

**Západočeská univerzita v Plzni**  
**Fakulta aplikovaných věd**  
**Katedra kybernetiky**

# **BAKALÁŘSKÁ PRÁCE**

**PLZEŇ, 2013**

**Vojtěch Vašíček**

zadání

## PROHLÁŠENÍ

Předkládám tímto k posouzení a obhajobě bakalářskou práci zpracovanou na závěr studia na Fakultě aplikovaných věd Západočeské univerzity v Plzni.

Prohlašuji, že jsem bakalářskou práci vypracoval samostatně a výhradně s použitím odborné literatury a pramenů, jejichž úplný seznam je její součástí.

V Plzni dne 9.5.2013

.....  
*vlastnoruční podpis*

## PODĚKOVÁNÍ

Na tomto místě bych velice rád poděkoval panu Ing. Jindřichu Liškovi, Ph.D za výborné odborné a osobní vedení práce, za trpělivost a ochotu při vedení jednotlivých konzultací a za věcné připomínky, které vedly k vypracování této práce.

Dále bych rád poděkoval panu Ing. Janu Jaklovi za poskytnutí dat z monitorovacího systému rubbingu a připomínky při testování aplikace.

Velký dík patří také mým rodičům, kteří mě při studiu podporovali.

## **Anotace**

Bakalářská práce se zabývá zpracováním dat diagnostického systému rubbingu, který monitoruje kontakt rotoru a statoru na parní turbíně, a předchází tak možným materiálním a ekonomickým ztrátám ve výrobě elektrické energie. Pro data z tohoto systému byl navržen efektivní způsob archivace do zvolené databáze Microsoft SQL Server. Pomocí aplikace RAMS Analyzer implementované v LabView je k těmto datům umožněn rychlý přístup a jejich snadná vizualizace pro potřeby monitorování časového průběhu jednotlivých měřených veličin. Navržená aplikace také umožňuje statistický report ukazující na místa nejvíce zatížená kontaktem rotoru a statoru - rubbingem.

## **Klíčová slova**

Rubbing, databáze, LabView, RAMS Analyzer

## **Annotation**

This bachelor thesis deals with the data processing of rubbing diagnostic system that monitors the contact of rotor and stator on a steam turbine to avoid possible material and economic losses during electricity production. The data from this system are effectively archived into Microsoft SQL Server database. RAMS Analyzer - application implemented in LabView - allows quick access and easy visualization of archived data in order to monitor the time behaviour of the measurements. The proposed application also allows statistical report that shows the areas where is the contact rotor and stator - rubbing - most common.

## **Keywords**

Rubbing, database, LabView, RAMS Analyzer

## Obsah

<b>1. ÚVOD</b>	<b>8</b>
<b>2. ÚVOD DO PROBLEMATIKY</b>	<b>10</b>
2.1. Analytický přístup k diagnostice rotujících částí .....	10
2.2. Měření výchylky středu disku a způsoby její reprezentace .....	11
2.2.1. Orbity .....	12
2.2.2. Fázory .....	13
2.2.3. Precese .....	13
2.3. Rubbing .....	15
2.3.1. Typy rubbingu .....	16
2.3.2. Projevy rubbingu .....	17
<b>3. STÁVAJÍCÍ STAV SYSTÉMU</b>	<b>19</b>
3.1. Hardware .....	19
3.2. SW pro online detekci rubbingu .....	20
3.3. LabView .....	21
3.4. Výstup stávajícího systému .....	22
<b>4. MOŽNOSTI ARCHIVACE DAT</b>	<b>25</b>
4.1. Microsoft SQL Server .....	25
4.2. Microsoft Office Access .....	26
4.3. SQL .....	26
4.4. ODBC .....	27
<b>5. IMPLEMENTACE SYSTÉMU ARCHIVACE DAT</b>	<b>29</b>
5.1. Formát tabulek databáze .....	29
5.2. Zápis do databáze - online data .....	31
5.3. Zápis do databáze - offline data .....	33
<b>6. RAMS ANALYZER – NÁSTROJ PRO VIZUALIZACI</b>	<b>36</b>
6.1. Záložka GRAPHS .....	38
6.2. Záložka UPLOAD DATA .....	39
6.3. Záložka PHASORS .....	41
6.4. Záložka ORBITS .....	42
6.5. Záložka STATISTICS .....	44
<b>7. ZÁVĚR</b>	<b>48</b>
<b>8. SEZNAM POUŽITÉ LITERATURY</b>	<b>49</b>
<b>A. STATISTICKÝ REPORT</b>	<b>50</b>
<b>B. OBSAH PŘILOŽENÉHO CD</b>	<b>51</b>

## Seznam obrázků

Obrázek 1-1: Parní turbína tepelné elektrárny Tušimice [1] .....	8
Obrázek 1-2: Zjednodušený model turbíny s ucpávkami .....	9
Obrázek 2-1: Zjednodušený model rotoru [4] .....	10
Obrázek 2-2: Orbíta a filtrovaná orbíta 1X .....	12
Obrázek 2-3: Časový průběh harmonického signálu a jeho fázor .....	13
Obrázek 2-4: Orbíta časového průběhu středu rotoru rozložená na dva protichůdné fázory... 14	
Obrázek 2-5: Rubbing [7] .....	15
Obrázek 2-6: Typy rubbingu [4] .....	16
Obrázek 2-7: Částečný rubbing - waterfall graf [7] .....	17
Obrázek 2-8: Vliv rubbingu na otáčkovou rychlost a počáteční fázi .....	18
Obrázek 2-9: Vliv rubbingu na počáteční fázi a amplitudu – fázorový diagram .....	18
Obrázek 3-1: NI PXI-1031 .....	19
Obrázek 3-2: NI PXI-1031 a NI PXI-8108 .....	20
Obrázek 3-3: Umístění čidel relativní rotorové vzdálenosti [10] .....	20
Obrázek 3-4: RAMS Online .....	21
Obrázek 3-5: Prostředí LabView .....	22
Obrázek 3-6: Výřez stávajícího výstupního souboru trendy.txt .....	23
Obrázek 3-7: Hanningova okénková funkce .....	24
Obrázek 3-8: Výřez stávajícího výstupního souboru log2.txt .....	24
Obrázek 4-1: Grafické uživatelské rozhraní SQL Server Management Studio .....	25
Obrázek 4-2: Použití SQL příkazu SELECT v prostředí SQL Server Management Studia ....	27
Obrázek 4-3: Komunikace mezi LabView a databází [13] .....	28
Obrázek 5-1: Vývojový diagram zápisu dat do tabulky EVENT .....	32
Obrázek 5-2: Algoritmus zápisu online dat do tabulky EVENT .....	32
Obrázek 5-3: Subsystém vytvářející tabulku EVENT (výřez) .....	33
Obrázek 5-4: Fáze archivace offline dat ze souboru trendy.txt do databáze .....	34
Obrázek 5-5: Převod offline dat do databáze pomocí prohledávání souborového systému ....	35
Obrázek 6-1: RAMS Analyzer po spuštění .....	36
Obrázek 6-2: Připojení k databázi .....	37
Obrázek 6-3: Zobrazení průběhů veličin v záložce GRAPHS .....	38
Obrázek 6-4: Načítání souborů do databáze v záložce UPLOAD DATA .....	39
Obrázek 6-5: Ukázka uživatelského hlášení na konci načítání souborů do databáze .....	40
Obrázek 6-6: Stav aplikace po načtení souborů do databáze .....	41
Obrázek 6-7: Průběh fázoru 1X ze směru y na rovině 1 .....	42
Obrázek 6-8: Zobrazení orbitů pro každou ze šesti rovin .....	43
Obrázek 6-9: Výpočet orbitů v MathScriptu .....	44
Obrázek 6-10: Statistiky z archivovaných dat .....	45
Obrázek 6-11: Algoritmus výpočtu statistik událostí rubbingu .....	46
Obrázek 6-12: Algoritmus výpočtu statistik otáčkové frekvence .....	47
Obrázek 6-13: Princip výpočtu MISSING DATA záložky STATISTICS .....	47
Obrázek A-1: Statistický report RAMS Analyzeru .....	50

## Seznam symbolů a zkratek

$1X$	.....	první harmonická
$A_d$	.....	amplituda fázoru s dopřednou precesí pro určitou frekvenci
$A_x$	.....	amplituda signálu $x(t)$
$A_y$	.....	amplituda signálu $y(t)$
$A_z$	.....	amplituda fázoru se zpětnou precesí pro určitou frekvenci
$b$	.....	tlumení systému
$fix$	.....	počáteční fáze fázoru s dopřednou precesí
$fiy$	.....	počáteční fáze fázoru se zpětnou precesí
$frot$	.....	pole hodnot otáčkové rychlosti
$i$	.....	index dat databáze
$i_{max}$	.....	počet záznamů v databázi za zvolené časové období
$j$	.....	imaginární jednotka
$k$	.....	tuhost hřídele
$m$	.....	hmotnost disku
$m_n$	.....	hmotnost nevývažku hřídele
$n$	.....	počet již načtených řádků souboru
$N$	.....	velikost pole dat
$N_r$	.....	počet řádků načítaného souboru
$O$	.....	osa rotace hřídele
$r$	.....	hmotnost disku
$s_1$	.....	souřadnicová osa afinního systému souřadnic
$s_1(t)$	.....	časový signál měřený ve směru osy $s_1$
$s_2$	.....	souřadnicová osa afinního systému souřadnic
$s_2(t)$	.....	časový signál měřený ve směru osy $s_2$
$t_0$	.....	čas spuštění načítání ze souboru
$t$	.....	aktuální čas
$t_1$	.....	diskrétní hodnota času záznamu v databázi
$t_2$	.....	diskrétní hodnota času záznamu v databázi
$T$	.....	těžiště disku
$T_z$	.....	čas zbývající do konce načítání souboru
$x$	.....	souřadnicová osa kartézského systému
$x(t)$	.....	spojitý časový signál popisující průběh výchylky středu disku v ose $x$
$x_{dn}(t)$	.....	dynamická složka časového signálu popisujícího průběh výchylky středu disku
$x_s$	.....	výchylka disku ve směru osy $x$
$x_{st}(t)$	.....	statická složka časového signálu popisujícího průběh výchylky středu disku
$y$	.....	souřadnicová osa kartézského systému
$y(t)$	.....	spojitý časový signál popisující průběh výchylky středu disku v ose $y$
$y_s$	.....	výchylka disku ve směru osy $y$
$z(t)$	.....	komplexní spojitá veličina

$\beta$ .....	úhel mezi spojnicí těžiště a středu disku a souřadnicovou osou x
$\delta_1$ .....	úhel svírající osa $s_1$ s osou x
$\delta_2$ .....	úhel svírající osa $s_2$ s osou y
$\varphi_d$ .....	počáteční fáze fázoru s dopřednou precesí
$\varphi_x$ .....	počáteční fáze signálu $x(t)$ , nebo některé z jeho frekvencí
$\varphi_y$ .....	počáteční fáze signálu $y(t)$ , nebo některé z jeho frekvencí
$\varphi_z$ .....	počáteční fáze fázoru se zpětnou precesí
$\omega$ .....	otáčková rychlost rotoru
DDR2.....	paměť s vysokorychlostní komunikací
IEEE 1284.....	vnější paralelní sběrnice
OS.....	operační systém
RS 232.....	vnější sériová sběrnice
SATA.....	počítačová sběrnice
USB.....	vnější čtyřvodičová sběrnice umožňující připojení až 127 zařízením



## 1. ÚVOD

Tato práce se zabývá archivací diagnostických veličin monitorovacího systému rubbingu, který je vyvíjen na katedře kybernetiky a řídicí techniky Západočeské univerzity v Plzni. Tento systém je fyzicky nasazen v tepelné elektrárně Tušimice v severních Čechách a jeho hlavním úkolem je snímat hodnoty veličin potřebných k detekci a lokalizaci výskytu zadíráání - rubbingu (viz kapitola 2). Vhodný způsob archivace těchto veličin usnadní v budoucnu přístup k datům z tohoto systému.

Elektrárna Tušimice je vybavena 200 MW parní turbínou [1] (viz obr. 1-1). Ta je složena z rotující části (dále jen rotor) a z nepohyblivé stacionární části (dále jen stator). Rotor se skládá z hřídele, na které jsou umístěny olopatkovaná kola, která jsou roztáčena hnacím médiem, tj. přehřátou párou. Olopatkovaná kola jsou pevně spojena s hřídelí a přenášejí tak točivý moment do generátoru elektrické energie. Na statoru jsou umístěny směrovací lopatky, které ženou páru s nejvyšší teplotou pod konstrukčně definovaným úhlem nejprve na kola s nejkratšími lopatkami. Stator a rotor je oddělen ucpávkami (viz obr. 1-2). Správný návrh této části má velký vliv na správný chod turbíny. Pokud je ucpávka příliš malá, mezi státorem a rotorem vznikají mezery. Přehřátá pára unikající tímto prostorem pak snižuje účinnost celé turbíny. Pokud je ucpávka naopak příliš velká, dochází ke zvýšení rizika kontaktu statoru s rotorem. Tento kontakt je mezi odbornou veřejností nazýván rubbing (z anglického slova to rub - dřít).



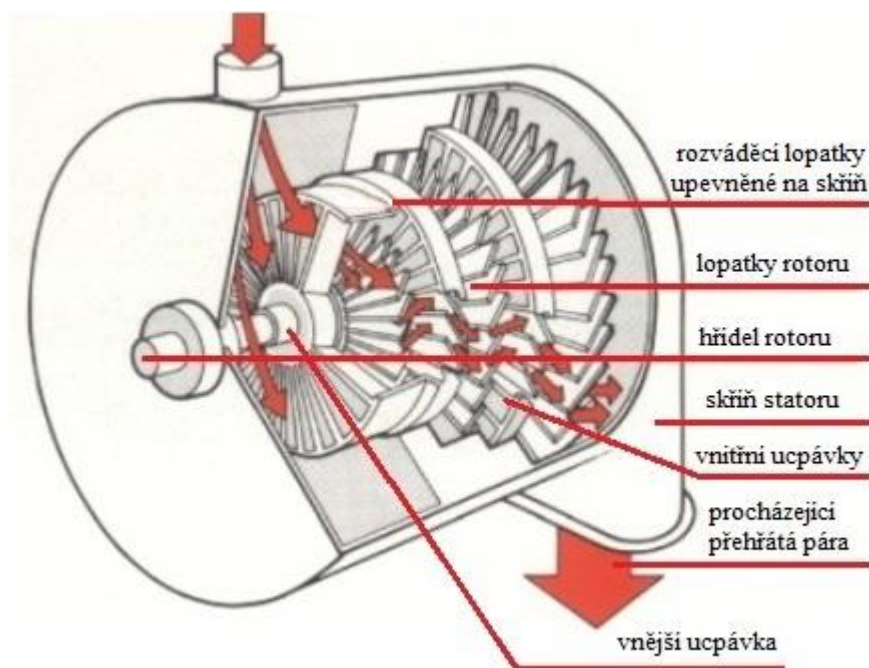
**Obrázek 1-1: Parní turbína tepelné elektrárny Tušimice [1]**

To, zda-li došlo k rubbingu, tj. zda-li došlo ke kontaktu rotoru se státorem, je důležité monitorovat a výsledky vyhodnocovat. Kontakt rotor-stator je totiž doprovázen zahřátím místa kontaktu vlivem tření. Vlivem nesymetrického ohřevu se rotor turbíny deformuje – ohýbá se – což za určitých podmínek může být doprovázeno dalším intenzivnějším kontaktem

rotor-stator. V kritickém okamžiku pak může dojít až k destrukci turbíny a k vysokým materiálním a ekonomickým škodám v důsledku zastavení výroby. Proto je vhodné kontakt rotor-stator detekovat, lokalizovat a dále analyzovat, a předejít tak možné katastrofě. Výsledky pozorování a měření je možné také použít jako zpětnou vazbu strojírenským firmám pro inovaci technologie výroby nových turbín.

Cílem této práce bude proto mimo jiné navrhnout a implementovat nástroj pro vizualizaci průběhů a charakteristik jednotlivých stavů a veličin, který vylepší uživatelský přístup k danému problému a usnadní přístup k archivovaným datům a jejich zobrazení pro následnou analýzu.

Na následujícím obrázku 1-2 (čerpáno z [3]) je ilustrován zjednodušený model turbíny, na kterém si lze všimnout dvou druhů ucpávek majících svá uplatnění na správné činnosti turbíny. Jsou jimi vnitřní ucpávky, které oddělují jednotlivé tlakové stupně, a vnější ucpávky, které oddělují samotný hřídel rotoru od skříň statoru [2].



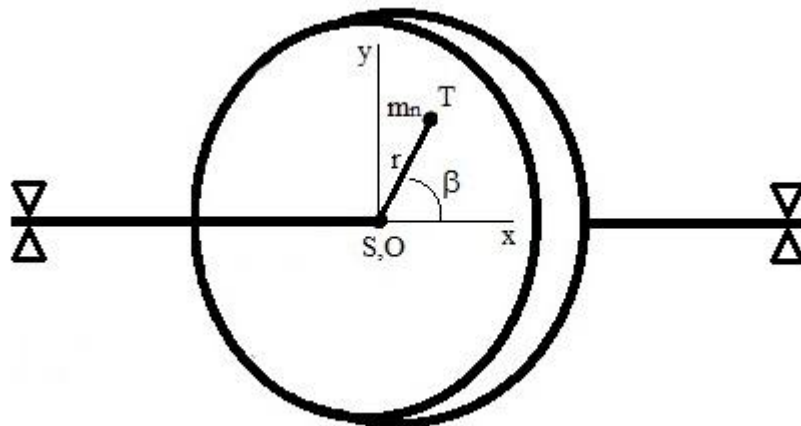
Obrázek 1-2: Zjednodušený model turbíny s ucpávkami

## 2. ÚVOD DO PROBLEMATIKY

Vyhodnocování pohybu rotujících částí může být prováděno buď analyticky za předpokladu, že známe nebo jsme schopni určit matematický model daného rotačního ústrojí, anebo měřením. Obecně lze říct, že se jedná o nelineární systémy, které jsme schopni popsat soustavou lineárních diferenciálních rovnic. Lineární systémy vzniklé linearizací systémů nelineárních však v praxi dávají dostatečně dobré výsledky. Při psaní této části jsem vycházel především ze zdrojů [4], [5], [6] a [7].

### 2.1. Analytický přístup k diagnostice rotujících částí

Jak již bylo řečeno v předchozím bodě, můžeme pohyb rotoru popsat soustavou diferenciálních rovnic [4]. Nejprve ale musíme definovat model daného systému. Předpokládejme, že je rotor reprezentován hmotným diskem o hmotnosti  $m$  a nehmotnou pružnou hřídelí. Dále uvažujme, že hmota disku není rovnoměrně rozložena okolo geometrického středu disku (v případě homogenního rozložení hmoty disku je geometrickým středem těžiště). Disk má nevyvážek o hmotnosti  $m_n$  v těžišti  $T$ , tj. ve vzdálenosti  $r$  od geometrického středu  $S$ . V kartézské soustavě souřadnic  $(x,y)$  svírá úsečka  $ST$  s osou souřadnic  $x$  úhel  $\beta$ . Osu rotace hřídele nazvěme  $O$ . Za klidu hřídele je poloha osy rotace  $O$  a středu disku  $S$  shodná. Při rotačním pohybu však na nevyvážek působí odstředivá síla, která změní polohy  $S$  vzhledem k  $O$ .



Obrázek 2-1: Zjednodušený model rotoru [4]

Uveďme nyní matematický model systému ilustrovaného na obr. 2-1 při zanedbání vlivu tíhové síly působící na disk. Odvození celého modelu a jeho řešení je velice dobře popsáno v publikaci [4]. Vyberme stěžejní rovnice popisující fyzikální charakter systému. Pohyb disku můžeme popsat dvěma diferenciálními rovnicemi druhého řádu (2.1)

$$\begin{aligned} m\ddot{x}_s(t) + b\dot{x}_s(t) + kx_s(t) &= m_n r \omega^2 \cos(\omega t + \beta) \\ m\ddot{y}_s(t) + b\dot{y}_s(t) + ky_s(t) &= m_n r \omega^2 \sin(\omega t + \beta) \end{aligned} \quad (2.1)$$

Časové proměnné  $x_s(t)$  a  $y_s(t)$  vyjadřují polohu středu disku vzhledem k ose  $x$  resp.  $y$  kartézského souřadného systému. Konstanta  $b$  značí tlumení systému,  $k$  je tuhost hřídele a  $m$  vyjadřuje hmotnost celého disku. Pro správné výsledky řešení sestaveného modelu je nutnou podmínkou přesná znalost jednotlivých parametrů, což v praxi nemusí být vždy realizovatelný požadavek. V takovém případě zhodnotíme analytický přístup jako jeden z možných přístupů řešení.

