

Diplomová práce



České  
vysoké  
učení technické  
v Praze

**F3**

Fakulta elektrotechnická  
Katedra řídicí techniky

## Poradní systém pro detekci klepání motoru

Advisory system for engine knock detection

Bc. Miroslav Kubíček

Vedoucí: prof. Ing. Pavel Sovka, CSc.

Obor: Systémy a řízení.

Studijní program: Systémy a řízení.

Leden 2018



## I. OSOBNÍ A STUDIJNÍ ÚDAJE

Příjmení: **Kubíček** Jméno: **Miroslav** Osobní číslo: **393177**  
Fakulta/ústav: **Fakulta elektrotechnická**  
Zadávající katedra/ústav: **Katedra řídicí techniky**  
Studijní program: **Kybernetika a robotika**  
Studijní obor: **Systémy a řízení**

## II. ÚDAJE K DIPLOMOVÉ PRÁCI

Název diplomové práce:

**Poradní systém pro detekci klepání motoru**

Název diplomové práce anglicky:

**Advisory system for engine knock detection**

Pokyny pro vypracování:

1. Prostudujte a popište jev klepání motoru
2. Popište základní přístupy k jeho detekci
3. Implementujte ve vhodném jazyce (Java, C, C#,...) program pro vyhodnocování a zobrazování potřebných charakteristik
4. Kód implementujte tak, aby jej bylo možné použít při online vyhodnocování v průběhu práce s motorem, i při off-line vyhodnocování záznamů
5. Proveďte validaci programu a diskutujte dosažené výsledky

Seznam doporučené literatury:

- [1] B. G. Quinn and E. J. Hannan: The estimation and tracking of frequency. Cambridge University Press, 2001  
[2] W. J. Fitzgerald and all: Nonlinear and nonstationary signal processing. Cambridge University Press, 2000

Jméno a pracoviště vedoucí(ho) diplomové práce:

**prof. Ing. Pavel Sovka, CSc., katedra teorie obvodů FEL**

Jméno a pracoviště druhé(ho) vedoucí(ho) nebo konzultanta(ky) diplomové práce:

Datum zadání diplomové práce: **10.10.2017** Termín odevzdání diplomové práce: **09.01.2018**

Platnost zadání diplomové práce: **30.09.2019**

\_\_\_\_\_  
Podpis vedoucí(ho) práce

\_\_\_\_\_  
Podpis vedoucí(ho) ústavu/katedry

\_\_\_\_\_  
Podpis děkana(ky)

## III. PŘEVZETÍ ZADÁNÍ

Diplomant bere na vědomí, že je povinen vypracovat diplomovou práci samostatně, bez cizí pomoci, s výjimkou poskytnutých konzultací. Seznam použité literatury, jiných pramenů a jmen konzultantů je třeba uvést v diplomové práci.

\_\_\_\_\_  
Datum převzetí zadání

\_\_\_\_\_  
Podpis studenta



## Poděkování

Děkuji mému vedoucímu práce panu profesorovi Ing. Pavlu Sovkovi, CSc. za cenné rady a ochotu při psaní této diplomové práce. Chtěl bych poděkovat mé rodině za podporu a především Aleksandru Simakovovi a dalším přátelům a spolužákům z ČVUT, s kterými jsem si užil spoustu zábavy a poznal zajímavé věci.

Speciální poděkování patří mé přítelkyni, která za mnou stála za každé situace po celou dobu mého magisterského studia.

## Prohlášení

Prohlašuji, že jsem předloženou práci vypracoval samostatně, a že jsem uvedl veškeré informační zdroje v souladu s Metodickým pokynem o dodržování etických principů při přípravě vysokoškolských závěrečných prací.

V Praze, 9. ledna 2018

.....

## Abstrakt

Tato práce se zabývá jevem klepání u zážehových motorů. Jedná se o jev, který vznikne "špatným" spálením směsi ve spalovací komoře motoru, které může vést až k jeho zničení. Rozkmitání pístu v důsledku tohoto jevu je na relativně stálé frekvenci, neboť je dáno především jeho mechanickými rozměry. Cílem této práce je klepání a jeho frekvenci detekovat pomocí akcelerometrů namontovaných poblíž spalovací komory. Součástí této práce je knihovna poradního systému s programem, který jí volá. To umožňuje zobrazení frekvenčních charakteristik potřebné k detekci klepání a jeho frekvence.

**Klíčová slova:** klepání, klepání motoru, FFT, STFT, Poradní systém

**Vedoucí:** prof. Ing. Pavel Sovka, CSc.

## Abstract

This thesis is focused on the phenomenon of knocking in spark-ignition engines. Knocking is caused by the "bad" burning of the air-fuel mixture in the combustion chamber of the engine. It can lead to its destruction. Oscillation of a piston caused by knocking appears on relatively constant frequency because it is mainly due to its mechanical dimensions. The object of this thesis is to detect the knocking and its frequency using accelerometers mounted near the combustion chamber. The main part of this thesis is the library of the advisory system with the program that calls it. This allows it to display the frequency characteristics required to detect knocking and its frequency.

**Keywords:** knock, engine knocking, FFT, STFT, Advisory system

## Obsah

<b>1 Úvod</b>	<b>1</b>	<b>4 Detekce klepání</b>	<b>15</b>
1.1 Cíl práce a motivace	1	4.1 Odhad klepací frekvence	16
1.2 Struktura práce	2	4.2 Možné algoritmy detekce klepání a zjištění klepací frekvence	17
<b>Část I Teoretická část</b>		<b>Část II Praktická část</b>	
<b>2 Jev klepání motoru</b>	<b>5</b>	<b>5 Laboratorní měření klepání motoru</b>	<b>21</b>
2.1 Předčasné vznícení / pre-ignition	6	5.1 Cíl měření	21
2.2 Dělení klepání podle typu	7	5.2 Průběh měření dat	22
2.3 Snímače klepání motoru	9	5.3 Digitalizace naměřených dat	24
2.3.1 Přímé měření	9	<b>6 Návrh poradního systému</b>	<b>25</b>
2.3.2 Nepřímé měření	10	6.1 Načtení a předzpracování dat	25
<b>3 Komplexní transformace</b>	<b>11</b>	6.1.1 Au formát [15]	25
3.1 Spojitá Fourierova transformace	11	6.1.2 Wave formát 16bit PCM [16]	27
3.2 Diskrétní Fourierova transformace (DFT)	12	6.1.3 16-bitová pulzně kódová modulace	28
3.3 Krátkodobá Fourierova transformace (STFT)	13	6.2 Zpracování dat	28
3.4 Vlnková transformace (WT)	14	6.2.1 Filtrace dat a nastavení mezí	28
		6.2.2 Výběr vhodné transformace	29

6.2.3 Rozdělení naměřených dat . . .	30
6.3 Statistiky . . . . .	32
<b>7 Implementace Poradního systému v C#</b>	<b>35</b>
7.1 Struktura programu . . . . .	35
7.1.1 Popis komponent programu .	36
7.2 Diagram tříd knihovny . . . . .	37
7.3 Možné vzniklé chybové hlášení .	42
7.4 Postup pro správné volání knihovny . . . . .	42
<b>8 Dosažené výsledky</b>	<b>43</b>
8.1 Experimenty na reálných datech	43
<b>9 Závěr (Rozšíření práce)</b>	<b>47</b>
9.1 Rozšíření práce . . . . .	48
<b>Literatura</b>	<b>49</b>

## Přílohy



## Obrázky

2.1 Průběh tlaku a teploty při klepání + vysvětlující obrázek klepání [4]...	6	7.2 Popis prvků programu . . . . .	38
2.2 Křivka pro rozhraní deflagrace a detonace [5] . . . . .	8	7.3 Diagram tříd pro poradní systém	39
2.3 Průběh zapálení směsi, který obsahuje předčasné vznícení, deflagraci a detonaci [4] . . . . .	9	8.1 Experiment na offline data diesel/ Online data 65% plyn snímač A ..	44
2.4 Přehled typů klepání [4] . . . . .	10	8.2 Experiment na klepajících datech D-20_11-04-2012_17-59- 07_#05.1_knocking.au . . . . .	45
4.1 Režimy vibrace válce [11] . . . . .	17	1 Detekce klepání pomocí experimentální metoda porovnání změny výkonových spekter v čase v MATLABU - Neklepající data . . . .	53
4.2 Dominance jednotlivých módů [10] . . . . .	18	2 Detekce klepání pomocí experimentální metoda porovnání změny výkonových spekter v čase v MATLABU - Klepající data . . . . .	53
5.1 Dieselový motor, pohled na snímače poblíž válců . . . . .	22		
5.2 Dieselový motor . . . . .	23		
6.2 Chyby 1. a 2. typu . . . . .	29		
6.3 Vybraná data z cyklu, okno 512, zpoždění 1150 . . . . .	31		
6.4 Zpracování a porovnání histogramů . . . . .	31		
7.1 Struktura programu . . . . .	36		

## Tabulky

6.1 Au formát, přehled významů jednotlivých bytů . . . . .	26
6.2 Wave formát, přehled významů jednotlivých bytů . . . . .	27

# Kapitola 1

## Úvod

### 1.1 Cíl práce a motivace

Spalovací motory, které jsou dnes a denně používány v především průmyslové a automobilové sféře, přináší drobné komplikace. Jde o jev, který se nazývá klepání (anglicky engine knocking) a vzniká při špatném vznícení paliva ve spalovací komoře válci. Slovem „špatném“ se rozumí, že palivo je zapáleno samovolně. Není zapálen postupujícím plamenem od jiskry ze zapalovací svíčky. To vede k mechanickému vychýlení pístu a po určitém čase může vést i zadření motoru. Detekcí klepání a následným řízením se zabývají články [1] a [2]. Na to zda dojde ke klepání, má velký vliv časové zapálení svíčky. Časové zapálení svíčky můžeme optimalizovat podle několika kritérií. To jsou například minimalizace spotřeby paliva, maximální točivý moment motoru nebo minimalizace oxidu dusíku  $NO_x$ , oxidu uhelnatého a dalších. Bohužel právě v těchto optimálních bodech musíme nejčastěji počítat s jevem klepání, který nám daná optima znepříjemňuje. To vede k tomu, že se pak pohybujeme na hranici optimalizace s výskytem klepání. Dojde-li ke vzniku klepání, následuje generace většího množství výfukových plynů a snížení točivého momentu motoru. Detekci samotného klepání můžeme zjistit právě za pomoci měření kvality spalování lambda sondou, detekce změny tlaku ve válci motoru nebo jako v našem případě pomocí piezoelektrických snímačů. Tyto signály jsou posílány do ECU (Engine Control Unit), která provede akční zásah například změnou poměru směsi nebo časové zapálení směsi k utlumení klepání. Klepání jako takové generuje specifický, pro některé zvuk typu udeření „kov na kov“ a je způsoben rezonancí vnitřní spalovací komory. Lze jí tedy určit z mechanické konstrukce. Pro správné fungování ECU je

nutné klepací frekvenci znát. Cílem této práce je navrhnout poradní systém, který bude zpracovávat signál z piezoelektrických snímačů pozicovaných poblíž válce motoru a odhadovat frekvenci na které ke klepání dochází. Dále zobrazení frekvenčních charakteristik a jejich statistik, které pomohou uživateli poradního systému určit nastavující parametr klepací frekvence.

Dodatek: Zdroje pro tuto diplomovou práci jsou čerpány, především z anglické literatury a v textu se mohou objevit nepřesnosti v důsledku překladu, za které se omlouvám. Například flame front, překládám jako ohnivá vlna. Jedná se o krajní část hořící směsi při zapálení svíčkou. Ta se postupně rozšiřuje, až prohoří směs v celém válci. Dále v práci nerozeznávám anglická slova „knock“ a „knocking“, které obě překládám jako klepání. Jedno však znamená, že se koukáme pouze na jeden cyklus a druhý obecný stav motoru, kde dochází k tomuto jevu.

## 1.2 Struktura práce

V práci se nejprve podíváme nejprve na to co vlastně klepání je a jak vzniká. Poté se budeme zabývat jeho detekcí, na které budeme určitě potřebovat nějaké teoretické znalosti o frekvenčních transformacích. Po teoretických kapitolách se dostaneme k samotné praktické části. Popíšeme si postup, jak jsme změřili data na motoru, který jsme se snažili rozklepat a které budeme později analyzovat. Následně popíšeme program poradního systému pro detekci klepání, jeho třídy a způsob detekce klepací frekvence. Toto řešení pak následně shrneme v poslední části a budeme diskutovat nad dosaženými výsledky.



# Část I

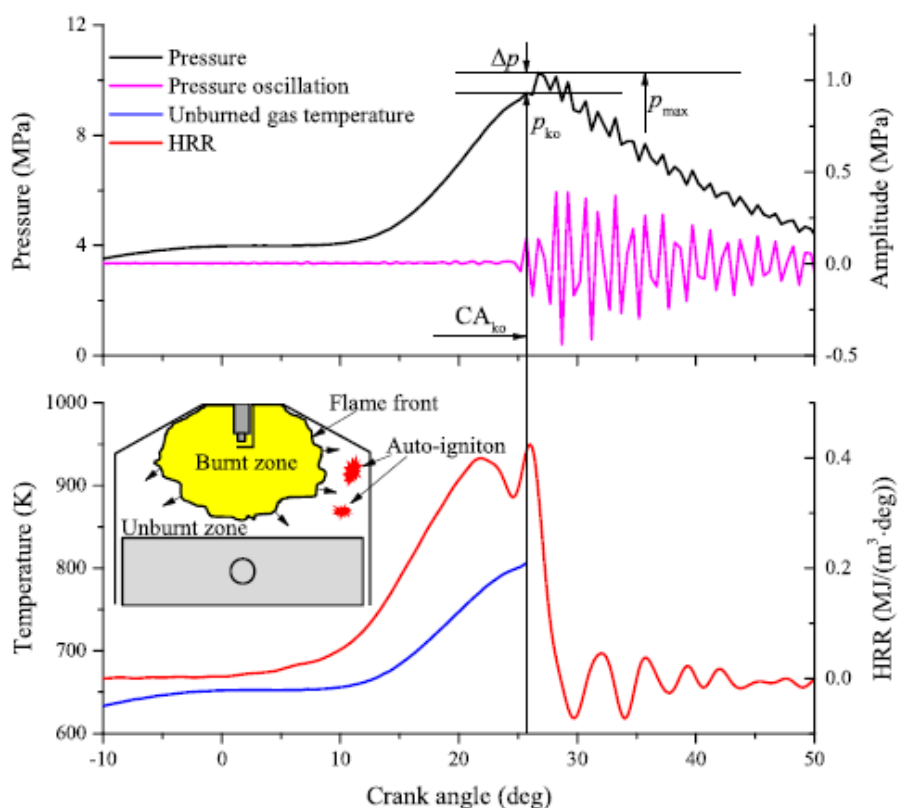
## Teoretická část

## Kapitola 2

### Jev klepání motoru

Jev klepání vzniká různými způsoby a má za následek rozkmitání pístu ve válci důsledkem špatného spálení paliva. Klepání vzniká především u spalovacích motorů, kde může mít destruktivní následky. U dieselových motorů tento jev také může nastat, avšak vzniká za jiných podmínek. Zde se budeme zabývat především klepáním ve spalovacích motorech. Běžný průběh fáze výbuchu v cyklu spalovacího motoru začíná jiskrou ve svíčke a následným zapálením směsi ve spalovací komoře. Jiskra zapaluje někdy kolem  $10^\circ$  až  $40^\circ$  klikové hřídele před horní úvratí pístu. To vede k vytvoření plamene ve válci, při kterém směs kompletně vyhoří. V další fázi je vyhořelá směs vyfouknuta ven z pístu.

