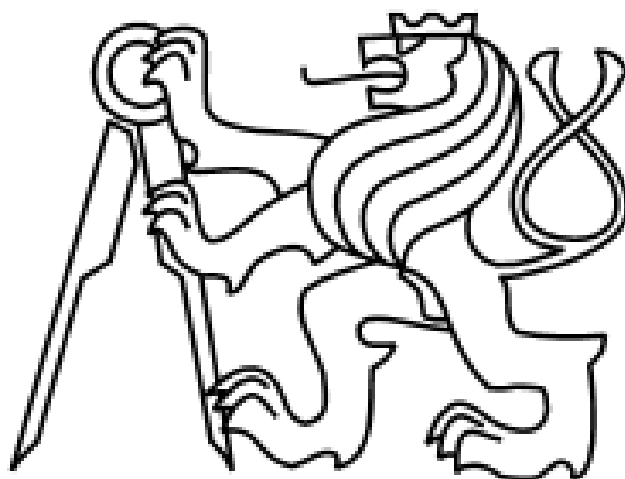


ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE  
Fakulta strojní – Ústav přístrojové a řídicí techniky



Diplomová práce

**2D diagnostika profilu svazku nabitých částic  
získaných pomocí částicových urychlovačů**

**Bc. Dušan Poklop**

**2016**

## ZADÁNÍ DIPLOMOVÉ PRÁCE

pro:

**Bc. Dušana POKLOPA**

obor: Přístrojová a řídicí technika

Název tématu:

**2D diagnostika profilu svazků nabitých částic získaných pomocí částicových urychlovačů**

Název anglicky:

2D diagnostics of a charged particle beams profile obtained from particle accelerators

### *Zásady pro zpracování*

Práce se zabývá návrhem algoritmu pro diagnostiku profilu svazků na urychlovačích Ústavu jaderné fyziky AV ČR (mikrotron, popřípadě cyklotron). Svazek urychlených částic je na výstupním okně elektronovodu či iontovodu urychlovače snímán rotačním drátkovým detektorem. Získaná data budou navrženým algoritmem zpracována jako 2D signál profilu svazku, který bude v barevné škále zobrazen na monitoru počítače. Simulace algoritmu bude nejdříve ověřena v prostředí MATLAB – zejména za účelem výběru vhodného filtru pro zpětnou filtrovanou projekci (zpětná Radonova transformace) a ověření funkčnosti algoritmu. Výsledný software bude realizován na systému Arduino. Součástí práce je i navrhnout řízení krokového motoru, jenž ovládá definované natáčení drátkového detektoru.

Cíle práce:

- Seznámit se s Radonovou transformací.
- Seznámit se se systémem Arduino.

- Provést simulace zobrazení rekonstruovaného obrazu v prostředí MATLAB.
- Navrhnout algoritmus pro rekonstrukci obrazu.
- Navrhnout řízení krokového motoru pro rotaci drátkového detektoru.
- Otestovat navržený systém na mikrotronu MT25.

*Rozsah grafických prací:* dle pokynů vedoucího DP

*Rozsah práce:* min. 50 stran + přílohy


*Seznam odborné literatury:*

- [1] Krist, P., Bíla, J. A mathematical model of the MT25 microtron. JINST 6, T10005, 2011
- [2] Krist, P. Řídicí systém mikrotronu – kruhového relativistického urychlovače elektronů. Praha: Ústav přístrojové a řídicí techniky, ČVUT Praha, 2012. 115 stran. Disertační práce, školitel Bíla, J.
- [3] Ullmann, V. Jaderná fyzika a fyzika ionizujícího záření – kruhové urychlovače [online], 2008, [18.10.2016]. Dostupné z <http://astronuklfyzika.cz/JadRadFyzika5.htm#KruhUrychlovace>
- [4] Walek, P., Lamoš, M., Jan, J. Analýza biomedicínských obrazů - Počítačová cvičení. Brno: Ústav biomedicínského inženýrství, VUT Brno, 2013. 138 stran.

*Vedoucí diplomové práce:* Mgr. Ing. Jakub Jura, Ph.D.

*Datum zadání diplomové práce:* 26. 10. 2016

*Termín odevzdání diplomové práce:* 6. 1. 2017

  
doc. Ing. Jan Chyský, CSc.  
*vedoucí ústavu*



  
prof. Ing. Michael Valásek, DrSc.  
*děkan fakulty*

*V Praze dne:* 22. 10. 2016

### **Poděkování**

Děkuji Mgr. Ing. Jakubu Jurovi, Ph.D. za odborné a organizační vedení při zpracování této práce. Také Ing. Pavlu Kristovi, Ph.D. za jeho odborné rady, technickou pomoc, čas a ochotu při řešení diplomové práce.

### **Prohlášení**

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně s tím, že její výsledky mohou být dále použity podle uvážení vedoucího diplomové práce jako jejího spoluautora. Souhlasím také s případnou publikací výsledků diplomové práce nebo její podstatné části, pokud budu uveden jako její spoluautor.

V Praze dne .....

.....

podpis

**Anotace**

Práce se zabývá návrhem systému pro 2D diagnostiku příčného profilu svazku na mikrotronu MT25 – kruhovém relativistickém urychlovači elektronů. Svazek urychlených elektronů je na výstupním okně elektronvodu urychlovače snímán drátkovým detektorem. Získaná data jsou navrženým systémem zpracována a pomocí filtrované zpětné projekce zobrazena jakožto 2D obrázek profilu svazku. Simulace zobrazování byla prováděna v prostředí GNU Octave, za účelem vybrat vhodný filtr pro filtrovanou zpětnou projekci. Navržený systém byl realizován na platformě Arduina a osobního počítače. Systém umožňuje lepší popis svazku, který zkvalitňuje ozařovací služby mikrotronové laboratoře. Systém může být využit k diagnostice svazků nabitých částic také v dalších laboratořích

**Klíčová slova:** Radonova transformace, filtrovaná zpětná projekce, rekonstrukce obrazu, Arduino, Mikrotron

**Abstract**

This work presents the design of a 2D diagnostic beam profile system for the MT25 microtron – a cyclic electron accelerator. The beam of the accelerated electrons at the output window of the beam line is scanned by the wire detector. These data are processed by a system which create image of the beam profile. Display simulation was designed in GNU Octave, allowing to choose the suitable filter for filtered back projection. The designed system was realized using the Arduino. The system allows better description of the beam, which improves irradiation service of the Microtron laboratory. The system is useful to describe beams of charged particles in other laboratories.

**Klíčová slova:** Radon transformation, Filtered Back Projection, image reconstruction, Arduino, Microtron

## Obsah

Obsah .....	3
Seznam použitých symbolů.....	4
1 Úvod.....	6
1.1 Mikrotron.....	6
1.2 Krokový motor.....	9
1.3 Drátkový detektor .....	11
1.4 Převodník pro drátkový detektor .....	12
1.5 Arduino .....	12
2 Teoretický základ .....	15
2.1 Radonova transformace .....	15
2.2 Fourierova rekonstrukce.....	18
2.3 Prostá zpětná projekce.....	21
2.4 Filtrovaná zpětná projekce .....	22
3 Simulace rekonstrukce obrazu.....	26
3.1 Obecné simulace rekonstrukce obrazu.....	26
3.1.1 Tvorba sinogramu .....	26
3.1.2 Rekonstrukce obrazu.....	28
3.2 Reálný systém .....	30
3.3 Simulace reálného systému .....	31
3.4 Výběr filtru .....	37
4 Realizace systému .....	41
4.1 Řízení krokového motoru.....	41
4.2 Sériová komunikace.....	42
4.3 Program.....	43
5 Testování navrženého systému .....	44
5.1 Výstupy měření .....	45
5.2 Diskuze.....	46
6 Závěr .....	48
7 Použitá literatura.....	50
Seznam obrázků.....	51
Seznam tabulek.....	52
Seznam příloh.....	53

## Seznam použitých symbolů

$F_e$ [N]	elektrická urychlující síla
$F_m$ [N]	Lorenzova síla
$E$ [V/m]	intenzita elektrického pole
$W_e$ [eV]	kinetická energie elektronů
$e$ [C]	náboj elektronu
$v$ [m/s]	rychlost
$c$ [m/s]	rychlost světla
$m$ [kg]	hmotnost
$B$ [T]	magnetická indukce
$R$ [m]	poloměr zakřivení
$\vartheta$ [°]	úhel snímané projekce
$\sigma$ [-]	směrodatná odchylka
$\omega$ [rad/s]	úhlová frekvence
$P$	paprskový integrál
$p$	směr paprskového integrálu
$t$	osa projekcí
$f$	předmětová funkce
$h$	impulzní funkce
$F$	funkce ve Fourierově prostoru/frekvenční oblasti
$F$	přenosová charakteristika
$R$	Radonův operátor
$J$	Jakobián
$x$	souřadnice
$y$	souřadnice
$s$	dráha po které se integruje
CT	rentgenová výpočetní tomografie
MRI	magnetická rezonance
PET	pozitronová emisní tomografie
SPECT	jednofotonová emisní výpočetní tomografie

1DFT	jednodimenzionální Fourierova transformace (One-Dimensional Fourier Transform)
2DFT	dvou dimenzionální Fourierova transformace (Two-Dimensional Fourier Transform)
IFFT	inverzní rychlá Fourierova transformace (Invert Fast Fourier Transform)
SBP	prostá zpětná projekce (Simple Back Projection)
FBP	filtrovaná zpětná projekce (Filtered Back Projection)
FIR	filtr s konečnou impulsní odezvou (Finite Impulse Response)
TM <sub>010</sub>	transverzální magnetická vlna
DC	stejnoseměrný (proud, napětí)
USB	univerzální sériový port (Universal Serial Bus)
COM	rozhraní sériového portu
TX	odesílací kanál komunikace (Transmitted)
RX	přijímací kanál komunikace (Received)
I/O	vstupy/výstupy (Input/Output)
PWM	pulzně šířková modulace
SRAM	statická paměť (Static Random Access Memory)
EEPROM	elektricky mazatelná paměť (Electrically Erasable Programmable Read-Only Memory)
UART	univerzální asynchronní přijímač/vysílač (Universal Asynchronous Receiver/Transmitter)
ICSP	způsob programování kontrolérů (In-Circuit Serial Programming)
ZIP	archivovaný soubor
8-N-1	zkrácený zápis pro konfiguraci sériového portu
A/D	analogově digitální převodník
Bd	Baud (jednotka komunikační rychlosti, počet bitů za sekundu)



# 1 Úvod

Cílem této diplomové práce je navrhnout systém pro diagnostiku příčného profilu svazku na výstupu relativistického urychlovače elektronů mikrotron MT25, kde na základě principu zpětné Radonovy transformace bude vytvořen algoritmus, který získaná data z drátkového detektoru zpracuje do grafické podoby. Pro optimální návrh algoritmu bude provedena jeho simulace v prostředí Matlab či Octave.

Systém bude sloužit operátorovi mikrotronu jako pomůcka pro nastavení profilu svazku resp. jeho těžiště či tvaru svazku. Různé tvary profilu svazku jsou vyžadovány pro různé velikosti a tvary ozařovaného terče (vzorku). Operátor však nadále bude nastavovat tvar svazku ručně pomocí fokusačních čoček umístěných na trase svazku.

V první kapitole budou uvedeny jednotlivé prvky systému tj. částicový urychlovač mikrotron a jeho funkce, dále krokový motor, drátkový detektor a jednočipový procesor Arduino. V druhé kapitole pak bude popsán princip zpětné Radonovy transformace a princip použitých filtrů. V následující třetí kapitole budou popsány jednotlivé simulace v prostředí Matlab či Octave. Ve čtvrté kapitole bude popsán algoritmus pro naprogramování jednočipového procesoru Arduino, jenž zahrnuje řízení krokového motoru i sběr naměřených dat. V poslední páté kapitole budou zhodnoceny výstupy z měření na mikrotronu MT25.

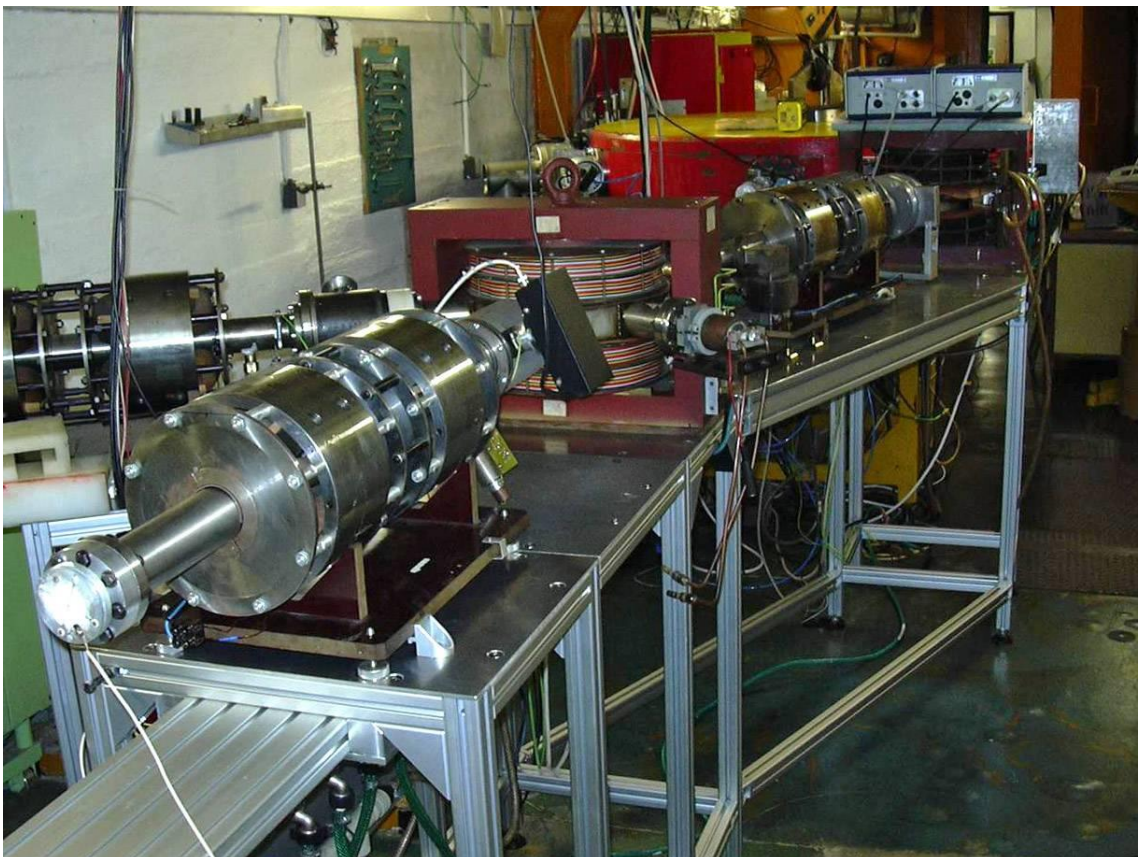
## 1.1 Mikrotron

Mikrotron je relativistický urychlovač elektronů pracující na principu objeveném v roce 1944 ruským fyzikem V. I. Vekslerem. Elektrony se v mikrotronu pohybují po kruhových drahách ve stacionárním magnetickém poli a jsou urychlovány během opětovných průchodu dutinovým rezonátorem, v němž je buzeno silné pulzní elektromagnetické pole s elektrickou složkou ve směru pohybu elektronu. Tento urychlovač má konstantní magnetické pole a konstantní amplitudu urychlovacího vf napětí. [5]

Kruhové urychlovače využívají velmi efektivní způsob, jak urychlit nabitě částice na vysoké energie. Jde o jejich mnohonásobné urychlení v elektrickém poli

$E_e$ , kam jsou elektrony opakovaně vráceny po kruhové dráze působením magnetického pole. Na elektron s nábojem  $e$  je zde aplikována nejen elektrická urychlující síla  $F_e = e \cdot E$ , ale i Lorentzova síla  $F_m = e [v \times B]$  působící v magnetickém poli (magnetická indukce  $B$ ) kolmo ke směru pohybu nabitě částice rychlostí  $v$ . Tato magnetická síla způsobuje, že nabitá částice o hmotnosti  $m$  se bude pohybovat po kruhové dráze o poloměru  $R = m \cdot v \cdot c / (e \cdot B)$ . Je-li ve vhodném místě této kruhové dráhy synchronně aplikováno elektrické urychlující pole (v tečném směru), budou částice periodicky urychlovány při každém svém oběhu. [6]

Vhodnou volbou velikosti magnetického pole při dané frekvenci vysokofrekvenčního elektromagnetického pole v rezonátoru lze dosáhnout toho, že všechny dráhy elektronu procházejí přibližně osou rezonátoru ve vhodné fázi, kdy elektrická složka pole elektrony urychluje. [5]

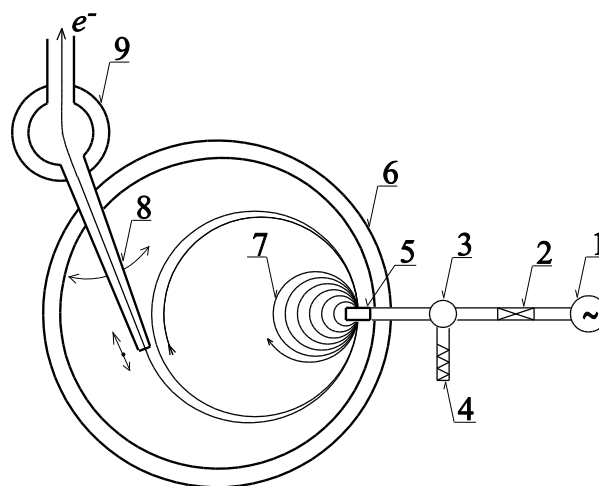


Obr. 1 Mikrotron MT25 – kruhový relativistický urychlovač elektronů [9]

Mikrotron MT25 [Obr. 1] je napájen z mikrovlnného zdroje, jímž je magnetron [viz Obr. 2]. Vlna postupuje vlnovodem, jehož součástí je posouvač fáze, cirkulátor a rezonátor. Posouvač fáze umožňuje nastavit vlnu tak, aby se její

kmitna nacházela v rezonátoru. Cirkulátor zamezuje průniku případné odražené vlny od rezonátoru zpět do magnetronu, což by vedlo k interferenci vln a možným výbojům v magnetronu. Odraženou vlnu odvádí do vodní zátěže, kde je její energie utracena. V rezonátoru je mikrovlnným zářením buzen vid  $TM_{010}$ , tedy intenzita elektrického pole uvnitř rezonátoru je rovnoběžná s osou cylindrického rezonátoru. Rezonátor urychluje elektrony emitované z přímo žhavené katody, která je umístěna vhodném místě v jedné s jeho stěn.

