

SVEUČILIŠTE U ZAGREBU  
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

DIPLOMSKI RAD br. 1076

## **Algoritmi za detekciju umora vozača**

Zvonimir Mandić

Zagreb, lipnja 2015

Zagreb, 6. ožujka 2015.

## DIPLOMSKI ZADATAK br. 1076

Pristupnik: **Zvonimir Mandić (0036457934)**  
Studij: Računarstvo  
Profil: Računalno inženjerstvo

Zadatak: **Algoritmi za detekciju umora vozača**

### Opis zadatka:

U okviru diplomskog rada potrebno je proučiti algoritme za detekciju umora vozača. Potrebno je razmotriti sve fiziološke pojave iz kojih se može detektirati umor i istražiti algoritme s različitim metodama detekcije te odrediti referentan s najvećom točnošću. Posebnu pažnju potrebno je posvetiti algoritmima koji detekciju baziraju na značajkama lica vozača. Potrebno je istražiti dostupne označene baze slika sa značajkama lica te ih testirati na gotovim algoritmima. Razviti vlastiti algoritam u Matlab okruženju temeljen na odabranoj fiziološkoj pojavi. Razvijeni algoritam testirati na označenim bazama slika i usporediti rezultate s referentnim algoritmom.

Zadatak uručen pristupniku: 13. ožujka 2015.  
Rok za predaju rada: 30. lipnja 2015.

Mentor:  


---

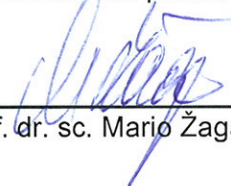
Prof. dr. sc. Davor Petrinović

Djelovođa:  


---

Prof. dr. sc. Danko Basch

Predsjednik odbora za  
diplomski rad profila:



---

Prof. dr. sc. Mario Žagar

Zahvaljujem se prof. dc. sc. Davoru Petrinoviću na mentorstvu

# Sadržaj

1. Uvod .....	1
2. Fiziološke pojave uzrokovane umorom.....	2
3. Sustavi i algoritmi za detekciju umora.....	5
3.1. Detekcija prvog stadija sna iz EEG-a .....	5
3.2. Detekcija umora analizom pritiska na volan .....	7
3.3. Detekcija umora analizom više karakteristika lica .....	10
3.3.1. Detekcija i praćenje oka .....	10
3.3.2. Parametri praćenja pomaka kapka .....	12
3.3.3. Estimacija orijentacije glave .....	13
3.3.4. Praćenje smjera pogleda .....	14
3.3.5. Analiza izraza lica.....	16
3.3.6. Modeliranje umora pomoću Bayesovih mreža.....	17
4. Postojeći sustavi za detekciju umora .....	22
4.1. Audi.....	22
4.2. BMW .....	22
4.3. Volkswagen.....	22
4.4. Bosch .....	23
4.5. Mercedes-Benz.....	23
5. Programska potpora i korišteni hardver .....	24
5.1. Zedboard.....	24
5.2. Matlab .....	25
5.3. ModelSim .....	26
5.4. Xilinx Vivado .....	26
6. Implementacija .....	27
6.1. Računanje udaljenosti.....	28
6.2. Normalizacija i skaliranje.....	31
6.3. Filtar .....	32
6.4. Detekcija zijevanja .....	37
6.4.1. Detektor lokalnog maksimuma .....	38
6.4.2. Detektor zijevanja .....	40
6.5. Brojač.....	41
6.6. Frekvencija zijevanja i detekcija umora.....	42
6.7. Integracija u IP jezgru .....	44
7. Zaključak .....	46

8. Literatura .....	47
Naslov .....	49
Sažetak .....	49
Ključne riječi .....	49
Title .....	50
Abstract .....	50
Keywords .....	50

# 1. Uvod

Razni su uzroci zbog kojih se na cestama događaju prometne nesreće. Brzina i alkohol glavni uzroci su većine nesreća. No vrlo podcijenjen faktor kod vozača je umor. Nesreće uzrokovane umorom vrlo često dovode do ozbiljnih posljedica jer vozač ne može pravovremeno reagirati i u sekundi može sletjeti s ceste ili udariti u drugo vozilo. Mladi i neiskusni vozači više su podložni umaranju jer ulažu više energije i koncentracije u vožnji od iskusnijih vozača. Osim fiziološkog gubitka energije, umor izaziva i monotonija vožnje, usamljenost, vožnja u nepovoljnim uvjetima,...

Sprječavanju takvih nesreća može se doskočiti ugradnjom sustava za detekciju umora. Takvi sustavi „promatraju“ vozača i na temelju njegovog fiziološkog stanja, odnosno razine umora, predlažu vozaču da se odmori ili zvučnim signalima upozoravaju vozača da obrati pozornost na cestu. Neke od fizioloških pojava iz kojih se može zaključiti je li vozač umoran su položaj glave, zatvorenost kapaka, učestalost zijevanja i puls. Kao referentni sustav za detekciju umora uzima se sustav koji koristi EEG (elektroencefalograf). Korištenje takvog sustava je nepraktično u stvarnom životu tako da se on koristi za usporednu provjeru ostalih. Kao sredstvo za dobivanje informacija o vozaču najčešće se koristi kamera i razni taktilni senzori na volanima.

Cilj diplomskog rada je proučiti dostupne algoritme za detekciju umora i fiziološke pojave na kojima se oni temelje te napraviti vlastiti sustav za detekciju koji detektira umor iz frekvencije zijevanja. Posebni naglasak će biti na algoritmima koji koriste kameru za detekciju.

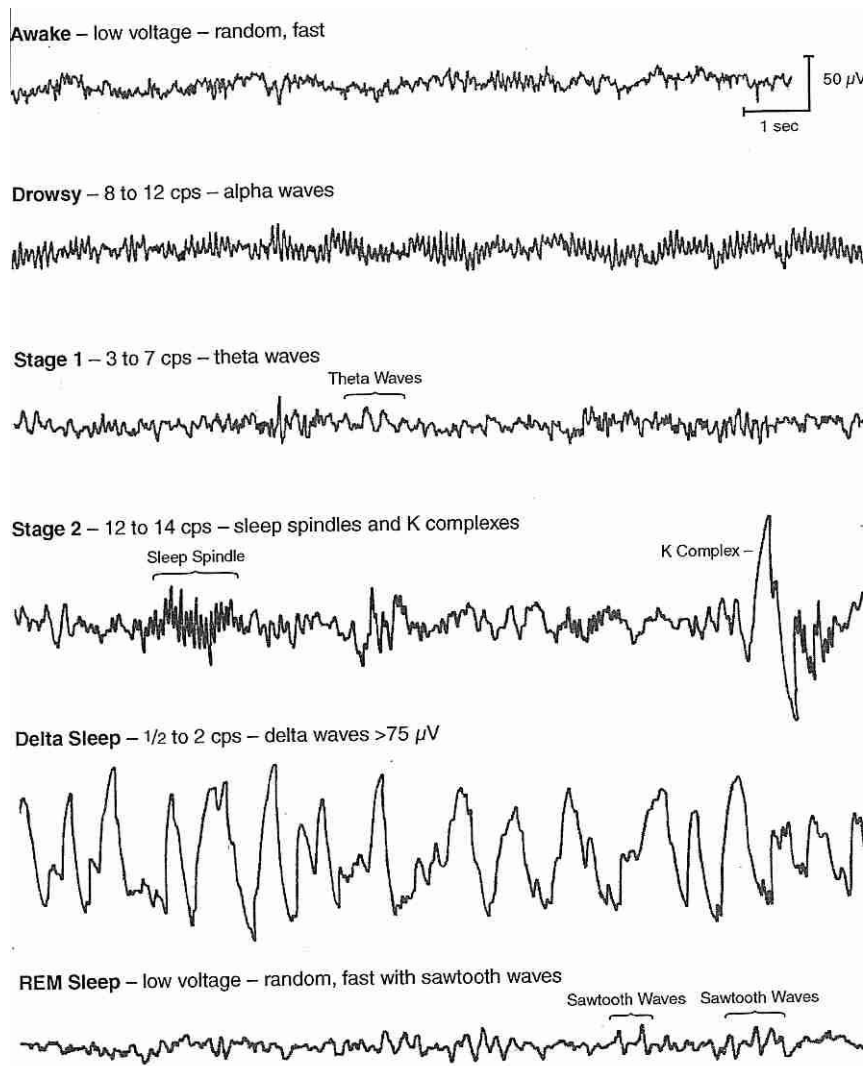
Sustav koji će se razviti u okviru rada bazirat će se na gotovim izmjerenim koordinatama gornje usne i brade. Iz tih koordinata računat će se njihova udaljenost a iz promjena udaljenosti detektirat će se zijevanje. Na kraju, mjerit će se učestalost zijevanja i iz toga će se zaključiti je li vozač umoran.