Klepání motoru je jev, který vzniká při samovolném vznícení směsi uvnitř spalovací komory válce. Samovolné vznícení vznikne působením vysokého tlaku a teploty na okolní nevyhořelou část směsi. Vyšší tlak je způsoben rozšiřující se ohnivou vlnou od jiskry svíčky (Obrázek 2.1 dole). Vznikne-li samovolné vznícení směsi, vytvoří se ohnivá vlna jdoucí proti pohybu hlavní vlny. To způsobí oscilace tlaku uvnitř spalovací komory, které poté rozklepou s pístem. Změny tlaku můžeme vidět na Obrázku 2.1. Klepání má za následek větší opotřebení pístu, a pokud řídicí jednotka nezasáhne, může dojít k zadření nebo i prasknutí válce motoru. Klepání je stochastický jev, a tedy pokud nastane v jednom cyklu, neznamená to, že v následujícím cyklu by měl nastat také. Klepání se také častěji vyskytuje v místech s vyšší okolní teplotou. Výskyt klepání je závislý na druhu a struktuře paliva. Některá paliva mají tendenci podléhat klepání více než jiná. Pokud ke klepání dojde, můžeme zasáhnout několika způsoby: [3]



**Obrázek 2.1:** Průběh tlaku a teploty při klepání + vysvětlující obrázek klepání [4]

- Použití paliva s vyšším oktanovým číslem, ta zvyšuje teplotu spalování paliva
- Obohacení poměru vzduch-palivo, ta naopak snižuje teplotu spalování paliva
- Snižením tlaku/teploty ve válci
- Snižením tlaku v potrubí za pomoci otevření škrticí klapky
- Snižením zatížení motoru
- Zpoždění doby zapálení svíčky
- Přidáním různých aditiv do paliva

## 2.1 Předčasné vznícení / pre-ignition

Tento jev vzniká, když je směs zapálena dříve, než svíčka vytvoří jiskru. V anglické literatuře to najdeme jako pre-ignition. Samovolné vznícení směsi



je způsobeno například z horkých míst ve spalovací komoře například ze zapalovací svíčky. Tento jev vzniká především při nízkých otáčkách s vysokou zátěží motoru. Předčasné vznícení pak může vést k tvorbě tzv. super-klepání viz níže. [4]

## 2.2 Dělení klepání podle typu

Klepání můžeme rozdělit do několika kategorií.

### ■ Podle příčiny vzniku

#### Ohnivou vlnou ze svíčky / Spark knock

Toto klepání vznikne z důvodu šíření ohnivé vlny ze svíčky, která zapříčiní zvýšení tlaku v okolních ještě nevyhořelých místech spalovací komory. Při tomto typu klepání se plamen šíří typem deflagrace. Toto klepání má většinou mnohem menší následky, než při klepání způsobené předčasným vznícením. [4]

#### Předčasné vznícení / Pre-ignition

Viz výše. Exploze se může šířit jako deflagrace nebo detonace (většinou super-klepání). Horké místo, které zapříčí samovolné vznícení má určitý tlak a teplotu. Pokud se tento tlak a teplota nachází pod křivkou z obrázku 2.2, pak dochází k deflagraci. Pokud tlak a teplota jsou nad křivkou poté dochází k detonaci. Na obrázku 2.3 je poté vidět rozdíl šíření vln pro deflagraci a detonaci v jednom cyklu. [5]

### ■ Typu exploze

#### Deflagrace

Plamen se šíří tepelnou vodivostí (teplý materiál ohřívá studenější a zapaluje ho), šíří se podzvukovou rychlostí.

#### Detonace

Plamen se šíří pomocí rázové vlny (detonační vlny). Tak stlačuje okolní materiál a tím mění jeho teplotu na bod vznícení. Uvolňovaná energie je pod vysokým tlakem a tím pomáhá k šíření rázové vlny.

### ■ Podle velikosti změny tlaku [5]

#### Neklepání / non-knocking

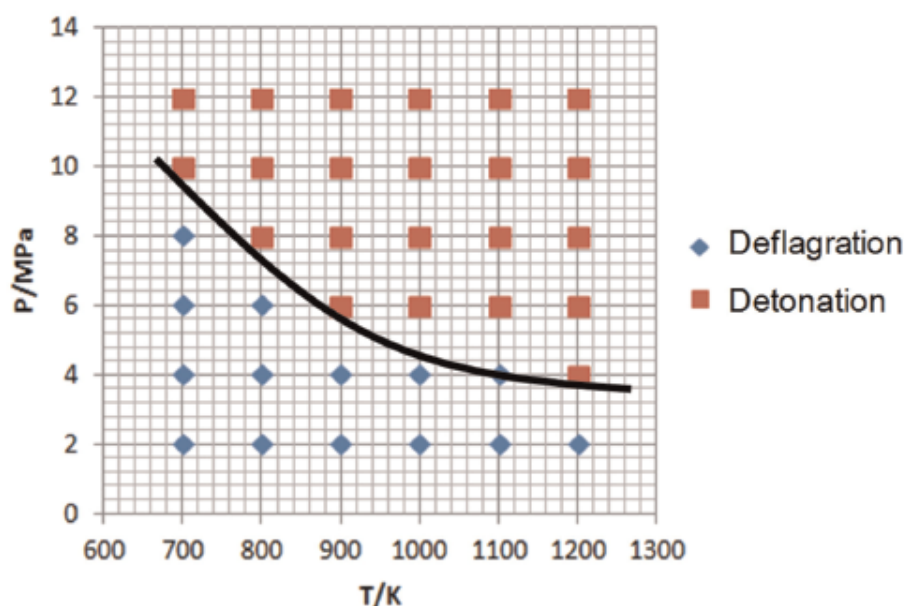
Pro  $|\Delta p| \approx 0$  MPa

#### Mírné-klepání / slight-knock

Pro  $|\Delta p| > 0$  MPa  $\cap$   $|\Delta p| < 0.2$  MPa

#### Silné-klepání / heavy-knock

Pro  $|\Delta p| > 0.2$  MPa  $\cap$   $|\Delta p| < 2$  MPa



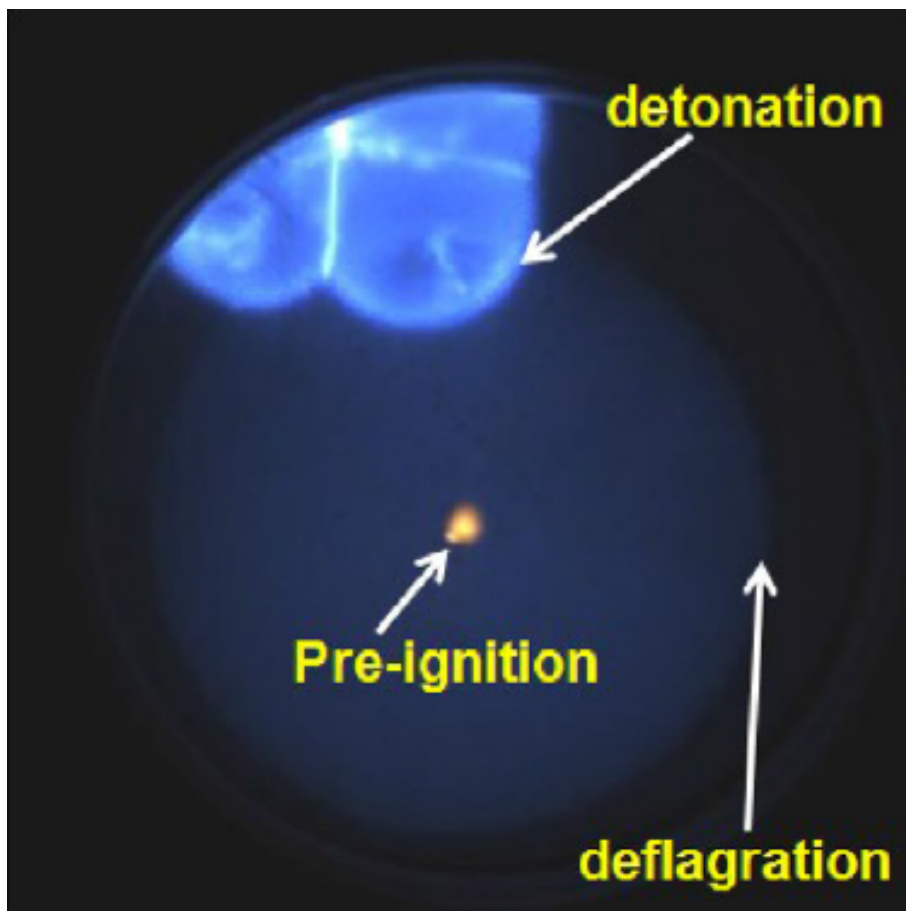
Obrázek 2.2: Křivka pro rozhraní deflagrace a detonace [5]

### Super-klepání / super-knock

Pro  $|\Delta p| > 2 \text{ MPa}$

Velikost  $\Delta p$  při super-klepání může v praxi dosahovat i hodnot nad 10MPa. V literatuře se s tímto pojmem také můžeme setkat jako unwanted pre-ignition, mega knock, LSPI (low-speed pre-ignition), deto-knock, SPI (stochastic pre-ignition), developing detonation či subsequent front propagation. Super-klepání má mnohem větší následky na poškození motoru. Motor může okamžitě poškodit už v jednom cyklu díky extrémně vysokým špičkám tlaku a následné oscilaci. Použitím standardních metod pro potlačení klepání jako jsou například zpoždění zapálení svíčky, chlazení vzduchu jdoucí do válce a zlepšení přenosu tepla nemusí vždy znamenat zamezení možnému vzniku super-klepání. Super-klepání je jedna z hlavních překážek ke zlepšení motorů s turbodmychadlem [4].

Porovnání všech typů klepání můžeme nalézt na Obrázku 2.4. Nastane-li předčasné vznícení (kvůli lokálně vyššímu tlaku a teplotě místa) může to vést ke všem typům klepání a to super-klepání, silné-klepání, mírné-klepání a neklepání. Velikou závislost na síle klepání určuje místo a čas, ve kterém samovznícení vzniklo. Pokud vznikne poblíž zapalovací svíčky, tak následné spojení dvou ohnivých vln nemusí mít za následek pozorovatelné oscilace tlaku. Super-klepání je vždy způsobeno kvůli předčasnému vznícení směsi [5]. Pokud tedy eliminujeme předčasné vznícení, eliminujeme také vznik super-klepání.

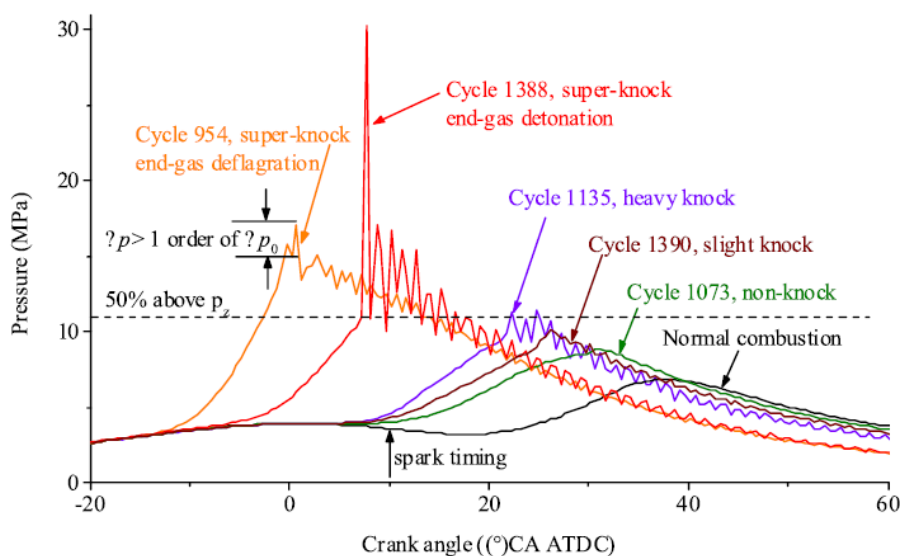


**Obrázek 2.3:** Průběh zapálení směsi, který obsahuje předčasné vznícení, deflagraci a detonaci [4]

## 2.3 Snímače klepání motoru

### 2.3.1 Přímé měření

Detekování klepání může zjištěno z několika různých signálů motoru. Nejzásadnější senzor je snímač tlaku ve válci, neboť změny tlaku jsou přímo dané vznikem klepání. Tento typ měření poskytuje nejlepší signál k detekci klepání. Nicméně, každý válec potřebuje svůj vlastní senzor a cena takového snímače je stále relativně vysoká. Tyto snímače se tedy nejvíce používají v laboratorních podmínkách.



