

VŠB – TECHNICKÁ UNIVERZITA OSTRAVA

FAKULTA STROJNÍ

KATEDRA AUTOMATIZIČNÍ TECHNIKY A ŘÍZENÍ



**Diagnostika spalovacích motorů na základě měření
kolísání jejich úhlové rychlosti**

*Internal combustion engine diagnostics based on rotational speed variation
measurements*

Zpracoval: Jan Benč

Ostrava: 22.4.2009

ANOTACE PRÁCE

Tato práce je věnována měření kolísání úhlové rychlosti při ustálených otáčkách, nazývané též úhlové kmity za rotace a jeho praktickému využití při diagnostice spalovacích motorů. Popis je zaměřen na softwarové zajištění měřicí úlohy. Za tímto účelem byla vytvořena aplikace zajišťující sběr a následné zpracování a zobrazení naměřených dat. V této práci se tedy seznámíme s jednotlivými prvky této aplikace zajišťující výpočet spektra signálu, výpočet nominální úhlové rychlosti, digitální filtraci, derivaci signálu a mnoho dalšího. Aplikace je vytvořena v programovacím jazyce Visual Basic a založena na platformě Microsoft .NET Framework.

ANNOTATION OF THE THESIS

Thesis deals with the measurement of the shaft angular vibration and the rotation uniformity at the quasi steady state rotation and its usage in diagnostics of engines. Describe is focused on the software support. For this purpose the software application was designed. The main tasks are follows signal acquirement, data processing and displaying output data. A user can check the validity signal using frequency spectrum. The software main functions consist of the evaluation of nominal angular velocity, digital filtration, derivation with respect to time etc. Software language is Visual Basic, witch is based on Microsoft .NET Framework.

Obsah

Seznam použitého značení	1
1 Úvod.....	2
2 Softwarové zajištění měřené úlohy	2
2.1 Sběr naměřených dat.....	2
2.2 Nominální rychlost otáčení (Tachometr).....	4
2.3 Autospectrum.....	5
2.4 Digitální filtry	6
2.5 Rovnoměrnost otáčení (Rotation Uniformity).....	8
3 Použitá literatura	11

Seznam použitého značení

DFT	Discrete Fourier transform (Diskrétní Fourierova transformace)
DI	Differential
FFT	Fast Fourier transform (rychlá Fourierova transformace)
FFTW	Fastest Fourier Transform in the West
FIR	Finite Impulse Response (konečná impulsní odezva)
FT	Fourierova transformace
IC	Internal Combustions (vnitřní detonace)
IRC	Incremental rotary encoder (inkrementální snímač)
LabVIEW	Laboratory Virtual Instruments Engineering Workbench
NI	National Instruments
OS	Operační Systém
PC	Personal Computer (osobní počítač)
PWR	Power (Celkový výkon signálu)
RMS	Root Mean Squared (Efektivní hodnota)
RPM	Revolution Per Minute (Otáčky za minutu)
SE	Single-ended
TEDS	(Transducer Electronic Data Sheet)
TTL	(Transistor – Transistor - Logic)
USB	Universal Serial Bus (univerzální sériová sběrnice)

Úvod

V této práci budou popsány nejpodstatnější metody potřebné pro sběr dat z měřicí karty NI USB-6009. Dále následuje popis základních vlastností nástroje Tachometr, který slouží k určení nominální úhlové rychlosti. Tento nástroj se vyznačuje především interpolací okamžiku průchodu signálu spouštěcí hladinou, což snižuje chybu odečtu asi padesátkrát. Na tento popis navazuje popis nástroje Autospectrum. DFT je zde realizována algoritmem FFTW, který se vyznačuje vysokým výkonem výpočtu a možností zadání libovolné délky vstupního signálu. Samozřejmostí jsou i připojená časová okna a měřítko v RMS, PWR a PWD. Pro zpracování signálu pomocí digitálních filtrů byly naprogramovány digitální FIR filtry dolní, horní, pásmová propust a filtr pro Hilbertovu transformaci. Poslední nástroj slouží k určení rozměrnosti otáčení. Tento nástroj je založen na metodě fázové demodulace a využívá dříve jmenovaných prvků.