## 2. Fiziološke pojave uzrokovane umorom

Umor vozača uzrokuje promjene u aktivnosti mozga, brzini rada srca i pokretljivosti očiju. Mentalne pojave koje karakteriziraju umor vozača su pospanost, smanjenje motivacije, smanjenje pažnje, povećano vrijeme reakcije, slabija kontrola nad vozilom. Posljedica umora može biti kratkotrajan san koji traje nekoliko sekundi a čega vozač najčešće nije ni svjestan. Na umor utječu zahtjevi, trajanje i monotonija vožnje. Zahtjevi vožnje su povećani kod jako gustog prometa ili loših vremenskih uvjeta. Trajanje vožnje je posebno bitan faktor kod profesionalnih vozača koji neprestano voze po osam sati. Monotonija vožnje najizraženija je kod vožnje autocestom gdje su najveće brzine i gdje sekunda nepažnje može dovesti do nesreće.

Fiziološke pojave koje se mogu detektirati su promjene u moždanim valovima, promjena brzine otkucaja srca, smanjenje pritiska vozača na volanu, učestalo zatvaranje kapaka, učestalo zijevanje, „padanje“ glave, smanjeno pokretanje očiju, monoton izraz lica. Mjerenja koja zahtijevaju fizički kontakt s vozačem najčešće izazivaju smetnju i ne koriste se u komercijalnim sustavima. Primjer takvih mjerenja je EEG ili mjerenje pulsa kontaktnim uređajem. Mjerenja koja ne zahtijevaju fizički kontakt uglavnom su bazirana na kameri iako postoje uređaji koji imaju senzore smještene na volanu. Pomoću kamere i algoritama računalnog vida moguće je promatrati sve gore navedene parametre (izuzev EEG-a) i izračunati razinu umora vozača.

Prva fiziološka pojava koja će se opisati je promjena u moždanim valovima. Analizom EEG signala može se točno vidjeti kad osoba počinje biti umorna i može se odrediti u kojoj fazi sna se ta osoba nalazi. Na Slika 1 prikazan je EEG signal kod osobe koja se nalazi u budnom i pospanom stanju te u četiri stanja sna.



Slika 1: EEG raznih stanja spavanja

Kod umora (pospanosti) pojavljuju se pojačani alfa valovi frekvencije 12 do 14 Hz. U prvom stupnju sna valovi su smanjenje amplitude po svim frekvencijama. Drugi stupanj sna karakterizira smanjenje tjelesne temperature i broja otkucaja srca. U tom stupnju pojavljuju se šiljci (pojačana aktivnost) u alfa frekvencijama. Treći stupanj karakteriziraju valovi niskih frekvencija i velike amplitude. Posljednji stupanj sna poznat je kao REM. On se najlakše identificira ubrzanim pomicanjem očiju.

Pojačane promjene u EEG-u pojavljuju se u frontalnom i zatiljnom režnju mozga. Praćenjem signala s tih područja mozga može se postići da EEG bude vrlo učinkovit u detekciji umora jer se iz očitavanja signala može zaključiti je li osoba umorna prije nego što krene to fizički pokazivati. Nedostatak EEG-a je nepraktičnost uređaja za snimanje koji često mora koristiti puno elektroda. Te



elektrode se ne mogu dobro staviti na glavu bez korištenja gela za bolji kontakt kože i uređaja. Taj gel se suši i to predstavlja problem kod dugotrajne uporabe.

Povećavanjem umora vozača generalno se smanjuje sila kojom on pritišće volan. Uzrok toga je povećanje opuštenosti mišića. Izvedba detekcije ovim putem je jeftinija od ostalih ali manje pouzdana jer nema apsolutne vrijednosti sile kojom prosječni vozač drži volan.

Iz načina upravljanja vozilom može se također zaključiti o umoru. Ako vozač često radi nagle ispravke putanje vozila (trzaje na volanu) velika je vjerojatnost da je izgubio koncentraciju u vožnji i da je umoran.

Metode koje se baziraju na računalnom vidu uzimaju vidljive fiziološke pojave u obzir. Primjer takvih pojava je položaj glave, zatvaranje kapaka, učestalo zijevanje, izraz lica, smanjeno kretanje očima. Umornoj osobi će glava opadati prema dolje, kapci će se sporije zatvarati i otvarati, lice će mu biti monotono, tj. smanjit će se aktivnost mišića i manje će mijenjati smjer pogleda u vožnji. Sve navedene pojave će se u sljedećim poglavljima opisati kroz već postojeće razvijene sustave.

### 3. Sustavi i algoritmi za detekciju umora

U ovom poglavlju opisan će se implementirani algoritmi za detekciju umora. Ti algoritmi su: detekcija EEG-om, detekcija analizom pritiska na volan i detekcija analizom karakteristika lica.

#### 3.1. Detekcija prvog stadija sna iz EEG-a

Kao izvor podataka koristi se rad "Real-time Stage 1 Sleep Detection and Warning System Using a Low-cost EEG Headset" autora Bryana Van Hala. U svom radu koristio je *NeuroSky Mindset* - uređaj za snimanje EEG signala. Uređaj koristi jednu suhu elektrodu koja se postavlja na čelo. Povezuje se Bluetoothom s računalom koje prikuplja informacije. Jeftin je i pristupačan uređaj i mogao bi se koristiti kod profesionalnih vozača za detekciju umora. Uređaj uzorkuje signale frekvencijom 512 Hz.



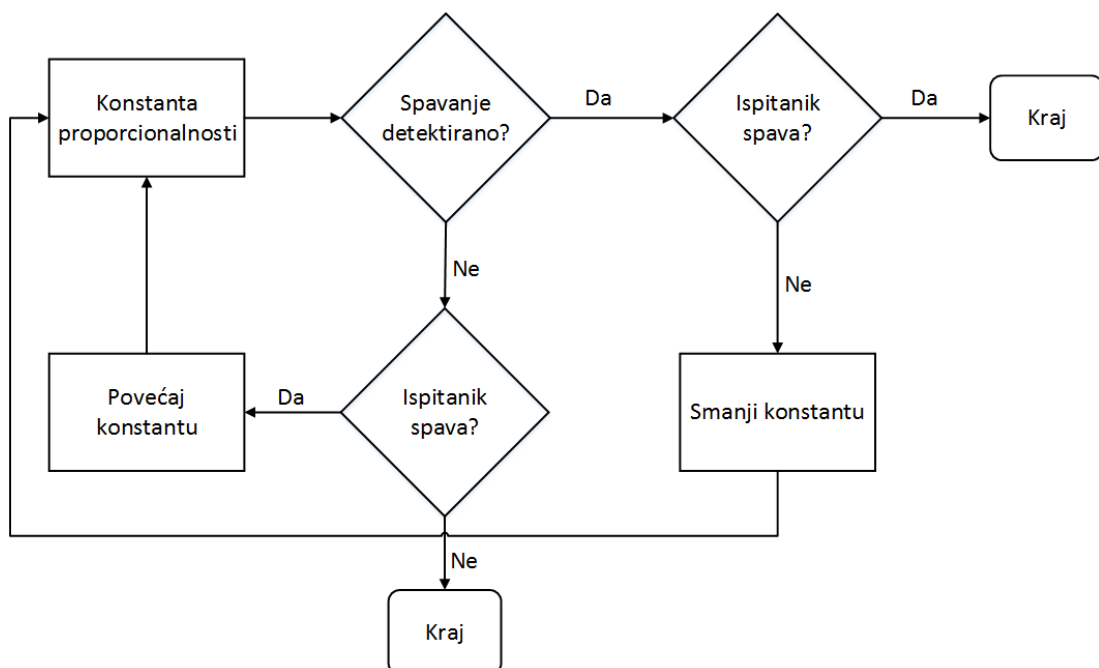
Slika 2: NeuroSky Mindset uređaj za EEG

Implementirani algoritam za detekciju bazira se na radu Lal et al. (2007) "Development of an algorithm for an EEG-based driver fatigue countermeasures". Algoritam koristi zadnjih 30 sekundi snimljene aktivnosti za analizu i izračun. Amplitudni prag za umor definiran je posebno za gornji i donji spektar alfa frekvencija. Prag se računa po formuli:

$$A_{LST} = a_{LS1} \cdot A_{LM} + a_{LS2} \cdot A_{LSD} \quad (1)$$

$A_{LST}$  je prag za niske alfa frekvencije nakon kojeg je detektiran umor.  $A_{LM}$  je srednja vrijednost amplitude a  $A_{LSD}$  je standardna devijacija.  $a_{LS1}$  i  $a_{LS2}$  su konstante pomoću kojih se dodjeljuje težina amplitudnim vrijednostima. Te konstante se dobivaju eksperimentalno a vrijednost im se dodjeljuje nakon vizualne detekcije umora iz EEG-a. Za gornji spektar alfa frekvencija koristi se ista formula, ali sa drugim konstantama.