Svazek urychlených elektronů opíše ve vakuové komoře s homogenním magnetickým polem kružnici a vrací se zpět do rezonátoru v době, kdy má elektrická složka mikrovlnného záření optimální amplitudu. Elektrony jsou opět urychleny a opíší v urychlovací komoře kružnici o větším poloměru. Tento děj se neustále opakuje až do té doby, kdy elektrony vletí do vývodu svazku, který odstíní magnetické pole komory. Od této chvíle se již svazek elektronů pohybuje po přímce a vylétává ven z urychlovací komory do elektronovodu, kde je pomocí kvadrupólových čoček a dipólových deflektorů fokusován a vyváděn na požadované pracoviště. [4, 5]



Obr. 2 Schéma mikrotronu MT25. 1 – magnetron, 2 – posouvač fáze, 3 – cirkulátor, 4 – vodní zátěž, 5 – urychlovací rezonátor, 6 – vakuová komora urychlovače, 7 – trajektorie urychlovaných elektronů, 8 – nastavitelný vývod svazku, 9 – první defleční magnet [4]

Tab. 1 Hlavní parametry mikrotronu MT25

Energie elektronů	6 – 25 MeV
Energetický zisk na jednu orbitu	0,9 MeV
Počet orbit	25
Proud urychlených elektronů v pulzu	maximálně 20 mA (při 24 MeV)
Frekvence magnetronu	2796 ± 5 MHz
Pulzní výkon magnetronu	3 MW
Délka vf pulzu	3,9 μs
Opakovací frekvence vf pulzů	423 Hz

## 1.2 Krokový motor

Krokový motor byl pořízen Ústavem jaderné fyziky AV ČR v.v.i. společně s řídicí jednotkou motoru. Jedná se o krokový motor SX23-1414 řady SX23 [Obr. 3] dodávaný firmou MICROKON, s.r.o.. Krokový motor je řízen výkonovým stupněm SD20x [Obr. 4], též dodávaný firmou MICROKON, s.r.o..

Údaje v Tab. 2 jsou uvedeny pro sériové zapojení s výkonovým stupněm.



Obr. 3 Krokové motory řady SX23: vlevo SX23-1414, vpravo SX23-2727D

Tab. 2 Technické parametry krokového motoru SX23-1414

Statický moment	1,4 Nm
Jmenovitý proud	1,45 A
Indukčnost	11,2 mH
Odpor [ $\Omega$ ]	3,4 $\Omega$
Moment setrvačnosti rotoru	0,03 kgm <sup>2</sup> x10 <sup>-3</sup>
Hmotnost [kg]	0,69 kg
Délka kroku [°]	1,8°
Tolerance kroku [°]	$\pm 0,1^\circ$

Výkonový stupeň je ovládán třemi vnějšími signály *STEP*, *DIR* a *ENABLE*. Signálem *STEP* se ovládá krokování motoru, kdy při každé náběžné hraně na vstupu se krokový motor otočí o jeden nastavený krok. Logickou úrovní na vstupu *DIR* je udáván směr otáčení. Signálem *ENABLE* je umožněno zapnutí koncového stupně, tedy zapnutí/vypnutí krokového motoru.

Napěťová úroveň vstupů výkonové jednotky je 0 - 24V, volitelně sen dá změnou odporové kaskády nastavit na 0 - 5V.

Tab. 3 Parametry výkonového stupně

Napájecí napětí	12 - 48 VDC
Amplituda proudu	0,4 - 2 A
Vhodný statický moment	0,1 - 2,8 Nm
Napěťová úroveň vstupů	0; 24V (volitelně 0; 5V)

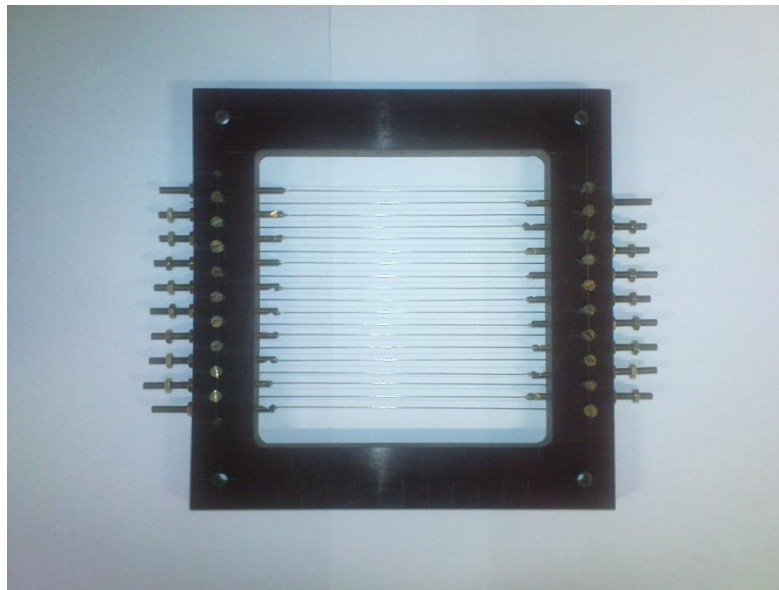


Obr. 4 Výkonový stupeň SD20x

Přepínačem SIP lze nastavit dělení kroku na 1, 2, 4, 8 a 16, kde nastavení dělení kroku na 8 nebo 16 se používá spíše pro zvýšení plynulosti chodu motoru než pro zvýšení rozlišení. Lze jím také nastavit amplitudu proudu motoru.

### 1.3 Drátkový detektor

Drátkový detektor použitý k diagnostice příčného profilu svazku je složen z několika tenkých ocelových drátků, které se při průchodu svazku vlivem emise sekundárních elektronů nabíjí. Střední proud sekundárních elektronů se pohybuje v jednotkách až desítkách nanoampér. Proud v pulsu dosahuje přibližně tisíckrát vyšší hodnoty.



*Obr. 5 Drátkový detektor*

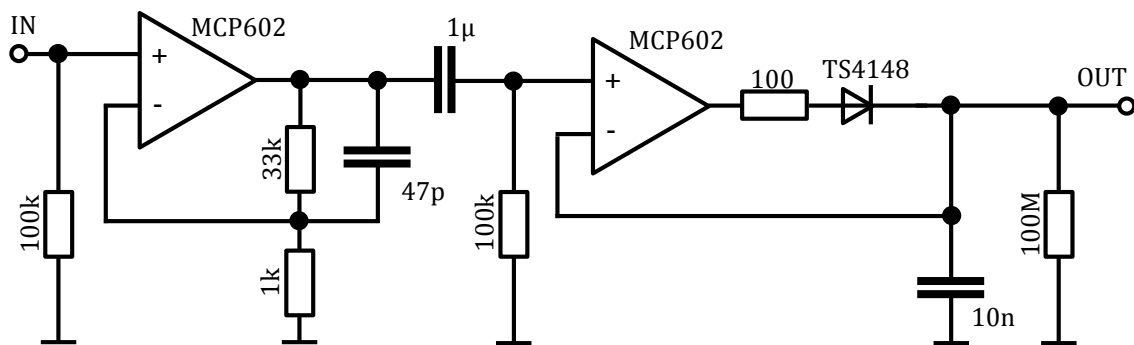
Na Obr. 5 je fotografie použitého detektoru. Celkový počet použitých detekčních drátků je 16. Každý drátek slouží jako zdroj proudu, kde čím je vyšší intenzita svazku v daném místě, tím je vyšší emise sekundární elektronů, tedy větší proud.

Drátkový detektor je připevněn na aparaturu, která je natáčena krokovým motorem.

## 1.4 Převodník pro drátkový detektor

Při průchodu urychlených elektronů detekčním drátkem dojde k emisi elektronů z drátku, což vytvoří proud pulsního charakteru (svazek urychlených elektronů je také pulsní). Za standartních podmínek, kdy střední proud urychlených elektronů je v desítkách mikroampér, můžeme naměřit hodnotu tohoto proudu v desítkách nanoampér. Analogové vstupy jednočipového procesoru Arduino MEGA 2560 jsou v rozmezí 0 – 5 V, proto je nutné proud na detekčních drátkách drátkového detektoru převést na napětí a zesílit do potřebného rozsahu napětí.

Převodník navrhli a vyrobili pracovníci mikrotronové laboratoře, jehož elektrický obvod je znázorněn na Obr. 6. Výsledkem je tištěný spoj se šestnácti kanály, každý pro jeden drátek detektoru.



Obr. 6 Elektrický obvod převodníku pro jeden kanál

## 1.5 Arduino

Vznik Arduina započal v italském městě Ivrea roku 2005. Kde se lidé z italského Interaction Design Institute rozhodli vytvořit jednoduchou prototypovací platformu pro studenty, která by byla volně dostupná a jednoduchá na používání. Samotný projekt byl pojmenovaný po významné postavě města Ivrey, Arduinovi Ivrejském.

Každá deska Arduina obsahuje procesor od firmy Atmel. Existuje celá řada oficiálních desek různých velikostí. Různí se svou velikostí, rozložením komponent, možností použití, množstvím vstupů a výstupů, výpočetním výkonem atd.. Také existuje celá řada tzv. shieldů, což jsou rozšiřující desky pro Arduino. Mezi

nejpoužívanější patří např. Ethernet Shield, Wifi Shield či Motor Shield. Na trhu jsou dostupné i tzv. klony Arduina. Jedná se o desky, které nepatří do oficiální řady.

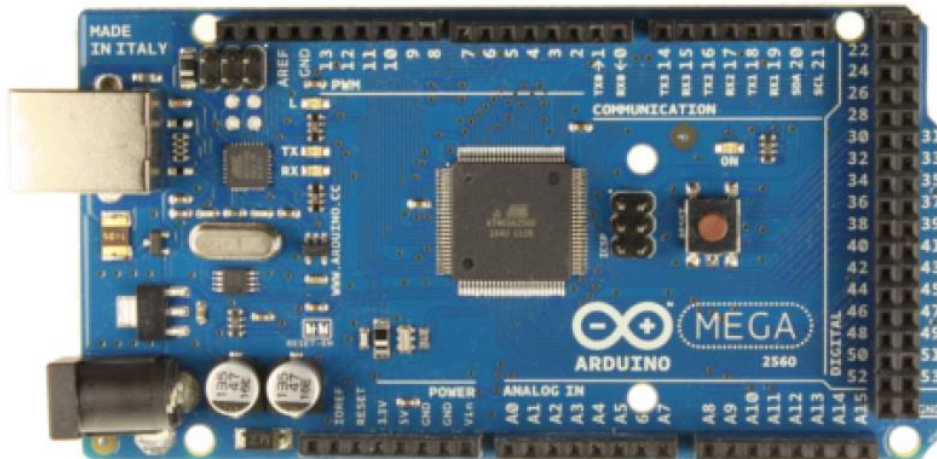
K programování Arduina se používá volně stažitelný software Arduino IDE, který je v současné době ve verzi 1.8.0. Instalace je možná na operační systém Windows, Linux a Mac OS. Instalace se liší podle použitého operačního systému. Pro Windows je možné stáhnout ZIP archiv, který je po rozbalení plně funkční.

Pro naše účely bylo pořízeno Ústavem jaderné fyziky AV ČR v.v.i. Arduino Mega2560 [Obr. 7]. Jedná se o desku s procesorem ATmega2560. Obsahuje 54 digitálních I/O pinů, 15 pinů lze použít jako PWM výstup, dále obsahuje 16 analogových vstupů, 4 UARTs (hardwarové sériové porty), 16 MHz krystalový oscilátor, USB konektor, napájecí konektor, přepínač ICSP a tlačítko reset. Technické parametry Arduina 2560 jsou k nahlédnutí v Tab. 4.

*Tab. 4 Technické specifikace Arduina Mega2560*

Pracovní napětí	5 V
Vstupní napětí (doporučené)	7 – 12 V
Vstupní napětí (mezní)	6 – 20 V
DC proud na I/O pin	20 mA
DC proud z 3,3 V pinu	50 mA
Rozlišení AD převodníků	10 bit (1024 hodnot)
Flash paměť	256 KB (8 KB pro proměnné)
SRAM	8 KB
EEPROM	4 KB
Délka	101,52 mm
Šířka	53,3 mm
Váha	37 g





Obr. 7 Arduino Mega2560 [7]

Výběr desky spočíval v možnosti použít co nejvíce analogových vstupů a proto je Arduino Mega2560 pro tento systém vyhovující. Desky s více analogovými vstupy se v oficiální řadě nenacházejí, ale v případě potřeby lze desku rozšířit o I/O shield.

## 2 Teoretický základ

Všechny moderní zobrazovací systémy jako jsou rentgenová výpočetní tomografie (CT), magnetická rezonance (MRI), pozitronová emisní tomografie (PET) nebo jednoduchá emisní tomografie (SPECT) pracují na projekčně rekonstrukčním principu. Znamená to, že všechny tyto modalities poskytující pouze soubor projekcí snímaného objektu. Z tohoto souboru projekcí nelze diagnostikovat, a proto je nutné z něj zrekonstruovat obraz snímaného objektu, k čemuž právě slouží algoritmy rekonstrukce z projekcí. [3]

V této se kapitole bude vysvětlen princip vytváření souboru projekcí, tedy Radonovu transformaci a seznámíme se základními principy rekonstrukce prostřednictvím frekvenční oblasti, prosté zpětné projekce a filtrované zpětné projekce.

### 2.1 Radonova transformace

Radonova transformace je matematická transformace pojmenovaná po Johannovi Radonovi, jenž ji zformuloval. Jedná se integrální transformaci, která spočívá v integraci funkce přes její nadroviny.

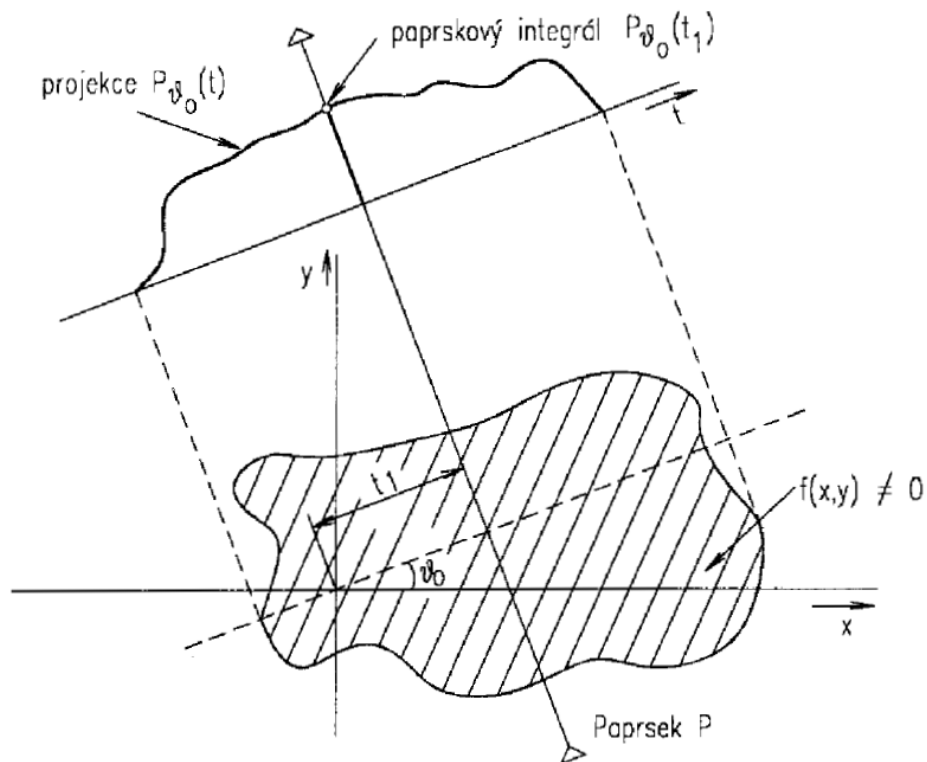
Pokud máme topografický řez tělesem, který leží v rovině  $xy$  a jsou dány jeho charakteristické vlastnosti v každém bodě funkcí  $f(x,y)$  (dále jen předmětová funkce). Můžeme zobrazit integrál této funkce po přímce  $p$  na osu  $t$ . Tím získáme bod, který se nazývá paprskový integrál, viz Obr. 8. Paprskový integrál lze popsat rovnicí [1]

$$P_{\vartheta_0}(t_1) = \int_p f(x,y) ds, \quad (2.1)$$

kde  $\vartheta_0$  je úhel, který svírá přímka  $p$  s osou  $x$ , parametr  $t_1$  je vzdálenost od počátku souřadnic a  $f(x,y)$  je snímaná funkce. Soubor paprskových integrálů zobrazených na ose  $t$  je nazýván projekce. Když budeme spojitě měnit úhel  $\vartheta$ , vznikne souvislý

soubor projekcí, který můžeme chápat jako spojitou funkci dvou proměnných a změni se nám rovnice (2.1) na rovnici [1]

$$P_{\vartheta}(t) = P(t, \vartheta) = \int_{t, \vartheta} f(x, y) ds = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x, y) \delta(x \cos \vartheta + y \sin \vartheta - t) dx dy. \quad (2.2)$$



Obr. 8 Princip vzniku projekce [1]

Dále platí:

$$P_{\vartheta}(t) = P(t, \vartheta) = R[f(x, y)], \quad (2.3)$$

kde  $R[f(x, y)]$  je Radonův operátor. A vyjádříme-li spojitou změnu souřadné systému v závislosti na  $\vartheta$  (pro následující rovnice nahradíme parametr  $t$  za  $x'$ ) [2]

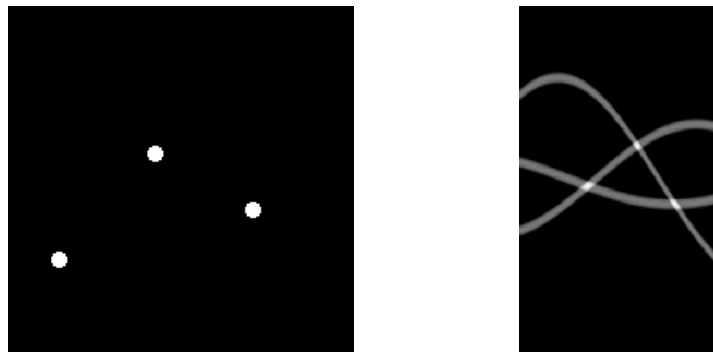
$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos \vartheta & \sin \vartheta \\ -\sin \vartheta & \cos \vartheta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}, \quad (2.4)$$

kde  $x'$  a  $y'$  je nově získaný souřadný systém.