## 2.2. Měření výchylky středu disku a způsoby její reprezentace

Pokud nelze z nějakého důvodu nalézt matematický model, který by dostatečně odpovídal skutečnému systému, anebo pokud nelze určit některý z parametrů, musíme se od analytického přístupu uchýlit k měření daného systému, a tím k určení aproximativního modelu. Pro měření výchylky středu disku  $S$  je dostačující měřit výchylku  $S$  vzhledem ke klidové poloze středu disku. Rotor je vhodné osadit dvěma čidly ve směru  $s_1$  a  $s_2$ . Veličiny  $s_1(t)$  a  $s_2(t)$  reprezentují časový signál vzdálenosti osy rotace  $S$  ve směru  $s_1$  resp.  $s_2$ . Existuje pak transformační přepočítání signálů ze směrů  $s_1$  a  $s_2$  do kartézských souřadnic  $(x, y)$  [4] uvedený pomocí následujících rovnic (2.2)

$$x(t) = \frac{s_2(t) \sin \delta_1 - s_1(t) \sin \delta_2}{\sin(\delta_1 - \delta_2)} \quad (2.2)$$

$$y(t) = j \frac{s_1(t) \cos \delta_2 - s_2(t) \cos \delta_1}{\sin(\delta_2 - \delta_1)}$$

Signály  $x(t)$  a  $y(t)$  vyjadřují časový průběh výchylek středu disku v kartézských souřadnicích  $(x, y)$ . Tyto signály můžeme rozložit na dvě části [4]

$$x(t) = x_{st}(t) + x_{dn}(t) \quad (2.3)$$

$$y(t) = y_{st}(t) + y_{dn}(t)$$

$x_{st}(t)$  resp.  $y_{st}(t)$  v rovnicích (2.3) interpretuje statickou složku signálu, která zahrnuje průběhy, které nezávisí na otáčení disku. Tato složka zahrnuje střední hodnotu signálu, což je konstantní funkce vyjadřující střední vzdálenost povrchu disku od čidla. Dále pak obsahuje neharmonický periodický signál, který je dán nerovností povrchu měřeného disku. Tento průběh můžeme určit, když měřený disk budeme pomalu otáčet. Při takovém otáčení se neprojeví odstředivá síla a čidlo bude měřit pouze nerovnosti disku. Dynamické složky signálů  $x(t)$  a  $y(t)$  mají harmonický charakter. Zapišeme je ve tvaru (2.4)

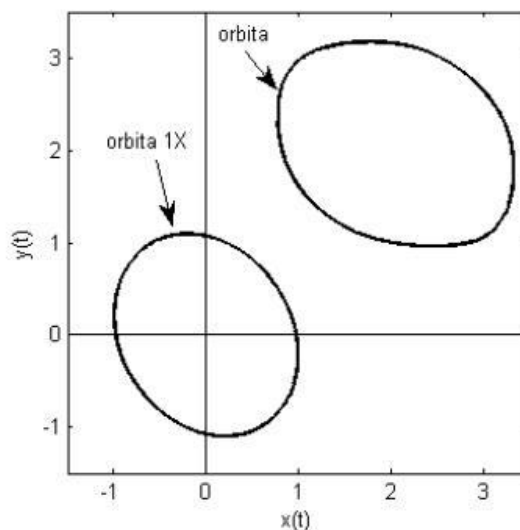
$$\begin{aligned}x_{dn}(t) &= A_x \sin(\omega t + \varphi_x) \\ y_{dn}(t) &= A_y \sin(\omega t + \varphi_y)\end{aligned}\tag{2.4}$$

Jelikož amplituda střední hodnoty signálu je oproti dynamické složce zanedbatelná, budou za průběhy výchylek středu disku uvažovány pouze dynamické složky spolu se střední hodnotou signálu.

Signál průběhu výchylky středu disku můžeme Fourierovou transformací, popsanou v publikaci [4], rozložit na frekvenční spektrum. Frekvenční složky, jejichž frekvence odpovídá celočíselnému násobku otáčkové frekvence, mají přívlastek "harmonické" (druhá harmonická - 2X, třetí harmonická 3X...). Frekvenční složky, jejichž frekvence odpovídá celočíselnému podílu otáčkové frekvence, nazýváme "subharmonické" (1/2X, 1/3X). Frekvenční složku, která odpovídá hodnotě otáčkové frekvence, nazýváme první harmonická a značíme ji 1X.

### 2.2.1. Orbity

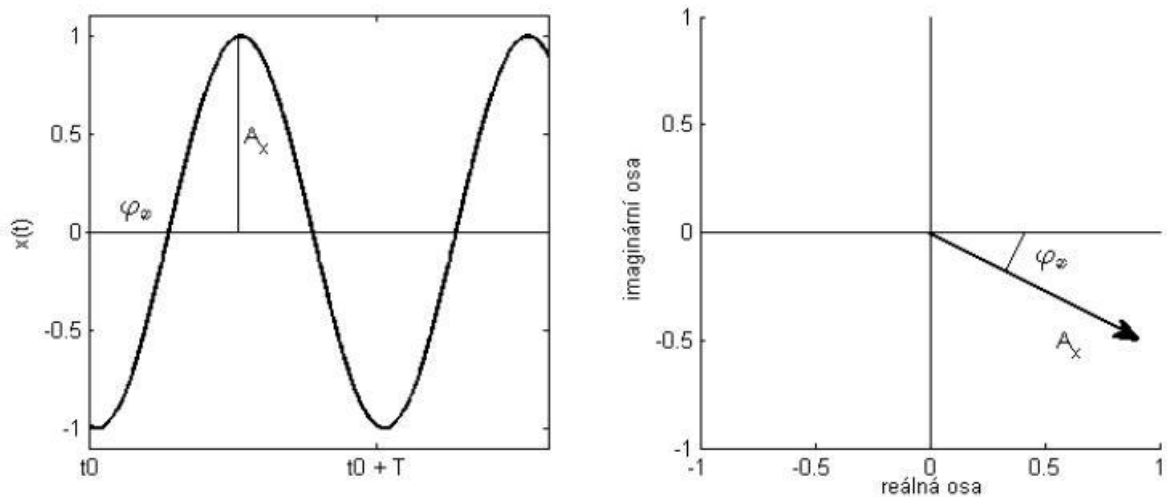
Pokud jsou změřeny signály výchylek středu disku, lze je převést do kartézských souřadnic (x, y) podle rovnice (2.2). Zavedme nyní komplexní veličinu  $z(t) = x(t) + jy(t)$ . Orbity je pak časový průmět této veličiny v rovině xy. Jinými slovy, orbity je průběh středu rotujícího disku promítnutý do roviny xy. Orbitou ovšem může být i jen některá z frekvenčních složek signálu. Taková orbity se nazývá filtrovaná [4]. Například orbity jedné frekvence - harmonické složky - je obecně elipsa.



Obrázek 2-2: Orbity a filtrovaná orbity 1X

### 2.2.2. Fázory

Fázory můžeme reprezentovat průběhy jednotlivých signálů a to pomocí zobrazení amplitudy a fáze signálů do komplexní roviny v jednotlivých časových okamžicích (viz obr. 2-3). Za předpokladu, že je signál  $x(t)$  periodický a harmonický, jej lze zapsat pomocí funkce sinus jako  $x(t) = A_x \sin(\omega t + \varphi_x)$ . Průběh fázoru při konstantní otáčkové rychlosti  $\omega$  v polárních souřadnicích můžeme psát jako  $A_x e^{j\varphi_x}$ .



Obrázek 2-3: Časový průběh harmonického signálu a jeho fázor

### 2.2.3. Precese

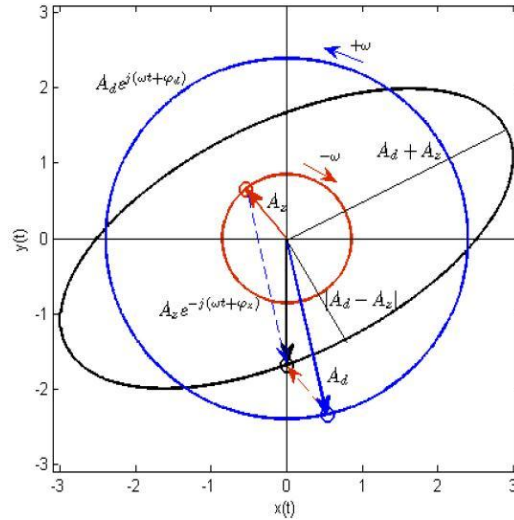
Precesí nazýváme trajektorii středu rotoru. V závislosti na tom, jakým směrem se pohybují body po této trajektorii, rozlišujeme precesi dopřednou (směr rotace rotoru a středu rotoru je shodný) a zpětnou (směr rotace rotoru a středu rotoru je opačný). Pokud se signál skládá z více frekvenčních složek, lze určit směr precese každé z těchto složek. Výsledný směr precese je pak dán směrem precese složky s dominantní amplitudou. Časovou reprezentaci precese, tj. elipsu, můžeme rozložit na dva protichůdně rotující fázory kruhových orbit. Precesi tak můžeme dostat jako vektorový součet těchto dvou fázorů. Mějme dva signály  $x(t)$  a  $y(t)$  ve tvaru (2.5)

$$\begin{aligned} x(t) &= A_x \sin(\omega t + \varphi_x) \\ y(t) &= A_y \sin(\omega t + \varphi_y) \end{aligned} \quad (2.5)$$

Orbitu  $z(t) = x(t) + jy(t)$  složenou z těchto signálů můžeme rozložit na součet dvou protichůdně rotujících fázorů [7]

$$x(t) + jy(t) = A_d e^{j(\omega t + \varphi_d)} + A_z e^{-j(\omega t + \varphi_z)} \quad (2.6)$$

Kde  $A_d$  a  $A_z$  jsou amplitudy protichůdných fázorů (viz obr. 2-4, čerpáno z [4]). Tyto hodnoty jsou pro jednotlivé orbity konstantní, neboť jejich průběhem je kružnice. Fázory jsou určeny ještě fázemi  $\varphi_d$  resp.  $\varphi_z$ .



Obrázek 2-4: Orbíta časového průběhu středu rotoru rozložena na dva protichůdné fázory

Definujme nyní komplexní tvar funkce sinus pomocí komplexní proměnné  $j$  [8]

$$\sin x = \frac{e^{jx} - e^{-jx}}{2j} \quad (2.7)$$

Rovnici orbity (2.6) můžeme upravit podle rovnic (2.5) a (2.7). Následující vztahy vychází z publikace [4]

$$\begin{aligned} & A_x \sin(\omega t + \varphi_x) + jA_y \sin(\omega t + \varphi_y) = \\ &= \frac{A_x}{2j} [e^{j(\omega t + \varphi_x)} - e^{-j(\omega t + \varphi_x)}] + \frac{A_y}{2j} [e^{j(\omega t + \varphi_y)} - e^{-j(\omega t + \varphi_y)}] = \\ &= e^{j\omega t} \left( -j \frac{A_x}{2} e^{j\varphi_x} + \frac{A_y}{2} e^{j\varphi_y} \right) + e^{-j\omega t} \left( j \frac{A_x}{2} e^{-j\varphi_x} - \frac{A_y}{2} e^{-j\varphi_y} \right) \end{aligned} \quad (2.8)$$

Pokud porovnáme členy u  $e^{j\omega t}$  a  $e^{-j\omega t}$

$$\begin{aligned} A_d e^{j\varphi_d} &= \frac{1}{2} (A_y e^{j\varphi_y} - jA_x e^{j\varphi_x}) = \\ &= \frac{1}{2} (A_y \cos \varphi_y + A_x \sin \varphi_x + jA_y \sin \varphi_y - jA_x \cos \varphi_x) \\ A_z e^{-j\varphi_z} &= \frac{1}{2} (-A_y e^{-j\varphi_y} + jA_x e^{-j\varphi_x}) = \\ &= \frac{1}{2} (-A_y \cos \varphi_y + A_x \sin \varphi_x + jA_y \sin \varphi_y + jA_x \cos \varphi_x) \end{aligned} \quad (2.9)$$

Získáme hledané vztahy k rozložení trajektorie středu rotoru na dva protichůdně rotující fázy

$$A_d = \frac{1}{2} \sqrt{A_x^2 + A_y^2 + 2A_x A_y \sin(\varphi_x - \varphi_y)}$$

$$\varphi_d = \tan^{-1} \left( \frac{A_y \sin \varphi_y - A_x \cos \varphi_x}{A_y \cos \varphi_y + A_x \sin \varphi_x} \right)$$
(2.10)

$$A_z = \frac{1}{2} \sqrt{A_x^2 + A_y^2 - 2A_x A_y \sin(\varphi_x - \varphi_y)}$$

$$\varphi_z = \tan^{-1} \left( -\frac{A_y \sin \varphi_y + A_x \cos \varphi_x}{-A_y \cos \varphi_y + A_x \sin \varphi_x} \right)$$

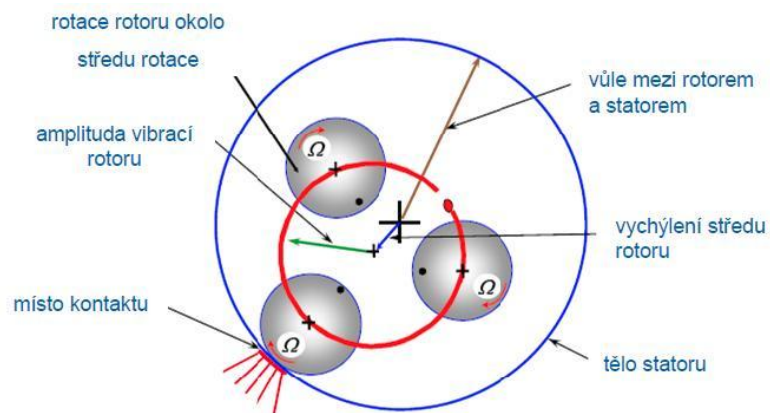
Jak bylo uvedeno výše, odvozené veličiny  $A_d$  a  $A_z$  jsou amplitudy protichůdných fázorů a jejich hodnota je konstantní. U každé orbity tedy může být určen charakter precese. Pokud je splněna první podmínka (2.11), jedná se o orbitu s dopřednou precesí. Při platnosti druhé podmínky (2.11) jde o orbitu se zpětnou precesí. Tyto vztahy vychází z [7].

$$A_d > A_z$$

$$A_d < A_z$$
(2.11)

### 2.3. Rubbing

Jak již bylo ukázáno v předchozích kapitolách, při pohybu rotoru dochází vlivem odstředivé síly spojené s nevyvážkem na hřídeli ke změně vzdálenosti mezi středem hřídele a osy jejího otáčení. V krajním případě může tato vzdálenost nabýt takové hodnoty, že dojde ke kontaktu mezi rotující částí turbíny, tj. rotorem, a její stacionární nepohyblivou částí, tj. statorem. Tento kontakt se nazývá rubbing (viz obr. 2-5).



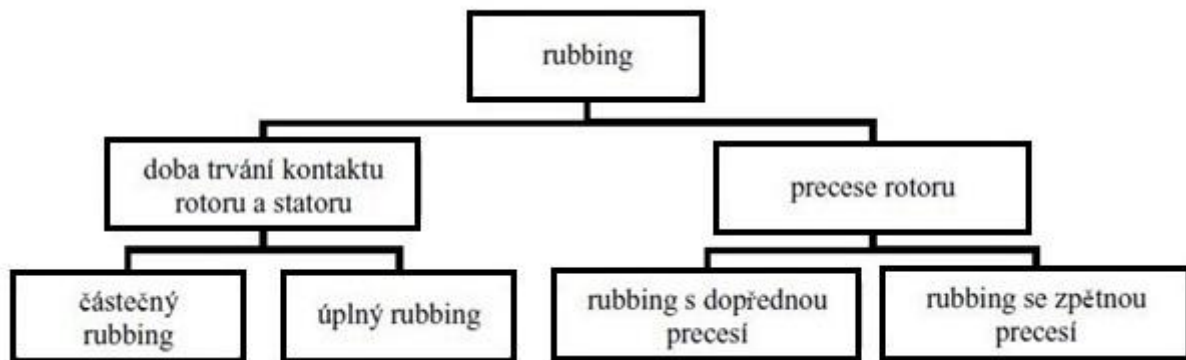
Obrázek 2-5: Rubbing [7]



Rubbing je vážná porucha zařízení, která nepříznivě ovlivňuje vlastnosti rotoru. Dochází k nárůstu vibrací, mechanický kontakt je doprovázen třením, při kterém vzniká teplo, a může tak dojít k deformaci rotoru. V případě kontaktu v oblasti ucpávky může dojít k jejímu poškození a unikající pára snižuje účinnost celé turbíny. V krajním případě může dojít až k havárii celého zařízení. Rubbing je ve většině případů pouze indikátorem některé další závady stroje.

### 2.3.1. Typy rubbingu

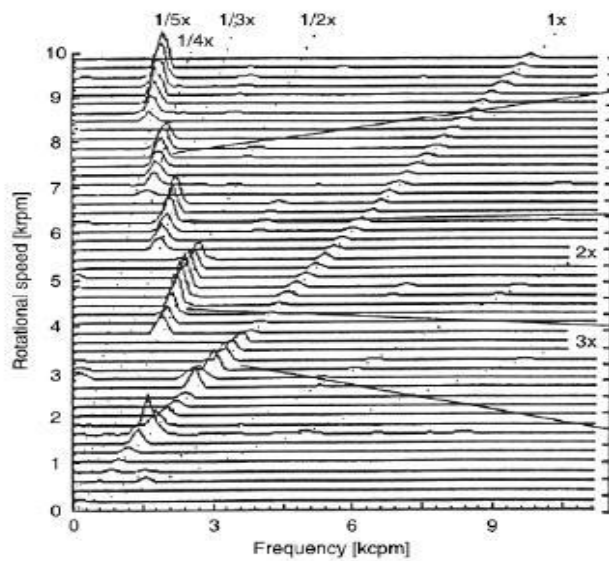
Rubbing můžeme rozdělit na několik typů (viz obr. 2-6). Především pak podle délky trvání samotného kontaktu a podle typu precese rotoru. V tomto odstavci jsem čerpal především z publikace [4] a [7].



Obrázek 2-6: Typy rubbingu [4]

#### Částečný rubbing

Pokud není rotor ve stálém kontaktu se statorem vzniká částečný rubbing. Tento typ rubbingu je typický svým obecně krátkodobým kontaktem během jedné periody otočení rotoru. Tento kontakt však může mít periodický ráz a trvat po dlouhou dobu. Podle charakteru nárazů vznikají ve spektru měřeného signálu subharmonické složky. Z těchto složek spektra můžeme určit intenzitu částečného rubbingu. Slabý částečný rubbing vzniká, když se ve spektru objevují subharmonické složky  $1/2X$  (kontakt každou druhou otáčku),  $1/3X$ ,  $1/4X$  atd. a jejich celočíselné násobky (viz obr. 2-7). Vzniká i několik vyšších harmonických složek  $2X$ ,  $3X$  atd. Silný částečný rubbing, tj. rubbing s vysokou intenzitou, můžeme ze spektra určit podle toho, že harmonické složky jsou potlačeny a oproti slabému rubbingu vzniká pouze dominantní subharmonická složka  $1/2X$  a její celočíselné násobky. Silný částečný rubbing může být doprovázen krátkodobou změnou precese rotoru. Částečný rubbing se může změnit v úplný rubbing.



Obrázek 2-7: Částečný rubbing - waterfall graf [7]

## Úplný rubbing

Úplný rubbing je charakterizovaný tím, že mezi rotorem a statorem dochází k téměř nepřerušovanému kontaktu. Je předcházen částečným rubbingem. Úplný rubbing můžeme dále rozdělit na rubbing synchronní a samobuzený.