Postupak dobivanja konstanti je iterativan i može se vidjeti na Slika 3. Akcija smanjivanja konstante proporcionalnosti utječe tako da smanjuje vjerojatnost detekcije spavanja a akcija povećavanja suprotno.



Slika 3: Graf postupka određivanja konstante

Nakon dobivenih konstanti, uređaj se može priključiti na bilo koju osobu. Tada računalo ima zadatak usporediti jačinu signala s pragom i odrediti je li osoba umorna ili ne. Poteškoće koje se javljaju kod rada su slab signal kojem može biti uzrok loš kontakt elektrode s kožom i prejak signal kojeg uzrokuju nestandardne pojave (npr. pojačano pomicanje mišića glave). Problem lošeg kontakta rješava konstrukcija korištenog uređaja koji samostalno drži dovoljan pritisak na glavu. Problem prejakog signala također rješava uređaj s ugrađenim sustavom detekcije signala s puno šuma, koji se zatim hardverski otklanja. Lal et al. (2007) u svom radu koristi maksimalnu amplitudnu ulaznu vrijednost iznad koje sve signale izbacuje iz izračuna.

Uređaj je testiran u 16 testova. U 13 testova je detektiran umor što daje točnost od 81%. U deset testova algoritam je detektirao umor prosječno 8,4 sekunde kasnije od vizualne detekcije. U ostalim testovima za detekciju je trebalo više od 30 sekundi. Tri slučaja su bila netočno pozitivna jer ispitanik u tom trenutku nije počeo spavati.

### 3.2. Detekcija umora analizom pritiska na volan

Chieh et al. u svom radu "*Driver Fatigue Detection Using Steering Grip Force*" opisali su metodu detekcije umora zasnovanu na analizi vozačevog pritiska na volan. Pritisak vozača na volan pada kako mu se povećava umor ili smanjuje oprez u vožnji zbog opuštanja mišića. U radu nije bilo moguće koristiti prag sile ispod kojeg bi umor bio detektiran nego je korištena statistička analiza. Prag se ne može koristiti jer svaki vozač drži volan na različit način primjenjujući različitu silu. Radnje kod skretanja vozila također unose neiskoristive vrijednosti u izračun. Implementirani algoritam prati prosječnu silu na volan i detektira značajan pad sile kroz vrijeme koji bi mogao biti uzrokovan umorom. Sila se mjeri dvama sensorima smještenima na volanu vozila. Cjelokupni algoritam se koristi kao dio većeg sustava za detekciju umora koji koristi i ostale ulaze (npr. analiza lica). Razvijeni detektor umora bazira se na algoritmu za detekciju promjena. Podaci o sili prikupljeni sa senzora uzimaju se kao sekvenca neovisnih slučajnih varijabli  $y$  sa gustoćom vjerojatnosti  $p(y)$  ovisne o skalarnom parametru srednje vrijednosti sekvence. Za nepoznato vrijeme promjene parametar  $\theta$  jednak je  $\mu_0$  a za poznato  $\mu_1$ . Osnova same detekcije je logaritamska vrijednost omjera funkcija gustoće vjerojatnosti prije i nakon vremena detekcije promjene sile. Formula je:

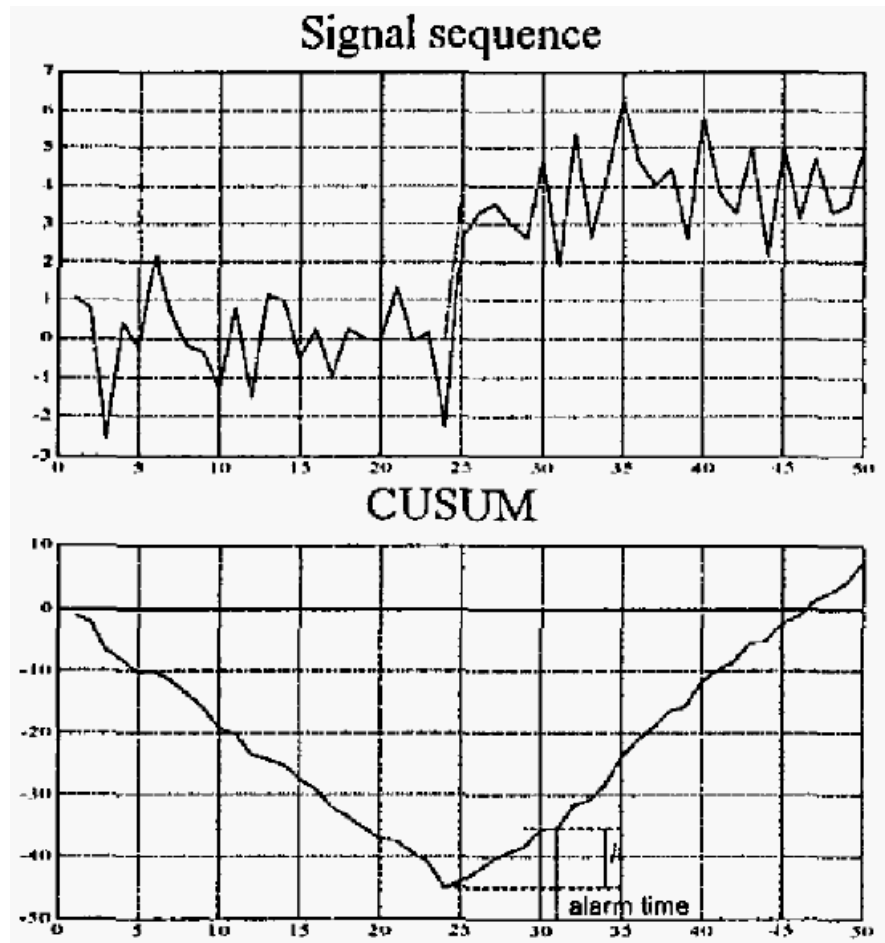
$$s_i = \ln \frac{p_{\mu_1}(y_i)}{p_{\mu_0}(y_i)} \quad (2)$$

$p_{\mu_1}$  je veći od  $p_{\mu_0}$  kad je lokalna srednja vrijednost sekvence jednaka  $\mu_1$  a manja kad je lokalna srednja vrijednost jednaka  $\mu_0$ . Rezultat toga je da se predznak funkcije mijenja iz negativnog u pozitivni kada se dogodi promjena lokalne

srednje vrijednosti sekvence  $y_1$  iz  $\mu_0$  u  $\mu_1$ . Algoritam detekcije promjene računa kumulativnu sumu vrijednosti iz funkcije 2 prema formuli:

$$S_k = \sum_{i=1}^k S_i \quad (3)$$

Promjena predznaka  $S_i$  u  $S_k$  se očituje kao lokalni minimum.



Slika 4: Prikaz ulaznog signala i izračuna kumulativne sume[3]

Promjena u srednjoj vrijednosti, odnosno detektirani umor, potvrđuje se ako se kumulativna suma nastavi povećavati u preddefiniranom vremenu nakon što je dosegla svoj lokalni minimum. Preddefinirano vrijeme čekanja je vrijeme aktivacije alarma koje se koristi kako bi se izbacio utjecaj šuma i nestabilnost u signalu.