Obrázek 2.4: Přehled typů klepání [4]

### 2.3.2 Nepřímé měření

Dalším typem senzoru jsou poté piezoelektrické akcelerometry, těmi se budeme zabývat v této práci. Fungují na principu působící síle na krystal uvnitř senzoru, které generují napětí úměrný působící síle. Akcelerometry jsou připevněny na blok motoru a snímají tak intenzitu vibrací. Tyto snímače trpí na špatný poměr signál/šum (SNR – signal noise ratio) zvláště při vyšších otáčkách motoru [6]. Existují dva typy těchto akcelerometrů.

- Prvním typem jsou rezonanční senzory, které se používají u mnohých low-endových systému pro detekci. Jistou nevýhodou je skutečnost, že zesiluje sílu signálu v okolí rezonanční frekvence.
- Druhým typem jsou snímače širokopásmové, které nemají rezonanční píky pod 20kHz a tím neovlivňují měření v závislosti na frekvenci. Můžeme tedy tyto typy snímačů dávat univerzálně na každý motor, nezávisle na jeho klepací frekvenci [7].

Dalším typem senzoru může být využití zapalovací svíčky, využitím měření ionizačního proudu. Chemická reakce po zapálení svíčkou vytvoří volné náboje, které můžeme měřit přiložením napětí na zapalovací svíčku těsně po zapálení směsi. Podle velikosti protékajícího proudu můžeme měřit kvalitu spálení směsi a tím případně detekovat klepání [8].

## Kapitola 3

### Komplexní transformace

Jelikož v návrhu poradního systému budeme potřebovat frekvenční charakteristiky a znalost těchto transformací je zásadní pro tuto diplomovou práci, sepíšu zde definice jednotlivých komplexních transformací. Později budu diskutovat, proč jsem jakou transformaci použil.

#### 3.1 Spojitá Fourierova transformace

Spojité Fourierova transformace funkce času je komplexní funkce frekvence, kde absolutní hodnota (amplitudové spektrum) představuje amplitudu sinusového signálu dané frekvence a argument (fázové spektrum) jeho fázový posun sinusového signálu na dané frekvenci. Pro reálnou časovou funkci je amplitudové spektrum sudou funkcí a fázové spektrum lichou funkcí.

##### **Přímá spojitá Fourierova transformace**

Nechť  $f(t)$  je komplexní funkce definována na  $\mathbb{R}$ .

Funkce

$$F(\omega) = \int_{-\infty}^{\infty} f(t)e^{-j\omega t} dt, \quad \omega \in \mathbb{R}$$

se nazývá **Fourierova transformace funkce  $f$**

**Zpětná/Inverzní spojitá Fourierova transformace** Nechť  $F(\omega)$  je komplexní funkce definována na  $\mathbb{R}$ .

Funkce

$$f(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} F(\omega) e^{j\omega t} d\omega, \quad t \in \mathbb{R}$$

se nazývá **zpětná/inverzní Fourierova transformace funkce f**

Postačující podmínka pro existenci Fourierovy transformace:

$$\int_{-\infty}^{\infty} |f(t)| dt < \infty$$

pro spojitou transformaci

$$\int_{-\infty}^{\infty} |F(\omega)| d\omega < \infty$$

pro inverzní transformaci

## 3.2 Diskrétní Fourierova transformace (DFT)

Podobně jako pro spojitý čas můžeme definovat Fourierovu transformaci pro diskrétní čas. Tato transformace vypočítaná z definice, měla velmi vysoké výpočetní nároky. Popsáním efektivního algoritmu FFT (Fast Fourier Transform) pány J. W. Cooley a J. W. Tukey se výpočet stal výrazně rychlejší. Rozdíl mezi FFT a DFT je tedy především v rychlosti výpočtu (pro DFT je složitost  $O(n^2)$  a pro FFT je složitost  $N(n \log n)$ ).

### Přímá diskrétní Fourierova transformace

Diskrétní Fourierova transformace transformuje posloupnost  $N$  komplexních čísel  $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N$  na posloupnost komplexních čísel  $\mathbf{X}_1, \mathbf{X}_2, \dots, \mathbf{X}_N$  podle

$$X_k = \sum_{n=1}^N x_n e^{-j2\pi kn/N}$$

### Zpětná/Inverzní diskrétní Fourierova transformace

Zpětná/Inverzní diskrétní Fourierova transformace transformuje posloupnost  $N$  komplexních čísel  $\mathbf{X}_1, \mathbf{X}_2, \dots, \mathbf{X}_N$  na posloupnost čísel  $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N$  podle

$$x_k = \frac{1}{N} \sum_{n=1}^N X_n e^{j2\pi kn/N}$$

### 3.3 Krátkodobá Fourierova transformace (STFT)

Tento typ transformace se používá, pokud chceme rozlišit frekvence v jednotlivých časových úsecích (okénkách) signálu. Jde tedy o opakované používání Fourierovy transformace na rozdělený signál po okénkách. Můžeme také okénka přes sebe překrývat a také je můžeme váhovat. Právě okénkovou reálnou funkcí  $w(t)$  definujeme délku daného okénka a váhu signálu. Typická funkce  $w(t)$  může vypadat například  $w(t) = 0, t \in (-\infty, -1) \cup (1, \infty)$ ;  $w(t) = 1, t \in \langle -1, 1 \rangle$ . Jde o pravoúhlý tvar, kde velikost okna jsou 2 a celý úsek signálu okně má stejnou váhu.

Výsledkem výpočtu STFT je komplexní spektrum poskytující informaci o amplitudě a fázi signálu pro daný čas a frekvenci

#### Krátkodobá spojitá Fourierova transformace

$$STFT\{x(t)\}(\tau, \omega) \equiv F(\tau, \omega) = \int_{-\infty}^{\infty} x(t)w(t - \tau)e^{-j\omega t} dt$$

kde  $x(t)$  je časový signál, který má být transformován,  $w(t)$  je reálná okénková funkce a  $\tau$  je časový posun okénka

#### Krátkodobá diskrétní Fourierova transformace

$$STFT\{x[n]\}(m, \omega) \equiv F(m, \omega) = \sum_{n=-\infty}^{\infty} x[n]w[n - m]e^{-j\omega n} dt$$

kde  $x[n]$  je vzorkovaný signál, který má být transformován,  $w(t)$  je reálná okénková funkce a  $m$  je posun okénka.

V praxi často používáme spektrogram, který nám říká výkon jednotlivých frekvencí v jednotlivém časovém úseku. Spektrogram vypočítáme jako

$$\text{spectrogram}\{x(t)\}(\tau, \omega) = |X(\tau, \omega)|^2$$

## 3.4 Vlnková transformace (WT)

V předchozí transformaci je problém v tom, že velikost okénka limituje vždy jeden nebo druhý parametr a to buď rozlišení v čase, nebo rozlišení ve frekvenci. Pro delší okna máme lepší rozlišení ve frekvenci, ale menší rozlišení v čase. Pro menší okna je to přesně opačně. Problém řeší Vlnková transformace (Wavelet transform).

### Spojité Vlnková transformace

$$[W_\psi f](a, b) = \int_{-\infty}^{\infty} f(t)\psi_{a,b}^*(t)dt$$

Skalární součin v integrálu určuje do jaké míry je tvar signálu korelovaný s lokální sondou danou příslušnou vlnkou  $\psi_{a,b}$ . Mateřskou vlnku  $\psi$  volíme tak, aby byla co nejvíce podobná našemu hledanému signálu.



## Kapitola 4

### Detekce klepání

Vznik klepání detekuje ECU (Engine Control Unit). Ta se snaží optimalizovat výkon motoru na základě dat ze senzorů. A upravuje poměr směsi, časové zapálení svíčky a volnoběh. Dříve tyto činnosti zařizovali mechanické a pneumatické komponenty. Algoritmus detekce klepání by měl nastavovat časové zapálení svíčky na hranici, kdy motor začíná klepat. Uvažujeme tedy fakt, že při zpoždění zapálení svíčky snižujeme riziko výskytu klepání a naopak předstihem riziko zvyšujeme. Samotná detekce je velice závislá na rychlosti otáčení motoru, kvůli většímu okolnímu ruchu, který zašumí data ze senzorů. Z tohoto důvodu spousty detekujících systémů vypínají detekci při otáčkách vyšších než 4000 RPM a nastavují velmi konzervativní časové zapálení svíčky, které garantuje, že ke klepání nebude docházet [7]. Dalším problémem, který může vznikat při detekci klepání je ten, že s postupem času komponenty motoru stárnou. To vede například k většímu zašumění signálu a tím horší detekci. Vyskytují se poté častěji detekce klepání, když klepání nenastane a nedetekování klepání když klepání nastane.

Dnešní systémy detekce jsou nastaveny tak, že vyhodnocují signál jen v určitém čase, ve kterém by mohlo ke klepání dojít. Okno je přibližně nastaveno od 10 stupňů do 70 stupňů před horní úvrati daného válce. To vede k větší robustnosti systému. Detekuje-li algoritmus klepání jsou dva přístupy, jak toto klepání řešit. Prvním přístupem je to řešit globálně a to tím, že provedeme zpoždění zapálení svíčky pro každý válec stejně, nezávisle u jakého válce ke klepání došlo. To samozřejmě šetří paměť a operační výkon CPU v ECU. Druhým přístupem je zasáhnout individuálně, pouze na daný válec. Toto řízení pak poskytuje motoru lepší výkon i spotřebu paliva. S nárůstem výpočetního výkonu ECU se tento přístup stává používanější.

## 4.1 Odhad klepací frekvence

Různé zdroje odkazují na možnost odhadnout klepací frekvenci pomocí šířky pístu. V článku „Detection of knocking in Spark Ignition (SI) engines using CMAC neural networks“ [9] je napsán vzorec pro výpočet frekvence tlakové vlny dané Draperovou rovnicí:

$$f = \frac{\rho_{1,0} c_s}{\pi B} \quad [Hz]$$

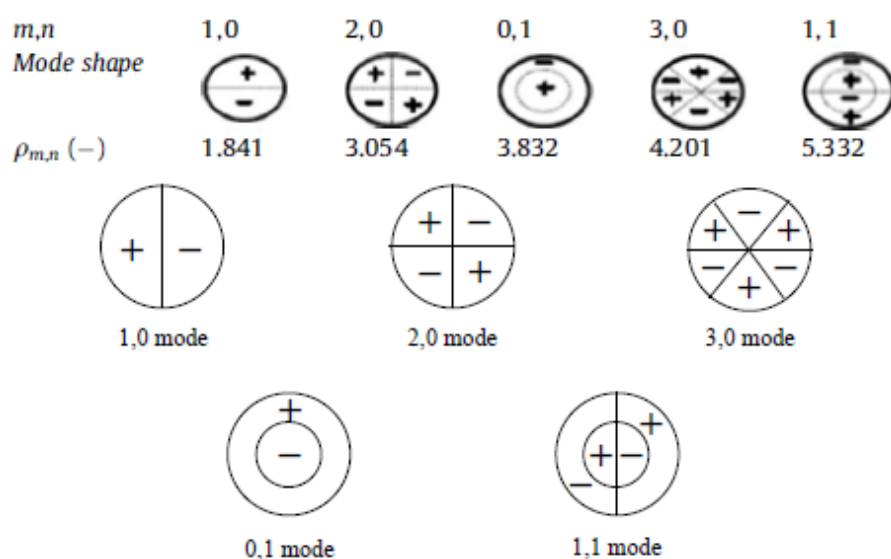
kde  $\rho_{1,0}$  je vlnový koeficient pro první radiální režim vibrací. V článku vypočítáno definováno jako  $\rho_{1,0} = 1.841$ .  $c_s$  je rychlost zvuku ve spalovací komoře, která brána jako 900 m/s.  $B$  je vnitřní průměr válce v metrech. Po dosazení koeficientů vychází výpočet frekvence jako

$$f = \frac{527,40}{B} \quad [Hz]$$

Článek [9] více nepopisuje odvození vzorce nebo jeho jednotlivých koeficientů. Další informace však můžeme najít v článku [10], [11] a [12], kde je více objasněno co koeficient  $\rho_{1,0} = 1.841$  znamená. Koeficient odpovídá akustickému režimu, kde tlakové fluktuace jsou v tangenciálním a radiálním směru. Koukneme-li na Obrázek 4.2, je dominance módu 1,0 nejvyšší a tu se budeme snažit v naší práci odhadnout.

Nejdominantnější mód 1,0 je nazýván jako fundamentální frekvence a mezi fundamentální frekvencí a 20kHz se typicky vyskytují další 2 až 4 frekvenční špičky (vibrační módy). Každá struktura motoru může mít různě vysoké a různě silné vibrační módy. Šum, v závislosti na poloze snímače může mít za následek, že některé módy nemusí být detekovatelné. Fundamentální frekvence se může měnit až o  $\pm 400Hz$  v závislosti na teplotě spalovací komory. [7]

Dalším zdrojem jsem našel internetovou kalkulačku pro odhad klepací frekvence [13]. Kalkulačka požaduje vstupní hodnotu průměru pístu a po stisknutí tlačítka „Calculate“ odhadne klepací frekvenci. Z výsledků lze lehce odhadnout vzorec výpočtu klepací frekvence jako



Obrázek 4.1: Režimy vibrace válce [11]

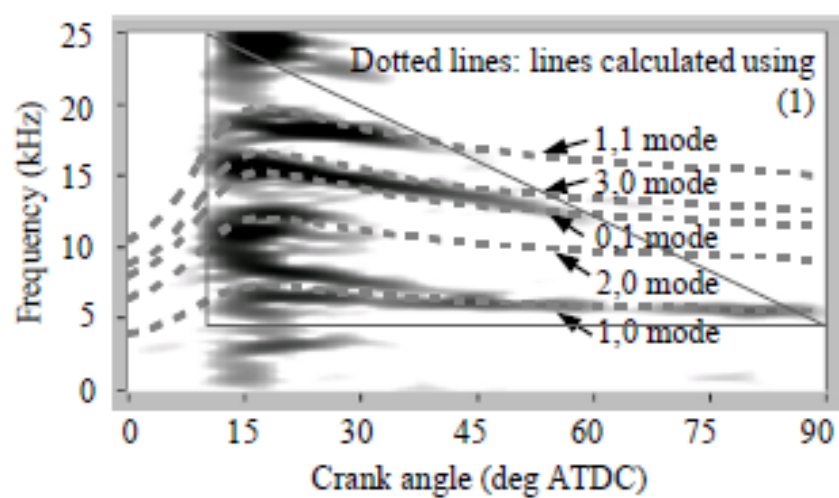
$$f = \frac{573,25}{B} \quad [Hz]$$

Pokud budeme vycházet z předchozího vzorce a a vezmeme rychlost šíření zvuku jako 900 m/s. Vyjde nám vlnový koeficient  $\rho_{1,0} = 2$ .