Můžeme napsat vztah [2]

$$R[f(x, y)] = \int_{-\infty}^{\infty} f(x' \cos \vartheta - y' \sin \vartheta, x' \sin \vartheta + y' \cos \vartheta) dy'. \quad (2.5)$$

Takto získaný soubor projekcí je ukládán do vymezené oblasti tzv. Radonův prostor, což je v podstatě obrazová matice  $P(t, \vartheta)$ . Radonovou transformací tedy rozumíme naplnění Radonova prostoru souborem projekcí. Prostorová reprezentace obrazové matice se nazývá sinogram. Název vyplývá ze skutečnosti, že výstupem Radonovy transformace Diracovy delta funkce je právě sinusoida, tedy každý bod popsaný předmětovou funkcí se po transformaci zobrazí v sinogramu jako část harmonické funkce. Amplitudy a fáze těchto funkcí jsou závislé na poloze v původní funkci [Obr. 9].



Obr. 9 Vliv pozice bodu na amplitudu harmonické funkce  
vlevo detekovaný objekt, vpravo sinogram

Radonova transformace se provádí pro snímané projekce v intervalech  $\langle 0, \pi \rangle$  nebo  $\langle 0, 2\pi \rangle$ , protože projekce se opakují s periodou  $2\pi$ .

Jak již bylo zmíněno výše, Radonova transformace je integrální metodou, pro použití v praxi to znamená, že bychom museli splnit následující podmínky. Snímaný objekt by musel být stacionární, tedy neměnicí se v čase. Museli bychom snímat nekonečný počet projekcí s nekonečným množstvím paprskových integrálů. Těchto podmínek nejsme schopni v reálném světě dosáhnout, a proto výpočetní algoritmy, které rekonstruují původní obraz z nasnímaných projekcí, musí být přizpůsobený těmto podmínkám. Při snímání projekcí v diskrétních soustavách se hodnota paprskového integrálů tvoří za pomoci sumace nikoli integrálu.

Formulace inverzní Radonovy transformace je snahou docílit nalezení vhodné aproximační diskrétní metody pro rekonstrukci obrazu z projekcí. K dispozici je několik algoritmů zabývajících se rekonstrukcí obrazů. V této práci se zaměříme na dvě z nich, jenž jest zpětná prostá projekce a filtrovaná zpětná projekce. Nejprve si však popíšeme princip rekonstrukce prostřednictvím frekvenční oblasti.

## 2.2 Fourierova rekonstrukce

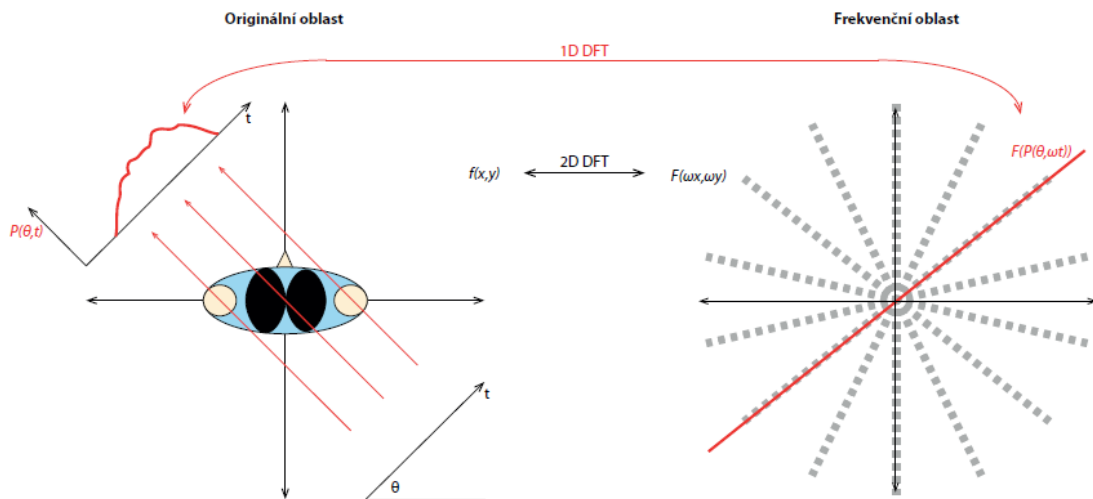
Rekonstrukce prostřednictvím frekvenční oblasti známá pod pojmem Fourierova rekonstrukce je jedna z metod rekonstrukcí obrazu a vychází z projekčního teorému. Nejedná se tedy o metodu na bázi inverzí Radonovy transformace.

Projekční teorém nebo též centrální řezový teorém je jeden ze základních fenoménů obrazových rekonstrukčních metod na bázi zpětné projekce. Která říká, že 1DFT (One-Dimensional Fourier Transform) jedné projekce nasnímané pod úhlem  $\vartheta$  je totéž jako radikální řez 2DFT (Two-Dimensional Fourier Transform) původní funkce ve stejném směru  $\vartheta$  [Obr. 10].

Projekční teorém lze popsat rovnicí [2]

$$P_{\vartheta}(\omega) = F(\omega \cos \vartheta, \omega \sin \vartheta) = F(\omega_{x'}, \omega_{y'})|_{\vartheta} \text{ nebo } F(\omega_x, \omega_y)|_{\vartheta} = F(\omega, \vartheta), \quad (2.6)$$

kde  $F(\omega_x, \omega_y)$  je 2D Fourierova transformace předmětové funkce  $f(x, y)$  a  $(\omega_{x'}, \omega_{y'})$  resp.  $(\omega_x, \omega_y)$  a  $(\omega, \vartheta)$  reprezentující kartézské/polární souřadnice ve frekvenční (Fourierově) doméně.

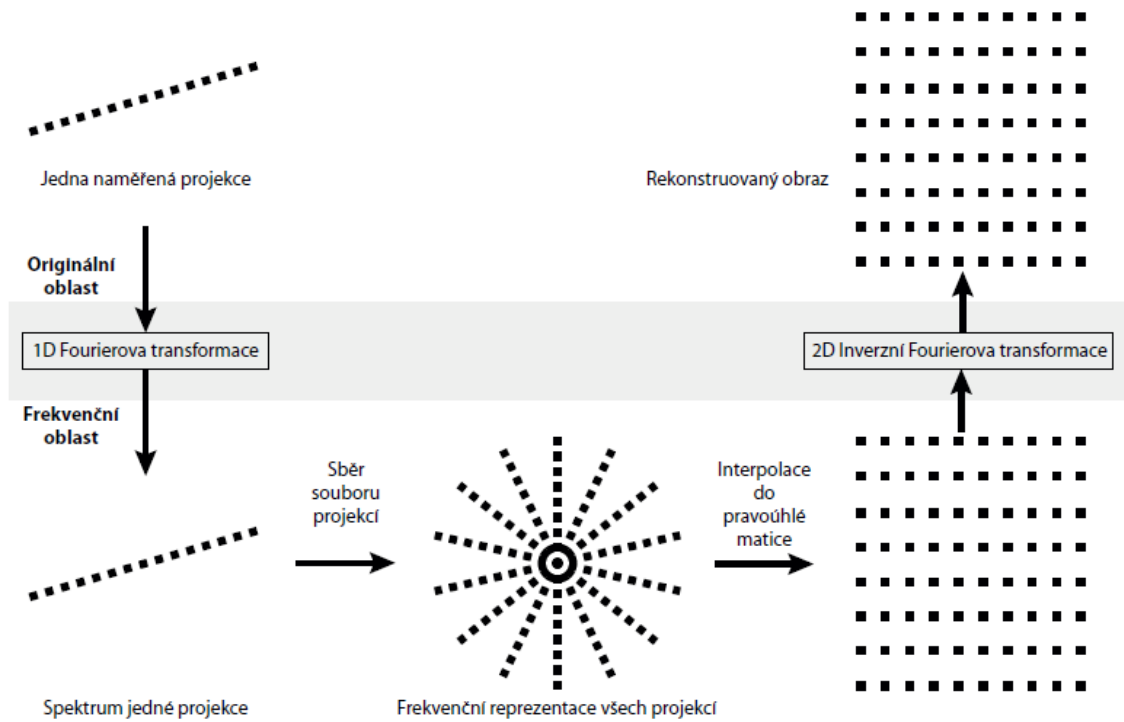


Obr. 10 Principiální vysvětlení projekčního teorému [3]

Princip Fourierovy rekonstrukce spočívá v tom, že do prázdného prostoru tzv. spektrální oblasti se převedou, použitím 1DFT, nasnímané projekce pro jednotlivé úhly  $\vartheta$ . Tím se získá soubor jednotlivých spekter. Tento soubor se, následnou interpolací, převede do pravoúhlé matice tzv. spektrální rekonstrukční matice. Posledním krokem k získání rekonstruovaného obrazu je aplikace 2DFT na spektrální rekonstrukční matici, viz Obr. 11.

Matematicky je to popsáno rovnicí [2]

$$f'(x, y) = F_{2D}^{-1}[F(\omega_x, \omega_y)] = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} F(\omega_x, \omega_y) e^{j(x\omega_x + y\omega_y)} d\omega_x d\omega_y = f'(r, \vartheta) dt. \quad (2.7)$$



Obr. 11 Princip rekonstrukce prostřednictvím frekvenční oblasti [3]

Též je možné provést převzorkování do kartézských souřadnic až výsledném obrazu a rekonstrukci počítat v polárních souřadnicích podle vztahu [2]

$$f'(r, \beta) = \int_0^\pi \int_{-\infty}^{\infty} F(\omega, \vartheta) e^{j\omega(x \cos \vartheta + z \sin \vartheta)} |J| d\omega d\vartheta, \quad (2.8)$$

kde  $J$  je Jakobián, pro který platí

$$J = \begin{bmatrix} \frac{\partial \omega_x}{\partial \omega} & \frac{\partial \omega_y}{\partial \omega} \\ \frac{\partial \omega_x}{\partial \vartheta} & \frac{\partial \omega_y}{\partial \vartheta} \end{bmatrix} = \begin{bmatrix} \cos \vartheta & \sin \vartheta \\ -\omega \sin \vartheta & \omega \cos \vartheta \end{bmatrix} = \omega, \quad (2.9)$$

$$\omega = \sqrt{\omega_x^2 + \omega_y^2}, \vartheta = \arctan \frac{\omega_x}{\omega_y}, r = \sqrt{x^2 + y^2}, \beta = \arctan \frac{y}{x}. \quad (2.10)$$

Avšak zde je úvaha pouze teoretická, protože hlavní výhodou Fourierovy rekonstrukce je rychlost navržených algoritmů na bázi rychlé zpětné Fourierovy transformace (IFFT – Invert Fast Fourier Transform).

### 2.3 Prostá zpětná projekce

Integrální Radonova transformace je teoreticky přesně inverzibilní, ale vzhledem k diskrétnímu charakteru sejmutých projekcí a jejich konečnému počtu je nutné pro rekonstrukci snímaného obrazu použít diskrétní aproximaci inverzní Radonovy transformace. Nejjednodušší a do značné míry heuristickou metodou, kterou nelze požadovat za aproximaci inverzní Radonovy transformace, je metoda tzv. prosté zpětné projekce (SBP – Simple Back Projection). [3]

Myšlena metody spočívá v získání předmětové funkce přesně opačným způsobem, jakým se získal soubor projekcí. Tedy vzít každou nasnímanou projekci z Radonova prostoru a rozprostřít ji do prázdného prostoru pod úhlem, pod jakým byly nasnímány. Takto získané prostory se sečtou, čímž získáme rekonstruovaný obraz.

Matematický zápis lze ve spojitém tvaru vyjádřit [2]

$$\hat{f}(x, y) = \frac{1}{\pi} \int_0^{\pi} p_{\vartheta}(x \cos \vartheta + y \sin \vartheta) d\vartheta, \quad (2.11)$$

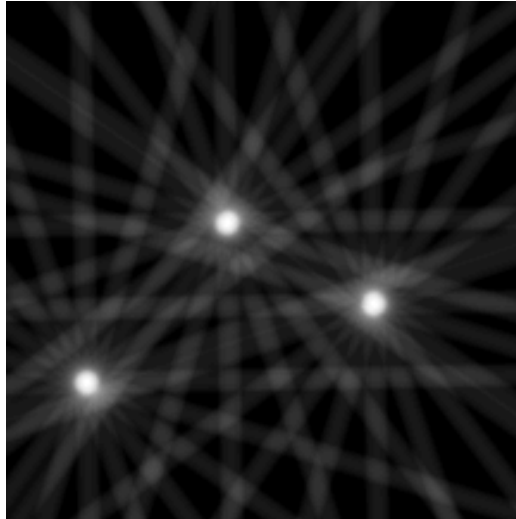
pro diskrétní tvar je pak rovnice následující

$$\hat{f}(x, y) = \frac{1}{N} \sum_{i=1}^N p((x \cos \vartheta_i + y \sin \vartheta_i, \vartheta_i)), \quad (2.12)$$

kde,  $\hat{f}(x, y)$  je rekonstruovaný obraz jako 2D obrazová funkce,  $N$  je počet projekcí a  $\vartheta_i$  je úhel  $i$ -té projekce.

Nevýhodou metody je značné rozostření obrazu, které je způsobeno sčítáním projekcí v celém obsahu prostoru v daném směru. Toto rozostření je nazýváno jako hvězdicový artefakt [viz Obr. 12], který je závislý na počtu použitých projekcí. Čím více projekcí použijeme, tím méně bude patrný hvězdicový artefakt, úplného zaostření obrazu se však nedočkáme.





Obr. 12 Hvězdicový artefakt při použití 10 projekcí

Tato metoda rekonstrukce obrazu se v praxi moc nevyužívá. Ačkoli je znám charakter rozostření, nabízí se možnost provést filtraci rekonstruovaného obrazu tzv. ostřícím filtrem. Výhodnějším je však provést filtraci projekcí ještě před rekonstrukcí obrazu. Tímto se zabývá následující metoda zvaná filtrovaná zpětná projekce.

## 2.4 Filtrovaná zpětná projekce

Prostou zpětnou projekci nelze považovat za teoreticky přesnou inverzní Radonovu transformaci. Lze matematicky dokázat, že přesnou inverzní Radonovu transformaci lze realizovat provedením zpětné projekce souboru tzv. filtrovaných projekcí, odkud je také odvozen název další rekonstrukční metody filtrované zpětné projekce. Filtrovaná zpětná projekce je velmi podobná prosté zpětné projekci, opět se soubor jednorozměrných projekcí rozprostírá přes rekonstrukční matici, čímž vzniká obraz. Tentokrát jsou ale rozprostírány projekce filtrované jednorozměrným FIR filtrem. [3]

Filtrovaná zpětná projekce (FBP – Filtered Back Projection) hledá rekonstruovanou funkci  $f'(x,y)$  podle vztahu [2]

$$f'(x,y) = \int_0^\pi \int_{-\infty}^{\infty} P_\vartheta(x') h(x \cos \vartheta + y \sin \vartheta - x') dx' d\vartheta, \quad (2.13)$$

kde funkce  $h(x,y)$  provádí korekci zkreslení prosté zpětné projekci. Z porovnání rovnic (2.11) a (2.13) lze vypočítat konvoluční integrál

$$\int_{-\infty}^{\infty} P_{\vartheta}(x')h(x \cos \vartheta + y \sin \vartheta - x')dx' d\vartheta, \quad (2.14)$$

který provádí danou filtraci. Dále je potřebné určit impulzní charakteristiku filtru, tedy funkci  $h(x)$ . Nejprve vyjádříme rovnici (2.14) v polárních souřadnicích, k tomu využijeme vztah (2.10) [2]

$$f'(r, \beta) = \int_0^{\pi} \int_{-\infty}^{\infty} P_{\vartheta}(x')h(r - x')dx' d\vartheta. \quad (2.15)$$

Ze srovnání vztahů (2.15) a (2.8) vyplývá že [2]

$$\int F(\omega, \vartheta)e^{j\omega r}|J|d\omega = \int_{-\infty}^{\infty} P_{\vartheta}(x')h(rx')dx', \quad (2.16)$$

kde výraz na levé straně představuje zpětnou Fourierovu transformaci a výraz na pravé straně představuje konvoluci dvou funkcí v prostorové oblasti. Můžeme tedy napsat, že [2]

$$F(\omega, \vartheta)|J| = \mathbf{F}[P_{\vartheta}(x') * h(x')] = \mathbf{F}[P_{\vartheta}(x')] \cdot \mathbf{F}[h(x')]. \quad (2.17)$$

Z poznatků vztahu (2.7) a po dosazení  $J$  ze vztahu (2.9), musí tedy platit [2]

$$\mathbf{F}[h(x')] = |\omega|. \quad (2.18)$$

Pomocí tohoto vztahu jsme schopni vypočítat impulzní charakteristiku filtru, tedy funkci  $h(x')$  jako zpětnou Fourierovu transformaci zpětnou transformaci známého Jakobíanu. Takto získané filtry jsou označovány jako ramp-filtry.

V reálných systémech jsou data z projekcí vzorkována, používá se výhradně číslicová filtrace. V tomto případě je spektrum každého filtru periodické s periodou rovnou vzorkovacímu úhlovému kmitočtu. U ideálního ramp-filtru požadujeme [2]

$$\mathbf{F}[h(x')] = \begin{cases} |\omega|, & |\omega| \leq \frac{x}{\Delta x} 2B \\ 0, & \text{jinde,} \end{cases} \quad (2.19)$$

Odtud můžeme získat  $h'(x)$  podle vztahu (2.18) [2]

$$h(x') = \frac{1}{2\pi} \int_{-\infty}^{\infty} H(\omega) e^{jx\omega} d\omega = \frac{1}{2\pi} \int_{-2\pi B}^{2\pi B} H(\omega) e^{jx\omega} d\omega = 2B^2 \text{sinc}(2\pi Bx) - B^2 \text{sinc}(\pi Bx), \quad (2.20)$$

kde  $\text{sinc}(x) = \sin(x)/x$ .