Eksperiment je proveden s volanom i simulatorom vožnje na računalu. Korišteni senzori su rezistivni kod kojih otpor ovisi o pritisnoj sili. Vrijednosti otpora pretvorene su u napon koji je zatim filtriran nisko propusnim filtrom. Podaci su analizirani na računalu u stvarnom vremenu. Inicijalno podaci su obrađivani po normalnoj razdiobi, sa srednjom vrijednosti  $\mu$  i varijancom  $\sigma^2$ , po formuli:

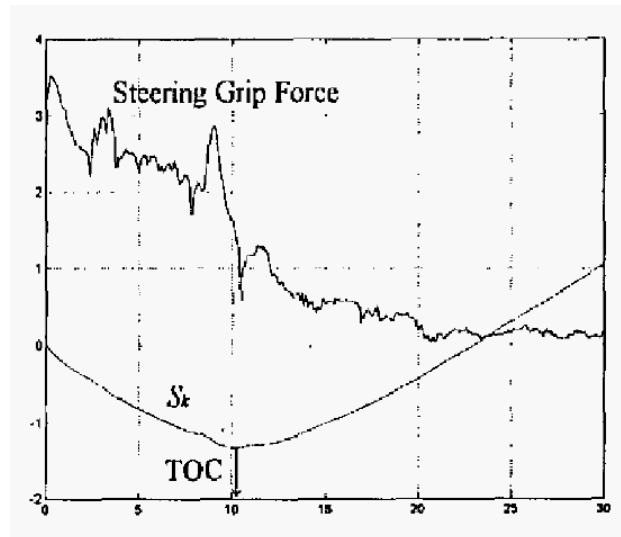
$$P_{\mu}(y) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(y-\mu)^2}{2\sigma^2}} \quad (4)$$

Zbog kompleksnosti izračuna formule 4 na računalima manjih procesnih mogućnosti u testiranju su korištene i trapezoidne funkcije razdiobe. Korištene formule su:

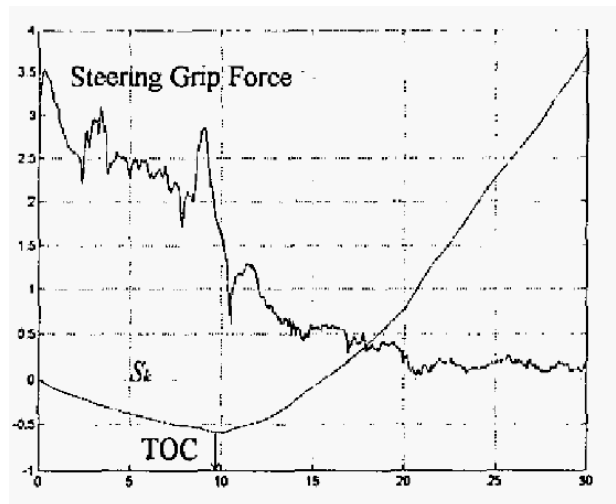
$$P_{\mu_0}(y) = \begin{cases} \frac{1}{\mu_0} y, & y < \mu_0 \\ 1, & y \geq \mu_0 \end{cases} \quad (5)$$

$$P_{\mu_1}(y) = \begin{cases} 1, & y \leq \mu_1 \\ \frac{1}{\mu_1 - \max(y)} y - \frac{\max(y)}{\mu_1 - \max(y)}, & y > \mu_1 \end{cases} \quad (6)$$

$\max(y)$  je maksimalna izlazna vrijednost senzora za pritisak. Rezultati eksperimenta mogu se vidjeti na slikama Slika 5 i Slika 6.



Slika 5: Rezultati testiranja s normalnom razdiobom[3]



Slika 6: Rezultati testiranja s trapezoidnom funkcijom[3]

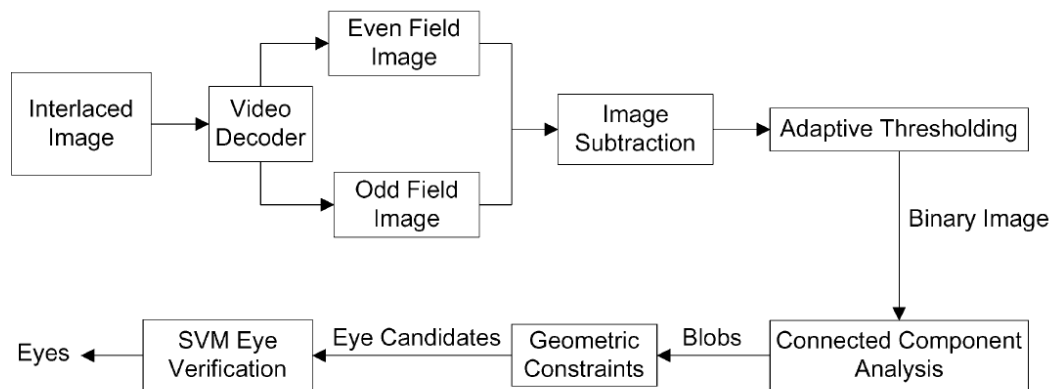
Razlika u vremenu detekcije korištenjem funkcije s normalnom razdiobom i trapezoidne funkcije ima grešku od 3.55%. Korištenjem trapezoidne funkcije vrijeme detekcije je ranije što znači da je ona osjetljivija i može imati više netočnih detekcija.

### 3.3. Detekcija umora analizom više karakteristika lica

U radu "*Real-Time Nonintrusive Monitoring and Prediction of Driver Fatigue*" opisan je sustav za nadzor vozača u stvarnom vremenu analizirajući karakteristike njegova lica. Ono koristi kamere s aktivnim infracrvenim svjetlom za poboljšanje dobivene slike. Karakteristike koje se promatraju su pomicanje kapka, kretanje očiju, kretanje glave i izraz lica. Za detekciju umora razvijen je vjerojatnosni model za detekciju. U izračunu se koriste sve promatrane karakteristike što daje točniji rezultat. Sustav je testiran u stvarnim uvjetima umora s ljudima raznih karakteristika.

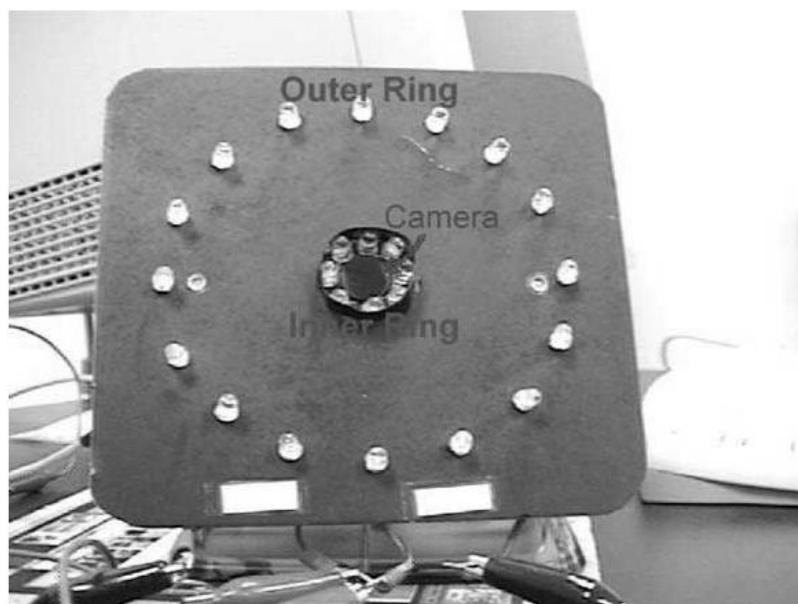
#### 3.3.1. Detekcija i praćenje oka

Na Slika 7 prikazan je graf sustava koji se koristi za detekciju i praćenje oka.



*Slika 7: Shema sustava za detekciju i praćenje oka[2]*

Sustav koristi dvije kamere: kameru sa širokim objektivom za nadzor lica i kameru s uskim objektivnom za nadzor očiju. Slika sa prve kamere koristi se za nadzor pokreta glave i izraza lica a slika s druge za praćenje kapaka i pokreta očiju. Pomoću aktivnog infracrvenog osvjetljenja ostvaruje se efekt refleksije na zjenici koji pomaže kod detekcije oka. Osim tog efekta, infracrveno svjetlo osigurava konzistentnu kvalitetu dobivene slike s obzirom na promjenljivo vanjsko osvjetljenje. Pošto je osvjetljenje nevidljivo vozaču, prednost mu je to što ga neće ometati tijekom vožnje. Položaj infracrvenog svjetla uz kameru prikazan je na Slika 8.



