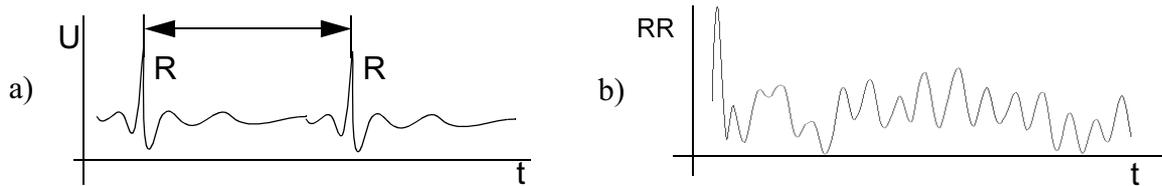


PROSTŘEDÍ PRO ZPRACOVÁNÍ BIOLOGICKÝCH SIGNÁLŮ

Adam Příbyl

ČVUT FEL, Katedra kybernetiky

Úkolem stanoveným pro mou úlohu bylo vytvoření především filtračního algoritmu pro použití v oblasti biomedicínských signálů. Konkrétně se jednalo o filtraci respirační sinusové aritmie ze signálů EKG, přesněji ze signálů RR intervalů (RR interval je vzálenost mezi R vlnami EKG signálu, což odpovídá tepové frekvenci), jak je znázorněno na dalším obrázku obr. 1 .

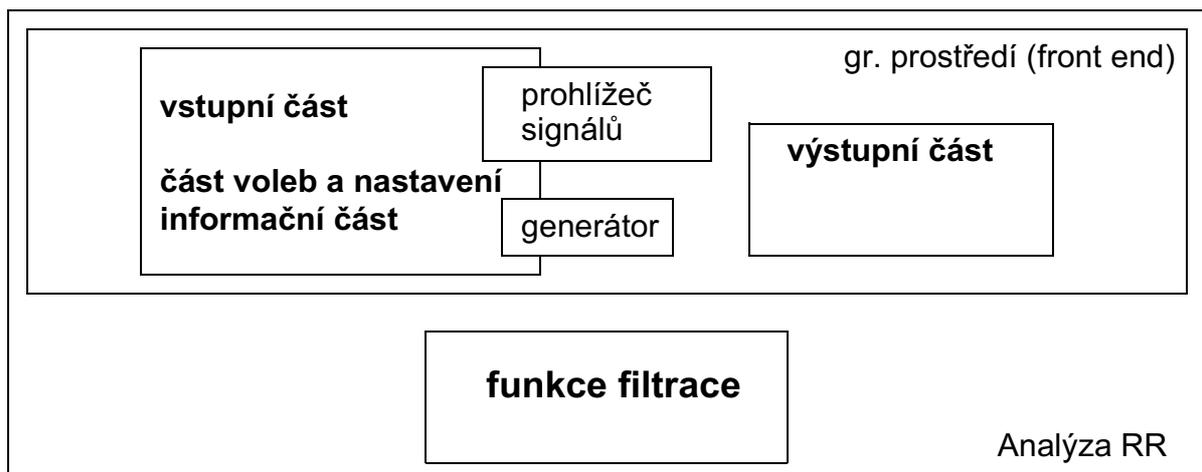


Obr. 1: a) EKG s R vlnami, b) signál RR intervalů

Pro tento účel byly vybrány algoritmy adaptivní filtrace a to regresní analýza a metoda nejmenších čtverců známá spíše jako LMS.

Vzhledem k tomu, že se jednalo o práci pro 3.lékařskou fakultu University Karlovy a bylo potřeba, aby filtrace byla přístupná i pro případné pracovníky nezabývající se programováním v Matlabu, a také z důvodu zpracování většího objemu dat bylo potřeba navrhnout rozumnou strukturu prostředí a přístupu k filtraci.

Protože jsem trochu unixově (open source) smýšlející člověk a při tvorbě jsem občas přebíhal mezi systémy, i struktura, kterou jsem dal svému prostředí, je do značné míry podobná modelu známému z těchto OS. Základní struktura tedy vypadá asi následovně (viz. obr. 2).