## 4.2 Možné algoritmy detekce klepání a zjištění klepací frekvence

Všechny algoritmy pro detekci klepání na motoru už počítají s apriorní informací o hodnotě klepací frekvence. Pokud jí ale neznáme, můžeme jí předběžně odhadnout (viz 4.1). Pokud je třeba zpracovat reálná měření, jsou používány frekvenční charakteristiky. Nejjednodušším přístupem je motor dostat do stavu klepání, naměřit data a z nich vypočítat Fourierovu transformaci. Maximum by nám mělo udávat klepací frekvenci. Po teoretickém úvodu však víme, že jev klepání nám motor může zničit, v případě, že algoritmus pro detekci klepání je nastaven špatně. Proto je v zájmu nezničení motoru, tento jev kontrolovat a vystavovat ho pouze na nezbytně nutnou dobu pro jeho detekci a zjištění frekvence.

Mnou navržené algoritmy dělají několik statistik na základě překročení mezí a poté statistiku vyhodnocují, na jakých frekvencích bylo překročení



Obrázek 4.2: Dominance jednotlivých módů [10]

nejčastější. Meze jsou stanoveny pomocí naměřených dat v době, kdy ke klepání nedochází a poté porovnávány s daty, ve kterých by klepání nastává. Více o implementaci daných algoritmů v kapitole 6.3.



## Část II

### Praktická část

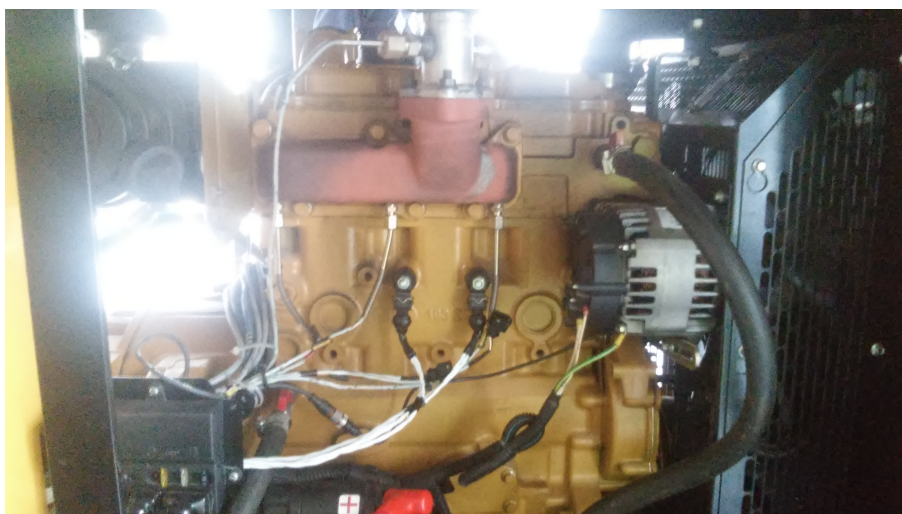
## Kapitola 5

### Laboratorní měření klepání motoru

Dne 10. listopadu 2017 jsme provedli laboratorní měření na dieslovém motoru (viz obrázky 5.1 a 5.2). Jedná se o tří válcový motor s vrtáním 105 mm (vnitřní průměr válce) a se zdvihem (vzdálenost mezi horní a dolní úvratí) 127 mm. Zdvihový objem válce je 3.3 litry [14]. Tento motor je zároveň dieslový generátor, který je upravený pro použití i plynu. Plyn snižuje náklady na běh motoru, avšak právě u něj může docházet ke klepání, které budeme chtít detekovat. K motoru je přidělena kontrolní jednotka, která řídí spalování plynu v závislosti na rychlosti motoru a zátěži. Tato jednotka se také stará o klepání motoru, které se při změně poměru směsi plyn/nafta může vyskytnout. Tato modifikace motoru je provedena především z ekonomického hlediska, kdy plyn je levnější palivo než nafta. Motor poté generuje méně emisí a celá modifikace nezpůsobuje ztrátu výkonu. Na motoru jsou nainstalovány dva piezoelektrické snímače A a B (vlevo snímač B, vpravo snímač A na Obrázku 5.1). Pořadí pálení válců je takové, že první válec je krajní vpravo. Druhý válec v pořadí pálení je uprostřed a třetí a poslední válec je na kraji vlevo. Snímač A je tedy mezi válci 2 a 3 a snímač B je mezi válci 1 a 2.

#### 5.1 Cíl měření

Cílem měření je naměřit data pro různé poměry směsi nafta/plyn. Z dat popsat jejich strukturu a případné rozdělení, aby mohli být aplikovány pravděpodobnostní odhady. Data budeme měřit vždy v ustáleném stavu motoru, při konstantních otáčkách a při zachování stejného poměru směsi.



**Obrázek 5.1:** Diesellový motor, pohled na snímače poblíž válců

## 5.2 Průběh měření dat

Před samotným měřením byl motor rozehrát, aby nevznikaly chyby v důsledku studeného motoru. Na motor je nastavena zátěž 18kW. Samotné odečítání dat je prováděno pomocí programu, který komunikuje s řídicí jednotkou u motoru. Pomocí něj lze ovládat například poměr směsi nafta/plyn, které během měření budeme měnit. Program komunikuje se všemi snímači na motoru a tedy i s piezoelektrickými snímači. Z nich se později pokusíme vyčíst, zda ke klepání došlo či nikoliv a pokud ano na jaké frekvenci klepání vzniklo. Program, který odčítá data je vytvořený firmou poskytující měření na motoru. Data je možné odčítat pouze z jednoho snímače zároveň, tím je program limitován. Motor je nejprve nastaven na 100% poměr nafty a 0% plynu a jsou odečtena data o délce 60 sekund pro snímač A. Dále následuje 60 sekundová data pro snímač B. Tento proces se opakuje třikrát a vzniknou výsledná data pro snímač A 3x60 sekund a snímač B 3x60 sekund. Poté je změněn poměr směsi na 40% plynu a 60% nafty. Z tohoto měření opět získáme data o velikosti 3x60 sekund pro snímač A a B. Při poměru 50% plynu a 50% nafty nastal neznámý problém a řídicí jednotka zasáhla vypnutím přívodu plynu do motoru. Problém nastal při měření druhé minuty na snímači B. Měření v tu dobu bylo přerušeno. Po krátké pauze byl opět nastaven poměr plynu/nafta 50%/50% a měření pokračovalo. Následné měření pro poměry 60% plynu a 65% plynu se obešli bez sebemenších problémů.

Naměřená data:

- 18kW 100% nafta 0% plyn





**Obrázek 5.2:** Diesellový motor

D-2\_2017-11-10\_10-07-14\_#A.au

D-2\_2017-11-10\_10-08-15\_#B.au

D-2\_2017-11-10\_10-09-58\_#A.au

D-2\_2017-11-10\_10-10-58\_#B.au

D-2\_2017-11-10\_10-12-20\_#A.au

D-2\_2017-11-10\_10-13-20\_#B.au

■ 18kW 100% nafta 40% plyn

D-2\_2017-11-10\_10-16-11\_#A.au

D-2\_2017-11-10\_10-17-11\_#B.au

D-2\_2017-11-10\_10-18-19\_#A.au

D-2\_2017-11-10\_10-19-19\_#B.au

D-2\_2017-11-10\_10-20-27\_#A.au

D-2\_2017-11-10\_10-21-27\_#B.au

■ 18kW 100% nafta 50% plyn

D-2\_2017-11-10\_10-40-16\_#A.au

D-2\_2017-11-10\_10-41-16\_#B.au

D-2\_2017-11-10\_10-42-37\_#A.au

D-2\_2017-11-10\_10-43-37\_#B incomplete (gas off).au

D-2\_2017-11-10\_10-45-21\_#A.au

D-2\_2017-11-10\_10-46-21\_#B.au

- 18kW 100% nafta 60% plyn

D-2\_2017-11-10\_10-53-33\_#A.au

D-2\_2017-11-10\_10-54-33\_#B.au

D-2\_2017-11-10\_10-55-39\_#A.au

D-2\_2017-11-10\_10-56-40\_#B.au

D-2\_2017-11-10\_10-57-45\_#A.au

D-2\_2017-11-10\_10-58-45\_#B.au

- 18kW 100% nafta 65% plyn

D-2\_2017-11-10\_11-02-01\_#A.au

D-2\_2017-11-10\_11-03-01\_#B.au

D-2\_2017-11-10\_11-04-07\_#A.au

D-2\_2017-11-10\_11-05-07\_#B.au

D-2\_2017-11-10\_11-06-10\_#A.au

D-2\_2017-11-10\_11-07-10\_#B.au

## 5.3 Digitalizace naměřených dat

Data jsou programem uložena ve formátu \*.au. Tento formát podporuje mnoho typů kódování, z nichž nepoužívanější jsou především 16-bit a 32-bit lineární pulzně kódová modulace. Na tyto data bylo použito kódování 16-bit pulzně kódové modulace. Popisem zvukového formátu se budeme zabývat v následující kapitole.

## Kapitola 6

### Návrh poradního systému

#### 6.1 Načtení a předzpracování dat

Z měření máme k dispozici naměřená data ve formátu \*.au. Tento formát si popíšeme, abychom mohli s daty pracovat v našem programu. Následně náš poradní systém je rozšířen i na podporu zvukových souborů \*.wav. Tyto zvukové soubory totiž patří k nejrozšířenějším bezztrátovým zvukovým souborům.

##### 6.1.1 Au formát [15]

Formát byl představen společností Sun Microsystem. Celkem podporuje 22 typů kódování včetně 8-bit, 16-bit, 24-bit a 32-bit pulzně kódové modulace. Tento typ kódování bude právě pro nás důležitý, protože data, která budeme zpracovávat, jsou právě kódována v 16-bit pulzně kódové modulaci. Všechna datová pole jsou typu big-endian, tedy nejvýznamnější byte (MSB) je na nejnižší adrese. Význam jednotlivých bytu shrnuje tabulka 6.1

32-bitové slovo	Název buňky	Popis
1	magické číslo	Hodnota 0x2E736E64h, ve všech souborech stejná. V ASCII je to hodnota „.snd“
2	datový offset	Toto číslo musí být dělitelné 8. Jedná se o velikost hlavičky v bytech, pokud nechceme přidat anotaci do souboru (extra informace) je tato velikost minimální a to 0x18h
3	velikost souboru	Velikost souboru v bytech. Pokud velikost není známa je použita hodnota 0xffffffff. Spousty programů, které čtou soubor toto pole ignorují.
4	kódování	Typ kódování, 1 = 8-bit G.711 u-law, 2 = 8-bit lineární pulzně kódová modulace, 3 = 16-bit lineární pulzně kódová modulace, 4 = 24-bit lineární pulzně kódová modulace, 5 = 32-bit lineární pulzně kódová modulace, 6 = 32-bit IEEE floating point, 7 = 64-bit IEEE floating point, 8 = Fragmented sample data, 9 = DSP program, 10 = 8-bit fixed point, 11 = 16-bit fixed point, 12 = 24-bit fixed point, 13 = 32-bit fixed point, 18 = 16-bit linear with emphasis, 19 = 16-bit linear compressed, 20 = 16-bit linear with emphasis and compression, 21 = Music kit DSP commands, 23 = 4-bit compressed using the ITU-T G.721 ADPCM voice data encoding scheme 24 = ITU-T G.722 SB-ADPCM, 25 = ITU-T G.723 3-bit ADPCM, 26 = ITU-T G.723 5-bit ADPCM, 27 = 8-bit G.711 A-law
5	vzorkovací frekvence	Počet vzorků za sekundu
6	počet kanálů	1 pro mono zvuk, 2 pro stereo
7 - ...	data	Data souboru kódované podle zadaného typu kódování

**Tabulka 6.1:** Au formát, přehled významů jednotlivých bytů

Náš data z piezoelektrického snímače obsahují 2 kanály. V prvním kanálu jsou obsažena naměřená data ze snímače. V druhém kanálu je synchronizační signál, kde jeho náběžná hrana definuje první válec v pořadí pálení v horní úvratí. Poradní systém poté může pracovat s daty jednotlivých cyklů

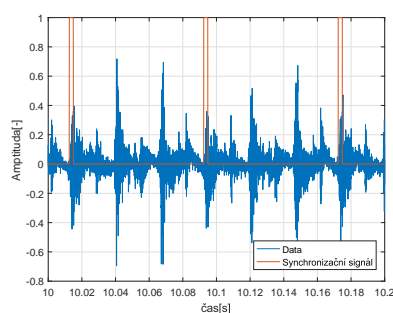
motoru. Data piezoelektrického snímače z našeho měření ve formátu \*.au jsou kódována pomocí 16-bitové pulzně kódové modulace.