Synchronní úplný rubbing, tj. úplný rubbing s dopřednou precesí, je charakteristický vznikem dominantní amplitudy 1X ve spektru. Amplituda 1X je omezena průměrem stacionární části systému. Vzniká zde "jump up" a "jump down" efekt dobře popsáný v [7].

Samobuzený úplný rubbing, tj. úplný rubbing se zpětnou precesí, je charakteristický vznikem nové dominantní vlastní frekvence ve spektru. Tato frekvence je vyšší než rezonanční frekvence systému bez rubbingu, což znamená, že se rotor neotáčí žádnou otáčkovou frekvencí (frekvence je dokonce téměř nezávislá na otáčkách rotoru). Tento stav a zároveň typ rubbingu je nejnebezpečnější a může vést až ke zničení stroje.

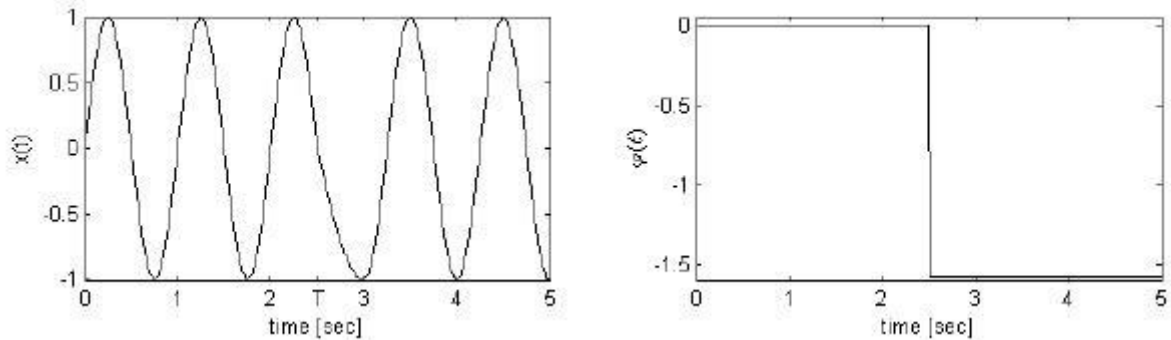
Rubbing může vznikat i při nájíždění nebo dojíždění otáček v oblasti rezonanční frekvence, tj. vlastní frekvence rotoru, která je doprovázena největšími amplitudami rotorových vibrací, při kterých je pravděpodobnost kontaktu největší.

### 2.3.2. Projevy rubbingu

Jak již bylo řečeno v předchozích kapitolách, vlivem rubbingu může být pozorováno několik charakteristických jevů, ze kterých lze právě na existenci rubbingu poukázat. Některé z nich nyní uvedeme.

Z mechanického pohledu dochází při rubbingu ke tření, které způsobuje deformaci povrchu rotoru a statoru. Ve spektru má pak za následek vznik vysokofrekvenčních komponent a šumu. Vlivem tření dochází k zahřívání místa kontaktu a tím k deformaci, která je zvláště nebezpečná u rotoru, který je citlivý na přesné rozložení hmoty. Jedním ze

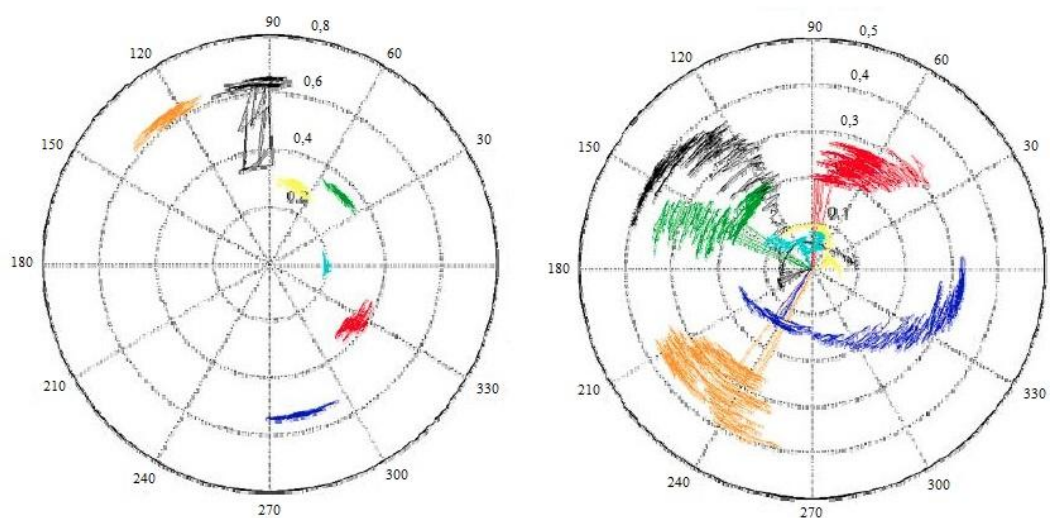
základních přístupů pro detekci rubbingu je fakt, že při výskytu rubbingu se mění otáčková frekvence rotoru (dojde k jejímu poklesu). Vliv změny otáčkové frekvence má vliv na změnu počáteční fáze měřeného signálu [4]. Vlevo na obrázku 2-8 si lze všimnout, že v čase 2,5 s došlo k rubbingu - perioda signálu se krátkodobě zvětšila, což je právě dáno snížením otáčkové frekvence. Vpravo pak lze vidět doprovodný jev této události, tj. změnu počáteční fáze.



**Obrázek 2-8: Vliv rubbingu na otáčkovou rychlost a počáteční fázi**

Ve skutečnosti dojde kromě změny počáteční fáze i ke změně amplitudy [7]. Tento jev je ilustrován na obrázku 2-9 [9]. V levé části je průběh fází jednotlivých signálů v případě, že rubbing nenastal. V pravé části je situace, kdy rubbing nastal. Porovnáním si lze všimnout, že se fáze signálů oproti situaci bez rubbingu výrazně změnila. Změnila se také amplituda, která s rostoucím kontaktem klesá.

Při rubbingu dochází kromě tření také k nárazům rotoru na stator doprovázeným širokým spektrem vybuzených frekvencí. Vlivem rubbingu dochází také ke změně tuhosti systému [7].



**Obrázek 2-9: Vliv rubbingu na počáteční fázi a amplitudu – fázorový diagram**

### 3. STÁVAJÍCÍ STAV SYSTÉMU

Stávající systém monitorování rubbingu, kterým se budu v tomto odstavci zabývat, je fyzicky implementován na jedné z turbín tepelné elektrárny Tušimice. Při psaní této části práce jsem vycházel zejména ze zdrojů [4], [10] a [11].

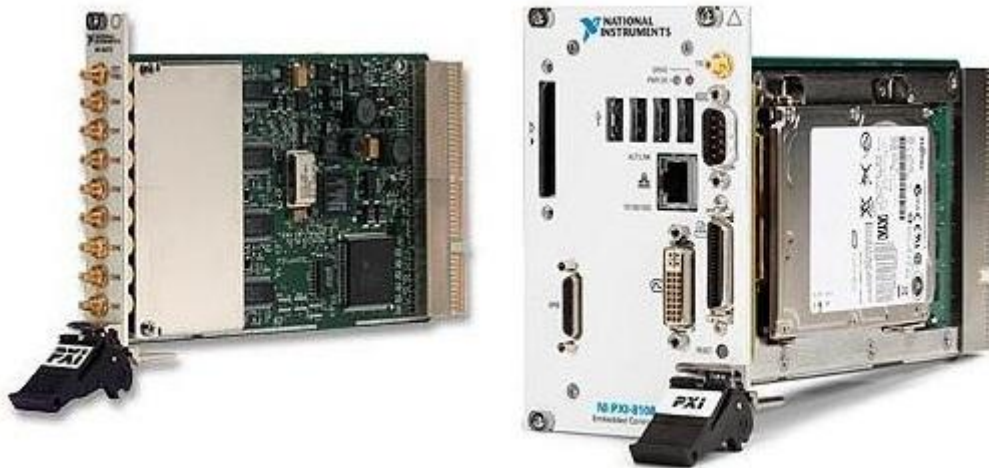
#### 3.1. Hardware

Hardware měřicího systému je realizován produkty firmy National Instruments (viz kapitola 3.1.). Na obrázku 3-1 je kostra výrobku NI PXI-1031, která obsahuje 4 sloty pro měřicí zařízení. Obsahuje také vestavěný zdroj napětí, který může být napájen přímo ze sítě napětím 230 V při frekvenci 50 Hz. Součástí je také velmi účinné chlazení.



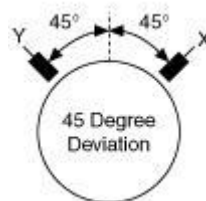
Obrázek 3-1: NI PXI-1031

Vpravo na obrázku 3-2 je výpočetní jednotka systému, která je reprezentována výrobním označením NI PXI-8108. Tato jednotka s 2,53 GHz procesorem Core 2 Duo T9400 od firmy Intel je kontrolerem, který pracuje pod operačním systémem Windows 7. V tomto OS běží software LabView (viz kapitola 3.2.), ve kterém je celý systém monitorování rubbingu programově implementován. Tento kontroler má standardně 1 GB 800 MHz operační paměť DDR2, pevný disk s kapacitou 80 GB se sběrnicí SATA, 6 MB cache, 4 USB konektory, sériovou linku RS 232, ethernetový port, paralelní port IEEE 1284 a další vstupně výstupní porty.



Obrázek 3-2: NI PXI-1031 a NI PXI-8108

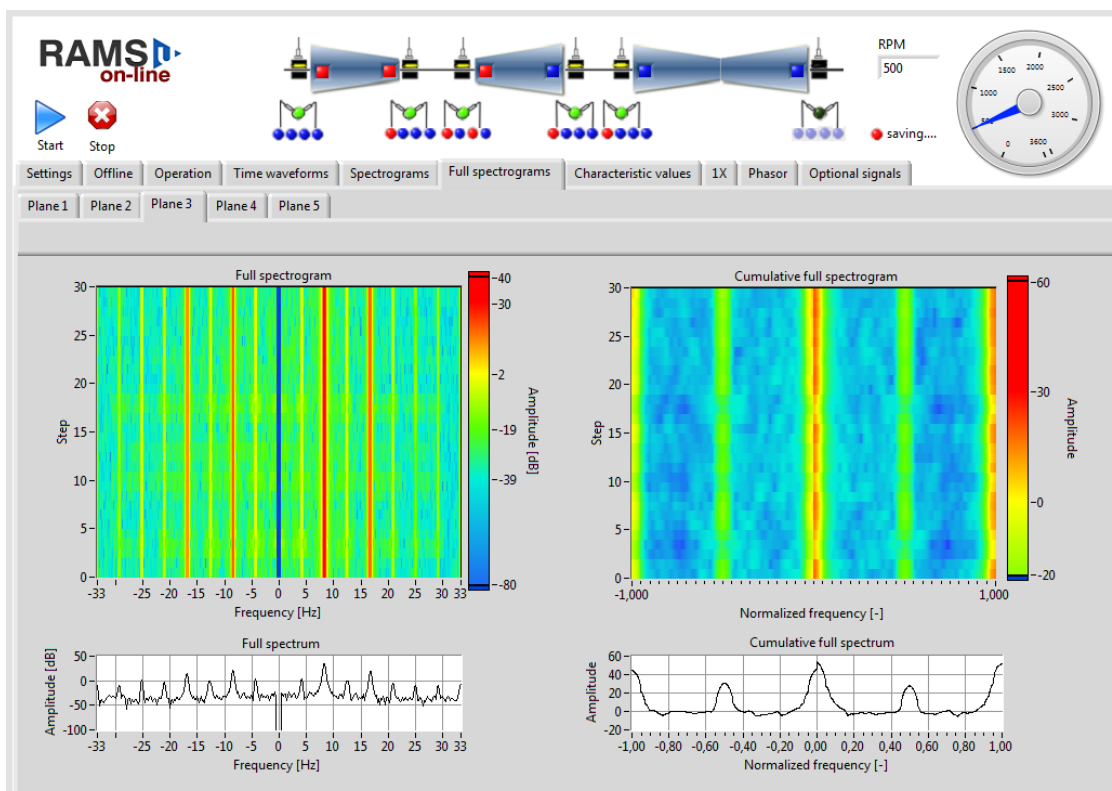
Hodnoty měřených veličin jsou signály vedeny a zpracovávány třemi kartami NI PXI-4472B (viz obrázek 3-2 vlevo). Každá karta obsahuje 8 24-bitových analogových kanálů. Každý kanál disponuje možností maximální vzorkovací frekvence 102400 vzorků za sekundu a rozsahu -10 až 10 V, přičemž je možné využít jednotlivé kanály pro měření napětí, nebo pro akcelerometrická měření. Dynamický rozsah je 111 dB. V našem případě využíváme 19 kanálů, neboť jsou měřeny signály ze 6 rovin turbíny a pro každou rovinu jsou využity 3 kanály – jeden pro měření absolutní statorové vzdálenosti a dva kanály zpracovávají signály z čidel měřících relativní rotorovou vzdálenost od turbíny. Poslední z kanálů je použit pro měření otáček rotoru – otáčková synchronizace měření. Umístění těchto čidel je ilustrováno na obrázku 3-3.



Obrázek 3-3: Umístění čidel relativní rotorové vzdálenosti [10]

### 3.2. SW pro online detekci rubbingu

Software systému pojmenovaný RAMS (Rub Advanced Monitoring System) Online (viz obr. 3-4) je naprogramován v prostředí LabView (viz kapitola 3.3.). V tomto softwaru jsou zpracovávány signály z karty provádějící měření na turbíně. Samotný systém pracuje na bázi metody úplného spektra, což znamená, že měření a následná diagnostika se provádí pomocí dvou signálů na sebe navzájem kolmých (viz obr. 3-3). Software stávajícího systému zaznamenává online data, tj. data aktuálně měřená na turbíně a následně je vyhodnocuje a zobrazuje průběhy na jednotlivých rovinách, spektrogramy, fázory, otáčkovou rychlost a jiné uživatelem žádané hodnoty.

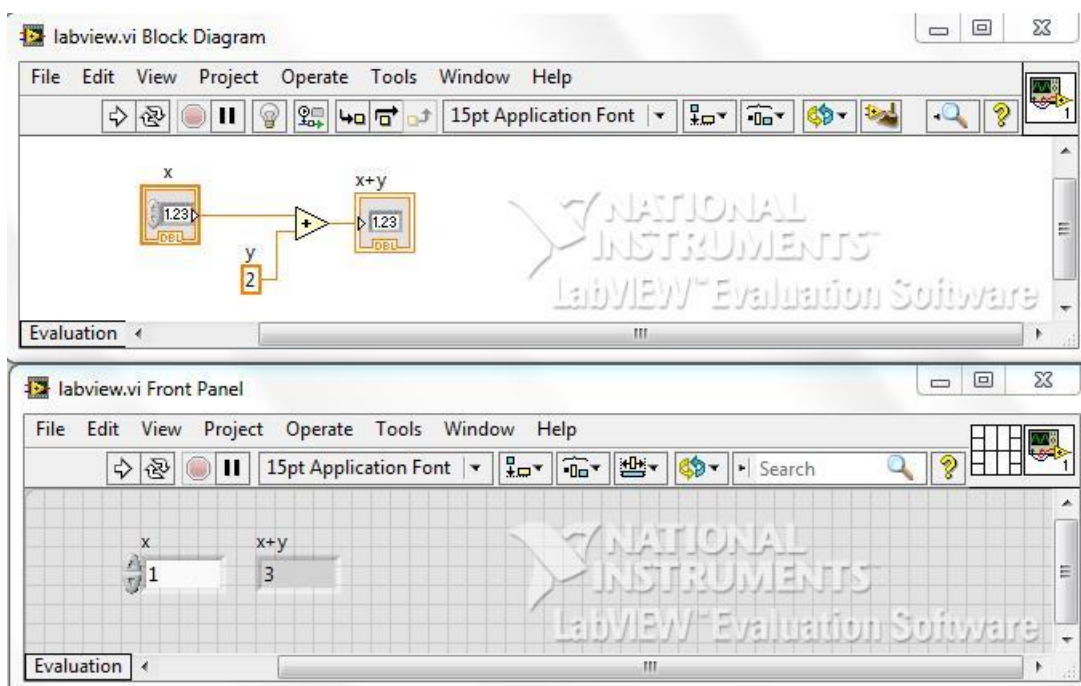


Obrázek 3-4: RAMS Online

### 3.3. LabView

LabView je nástroj od firmy National Instruments založený na grafickém programování. Každý program je tvořen z bloků, z nichž každý má svou konkrétní specifikaci, tj. vnitřní uspořádání a kombinaci vstupů a výstupů. Propojení jednotlivých bloků zajišťuje správný směr informačního toku a tedy chod programu. Tento software je vhodný k použití v aplikacích reálných systémů, neboť komunikuje s celou řadou měřicích a regulačních karet a systémů. Implementace složitých programů je mnohdy snadnější než pomocí jiných nástrojů nevyužívajících grafické programování. To jsou jedny z hlavních důvodů, proč je náš systém monitorování a detekce rubbingu implementován právě v tomto prostředí. Archivace diagnostických veličin a nástroj pro vizualizaci, což jsou hlavní cíle této práce, jsou implementovány také pomocí programu LabView.

Samotné programovací prostředí je rozděleno na dvě části - blokové schéma a uživatelské rozhraní – front panel (viz obr 3-5). Blokové schéma je samotný program, který je tvořen z bloků vzájemně propojených informačními vazbami. Uživatelské rozhraní je ta část, která zajišťuje výměnu informací mezi uživatelem a programem.



Obrázek 3-5: Prostředí LabView

### 3.4. Výstup stávajícího systému

Stávající systém archivuje požadované výstupní veličiny do textového souboru. Po bližším seznámení se s tímto systémem jsem zjistil, že výstupní tok dat, které se archivují, je ve formátu dvou textových souborů .txt, kde jsou všechny určující hodnoty veličin zapsány s přesností na šest desetinných míst. Jednotlivé bloky v řádku souboru jsou odděleny tabulátorem.

První z těchto souborů pojmenovaný trendy.txt (viz obr. 3-7) zaznamenává jednotlivé trendy měřených veličin. Každý řádek tohoto souboru obsahuje hodnoty archivovaných veličin dané roviny a je v následujícím formátu:

- Čas události – Klíčový údaj každého řádku.
- Jméno roviny – Číslo roviny, ke kterému se údaje vztahují.
- Amplitudy 1X
  - Amplituda 1X složky spektra signálu měřeného ve směru  $x$  –  $A_x$ .
  - Amplituda 1X složky spektra signálu měřeného ve směru  $y$  –  $A_y$ .
  - Amplituda fázoru dopředné precese 1X složky spektra –  $A_d$  (viz obrázek 2-4).
  - Amplituda fázoru zpětné precese 1X složky spektra –  $A_z$  (viz obrázek 2-4).
  - Počáteční fáze 1X složky signálu měřeného ve směru  $x$  –  $\varphi_x$  [rad].
  - Počáteční fáze 1X složky signálu měřeného ve směru  $y$  –  $\varphi_y$  [rad].

(Hodnoty obou počátečních fází obsahuje pouze novější verze výstupního souboru stávajícího systému.)

- Charakteristické hodnoty  $1/2X$ 
  - Amplituda  $1/2X$  složky spektra signálu měřeného ve směru x.
  - Amplituda  $1/2X$  složky spektra signálu měřeného ve směru y.
- Charakteristické hodnoty  $1/3X$ 
  - Amplituda  $1/3X$  složky spektra signálu měřeného ve směru x.
  - Amplituda  $1/3X$  složky spektra signálu měřeného ve směru y.
- Charakteristické hodnoty  $1/4X$ 
  - Amplituda  $1/4X$  složky spektra signálu měřeného ve směru x.
  - Amplituda  $1/4X$  složky spektra signálu měřeného ve směru y.
- Charakteristické hodnoty  $1/5X$ 
  - Amplituda  $1/5X$  složky spektra signálu měřeného ve směru x.
  - Amplituda  $1/5X$  složky spektra signálu měřeného ve směru y.
- Otáčková frekvence – Úhlová rychlost rotoru [Hz].
- $S_{max}$  – Největší hodnota výchylky středu rotoru od osy otáčení při změně úhlu rotoru o hodnotu  $2\pi$  rad.  $S_{max} = A_d + A_z$  (viz obr. 2-4).
- Flag – Binární hodnota nesoucí uživatelskou informaci

Měření signálů ve směru x a y bylo realizováno snímači umístěnými podle obrázku 3-3, což znamená, že osa x a y není ve vodorovném resp. svislém směru, ale obě jsou pootočený o  $45^\circ$ .

