

Detekcija QRS kompleksov

Projektna naloga pri OBSS

Zala Štebljaj

Povzetek

V poročilu je predstavljen real-time algoritem za detekcijo QRS kompleksov, ki sem ga implementirala na podlagi članka *A Moving Average based Filtering System with its Application to Real-time QRS Detection* avtorjev H. C. Chena in S. W. Chena in je sestavljen iz dveh faz filtriranja in faze odločanja. Dodane so tudi nekatere izboljšave, delovanje algoritma pa je preverjeno na Long-Term ST database bazi posnetkov in da zelo zadovoljive rezultate. Podane so tudi pomankljivosti in možne izboljšave algoritma.

1 Uvod

Cilj te projektne naloge je bil na podlagi izbranega članka izdelati detektor QRS kompleksov. Izbrala sem članek [3] z naslovom *A Moving Average based Filtering System with its Application to Real-time QRS Detection* avtorjev H. C. Chena in S. W. Chena. Članek predlaga, da signal najprej procesiramo z linearnim visokoprepustnim filtrom, nato z nelinearnim nizkoprepustnim filtrom, na koncu pa s prilagodljivo mejo določimo, kateri izmed vrhov procesiranega signala so dovolj visoki, da predstavljajo dejanske srčne utripe. S procesiranjem želimo odstraniti nepomembne dele signala, kot so na primer šumi, P in T valovi ter napake, ki nastanejo zaradi premikanja elektrod, da lažje določimo utripe v fazi odločanja. V nadaljevanju bom predstavila, kako sem algoritem implementirala, njegove rezultate na bazi posnetkov Long-Term ST database [2], ki jo lahko najdemo na strani PhysioNeta [1], ter možne izboljšave. Kot zahteva članek, je to real-time algoritem, naredila pa sem ga v programskem jeziku MATLAB.

2 Metode

Implementirani detektor je sestavljen iz treh faz; najprej signal procesiramo z linearnim visokoprepustnim filtrom, nato z nelinearnim nizkoprepustnim filtrom, na koncu pa sledi še faza odločanja, v kateri s spremenljivo mejo določimo dejanska mesta utripov.

Najprej torej uporabimo linearnen visokoprepusten filter, s katerim skušamo poudariti QRS komplekse in zatreti nizke frekvence, kot so P in T valovi in nihanje osnovne linije. To naredimo z M-točkovnim filtrom s tekočim povprečjem

$$y_1(n) = \frac{1}{M} \sum_{m=0}^{M-1} x(n-m)$$

in filtrom, s katerim signal zakasnimo za $\frac{M+1}{2}$ vzorcev

$$y_2(n) = x\left(n - \frac{M+1}{2}\right),$$

pri čemer $x(n)$ predstavlja trenutni vhod. M je stopnja filtra, ki jo omejimo na lihe vrednosti, avtorja članka priporočata $M = 5$ ali $M = 7$ (v svoji implementaciji sem izbrala $M = 7$). Končni linearni filter dobimo z odštevanjem:

$$y(n) = y_2(n) - y_1(n).$$

Signal nato naprej procesiramo z nelinearnim nizkoprepustnim filtrom, da ga zgladimo, pri tem pa še vedno ohranimo izrazite vrhove, ki ustrezajo lokacijam QRS kompleksov. Ta filter je sestavljen iz

zaporednega kvadriranja po točkah ter seštevanja elementov v pomičnem oknu. Širina okna naj bi bila izbrana glede na vzorčno frekvenco tako, da ustreza približno 150 ms. Baza posnetkov, ki sem jo izbrala, ima frekvenco vzorčenja 250 Hz, zato sem za širino okna vzela 38 vzorcev.

Tako sprossen signal ima izrazite vrhove na lokacijah QRS kompleksov, zato lahko konstruiramo spremenljivo mejo, ki nam pomaga določiti, kateri vrhovi res predstavljajo srčni utrip. Ta meja je podana z enačbo

$$T(i) = \alpha \cdot \gamma \cdot V + (1 - \alpha) \cdot T(i - 1).$$

Pri tem je V trenutni vrh (ki smo ga ravno določili in predstavlja srčni utrip), α je faktor pozabljanja ($0 \leq \alpha \leq 1$), γ pa utež, ki določa prispevek trenutnega vrha k naslednji meji. V moji implementaciji sta dobre rezultate dali $\alpha = 0.05$ in $\gamma = 0.2$, dva izmed predlogov za konstanti avtorjev članka.