## 6.1.2 Wave formát 16bit PCM [16]

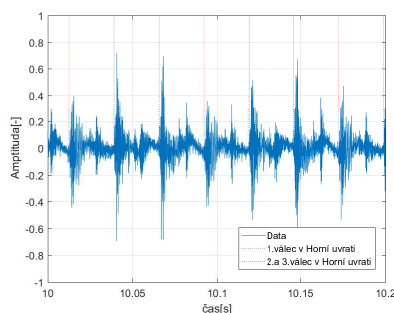
Náš poradní systém bude podporovat i formát typu wav modulovaný v 16-bit PCM, proto si také velmi stručně popíšeme jeho strukturu. Tento zvukový formát vytvořili firmy IBM a Microsoft. Tento formát je podmnožina specifikace RIFF pro ukládání multimediálních souborů. Strukturu dat vysvětluje tabulka 6.2:

Endian	Offset v bytech	Název	Popis
big	0	ChunkID	Obsahuje v ASCII hodnotu „RIFF“ ,(0x52494646h)
little	4	ChunkSize	Obsahuje hodnotu 20+SubChunk1Size+SubChunk2Size
big	8	Format	Obsahuje v ASCII hodnotu „WAVE“ (0x57415645h)
big	12	Subchunk1 ID	Obsahuje v ASCII hodnotu „fmt „ (0x666d7420h)
little	16	Subchunk1Size	Hodnota 16 pro pulzně šířkovou modulaci
little	20	AudioFormat	Hodnota 1 pro pulzně kódovou modulaci
little	22	NumChannels	Počet kanálů ve zvukovém souboru. 1 pro mono zvuk, 2 pro stereo
little	24	SampleRate	Vzorkovací frekvence
little	28	ByteRate	= SampleRate * NumChannels * BitsPerSample/8
little	32	BlockAlign	= NumChannels * BitsPerSample/8
little	34	BitsPerSample	Velikost dat na jeden vzorek v bitech
big	36	Subchunk2ID	Obsahuje v ASCII hodnotu „data“ (0x64617461)
little	40	Subchunk2Size	= NumSamples * NumChannels * BitsPerSample/8 (Velikost dat v bytech)
little	44	Data	Data odpovídající hlavičce souboru

**Tabulka 6.2:** Wave formát, přehled významů jednotlivých bytů



(a) : Vstupní data ze zvukového souboru, neupravená



(b) : Rozdělení signálu na horní úvratě jednotlivých píští

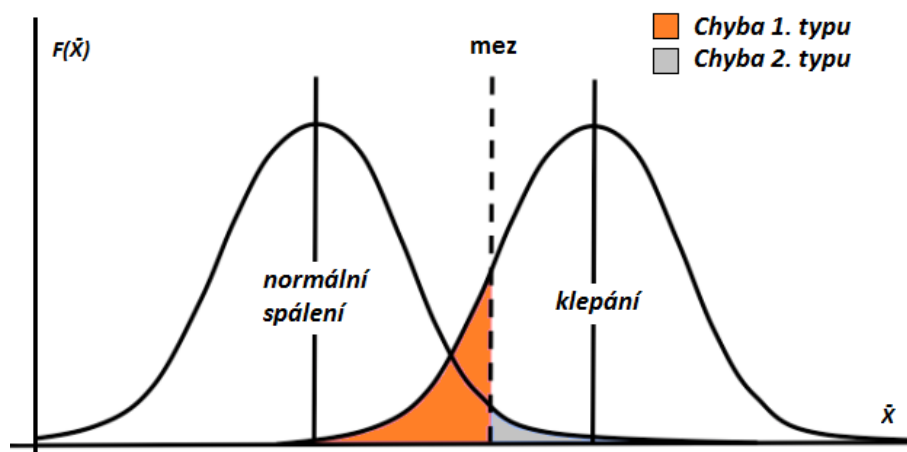
### 6.1.3 16-bitová pulzně kódová modulace

Jde o modulační metodu, která převádí analogový signál na digitální. Klíčovým parametrem je vzorkovací frekvence a rozlišení hodnot, které je dáno bitovou velikostí samotného vzorku. Signál je modulován na hodnoty v rozmezí (-1 a 1). Hodnota 0 ( 0x0000h ) je střed rozsahu a odpovídá hodnotě 0. Záporné hodnoty jsou reprezentovány dvojkovým doplňkem, tedy jedná-li se o záporné číslo, poznáme to z prvního binárního čísla. Pokud binární číslo začíná 1, pak se jedná o záporné číslo. Záporných čísel pro 16-bitové číslo je 32768 a kladných 32767. Kvantovací krok (rozlišení) je dáno jako  $\frac{1}{32768}$ , což odpovídá přibližně  $3.0518 \cdot 10^{-5}$ . Minimální hodnota je -1 zapíšeme jako 0x8000h. Maximální hodnotu  $1 - \frac{1}{32768}$ , kterou zapíšeme jako 0x7FFFh. Tento zvukový formát je bezztrátový, avšak vzniká zde ztráta při převodu na dané kvantum. Velikost kvantizační chyby, je vzdálenost mezi kvantovanými a původními navzorkovanými body.

## 6.2 Zpracování dat

### 6.2.1 Filtrace dat a nastavení mezí

Z teorie víme, že klepací (fundamentální) frekvence by se měla nacházet v pásmu 1kHz – 20kHz. Prvním krokem se tedy můžeme vymežit na toto pásmo. Obecně jsou data ze snímačů velice zašuměná, proto je dobré vymežit se pouze na válec, který je poblíž snímače. Samotná velikost a posunutí okna by měly být upravené tak, abychom pozorovali úsek, kde s největší pravděpodobností bude docházet ke klepání. Máme-li okno správně posunuté



Obrázek 6.2: Chyby 1. a 2. typu

a nastavené, vytyčíme si meze. Tyto meze nám budou rozlišovat stavy mezi standardním shořením paliva a vznikem klepání. Pro tyto meze mohou nastat celkem 4 stavy. První stav nastane, když průběh spalování je normální a nastavené meze nepřekročíme. Druhým možným stavem je, když ke klepání ve spalovací komoře dojde a nastavené meze překročíme. V obou těchto případech jde o správné detekování jevu, který nastal. Klíčové stavy pro správné nastavení mezí jsou zbylé dva. Třetím možným stavem je, pokud jsou meze překročeny, ale ke skutečnému klepání nedojde. To může být například způsobeno zašuměním snímačů, nebo případně rezonance nějaké součástky na frekvenci, kterou pozorujeme, která při vytyčování mezí nerezonovala. Možností je celá řada. Tomuto stavu říkáme chyba prvního typu nebo také falešně pozitivní. Posledním možným stavem je stav, kdy ke klepání dojde, avšak vytyčené meze nepřekročíme. Tomuto stavu říkáme chyba druhého typu nebo také falešně negativní. Rozdělení pro klepání můžeme předpokládat různé, neboť klepání může vznikat s různou intenzitou a různým způsobem vzniku. Porovnání chyb 1. a 2. typu jsou zobrazeny na 6.2.

## 6.2.2 Výběr vhodné transformace

Pro vytvoření detekce, kdy ke klepání dojde, je nutné provést frekvenční transformaci, která přiřadí každé frekvenci pro každý cyklus motoru hodnotu. Tuto hodnotu porovnáme s mezí a posoudíme, zda došlo ke klepání či nikoliv. Náhodnou veličinu  $\bar{X}$  (viz 6.2) budeme generovat pomocí výkonového spektra Fourierovy transformace vybraného okénka. Počet náhodných proměnných bude odpovídat počtu naměřených cyklů. Ve výsledku se tedy bude jednat o Krátkodobou Fourierovu transformaci. V našem případě je možné Fourierovu transformaci zobecnit na vlnkovou transformaci. Je nutné si však nadefinovat vlnku, která by se měla podobat průběhu signálu při vzniku klepání. To díky

nedostatku dat a především z obecnosti řešení nejsme schopni odhadnout, proto signál korelujeme se sinem a cosinem (Fourierova transformace).

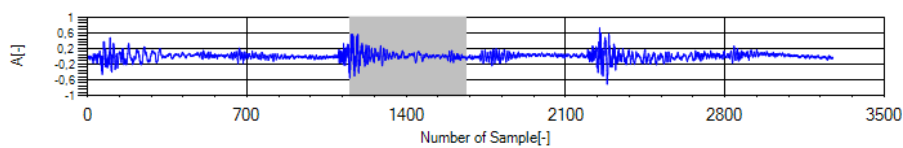
### 6.2.3 Rozdělení naměřených dat

V této části se pokusíme odhadnout rozdělení dat v oblasti klepací frekvence. Data budeme brát z laboratorního měření a posuneme okno nad spalovací část druhého válce. Velikost okna jsem zvolil 512 vzorků. Posun od horní úvratí prvního válce jsem zvolil 1150 vzorků (viz 6.3). Nyní provedeme FFT nad zvoleným úsekem daným velikostí a zpožděním pro každý naměřený cyklus motoru (mimo dat, během kterých nastaly nečekané komplikace). Z vypočítaných FFT získáme výkonové spektrum pomocí kvadrátu absolutní hodnoty. Data omezíme na rozmezích 4kHz až 6kHz. To nám dává datovou základnu o 21713 cyklů a 26 frekvencích (3923Hz, ..., 5926Hz). Celkem tedy asi 500 tisíc dat.

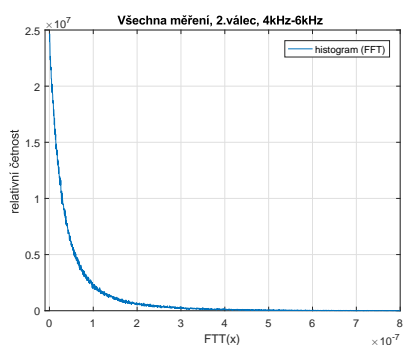
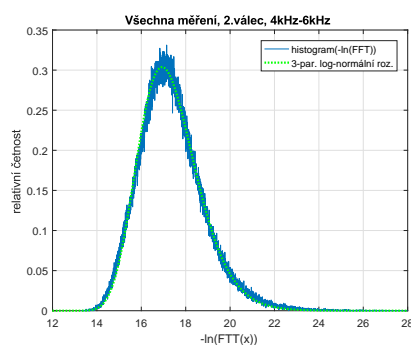
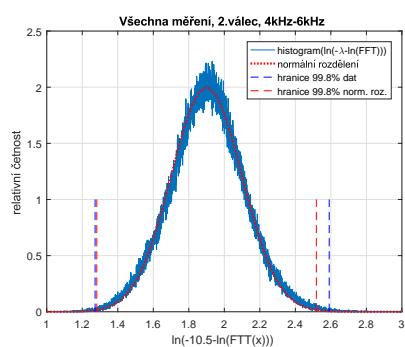
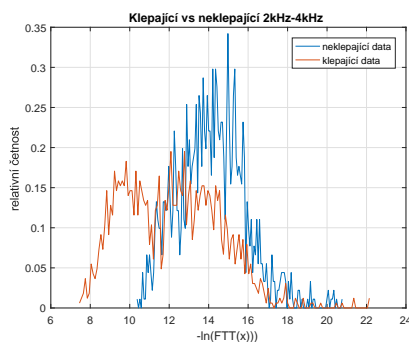
Omezení na frekvenci je nastaveno proto, abychom pozorovali oblast, kde pro tento typ motoru klepání vzniká. Na tento odhad jsem použil vzorec z kapitoly 4.1. Pro tato data vykreslíme histogram, který můžeme vidět na 6.4a. Nejvyšší četnost můžeme vidět kolem hodnoty 0. Pro zvýraznění hodnot kolem 0 a omezení vyšších hodnot provedeme logaritmus z vypočítaného výkonového spektra FFT a z něj až poté uděláme histogram. Zlogaritmováním hodnot menších než 1 a větších než 0 dostaneme záporné hodnoty. Rozdělení se podobá logaritmicko-normálnímu rozdělení, které je stranově otočené podle osy  $y$ . Znaménkem „-“ před logaritmem docílíme otočení. Výsledná data porovnáme s tří parametrickým logaritmicko normálním rozdělením, kde mimo střední hodnoty a rozptylu je také parametr  $\lambda$ , který posune  $X$  o  $\lambda$ . Porovnání můžeme vidět na Obrázku 6.4b. Na Obrázku 6.4c jsou pak data posunuta o  $\lambda$  a opět zlogaritmována přirozeným logaritmem. Tento výsledný histogram poté porovnáme s normálním rozdělením o stejné střední hodnotě a rozptylu jako pro logaritmicko-normální rozdělení. Odhadované parametry pro toto měření motoru pro frekvence 4kHz až 6kHz jsou  $\mu=1.9$ ,  $\sigma=0.2$  a  $\lambda=10.5$ . Pokud bychom chtěli provést matematické ověření normálního rozdělení, provedli bychom chí-kvadrát test. Ten dělat nebudeme, jelikož důležitým kritériem je pro nás spíše stanovení mezí (konfidenčních intervalů). Na detekci klepání budeme porovnávat naměřená data se stanovenou horní hranicí výkonového spektra FFT. Z Obrázku 15 je pro nás tedy důležitá spodní hranice, neboť je celý obrázek překlopený podle osy  $y$  a nejmenší hodnota nám říká nejvyšší výkon ve spektru.

Na Obrázku 6.4d vidíme porovnání dat, ve kterých ke klepání docházelo a nedocházelo. Pro neklepající data můžeme hrubě odhadnout, že se pravděpo-





Obrázek 6.3: Vybraná data z cyklu, okno 512, zpoždění 1150

(a) : Histogram  $|FFT(x)|^2$  pro všechna data v oblasti klepání(b) : Histogram  $-\ln|FFT(x)|^2$  + odhad log-norm rozdělení(c) : Histogram  $\ln(\lambda - \ln(|FFT(x)|^2))$  + porovnání s normálním rozdělením(d) : Histogram  $-\ln(|FFT(x)|^2)$  pro klepající a neklepající signál. Použit signál z dat D-20\_11-04-2012\_17-59-07\_#05.1\_knocking a D-20\_11-04-2012\_15-48-50\_#05

Obrázek 6.4: Zpracování a porovnání histogramů

dobně opět jedná o logaritmicko-normální rozdělení. O klepajících datech to už říci nemůžeme. Může se jednat o nějakou směs rozdělení. Nastavením mezi můžeme rozlišit, která data patří pro klepací signál a která pro neklepací signál s tím, že vždy budou nastávat chyby 1. a 2. typu. Naším cílem však není určit, jaká data k jakému průběhu spalování patří, ale spíše detekovat zda ke klepání dochází či nikoliv. Roli tedy bude hrát i vzdálenost (intenzita výkonového spektra) od stanovené meze. Stanovením zda dochází ke klepání či nikoliv můžeme dělat několika způsoby. Statistiky, které jsou implementovány do poradního systému, vám popíši v následující části.