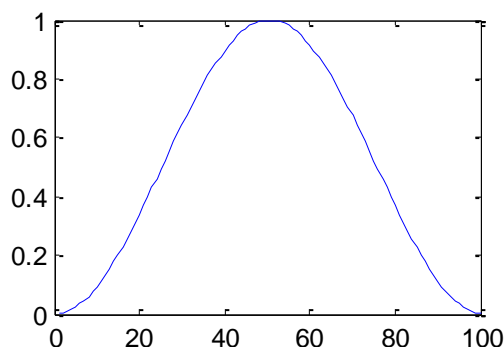
22.10.2012 9:09:27	rovina 1	amplitudy 1x:	1,393471	1,912150
22.10.2012 9:09:27	rovina 2	amplitudy 1x:	4,531157	6,008423
22.10.2012 9:09:27	rovina 3	amplitudy 1x:	8,998925	7,215060
22.10.2012 9:09:27	rovina 4	amplitudy 1x:	10,522817	9,189394
22.10.2012 9:09:27	rovina 5	amplitudy 1x:	0,809142	2,074504
22.10.2012 9:15:27	rovina 1	amplitudy 1x:	1,396247	1,910469
22.10.2012 9:15:27	rovina 2	amplitudy 1x:	4,529375	6,004702
22.10.2012 9:15:27	rovina 3	amplitudy 1x:	9,002298	7,213086
22.10.2012 9:15:27	rovina 4	amplitudy 1x:	10,528340	9,186927
22.10.2012 9:15:27	rovina 5	amplitudy 1x:	0,809296	2,075780
22.10.2012 9:21:27	rovina 1	amplitudý 1x:	1,397000	1,915867

Obrázek 3-6: Výřez stávajícího výstupního souboru trendy.txt

Druhý textový soubor pojmenovaný log2.txt (případně log.txt) obsahuje informace o změnách v nastavení systému, systémových parametrech (viz obr. 3-8) a událostech - zda-li byl systém RAMS Online spuštěn a s jakými parametry, nebo vypnut, zda-li byl změněn některý z parametrů, případně informace o detekci rubbingu, hodnoty  $S_{max}$ , či jejich ukončení. Oproti souboru trendy.txt nemá soubor log2.txt (případně log.txt) jednotný formát a jsou do něho pouze zapisovány logované události v pořadí, v jakém v systému nastaly. Existují tři formáty daného souboru – česká verze, anglická verze a česká verze bez diakritiky. Uvedme některé obsažené veličiny (parametry), při jejichž popisu jsem vycházel především ze zdroje [7]:

- Vzorkovací frekvence – Frekvence s jakou jsou snímány hodnoty signálu [Hz].
- Typ okénkové funkce – Rozeznáváme několik typů okénkových funkcí [7]. Nejběžněji používanou funkcí je Hanningovo okénko (viz obr. 3-7). Okénková funkce zajišťuje, aby nedošlo k úniku ve spektru tzn. zajišťuje periodicitu, nutnou pro správné určení spektra.





**Obrázek 3-7: Hanningova okénková funkce**

- Délka okénkové funkce – Velikost okénkové funkce [s].
- Překryv okének – Udává jak velká část signálu [%] předchozího okénka se bude uvažovat při dalším výpočtu nového spektra např. při krátkodobé Fourierovo transformaci. Překryv okének zlepšuje rozlišení v čase.
- Maximální frekvence spektra – [Hz].
- Délka signálu před rubbingem – Udává časovou velikost signálu v bufferu, která se při detekci rubbingu zaznamenává a archivuje pro následnou analýzu.
- Úroveň signálu pro výpočet otáček – Hodnota, kterou když signál sestupným směrem překročí, vyvolá impuls. Obrácená hodnota časové vzdálenosti dvou sousedících impulsů je otáčková frekvence.
- Prahy detekce – Hodnoty udávající hladiny, které při překročení signálu pro dané harmonické, subharmonické složky spektra nebo hodnotu Smax, vyvolají detekci rubbingu.
- Šířka pásma pro normování – Velikost pásma určeného pro normování kumulativního spektra pro odstranění šumu.

```

22.10.2012 9:00:33      aplikace sputna
Parametry -          vzorkovac frekvence [Hz]:      4096,000000      vzorkovac frekv
22.10.2012 9:00:48      aplikace sputna
22.10.2012 9:01:57      aplikace sputna
Parametry -          vzorkovac frekvence [Hz]:      4096,000000      vzorkovac frekv
22.10.2012 9:02:04      zmna hodnoty parametru
22.10.2012 9:03:19      aplikace sputna
Parametry -          vzorkovac frekvence [Hz]:      4096,000000      vzorkovac frekv
23.10.2012 9:27:58      Smax              rovina 1          hladina
23.10.2012 9:27:58      Smax              rovina 5          hladina
23.10.2012 9:27:58      Smax              rovina 4          hladina
23.10.2012 9:27:58      Smax              rovina 3          hladina
23.10.2012 9:27:58      Smax              rovina 2          hladina
23.10.2012 9:46:10      rubbing           rovina 2- 1/2X
23.10.2012 9:46:10      rubbing           rovina 4- 1/2X

```

**Obrázek 3-8: Výřez stávajícího výstupního souboru log2.txt**

Jedním z úkolů bylo vylepšit archivaci dat stávajícího systému, a tím přejít z ukládání a uchovávání dat ve dvou textových logsouborech k efektivnějšímu způsobu archivace.



## 4.2. Microsoft Office Access

Z dalších nástrojů vhodných pro archivaci dat uveďme Microsoft Office Access. Tento relační databázový systém od společnosti Microsoft kombinuje relační Microsoft Jet Database Engine s uživatelským rozhraním a nástroji pro vývoj softwaru. Microsoft Access ukládá data ve svém vlastním formátu založeném na Access Database Jet. Všechny databázové tabulky, dotazy, formuláře, sestavy, makra a moduly jsou uloženy v databázi aplikace Access Jet jako jeden soubor. Microsoft Access je součástí sady Microsoft Office a je určený především pro výuku a pochopení relačních systémů nebo pro databázové systémy menších rozměrů, proto jsem při volbě vhodného nástroje pro archivaci dat a správu databází upřednostnil Microsoft SQL Server, který je vhodnější pro velmi objemná data, která vzhledem k periodickému snímání veličin diagnostickým systémem rubbingu můžeme očekávat.

## 4.3. SQL

Jak již název napovídá SQL (Structured Query Language) je standardizovaný strukturovaný dotazovací jazyk používaný pro přístup k databázím. Můžeme jej rozdělit na dvě části. Část pro práci s databázovými strukturami DDL (The Data Definition Language) a část pro práci s daty DML (The Data Manipulation Language). Uveďme pro představu syntaxi několika základních příkazů pro práci s databázovými strukturami:

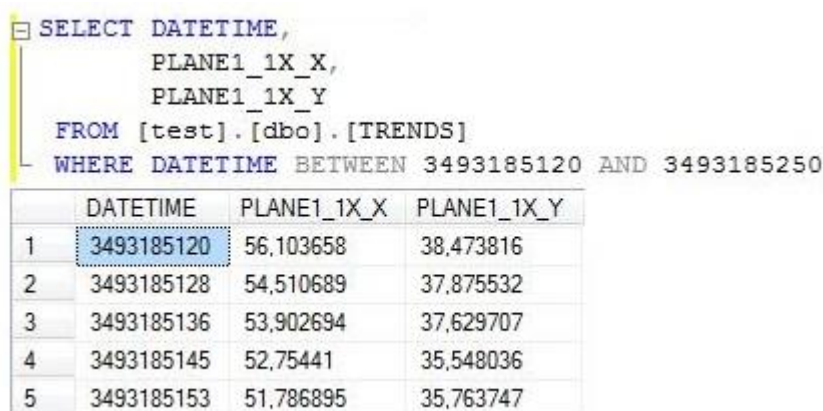
- `CREATE DATABASE database_name`  
Tento příkaz vytváří novou databázi.
- `CREATE TABLE table_name (column_name1 data_type, column_name2 data_type, column_name3 data_type,...).`  
Příkaz `CREATE TABLE` vytváří tabulku databáze se zadanými parametry.
- `DROP TABLE table_name`  
Tento příkaz vymaže zvolenou tabulku v databázi.

A několik příkazů pro práci se samotnými daty:

- `SELECT column_name(s) FROM table_name WHERE column_name operator value`  
Příkaz `SELECT` slouží pro výběr dat z databáze. Vybere informace ze sloupců zadané tabulky. `Column_name(s)` může být nahrazeno znakem `*` značícím, že mají být vybrána data ze všech sloupců. Příkaz `WHERE` specifikuje výběr dat zadanou podmínkou.
- `INSERT INTO table_name (column1, column2, column3,...) VALUES (value1, value2, value3,...)`  
Příkaz `INSERT` zajišťuje vkládání nových dat do databáze.

- UPDATE table\_name SET column1 = value1, column2 = value2,...WHERE some\_column = some\_value  
Příkaz UPDATE se používá k aktualizaci již stávajících záznamů v databázi.
- DELETE FROM table\_name WHERE some\_column = some\_value  
Příkaz DELETE vymaže vybraný záznam z tabulky databáze.

Jazyk SQL není case sensitive, což znamená, že není závislý na velikosti písma. Výše uvedené příklady příkazů je tedy možné psát i malými písmeny. Na obrázku 4-2 je znázorněno použití příkazu SELECT v prostředí SQL Server Management Studia. V horní části obrázku je samotný SQL příkaz, kde vybíráme hodnoty ze sloupců PLANE1\_1X\_X a PLANE1\_1X\_Y tabulky TRENDS, které jsou mezi uvedenými hodnotami. Tato tabulka je objektem databáze pojmenované test. Za rezervovaným slovem FROM jazyka SQL je tedy celá cesta k dané tabulce. V dolní části obrázku je výsledný výstup SQL Server Management Studia na uvedený příkaz SELECT v závislosti na datech v tabulce TRENDS.



```

SELECT DATETIME,
       PLANE1_1X_X,
       PLANE1_1X_Y
FROM [test].[dbo].[TRENDS]
WHERE DATETIME BETWEEN 3493185120 AND 3493185250

```

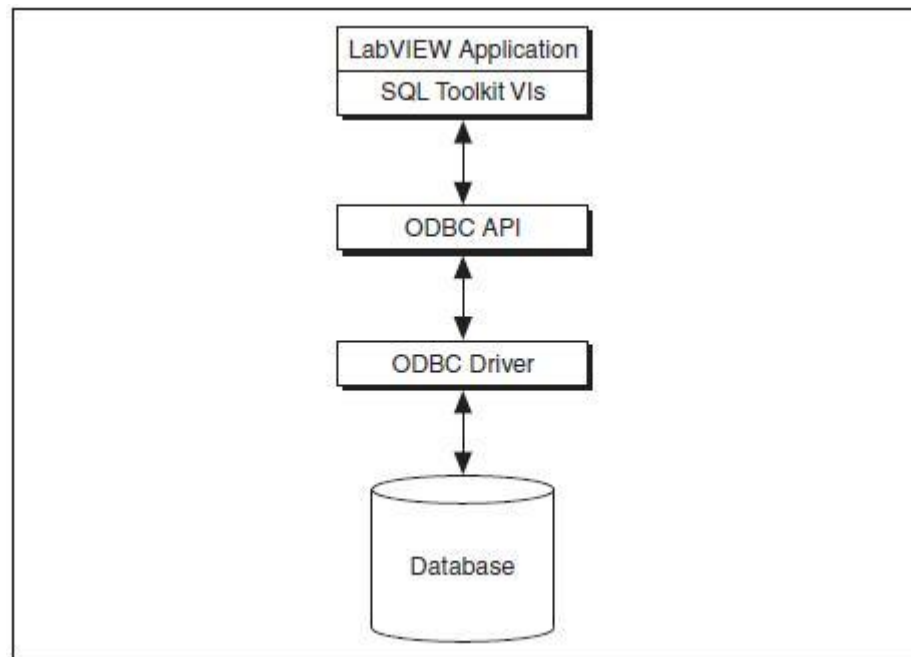
	DATETIME	PLANE1_1X_X	PLANE1_1X_Y
1	3493185120	56,103658	38,473816
2	3493185128	54,510689	37,875532
3	3493185136	53,902694	37,629707
4	3493185145	52,75441	35,548036
5	3493185153	51,786895	35,763747

Obrázek 4-2: Použití SQL příkazu SELECT v prostředí SQL Server Management Studia

## 4.4. ODBC

ODBC (open database connectivity) byl vytvořen jako standart jednotné metody pro přístup aplikací k databázím. Tato norma se skládá z víceúrovňového API (Application Programming Interface), ovladače, SQL implementace založené na ANSI SQL a prostředkem pro definování a udržování jmen zdrojových dat DSN (data source name). DSN je rychlý způsob, jak se odkazovat na konkrétní databázi. Pokud nadefinujeme DSN jedinečným názvem, ovladač ODBC nás může spojit s danou místní nebo vzdálenou databází. Každá databáze, ke které chceme tímto způsobem přistupovat musí mít nadefinovanou svojí DSN.

Součástí LabView je SQL Toolkit VIs, který podporuje ODBC a pro komunikaci s databází můžeme použít právě tento způsob. Na obrázku 4-3 je tento způsob komunikace mezi LabView a databází ilustrován. SQL Toolkit VIs volá Microsoft API pro použití ODBC. Ovladač ODBC následně komunikuje s databází specifickým ovladačem, který překládá volání do nízkourovňového jazyka databáze.



Obrázek 4-3: Komunikace mezi LabView a databází [13]

## 5. IMPLEMENTACE SYSTÉMU ARCHIVACE DAT

Nový systém archivace dat je implementován stejně jako stávající systém v prostředí LabView (viz kapitola 3.1.). Pro propojení obou systémů jsem použil algoritmy, při jejichž implementaci jsem využil informací především ze zdrojů [10], [13] a [14]. V posledních dvou zdrojích je popsáno použití knihovny programu LabView a to konkrétně LabView Connectivity Toolkit, který hraje důležitou roli v práci s databázemi.

### 5.1. Formát tabulek databáze

Po seznámení se se stávajícím výstupním systémem v odstavci 3.2., kde výstupní veličiny také popisují, jsem mohl přistoupit k fázi návrhu nového formátu archivovaných dat. Informace o stavech systému, které se dosud ukládaly do dvou textových souborů, jsem se rozhodl ukládat do tří tabulek databáze uložené v Microsoft SQL Server (viz kapitola 4.1). Tabulky jsem navrhl v následujícím formátu, kde v prvním sloupci je vždy název veličiny a její datový typ a ve druhém sloupci její krátký popis vycházející z veličin stávajícího systému popsaných v kapitole 3.3:

tabulka SETTINGS	
jméno sloupce (datový typ)	popis
DATETIME (DBL)	klíčový sloupec udávající čas v sekundách od 1.1.1904 12:00
ACTION (I32)	0 - aplikace spuštěna, 1 - ukončena, 2 - změna parametrů
parametry pro zpracování signálu	
FS (I32)	vzorkovací frekvence
WINLEN (I32)	délka okénkové funkce
WINTYPE (I32)	typ okénkové funkce (0 - Hanning)
OVERLAP (I32)	překryv okének
FMAX (I32)	maximální frekvence spektra
PRETRIGGER (I32)	délka záznamu před událostí
KP_THRESHOLD (DBL)	úroveň signálu kp pro výpočet otáček
BANDWIDTH (DBL)	šířka pásma pro normování
C_POINTS (I32)	počet bodů kumulativního úplného spektra
C_INTERVALS (I32)	počet intervalů pro výpočet kumulativního úplného spektra
parametry pro detekci rubbingu	
SMAX_THRESHOLD (DBL)	práh detekce Smax
THRESHOLD1 (DBL)	práh detekce 1/2X
THRESHOLD2 (DBL)	práh detekce 1/3X
THRESHOLD3 (DBL)	práh detekce 1/4X
THRESHOLD4 (DBL)	práh detekce 1/5X

**Tabulka 5-1: Formát databázové tabulky SETTINGS**

tabulka EVENT	
jméno sloupce (datový typ)	popis
DATETIME (DBL)	klíčový sloupec udávající čas v sekundách od 1.1.1904 12:00
sloupce pro data naměřená v rovině 1	
SMAX1 (I32)	1 - Smax, 2 - Smax ukončeno, 3 - Smax + Smax ukončeno ve stejný čas
THRESHOLD1_S (DBL)	hodnota překročeného thresholdu Smax
VALUE1_S (DBL)	hodnota Smax
RUBBING1_1/2X (I32)	1 - rubbing, 2 - rubbing ukončeno, 3 - rubbing + rubbing ukončeno ve stejný čas pro char. veličinu 1/2X
THRESHOLD1_1/2X (DBL)	hodnota překročeného thresholdu char. veličinu 1/2X
VALUE1_1/2X_F (DBL)	hodnota charakteristické veličiny 1/2X dopředné precese
VALUE1_1/2X_B (DBL)	hodnota charakteristické veličiny 1/2X zpětné precese
RUBBING1_1/3X (I32)	1 - rubbing, 2 - rubbing ukončeno, 3 - rubbing + rubbing ukončeno ve stejný čas pro char. veličinu 1/3X
THRESHOLD1_1/3X (DBL)	hodnota překročeného thresholdu char. veličinu 1/3X
VALUE1_1/3X_F (DBL)	hodnota charakteristické veličiny 1/3X dopředné precese
VALUE1_1/3X_B (DBL)	hodnota charakteristické veličiny 1/3X zpětné precese
RUBBING1_1/4X (I32)	1 - rubbing, 2 - rubbing ukončeno, 3 - rubbing + rubbing ukončeno ve stejný čas pro char. veličinu 1/4X
THRESHOLD1_1/4X (DBL)	hodnota překročeného thresholdu char. veličinu 1/4X
VALUE1_1/4X_F (DBL)	hodnota charakteristické veličiny 1/4X dopředné precese
VALUE1_1/4X_B (DBL)	hodnota charakteristické veličiny 1/4X zpětné precese
RUBBING1_1/5X (I32)	1 - rubbing, 2 - rubbing ukončeno, 3 - rubbing + rubbing ukončeno ve stejný čas pro char. veličinu 1/5X
THRESHOLD1_1/5X (DBL)	hodnota překročeného thresholdu char. veličinu 1/5X
VALUE1_1/5X_F (DBL)	hodnota charakteristické veličiny 1/5X dopředné precese
VALUE1_1/5X_B (DBL)	hodnota charakteristické veličiny 1/5X zpětné precese

**Tabulka 5-2: Formát databázové tabulky EVENT pro rovinu 1**

Celá tabulka EVENT obsahuje 6 rovin tzn. uvedený formát pro rovinu 1 je kopírován pro každou rovinu do jedné velké tabulky obsahující celkově 115 sloupců.

tabulka TRENDS	
jméno sloupce (datový typ)	popis
DATETIME (DBL)	klíčový sloupec udávající čas v sekundách od 1.1.1904 12:00
FROT (DBL)	hodnota otáčkové frekvence
RD_OPERATION (I32)	0 - provoz na natáčedle, 1 - náběh, 2 - doběh, 3 - nominální provoz
sloupce pro data naměřená v rovině 1	
PLANE1_SMAX (DBL)	hodnota Smax
PLANE1_1X_X (DBL)	signál x - amplituda 1X

PLANE1_1X_Y (DBL)	signál y - amplituda 1X
PLANE1_1X_F (DBL)	amplituda dopředné precese 1X
PLANE1_1X_B (DBL)	amplituda zpětné precese 1X
PLANE1_1X_PHASE_X (DBL)	signál x - fáze 1X
PLANE1_1X_PHASE_Y (DBL)	signál y - fáze 1X
PLANE1_1/2X_F (DBL)	hodnota char. veličiny 1/2X dopředné precese
PLANE1_1/2X_B (DBL)	hodnota char. veličiny 1/2X zpětné precese
PLANE1_1/3X_F (DBL)	hodnota char. veličiny 1/3X dopředné precese
PLANE1_1/3X_B (DBL)	hodnota char. veličiny 1/3X zpětné precese
PLANE1_1/4X_F (DBL)	hodnota char. veličiny 1/4X dopředné precese
PLANE1_1/4X_B (DBL)	hodnota char. veličiny 1/4X zpětné precese
PLANE1_1/5X_F (DBL)	hodnota char. veličiny 1/5X dopředné precese
PLANE1_1/5X_B (DBL)	hodnota char. veličiny 1/5X zpětné precese
PLANE1_FSPEC (Binary)	úplné spektrum (100 hodnot-decimace)

**Tabulka 5-3: Formát databázové tabulky TRENDS pro rovinu 1**

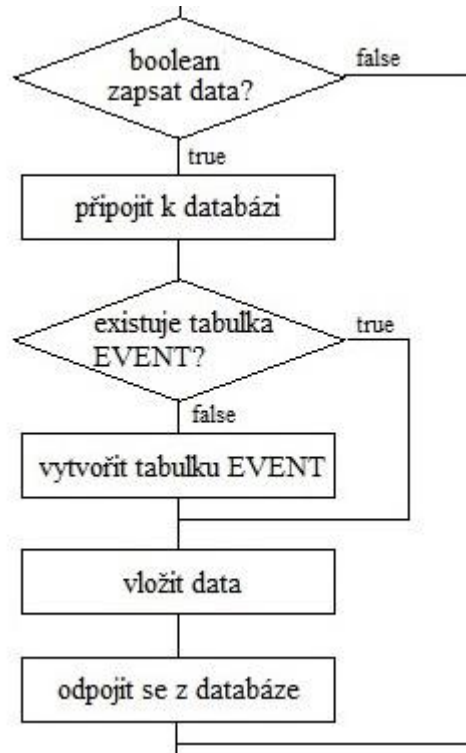
Celá tabulka TRENDS obsahuje 6 rovin tzn. uvedený formát pro rovinu 1 (PLANE1) je kopírován pro každou rovinu do jedné velké tabulky obsahující celkově 99 sloupců.