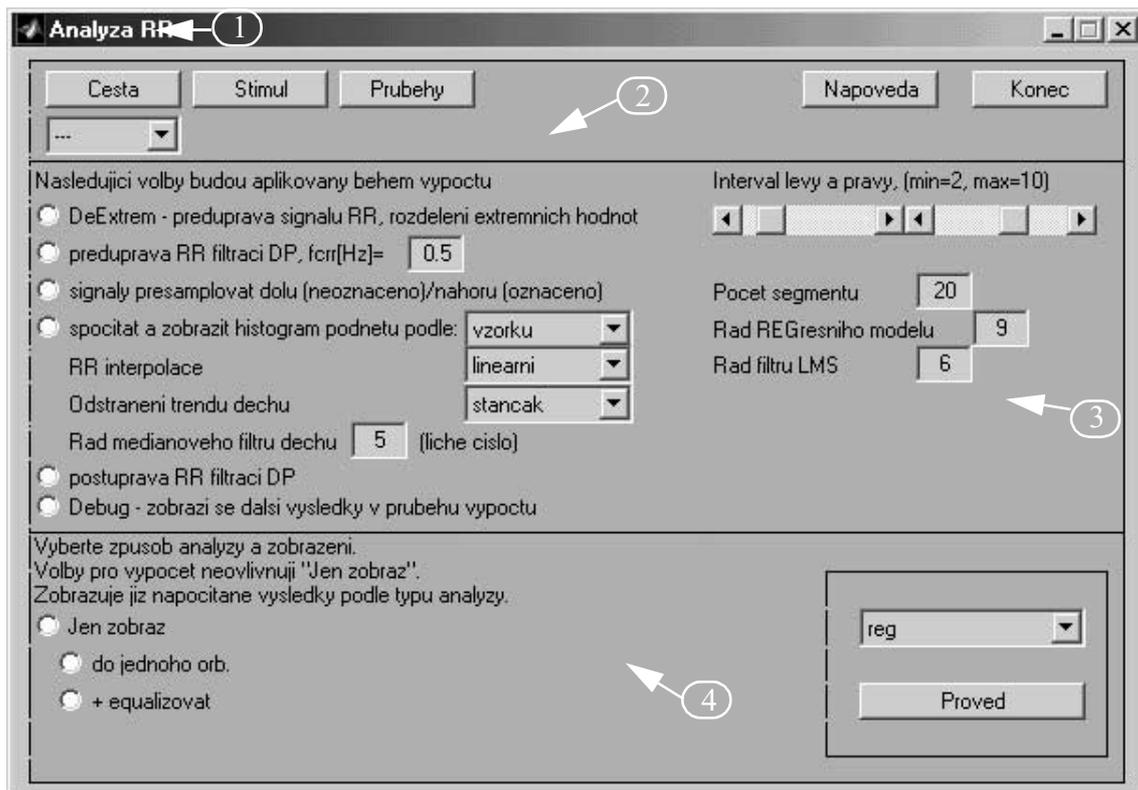
Obr. 2: Struktura prostředí

Funkce filtrace je vlastní výkonná část umožňující přímý vstup a přímý výstup výsledků pomocí vstupně výstupních parametrů (proměnných) a zjednodušený grafický výstup - takto byla vytvořena z důvodu možného samostatného použití v jiných aplikacích.

Pod grafické prostředí (front end) je zahrnována vstupní část (ta může být realizována i ne-grafickým vstupem - malým pomocným programkem, který nastavuje parametry filtrace pomocí ručně editovaného konfiguračního souboru). Součástí vstupní části je i prohlížeč signálů a generátor umělých signálů pro testování parametrů filtrace. Druhou částí je funkce grafického vstupu, která obstarává zobrazení výsledků podle nastavených parametrů.

Osobně si myslím, že to je jeden z nejlepších modelů, které se pro tento typ aplikací hodí, a je pro něj velmi dobře využitelný dostupný potenciál Matlabu.

Grafický vstupní interface pak vypadá následovně (obr. 3).