Softwarové zajištění měřené úlohy

Pro tvorbu aplikace bylo zvoleno prostředí Microsoft Visual Studio, programovací jazyk Visual Basic a platforma Microsoft .NET Framework. Pro doplnění jsou uvedena vývojová prostředí, které se nabízejí pro tvorbu měřicích aplikací:

- LabVIEW
- MATLAB
- Microsoft Visual Studio

2.1 Sběr naměřených dat

Ze studia vzorových příkladů, lokální nápovědy i článků na internetu vyplynulo deset základních funkcí pro založení vlastní aplikace. Mimo tyto funkce jsou v aplikaci použity i další prvky knihovny NI - DAQmx.NET, které zjednodušují uživatelské rozhraní a stabilizují měřicí proces.



Obr. 2.1 Postup tvorby algoritmu pro sběr dat

První krok při tvorbě měřicí aplikace je vytvořit task. Task by se dal volně přeložit do češtiny jako měřicí úloha, která reprezentuje vstupní či výstupní operace měřicí karty.

```
myTask = New Task()
```

Po vytvoření tasku následuje vytvoření virtuálních kanálů pro vstup či výstup dat z karty. V naší aplikaci je využita třída pro analogové vstupní signály AIChannel:

```
myTask.AIChannels.CreateVoltageChannel(CardSet.PhysicalChannel, "",
AITerminalConfiguration.Differential, -CardSet.Ranges_
, CardSet.Ranges, AIVoltageUnits.Volts)
```

`CardSet.PhysicalChannel` je řetězec reprezentující reálné vstupy a výstupy zařízení. Existují přitom dva formáty jak daný řetězec zapsat. Prvním je řetězec `"Dev1/ai0:2"`, který zahrnuje vstupy ai0, ai1 a ai2 a druhým `"Dev1/ai0, Dev1/ai2"`, který na rozdíl od předchozího řetězce zahrnuje pouze vstupy ai0 a ai2. Dev1 je v tomto řetězci název zařízení, který mu je přidělen prostřednictvím MAX.

`AITerminalConfiguration.Differential` je výčtový datový typ pomocí něj nastavíme režim čtení na „Single-ended“, nebo na „Differential“. Pro měření je důležité také nastavit vlastnosti vzorkování. K tomuto účelu slouží níž funkce `ConfigureSampleClock`.

```
myTask.Timing.ConfigureSampleClock("", CardSet.SampleRate,
SampleClockActiveEdge.Rising, SampleQuantityMode.FiniteSamples_
, CardSet.SampPerChan)
```

První parametr udává zdroj časového signálu. Tento parametr je v našem případě prázdný řetězec. Používáme tedy vzorkovací hodiny měřicí karty. Při použití jiného časového zdroje by řetězec vypadal následovně `"/Dev1/PFI0"`. `CardSet.SampleRate` parametr je typu integer a udává vzorkovací frekvenci.

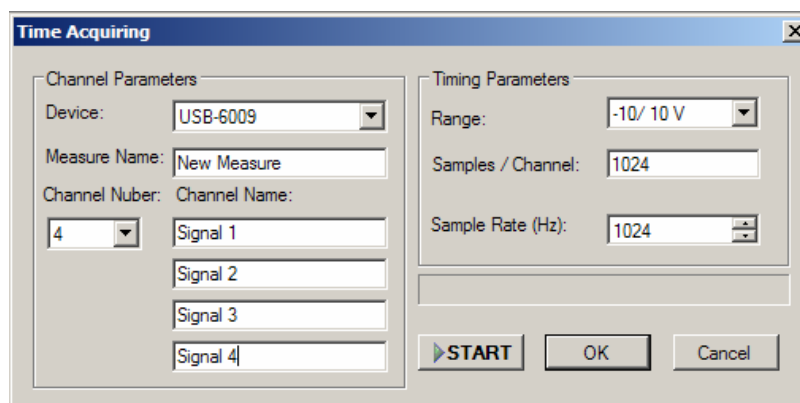
`SampleQuantityMode.FiniteSamples` tento parametr je opět výčtovým typem. Je možné volit ze tří možností. `FiniteSamples` zvolený v naší aplikaci po naměření požadovaného počtu vzorků ukončí měření. Při volbě `ContinuousSamples` měření probíhá průběžně, přičemž po každém dosažení nastaveného počtu vzorků, odesílá data programu přes funkci `ReadMultiSample`. Obě předchozí nastavení využívají buffer. Opakem je tomu u poslední možnosti `HardwareTimedSinglePoint`, kde je čtení a zápis prováděn po jednom vzorku. Posledním parametrem `CardSet.SampPerChan` nastavíme počet požadovaných vzorků.

```
' Verifikace Tasku
```

```
myTask.Control(TaskAction.Verify)
reader = New AnalogMultiChannelReader(myTask.Stream)
```

```
' Ctení dat
```

```
Measur.AcquiredData=reader.ReadMultiSample(CardSet.SampPerChan)
```