## 5.2. Zápis do databáze - online data

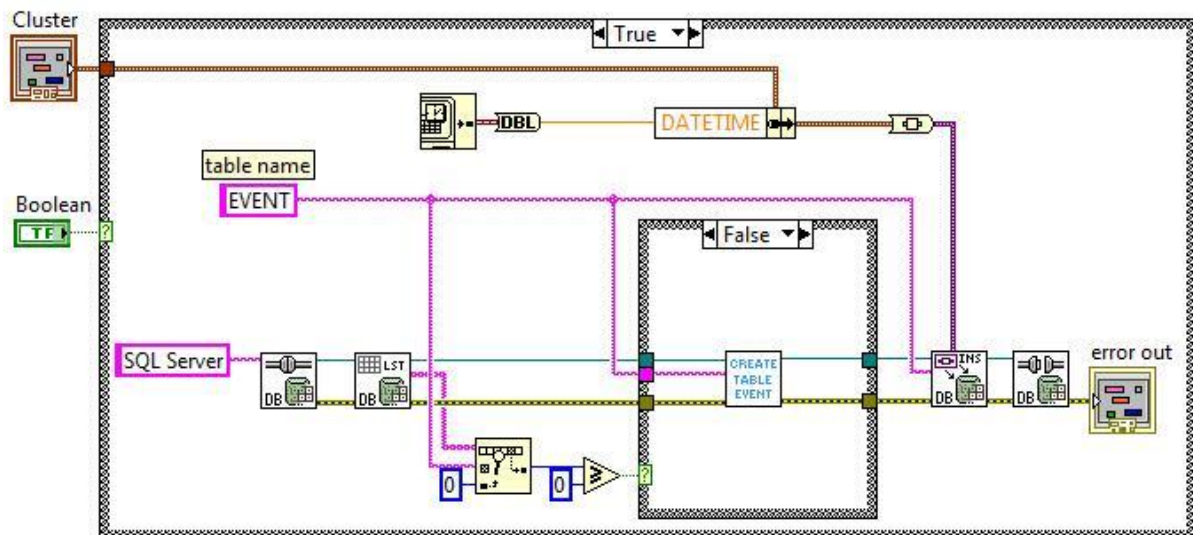
Následující algoritmy zajišťují propojení systému rubbingu konkrétně RAMS Online a databázové struktury, při kterém dochází k ukládání aktuálně měřených dat (dále jen online data). Obecně můžeme říct, že pro tabulky EVENT, TRENDS i SETTINGS je použit podobný algoritmus. Hlavní odlišnost je v deklaraci proměnných při vytváření tabulky. Pro tabulku TRENDS se proměnné deklarují pomocí tabulky 5-3, pro tabulku EVENT podle tabulky 5-2 a pro tabulku SETTINGS podle tabulky 5-1.

Na obrázku 5-1 a 5-2 je uveden vývojový diagram algoritmu resp. samotná implementace algoritmu zápisu dat do tabulky EVENT v prostředí LabView. Vstupní signály systému jsou Cluster, který obsahuje jednotlivé hodnoty měřených veličin (viz tabulka 5-2 v kapitole 5.1.) a signál typu boolean, který nese informaci o tom, zda-li se data do tabulky mají nebo nemají zapsat. V případě, že je boolean nastaven na hodnotu logické jedničky, tak se program připojí k databázi. Algoritmus dále zkontroluje, zda-li už tabulka EVENT v databázi existuje. Pokud tomu tak není, tak tabulku vytvoří a následně do ní vloží jednotlivé naměřené hodnoty z měřícího systému rubbingu s klíčovou časovou informací. Po úspěšném vložení dat do tabulky se program od databáze odpojí, případně vypíše chybu, která při běhu nastala, což je také jediný výstupní signál systému zápisu dat. V tomto případě se předpokládá, že se program připojí k databázi konfigurované ve Windows pomocí ODBC (viz kapitola 4.4), neboť tento program pro online archivaci dat poběží přímo v elektrárně, kde bude společně s programem Microsoft SQL Server nainstalován a je tedy žádoucí, aby se připojoval k jedné databázi.



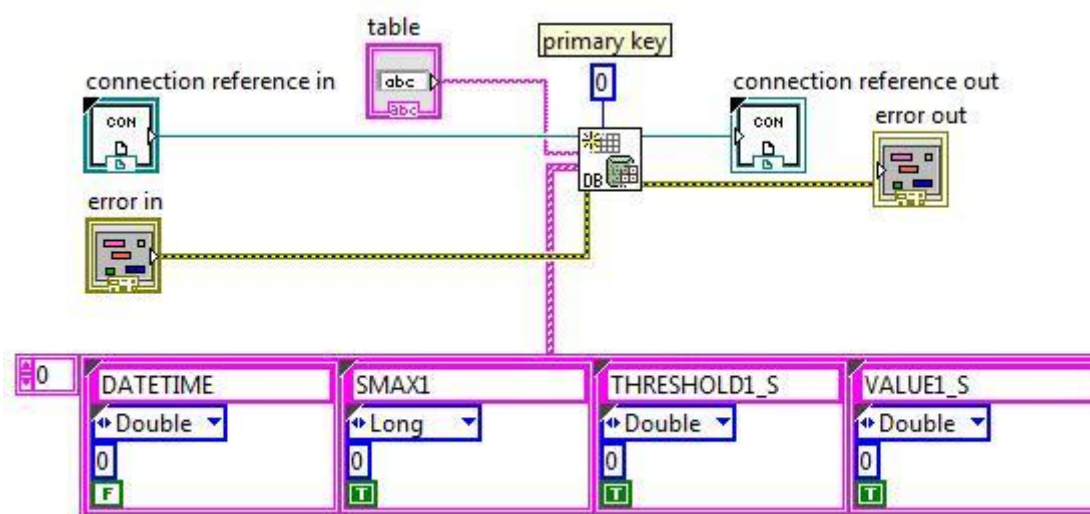


Obrázek 5-1: Vývojový diagram zápisu dat do tabulky EVENT



Obrázek 5-2: Algoritmus zápisu online dat do tabulky EVENT

Na obrázku 5-3 je znázorněno samotné vytvoření tabulky EVENT, kde jsou nadefinovány sloupce podle tabulky 5-2 (viz kapitola 5.1.). Jako primary key tedy primární klíč je zvolen čas DATETIME.



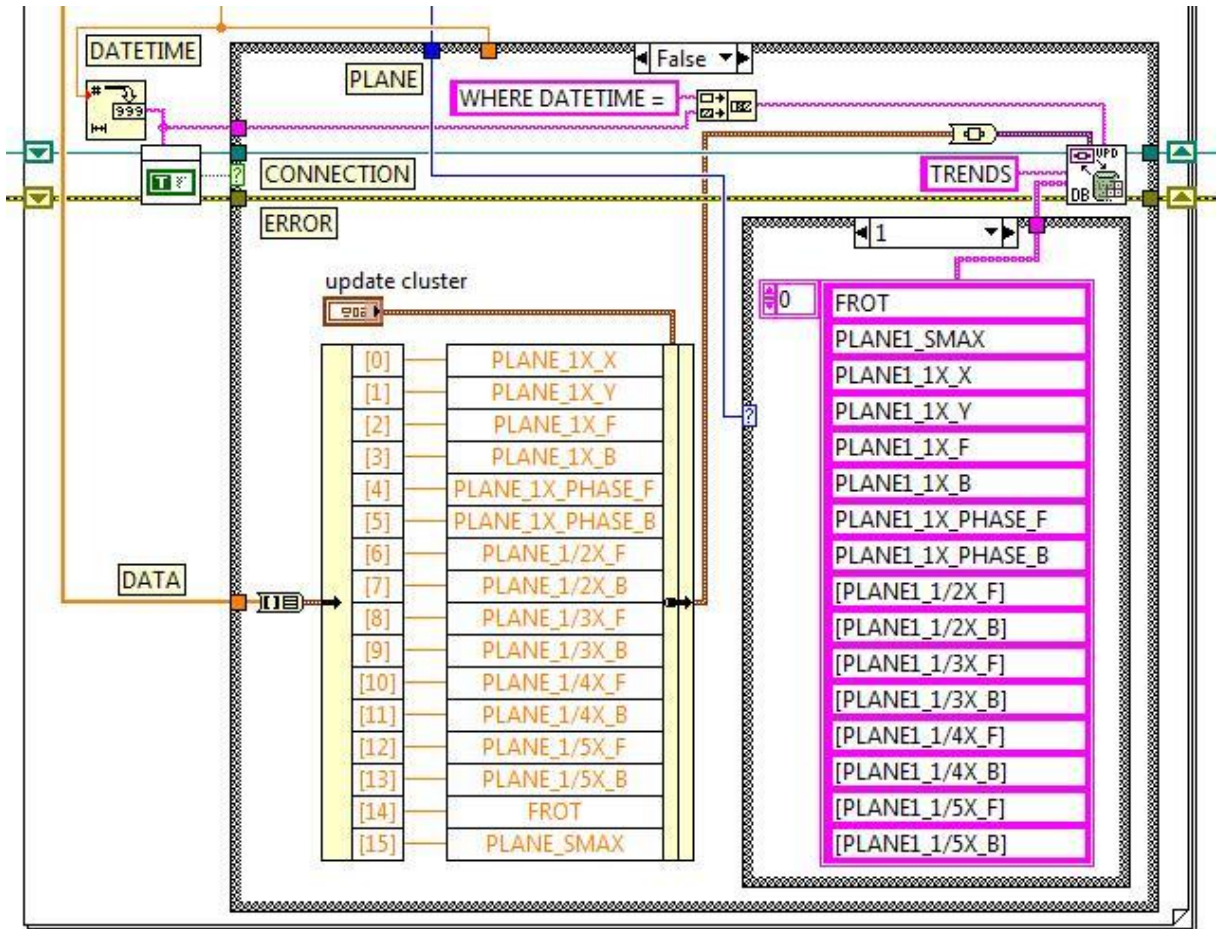
Obrázek 5-3: Subsystém vytvářející tabulku EVENT (výřez)

### 5.3. Zápis do databáze - offline data

Dalším úkolem bylo zajistit propojení stávajících výstupů v podobě textových souborů pojmenovaných `trendy.txt` a `log2.txt` (případně `log.txt`) s novou databází v programu Microsoft SQL Server.

#### TRENDY.TXT

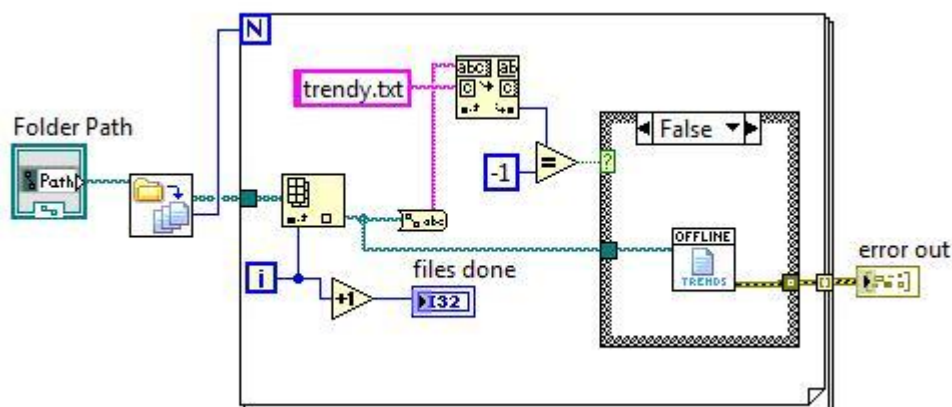
U souborů pojmenovaných `trendy.txt` se předpokládá formát uvedený v kapitole 3.4. Pro jejich načítání do databáze jsem vytvořil jednoduchý modul nazvaný OFFLINE TRENDS, jehož vstupními hodnotami jsou absolutní cesta souboru v souborovém systému a dvě hodnoty řádků charakterizující od kterého řádku do kterého řádku se mají ze souboru data načítat. Defaultně je zde nastaveno čtení celého souboru. Případné uživatelsky špatně nastavené hodnoty jsou ošetřeny. Modul tedy provádí N iterací, kde N je počet řádků souboru určených k načítání. V každé iteraci se z datového souboru čtou hodnoty archivovaných veličin, přičemž jsou převáděny z hodnoty String na hodnotu double. Předpokládá se, že vstupní datový soubor je jednoho ze dvou formátů uvedených v kapitole 3.4. Modul jednotlivé formáty rozezná a podle předem definovaných pozic parsuje do pole. V první pozici pole je časový údaj měření, který se převede na sekundy a testuje se podmínka, zda-li tuto klíčovou časovou hodnotu databáze obsahuje (viz obr. 5-4). V případě prázdné databáze se vytvoří tabulka TRENDS. Pokud databáze klíčový údaj neobsahuje, vytvoří se nový řádek databáze a klíčová časová hodnota se spolu s hodnotami archivovaných veličin daného řádku vstupního souboru zapíše do databáze (příkaz INSERT viz kapitola 4.3.). Ve sloupcích, jejichž data načítaný řádek neobsahuje, jsou hodnoty 0. V případě, že databáze klíčový údaj obsahuje, zapíšou se pouze hodnoty archivovaných veličin odpovídající danému řádku vstupního souboru na příslušné pozice (příkaz UPDATE viz kapitola 4.3.).



Obrázek 5-4: Fáze archivace offline dat ze souboru trendy.txt do databáze

Při vývoji tohoto modulu došlo několikrát k zefektivnění implementace, a tím i ke zkrácení doby běhu. V první verzi byla fáze zpracování textového souboru uskutečněna v MathScriptu (což je blok umožňující interpretaci kódu z programu Matlab) a každý řádek provedl připojení a odpojení databáze. Ve druhé verzi bylo zlepšeno připojení databáze. Modul se připojil k databázi před načítáním prvního řádku a po načtení posledního řádku se odpojil. Ve třetí a finální verzi došlo k nahrazení MathScriptu odpovídajícími bloky LabView. Jednotlivé verze jsem porovnal experimentem při němž bylo načítáno 400 řádků dat ze souboru trendy.txt do databáze. Změřil jsem průměrný čas běhu jednotlivých verzí a spočítal rychlost načítání. Verze jedna s průměrnou rychlostí 210 sekund načítá téměř 2 řádky za sekundu. Verze dva s časem 120 sekund načítá téměř 3,5 řádku za sekundu. Finální verze načte uvedený počet řádků za 37 sekund, což je téměř 11 řádek za sekundu. Tuto rychlost již považuji za dostatečnou, neboť tento modul bude použit pro jednorázové vložení dat do databáze.

Jelikož výstupní tok informací stávajícího systému je archivován rozsáhlou množinou textových souborů stejného formátu a jména, bylo by velice neefektivní spouštět výše uvedený program offline dat pro každý soubor zvlášť. Z tohoto důvodu byl vytvořen modul (viz obr. 5-5), který umožňuje automaticky prohledat celý podstrom souborů a převádět informace ze stávajícího umístění do nové SQL databáze. Výstupními hodnotami jsou výpisy chyb a počet načtených souborů, který se během chodu aplikace sám iteruje.



Obrázek 5-5: Převod offline dat do databáze pomocí prohledávání souborového systému

## LOG2.TXT

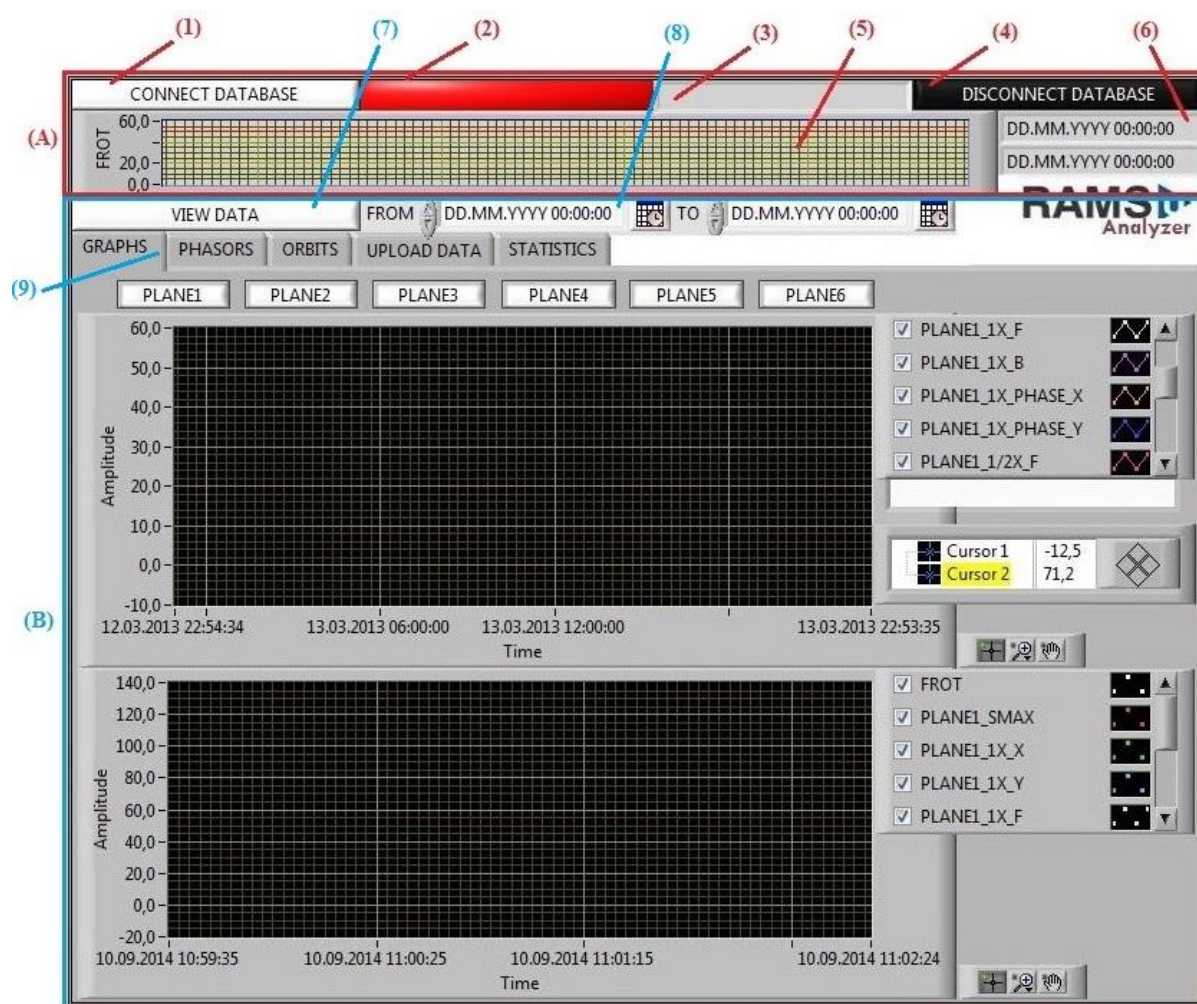
Soubory pojmenované log2.txt (případně log.txt) nemají jednotný formát (viz kapitola 3.4.). Nastavení aplikace a logované události jsou do souboru zaznamenávány v pořadí, v jakém nastaly. Před samotným ukládáním do databáze je tedy nutné zjistit, jaká informace se na právě načítaném řádku vyskytuje. Řádek může obsahovat informaci o rubbingu, ukončení rubbingu, Smax, ukončení Smax, zapnutí systému RAMS Online (v tomto případě je dalším řádkem řádek parametrů podle kapitoly 3.4., se kterými se aplikace spustila), ukončení systému RAMS Online nebo změnu jednotlivých parametrů definovaných v kapitole 3.4. Ještě před klasifikací informace obsažené v řádku se každý řádek přiřazuje jednomu z povolených formátů - česká verze, anglická verze a česká verze bez diakritiky (viz kapitola 3.4.). Po rozpoznání formátu souboru a informace obsažené v řádku je algoritmus podobný s algoritmem pro načítání souborů trendy.txt - aktuální řádek je parsován do pole podle předem daných pozic a ukládán do databáze podle tabulek 5-1 a 5-2 v kapitole 5.1. Podle popsaného algoritmu pracuje modul, který jsem pojmenoval OFFLINE LOGS.