*Slika 8: Kamera s dva prstena LE dioda[2]*

Na slici se mogu vidjeti dva kruga LE dioda. Svaki krug ima svoju funkciju i s njima se upravlja preko video dekodera. Funkcija unutarnjeg kruga je da se dobije slika s efektom refleksije zjenice a vanjskog da se dobije slika gdje nema



tog efekta, iako i na toj slici postoji refleksija iz sredine zjenice ali manjeg intenziteta.

Za lakšu detekciju zjenice, razvijen je sklop koji alternirajuće pali dva kruga LE dioda kako bi se kreirale dvije slike s različitim izvorom osvjetljenja. Obje slike imaju jednaku pozadinu i vanjsko osvjetljenje. Oduzimanjem jedne slike od druge dobit će se slika na kojoj su istaknute zjenice zbog refleksije kod korištenja unutarnjeg kruga LE dioda. Nakon toga iz slike se uklanja šum korištenjem praga minimalnog osvjetljenja te se detektiraju binarni okrugli oblici koji ispunjavaju uvjete veličine i oblika zjenice. U dobivenoj slici se mogu još uvijek pojaviti lažno detektirani oblici pa se za izbacivanje pogrešno detektiranih oblika koristi stroj s potpornim vektorima (engl. *support vector machine*). Stroj koristi karakteristike regije oko detektiranih oblika da utvrdi da li se oni nalaze u području očiju.

Detektirane oči prate se u svakoj sljedećoj slici. U radu su implementirana dva algoritma za praćenje oka. Prvi je Kalmanov filter<sup>1</sup> koji se bazira na refleksiji zjenice a drugi je algoritam koji koristi prosjek pomaka očiju. Kalmanov filter dobro radi u normalnim uvjetima dok je snimana osoba okrenuta prema kameri. Drugi algoritam se koristi kad osoba zatvori oči, okrene glavu ili kad vanjsko osvjetljenje poremeti efekt refleksije zjenice. On detektira i prati regiju očiju tako da se ne koristi refleksija zjenice. Ni jedan od dva algoritma nije u potpunosti pouzdan i zbog toga se moraju koristiti oba da bi jedan premostio nedostatke drugog.

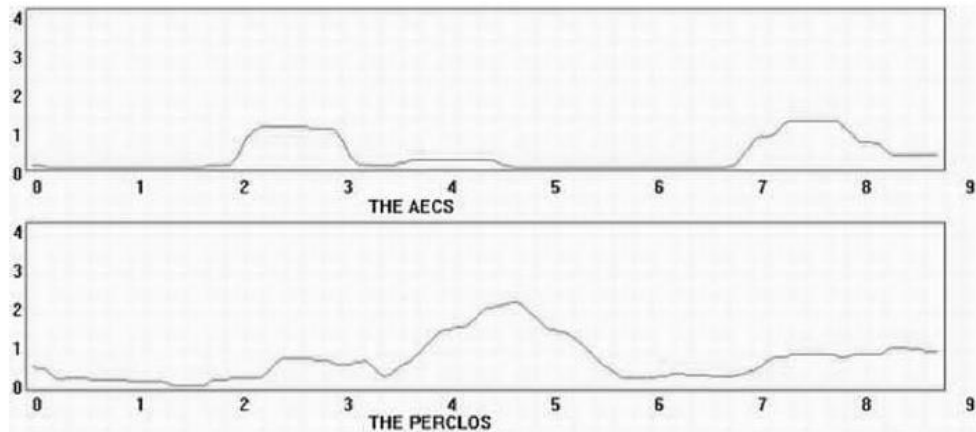
### 3.3.2. Parametri praćenja pomaka kapaka

Pomaci kapaka su jedan od parametara koji odražavaju stanje umora promatrane osobe. Primarna namjena detekcije očiju je praćenje kapaka i izračun relevantnih parametara. Prate se dva parametra: postotak zatvorenosti oka u vremenu (PERCLOS) i srednja brzina zatvaranja očiju (AECS). PERCLOS se smatra jednim od najtočnijih parametara za praćenje umora. Brzina otvaranja i zatvaranja očiju je također dobar pokazatelj umora. U radu se definira kao vrijeme koje je potrebno da osoba potpuno zatvori i otvori oči a osobi koja je umorna taj proces će trajati primjetno duže.

---

<sup>1</sup> <http://www.cs.unc.edu/~welch/kalman/media/pdf/Kalman1960.pdf>

Stupanj otvorenosti oka karakterizira se oblikom zjenice. Primijećeno je da kod zatvaranja oka zjenice postaju više eliptične pa se kod izračuna koriste omjeri osi elipse. Kumulativno vrijeme zatvorenosti oka koristi se kod izračuna PERCLOS-a. Korišteni vremenski prozor je 30 sekundi. Prikaz grafa izračuna PERCLOS-a i AECS-a nalazi se na Slika 9.

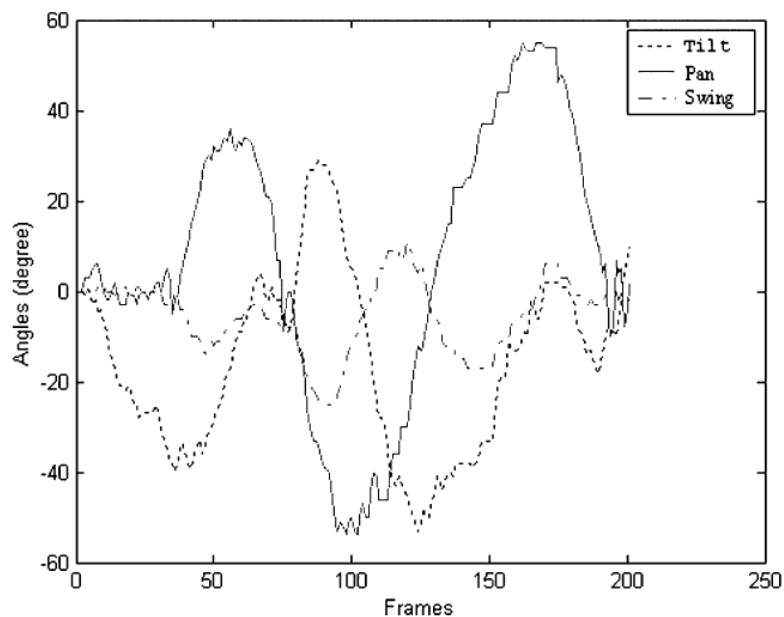


Slika 9: Grafovi pomaka kapaka - gornji prikazuje AECS a donji PERCLOS[2]

### 3.3.3. Estimacija orijentacije glave

Iz položaja glave mogu se saznati informacije o nečijoj razini pažnje i umora. Česti trzaji glave označavaju pojavu umora. Kako je normalan položaj glave frontalni s obzirom na upravljač, dugotrajno okrenuta glava u nekom smjeru označava da je vozač umoran ili da ne pridaje dovoljnu pažnju stanju na cesti. Zbog toga, orijentacija glave se može koristiti za detekciju umora i pažnje vozača. U radu je fokus na detekciji čestih trzaja glave.

Detekcija se bazira na trodimenzionalnom izračunu pozicije glave. Sustav zahtjeva prethodnu inicijalizaciju gdje vozač mora biti okrenut frontalno prema kameri kako bi se izmjerila udaljenost između očiju i njihova pozicija što će se kasnije koristiti. Za praćenje pomaka glave koristi se Kalmanov filter. Njegov nedostatak što ne može raditi kad su pomaci glave veliki rješava se korištenjem prethodno razvijenog pouzdanijeg algoritma za detekciju očiju. Konačni izračun koristi kombinaciju izračuna Kalmanovog filtra i algoritma detekcije očiju.