## 6.3 Statistiky

Naměřená online data jsou porovnána se stanovenými mezemi nebo s předchozím měřením a uložena do statistiky. Tyto statistiky jsou poté zobrazeny v jednom grafu. Každá statistika obsahuje různé veliké hodnoty pro každou frekvenci. Počet hodnot pro jednu statistiku je dán velikostí pozorovaného okénka. Počet je dán samotnou FFT transformací. Pro porovnatelné zobrazení do jednoho grafu jsou všechny statistiky normovány tak, aby součet všech hodnot ve statistice byl roven hodnotě 1 (mimo situace kdy je celá statistika nulová).

### Over Max

Parametrem této statistiky je mez maximálního výkonu spektra FFT pro každou jeho frekvenci. Tato mez se stanovuje na základě výkonového spektra offline dat pro zadané okénko, z kterých se vezme kvantil 0.99. Jinými slovy pouze 1% dat je větší než tato stanovená mez. Každý celý naměřený cyklus online dat je zpracován do FFT výkonového spektra a pro každou frekvenci porovnán s příslušnou mezí. Pokud mez není překročena, nebo je měření shodné s mezí do statistiky, je přičtena hodnota 0. Pokud je mez překročena, do statistiky se přičítá kvadrát vzdálenosti o kolik je mez překročena. Důvodem volby kvadrátu je zvýraznění vzdálenějších naměřených hodnot.

$$S_t[f] = S_{t-1}[f] + (X_t[f] - X_{max}[f])^2, X_t[f] > X_{max}[f]$$

$$S_t[f] = S_{t-1}[f], X_t[f] \leq X_{max}[f]$$

kde  $S_t[f]$  je statistika v pro cykly do času  $t$  pro frekvenci  $f$ .

$S_{t-1}[f]$  je statistika pro cykly do času  $t - 1$  pro frekvenci  $f$ .

$X_t[f]$  je výkonové spektrum FFT v cyklu  $t$  pro frekvenci  $f$ .

$X_{max}[f]$  je stanovená maximální mez z offline dat pro frekvenci  $f$ .

### ■ Over Mean

Tato statistika je naprosto shodná s výše uvedenou statistikou až na parametr, kterým je pro tuto statistiku střední hodnota výkonu spektra FFT pro každou jeho frekvenci. Opět zde přičítáme kvadrát vzdálenosti od průměrné hodnoty.

$$S_t[f] = S_{t-1}[f] + (X_t[f] - X_{mean}[f])^2, X_t[f] > X_{mean}[f]$$

$$S_t[f] = S_{t-1}[f], X_t[f] \leq X_{mean}[f]$$

kde  $X_{mean}[f]$  je stanovená střední hodnota z offline dat pro frekvenci  $f$ .

### ■ Abs Diff

U této statistiky koukáme na celkovou absolutní hodnotu změny výkonového spektra ve dvou za sebou jdoucích cyklech. Předpokládáme, že pokud v prvním cyklu nedojde ke klepání a v následujícím cyklu ke klepání dojde, je klepání na té frekvenci, kde je změna výkonového spektra nejvyšší. Tato statistika je výhodná na úsecích, kde s velkým procentem dochází ke klepání. Pokud však budeme mít dlouhý úsek dat, kde ke klepání dojde pouze jednou, je pak tato změna silně zprůměrována díky ostatním změnám výkonového spektra. Tato statistika se zvyšuje každým dalším měřením, nehledě na nějaké mezní hodnoty.

$$S_t[f] = S_{t-1}[f] + |X_t[f] - X_{t-1}[f]|$$

kde  $X_{t-1}[f]$  je výkonové spektrum FFT v cyklu  $t - 1$  pro frekvenci  $f$ .

### ■ Max Abs Diff

Tato statistika potlačuje nevýhody statistiky předchozí. Do statistiky je zanesena pouze nejvyšší změna v absolutní hodnotě výkonového spektra. Pro dlouhé záznamy, kde ke klepnutí dojde pouze jednou, by měla mít silnější vypovídající schopnosti než u předchozí statistiky.

$$S_t[f] = |X_t[f] - X_{t-1}[f]|, |X_t[f] - X_{t-1}[f]| > \max(|X_{t-1}[f] - X_{t-2}[f]|, \dots, |X_2[f] - X_1[f]|)$$



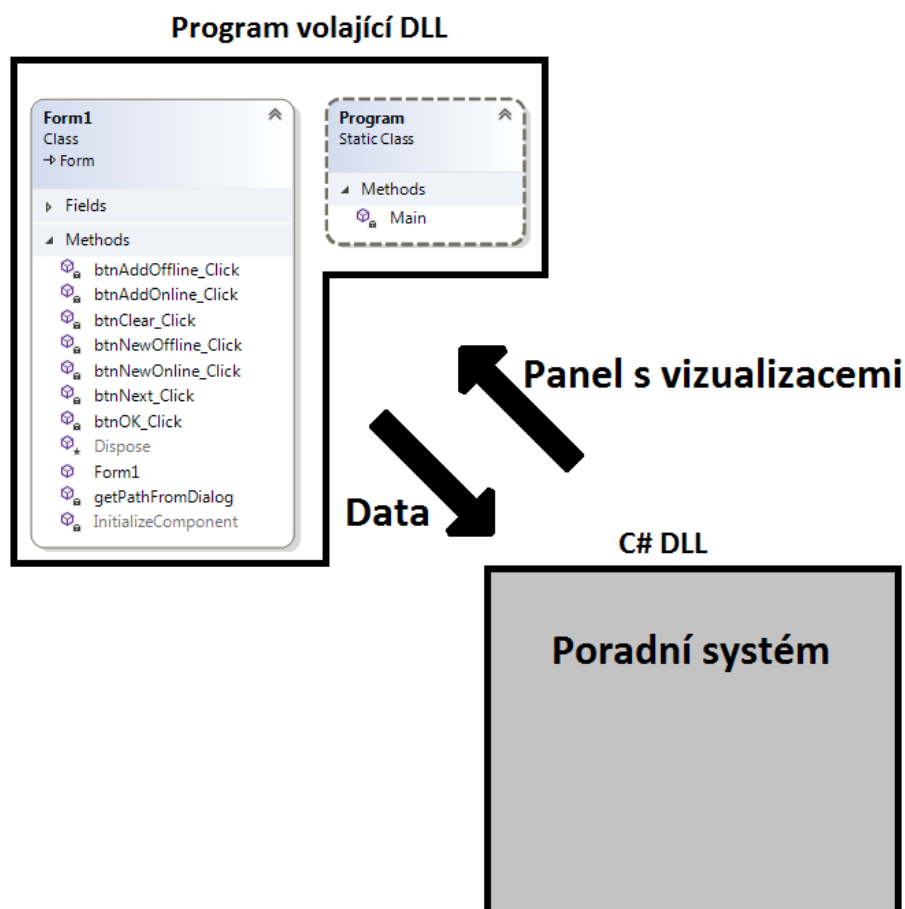
## Kapitola 7

### Implementace Poradního systému v C#

Jako programovací jazyk jsem si vybral C#, protože se jedná o velmi rozšířený vyšší jazyk, kde se nemusíme starat o správu paměti. Hlavní logika a jádro aplikace bude v dynamické knihovně dll, která bude načtena a volána „zvenku“. Budeme volat funkce knihovny, které se postarají o funkcionalitu systému. Jelikož naše aplikace bude vytvořena jako dll, tak bude implementace omezená pouze na operační systém Windows. Dll je zkratka z anglického slova „Dynamic Link Library“. Jde o typ souboru obsahující funkce, které mohou jiné programy volat. Slovem dynamický se poté myslí, že knihovna je načtena až ve chvíli, kdy jí program zavolá během programu. Na rozdíl od statické knihovny, která je v paměti načtena od začátku běhu programu. Tento typ byl vytvořen společností Microsoft pro operační systém Windows.

#### 7.1 Struktura programu

Program tedy bude rozdělen na dvě části. Na hlavní část, kterou je dynamická dll knihovna, v ní budou veškeré navržené výpočty a zpracování FFT signálu, a na druhou část, která bude knihovnu volat a zobrazovat výsledné zpracování. Je rozumné tyto programy od sebe oddělit z důvodu lepší kompatibility s již vzniklými programy komunikující s motorem. Budeme-li chtít implementovat náš program do nějakého již zaběhlého systému, stačí pouze načíst naši knihovnu a zavolat potřebné funkce pro předání dat a zobrazení vizualizace. Data pro poradní systém jsou offline data (neklepající data, vytyčení mezí), online data (aktuální naměřená data, klepající), velikost a zpoždění okénka. Vizualizace je zprostředkována objektem Panel, který je třídy Sys-



Obrázek 7.1: Struktura programu

tem.Windows.Forms. Pro podporu je tedy nutné, aby program volající DLL byl schopný tento Panel zobrazit. V našem případě se jedná o jednoduchý program v C#, který pomocí tlačítek volá funkce knihovny a zobrazuje Panel na příslušném místě. Volání knihovny je zobrazeno na Obrázku 7.1.

### 7.1.1 Popis komponent programu

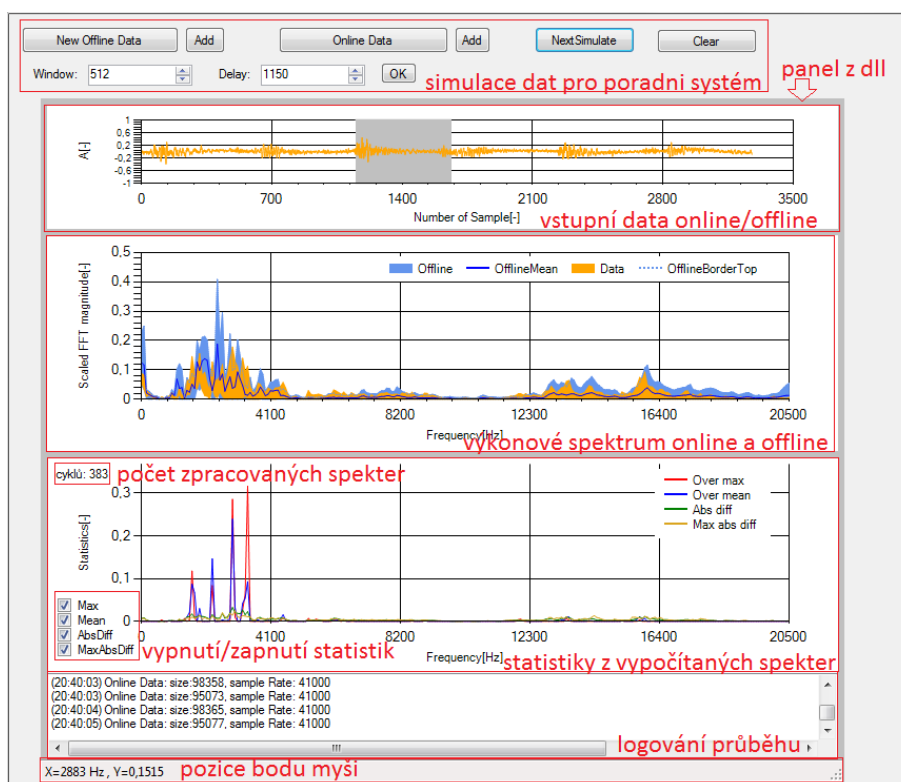
Na Obrázku 7.2 můžeme vidět spuštěný program s popisem jednotlivých jeho částí. Program volající dll se stará o simulaci dat z obrázku popsané jako „simulace dat pro poradní systém“. Zde je tlačítko pro načtení offline dat (neklepající data) nazvané „New Offline Data“, které při kliknutí otevře dialog s vybráním audio souboru pro výpočet offline dat. Tento dialog podporuje i výběr více souborů najednou. Tím můžeme spojit více měření do jednoho dlouhého měření. Data jsou automaticky oříznuta na začátku i na konci na

celé cykly, tudíž nedojde k nějaké neočekávané chybě při naměření neúplného cyklu. Tlačítkem „Online Data“ načteme data pro simulaci reálného měření z motoru. Opět zde můžeme vybírat více dat, které pak program spojí na jeden dlouhý záznam. Tyto data jsou však pouze uložena do operační paměti a nejsou poslána do dll knihovny. Spouštěcím signálem je tlačítko „Next Simulate“, které odešle do knihovny poradního systému první vzorek online dat o velikosti 100 tisíc prvků. Po opětovném stisknutí odešle další vzorek v pořadí stejné velikosti a tak dále, než nedojde na konec načtených online dat. Offline a online je možné rozšířit stisknutím tlačítka „Add“ o další měření, které bude přidáno na konec původních dat. Vždy při přijetí offline nebo online dat knihovnou, je do logovacího okénka zapsána velikost a typ zpracovaných dat. Pod těmito tlačítky máme kolonky pro nastavení velikosti okénka (počet vzorků pro FFT) a zpoždění (v počtech vzorků) od náběžné hrany prvního válce. Stisknutím tlačítka „OK“ jsou nastavení velikosti a zpoždění odeslány do knihovny poradního systému. Posledním tlačítkem „Clear“ vymažeme veškerá data, nastavení velikosti a zpoždění uložená (pouze) v knihovně.

Poradní systém vrací panel na němž jsou zobrazeny 3 grafy, logovací okénko a lišta s popisem o pozici. Prvním grafem je graf zobrazující aktuálně poslaná data do knihovny. Zobrazí vždy první celý cyklus v datech. Poté k těmto datům vykreslí vybrané okénko podle parametrů velikosti a zpoždění. To nám pomůže lépe vybrat okénko cyklu. Druhým grafem je zobrazení výkonového spektra dat. Pro první i druhý graf **modrá** barva značí offline data a **oranžová** barva online data. Data obsahují určitý počet cyklů, z nichž každý 1 cyklus nám dá právě jedno výkonové spektrum. V grafu jsou zobrazeny meze jako plocha pro offline data a online data. Pro offline data je dolní mez stanovena jako minimální hodnota spektra a horní mez jako kvantil 99% ze spekter offline dat. Pro online data jsou meze stanovené čistě jako minimální a maximální hodnota výkonového spektra na dané frekvenci. Poslední významným průběhem je střední hodnota výkonových spekter offline dat a je značena tmavě modrou barvou. Třetím grafem jsou zobrazené normované statistiky, které jsou popsány v sekci 6.3. V levém horním rohu grafu je počítadlo zpracovaných spekter pro přehled rozsahu statistiky. V levém dolním rohu je pak možnost jednotlivé statistiky zapínat a vypínat.