Pro načítání více souborů stejného jména ze složky jsem implementoval obdobný modul jako v případě souborů trendy.txt (viz obr. 5-6). Jediným rozdílem je String konstanta změněná na log2.txt (případně log.txt) a jméno modulu – OFFLINE LOGS. Algoritmus načítání dat ze souboru log2.txt (případně log.txt) není tak efektivní a rychlý jako algoritmus načítání dat ze souboru trendy.txt. Samotná velikost souboru log2.txt (případně log.txt) je oproti velikosti souboru trendy.txt zanedbatelná, a proto výsledky načítání mohou být hodnoceny jako dostatečně uspokojivé.

## 6. RAMS ANALYZER – NÁSTROJ PRO VIZUALIZACI

Pro potřeby vizualizace a analýzy archivovaných dat byla vytvořena aplikace, která nese označení RAMS Analyzer (Rub Advanced Monitoring System). Implementace této aplikace proběhla v prostředí LabView (viz kapitola 3.3). Aplikace má uživateli poskytnout náhled na archivovaná data, na průběhy jednotlivých měřených veličin, na charakteristické ukazatele rubbingu a s tím spojenou statistiku.

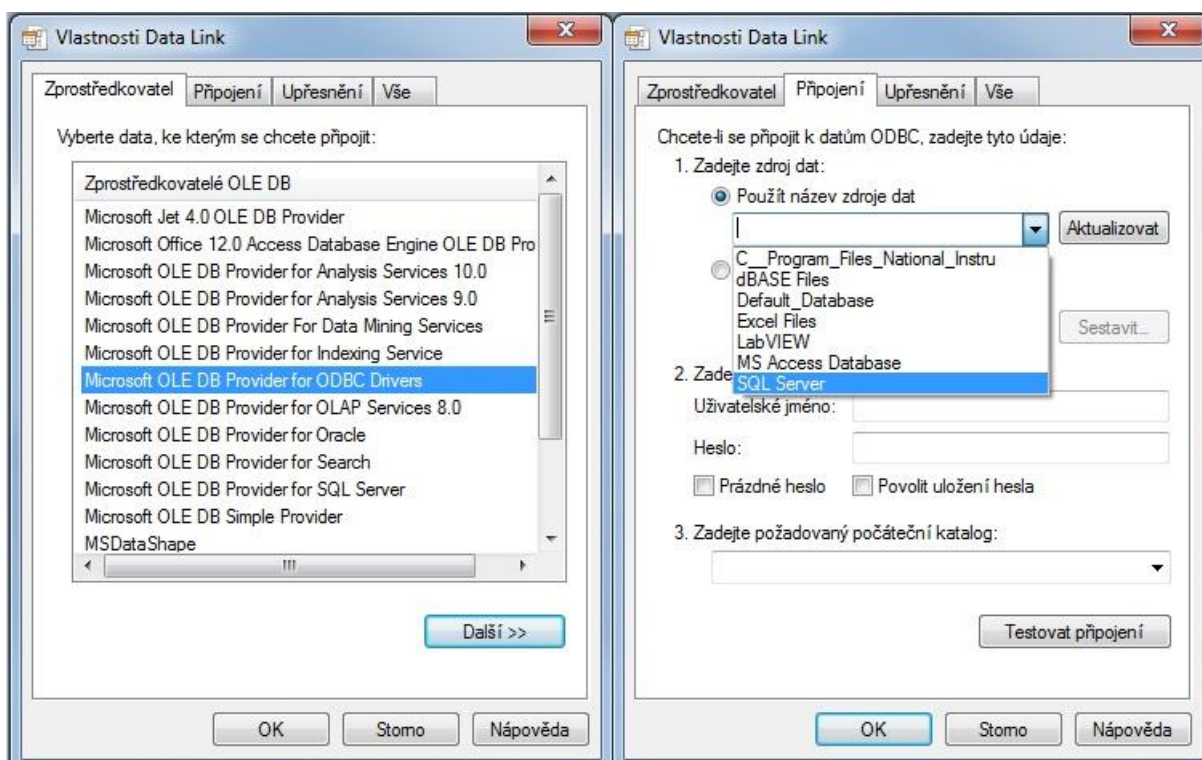
Samotná aplikace je rozdělena na dvě části – část pro připojení k databázi (A) a část pro vizualizaci dat (B). Označení částí aplikace v závorkách v následujícím odstavci vychází z obrázku (6-1). Funkce aplikace jsou ilustrovány pro data naměřená na turbíně tepelné elektrárny Tušimice. Jako počáteční stav aplikace je v tomto případě zvoleno období od 20.2.2012 10:09:32 do 23.02.2012 13:00:12.



Obrázek 6-1: RAMS Analyzer po spuštění

## Připojení k databázi (A)

Tato část aplikace je tvořena dvěma tlačítky – CONNECT DATABASE (1) a DISCONNECT DATABASE (4). Jedná se o tlačítka pro připojení resp. pro odpojení databáze. Po stisknutí tlačítka (1) je uživateli nabídnuto, ke které konkrétní databázi se chce připojit, a jakým zprostředkovatelem má být spojení uskutečněno. Vyžaduje-li si to zvolená databáze, je uživatel nucen zadat uživatelské jméno a heslo. Ve fázi testování této aplikace bylo použito spojení přes ODBC (viz kapitola 4.4) ilustrovaném na obrázku 6-2.



Obrázek 6-2: Připojení k databázi

Úspěšnost spojení interpretují dva indikátory (2) a (3). Indikátor (2) je při spuštění aplikace červené barvy, po úspěšném připojení změní svou barvu na zelenou a při úspěšném odpojení na černou. Indikátor (3) sleduje stejné změny stavů aplikace, jeho interpretace je ovšem v textové podobě (connection succesfull – pro úspěšné připojení, connection failed – při neúspěšném připojení,...). Další prvek tvořící část (A) je graf otáček turbíny (5). Tento graf zobrazuje všechna data o otáčkové rychlosti z tabulky EVENT zvolené databáze. Jedná se o první náhled na data databáze. S grafem jsou spojeny indikátory (6), které zobrazují datum nejstaršího a nejaktuálnějšího záznamu. Pokud zvolená databáze neobsahuje tabulku EVENT, nebo je tato tabulka prázdná, je vyvolána uživatelská zpráva - NO FROT TO DISPLAY (EMPTY DATABASE). Po úspěšném připojení k databázi přecházíme do vizualizační části aplikace.

## Vizualizační část (B)

Tato část se skládá z pěti záložek (9) – GRAPHS, PHASORS, ORBITS, UPLOAD DATA a STATISTICS, z tlačítka VIEW DATA (7) a dvou formulářů (8). Formuláře reprezentují časové rozmezí dat, které bude zobrazeno. Tlačítkem VIEW DATA se provádí samotná vizualizace, která probíhá ve třech záložkách - GRAPHS, PHASORS a ORBITS.

### 6.1. Záložka GRAPHS

Záložka GRAPHS zobrazuje časové průběhy měřených veličin. Funkčnost této části aplikace je ukázána na obrázku 6-3, ze kterého je také v tomto odstavci použito číslování.



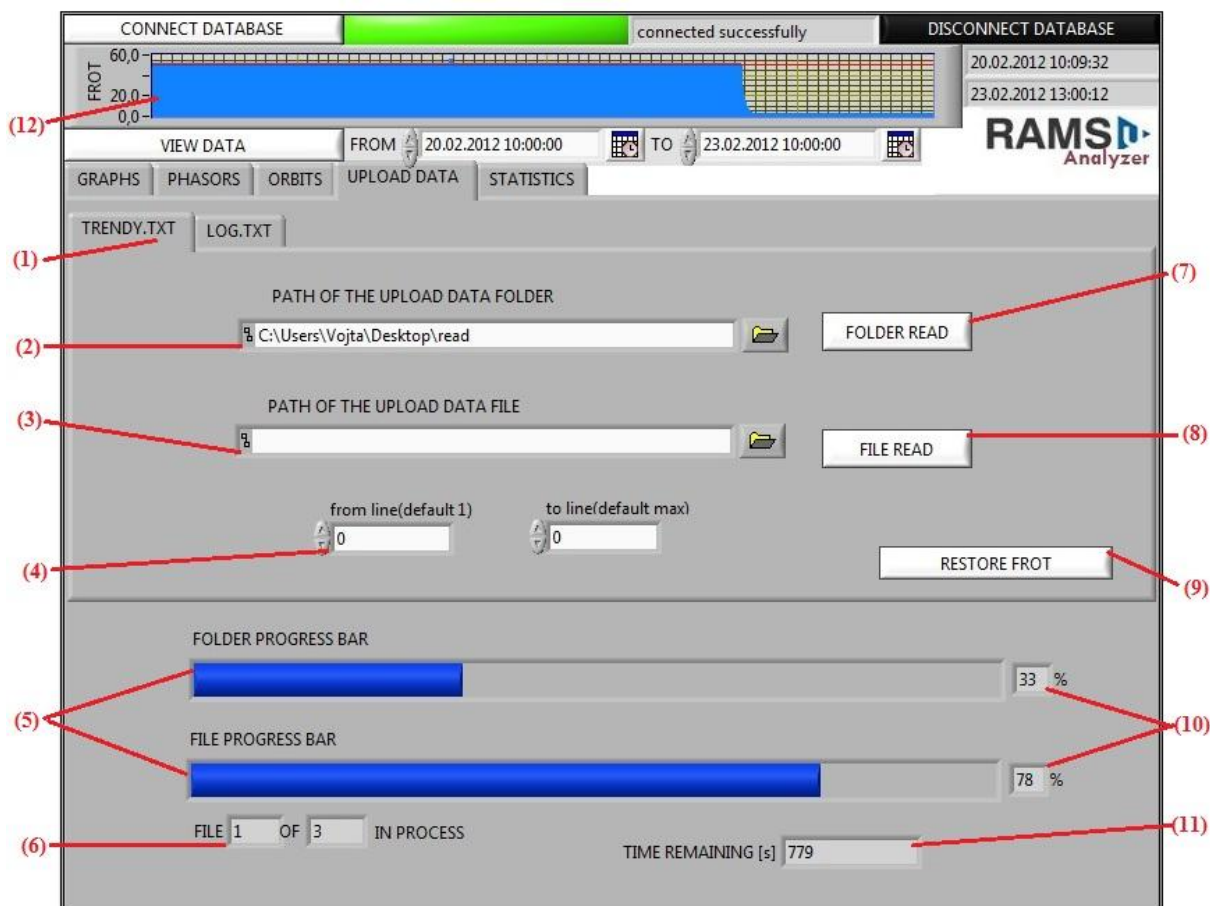
Obrázek 6-3: Zobrazení průběhů veličin v záložce GRAPHS

Uvnitř záložky GRAPHS jsou umístěny dva grafy. Horní graf (2) zobrazuje průběhy měřených veličin definovaných podle tabulky TRENDS (viz kapitola 5.1.) v celém zadaném časovém intervalu. Na obrázku 6-3 je zvolen interval od 22.02.2012 10:00:00 do 23.02.2012 10:00:00. Obecně se předpokládá, že maximální zobrazovaný časový interval bude v řádu dnů až týdnů v závislosti na intervalu mezi jednotlivými měřeními veličin na turbíně (obvykle v řádu sekund až minut). V opačném případě může dojít k přetečení vnitřní paměti LabView z důvodu obecně velmi vysokého počtu zobrazovaných hodnot (99 hodnot za jedno měření).

Tlačítka (1) definují roviny, jejichž veličiny se budou zobrazovat. Jejich funkčnost je založena na principu událostí, a tak mohou pracovat nezávisle na tlačítku VIEW DATA. Formulář (4) je součástí grafu (2) a umožňuje libovolný výběr zobrazovaných veličin. Graf (2) obsahuje dva kurzory (6), které vymezují oblast v ose y. Společně s veličinou definovanou ve formuláři (5) určují časovou podmínku pro zobrazení veličin ve spodním grafu (3). Jinými slovy – v grafu (3) se zobrazí průběhy veličiny v časech, kdy je zadaná veličina ve formuláři (5) v oblasti mezi kurzory. Na obrázku 6-3 jsou kurzory umístěny vzhledem k veličině FROT (otáčková frekvence) a určují oblast, kdeotáčková frekvence klesá z 48 Hz na 5 Hz. V grafu (3) je zobrazen průběh zvolených veličin (8) FROT a SMAX roviny 1. Panely (7) a (9) rozšiřují možnosti práce s grafy.

## 6.2. Záložka UPLOAD DATA

Záložka UPLOAD DATA zajišťuje načítání výstupních souborů stávajícího systému popsaných v kapitole 3.4. Jedná se o implementaci algoritmů načítání souborů trendy.txt a log2.txt (případně log.txt) popsaných v kapitole 5.3. a jejich vhodné zařazení do aplikace. Algoritmy načítání jsou navíc doplněny o ukazatele stavů jednotlivých procesů. Funkčnost záložky je ukázána na obrázku 6-4, ze kterého je v tomto odstavci použito číslování.



Obrázek 6-4: Načítání souborů do databáze v záložce UPLOAD DATA



Záložka UPLOAD DATA obsahuje další dvě záložky (1) – TRENDY.TXT a LOG.TXT. Volbou jedné ze záložek specifikujeme, jaký z výstupních souborů stávajícího systému rubbingu chceme načíst do databáze. Interface obou záložek je podobný. Jediným rozdílem je, že záložka LOG.TXT neobsahuje tlačítko RESTORE FROT (9) a číselné vstupy (4). Na obrázku 6-4 je zobrazena záložka TRENDY.TXT. Ta obsahuje dvě tlačítka FOLDER READ (7) a FILE READ (8). Uživatel má tak na výběr zda-li chce načítat jeden soubor, nebo více souborů umístěných ve stromové struktuře některé ze složek souborového systému. Po stisknutí jednoho z tlačítek je z formuláře (2) resp. (3) čtena cesta k souboru resp. složce určené k načítání. V případě záložky TRENDY.TXT můžeme specifikovat oblast řádků, které se budou načítat. V případě nesmyslného údaje je uživatel upozorněn uživatelskou zprávou. Stejně tak tomu je i v případě, kdy je stisknuto některé z tlačítek (7), (8) bez předchozí specifikace cesty k souboru resp. složce.

Pod záložkami TRENDY.TXT a LOG.TXT je panel zobrazující průběh načítání. Vizualizace je provedena pomocí dvou progress bar ovladačů (5), které zobrazují aktuální stav načítání. Jejich procentuální podoba je pak uvedena v indikátorech (10). Indikátor (6) uvádí číslo právě načítaného souboru a počet všech souborů určených ke zpracování. Indikátor (11) reprezentuje čas  $T_z$ , který zbývá do konce načítání právě zpracovávaného souboru. Jeho výpočet se provádí jednoduchou trojčlenkou, kdy je znám čas spuštění  $t_0$ , aktuální čas  $t$ , počet řádků souboru  $N_r$  a počet již načtených řádků  $n$ . Zbývající čas je pak určen předpisem:

$$T_z = \left( \frac{t - t_0}{n} \right) \cdot (N_r - n) \quad (6.1)$$

V případě, že by byl každý řádek načítán stejně dlouho, čas zbývající do konce načítání vypočítaný podle výše uvedeného vztahu by byl přesný, v ostatních případech se jedná o aproximativní čas. S rostoucím počtem načtených řádků se vypočítaná průměrná doba načítání jednoho řádku blíží přesné hodnotě, z tohoto důvodu je s rostoucím počtem načtených řádků zpřesňován odhad zbývajícího času.

Na obrázku 6-4 je ilustrován průběh načítání tří souborů umístěných v souborovém systému pod složkou s cestou C:\Users\Vojta\Desktop\read.

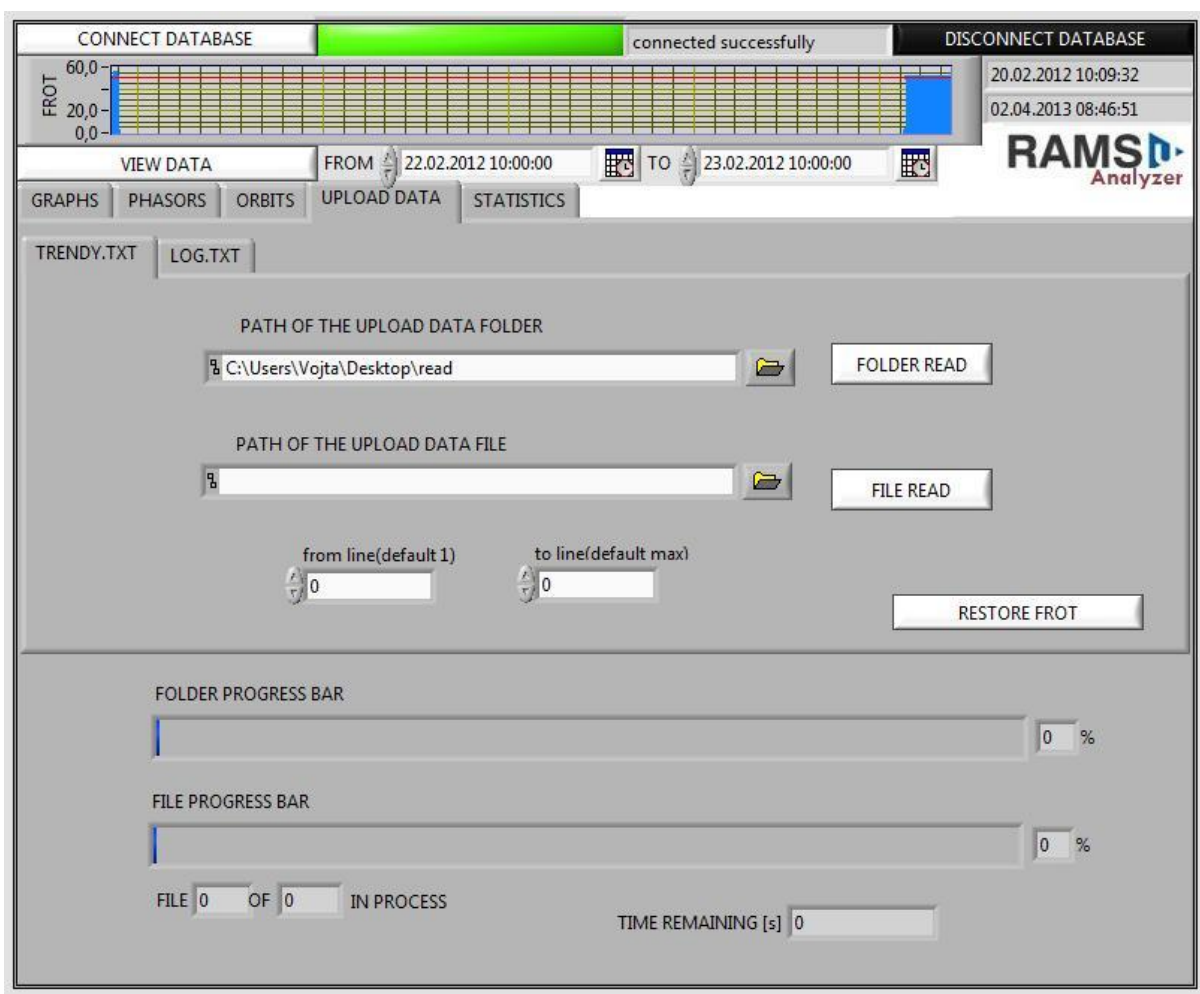


**Obrázek 6-5: Ukázka uživatelského hlášení na konci načítání souborů do databáze**

Po načtení souborů je vyvolána uživatelská zpráva oznamující konečný stav načítání (viz obr. 6-5). Pro načítání souboru trendy.txt nabývá dvou hodnot – COMPLETELY DONE

a INCOMPLETELY DONE. V případě načítání souborů log.txt je v případě vzniku chyby během načítání vyvoláno hlášení o počtu chyb, tj. počet nenačtených řádků. Chyby při načítání je možné pozorovat pouze v případě nesprávného formátu načítaných souborů.