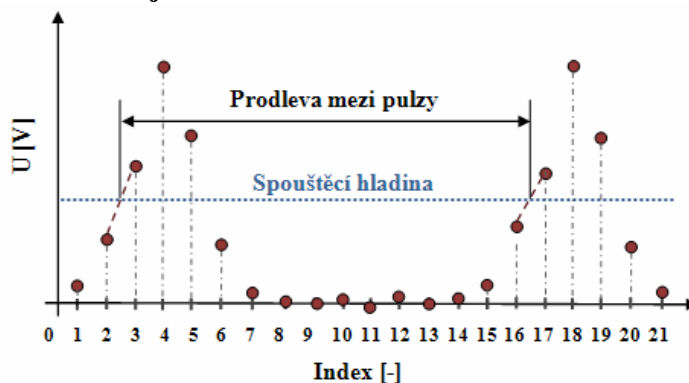
Obr. 2.2 Dialogové okno pro nastavení karty

Po nastavení všech potřebných vlastností tasku provedeme jeho verifikaci. Čtení dat je provedeno členskou funkcí `ReadMultiSample` nově vytvořeného objektu `reader`. Po ukončení měření se `Task` ruší pomocí metody `dispose`. Algoritmus pro měření dat z karty je v aplikaci použit na několika místech. Jedním z nich je i formulář sloužící k měření signálů určité časové délky, který je prezentován na obr. 2.2. Při měření většího počtu vzorků docházelo u předchozího způsobu čtení dat k situacím, kdy se buffer nestačil dostatečně rychle vyprazdňovat. Docházelo tak k častým kolizím. Tuto vadu odstraňuje `analogCallback`. Zde jsou data brány v menších dávkách a postupně ukládány do pomocné proměnné. Celý proces se rekurzivně opakuje, dokud není naměřen potřebný počet vzorků, nebo dokud není zrušen `Task`.

```
reader.SynchronizeCallbacks = True
analogCallback = New AsyncCallback(AddressOf AnalogCallbackGetData)
reader.BeginReadMultiSample(Convert.ToInt32(OptimalSample), _,
analogCallback, myTask)
```

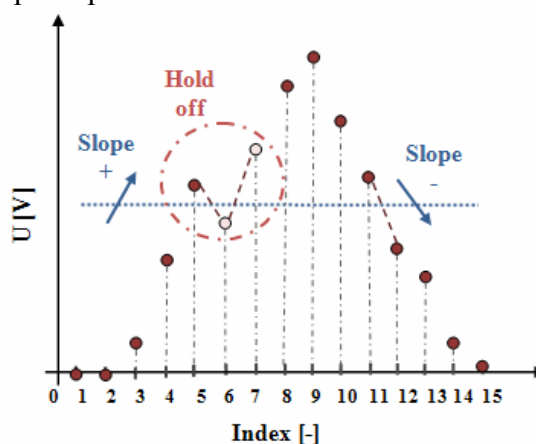
2.2 Nominální rychlost otáčení (Tachometr)

Předmětem zpracování tohoto nástroje je pulzní signál, generovaný tachosondou, nebo IRC snímačem. Rychlost otáčení je v tomto signálu určena časovou prodlevou mezi jednotlivými impulsy viz **Chyba! Nenalezen zdroj odkazů.**



Obr. 2.3 Ukázka vzorkovaného pulzního signálu (detekce impulsu algoritmem)

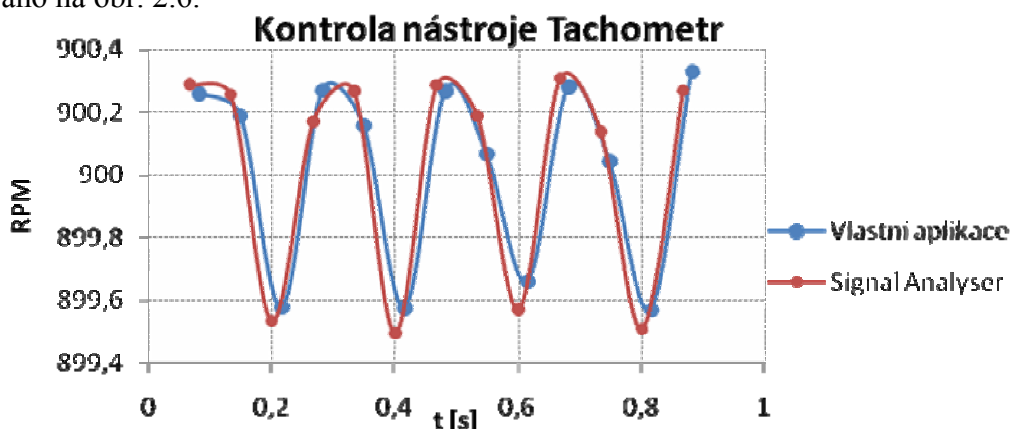
Pro detekci impulsu algoritmem je zapotřebí nejdříve určit spouštěcí hladinu, kterou lze nastavit v dialogovém okně pro tento nástroj za pomoci položky `Level`. Jedná se o procentuální hodnotu z maximální hodnoty naměřeného signálu. Časová prodleva mezi impulsy je tedy určena dvěma průchody přes spouštěcí hladinu.