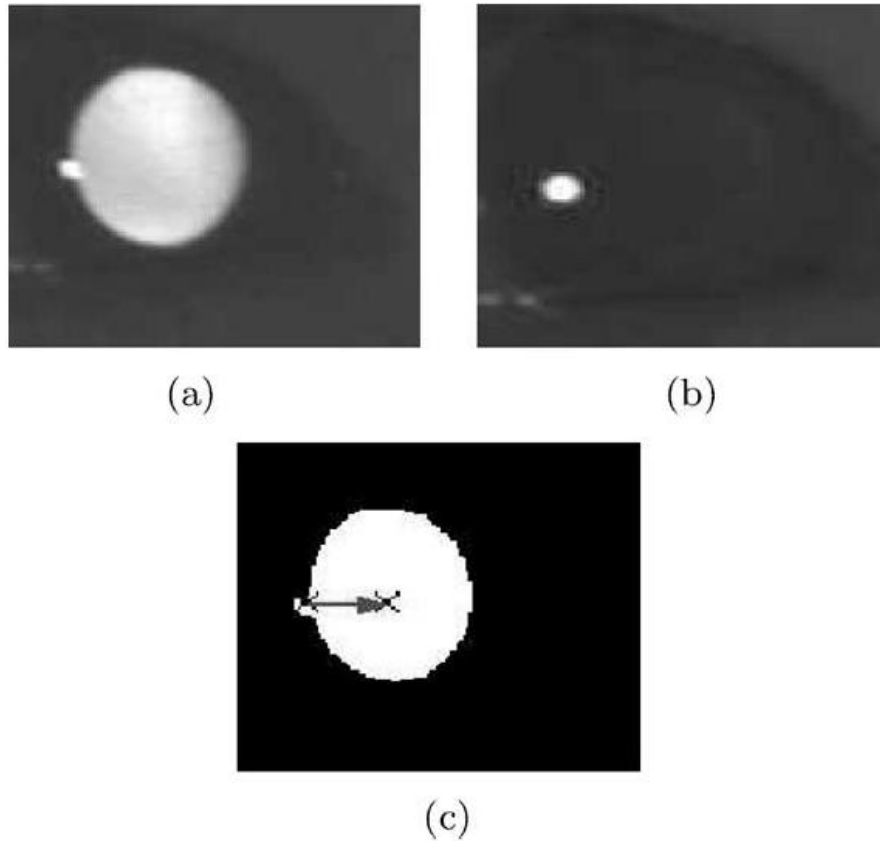


*Slika 10: Graf stupnja rotacije glave[2]*

#### 3.3.4. Praćenje smjera pogleda

Smjer pogleda vozača ukazuje na njegovu razinu budnosti tako da umorniji vozači teže bližem pogledu. Pogled se također može koristiti za analizu nečijih potreba (u trgovinama, kod gledanja reklama) ili razine opreza. Smjer pogleda određuje se iz orijentacije glave i orijentacije očiju. Orijehtacija glave određuje generalni smjer pogleda dok orijentacija očiju daje konačan precizni rezultat. Najčešće korišten pristup određivanja smjera pogleda je određivanje relativne pozicije između zjenice i odsjaja sa rožnice oka. Ta metoda se pokazuje uspješna samo za statičnu glavu i ne može se koristiti kod promatranja vozača. U radu je razvijen algoritam koji kombinirano koristi položaj glave i očiju.

Računanje smjera pogleda ima tri faze: detekcija i praćenje odsjaja zjenice, kalibracija smjera pogleda i mapiranje smjera pogleda. Za detekciju odsjaja zjenice koriste se već opisane infracrvene lampice. Na Slika 11a prikazane su oči snimanje vanjskim krugom lampica a na Slika 11b unutarnjim krugom.



*Slika 11: Prikaz slika očiju snimljenih dok je uključen vanjski (a), odnosno unutarnji (b) krug ledica te detektirani centar odsjaja i zjenice(c) [2]*

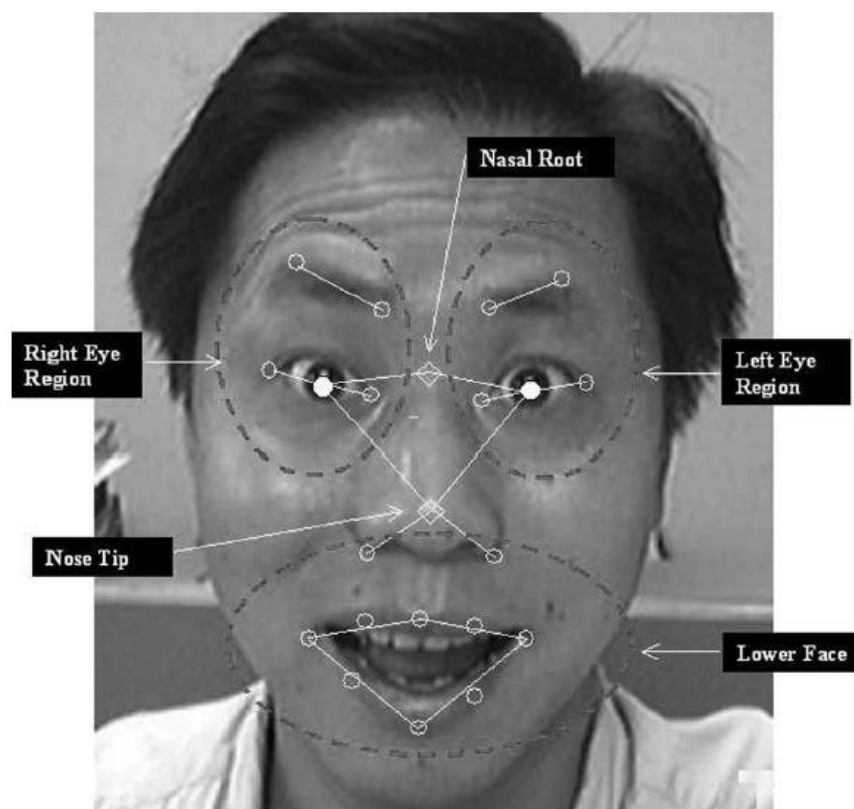
Iz tih slika izračunava se lokalni i globalni smjer pogleda. Kod izračuna konačnog smjera pogleda koriste se parametri,  $\Delta_x$ ,  $\Delta_y$ ,  $r$ ,  $\theta$ ,  $G_x$  i  $G_y$ . Parametri  $\Delta_x$  i  $\Delta_y$  su razlike u poziciji detektiranih centara odsjaja i zjenice i označavaju lokalni smjer pogleda. Duljina vektora između tih centara označava udaljenost objekta kojeg vozač gleda.  $r$  je omjer velike i male osi elipse detektirane na zjenici i označava rotaciju glave izvan ravnine kamere (gore-dolje, lijevo-desno) dok je  $\theta$  orijentacija elipse koja označava samu rotaciju glave (nagnuta glava).  $G_x$  i  $G_y$  su koordinate centra odsjaja i označavaju translaciju glave u ravnini kamere. Svi navedeni parametri koriste se kod funkcije za stvarni izračun smjera. Funkcija je aproksimirana općenitom regresijskom neuronskom mrežom<sup>2</sup> (GRNN) koju karakteriziraju brzo vrijeme učenja, mogućnost modeliranja nelinearnih funkcija i koja je pokazala dobre rezultate u signalima s dosta šuma.

<sup>2</sup> <http://www.mathworks.com/help/nnet/ug/generalized-regression-neural-networks.html>

Smjer pogleda koristi se pri izračunu dodatnog parametra umora: udaljenosti pogleda. Taj parametar koristi podatke o smjeru pogleda da bi se analizirala vozačeva razina osviještenosti o situaciji u prometu. Dodatan parametar koji se uvodi je brzina pomicanja očiju kroz vrijeme. Vozač koji je u normalnom stanju će s vremena na vrijeme brzo pomicati oči i fokusirati se na razne objekte ispred sebe. Prestanak tih akcija upućuje na pojavu umora.

### 3.3.5. Analiza izraza lica

Izraz lica umorne osobe karakteriziraju usporeni mišići lica, smanjeno pokretanje mišića i učestalo zijevanje. Značajke lica oko očiju i usta mogu se najefikasnije koristiti kod izračuna. Na slici 12 se mogu vidjeti karakteristične točke lica koje razvijeni algoritam koristi.