Po stisknutí tlačítka OK přejdou všechny indikátory v panelu pod záložkami TRENDY.TXT a LOG.TXT do počátečního nulového stavu a jsou připraveny pro další načítání. Tlačítko RESTORE FROT v záložce TRENDY.TXT aktualizuje počáteční graf otáčkové rychlosti turbíny (12). Tato situace je ilustrována na obrázku 6-6, kde je vidět, že po načtení souboru, obsahuje databáze již dva úseky naměřených dat – původní od 20.2.2012 10:09:32 do 23.02.2012 13:00:12 a nově načtený úsek měřených dat do 02:04:2013 08:46:51.



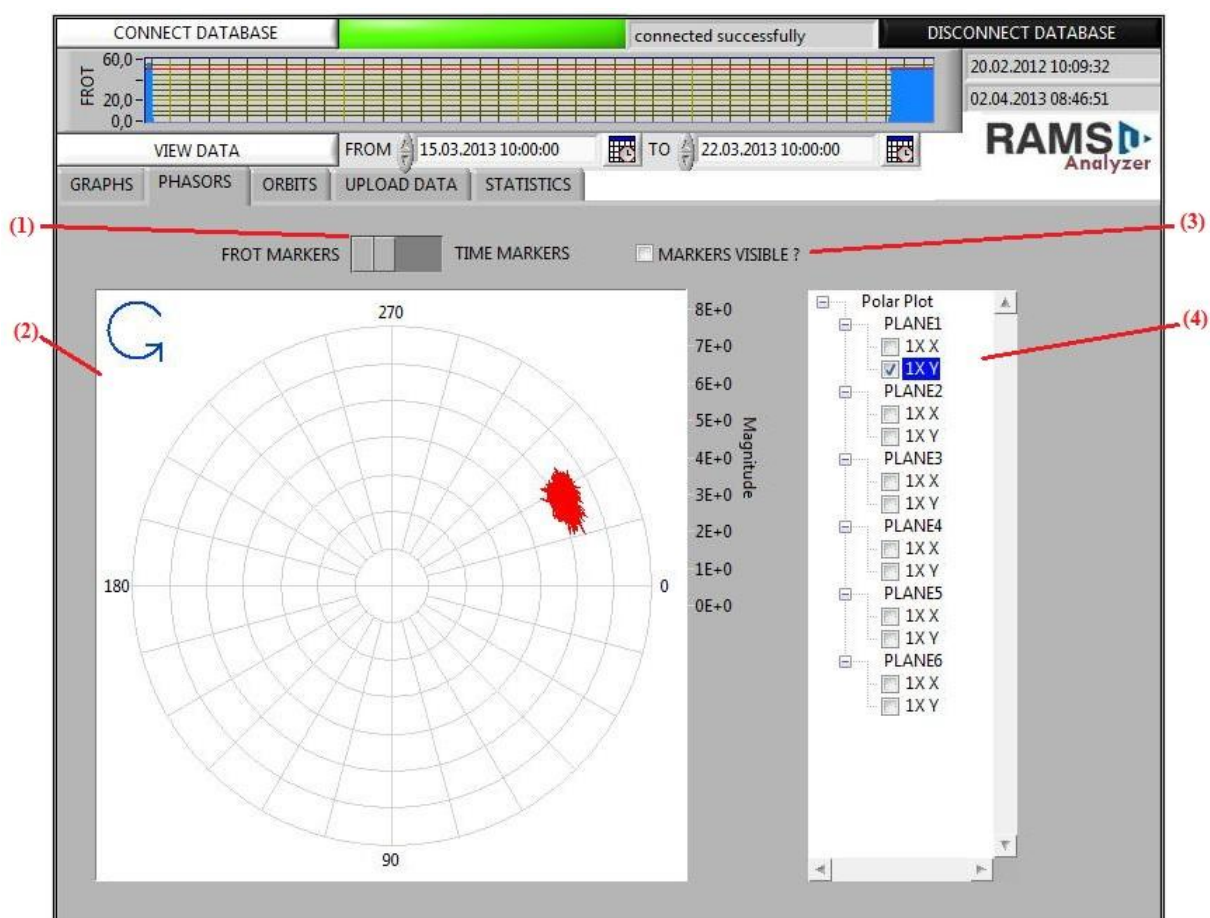
Obrázek 6-6: Stav aplikace po načtení souborů do databáze

### 6.3. Záložka PHASORS

Záložka PHASORS zobrazuje průběh fázorů signálů 1X ve směru x a y měřených podle obrázku 3-3. Vychází z teoretické části popsané v kapitole 2.2.2. Číslování v tomto odstavci je použito dle obrázku 6-7. Samotné zobrazení je realizováno pomocí bloku polar plot, do něhož jsou přivedeny polární souřadnice fázoru v daném čase, tj. amplituda a fáze. Na obrázku je vidět zobrazení průběhu fázorů v grafu (2). Pomocí formuláře (4) můžeme přepínat

mezi jednotlivými fázory signálu 1X měřených ve směrech x a y v každé ze šesti rovin. Přepínací tlačítko (1) indikuje, jaké značení má být u každého z vypočítaných bodů grafu, tj. chceme-li zobrazit otáčkovou rychlost, anebo čas. Ten je počítán od prvního vypočítaného bodů grafu, tj. první fázor v grafu má označení času  $t = 0$ . V případě, že žádné z výše uvedených značení není žádoucí např. z důvodu přehlednosti, může být značení vypnuto formulářem (3), který pracuje na principu události, a je proto narozdíl od přepínacího tlačítka (1) nezávislý na tlačítku VIEW DATA.

Na obrázku 6-7 je vidět průběh fázoru 1X ze směru y na rovině 1 v zadaném čase od 15.03.2013 10:00:00 do 22.03.2013 10:00:00, tj. z dat načtených podle obrázku 6-4.



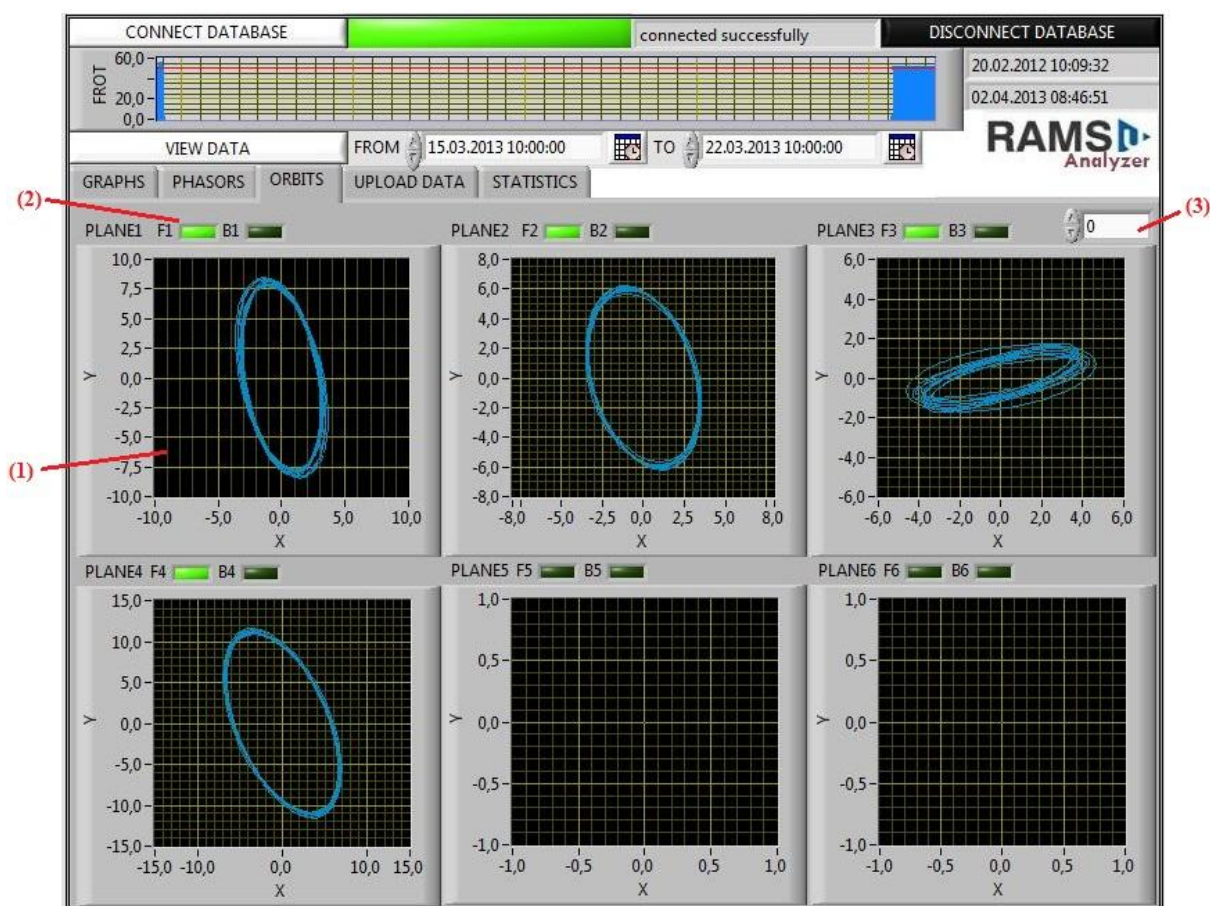
Obrázek 6-7: Průběh fázoru 1X ze směru y na rovině 1

## 6.4. Záložka ORBITS

Záložka ORBITS zobrazuje průběhy výchylek středu rotoru vzhledem ke své ose v každé ze 6 měřených rovin. Tato záložka vychází z teoretické části popsané v kapitole 2.2.1. a 2.2.3. Číslování funkcí záložky v tomto odstavci se řídí obrázkem 6-8. Záložka je členěna do šesti částí. Každá z nich odpovídá jedné z rovin a obsahuje samotný graf (1) a dva boolean indikátory (2). V grafu (1) jsou vykresleny orbity barev červená a modrá, kde červená barva značí orbitu se zpětnou precesí a modrá barva orbitu s dopřednou precesí. Boolean indikátory dopředné a zpětné precese (2) nabývají hodnoty true pokud je mezi vykreslenými orbitami

alespoň jedna s dopřednou resp. se zpětnou precesí. Formulář (3) reprezentuje počet orbit, které budou vykresleny v každém grafu. Defaultně je nastavena hodnota 10. Volba uživatele může nabývat hodnot z celočíselného intervalu 1 až 20. V jiném případě je vykresleno 10 orbit. V případě není-li v databázi pro zvolený časový úsek dostatečný počet záznamů požadovaný ve formuláři (3), je vykresleno pouze tolik orbit, kolik je časových záznamů v databázi pro zvolený časový úsek. Pro vykreslení  $n$  orbit se zvolený časový úsek rozdělí na  $n-1$  rovnoměrných částí a vykreslí se vždy orbita pro poslední časový úsek dané části. Poslední orbitou je orbita odpovídající prvnímu záznamu v databázi pro zvolený časový interval. Samotné vykreslení orbit je zajištěno tlačítkem VIEW DATA. Při prvním stisknutí jsou vykresleny orbity a je proveden autoscale, tj. měřítka jednotlivých souřadnicových os jsou transformována na velikost dané orbity. Při druhém stisku tlačítka VIEW DATA jsou zobrazeny již nezkrácené orbity (viz obr. 6-8).

Na obrázku 6-8 je vykresleno 10 orbit pro každou z rovin kromě roviny 5 a 6, na kterých ve zvoleném časovém úseku neprobíhalo měření. Všechny zobrazené orbity mají dopředný charakter precese.



Obrázek 6-8: Zobrazení orbitů pro každou ze šesti rovin

Výpočet jednotlivých orbit je implementován v MathScriptu (což je blok umožňující interpretaci kódu z programu Matlab) v modulu nazvaném ORBIT PLOT. Klíčová část výpočtu je uvedena na obrázku 6-9. Celý výpočet běží ve for cyklu a pro každý průchod je vypočtena jedna ze zadaného počtu orbit. Vstupními parametry MathScriptu jsou velikost

pole dat – počet orbit – N, pole otáčkové rychlosti – frot, pole amplitud signálu měřeného ve směru x a y – Ax resp. Ay, pole počátečních fází signálu měřeného ve směru x a y – fix resp. fiy, pole amplitud fázorů dopředné a zpětné precese – Ad resp. Az a počet záznamů v databázi pro zvolené období – i\_max. Rozdělení časového intervalu na stejnoměrné úseky je realizováno příkazem na řádku 24. Z důvodu obecně rozdílné velikosti dvou orbit je nutná normalizace kroku pro výpočet jednotlivých orbit. Požadavkem je, aby každá orbita byla reprezentována 100 body v rovině xy. Tato normalizace je provedena na řádku 11. Po definování stohodnotového časového vektoru na řádku 12 je na řádku 13 proveden samotný výpočet orbity podle vzorce (2.6). Na řádce 14 je na vypočítanou orbitu aplikováno otočení o 45° a tím zajištěn přechod od souřadných os umístěných podle obrázku 3-3 ke klasickým Euklidovským souřadnicím. Vzorec pootočení o úhel  $\varphi$  je určen vztahem (6.2).

$$A = A \cdot e^{j \cdot \varphi} \quad (6.2)$$

Na řádku 15 a 16 je vypočtena reálná a imaginární část orbity a tím i souřadnice jednotlivých bodů orbity v ose x resp. y. Na řádkách 18 až 23 se podle vztahů (2.11) určuje, zda-li výše vypočítaná orbita vykazuje dopředný nebo zpětný charakter precese.

```

8   prec=zeros(1,N);
9   in=1;
10  for i=1:N
11      step=0.01/frot(in);
12      t = 0:step:0.99*(1./frot(in));
13      S(i,:)= Ax(in)*sin(frot(in)*2*pi.*t+fix(in))+j*Ay(in)*sin(frot(in)*2*pi.*t+fiy(in));
14      S(i,:)=S(i,:)*exp(j*(-pi/4));
15      x(i,:)=real(S(i,:));
16      y(i,:)= imag(S(i,:));
17
18      if(Ad(in)<Az(in))
19          prec(i)=-1;
20      end
21      if(Ad(in)>Az(in))
22          prec(i)=1;
23      end
24      in=in+floor(i_max/N);
25  end

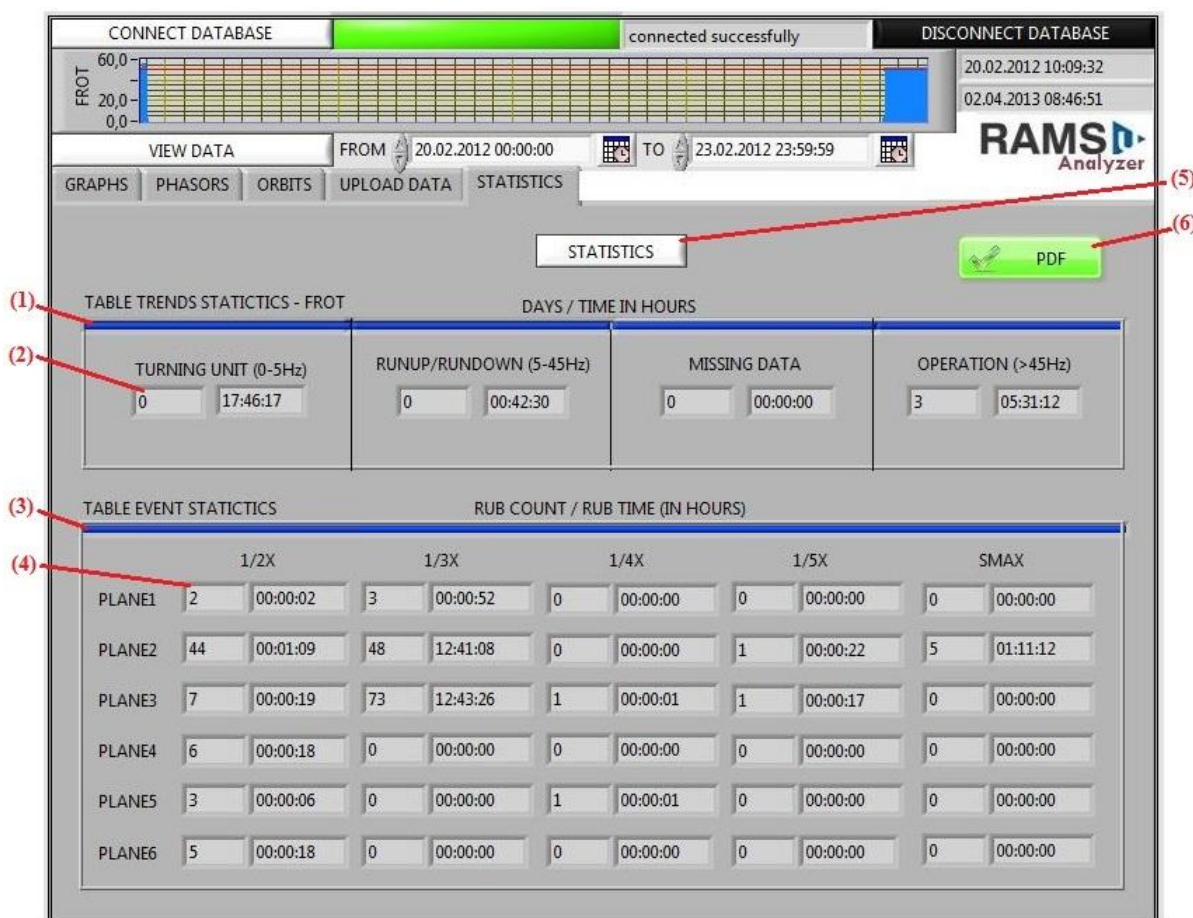
```

Obrázek 6-9: Výpočet orbitů v MathScriptu

## 6.5. Záložka STATISTICS

Záložka STATISTICS poskytuje výpočet statistik z dat uložených v databázi. Pro popis funkčnosti této záložky je v tomto odstavci užito číslování podle obrázku 6-10. Záložka STATISTICS se skládá z tlačítka STATISTICS (4), jehož stiskem se spouští výpočet statistik, tlačítka PDF (6) a dvou oblastí – TABLE TRENDS STATISTICS - FROT a TABEL EVENT STATISTICS. První ze zmíněných oblastí je tabulka obsahující čtyři okna – TURNING UNIT (0-5Hz) – provoz na natáčetle, RUNUP/RUNDOWN (5-45Hz) – náběh a doběh, MISSING DATA a OPERATION (>45Hz) – nominální provoz. Každé z oken obsahuje údaj