Osnovna ideja je, da za dejanski srčni utrip vzamemo le vrhove, ki presežejo to mejo, vendar pa se izkaže, da ta pristop v nekaterih primerih odpove, zato sem dodala še nekatere izboljšave. Metoda je implementirana s premikajočim se oknom, v katerem najdemo maksimum in najprej preverimo, ali je nad mejo in ali je boljši od prejšnjega zaznanega. Z zaznanim najboljšim vrhom nad mejo, ki naj bi predstavljal srčni utrip, potem čakamo, da je maksimum v naslednjih oknih nižji od 0.9 meje. S tem preprečimo dodajanje veliko vrhov, če procesirani signal oscilira okoli meje. Včasih pa se zgodi, da je meja tako nizka, da maksimumi v naslednjih oknih nikoli ne bi prišli pod 0.9 meje. V ta namen dodamo števec, ki šteje, kolikokrat zaporedno je maksimum okna že nižji od zadnjega najvišjega vrha. Ko dosežemo določeno število, ta najvišji vrh proglasimo za srčni utrip in posodobimo mejo. Še en velik problem pa nastane, če je originalni signal tako razgiban, da dobimo zares visoke špice v procesiranem signalu. To privede do tega, da je tudi meja ekstremno visoka, in na ta način ne bi zaznali prav nobenega vrha, saj nobeden ne pride čez mejo. Kot rešitev uvedemo še en števec, ki šteje, koliko oken smo že pregledali, ne da bi prešli to visoko mejo. Ko je takih oken dovolj veliko, mejo ponastavimo tako, da upoštevamo le nazadnje določen vrh V , torej $T(i) = \alpha \cdot \gamma \cdot V$. S tem sicer lahko izgubimo nekaj pravih srčnih utripov, vendar pa je to vseeno bolje, kot pa da ne zaznamo nobenega več.

3 Rezultati

Svoj algoritem sem preizkusila na bazi posnetkov Long-Term ST database, v kateri je 86 posnetkov. Kot utrip mi je pravilno zaznal 8864057 utripov ($8791508 + 71970 + 579$), dodatno je zaznal 75706 utripov, ki jih v resnici ni, ni pa zaznal 33723 ($32749 + 953 + 21$) utripov, ki bi jih moral. S temi podatki lahko izračunamo občutljivost in pozitivno prediktivnost:

$$Se = \frac{TP}{TP + FN} = \frac{8864057}{8864057 + 33723} = 0.9962 = 99.62\%$$

$$+P = \frac{TP}{TP + FP} = \frac{8864057}{8864057 + 75706} = 0.9915 = 99.15\%$$

Vidimo lahko, da je algoritem po obeh merilih zelo dober.

4 Diskusija

Na preizkušanih posnetkih se je moj algoritem kar dobro izkazal. Tako občutljivost kot pozitivna prediktivnost sta zelo visoki, kar nam pove, da je zaznal veliko pravih srčnih utripov in ni zaznal preveč dodatnih utripov, ki jih v resnici ni. Seveda je število posnetkov v bazi omejeno, zato bi lahko algoritem preizkusili še na kakšnem drugem posnetku, morda bolj problematičnem, in poskusili ugotoviti, kje srčnih utripov še ne detektira tako dobro. Pomanjkljivost vidim tudi v tem, da je faza odločanja narejena precej eksperimentalno in so v njej neke konstante, ki morda pri drugih posnetkih (na primer z drugo frekvenco vzorčenja) ne bi tako dobro delovale. Da pa bi lahko določila, kako jih je treba nastaviti glede na vhodni posnetek, pa bi spet potrebovala še več različnih posnetkov in ne le iz ene baze.

Literatura

- [1] A. Goldberger, L. Amaral, L. Glass, J. Hausdorff, P. C. Ivanov, R. G. Mark, J. E. Mietus, G. B. Moody, C. K. Peng in H. E. Stanley, *PhysioBank, PhysioToolkit, and PhysioNet: Components of a new research resource for complex physiologic signals*, Circulation [Online], **101**(23) (2000) 215–220.
- [2] F. Jager, A. Taddei, G. B. Moody, M. Emdin, G. Antolic, R. Dorn, A. Smrdel, C. Marchesi in R. G. Mark, *Long-term ST database: a reference for the development and evaluation of automated ischaemia detectors and for the study of the dynamics of myocardial ischaemia*, Medical & Biological Engineering & Computing **41**(2) (2003) 172-183.
- [3] H. C. Chen in S. W. Chen, *A Moving Average based Filtering System with its Application to Real-time QRS Detection*, v: Computers in Cardiology 2003 (ur. Alan Murray, PhD), IEEE, Thessaloniki Chalkidik, 2003, 585-588.