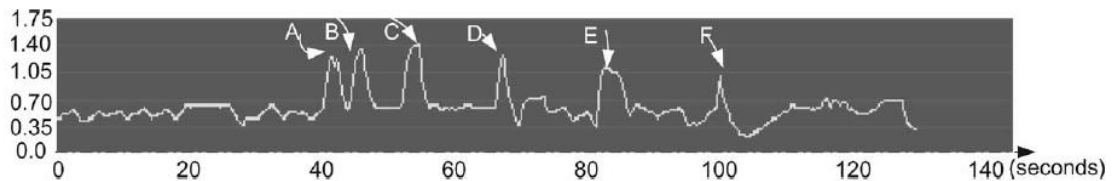


*Slika 12: Karakteristične točke lica[2]*

Karakteristične točke se iz slike dobivaju Gaborovom valičnom transformacijom<sup>3</sup>. Nakon detekcije točkaka, koristi se Kalmanov filtar za njihovo praćenje koji koristi prethodno detektirane oči kao referencu za pronalazak

<sup>3</sup> <http://www.cnbc.cmu.edu/~tai/papers/pami.pdf>

ostalnih točaka. Fokus u ovom dijelu rada je na detekciji zijevanja. Zijevanje se detektira ako se u području usta karakteristične točke znatno pomiču, posebno u vertikalnom smjeru. Visina usta je tu predstavljena s udaljenosti između gornje i donje usne a širina s udaljenosti između lijevog i desnog kraja usta. Pomoću tih parametara analizira se otvorenost usta i detektiraju se zijevanja. Otvorenost usta predstavlja ukupnu površinu. Na slici 13 može se vidjeti graf otvorenosti usta na kojem su slovima A do F označena detektirana zijevanja.



Slika 13: Graf otvorenosti usta s detektiranim zijevanjem[2]

### 3.3.6. Modeliranje umora pomoću Bayesovih mreža

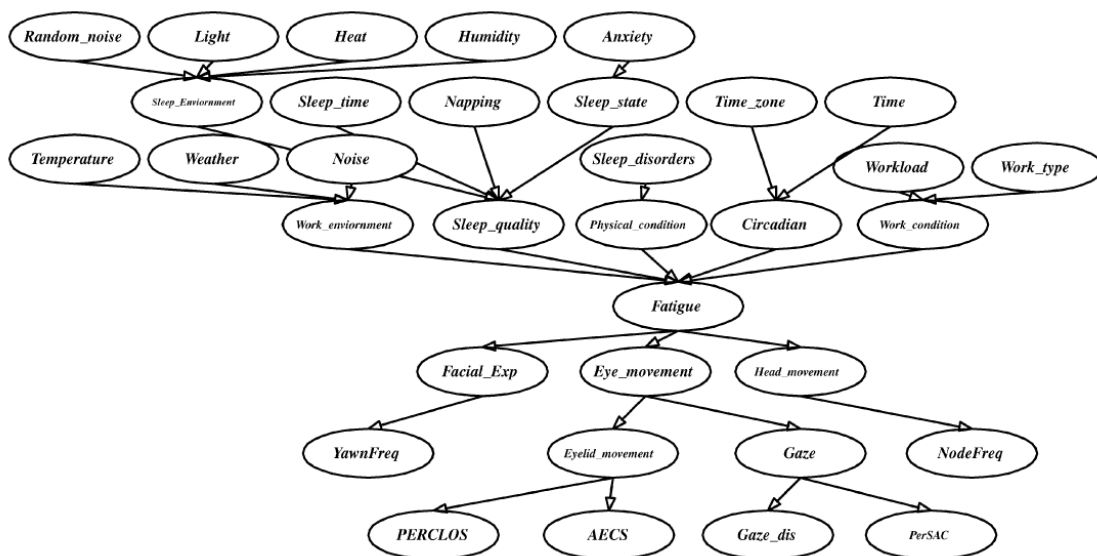
Modeliranje umora je vrlo kompleksan zadatak. Poteškoće modeliranja umora uključuju i nemogućnost zaključivanja o umoru trenutačnim promatranjem. Stoga se svi zaključci izvode iz slijedno dobivenih informacija agregiranih kroz vrijeme. Umor se smatra rezultatom brojnih vanjskih utjecaja poput radne atmosfere, zdravlja i povijesti spavanja. Umor karakteriziraju vizualni simptomi poput neuobičajenih pokreta kapaka, zijevanje i učestalo trzanje glavom. Na karakteristike simptoma umora utječu godine, visina, zdravlje i oblik lica. Da bi se umor mogao efektivno nadzirati, nužno je koristiti sustav koji će integrirati više izvora informacija i na temelju njihovih kombinacija donositi odluke. Bayesova mreža u ovom slučaju je najbolja opcija za zadani problem. Ona pruža mehanizam grafičkog prikaza nepouzdatih i nesigurnih informacija i mehanizam kombiniranja tih informacija na visokoj razini. Mreža se sastoji od čvorova i lukova koji zajedno čine usmjereni aciklički graf. Svaki čvor grafa je zasebna varijabla koja može biti diskretna ili kontinuirana. Lukovi predstavljaju vjerojatnosnu međuovisnost čvora roditelja s čvorom djeteta.

Primarna zadaća Bayesove mreže je izvlačenje neprimjetnih događaja iz primjetnih i kontekstualnih podataka. Prvi korak u kreiranju takve mreže je

identificiranje tih događaja i njihovo grupiranje u međusobno isključive događaje kako bi se stvorila slika o ciljnom događaju kojeg se modelira. Sljedeći korak identificira dostupne podatke koji daju informacije o događaju i grupira ih u informacijske varijable. Osim ova dva koraka, postoji još nekoliko njih u kojima se nastoji bolje povezati hipotetske varijable na visokoj razini s informacijskim na niskoj.

Za modeliranje umora, ciljna hipotetska varijabla je umor. Sve ostale varijable, koje uzrokuju umor i pojavu vizualnih znakova umora, su informacijske varijable. Najznačajnija varijabla je povijest spavanja a za njom slijede cirkadijski ritam, uvjeti rada, okolišni uvjeti i fizičko stanje osobe. Okolišni uvjeti koji se uzimaju u obzir su temperatura, vrijeme i buka. Za fizičko stanje osobe u izračun se uzimaju dob i nepravilnosti kod spavanja. Cirkadijski ritam karakteriziraju doba dana i promjena vremenske zone. Faktori uvjeta rada su radno opterećenje i tip posla. Na kvalitetu sna utječu okruženje i vrijeme sna a na okruženje tokom spavanja utječu buka, razina osvjetljenja, toplina i vlaga.

Sustav za vizualnu detekciju razvijen u radu koristi PERCLOS i AECS za pomake kapaka, NodFreq za pomake glave, GAZEDIS i PERSAC za pomake zjenice i YawnFreq za analizu izraza lica. Kreirana Bayesova mreža koja koristi sve navedene module nalazi se na slici 14.



Slika 14: Bayesova mreža za detekciju umora[2]

Čvorovi iznad čvora umora (engl. *fatigue*) označavaju razne faktore koji utječu na umor. Oni se mogu smatrati kontekstualnim informacijama. Čvorovi ispod čvora umora su vizualne opservacije dobivene iz opisanog sustava računalnog vida i one se zovu opservativnim informacijama.

Bez vjerojatnosnih parametara Bayesova mreža nije potpuna. Za kreiranje tih parametara potrebno je odrediti početne, odnosno težinske, vrijednosti početnih čvorova mreže i uvjetne vjerojatnosti veza između čvorova. Te vrijednosti se uobičajeno dobivaju statističkom analizom velike količine ulaznih podataka za treniranje mreže. U ovom radu podaci kojima su dobivene vjerojatnosti izvedene su pomoću tri različita izvora podataka. Prve informacije dobivene su od testiranih osoba pomoću čega su se trenirale niže razine mreže. Drugi stupanj adaptacije mreže vršio se pomoću velikog broja anketa koje su ispitanici ispunjavali. Ankete su bile subjektivne i iz njih je bilo potrebno izbaciti beskorisne podatke i modelirati vjerojatnosti. U trećem stupnju preostale vjerojatnosne vrijednosti za koje se nisu mogli dobiti podaci zadane su subjektivnom estimacijom. Dobivene vjerojatnosti nalaze se u tablici 1.