Obr. 2.4 Ukázka vlastností nástroje Tachometr na samostatném pulzu

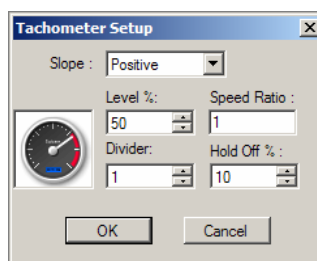
Časová prodleva mezi dvěma náběžnými hranami je pak dána podílem počtu vzorků a vzorkovací frekvence. Pro vzorkovací frekvenci tedy platí nepřímá úměra mezi počtem impulsů generovaných za otáčku a relativní přesností měření délky časového intervalu. Přesnost

vyhodnocení je přitom dána poměrem frekvence první harmonické signálu a vzorkovací frekvence. Neurčitost z titulu vzorkovací periody je v našem algoritmu snížena interpolací okamžiku průchodu signálu spouštěcí hladinou. Chyba odečtu se tak sníží asi padesátkrát. Mimo dříve popsané vlastnosti nástroj Tachometr disponuje ještě Speed Ratio, jenž slouží k zadání převodového poměru a dále funkci Hold - Off. Zde se jedná o procentuální hodnotu z časové prodlevy mezi jednotlivými pulzy, na které je vypnuta detekce průchodu spouštěcí hladinou. Slouží tedy jako prevence před šumem kolem spouštěcí hladiny. Poslední vlastnost Divider využijeme, pokud je generováno více, než jeden pulz za otáčku. V tomto případě je zadán počet pulzů za otáčku. Dialogové okno pro nastavení vlastností nástroje Tachometer je prezentováno na obr. 2.6.



Obr. 2.5 Porovnání výsledků vlastní aplikace s programem Signal Analyser

Pro kontrolu byl na vstup tachometru přiveden pulzní signál o frekvenci 15 Hz, který se svou frekvencí blíží volnoběžným otáčkám automobilu. Výslednou nominální rychlost otáčení vlastní aplikace jsem následně porovnal s výsledky programu Signal Analyser. Toto srovnání je prezentováno na obr. 2.5.



Obr. 2.6 Dialogové okno pro nastavení Nástroje Tachometer

2.3 Autospectrum

Pro výpočet DFT byla použita komponenta založená na algoritmu FFTW, jenž vyvinuli Matteo Frigo a Steven G. Johnson. Tato komponenta je schopna provádět výpočet DFT pro reálná i komplexní data o libovolné dimenzi a libovolné délce (včetně délky rovnající se prvočíslu). Další velkou předností tohoto algoritmu je i vysoká rychlost. Komponenta byla do programu zařazena teprve po odzkoušení na jednoduchých příkladech. Zde byly pomocí níže uvedené funkce generovány harmonické signály a po průchodu transformací kontrolovány výsledky. [Tůma 2009]

```
Private Function f(ByVal x As Double) As Double
    f = AMPLITUDE * Math.Sin(2 * Math.PI * x / SAMPLERATE * FREQUENCY)
End Function
```

Výsledkem výpočtu DFT je dekompozice signálu na harmonické frekvenční složky. Při vstupním vektoru o délce n , je absolutní hodnota složky signálu n násobně větší než jeho amplituda a harmonická složka spektra připadá na frekvenční pásmo o šířce převrácené hodnoty doby měření t vstupním vektoru. Pro praxi je toto měřítko nepřehledné a proto byl výstup

komponenty FFTW přepočítán do měřítka nového. Zde na ose x vystupuje frekvence v hertzech a pro amplitudu je možné zvolit tři varianty: [Tůma 2009]