Obr. 3: Vstupní část grafického rozhraní

Základní rozdělení obrazovky je na 4 části:

1. je informační lišta, která i během analýzy zobrazuje právě probíhající akce.
2. ve které se zobrazují informace o vybrané skupině souborů pro analýzu a nacházejí se zde ovládací prvky pro spuštění podprogramů pro prohlížení průběhů signálů a generátoru modelových signálů
3. slouží k výběru parametrů analýzy
4. je oblast obsahující volby zobrazení výsledků a její důležitou podčástí je výběr a spuštění analýzy

Procedura zpracování

Jak je vidět z možností grafického prostředí, nastavují se zde parametry pro dva typy analýzy (filtrace) - regresní analýzu a LMS (metoda nejmenších čtverců) - a parametry pro předúpravu signálů, především RR intervalů. Do analýzy však vstupují ještě další dva signály - signál dechu (většinou označovaný jako RESP) a signál podnětů, kterým se však nebudeme zabývat a má souvislost s dalším úkolem vážícím se k této filtraci. V následujících odstavcích pak popíšete svou implementaci algoritmů obou analýz.

Regresní analýza - implementace Matlab

Implementaci popíši jen velmi stručně, případné hlubší pochopení je nutné hledat v matematickém aparátu vážícím se k regresnímu algoritmu v literatuře.

Při implementaci v Matlabu musíme tedy nejdříve splnit podmínku stacionárního a ergodického signálu. Toho dosáhneme segmentováním

```
1      d=(rr-Stredni); %odecti od RR stredni hodnotu
1      for k=1:PocSegment-1
2          % Vyriznuti segmentu, prekryv delky m
3          resp1=resp((k-1)*N+1:k*N+m);
4          r1=d((k-1)*N+1:k*N+m);
5          uu=resp1; vv=r1;
```

Nyní podle řádu modelu spočteme postupně kovarianční funkce.

```
6          % Spocti potrebne kovarianční funkce
7          Kxx=[];
8          Kxy=[];
9          for kk=1:m+1
10             Kxx(kk)=(uu(m+1:N+m-kk+1)'*uu(m+kk:N+m))/N;
11             Kxy(kk)=(uu(m+1:N+m-kk+1)'*vv(m+kk:N+m))/N;
12          end
```

Vyřešíme soustavu rovnic.

```
13         % Spocti g (vyres soustavu rovnic)
14         xxx=toeplitz(Kxx);
15         g(:,k)=xxx\Kxy';
```

A odečteme nastřádanou RSA od RR.

```
16         % Ocisti RR-intervalu od RSA
17         for ll=m+1:N+m
18             rsa(ll,k)=g(:,k)*flipud(resp1(ll-m:ll));
19             r(ll,k)=r1(ll)-rsa(ll,k);
20         end
21     end
```

Protože dosud jsme pracovali s maticí, která měla sloupce délky N (N je délka segmentu) a my potřebujeme dostat opět kontinuální signál původní délky bez přechodových dějů modelu, sloučíme sloupce do jednoho vektoru.

```
22         % Slouci se RR-RSA v matici r do jednoho vektoru r2
23         r2=[];
24         for k=1:PocSegment-1
25             tmp=r(m+1:N+m,k);
26             r2=[r2 tmp'];
27         end
28         r2=[zeros(1,m+1) r2]';
```

To je tedy stručný popis algoritmu tak, jak je ho možné najít ve zdrojových kódech.

LMS - implementace Matlab

```
1      for k=1+M:N
```

Pouze převrátíme sloupcový vektor *resp* podél vodorovné osy.

```
2          u=flipud(resp(k-M:k-1));
```

Provedeme vynásobení vstupního signálu *u* koeficienty filtru a jejich sečtení. (Násobíme vektor řádkový, vektorem sloupcovým.)

```
3          % 1. Filtrace (adaptivni)
4          dodhad(k)=u'*wodhad(:,k-1);
```

Přičteme (fakticky odečteme) odhadnutý signál *dodhad* k původnímu signálu RR= *d*.

```
5          e(k)=d(k)+dodhad(k);
```