Diskretizací vztahu (2.20) v bodech  $x_k = k \cdot \Delta x$  získáváme [2]

$$h_{RL}(k) = \begin{cases} B^2 = \frac{1}{4\pi\Delta x^2} & \text{pro } k = 0 \\ 0, & \text{pro } k \text{ sudé.} \\ \frac{4B^2}{\pi^2 k^2} = \frac{-1}{\pi^2 k^2 \Delta x^2} & \text{pro } k = 0 \end{cases} \quad (2.21)$$

Tento filtr je nazýván Ram-Laken filtr.

Filtry s takto získanou přenosovou charakteristikou nejsou však fyzikálně realizovatelné, jelikož vykazují lineární nárůst přenosové charakteristiky s frekvencí a při velkém zesílení na vysokých kmitočtech dochází k nárůstu šumu a snižování poměru signál-šum. Proto se musíme spokojit s jeho aproximací. V praktických aplikacích se proto frekvenční charakteristika ramp filtru násobí jinými funkcemi, které omezí vysoké kmitočty tzv. okny. Výsledné filtry jsou pojmenovány podle funkce, kterou byl ramp filtr vynásoben. Mezi nejpoužívanější spadají tyto: [8]

Shepp-Logan Filtr:

$$\mathbf{F}[h(x')] = \begin{cases} |\omega| \cdot \sin \frac{|\omega|}{4B}, & |\omega| \leq 2B ; \\ 0, & \text{jinde,} \end{cases} \quad (2.22)$$

Hamming filtr:

$$\mathbf{F}[h(x')] = \begin{cases} |\omega| \cdot [0,54 + 0,46 \cos(\frac{\omega}{2B})], & |\omega| \leq 2B ; \\ 0, & \text{jinde,} \end{cases} \quad (2.23)$$

Hann filtr:

$$\mathbf{F}[h(x')] = \begin{cases} |\omega| \cdot 0,5 \left(1 + \cos \frac{\omega}{2B}\right), & |\omega| \leq 2B ; \\ 0, & \text{jinde,} \end{cases} \quad (2.24)$$

Cosine filtr:

$$\mathbf{F}[h(x')] = \begin{cases} |\omega| \cdot \cos\left(\frac{\omega}{4B} - \frac{\pi}{2}\right), & |\omega| \leq 2B ; \\ 0, & \text{jinde,} \end{cases} \quad (2.25)$$

### 3 Simulace rekonstrukce obrazu

V této kapitole bude popsán postup, jakým jsem postupoval při návrhu simulací rekonstrukce obrazu pomocí filtrované zpětné projekce. Nejprve si ukážeme tvorbu sinogramu pro různý počet snímaných projekcí na obrázku fantomu. Dále si popíšeme rekonstrukci obrázku fantomu ze souboru projekcí filtrovanou zpětnou projekcí pro různé typy filtrů v závislosti na počtu nasnímaných projekcí. Poté se seznámíme s omezeními navrhovaného systému a jejich dopad na výsledný skript a provedeme sérii simulací na několika obrázcích odpovídajících profilům svazků urychlených elektronů.

Pro simulace jsem použil software GNU Octave, jazyk vyšší úrovně určený nejenom pro numerické výpočty. Jedná se o volně šiřitelný software, který je navržen tak, aby byl co nejvíce kompatibilní se softwarem Matlab. Při použití softwaru GNU Octave bude též zajištěno, že výsledná práce bude moci být použita na pracovištích, kde Matlab není k dispozici.

Pro práci s obrázky je potřeba GNU Octave rozšířit o balíček *Image*, který rozšiřuje Octave o funkce pro zpracování obrazu.

#### 3.1 Obecné simulace rekonstrukce obrazu

##### 3.1.1 Tvorba sinogramu

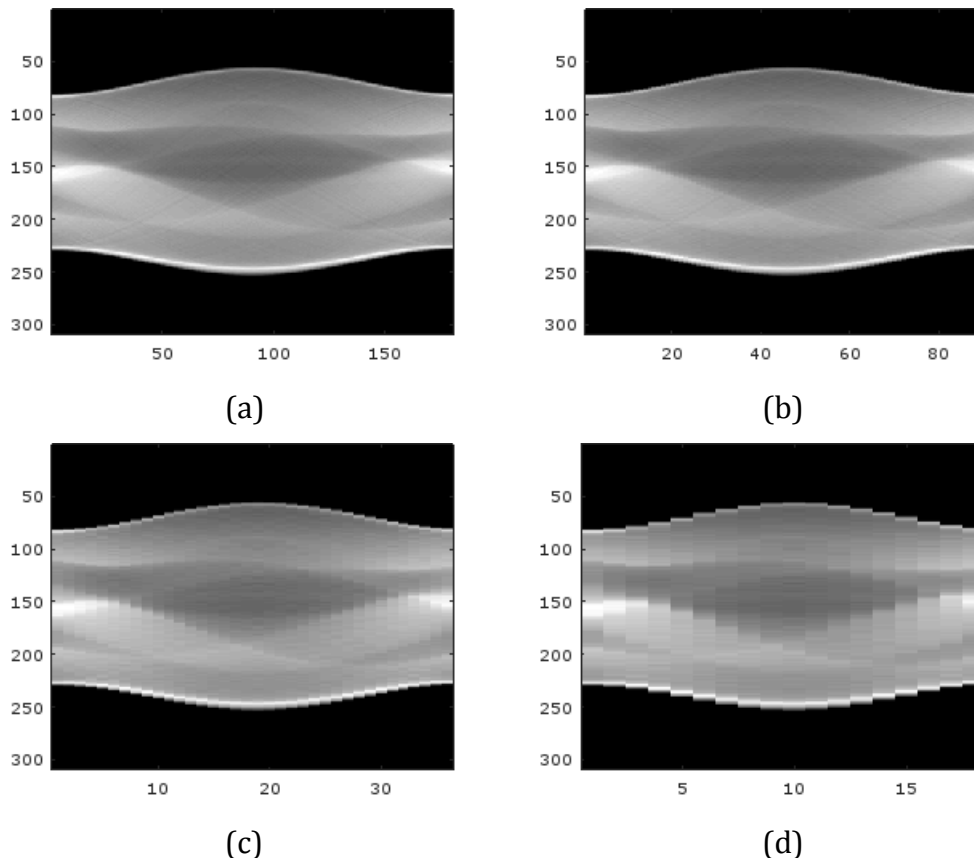
Jak již bylo popsáno v kapitole 2.1, pro rekonstrukci obrazu je nejprve potřeba vytvořit soubor projekcí původní obrazu, tedy sinogram. K tomuto účelu slouží funkce *radon*, která je součástí rozšiřujícího balíčku *Image*.

Tvorbu souboru projekcí si ukážeme na základním fantomovém obrázku [Obr. 13]. Shepp-Logan phantom je známý matematický model skládající se ze souboru překrývajících se elips, který umožňuje přesné testování počítačové tomografie. Obrázek fantomu je též součástí rozšiřujícího balíčku *Image* programu GNU Octave.



Obr. 13 Shepp-Logan phantom

Standartní zápis Octave funkce pro získání souboru projekcí pomocí funkce *radon* je následující: `SoubourProjekci = radon(I, theta);`, kde *I* je snímaný obrázek a *theta* je počet snímaných projekcí ve stupních v intervalu od  $0^\circ$  do  $179^\circ$ . Na Obr. 14 jsou znázorněné soubory projekcí Obr. 13 pro různý počet snímaných projekcí. Obrázky jsou z důvodů přehlednosti zobrazeny ve stejném měřítku.

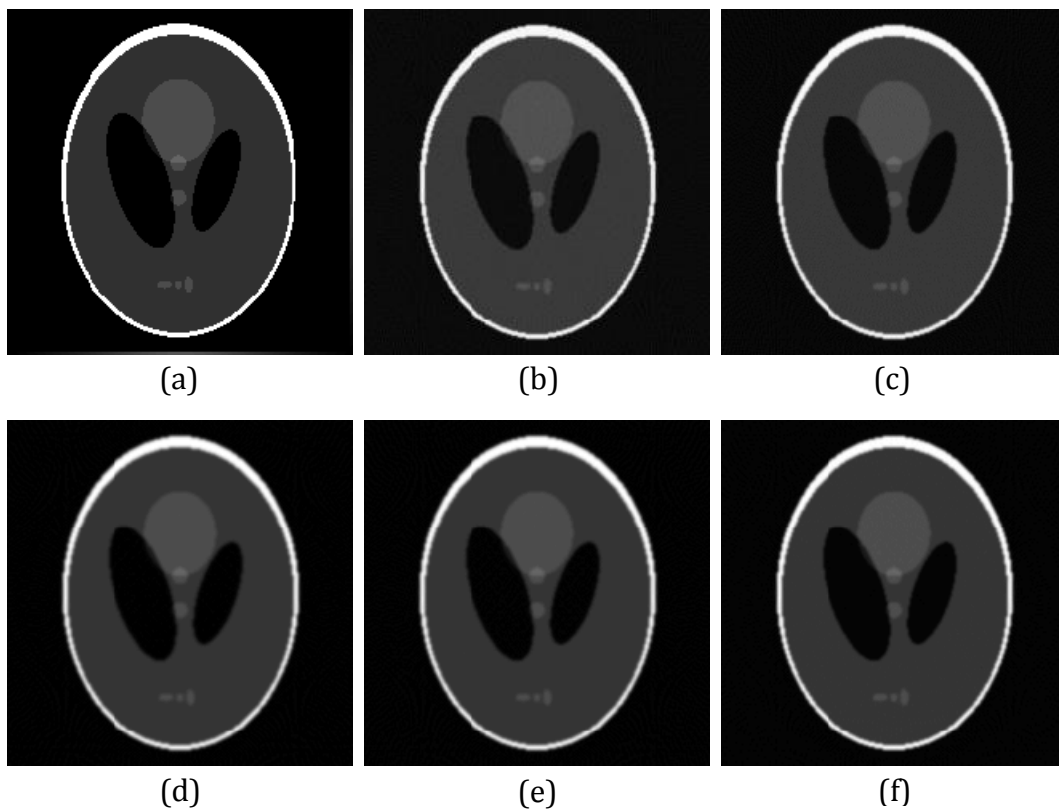


Obr. 14 Soubory projekcí: (a) ze 180 projekcí, (b) z 90 projekcí, (c) z 36 projekcí, (d) z 18 projekcí

### 3.1.2 Rekonstrukce obrazu

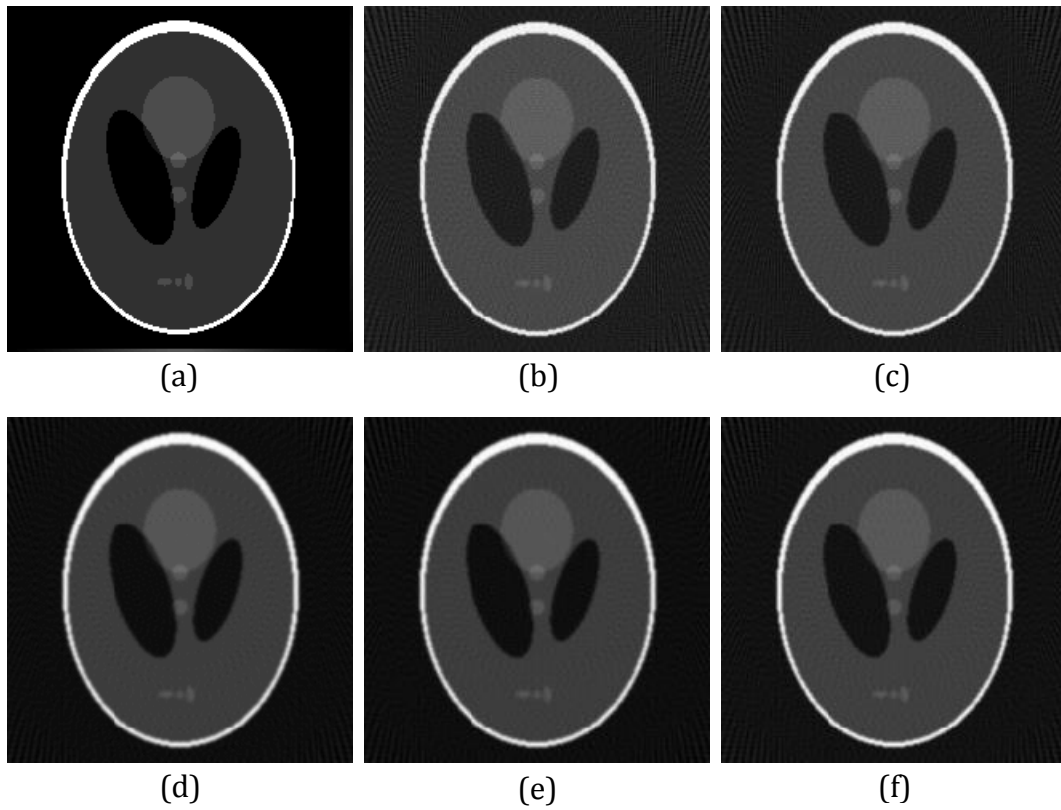
Pro rekonstrukci obrazu potřebujeme soubory projekcí, které jsme vytvořili v přechozí podkapitole [Obr. 14]. K tomu nám poslouží funkce *iradon*, která je stejně jako funkce *radon*, součástí rozšiřujícího balíčku *Image*.

Příkaz pro volání funkce *iradon* v programu GNU Octave je následující:  
`Rekonstrukce = iradon (SouborProjekci, theta, interpolace, filtr);`, kde *interpolace* znamená, jaká metoda interpolace bude použita. Pro nadcházející simulace jsem použil metodu lineární. Parametrem *filtr* nastavujeme, jaký filtr bude použit při zpětné filtrované projekci. Na výběr máme možností pro zpětnou filtrovanou projekci 'Ram-Lak', 'Shepp-Logan', 'Cosine', 'Hann' a 'Hamming'. Též může použít zpětnou prostou projekci, když do parametru *filtr* vložíme 'None'.

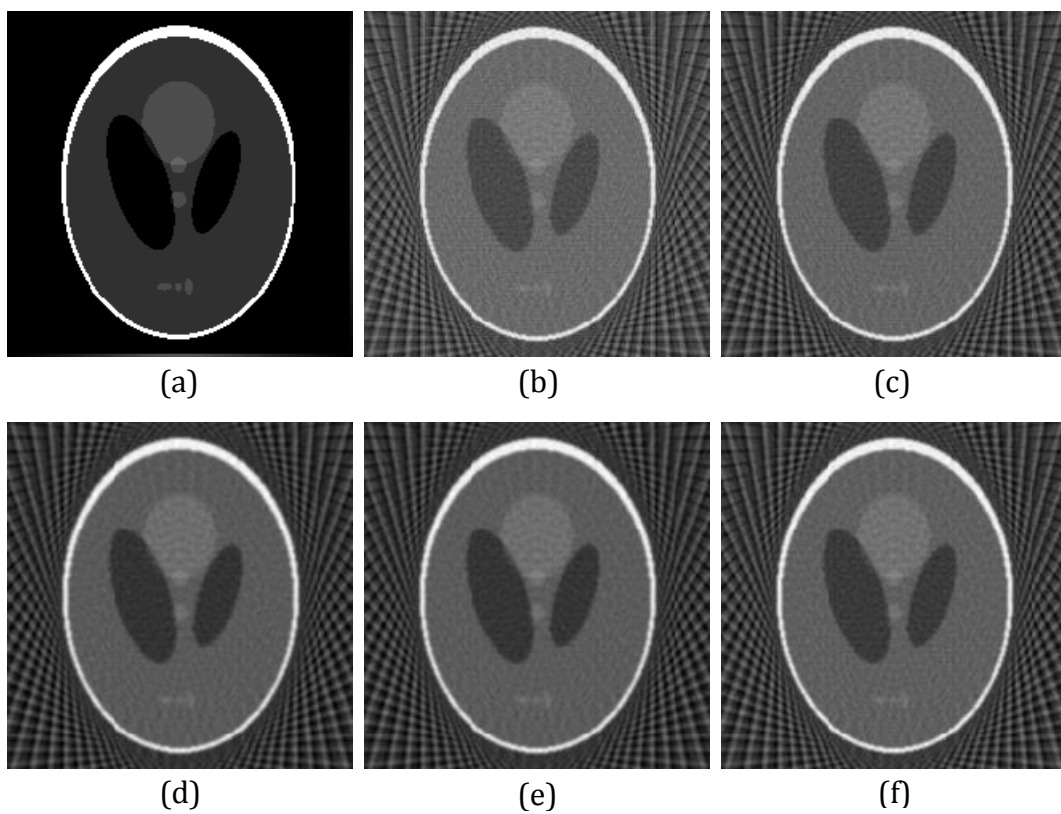


Obr. 15 Rekonstrukce obrazu ze 180 projekcí: (a) původní obraz, (b) Ram-Laken, (c) Shepp-Logan, (d) Hann, (e) Hamming, (f) Cosine

Na Obr. 15 můžeme vidět rekonstrukci obrazu ze 180 nasnímaných projekcí [Obr. 14a] pro různé použité filtry. Z pohledového hlediska není patrné, jak který filtr má dopady na rekonstruovaný obraz.