## 7.2 Diagram tříd knihovny

Knihovna je programována objektově a obsahuje několik tříd. Diagram můžeme vidět na Obrázku 19. Knihovna obsahuje celkem 8 tříd, z nichž třídy ASManager a ASData jsou veřejné a právě s nimi bude pracovat vnější program volající tuto knihovnu. V této sekci si popíšeme funkcionalitu jednotlivých



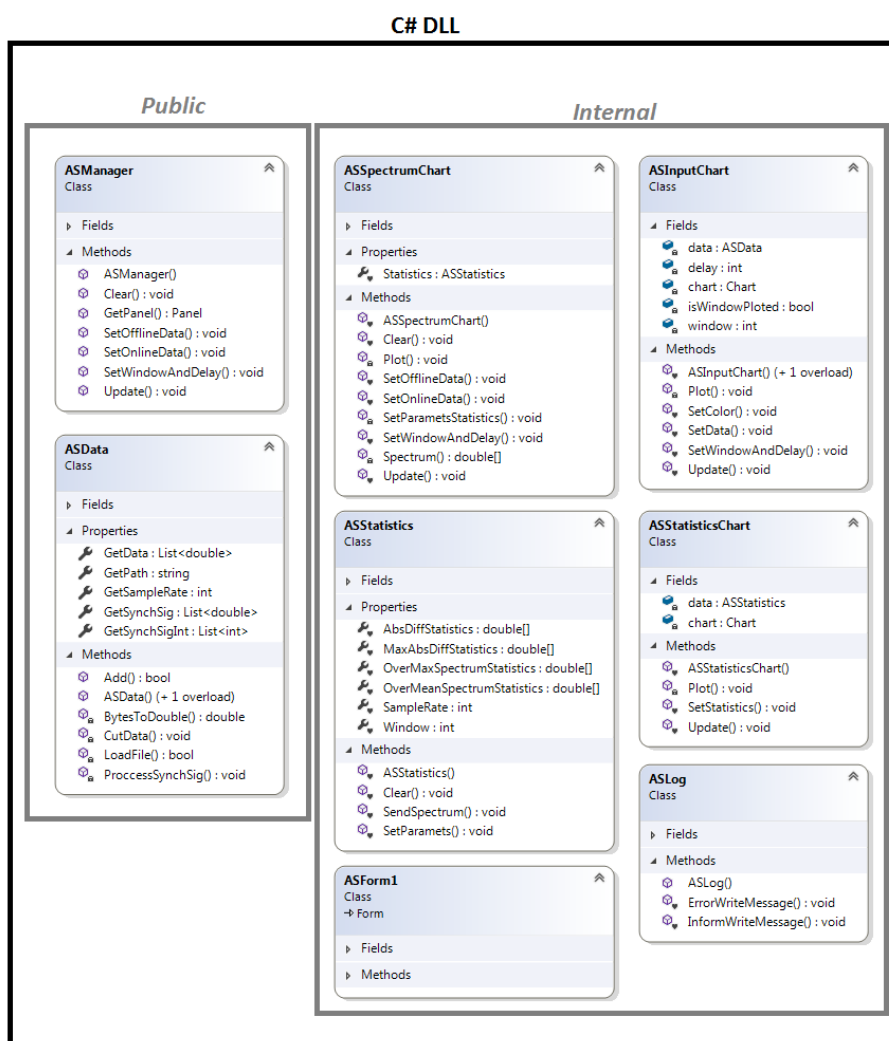
Obrázek 7.2: Popis prvků programu

tříd v knihovně.

#### ■ ASManager

Jedná se o veřejnou třídu přes kterou vedeme hlavní komunikaci. Obsahuje konstruktor bez parametrů, a metody `SetOfflineData`, `SetOnlineData`, `SetWindowAndDelay`, `Update`, `GetPanel` a `Clear`. Pomocí metody **SetOfflineData** nastavíme krajní meze pro porovnání s online daty. Při každém přiřazení dat jsou data vykreslena až po zavolání metody `Update`. Data jsou vykreslena modrou barvou do grafu výkonových spekter. Jako parametr je třída `ASData`, která je popsána níže. Chceme-li offline data pouze rozšířit o nějaký úsek, je nutné vložit veškerá data včetně nového úseku. Metodou **SetOnlineData** nastavujeme nově naměřený úsek, který opět po zavolání metody `Update` vykreslí online data oranžovou barvou. Online data zatím není možné vykreslit, pokud nejsou nastavena offline data. Metoda **SetWindowAndDelay** má dva parametry typu `integer`, kterými nastavíme velikost a zpoždění vybraného okénka. To je nutné nastavit před zavoláním metody `Update`. Metoda **Update** je bez parametru a nám nastavená offline/online data zpracuje a zobrazí v grafech. Stejně jako změna nastavení velikosti a zpoždění okénka se projeví až po zavolání metody `Update`. Veškeré zobrazení se provádí na panelu, který vrací metoda `GetPanel`. Na panelu jsou poté komponenty grafu, které





Obrázek 7.3: Diagram tříd pro poradní systém

jsou třídy `System.Windows.Forms.DataVisualization.Charting.Chart`, která je dostupná až od verze `.NET framework 4.0`. Metoda `Clear` vymaže nastavená online a offline data a nastavení velikosti okénka a jeho zpoždění.

### ■ ASData

Touto třídou specifikujeme formát vstupních dat pro poradní systém. Data se vytváří přes konstruktor dvěma způsoby. Zavoláním konstrukturu s parametrem cesty k souboru typu `string` nebo zavoláním konstrukturu s prvním parametrem typu `List<double>`, který obsahuje naměřená data, druhým parametrem opět typu `List<double>`, který obsahuje naměřená data synchronizačního signálu (náběžná hrana signálu určuje první válec v horní úvrati) a třetím parametrem je vzorkovací frekvence, kterými byla data navzorkována. Data vytvoří novou proměnou `synchSigInt` typu `List<int>`. Jde o zpracovaná data ze synchronizačního

signálu, kde náběžná hrana má hodnotu 1 jinak 0. Při tomto převodu se provede kontrola, že náběžné hrany byly detekovány správně. Algoritmus otestuje, jestli maximální délka cyklu v datech je větší než dvojnásobek nejmenšího cyklu v datech. Pokud tomu tak je, vyhodí chybovou hlášku viz 7.3. Poradní systém tedy předpokládá stacionární rychlost motoru, a pokud je načteno měření, kdy motor zrychluje či brzdí, tak změny ve velikosti naměřených cyklech budou zanedbatelné. To můžeme zařídit vhodně zvolenou velikostí dat. Pokud ovšem chceme vytvořit dlouhá data typu ASData s kolísavou velikostí cyklů, jde to udělat pomocí metody **Add**. Tato metoda má jako parametr data typu ASData, které chceme přidat k aktuálním datům. Metoda akorát otestuje shodnou vzorkovací frekvenci a data spojí. Veškerá vytvořená data typu ASData jsou oříznuta na začátku a na konci, tak aby začínala náběžnou hranou pro první cyklus a končila posledním datem posledního celého cyklu. Privátní metoda **LoadFile** načte zvukový soubor dané jako parametr. Tato metoda přiřadí 1. zvukový kanál do k datům a 2. kanál k synchronizačnímu signálu. Metoda podporuje pouze čtení souborů \*.au a \*.wav, které jsou kódovány pomocí PCM 16bit. Pokud soubor nesplňuje tyto podmínky je vrácen NotImplementedException.

#### ■ ASInputChart

Třídy končící slovem „Chart“ obsahují vždy metody Update a Plot. Tato třída vykresluje první cyklus nastavených dat typu ASData, které jsou atributem této třídy. Metoda **Update** zavolá metodu **Plot** a ta data vykreslí do přidělené komponenty Chart. A pro správné využití této třídy musí mít vstupní parametr typu Chart správně nastavené určité parametry, které jsou uloženy v třídě ASForm1.

#### ■ ASSpectrumChart

Hlavními atributy této třídy jsou online data a offline data. Při zavolání metody **Update** jsou vypočítána výkonová spektra jednotlivých zvolených okének. Při zpracování dat jsou spektra jednotlivých okének cyklu uložena do proměnné typu List<double>[] (proměnné spektrumDataOffline a spektrumDataOnline). Jde o dvourozměrné pole, kde jedním rozměrem je počet frekvencí v okénku (ta je konstantní a je daná velikostí okénka) a druhým rozměrem je počet cyklů/spekter v příslušných datech. Pomocí této proměnné pak určíme meze pro offline a online data. Pro offline data vybíráme kvantil 99% pro maximální mez pomocí seřazení pole. Dolní mez je vybrána jako minimální hodnota pole. U online dat vybíráme první a poslední prvek ze seřazeného pole, tedy maximum a minimum. Metoda **Plot** vykreslí meze pro offline a online data pomocí modré a oranžové plochy a střední hodnotu offline výkonových spekter pro každou frekvenci tmavě modrou barvou. K této třídě je přiřazen také atribut třídy ASStatistics, kam po každém výpočtu výkonového spektra okénka posíláme vypočítané spektrum, pro aktualizování statistiky. Nastavení mezí je prováděno pomocí metody třídy ASStatistics a to je prováděno při každé změně offline dat.

### ■ ASStatistics

Tato třída obsahuje statistická data z vypočítaných spekter. Spektra jsou do třídy odeslána pomocí metody **SendSpectrum**. Metoda aktualizuje statistiky pomocí algoritmů popsanych v sekci 6.3. Metoda **Clear** statistiky vymuluje. Statistika také obsahuje atribut `size` typu `integer`, která nám říká z kolika spekter je statistika utvořena a je inkrementováno při přijetí výkonového spektra. Toto číslo je také zobrazeno na grafu zobrazující tuto statistiku spolu se slovem „cycles“.

### ■ ASStatisticsChart

Tato třída se stará o vykreslení statistiky uložené ve třídě `ASStatistics`. Statistická data jsou znormována na součet 1, aby statistiky byly navzájem porovnatelné, a vykreslí je pomocí metody **Plot**. Dále se stará o číselnou hodnotu labelu, který počítá velikost statistiky.

### ■ ASForm1

Tato třída se stará o vytvoření a pozicování veškerých grafických komponent v aplikaci, především obecné nastavení komponent `Chart`. Při vytváření panelu ve třídě `ASManager` je poté zavolána instance panelu této třídy. To má své výhody a nevýhody. Výhodou je určitě to, že pokud chceme upravit/ přidat nějaké nastavení můžeme to udělat přímo v grafickém prostředí této třídy při vývoji metodou „naklikání“ parametrů. Nevýhodou je to, že třídy, kam posíláme datový typ `Chart`, pak ztrácí na univerzálnosti použití a rozhodně se nejedná o optimální přístup, neboť například samotný formulář této třídy není nikde použit. V této třídě také nastavujeme schopnost grafu funkci `zoom`, kterou použijeme podržením levým tlačítkem myši. Oddálení provedeme kliknutím na malé kolečko vedle posuvníku. V této třídě je nad každým grafem vytvořen `EventHandler` na pohyb myši, který zavolá metodu `chart_MouseMove`. Ta zobrazuje v dolní liště pozici bodu, na které právě ukazuje myš.

### ■ ASLog

Třída zapisuje chybové a informační zprávy o průběhu zpracování dat. Obsahuje statickou metodu **ErrorWriteMessage**, kterou můžeme volat bez vytvoření instance. Metoda přidá zprávu do souboru `ErrorLog.txt`, který je vytvořený (nebo se vytvoří) v adresáři se samotným programem. Zpráva je předána pomocí parametru. Třída vypisuje i informační zprávy, do komponenty `ListBox`, která se nachází v dolní části aplikace, pomocí metody **InformWriteMessage**. Jelikož pracujeme s instancí třídy, už nemůžeme použít statickou metodu. `ListBox` do kterého zapisujeme je atributem této třídy. Zavoláním metody `InformWriteMessage` přidáme řádek zprávy z parametru této metody do `ListBoxu`.

## 7.3 Možné vzniklé chybové hlášení

- Chyby v důsledku špatného vstupního souboru

`Bad detection rising edge of sync signal. More ErrorLog.txt.`  
 Chyba nastane, pokud maximální délka cyklu v datech je větší než dvojnásobek nejmenšího cyklu v datech. Knihovna vyhodí `Exception` s touto chybovou hláškou. Do souboru `ErrorLog.txt` jsou poté vypsány jednotlivé detekované délky cyklu. Pokud jsou data na tolik dlouhá, že obsahují cyklus o rychlosti  $v$  a zároveň o rychlost  $2v$ , tak jedinou možností jak se této hlášky vyvarovat je rozdělit data, tak aby tuto podmínku nesplnili a poté je spojit metodou `Add` v třídě `ASData`.

`Encoding is not supported, Audio Format is not supported`  
 Knihovna vyhodí `NotSupportedException` s touto chybovou hláškou, když se snažíme načíst data ze souboru, který není modulován 16-bitovou pulzně šířkovou modulací nebo pokud zvolený zvukový soubor je jiného formátu než `*.wav`, `*.au`.

- Programové chyby v důsledku špatného volání funkcí knihovny

`Window size is zero`

Pokud zvolíme nulovou velikost okna a voláme metodu `Update`

## 7.4 Postup pro správné volání knihovny

- Načtení offline dat a vytyčení mezí – Použijeme postupně metody `SetOfflineData`, `SetWindowAndDelay` a `Update` pro vykreslení offline dat. Pro rozšíření/přidání nových offline dat je nutné celý tento postup opakovat, ale na nově vytvořená `ASData`, které vzniknou spojením předchozích dvou a využití funkce `Add`.
- Posílání online dat při měření motoru – Použijeme metodu `SetOnlineData`, `SetWindowAndDelay` a `Update`. Samotná data jsou zpracována až po zavolání metody `Update`. Pro poslání dalších dat je možné změnit referenci na nová data typu `ASData` a opět zavolat metodu `Update`, nebo nová data opět nastavit stejně jako u prvních dat.
- Chceme-li změnit nastavení okénka a zpoždění musíme vždy spustit metodu `Clear`, která vymaže vykreslené grafy a odstraní data a poté viz předchozí kroky meze i data opět nastavit.