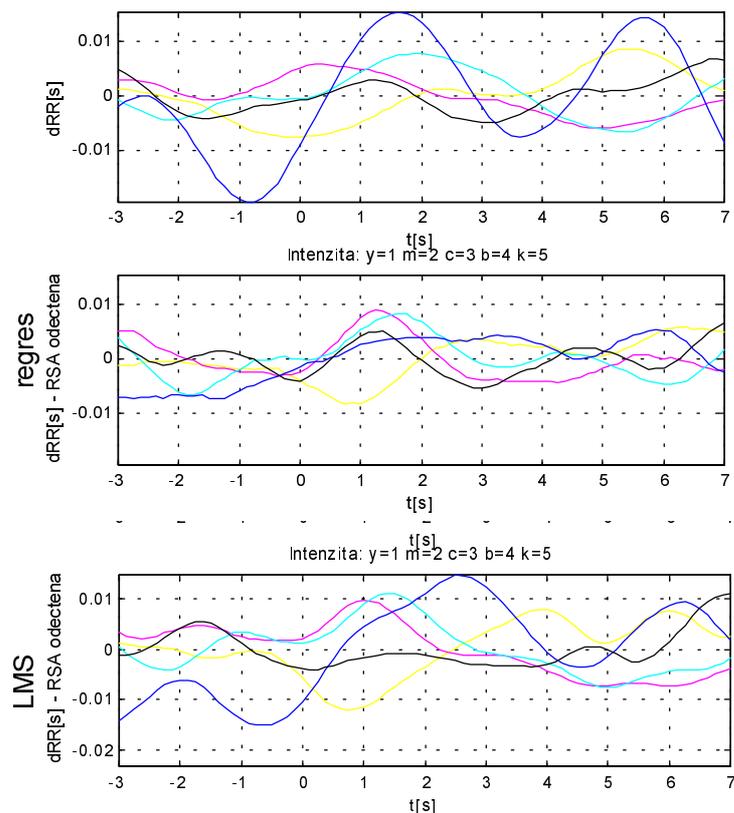
A na základě LMS algoritmu odhadneme koeficienty pro další cyklus.

```
6          % 2. Aktualizace
7          wodhad(:,k)=wodhad(:,k-1)-mi*e(k).*u;
8          end
9          e=e';
```

Teoreticky by nám aplikací tohoto algoritmu měl v \bar{e} zůstat signál očištěný od šumu RSA, který do signálu RR intervalů vnáší dech, a signál \bar{e} by měl tedy obsahovat pouze změny vyvolané dalšími činnostmi organismu

Účinnost těchto metod

Podle dostupných experimentálních výsledků se ukázalo, že použití LMS nepřineslo očekávané zlepšení oproti regresní analýze, a tak regrese zůstává jediným algoritmem, který bych si troufl pro tento typ zpracování doporučit. Jeden z příkladných výsledků je možno vidět na obrázku obr. 4. Na signálu, zvláště u regresní analýzy, je dobře vidět sklon ke snižování frekvence tepu po čase $t=0$. K popisu je potřeba dodat, že se jedná o signál průměrovaný přes mnoho period za účelem získání odezev na konkrétní podnět přichází právě v čase $t=0$. Průměrování je použito z důvodu sledování extrémně malých změn v signálu.



Obr. 4: Výsledek po aplikování filtrací

Doufám, že pro případného zájemce o tuto problematiku bude tento stručný článek, který měl sloužit jako konkrétní příklad implementace regresní analýzy, LMS a grafického prostředí v Matlabu, podnětný a pomůže mu při základech implementace těchto typů analýz a při rozhodování o struktuře psaných aplikací.

Ing. Adam Příbyl, xpribyla@cs.felk.cvut.cz, katedra kybernetiky ČVUT FEL, Karlovo nám 13, Praha 2

Literatura:

- [1] Příbyl Adam; Číslíkové metody odhadu vyvolané tepové odpovědi; Diplomová práce, ČVUT FEL 2002
- [2] Uhlíř Jan, Sovka Pavel; Číslíkové zpracování signálů; skriptum ČVUT 1995
- [3] Matlab - referenční příručka a online dokumentace