*Obr. 16 Rekonstrukce obrazu z 90 projekcí: (a) původní obraz, (b) Ram-Laken, (c) Shepp-Logan, (d) Hann, (e) Hamming, (f) Cosine*

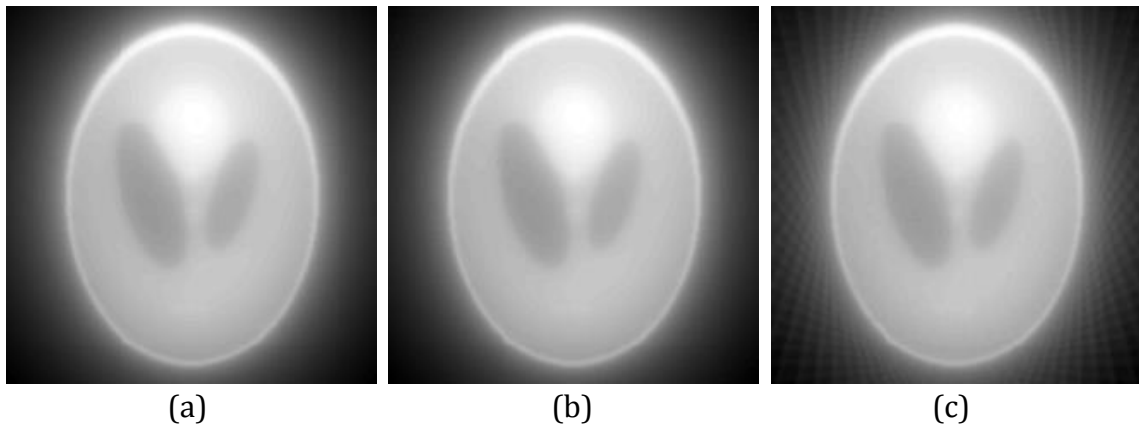


*Obr. 17 Rekonstrukce obrazu z 36 projekcí: (a) původní obraz, (b) Ram-Laken, (c) Shepp-Logan, (d) Hann, (e) Hamming, (f) Cosine*



Drobné rozdíly použité filtrace jsou patrné z Obr. 16, kde byla provedena rekonstrukce z 90 nasnímaných projekcí [Obr. 14b]. Jsou zde patrné vytvořené artefakty vznikající nedostatečným počtem projekcí.

Na Obr. 17 byla provedena rekonstrukce obrazu z 36 nasnímaných projekcí [Obr. 14c]. Na kterých je znázorněn vliv počtu projekcí na rekonstrukci obrazu.



Obr. 18 Zpětná prostá projekce: (a) ze 180 projekcí, (b) z 90 projekcí, (c) z 36 projekcí

Pro představu jsem provedl rekonstrukce použitím zpětné prosté projekce, z nichž je patrné rozostření obrazu [Obr. 18].

### 3.2 Reálný systém

V předchozí kapitole jsme si ukázali, jak obecně vytvořit rekonstrukci obrazu pomocí funkce *radon* a *iradon* v prostředí GNU Octave. Pro reálný systém jsou však stanovena omezení, kvůli kterým nebudeme moci tyto funkce použít.

Funkce *radon* při snímání jedné projekce počítá se všemi hodnotami v dané ose snímání, tedy s každým pixelem na ose. Při každém pootočení původního obrazu je vytvořen nový obraz s jiným rozměrem. Výstupem je tedy matice  $[m \times n]$ , kde  $m$  je uhlopříčka obrázku v pixelech a  $n$  je počet snímaných projekcí. V reálném systému používáme pouze šestnáct drátků, které jsou od sebe vzdáleny přibližně 5 mm, tudíž při použití tohoto způsobu snímání projekcí a tvorby sinogramu nemůžeme funkci *radon* použít. Napsal jsem tedy funkci *sinogram*, která nám nahrazuje funkci *radon* a nasnímá nám obrázek v určitých rozestupech odpovídající reálnému systému. Okomentovaná funkce *sinogram* je uvedena

v Příloze 1. Výstupem je také matice  $[m \times n]$ , kde ale  $m$  je počet použitých drátků, tedy 16 a  $n$  je počet projekcí. V našem případě je počet projekcí roven 200, vyplývá to z technologie použitého krokového motoru, viz kapitola 1.2 Krokový motor. V kapitole 3.1 Obecné simulace rekonstrukce obrazu jsme prokázali, že počet projekcí výrazně ovlivňuje kvalitu rekonstruovaného obrazu, proto jsem zvolil krok motoru po 0,9 stupních, tedy již zmíněných 200 projekcí. Též funkce *sinogram* pro každé otočení původního obrazu zachovává původní rozměr, což je dáno opět z použité technologie. Drátkový detektor je čtvercového tvaru, a pokud bychom jej centrálně rotovali kolem obrázku stejných rozměrů, uvidíme, že rohy obrázku se nám ve výsledné projekci nezobrazí, jelikož jsou překryté rámem. Nachází se tedy mimo oblast měření. Opačným jevem je pak ten, že rohy detektoru jsou umístěny mimo snímaný obrázek, takto vzniklý prázdný prostor je pak programově vyplněn nulami.

Z výše uvedených důvodů je pak dán předpoklad, že měřený svazek elektronů se musí nacházet v pracovní oblasti drátkového detektoru, tj. ve vestavěné kružnici. Čím více by byl svazek mimo pracovní oblast detektoru, tím více by bylo měření zkreslené.

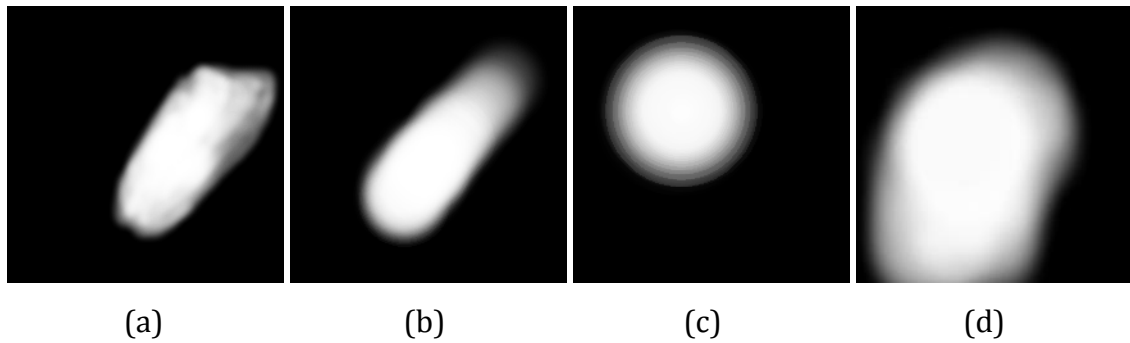
Funkce *iradon* nevrací zrekonstruovaný obraz v požadovaném rozměru, jaký potřebujeme. Jestliže použijeme soubor projekcí o velikosti 16x200, požadujeme, aby zrekonstruovaný obraz měl rozměr 16x16, který se nadále zvětší na požadovaný rozměr. Funkce *iradon* však vrací oříznutý obraz o velikosti 10x10, což je pro naše účely nepřijatelné. Napsal jsem tedy další funkci *rekonstrukce*, která provádí zpětnou filtrovanou projekci obrazu s požadovaným rozměrem a s možností volby filtru. Použití filtru zastává funkce *rho\_filter*, která je součástí i funkce *iradon*. Okomentovaná funkce *rekonstrukce* je uvedena v Příloze 2.

Výsledný okomentovaný skript *simulace*, ve kterém je provedena kompletní filtrovaná zpětná projekce, naleznete v Příloze 3.

### 3.3 Simulace reálného systému

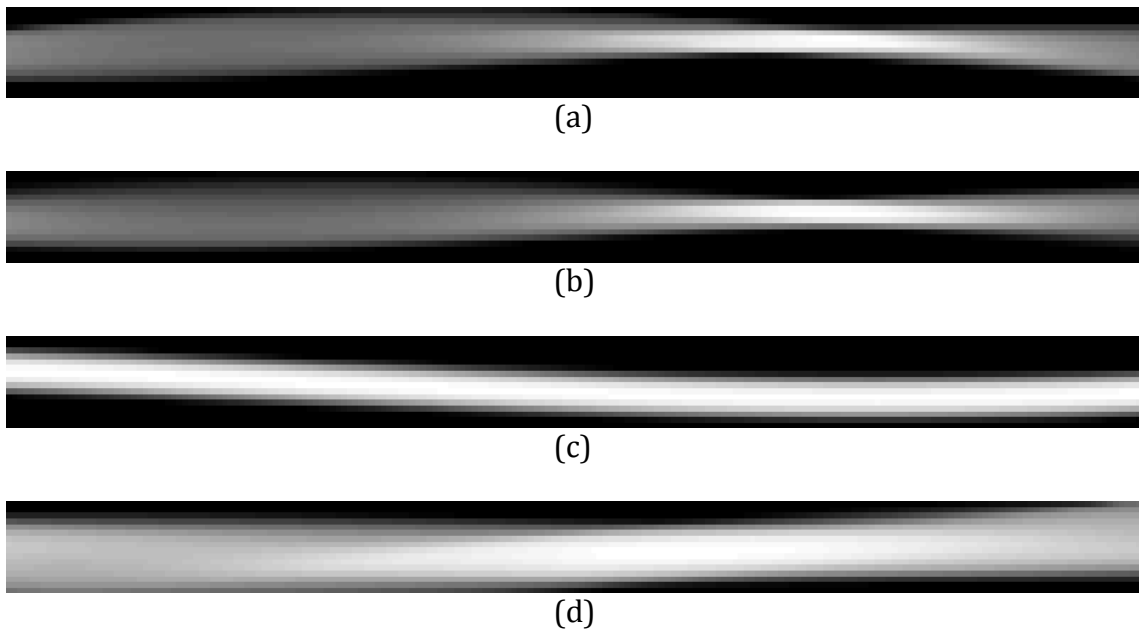
Pro simulaci navrhovaného systému mi pracovníci mikrotronové laboratoře dodali několik testovacích obrázků, reprezentující různé tvary svazků urychlených

elektronů [Obr. 19]. Na těchto obrázcích si ověříme funkci vytvořeného rekonstrukčního algoritmu.

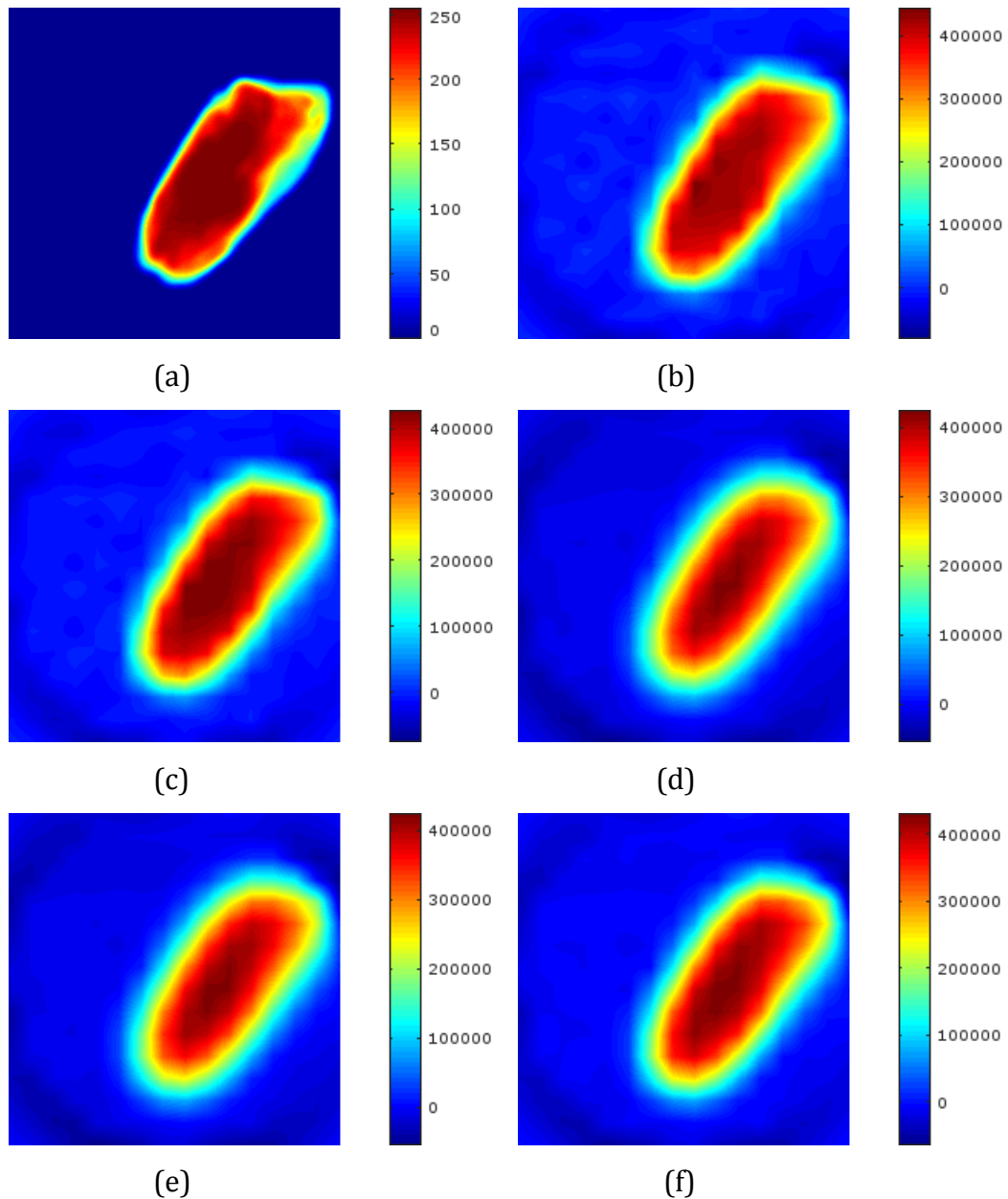


Obr. 19 Testovací obrázky

Na Obr. 20 jsou znázorněny sinogramy vytvořené funkcí *sinogram*, na nichž je patrné bitové rozlišení jednotlivých řádků, které je právě dáno množstvím snímaných pozic pro danou projekci.

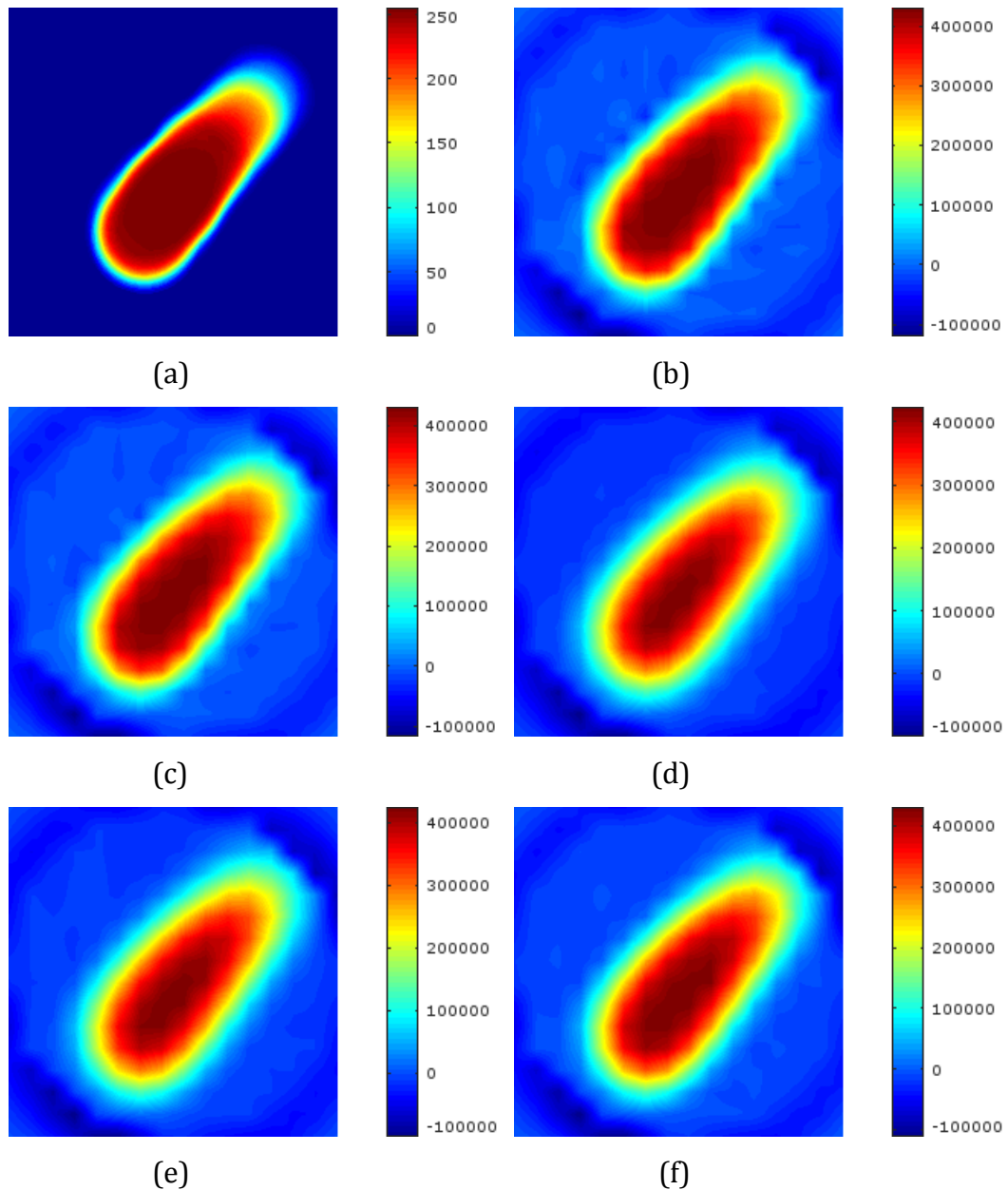


Obr. 20 Soubory projekcí testovacích obrázků:  
(a) Obr. 19a, (b) Obr. 19b, (c) Obr. 19c, (d) Obr. 19d



Obr. 21 Filtrovaná zpětná projekce Obr. 19a pro různé filtry: (a) původní obraz, (b) Ram-Laken, (c) Shepp-Logan, (d) Hann, (e) Hamming, (f) Cosine