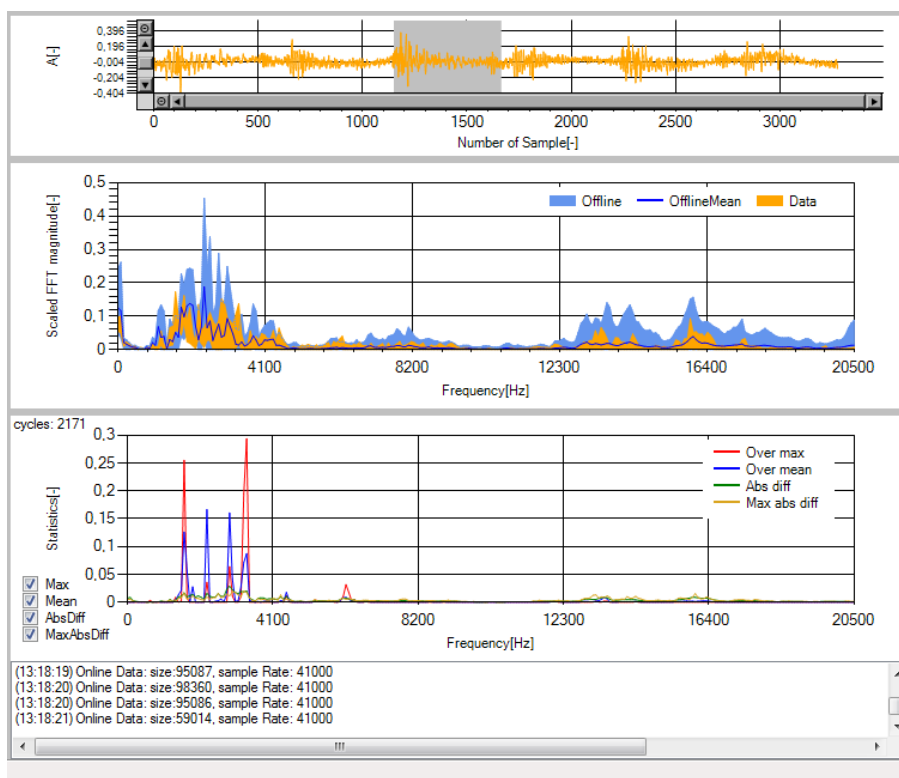
## Kapitola 8

### Dosažené výsledky

#### 8.1 Experimenty na reálných datech

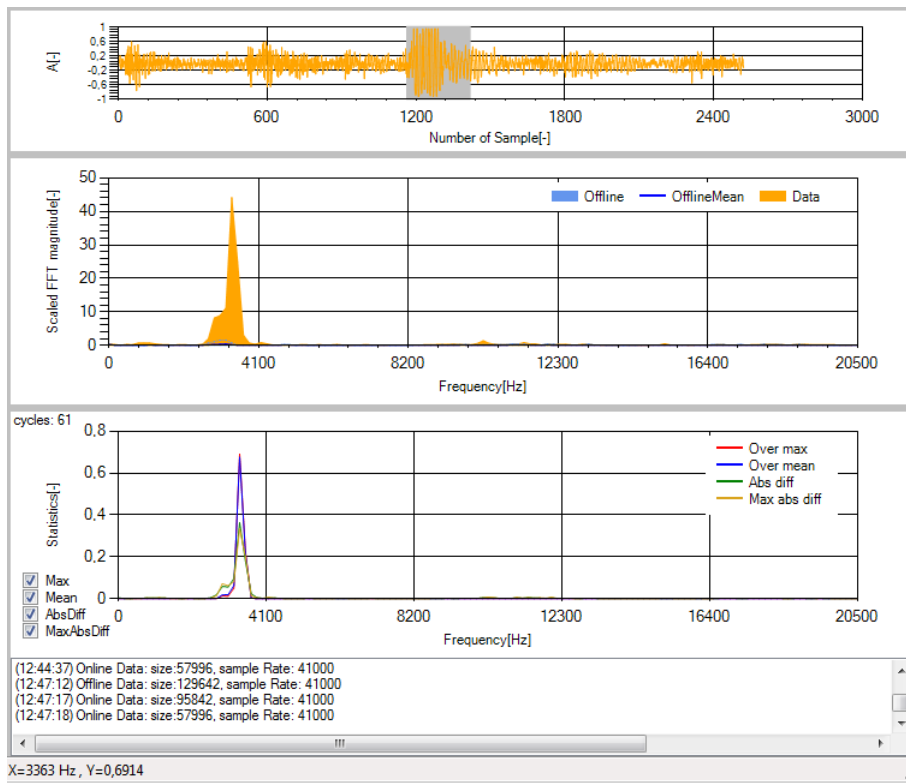
Otestujeme naše naměřená data z laboratorního měření. Vnitřní průměr válce je pro naše měření 105 mm. Podle Draperovi rovnice odhadneme klepací frekvenci 5023 Hz. Tento motor jsme měřili v různém poměru nafta/plyn a předpokládáme, že pro měření 100% nafty ke klepání nedocházelo. Vložíme tyto data jako offline měření a stanovíme meze. Jako online měření zvolíme naopak nejvyšší zastoupení plynu, tedy 65% plynu. Tento motor je 3 válcový a máme celkem 2 snímače A a B. Provedeme tedy celkem 6 experimentů pro každý snímač a válec zvlášť. Okénko volíme tak, aby bylo kousek po výbuchu ve spalovací komoře. Na Obrázku 8.1 jsou zpracovány data pro druhý válec snímače A. Podíváme-li se na statistiku kolem odhadované klepací frekvence, nevidíme žádnou frekvenci, která by převyšovala hranici 10%. Ani výkonové spektrum online dat kolem této frekvence nedominuje. Můžeme s velkou pravděpodobností tvrdit, že při našem měření na tomto válci nedošlo ke klepání. To samé provedeme pro ostatní data nad ostatními válci a zjistíme, že žádná frekvence nedominuje. Při laboratorním měření pravděpodobně nedošlo k rozklepání motoru.

Samé měření jsem provedl i na datech, které jsem dostal k dispozici, avšak ke kterým nemám podrobné údaje o typu motoru ani velikosti vnitřního vrtání válce (Obrázek 8.2). Tyto data obsahují klepání. Zde je klepání tak dominantní, že je průkazné hned ve výkonovém spektru online dat. Klepací frekvence pro tyto data určíme jako maximum vybraných statistik. Klepací frekvence je 3363 Hz. Výkonové spektrum pro online data u klepací frekvence



**Obrázek 8.1:** Experiment na offline data diesel/ Online data 65% plyn snímač A

je přibližně 20x vyšší než u neklepajících dat. Statistiky Over max spolu s Over mean a Abs diff spolu s Max abs diff se téměř prolínají.



**Obrázek 8.2:** Experiment na klepajících datech D-20\_11-04-2012\_17-59-07\_#05.1\_knocking.au





## Kapitola 9

### Závěr (Rozšíření práce)

V této práci byl prostudován jev klepání ve spalovacích motorech. Klepání způsobuje zapálení okolních míst nevyhořelé směsi, způsobené jinak než plynnule rozšiřujícího se plamene jdoucí od svíčky. Dojde k samovznícení směsi a to buď lokálním zvýšením tlaku a teploty při rozšiřování plamene nebo jako předčasné vznícení před zapálením svíčky například od rozpálené svíčky. Plamen ve válci se může šířit dvěma způsoby a to detonací a deflagrací. Klepání může nabývat různé síly, kterou určujeme velikostí změny tlaku ve válci s tím, že detonace většinou nabývá větší síly. Oscilace pístu při klepání jsou dána mechanickou konstrukcí, které můžeme odhadnout pomocí tzv. „Draperovi rovnice“, která nám podle průměru válce specifikuje 5 frekvencí, z nichž ta nejdominantnější je nazývána jako klepací frekvence.

Bylo provedeno měření na motoru a zaznamenáno přibližně 30 minut záznamu, při kterém pravděpodobně nedošlo k žádnému výraznému klepání. Byl vytvořen poradní systém pro detekci klepací frekvence psaný v C#. Hlavní jádro programu je dll knihovna, která zpracuje data z piezoelektrických snímačů poblíž válce motoru a vrátí panel se třemi grafy. V prvním grafu první cyklus motoru načtených neupravených dat. V druhém grafu výkonová spektra s mezemi pro běžný/offline průběh motoru a aktuálně/online měřený a v posledním grafu vyhodnocení statistik a možné potenciální klepací frekvence motoru. Tato dll knihovna může být připojena k jiným programům komunikující přímo s motorem pro vyhodnocení dat za běhu motoru. K této knihovně je vytvořen program simulující běh motoru, který načítá data do knihovny pomocí uložených zvukových souborů ve formátech \*.au a \*.wav, ve kterých jsou uložena data ze snímačů a synchronizační signál, detekující první válec v horní úvratí motoru. Pro laboratorní měření nebyla detekována jedna určitá frekvence okolo odhadnuté klepací frekvence pomocí Draperovi rovnice. Podle výsledků můžeme s jistou pravděpodobností usuzovat, že ke klepání v našem měření nedocházelo a pokud ano tak v zanedbatelné míře.





## Literatura

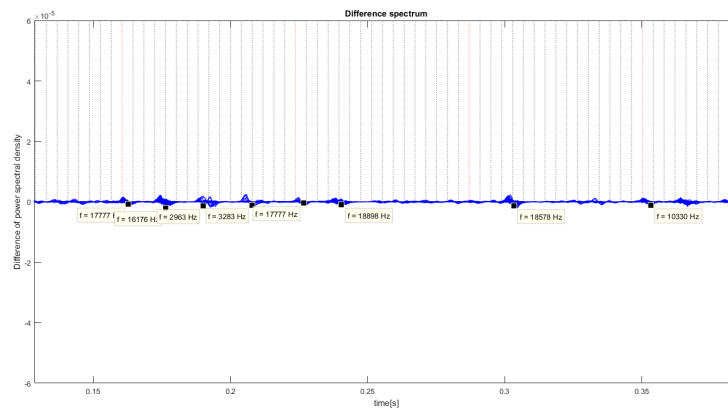
- [1] Y. S. M. S. Tatsuya Ibuki, Yasuhiro Awai and J. Kako, “Knocking detection in gasoline engines based on probability density functions: A mixed gaussian distribution approach,” *IEEE 54th Annual Conference on Decision and Control (CDC)*, 2015.
- [2] J. M. S. James C. Peyton Jones and J. Frey, “Likelihood-based control of engine knock,” *IEEE TRANSACTIONS ON CONTROL SYSTEMS TECHNOLOGY*, VOL. 21, NO. 6, 2013.
- [3] *Engine knocking* [https://en.wikipedia.org/wiki/Engine\\_knocking](https://en.wikipedia.org/wiki/Engine_knocking).
- [4] R. D. R. Zhi Wang, Hui Liu, “Progress in energy and combustion science,” *Journal of The Combustion Institute*, 2017.
- [5] X. H. S. S. Zhi Wang, Hui Liu, “Relationship between super-knock and pre-ignition,” *International Journal of Engine Research*, 2015.
- [6] J. Fiolka, “Fractional fourier transform and its application to engine knock detection,” *22nd International Conference "Mixed Design of Integrated Circuits and Systems"*, 2015.
- [7] M. o. G. T. S. Thomas G. Horner, “Engine knock detection using spectral analysis techniques with a tms320 dsp,” *Texas Instruments*, 1995.
- [8] I. Andersson, “Cylinder pressure and ionization current modeling for spark ignited engines,” *Division of Vehicular Systems Department of Electrical Engineering*, 2002.
- [9] S. M. Khurram Kamal, Mohammad Farid, “Detection of knocking in spark ignition (si) engines using cmac neural networks,” *IEEE magazine*, 2012.

- [10] A. K. Kenji Akimoto, Hirotaka Komatsu, “Development of pattern recognition knock detection system using short-time fourier transform,” *Symposium on Advances in Automotive Control*, 2013.
- [11] A. T. K. Colin R. Ferguson, *Internal Combustion Engines: Applied Thermosciences 3rd edition*. John Wiley & Sons, Inc., 2016.
- [12] S. S. E. Galloni, G. Fontana, “Numerical and experimental characterization of knock occurrence in a turbo-charged spark-ignition engine,” *Elsevier*, 2014.
- [13] *Knock Calculator*:<https://www.phormula.com/KnockCalculator.aspx>.
- [14] *Operation and Maintenance Manual Industrial Engines*, 2008.
- [15] *Au Format*: [https://en.wikipedia.org/wiki/Au\\_file\\_format](https://en.wikipedia.org/wiki/Au_file_format).
- [16] *Wave format*: <https://web.archive.org/web/20141213140451/>  
<https://ccrma.stanford.edu/courses/422/projects/WaveFormat/>.

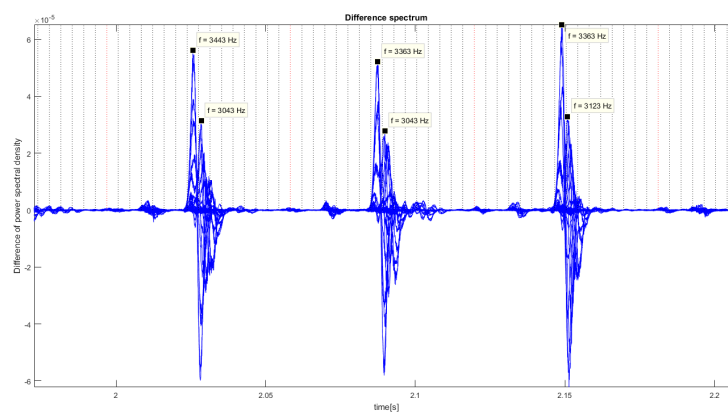




## Přílohy



**Obrázek 1:** Detekce klepání pomocí experimentální metoda porovnání změny výkonových spekter v čase v MATLABU - Neklepající data



**Obrázek 2:** Detekce klepání pomocí experimentální metoda porovnání změny výkonových spekter v čase v MATLABU - Klepající data