- RMS (Efektivní hodnota)
- PWR (Celkový výkon signálu)
- PSD (Výkonová spektrální hustota)

Pro ukázkou je uvedena část kódu zabývající se výpočtem RMS. Výpočet efektivní hodnoty je dán vztahem 3.1, Kde x jsou hodnoty odpovídající výstupnímu vektoru FFTW.

$$RMS(k) = \begin{cases} \frac{x_0}{n} & ; k = 0 \\ 2 \cdot \frac{|x_k|}{n \cdot \sqrt{2}} & ; k > 0 \end{cases} \quad (2.1)$$

```
Case Scale.RMS
  For k As Integer = 0 To UBound(DataIn)
    If k = 0 Then
      Ydata(k) = Math.Abs(DataIn(k) / NumOfSample)
    Else
      Ydata(k) = Math.Abs((2 * DataIn(k)) / (Math.Sqrt(2) * NumOfSample))
    End If
  Next
Case Scale.PWR
```

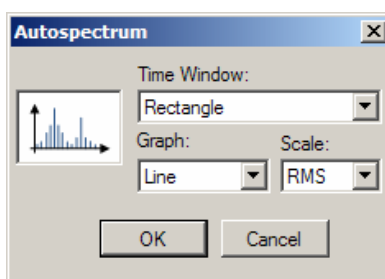
Dále byla v nástroji Autospectrum doprogramována i časová okna. Do této skupiny patří:

- Rectangular
- Hanning
- Flat Top
- Kaiser Bessel

Jako názorná ukázkou bylo zvoleno okno Hanning, které je popsáno vztahem 3.2, kde k je index a n je délka výstupního vektoru FFTW:

$$w(k) = 1 - \cos\left(\frac{2\pi k}{n}\right) \quad (2.2)$$

```
Public Function HanningWindow(ByVal x As Long, _
                               ByVal Length As Long) As Double
  Return (1 - Math.Cos((2 * Math.PI * x) / Length))
End Function
```



Obr. 2.7 Dialogové okno pro nastavení Nástroje Autospectrum

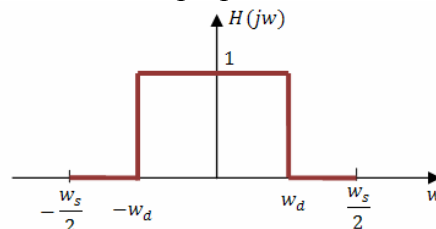
2.4 Digitální filtry

Rozeznáváme dva druhy digitálních filtrů, s nekonečnou odezvou IIR a s konečnou odezvou FIR, kterým dávám přednost. Filtry typu FIR představují typ, jenž nemá analogii u

analogových filtrů. Jsou určeny svou přenosovou funkcí, která má M – násobný pól v nule, jsou tedy nezávislé na koeficientech a vždy stabilní. Přenosová funkce FIR filtru má následující tvar: [Tůma 2009]

$$H(z) = b_0 + b_1 z^{-1} + \dots + b_{M-1} z^{-M+1} + b_M z^{-M} \quad (2.3)$$

Návrh filtru vychází z ideálního dolnoproústného filtru znázorněného na obr. 2.8.



Obr. 2.8 Zadání ideálního dolnoproústného filtru

Nejdříve tedy potřebujeme získat koeficienty přenosové funkce. Jako příklad návrhu poslouží dolnoproústný filtr, který lze definovat frekvenční přenosovou funkcí [Tůma 2009]

$$H(w) = \begin{cases} 1; & -w_d \leq w \leq w_d \\ 0; & \frac{w_s}{2} > w > w_d, -\frac{w_s}{2} < w < -w_d \end{cases} \quad (2.4)$$

Je také třeba uvažovat frekvenční omezení měření podle Shannon – Kotelnikova teorému:

$$\frac{f_s = w_s}{2\pi} \quad (2.5)$$

Zpětnou transformací frekvenční přenosové funkce pro omezený rozsah úhlové frekvence do poloviny kladné a záporné vzorkovací frekvence je tedy impulzní odezva a platí: [Tůma 2009]

$$h_n = \frac{1}{2\pi} + \int_{-\frac{w_s}{2}}^{\frac{w_s}{2}} H(jw) \exp(jwnT_s) dw = \frac{\sin(2\pi n f_d T_s)}{\pi n}; n = 0, \pm 1, \pm 2 \dots \quad (2.6)$$