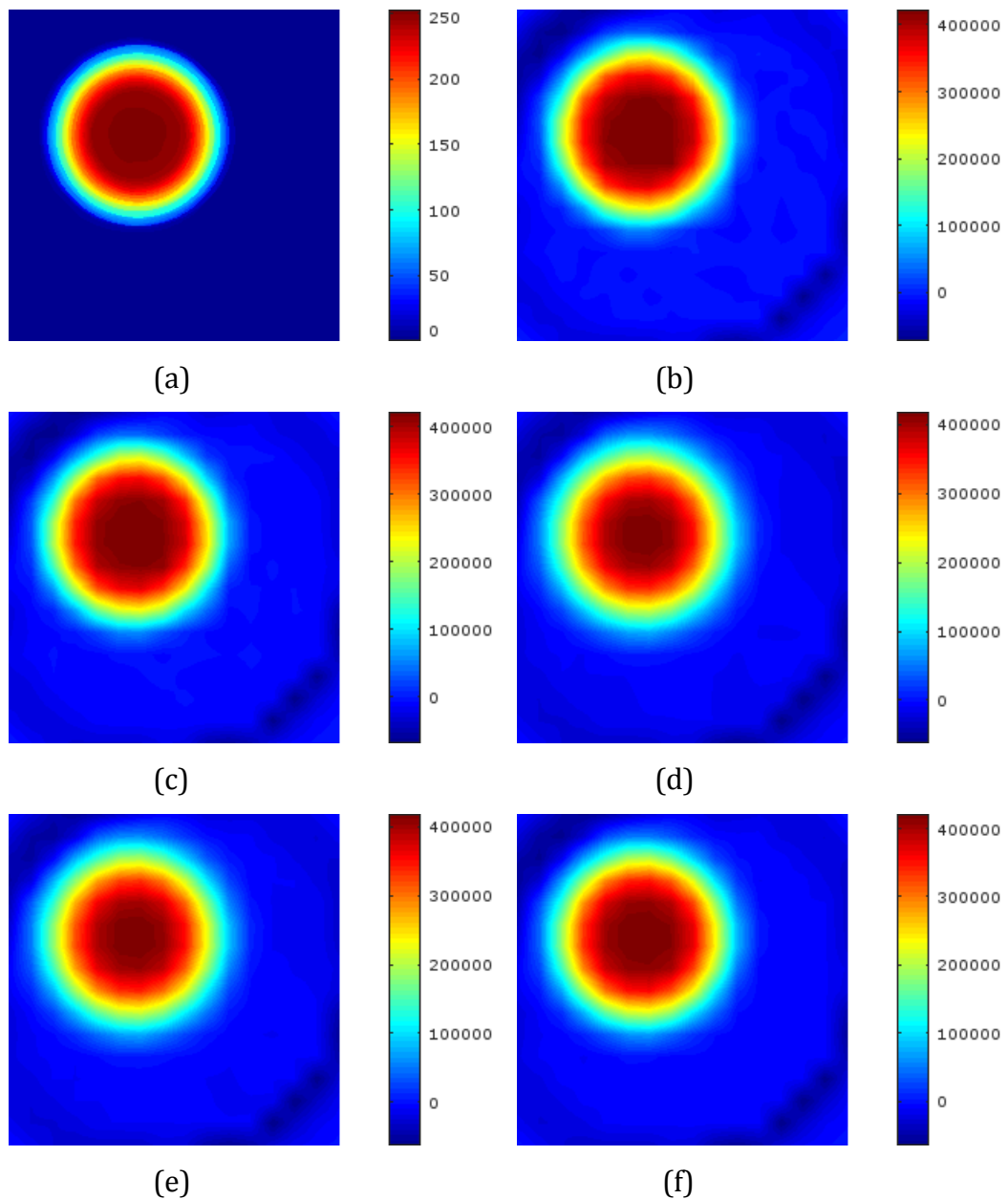
Provedené simulace rekonstrukčního algoritmu pro testovací obrázky jsou znázorněny na Obr. 21, Obr. 22, Obr. 23, Obr. 24. Obrázky jsou pro lepší přehlednost zobrazeny v barevné škále. Hodnoty na barevném panelu se v původním obrázku liší od hodnot zrekonstruovaného obrazu. To je zapříčiněno rekonstrukčním algoritmem, kde se sčítá každá nasnímaná projekce. Za vznik záporných hodnot je pak zodpovědný vybraný filtr, jehož impulsní charakteristika, která může nabývat záporných hodnot, je násobena každou projekcí.



Obr. 22 Filtrovaná zpětná projekce Obr. 19b pro různé filtry: (a) původní obraz, (b) Ram-Laken, (c) Shepp-Logan, (d) Hann, (e) Hamming, (f) Cosine

Z obrázků je patrná relativně velká míra rozostření vůči původnímu obrazu. Ta je závislá na množství snímaných pozic pro jednu projekci, tedy množství použitých drátků. Pro lepší zaostření výsledného obrazu by bylo nutné použít větší počet drátků, což by ale vedlo k nutnosti rozšířit navrhovaný systém o další měřicí kanály. Tedy rozšířit Arduino o další analogové vstupy, předělat tištěný spoj pro více kanálů a v neposlední řadě vyrobit nový detektor pro více drátků. Po konzultacích s pracovníky mikrotronové laboratoře jsme došli k závěru, že míra

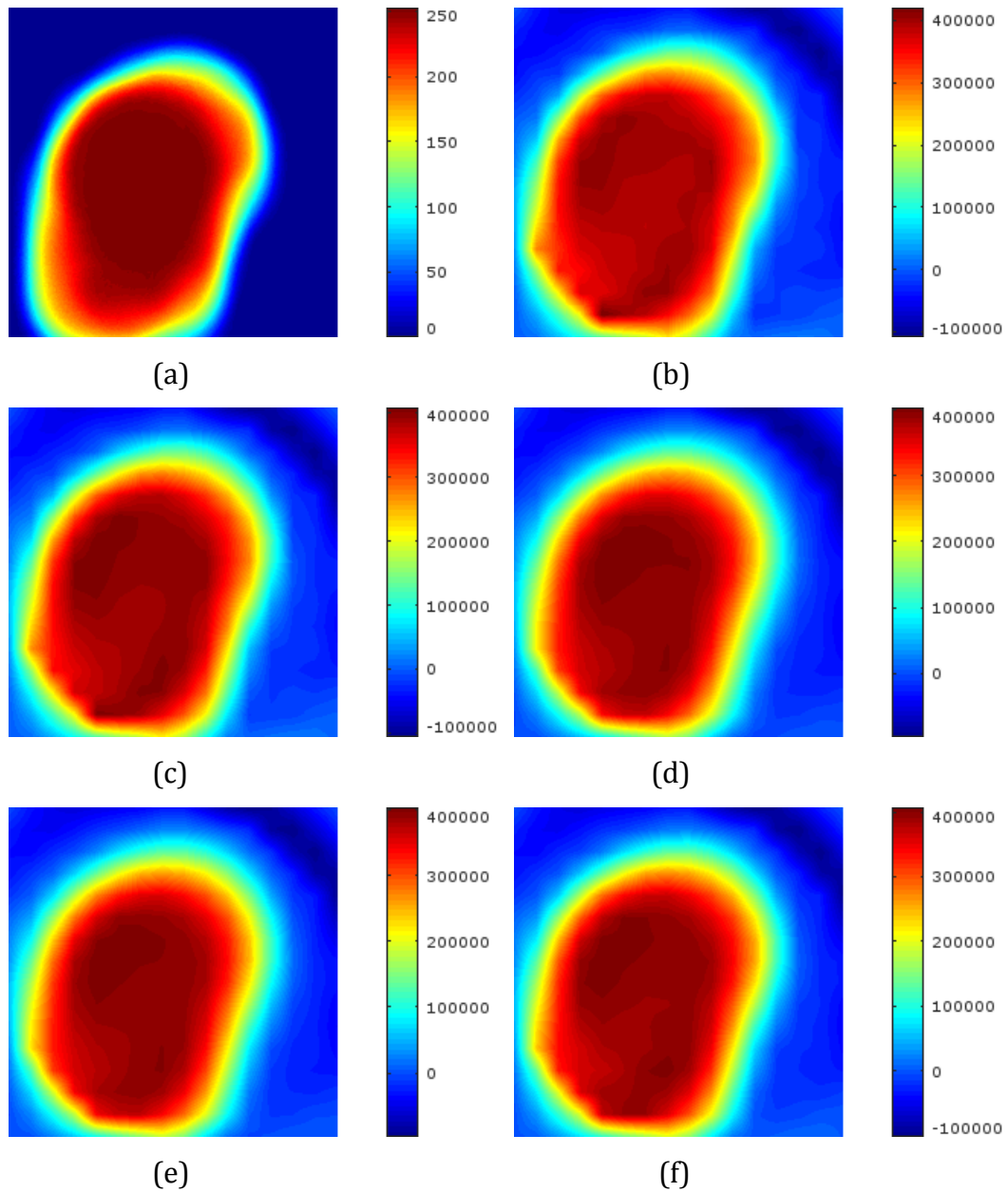
zobrazení je pro účely 2D diagnostiky dostatečná. A navržený rekonstrukční algoritmus lze použít pro praktické testování.



Obr. 23 Filtrovaná zpětná projekce Obr. 19c pro různé filtry: (a) původní obraz, (b) Ram-Lake, (c) Shepp-Logan, (d) Hann, (e) Hamming, (f) Cosine

V předchozí kapitole 3.2 Reálný systém, byl dán předpoklad, že svazek urychlených elektronů se musí nacházet v pracovní oblasti detektoru tj. ve vepsané kružnici. Na Obr. 24 je vidět důsledek porušení tohoto předpokladu. Svazek se svojí částí nachází mimo pracovní oblast (vlevo dole), což má za následek ořezání přečnickující části při rekonstrukci. Též si lze povšimnout, že takto neúplně

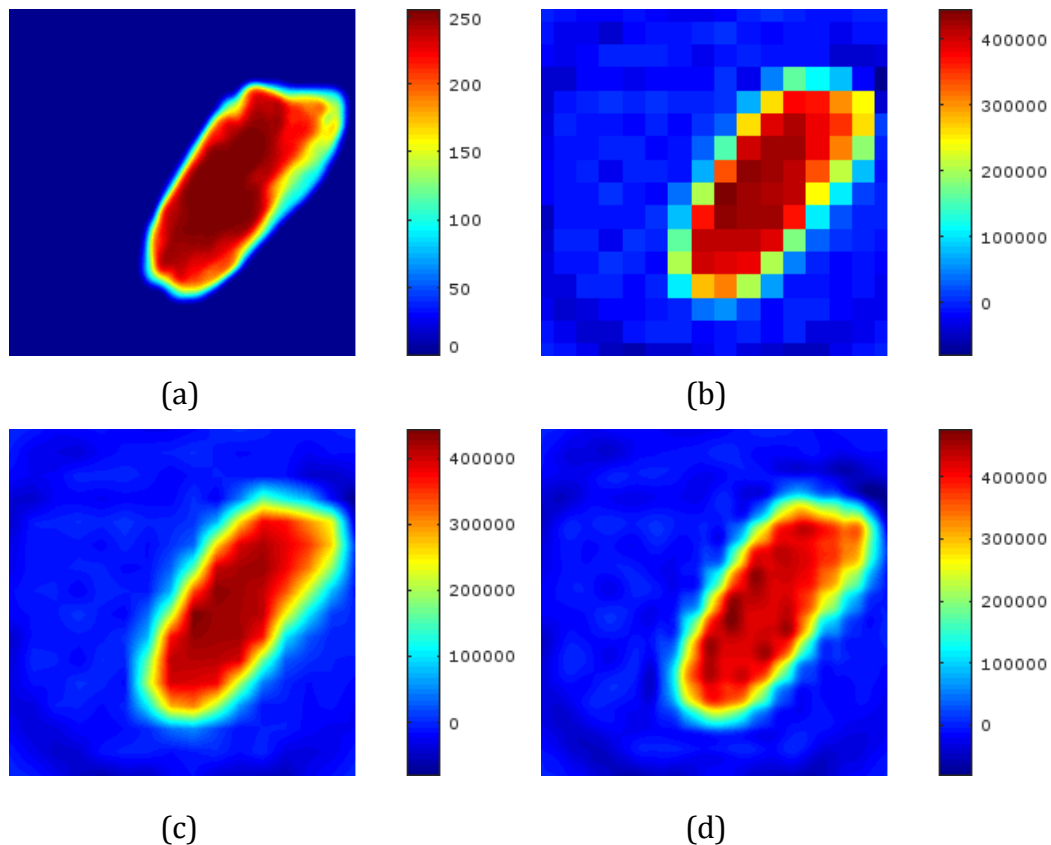
nasnímaná data mají vliv na výslednou rekonstrukci. Konkrétně pak samotné těžiště, které se při rekonstrukci ztrácí.



Obr. 24 Filtrovaná zpětná projekce Obr. 19d pro různé filtry: (a) původní obraz, (b) Ram-Laken, (c) Shepp-Logan, (d) Hann, (e) Hamming, (f) Cosine

V kapitole 3.1.2 Rekonstrukce obrazu jsem se zmínil, že používám lineární metodu interpolace. Program GNU Octave umožňuje použít tři základní metody interpolace: *'nearest'*, *'bilinear'* a *'bicubic'*. Metoda *'nearest'*, tedy metoda nejbližšího souseda spočívá v tom, že nově vzniklý pixel přebírá hodnotu od nejbližšího pixelu. Lineární metoda interpolace, ve 2D prostoru označována za bilineární, dopočítává

hodnotu nového pixelu ze čtyř sousedních hodnot. Naopak interpolace kubická resp. bikubická dopočítává hodnotu z šestnácti sousedních hodnot. Vliv použité interpolace je znázorněn na Obr. 25, jedná se o obrázek Obr. 19a zrekonstruovaný s Ram-Lakenovým filtrem. Bikubická metoda interpolace není pro navrhovaný systém vhodná, protože vycházíme ze základního obrázku o rozměrech 16x16, tedy samotná metoda nemá dostatečný počet dat pro optimální interpolaci.



Obr. 25 Vliv interpolace na rekonstruovaný obraz: (a) původní obraz, (b) metoda nejbližšího souseda, (c) bilineární metoda, (d) bikubická metoda

### 3.4 Výběr filtru

Pro výběr vhodného filtru použijeme obrázky Obr. 21, Obr. 22, Obr. 23 a Obr. 24, kde pomocí tří respondentů a jednoho pohledového kritéria provedeme prostou metodou pořadí výběr vhodného filtru. Respondenti přiřadí ke každému sérii obrázků pořadí, v jakém se jim rekonstruovaný obraz pro daný filtr nejvíce podobá původnímu obrázku. Pro každý tak tedy bude vytvořeno pořadí použití filtrů, které pak budou vyhodnoceny opět metodou pořadí mezi sebou, viz Tab. 10.



Pro jednotlivé obrázky jsem spočítal kvadratickou směrodatnou chybu, podle vzorce

$$\sigma = \sqrt{\frac{1}{N} \sum_{i=1}^N (x_i - \bar{x})^2}, \quad (3.1)$$

kde  $N$  je počet prvků matice,  $x_i$  je  $i$ -tá buňka matice vzniklá odečtem originálního obrazu od rekonstruovaného obrazu, a  $\bar{x}$  aritmetický průměr všech buněk takto vzniklé matice.

V Tab. 5 jsou vypočtené směrodatné odchylky pro jednotlivé použité filtry použité při rekonstrukci testovacích obrázků. Takto vypočítané směrodatné odchylky se nám při hodnocení projeví jako čtvrtý respondent. Určení pořadí pomocí kvadratické směrodatné odchylky se provádí z vypočtených hodnot, tedy nejnižší hodnota bude mít první pořadí, naopak nejvyšší hodnota bude mít pořadí páté.

Tab. 5 Vypočítané směrodatné odchylky rekonstrukcí testovacích obrázků

směrodatná odchylka	Ram-Laken	Shepp-Logan	Hann	Hamming	Cosine
Obr. 21	19,191	19,876	21,622	21,314	20,416
Obr. 22	23,506	23,891	25,620	25,429	24,721
Obr. 23	16,404	16,651	18,522	18,316	17,394
Obr. 24	24,985	25,367	26,659	26,527	25,898

Tab. 6 Pohledové kritérium Obr. 21

Obr. 21	Ram-Laken	Shepp-Logan	Hann	Hamming	Cosine
respondent1	5	1	4	3	2
respondent2	5	4	2	3	1
respondent3	5	1	2	4	3
respondent4	1	2	5	4	3
$\Sigma$	16	8	13	14	9
výsledné pořadí	5	1	3	4	2

Tab. 7 Pohledové kritérium Obr. 22

Obr. 22	Ram-Laken	Shepp-Logan	Hann	Hamming	Cosine
respondent1	1	3	4	5	2
respondent2	4	5	3	2	1
respondent3	3	1	5	4	2
respondent4	1	2	5	4	3
$\Sigma$	9	11	17	15	8
výsledné pořadí	2	3	5	4	1

Tab. 8 Pohledové kritérium Obr. 23

Obr. 23	Ram-Laken	Shepp-Logan	Hann	Hamming	Cosine
respondent1	5	4	3	2	1
respondent2	5	4	1	3	2
respondent3	5	4	2	3	1
respondent4	1	2	5	4	3
$\Sigma$	16	14	11	13	7
výsledné pořadí	5	4	2	3	1

Tab. 9 Pohledové kritérium Obr. 24

Obr. 24	Ram-Laken	Shepp-Logan	Hann	Hamming	Cosine
respondent1	5	4	1	2	3
respondent2	5	4	2	1	3
respondent3	5	4	1	3	2
respondent4	1	2	5	4	3
$\Sigma$	16	14	9	10	11
výsledné pořadí	5	4	1	2	3

Z Tab. 10 je patrné, že filtrem pro náš navrhovaný systém bude Cosinův filtr. Je zajímavé si povšimnout, že ačkoli si podle výpočtů směrodatných odchylek umístily nejlépe Ram-Lakenův a Shepp-loganův filtr. Z pohledového hlediska tomu už tak není.

Tab. 10 Výběr vhodného filtru

	Ram-Laken	Shepp-Logan	Hann	Hamming	Cosine
Obr. 21	5	1	3	4	2
Obr. 22	2	3	5	4	1
Obr. 23	5	4	2	3	1
Obr. 24	5	4	1	2	3
$\Sigma$	17	12	11	13	7
výsledné pořadí	5	3	2	4	1

## 4 Realizace systému

V této kapitole bude popsán postup, jakým jsem navržený systém realizoval. Nejprve se seznámíme s problematikou realizace systému, dále si popíšeme, co obnášelo řízení krokového motoru, poté si ukážeme, jak probíhá sériová komunikace mezi Arduinem a počítačem a nakonec si popíšeme si zpracování naměřených dat a tvorbu výsledného skriptu.

Při práci na realizaci systému jsem se potýkal s řadou problémů, ukázalo se, že jednočipový procesor Arduino nemá dostatečnou paměť pro proměnné, aby mohlo rekonstrukci obrazu provádět samostatně. A/D Převodník analogového vstupu Arduina má rozlišovací schopnost 10 bitů. Programově tedy jedna hodnota načtená z analogového vstupu Arduina odpovídá dvěma bytům. Při navržené matici souboru projekcí s rozměry 16x200 se tedy jedná o 6,2 kB pouze naměřených dat. Arduino má k dispozici paměť pro proměnné o velikosti 8 kB. Rozhodl jsem se tedy využít Arduino pro sběr naměřených dat a řízení krokového motoru.