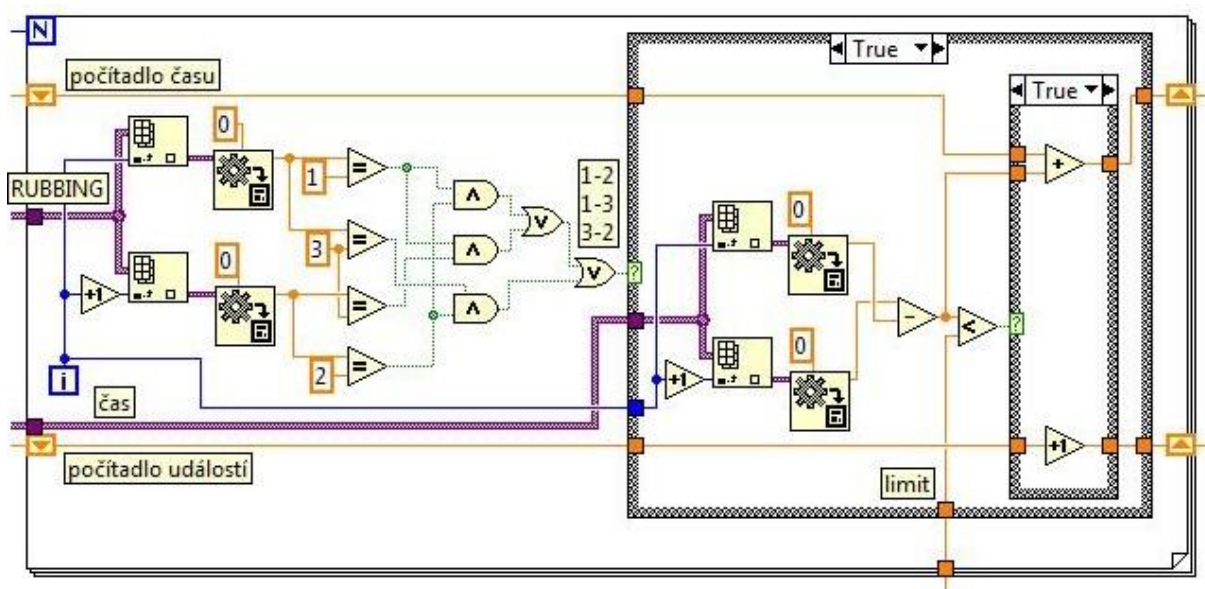
o čase, v kterém byla otáčková frekvence v definovaném intervalu. Pokud ve zvoleném časovém intervalu pro výpočet statistik chybí v tabulce TRENDS data v intervalu delším než je jeden den, sečtou se všechny takto chybějící časové intervaly a zobrazí se do okna MISSING DATA. Z tohoto důvodu se předpokládá, že tabulka TRENDS a tabulka EVENT budou naplněny daty za stejné časové období. Časový údaj oken je rozdělen do dvou indikátorů (2), kde první z nich je ve formátu dnů a druhý ve formátu hodin. Průběh výpočtu statistik každého okna znázorňuje progress bar (1). Oblast TABEL EVENT STATISTICS je tabulkou o velikosti šest řádků krát pět sloupců. Obsahuje časové údaje o detekci rubbingu na příslušné subharmonické frekvenci, či o překročené hladině Smax. Průběh výpočtu této části záložky znázorňuje progress bar (3). Každý prvek tabulky obsahuje dva indikátory (4). První z nich informuje o počtu detekovaných událostí a druhý o časovém trvání všech událostí ve formátu hodin. Tlačítko PDF (6), v případě nastavení hodnoty true před spuštěním samotného výpočtu statistik, exportuje definované statistiky do souboru .pdf, jehož jméno a umístění si uživatel sám zvolí. Vytvoření a stylizace souboru .pdf je realizována pomocí knihovny PDF Toolkit firmy Carya, jež není součástí standartního LabView (viz kapitola 3.3.). Knihovna podporující tuto funkci byla stažena ze zdroje [17]. Jedná se o free verzi, která však není určena pro komerční využití.



Obrázek 6-10: Statistika z archivovaných dat

Na obrázku 6-10 jsou vypočítané statistiky za období od 20.02.2012 00:00:00 do 23.02.2012 23:59:59. Výstup v podobě souboru .pdf je uveden v příloze A. Soubor obsahuje logo RAMS Analyzer pouze v případě, že je tento obrázek pojmenovaný logo.jpg uložen ve stejné složce jako samotný zdrojový kód aplikace. Obecně se předpokládá, že výpočet statistik může být prováděn za dlouhé časové období.

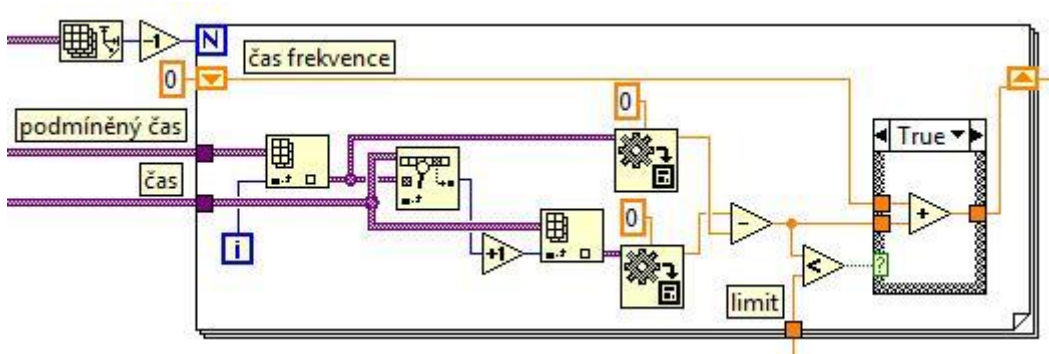
Na obrázku 6-11 je ilustrována implementace samotného výpočtu statistik událostí rubbingu. Z databáze se vyberou takové sloupce DATETIME, jejichž řádky obsahují zvolenou rovinu a frekvenci hodnoty 1, 2, nebo 3 (viz tabulka 5.2. kapitoly 5.1. sloupce RUBBING). Pokud je výsledná velikost pole dat N nenulová, provede se N-1 krát for cyklus uvedený na obrázku 6-11. Počáteční hodnota počítadla času a událostí je nula. V každé iteraci se porovnají dvě po sobě jdoucí události z pole dat. Pokud nabývají hodnot 1-2 (událost nastala – událost skončila), 1-3 (událost nastala – událost skončila a ve stejném čase nastala jiná událost), nebo 3-2 (jiná událost skončila a ve stejném čase nastala nová událost – nová událost skončila) vypočítá se rozdíl časů začátku a konce události, tj. trvání jedné události. Pokud je tato hodnota menší než zvolený limit, je počítadlo času zvětšeno právě o tento rozdíl a počítadlo událostí je zvětšeno o hodnotu 1. Limit je v aplikaci volen 86400 sekund, tj. 1 den. Limit je zde ochranou proti situaci, kdy v databázi chybí ve zvoleném období požadované informace (rubbing je obecně krátkodobá událost, a proto by legitimní rubbing neměl limit 1 den překročit). Výstupem for cyklu je počet detekovaných událostí zvolené frekvence a roviny a jejich čas v sekundách, který se formátuje na čas v hodinách.



Obrázek 6-11: Algoritmus výpočtu statistik událostí rubbingu

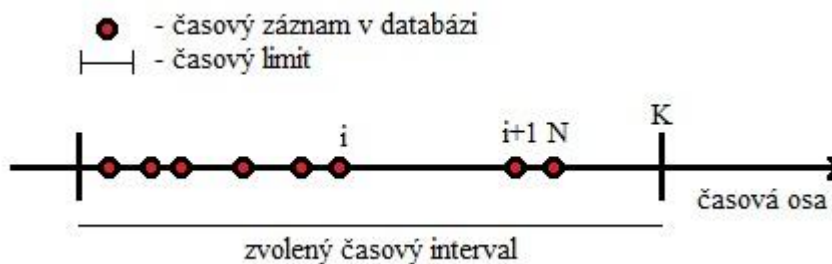
Na obrázku 6-12 je implementace hlavní myšlenky výpočtu otáčkové frekvence v určitém rozmezí hodnot (dále jen frekvenční podmínka) a v zadaném časovém intervalu (dále jen časová podmínka). Nejprve se z databáze vyberou všechny časové okamžiky splňující časovou i frekvenční podmínku (viz obr. 6-12 - podmíněný čas) a všechny časové okamžiky splňující časovou podmínku (viz obr. 6-12 - čas). Velikost pole podmíněného času N je menší

nebo rovna velikosti pole času. V následujícím for cyklu, který běží N-1 krát je pro každou hodnotu podmíněného času nalezen následující časový okamžik z pole celkového času a je vypočítán jejich rozdíl. V případě, že je tento rozdíl menší než zvolený limit (viz popis obrázku 6-11), je tento časový interval načítáván do pomocné proměnné (viz obr. 6-12 – čas frekvence). Obecně lze říci, že z naměřených a archivovaných diskrétních hodnot je vytvořena po částech konstantní funkce, jež v intervalu od diskrétní časové hodnoty  $t_1$  odpovídající funkční hodnotě  $f(t_1)$  do diskrétní časové hodnoty  $t_2$  odpovídající funkční hodnotě  $f(t_2)$  nabývá hodnoty  $f(t_1)$ . Jelikož převádíme diskrétní hodnoty do spojité oblasti času je zřejmé, že volbou metody aproximace lze docílit různých výsledků. V tomto případě aproximace po částech konstantní funkcí poskytuje dostatečně dobré výsledky.



Obrázek 6-12: Algoritmus výpočtu statistik otáčkové frekvence

Výpočet hodnoty MISSING DATA je znázorněn na obrázku 6-13. Ve zvoleném časovém intervalu se vyberou všechny časové záznamy. V případě, že se zde žádné záznamy nevyskytují, je celý zvolený časový interval ohodnocen jako MISSING DATA. V případě, že se zde vyskytuje alespoň jeden časový záznam, je vypočtena vzdálenost od začátku zvoleného intervalu k prvnímu záznamu a vzdálenost od posledního záznamu ke konci zvoleného intervalu. Pokud se ve zvoleném intervalu vyskytuje více než jeden záznam, vypočítávají se navíc postupně časové vzdálenosti po sobě jdoucích časových záznamů. V případě, že je některá ze vzdáleností větší než zvolený limit (v aplikaci 86400 s, tj. 1 den), je přičtena do pomocné proměnné reprezentující celkový čas chybějících dat v databázi. Z obrázku 6-13 je zřejmé, že rozdíl časových záznamů s indexem  $i$  a  $i+1$  je větší než limit. Vzdálenost posledního časové záznamu s indexem  $N$  a konce zvoleného intervalu označeného  $K$  je také větší než limit. Součet těchto vzdáleností bude výsledný čas MISSING DATA.



Obrázek 6-13: Princip výpočtu MISSING DATA záložky STATISTICS



## 7. ZÁVĚR

Rubbing je označení pro kontakt rotující části turbíny s její stacionární částí. Monitorování běhu turbíny za účelem detekce tohoto kontaktu je jedním ze základních principů odhalování nežádoucího chování turbíny. Včasná detekce kontaktu rotor-stator může zabránit velkým finančním a materiálním škodám. Princip funkce turbíny je vysvětlen v první kapitole, ve které je také ukázáno uložení rotoru ve statorové skříni. V druhé kapitole je popsán samotný rubbing, jeho rozdělení, výskyt a jevy, které ho doprovázejí. Rubbing jako takový je většinou jen ukazatelem nesprávné funkce některé z částí turbíny. Vznik rubbingu je často sledován z vývoje fázoru 1X, který s rostoucím kontaktem zvyšuje svoji amplitudu a mění fázi. Ve třetí kapitole byl popsán monitorovací systém rubbingu RAMS Online měřící vibrační charakteristiky v šesti ložiskových rovinách turbíny, implementovaný na jedné z turbín tepelné elektrárny Tušimice. Kromě nezbytného měřícího zařízení je v této kapitole popsán i program, v němž je celá aplikace implementována - LabView. Výstup systému, do něhož se archivují průběhy jednotlivých měřených veličin, je ve formátu dvou textových souborů. Význam informací a veličin v nich obsažených je vysvětlen v téže části.

Jedním z cílů této práce bylo navrhnout efektivnější způsob archivace výstupních dat aplikace. Čtvrtá kapitola se proto věnuje databázovým systémům a jejich komunikaci s programem LabView. Jedním z vhodných nástrojů pro archivaci takto získávaných dat je SQL Server od společnosti Microsoft, který byl v této práci použit. V páté kapitole je uveden návrh nového způsobu archivace dat z měřícího systému RAMS Online. Byly navrženy tři tabulky – EVENT, TRENDS a SETTINGS. První z nich archivuje data událostí, druhá průběhy jednotlivých veličin a třetí systémová nastavení aplikace. Samotný přechod od archivace dat pomocí dvou textových souborů k archivaci pomocí tří tabulek v programu SQL Server byl realizován pomocí tří modulů, jejichž algoritmus je objasněn v páté kapitole.

Dalším krokem a hlavním cílem této práce bylo vytvořit aplikaci, která bude nad archivovanými daty pracovat a vizualizovat je. Vytvořená aplikace RAMS Analyzer je implementována stejně jako monitorovací systém rubbingu v prostředí LabView a nabízí několik možností zobrazení dat. Po připojení k databázi je uživateli umožněno vybrat si jedno z pěti záložek. Záložka GRAPHS nabízí vizualizaci průběhů jednotlivých měřených veličin. Záložka ORBITS vykresluje orbity, tj. trajektorie středu rotoru během otočení kolem své osy. Záložka PHASORS zobrazuje fázory signálů 1X ve směru x a y každé z rovin. Záložka UPLOAD DATA umožňuje manuálně naplnit databázi daty ze starého formátu dvou textových souborů. Záložka STATISTICS počítá čas provozu na natáčedle, čas náběhu a doběhu, nominální provoz a časový interval dat, která za zvolené období nejsou v databázi. Poskytuje také informaci o počtu událostí detekovaných na turbíně – rubbingu. Výstupním formátem je počet událostí a časový interval jejich trvání. Tyto údaje je možno exportovat do souboru .pdf.

Vytvořená aplikace tedy poskytuje uživateli možnost práce s daty naměřenými na turbíně. Aplikace byla testována na datech za časové období mezi roky 2012 a 2013. V případě použití přibližném v kapitole šest prokazovala robustní charakter – vykazovala stabilitu při svém běhu. V současné době je však nutné další testování a v případě nalezení některého z nestabilních chování je nezbytné tuto chybu v další verzi aplikace odstranit.

## 8. SEZNAM POUŽITÉ LITERATURY

- [1] ČEZ: *Elektrárny Tušimice*, 2013 [online]. [cit. 14.4.2013]. Dostupné z: <http://www.cez.cz/cs/vyroba-elektriny/uhelne-elektrarny/cr/tusimice.html>
- [2] Škorpík, J.: *Transformační technologie*, ISSN 1804–8293, [online]. VUT v Brně, 2006. [cit. 16.4.2013]. Dostupné z <http://oei.fme.vutbr.cz/jskorpik/index.html>
- [3] *Bez názvu* [Parní turbína] [obrázek], 2007 [online]. [cit. 1.4.2013]. Dostupné z: <http://www.leander-project.homecall.co.uk/Engines/turbine.jpg>
- [4] Jakl, J.: *Metody zpracování signálů a detekce kontaktu rotor-stator v diagnostice parních turbín*. ZČU v Plzni, Práce ke státní doktorské zkoušce, Plzeň 2011.
- [5] Musyznska, A.: *Rotordynamics*. Tylor & Francis Group, CRC Press, 2005, ISBN 0-8247-2399-6.
- [6] Vaseghi, S.V.: *Advanced Digital Signal Processing and Noise Reduction*. John Wiley & Sons, New Jersey, 1987.
- [7] Liška, J.: *Modely periodických procesů. Vybuzené kmitání v točivých strojích*. 6. přednáška předmětu Modelování a simulace 1, Plzeň, 2012.
- [8] WolframMathWorld: *Sine*, 2013 [online]. [cit. 15.3.2013]. Dostupné z: <http://mathworld.wolfram.com/Sine.html>
- [9] Liška, J.: *Neue Methode zur Erkennung von Anstreifen einer Welle und zur dessen Lokalisierung*. Mülheim, 2013.
- [10] National Instruments: *LabView User Manual*, 2003 [online]. [cit. 25.2.2013]. Dostupné z: [www.ni.com](http://www.ni.com)
- [11] Flídr, M.: *Vnější sběrnice, Vnitřní sběrnice*. 8. a 9. přednáška předmětu Počítačové systémy, Plzeň, 2008.
- [12] Kučerová, H.: *Databázový software*, [online]. Vyšší odborná škola informačních služeb, Praha, 2005. [cit. 21.4.2013]. Dostupné z: <http://web.sks.cz/users/ku/soft.htm>
- [13] National Instruments: *Database Connectivity Toolset User Manual*, 2001 [online]. [cit. 20.3.2013]. Dostupné z: <http://www.ni.com/pdf/manuals/321525c.pdf>
- [14] Hans-Petter Halvorsen: *Database Communication in LabVIEW*, 2011 [online]. Telemark University College, Porsgrunn, Norway. [cit. 22.4.2013]. Dostupé z: <http://home.hit.no/~hansha/documents/labview/training/Database%20Communication%20in%20LabVIEW/Database%20Communication%20in%20LabVIEW.pdf>
- [15] Microsoft: *SQL Server*, 2013 [online]. [cit. 12.3.2013]. Dostupné z: <http://www.microsoft.com/sqlserver/en/us/default.aspx>
- [16] w3schools.com: *SQL Tutorial*, 2013 [online]. [cit. 12.3.2013]. Dostupné z: <http://www.w3schools.com/sql/>
- [17] Carya Automatisering: *PDF Toolkit for LabVIEW*, [online]. [cit. 17.4.2013]. Dostupné z: <http://www.carya.nl/index.php?catid=730&lang=uk>

## A. STATISTICKÝ REPORT

### STATISTICS REPORT

FROM: 20.2.2012 0:00:00  
TO: 23.2.2012 23:59:59



Rub count / Rub time	SMAX	1/2X	1/3X	1/4X	1/5X
PLANE1	0 / 00:00:00	2 / 00:00:02	3 / 00:00:52	0 / 00:00:00	0 / 00:00:00
PLANE2	5 / 01:11:12	44 / 00:01:09	48 / 12:41:08	0 / 00:00:00	1 / 00:00:22
PLANE3	0 / 00:00:00	7 / 00:00:19	73 / 12:43:26	1 / 00:00:01	1 / 00:00:17
PLANE4	0 / 00:00:00	6 / 00:00:18	0 / 00:00:00	0 / 00:00:00	0 / 00:00:00
PLANE5	0 / 00:00:00	3 / 00:00:06	0 / 00:00:00	1 / 00:00:01	0 / 00:00:00
PLANE6	0 / 00:00:00	5 / 00:00:18	0 / 00:00:00	0 / 00:00:00	0 / 00:00:00

#### FROT STATISTICS (Days / Hours)

TURNING UNIT (0-5Hz): 0 / 17:46:17  
 RUNUP/RUNDOWN (5-45Hz): 0 / 00:42:30  
 OPERATION (>45Hz): 3 / 05:31:12  
 MISSING DATA: 0 / 00:00:00

University of West Bohemia  
Department of Cybernetics

[www.carya.nl](http://www.carya.nl), PDF Toolkit For LabVIEW!

Obrázek A-1: Statistický report RAMS Analyzeru

## B. OBSAH PŘILOŽENÉHO CD

- Backup database – adresář obsahující zálohu databáze Microsoft SQL Server za období testované v kapitole 6 a textový soubor s popisem konfigurace připojení se k databázi pomocí ODBC (viz kapitola 4.4.)
- Bachelor thesis – adresář s bakalářskou prací ve formátu pdf.
- Data files – adresář obsahující ukázkou tří .zip souborů s daty měřených veličin stávajícího systému rubbingu, tj. dva textové soubory – trendy.txt a log.txt (viz kapitola 3.4.)
- RAMS Analyzer – adresář se spustitelným souborem RAMS\_Analyzer.exe aplikace popsané v kapitole 6.
- Statistics report – adresář obsahující ukázkou statistického výstupu aplikace RAMS Analyzer ve formátu pdf (viz příloha A).