Filtr je nekauzální, což znamená, že odezva předchází vstupní impuls. Pro dodatečné zpracování naměřených není nutné na kauzalitě filtru trvat, protože známe předcházející vstupní i výstupní vzorky signálu. Rovněž vzhledem k zmenšující se absolutní hodnoty koeficientů, vzhledem k rostoucímu n je možné zkrátit odezvu na vybraný počet koeficientů tak, že vznikne FIR filtr. Počet koeficientu je roven řádu filtru (`FilterOdder`). Z předešlého vzorce lze již odvodit následující algoritmus pro návrh koeficientů filtru dolní propust, přičemž obdobným způsobem bylo postupováno i u dalších filtrů:

```

ReDim FilterCoef((FilterOdder - 1))
m = (FilterOdder - 1) \ 2
Wcutoff = Math.PI * f
For n = -m To m
  If n = 0 Then
    FilterCoef(n + m) = Wcutoff / Math.PI
  Else
    FilterCoef(n + m) = Math.Sin(Wcutoff * n) / (Math.PI * n)
  End If
Next n

```

V aplikaci je tedy možné použít těchto filtrů dolní, horní, pásmová propust a filtr pro Hilbertovu transformaci.

```

Dim Offset = UBound(FilterCoef) \ 2
For VectorIndex As Integer = 0 To UBound(DataVector)
  Dim FilterIndex = UBound(FilterCoef)
  For i As Integer = VectorIndex - Offset To VectorIndex + Offset
    If (i >= 0 And i <= UBound(DataVector)) Then
      OutVector(VectorIndex) += FilterCoef(FilterIndex) * DataVector(i)
    End If
  Next i
Next VectorIndex

```

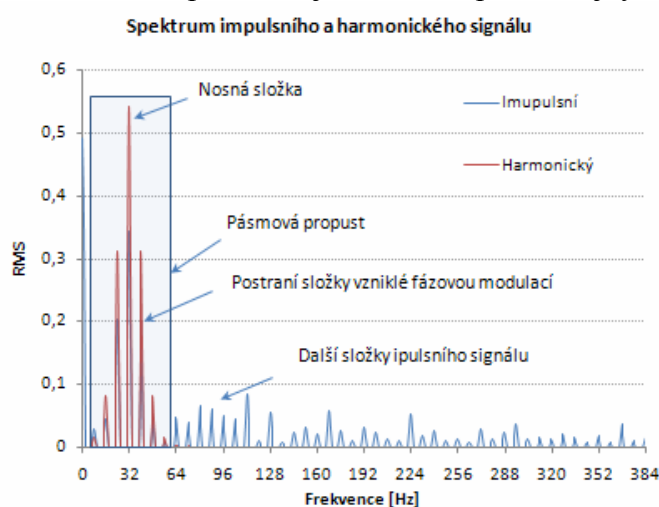
```

FilterIndex -= 1
Next i
Next VectorIndex

```

2.5 Rovnoměrnost otáčení (Rotation Uniformity)

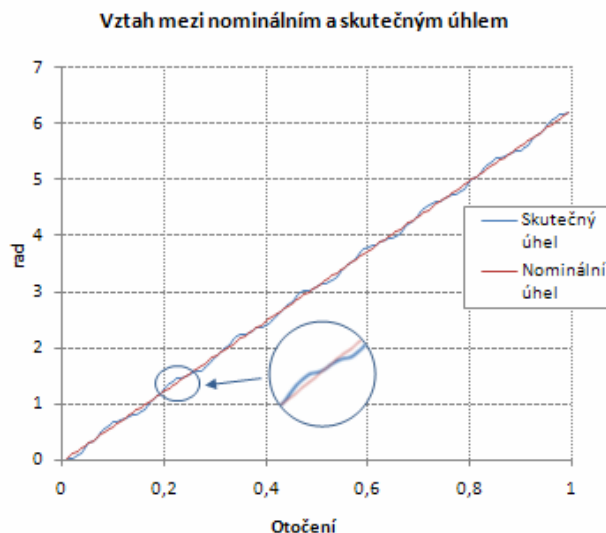
Tento nástroj je založen na metodě fázové demodulace. Postup zpracování signálu je tedy následující. Při otáčení obdržíme na snímači impulsní signál s tvarem blízkým obdélníku. V případě rovnoměrného pohybu (rychlost otáčení je konstantní) tento signál obsahuje harmonické složky příslušející složce o základní frekvenci impulsu. Tyto složky se nazývají nosnými. Při jakékoli nepatrné změně rychlosti poté dochází k fázové modulaci tohoto signálu, která se projeví vznikem postraních složek ke složkám nosným. Jako nosnou složku je dobré vybrat složku se základní frekvencí impulsů. Ta je v daném spektru nejvyšší. [Tůma 2002]