Naměřená data jsem se rozhodl zpracovat v programu GNU Octave, který jsem používal pro simulace navrženého systému. Výsledkem je napsaná funkce *filtravana*, jedná se o upravenou verzi funkce *rekonstrukce*, která provádí filtrovanou zpětnou projekci naměřených dat se zvoleným filtrem a vrací obraz v dostatečném rozlišení.

GNU Octave nemá plně podporovaný nástroj pro tvorbu vizualizace a tak se výsledný skript spouští příkazem v příkazovém okně prostředí GNU Octave.

### 4.1 Řízení krokového motoru

K řízení krokového motoru nám slouží výkonový stupeň. Součástí výkonového stupně je přepínač SIP8, kterým lze nastavit mikrokrokování motoru, resp. nastavuje se počet mikrokroků na celokrok. Jak bylo zmíněno výše, budeme snímat 200 projekcí. Při celokroku 1,8 stupně musíme tedy nastavit přepínač SIP8 na dva mikrokroky na celokrok, tedy na krokování po 0,9 stupních. Dále se přepínačem nastavuje proud motoru, který jsem nechal ve výchozí pozici 1,6 A. Ještě je nutné

podotknout, že po nastavení přepínače se musí jednotka restartovat, jinak se změny neprovedou.

Výkonový stupeň je řízen třemi externími signály STEP, DIR a ENABLE, které jsou ovládány Arduinem. Arduino dává na binárních výstupech logickou úroveň v rozmezí 0 – 5V. Vstupy výkonového stupně jsou v továrním nastavení nastaveny na napěťovou úroveň 0 – 24 V. Musela se tedy vyměnit odporová kaskáda, abychom dostali napěťovou úroveň vstupů na 0 – 5 V.

Dále bylo provedeno testování rychlosti otáčení krokového motoru. Krokování bylo nastaveno programově v Arduinu na 40 ms na krok ve směru snímání projekcí a na 20 ms na krok při návratu do počáteční pozice.

Poslední věcí k řízení krokového motoru bylo zaručit, aby se otáčení krokového motoru spouštělo na náš příkaz. To řeší Arduino společně s GNU Octave přes sériovou komunikaci.

## 4.2 Sériová komunikace

Arduino Mega2560 používá pro sériovou komunikaci převodník UART. Univerzální asynchronní přijímač / vysílač UART je blok obvodů odpovědný za provádění sériové komunikace, který pomocí dvou pinů RX a TX odesílá a přijímá data. Obsahuje vlastní generátor hodinového signálu, který lze volně nastavovat a řídit a tak rychlost těchto hodin, stejně jako velikost jednoho bajtu, paritního bitu či stop bitů.

Otevření kanálu pro komunikaci se v Arduinu řeší programově příkazem `Serial.begin(speed)` ;, kde parametrem `speed` nastavujeme přenosovou rychlost v bitech za sekundu (Bd – Baud). Pro komunikaci s počítačem můžeme použít jednu z následujících voleb: 300, 600, 1200, 2400, 4800, 9600, 14400, 19200, 28800, 38400, 57600 nebo 115200. Za parametrem `speed` můžeme ještě provést nastavení velikosti délky slova. Pokud se tento parametr neuvede, bude Arduino komunikovat v přednastaveném režimu 8-N-1.

8-N-1 je běžný zkrácený zápis pro sériový port nebo pro konfiguraci v asynchronním režimu. 8 značí počet přenášených datových bitů, N znamená bez parity a 1 je jeden stop bit. Tento zápis používá i GNU Octave.

K nastavení komunikačního kanálu v prostředí GNU Octave je nejprve zapotřebí nainstalovat rozšiřující balíček *instrument-control*, který obsahuje funkce pro sériovou, ale i paralelní komunikaci. Příkazem `s = serial("\\\\.\\COMX");` se otevře komunikační kanál. COMX pak představuje port, na kterém bude kanál otevřen. Dále je potřeba nastavit komunikační rychlost a časovou prodlevu pomocí příkazů `set(s, 'baudrate', speed); a set(s, 'timeout', time);`. Nesmíme zapomenout na nastavení stejné komunikační rychlosti jak na straně počítače, tak na straně Arduina.

Nyní, když máme otevřený komunikační kanál, můžeme začít posílat data. V tomto bodě je důležité si uvědomit, že data, která budeme posílat do počítače, jsou získaná z A/D převodníku, který má rozlišení 10 bit. Budou tedy uložena v datovém typu *integer*. Je tedy na straně Arduina nutné provést rozložení posílaných dat na 8 bitová slova. Na druhé straně v prostředí GNU Octave je nutné přijatá data složit do požadovaného datového typu.

### 4.3 Program

Program v Arduinu je vykonáván cyklicky a jelikož nechceme, aby se nám motor otáčel neustále dokola, napsal jsem podmínku, která spustí řízení krokového motoru na příkaz ze sériové linky. Po nasbírání všech dat se data odešlou přes sériovou linku a motor se vrátí do počáteční polohy. Poté program opět čeká na příkaz ze sériové linky. Okomentovaný výpis programu je uveden v Příloze 7.

Pro GNU Octave jsem vytvořil funkci *filtrvana*, která provádí filtrovanou zpětnou projekci s cosinovým filtrem a vrací nám už zvětšený obrázek. Výpis funkce je uveden v Příloze 5. Dále jsem napsal funkci *prevod*, která převádí data ze sériové linky do tvaru sinogramu. A v neposlední řadě jsem napsal skript *program*, který při spuštění otevírá komunikační kanál a odesílá příkaz pro Arduino o spuštění řízení motoru a snímání dat. Poté skript čeká na příkaz od Arduina, že začíná posílat data. Po přijetí všech dat skript ukončuje sériovou komunikaci a převede načtená data na požadovaný datový typ. Poté je zavolána funkce pro převod dat na sinogram a následně funkce pro rekonstrukci obrazu. V poslední řadě zobrazí rekonstruovaný obraz profilu svazku. Okomentovaný skript *program* je uveden v Příloze 6.

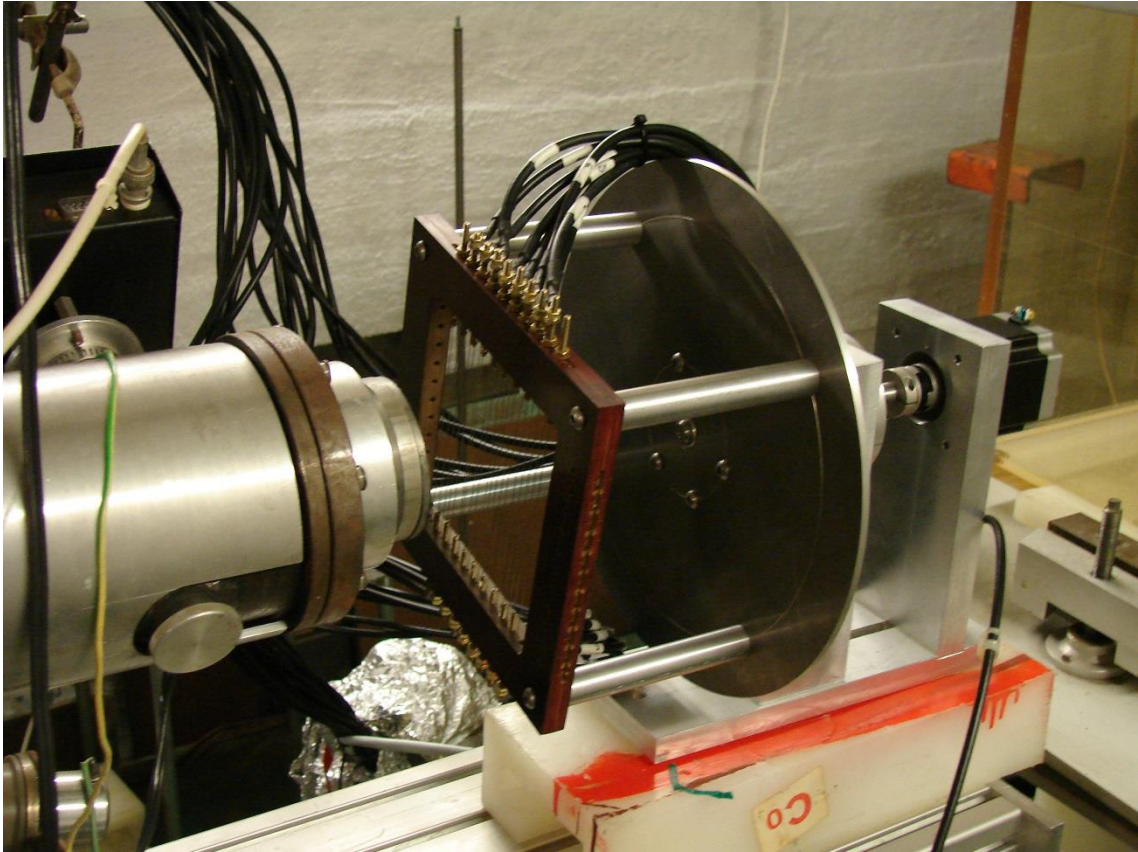
## 5 Testování navrženého systému

V této kapitole bude popsáno testování navrženého systému a seznámíme se s úskalími, která se vyskytla při testovacím provozu. Dále si ukážeme výstupy měření a provedeme diskuzi možných vlivů na výsledný rekonstruovaný obraz.

Měření probíhalo na pražském Karlíně v mikrotronové laboratoři spadající pod Ústav jaderné fyziky AV ČR v.v.i. pod dohledem odborných pracovníků mikrotronové laboratoře.

Systém byl navržen na připojení Arduina k počítači přes 14 metrový USB kabel s opakovačem. Aby cesta signálu z drátkového detektoru k Arduinu byla co nejkratší a nedocházelo ke znehodnocení signálu šumem, bylo Arduino umístěno v místnosti s mikrotronem. Po zapnutí urychlovače docházelo však na USB kabelu k velkému rušení a přenos dat neprobíhal korektně. Docházelo ke ztrátě paketů a výsledný soubor dat, tak nebyl kompletní a data, která se podařilo přenést, byla zkreslená. Jak již bylo zmíněno, A/D převodník Arduina má rozlišení 10 bitů, tedy může nabývat hodnot v intervalu od 0 do 1023. Přenesená data byla často vyšší než právě hodnota 1023. Tuto problematiku jsem vyřešil zkrácením USB kabelu na nezbytné minimum a počítač byl přesunut do místnosti s mikrotronem a následně přes ethernet kabel připojen přes vzdálenou plochu druhého počítače umístěného na velině urychlovače. Poté už přenos dat fungoval správně.

Další úskalí, které se projevilo během měření, bylo umístění počítače v místnosti mikrotronu (ozařovací kobce). Po několika měřeních se počítač začal chovat neobvykle a odmítal otevřít komunikační kanál. Nejprve jsem si myslel, že byla porucha na straně Arduina, ale nakonec se ukázala chyba na straně počítače. Předpokládalo se, že radiační pozadí, kde byl počítač umístěn, nebude mít na počítač vliv. K úspěšnému vyřešení tohoto problému pomohl restart počítače a jeho umístění za hradbu z olověných cihel (masivní olověné stínění) tak, aby byl počítač co možná nejlépe odstíněn. Pro jistotu jsem za olověnou zeď umístil i zbytek elektroniky. Poté už počítač i zbytek elektroniky pracovaly správně a bez dalších problémů.

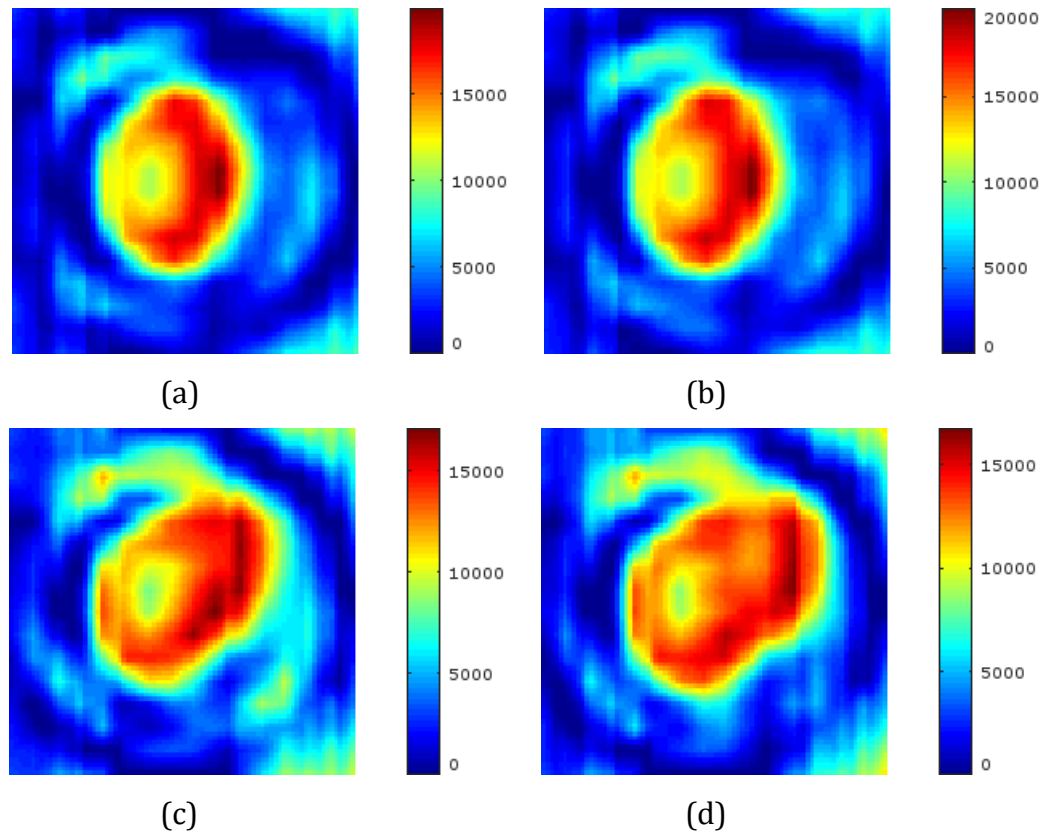


*Obr. 26 Aparatura s drátkovým detektorem a krokovým motorem u výstupního okna elektronvodu urychlovače*

## 5.1 Výstupy měření

Měření pro nízké proudy ve svazku probíhalo pro centrovaný kruhový svazek urychlených elektronů o průměru přibližně 30 milimetrů. Na Obr. 27a je zobrazena rekonstrukce výstupního svazku urychlených elektronů pro  $2,8 \mu\text{A}$  a na Obr. 27b pro  $3 \mu\text{A}$  ve svazku. Při přechodu proudu nad  $3 \mu\text{A}$  se svazek elektronů mírně vychýlil mimo osu a roztáhl se. Na Obr. 27c je proud ve svazku  $3,3 \mu\text{A}$  a na Obr. 27d je proud  $3,7 \mu\text{A}$ . Pro ověření tvaru svazku má operátor mikrotronu k dispozici kameru, která snímá ústí elektronvodu. Pokud je na ústí elektronvodu vložena folie pokrytá luminoforem, může pak operátor pozorovat tvar svazku, nikoli však jeho profil. Tvar svazku na luminoforu odpovídal tvaru získaným rekonstrukčním algoritmem. Výsledky měření ukazují, že navržený systém lze pro praktická měření využít.





Obr. 27 Rekonstrukce obrazu pořízené pro různé proudy ve svazku  
 (a)  $2,8 \mu\text{A}$ , (b)  $3 \mu\text{A}$ , (c)  $3,3 \mu\text{A}$  (d)  $3,7 \mu\text{A}$

## 5.2 Diskuze

Na rekonstruovaných obrázcích jsou jasně patrné artefakty na okrajích. Hodnoty by měli být blízké nule, čemuž odpovídá tmavě modrá barva. Tyto artefakty jsou pravděpodobně způsobeny nízkým odstupem signál-šum. Bohužel v ozařovací kobce mikrotronu je velké elektromagnetické rušení a vysoká radiace při provozu urychlovače. Oba jevy narušují korektní funkci převodníku a procesorové jednotky. Jednou možností je zajistit lepší stínění veškeré elektroniky, to je bohužel problematicky řešitelné.

Při řešení této práce jsem naštěstí měl k dispozici několik tun olověných cihel, které se použili jako stínící kryt. U elektromagnetického rušení je situace ještě horší a to z důvodu, že snímací drátky musí být z principu detekce obnažené (nestíněné) a tedy fungují jako anténa. Řešením by mělo být použití filtrů na vstupech měřicích převodníků. Tedy do budoucna je vhodné navrhnout a vyrobit novou desku tištěných spojů pro převodníky opatřenou kvalitními filtry,

kteřé ale nesmí znehodnotit měřený signál trvající přibližně 3  $\mu\text{s}$  a opakuje se s frekvencí 420 Hz.

Silné elektromagnetické rušení vzniká při spínání zdroje mikrovlnného záření, které je využito pro napájení urychlovače. Zdrojem je pulzní magnetron, který v krátkém pulzu (přibližně 4  $\mu\text{s}$ ) vygeneruje mikrovlnný puls o výkonu až 3 MW. Pulsní příkon je až 5 MW. Tedy dochází k sepnutí proudu až 100 A při napětí 50 kV s velmi krátkými náběžnými hranami (okolo 100 ns). Tento krátký puls přenášený poměrně dlouhým kabelem zároveň způsobí zakolísání rozvodné sítě 230 V na poměrně vysoké frekvenci, přes kterou je napájena citlivá elektronika měřicích převodníků. Bude tedy vhodné při úpravě převodníků myslet na pečlivou filtraci rušení přenášeného po napájecích kabelech. Tato problematika by měla být řešena pracovníky Oddělení urychlovačů Ústavu jaderné fyziky AV ČR, v.v.i. v rámci grantu, na kterém spolupracují s pracovníky Laboratoře jaderných reakcí ve Spojeném ústavu jaderného výzkumu (ОИЯИ) v Dubně v Ruské federaci.

Tato práce ukázala, že rotační drátkový detektor je schopen zobrazení profilu svazku. Otázkou je, zda rozlišení je vyhovující. Pro rutinní měření patrně ano, pro přesnou diagnostiku svazku navrhuji využití většího počtu snímacích drátků, což znamená zároveň zvýšení počtu měřicích převodníků a analogových vstupů Arduina. V tomto případě bude nutné zakoupit buď I/O shield a nebo využít zcela jinou platformu než je Arduino.

