. Dundee, UK, 2015 [cit. 2017-04-07]. Dostupné z: <http://www.maths.dundee.ac.uk/>
- [7] HERINGOVÁ, Blanka a Petr HORA. *MATLAB Díl I. – Práce s programem* [online]. Plzeň, 1995 [cit. 2017-04-07]. Dostupné z: <http://www.cdm.cas.cz/>
- [8] ZVONČÁK, Vojtěch. *Vizualizace biomedicínských dat v prostředí MATLAB: diplomová práce* [online]. Brno: Vysoké učení technické v Brně. Fakulta elektrotechniky a komunikačních technologií. 2016 [cit. 2017-04-07]. Dostupné z: <https://dspace.vutbr.cz/>
- [9] GALÁŽ, Zoltán. *Analýza ručně psaného projevu u pacientů postižených neurologickými onemocněními* [online]. Ústav telekomunikací, FEEC, VUT Brno, 2013 [cit. 2017-04-07]. Dostupné z: <https://dspace.vutbr.cz/>
- [10] MEKYSKA, Jiří. *Analýza řečových promluv pro IT diagnostiku neurologických onemocnění: dizertační práce* [online]. Ústav telekomunikací, FEEC, VUT Brno, 2014 [cit. 2017-04-07]. Dostupné z: <https://dspace.vutbr.cz/>