Obr. 2.9 Spektrum frekvenčně modulovaného impulsního a harmonického signálu

Výběr je proveden tak, že se za pomoci filtru pásmové propusti odstraní všechny složky spektra s výjimkou nosné složky a jejího postranního pásma. Je velmi vhodné, aby se šířka horního a dolního pásma rovnala polovině základní frekvence. Na této myšlence je založeno automatické nastavování filtru v tomto nástroji. Po odstranění nadbytečných složek spektra získáme harmonický fázově modulovaný signál. Základem nejen pro fázovou, ale i pro amplitudovou demodulaci signálu je teorie analytického signálu: [Tůma 2002]

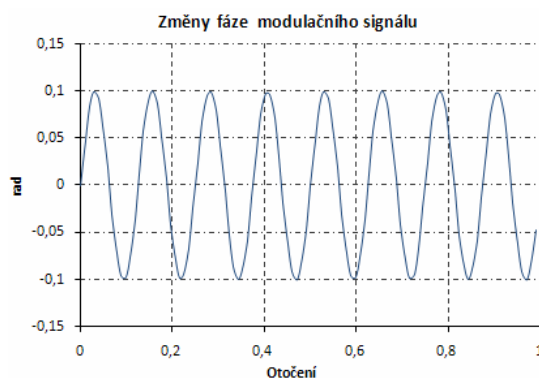
$$z(t) = x(t) + jy(t) = |z(t)| \exp(j\varphi(t)) \quad (2.7)$$



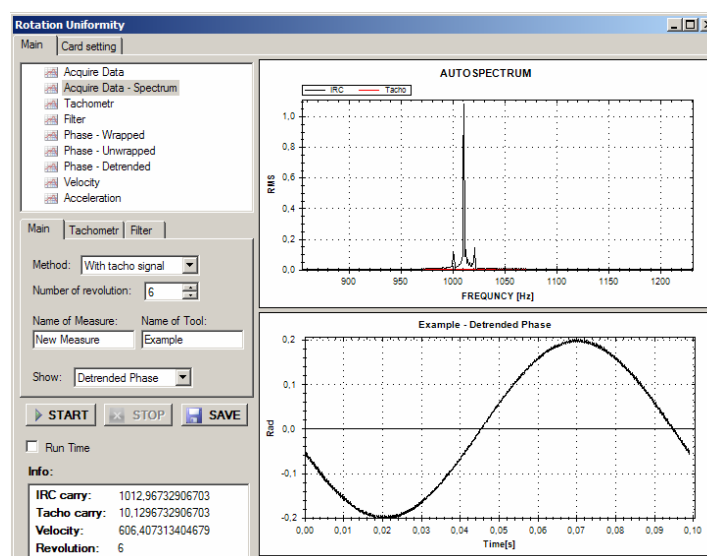
Obr. 2.10 Vztah mezi nominálním a skutečným úhlem natočení

Předchozími úpravami jsme získali jeho reálnou část $x(t)$. Imaginární část $y(t)$ získáme z reálné složky pomocí filtru pro Hilbertovu transformaci. Pomocí funkce arkustangens lze dále z individuálních hodnot reálné a imaginární složky určit hlavní hodnoty fáze a to v intervalech od $-\pi$ do π .

$$ARG\{z_t\} = \arctan\left(\frac{y_t}{x_t}\right), \quad t = 0, 1, 2, \dots, n-1 \quad (2.8)$$

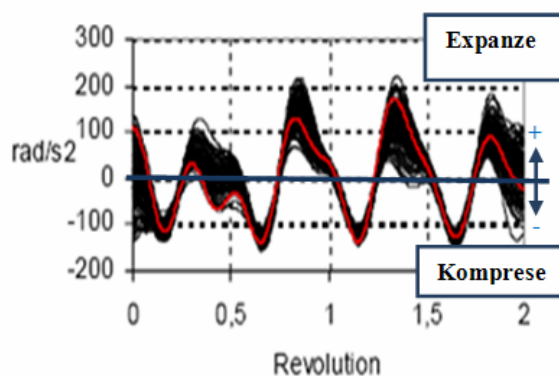
Obr. 2.11 průběh odchylek od lineární funkce $w_{\theta}t$