Využití detekčního systému je tedy plánováno především na elektronových urychlovačích v Ústavu jaderné fyziky a ve Spojeném ústavu jaderného výzkumu, ale je možné využít i na jiných svazcích nabitých částic. Po úpravě detekčního systému do menších rozměrů bude systém zkoušen na cyklotronech U120-M a TR 24 v Ústavu jaderné fyziky, které disponují svazky protonů a lehkých jader. Zde se také v současné době pracuje na jiném systému profilování svazku založeném na digitálním zpracování obrazu na luminoforu získaného pomocí kamery. Tento systém pak může být nápomocen k hodnocení kvality měření pomocí drátkového detektoru, které v dnešní době nejsem schopen bohužel provést.

## 6 Závěr

V rámci této diplomové práce byl navrhnout systém pro 2D diagnostiku příčného profilu svazku. Systém jsem navrhl na základě provedených simulací rekonstrukcí obrazu pomocí filtrovací zpětné projekce (inverzní Radonovy transformace).

Systém spojuje jednočipový procesor Arduino Mega2560 s programem GNU Octave, kde Arduino řídí krokový motor a odesílá naměřená data z drátkového detektoru přes sériovou linku do počítače. GNU Octave získaná data následně zpracuje rekonstrukčním algoritmem do grafické podoby a zobrazí rekonstruovaný obraz v barevné škále.

Součástí práce bylo navrhnout algoritmus pro řízení motoru Arduinem. Arduino čeká na příchozí příkaz od sériové linky a po obdržení příkazu začne být motor řízen. Motor se krokuje po 0.9 stupních v intervalu 40 ms. Po otočení motoru o 200 kroků, reálné natočení motoru je tedy 180°, Arduino zpracovává data. Po zpracování dat je motor v intervalu 20ms na krok vrácen do počáteční polohy.

Testovací provoz probíhal na pražském mikrotronu MT25 – kruhovém relativistickém urychlovači elektronů, který spadá pod Ústav jaderné fyziky AV ČR, v.v.i.. Výstupy měření jsou uvedeny v kapitole 5, kde ze zpracovaných dat je patrný vliv šumu na výsledný obraz. S pracovníky mikrotronové laboratoře byla provedena diskuze o možných příčinách rušení a došli jsme k závěru, že bude potřeba upravit elektroniku. Převodník bude potřeba doplnit o filtr na vstupu a o filtr na napájecím napětí.

V práci lze i nadále pokračovat. Arduino může být rozšířeno o další moduly, zejména o přídatnou paměť, aby mohlo provádět algoritmus rekonstrukce samostatně z naměřených dat. Dále o modul s ethernet konektorem, aby pro měření nebylo potřeba dvou počítačů. Takto rozšířený systém by pak mohl být ovládán vizualizačním softwarem.

Práce ukázala, že systém pro 2D diagnostiku příčného profilu svazku je možné využít pro praktické měření. Finální řešení systému bude provedeno v roce 2017 v rámci grantu, který se řeší ve spolupráci se Spojeným ústavem jaderného

výzkumu v Dubně – Ruská federace. V tomto grantu jsem členem řešitelského týmu.

## 7 Použitá literatura

- [1] Jan, J. *Číslicová filtrace, analýza a restaurace signálů*. Brno : VUTIUM, Brno, 2002. str. 427. ISBN 80-214-2911-9.
- [2] Drastich, A. *Tomografické zobrazovací systémy*. Skriptum. Brno : FEKT VUT, Brno, 2004. str. 208. ISBN 80-214-2788-4.
- [3] Walek, P., Lamoš, M., Jan, J. *Analýza biomedicínských obrazů - Počítačová cvičení*. Brno : Ústav biomedicínského inženýrství, VUT Brno, 2013. str. 138. ISBN 978-80-214-4792-9.
- [4] Krist, P., Bíla, J. *A mathematical model of the MT25 microtron*. JINST 6, T10005, 2011.
- [5] Krist, P. *Řídicí systém mikrotronu - kruhového relativistického urychlovače elektronů*. Praha : Ústav přístrojové a řídicí techniky, ČVUT Praha, 2012. str. 115. Disertační práce, školitel Bíla, J.
- [6] Ulmann, V. *Jaderná fyzika a fyzika ionizujícího záření - kruhové urychlovače*. [Online] 2008. [Citace: 4. 1 2017.] Dostupné z: <http://astronuklfyzika.cz/JadRadFyzika5.htm#KruhUrychlovace>.
- [7] Voda, Z. & tým HW Kitchen. *Průvodce světem Arduina*. 2015. ISBN 978-80-87106-90-7.
- [8] Window function. *Wikipedia*. [Online] 3. 1 2017. [Citace: 4. 1 2017.] Dostupné z: [https://en.wikipedia.org/wiki/Window\\_function](https://en.wikipedia.org/wiki/Window_function).
- [9] Department of accelerator. *The Czech Academy of Sciences*. [Online] (c) 2010. [Citace: 2. 1 2017.] Dostupné z: <http://accs.ujf.cas.cz/>.

## Seznam obrázků

Obr. 1	Mikrotron MT25 – kruhový relativistický urychlovač elektronů.....	7
Obr. 2	Schéma mikrotronu MT25.....	8
Obr. 3	Krokové motory řady SX23 .....	9
Obr. 4	Výkonový stupeň SD20x.....	10
Obr. 5	Drátkový detektor.....	11
Obr. 6	Elektrický obvod převodníku pro jeden kanál.....	12
Obr. 7	Arduino Mega2560 .....	14
Obr. 8	Princip vzniku projekce .....	16
Obr. 9	Vliv pozice bodu na amplitudu harmonické funkce.....	17
Obr. 10	Principiální vysvětlení projekčního teorému.....	19
Obr. 11	Princip rekonstrukce prostřednictvím frekvenční oblasti.....	20
Obr. 12	Hvězdicový artefakt při použití 10 projekcí .....	22
Obr. 13	Shepp-Logan phantom .....	27
Obr. 14	Soubory projekcí.....	27
Obr. 15	Rekonstrukce obrazu ze 180 projekcí.....	28
Obr. 16	Rekonstrukce obrazu z 90 projekcí.....	29
Obr. 17	Rekonstrukce obrazu z 36 projekcí.....	29
Obr. 18	Zpětná prostá projekce .....	30
Obr. 19	Testovací obrázky .....	32
Obr. 20	Soubory projekcí testovacích obrázků.....	32
Obr. 21	Filtrovaná zpětná projekce Obr. 19a pro různé filtry .....	33
Obr. 22	Filtrovaná zpětná projekce Obr. 19b pro různé filtry .....	34
Obr. 23	Filtrovaná zpětná projekce Obr. 19c pro různé filtry.....	35
Obr. 24	Filtrovaná zpětná projekce Obr. 19d pro různé filtry .....	36
Obr. 25	Vliv interpolace na rekonstruovaný obraz.....	37
Obr. 26	Aparatura s drátkovým detektorem a krokovým motorem .....	45
Obr. 27	Rekonstrukce obrazu pořízené pro různé proudy ve svazku .....	46

## Seznam tabulek

Tab. 1	Hlavní parametry mikrotronu MT25.....	9
Tab. 2	Technické parametry krokového motoru SX23-1414 .....	10
Tab. 3	Parametry výkonového stupně.....	10
Tab. 4	Technické specifikace Arduina Mega2560 .....	13
Tab. 5	Vypočítané směrodatné odchyly rekonstrukcí testovacích obrázků....	38
Tab. 6	Pohledové kritérium Obr. 22 .....	38
Tab. 7	Pohledové kritérium Obr. 23 .....	39
Tab. 8	Pohledové kritérium Obr. 24 .....	39
Tab. 9	Pohledové kritérium Obr. 25 .....	39
Tab. 10	Výběr vhodného filtru .....	40

## Seznam příloh

Příloha 1.....	I
Příloha 2.....	II
Příloha 3.....	III
Příloha 4.....	IV
Příloha 5.....	V
Příloha 6.....	VI
Příloha 7.....	VII



## Příloha 1

Okomentovaný výpis funkce *sinogram.m* v prostředí GNU Octave

```
1 function [SouborProjekci]=sinogram(I, theta)
2 % Funkce rekonstrukce vrací Soubor Projekci (sinogram) použitého
3 % obrázku pro zadaný krok snímání projekcí.
4
5 PocetProjekci = length(theta);
6
7 % Simulování pozic drátků
8 Dratky =
   round((length(I(:,1))/32):length(I(:,1))/16:length(I(:,1))-1);
9
10 % Vytvoření prázdné matice tzv. Radonův prostor.
11 SouborProjekci = zeros(16,PocetProjekci);
12
13 % Snímání jedné projekce na jeden cyklus.
14 for i = 1 : PocetProjekci
15 Irot = imrotate(I, theta(i), 'bilinear', 'crop');
16 SouborProjekci(:,i) = sum(Irot(Dratky,:),2);
17 end
18
19 endfunction
```

## Příloha 2

Okomentovaný výpis funkce *rekonstrukce.m* v prostředí GNU Octave

```
1 function [Obraz]=rekonstrukce(SouborProjekci, theta, filtr)
2 % Funkce rekonstrukce vrací matici rekonstruovaného obrazu.
3
4 PocetProjekci = length(theta);
5
6 % Filtrace Souboru Projekcí (sinogramu) zadaným filtrem.
7 Filtrace = rho_filter(SouborProjekci, filtr);
8
9 % Vytvoření prázdné matice, do které se vkládají jednotlivé
10 % rekonstruované projekce.
11 Obraz = zeros(16,16);
12
13 % Rekonstrukce jedné projekce ze sinogramu za jeden cyklus.
14 for i = 1 : PocetProjekci
15 Rekonstrukce = repmat(Filtrace(:,i)', 16, 1);
16 Rekonstrukce = imrotate(Rekonstrukce, 270-theta(i), 'bilinear',
'crop');
17 Obraz = Obraz + Rekonstrukce(:,,:);
18 end
19
20 endfunction
```

## Příloha 3

Okomentovaný výpis skriptu *simulace.m* v prostředí GNU Octave

```
1 % Skript pro filtrovanou zpětnou projekci.
2 close all;
3 clear all;
4 clc;
5
6 % Načtení souborů pro práci s obrázky.
7 pkg load image;
8
9 % Načtení obrázku, který budeme rekonstruovat.
10 I = imread('Obr1.png');
11
12 % Převod na obrázek ve stupních šedi.
13 I = rgb2gray(I);
14
15 % Krok úhlu, pod kterým budeme snímat obrázek.
16 theta = 0 : 0.9 : 179.999;
17
18 % Volání funkce sinogram.
19 SouborProjekci = sinogram(I, theta);
20
21 % Volání funkce rekonstrukce.
22 Obraz = rekonstrukce(SouborProjekci, theta, 'Cosine');
23
24 % Zvětšení rozměrů rekonstruovaného obrazu.
25 ObrazZvetseny = imresize(Obraz, [216 216], 'bilinear');
26
27 % Zobrazení rekonstruovaného obrazu v barevné škále.
28 figure;
29 imshow(ObrazZvetseny, []);
30 colormap(jet);
```

## Příloha 4

Okomentovaný výpis funkce *prevod.m* v prostředí GNU Octave

```
1 function [VsData]=prevod(DataSerial)
2 % Funkce prevod převádí data ze sériové linky do tvaru
3 % sinogramu (matice 16x200).
4
5 % Vytvoření prázdných matic pro data.
6 VsData = zeros(16,200);
7 Sloupec = zeros(1,16);
8
9 % Převod 16 dat do sloupce matice sinogramu za jeden cyklus.
10 for i = 1 : 200
11 S1 = ([DataSerial(:,0+i),
12 DataSerial(:,200+i),
13 DataSerial(:,400+i),
14 DataSerial(:,600+i),
15 DataSerial(:,800+i),
16 DataSerial(:,1000+i),
17 DataSerial(:,1200+i),
18 DataSerial(:,1400+i),
19 DataSerial(:,1600+i),
20 DataSerial(:,1800+i),
21 DataSerial(:,2000+i),
22 DataSerial(:,2200+i),
23 DataSerial(:,2400+i),
24 DataSerial(:,2600+i),
25 DataSerial(:,2800+i),
26 DataSerial(:,3000+i)]);
27 VsData (1:16,i) = Sloupec;
28 end
29
30 endfunction
```

## Příloha 5

Okomentovaný výpis funkce *filtravana.m* v prostředí GNU Octave

```
1 function [ObrVysledny]=filtravana(Sinogram)
2 % Funkce filtravana vrací zvětšený rekonstruovaný obraz.
3
4 % Načtení souborů pro práci s obrázky.
5 pkg load image;
6
7 % Definice počtu snímání (četnost sběru dat).
8 UhelTheta = 0 : 0.9 : 179;
9 PocetProjekci = length(UhelTheta);
10
11 % Filtrace naměřených dat.
12 Filtrace = rho_filter(Sinogram, 'Cosine');
13
14 % Vytvoření prázdné matice, do které se vkládají jednotlivé
15 % rekonstruované projekce.
16 ObrRekonstrukce = zeros(16,16);
17
18 % Rekonstrukce jedné projekce ze sinogramu za jeden cyklus.
19 for i = 1 : PocetProjekci
20 Rekonstrukce = repmat(Filtrace(:,1)', 16, 1);
21 Rekonstrukce = imrotate(Rekonstrukce, 270-UhelTheta(i), 'bilinear',
'crop');
22 ObrRekonstrukce = ObrRekonstrukce + Rekonstrukce(:,,:);
23 end
24
25 % Zvětšení rozměrů rekonstruovaného obrazu.
26 ObrVysledny = imresize(ObrRekonstrukce, 5, 'bilinear');
27
28 endfunction
```

## Příloha 6

Okomentovaný výpis skriptu *program.m* v prostředí GNU Octave

```
1 % Skript GNU Octave sloužící pro navázání komunikace s Arduinem
2 % a provedením filtrované zpětné projekce naměřených dat.
3 clc;
4
5 % Načtení souborů pro možnosti komunikace.
6 pkg load instrument-control
7
8 % Otevření kanálu pro seriovou komunikaci.
9 s = serial("\\\\.\\COM4");
10 set(s, 'baudrate', 9600);
11 set(s, 'timeout', 100);
12
13 % Vymazání zásobníku seriové linky.
14 srl_flush(s);
15
16 % Vyslání příkazu pro Arduino.
17 srl_write(s, uint8 (1));
18 srl_flush(s);
19
20 % Čekání na odpověď Arduina, že posílá data.
21 j = 0;
22 while (j == 0)
23 j = srl_read(s,1);
24 if (j > 0)
25 j = 1;
26 else
27 j = 0;
28 endif
29 endwhile
30
31 % Načtení dat ze zásobníku.
32 DataArduino = srl_read(s,6400);
33
34 % Převod dat na datový typ int16.
35 DataSerial=typecast(uint8(DataArduino), 'int16');
36
37 % Zavření komunikačního kanálu.
38 fclose(s);
39 clear s;
40
41 % Volání funkce převod.
42 Sinogram = převod(DataSerial);
43
44 % Volání funkce filtrovana.
45 Obr = filtrovana(Sinogram);
46
47 % Zobrazení rekonstruovaného obrazu v barevné škále.
48 figure;
49 imshow(Obr, []);
50 colormap(jet);
```

## Příloha 7

### Okomentovaný výpis programu pro Arduino

```
// Deklarace proměných
const int OutputStep = 53;
const int OutputDirection = 52;
const int OutputEnable = 51;

int incomingByte = 0;
int OutputData[16][200];
int InputValue[16];
int i, j, k;

int InputA0 = A0;
int InputA1 = A1;
int InputA2 = A2;
int InputA3 = A3;
int InputA4 = A4;
int InputA5 = A5;
int InputA6 = A6;
int InputA7 = A7;
int InputA8 = A8;
int InputA9 = A9;
int InputA10 = A10;
int InputA11 = A11;
int InputA12 = A12;
int InputA13 = A13;
int InputA14 = A14;
int InputA15 = A15;

void setup() {
// Nastavení a spuštění sériové komunikace
Serial.begin(9600);
// Nastavení výstupních pinů
pinMode(OutputStep, OUTPUT);
pinMode(OutputDirection, OUTPUT);
pinMode(OutputEnable, OUTPUT);
}

void loop() {
// Čekání na příkaz od sériové linky
if (Serial.available() > 0)
{
    incomingByte = Serial.read();
    // Nastavení směru a běhu krokového motoru
    digitalWrite(OutputEnable, HIGH);
    digitalWrite(OutputDirection, HIGH);
    // Načtení dat z analogových vstupů na krok motoru
    for (j = 0; j < 200; j++)
    {
        InputValue[1] = analogRead(InputA1);
        InputValue[2] = analogRead(InputA1);
        InputValue[3] = analogRead(InputA2);
        InputValue[4] = analogRead(InputA3);
        InputValue[5] = analogRead(InputA4);
        InputValue[6] = analogRead(InputA5);
    }
}
```

```
    InputValue[7] = analogRead(InputA7);
    InputValue[8] = analogRead(InputA7);
    InputValue[9] = analogRead(InputA8);
    InputValue[10] = analogRead(InputA9);
    InputValue[11] = analogRead(InputA10);
    InputValue[12] = analogRead(InputA11);
    InputValue[13] = analogRead(InputA12);
    InputValue[14] = analogRead(InputA13);
    InputValue[15] = analogRead(InputA14);
    InputValue[16] = analogRead(InputA15);
    // Zápis naměřených dat do paměti
    for (i = 0; i < 16; i++)
    {
        OutputData[i][j] = InputValue[i+1];
    }
    // Krokování motoru
    digitalWrite(OutputStep, HIGH);
    delay(20);
    digitalWrite(OutputStep, LOW);
    delay(20);
}
// Příkaz pro Octave o posílání dat
Serial.write(1);
// Uprava dat a jejich odeslání
Serial.write((uint8_t*)OutputData, sizeof(OutputData));
// Navrácení motoru do výchozí polohy
for (k = 0; k < 200; k++)
{
    digitalWrite(OutputDirection, LOW);
    digitalWrite(OutputStep, HIGH);
    delay(10);
    digitalWrite(OutputStep, LOW);
    delay(10);
}
}
```