Závislost hlavních hodnot fáze na čase obsahuje při dosažení hodnoty $\pm\pi$ body nespojitosti. Jedná se tedy o tzv. zabalenou (wrap) fázi. Pokud však posuneme hlavní body fáze o celočíselný násobek π , získáme fázi (unwrap) rozbalenou. Názorná ukázka rozbalené normované fáze je prezentována na obr. 2.10. Odstraněním lineárně narůstající složky v rozbalené normované fázi, která odpovídá konstantní úhlové rychlosti, získáme průběh odchylek od lineární funkce $w_{\theta}t$, jejíž fyzikální rozměr je úhlová výchylka viz obr. 2.11. Pro hodnocení okamžité rychlosti je nutné úhlovou výchylku derivovat podle času. A pro určení zrychlení je potřeba dvojnásobné derivace. Derivace je v naší aplikaci řešena jako rozdíl aktuální hodnoty a hodnoty jí předešlé násobené převrácenou hodnotou délky časového intervalu mezi těmito vzorky. [Tůma 2002]



Obr. 2.12 Formulář nástroje Rotation Uniformity

Formulář nástroje Rotation uniformity je rozdělen na dvě záložky. Po nastavení vlastností měřicí karty v záložce druhé uživatel přejde do záložky hlavní. Zde je realizováno měření. Místo požadovaného počtu vzorků v tomto nástroji zadáváme požadovaný počet otáček. Po dosažení poslední otáčky je měření ukončeno a přechází se na zpracování signálu podle výše uvedeného postupu. Formulář obsahuje dva grafy, což umožňuje uživateli pozorovat zároveň mezivýsledek i konečné výsledky. Výsledná data jsou vykreslena do grafu po otáčkách, což je výhodné zvláště u diagnostiky spalovacích motorů viz obr. 2.13. Volba grafu znázorňující kolísání úhlového zrychlení byla záměrná, protože úhlové zrychlení má nejvyšší vypovídací schopnost o rovnoměrnosti otáčení, protože jej lze přímo spojit s hnacím momentem stroje.



Obr. 2.13 Průběh kolísání úhlového zrychlení zážehového motoru automobilu [Tůma a spol. 2002]

Závěr

Předložená práce obsahuje popis softwaru pro fázovou demodulaci, který je užít k hodnocení úhlových kmitů za rotace. Ve výpočtech je použita teorie analytického signálu vyžadující Hilbertovu transformaci. K výpočtu Hilbertovy transformace je použit digitální filtr. Software obsahuje další pomocné funkce jako je vyhodnocení rychlosti otáčení na základě vyhodnocení vzdálenosti impulsu generovaných tachosondou nebo jiným zařízením a kontrolu autospekter signálů. Pro tvorbu softwaru bylo zvoleno prostředí Microsoft Visual Studio, programovací jazyk Visual Basic a platforma Microsoft .NET Framework.

Použitá literatura

KREIDL M., ŠMÍD R. *Technická diagnostika - senzory, metody, analýza signálu*, Praha: BEN, 30.11.2006, s.408. ISBN 80-7300-158-6

TŮMA, J. *Měření okamžitých otáček úhlovým snímačem otočení*. In *Technická diagnostika strojů a výrobních zařízení „DIAGO 2002“*, Ostrava : VŠB-TU Ostrava, 30. – 31. leden 2001. pp. 458-464. ISBN 80-248-0045-4.

TŮMA, J. *Phase demodulation in angular vibration measurements*. In *Proceedings of 3rd International Carpathian Control Conference*. Ostrava : VŠB-TU Ostrava, 27. - 30. 5. 2002, s. 99-104. ISBN 80-248-0089-6.

TŮMA, J., KOČÍ, P., ŠKUTA, J., JURÁK, M. *Analysis of a car burst shaking while the car engine is running at idle*. In *Proceedings of 3rd International Carpathian Control Conference*. Ostrava : VŠB-TU Ostrava, 27. - 30. 5. 2002, s. 105-110. ISBN 80-248-0089-6.

TŮMA, J. *Digital filter*, [on line]. [cit. 2009-04-1] Available from world wide web: <http://fs1.vsb.cz/~tum52/SignalAnalyser/Prezentace/Filtry_EN.pdf>

TŮMA, J. *Autospectrum*, [on line]. [cit. 2009-04-5] Available from world wide web: <<http://fs1.vsb.cz/~tum52/SignalAnalyser/Prezentace/Lecture4-Autospectrum.pdf>>