*Tablica 1: Vjerojatnosti veza Bayesove mreže*

Čvor	Stanje	Vjerojatnost
Nasumična buka	Da	0.15
	Ne	0.85
Svjetlo	Upaljeno	0.13
	Ugašeno	0.87
Vrućina	Velika	0.24
	Normalna	0.76
Vlažnost	Visoka	0.19
	Normalna	0.81
Duljina sna	Dostatno (> 6 sati)	0.90
	Nedostatno (< 6 sati)	0.10
Drijemanje	> 30 minuta	0.22
	Ne	0.78
Tjeskoba	Da	0.28
	Ne	0.72



Nepravilnost spavanja	Da	0.08
	Ne	0.92
Radno opterećenje	Veliko	0.15
	Normalno	0.85
Doba dana	Neaktivno	0.26
	Aktivno	0.74
Temperatura	Visoka	0.15
	Normalna	0.85
Vrijeme (atmosfersko)	Neuobičajeno	0.10
	Normalno	0.90
Buka	Velika	0.15
	Normalna	0.85
Tip posla	Dosadan	0.20
	Normalan	0.80

Parametrizirani sustav može početi donositi zaključke o umoru propagacijom vjerojatnosti kada vizualne informacije krenu dolaziti. Za izvođenje sustava i donošenje zaključaka korišten je program MSBNX<sup>4</sup>. U tablici 3 nalaze se dobivene vjerojatnosti tijekom izvođenja programa. U prvom retku se može vidjeti da početna vjerojatnost umora prije dobivanja podataka iznosi 0.5755. Prag vjerojatnosti iznad koje se zaključuje da je osoba umorna je 0.95. U tablici se može vidjeti kako moduli sustava ne mogu samostalno uzrokovati povećanje vjerojatnosti iznad praga ali u kombinaciji s kontekstualnim informacijama može doći do skoka u izračunu.

*Tablica 2: Rezultati izračuna vjerojatnosti umora*

Redni broj	Podaci i parametri	Vjerojatnost umora
1	Bez podataka	0.5755
2	YawnFreq visok	0.8204
3	PERCLOS visok	0.8639

<sup>4</sup> Online MSBNX Editor Manual and Software Download, M.R. Center; <http://research.microsoft.com/adapt/MSBNx>

4	AECS spor, duljina sna nedovoljna, doba dana za drijemanje	0.9545
5	YawnFreq visok, AECS spor	0.9552
6	Duljina sna nedovoljna, doba dana za drijemanje, temperatura visoka	0.8363

## 4. Postojeći sustavi za detekciju umora

U prethodnom poglavlju opisani su znanstveni radovi bazirani na temi detekcije umora. Kako su neki od tih radova stariji od deset godina, prirodno je za očekivati da takvi sustavi već postoje i da su ugrađeni u današnje moderne automobile.

U ovom poglavlju navest će se postojeći ugrađeni sustavi detekcije umora. Kako se sam koncept detekcije bazira na opsežnim znanstvenim istraživanjima, cijena im je vrlo visoka i zasad su takvi sustavi rezervirani samo za automobile visoke klase.

### 4.1. Audi

Audi je razvio „*Rest Recommendation System*“ koji obavještava vozača kad postane umoran. Informacije prikuplja iz pritiska volana, pedala kvačila, gasa i kočnice i pokreta kod mjenjača brzina.

### 4.2. BMW

BMW koristi „*Driving Assistant*“ i „*Driving Assistant Plus*“ u svojim automobilima. Sustav upozorava vozača ako izlazi iz trake u kojoj vozi, ako ima pješaka ispred njega i općenito detektira opasne situacije. U vidu detekcije umora, sustav analizira ponašanje vozača i njegove reakcije u vožnji i na temelju toga predlaže vozaču da odmori.

### 4.3. Volkswagen

Volkswagenov sustav „*Fatigue Detection*“ prati stanje vozača i automatski, ukoliko je potrebno, daje signal vozaču. Sustav svoje zaključke donosi na temelju analize vozačevog upravljanja volanom i drugim raznim signalima iz automobila. Sustav radi samo za brzine veće od 65 km/h i ako je vozač umoran alarmira ga

u intervalima od 15 minuta. Volkswagen navodi da se sustav ne koristi ako vozač ima sportski stil vožnje, ako je cesta krivudava i ako ima dosta rupa.

#### 4.4. Bosch

Bosch je također razvio sustav detekcije umora. Njihov proizvod bazira se na analizi upravljanja volanom. Ako vozaču padne koncentracija i na trenutak pusti volan, ubrzo će mu se naglo vratiti koncentracija i pokušat će ispraviti putanju vozila. Te pojave lako je povezati s umorom. Boschov sustav kombinira frekvenciju i korištenu snagu takvih događaja te uzima u obzir brzinu vozila, doba dana i vozačevo korištenje ostalih instrumenata u automobilu (žmigavci, brisači, upravljačke tipke, ...).

#### 4.5. Mercedes-Benz

Mercedes-Benz u svoje automobile ugrađuje sustav „*Attention Asisst*“ koji promatra vozača i na početku svakog putovanja kreira vozačev profil i u vožnji ga analizira. Sustav se koristi na brzinama od 80 do 180 km/h. Svoj detektor su bazirali na analizi vozačevog upravljanja volanom. Umoran vozač će imati poteškoće kod održavanja jednakog smjera u traci i radit će, kao što je kod Boschovog sustava opisano, nagle ispravke u putanji.

## 5. Programska potpora i korišteni hardver

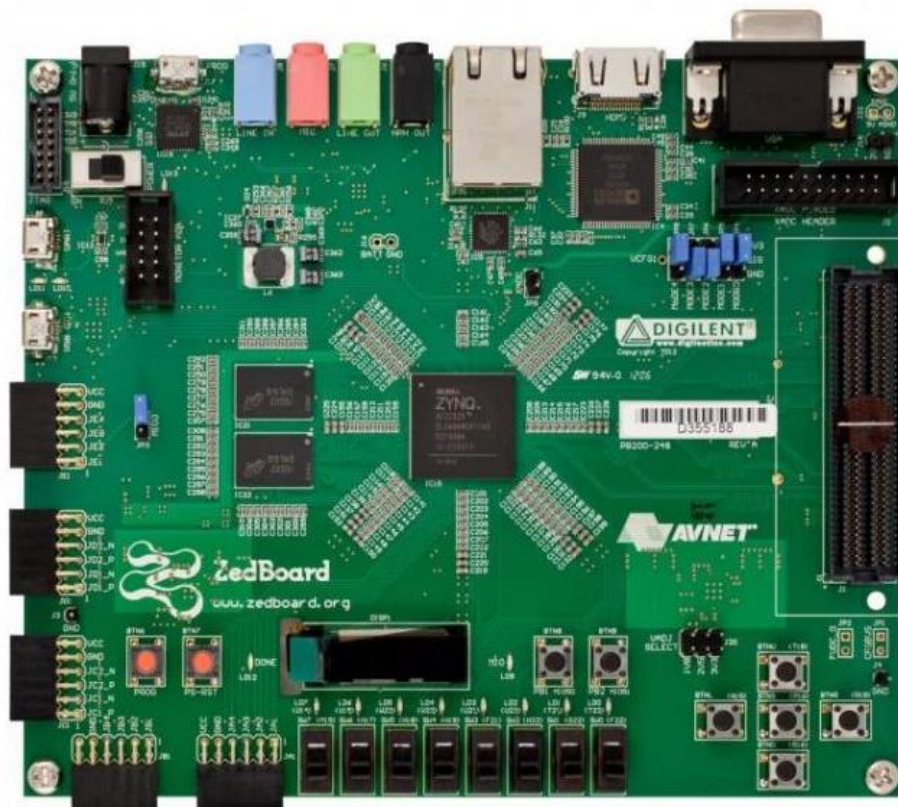
U ovom poglavlju ukratko će se opisati korišteni alati i hardver. Za izradu rada korišten je Zedboard razvojni sustav i programski alati Matlab, Modelsim i Xilinx Vivado.

### 5.1. Zedboard

Zedboard je razvila tvrtka Avnet. On je *low-cost* razvojni sustav za Xilinxov *Zync-7000 All Programmable* sustav na čipu. Sustav sadrži sve potrebno da bi se kreirao dizajn koji se može izvoditi na Linux, Android i Windows operacijskim sustavima. Zync 7000 aplikacijski procesor koji se nalazi na sustavu baziran je na dvojezrenom ARM 9 procesoru uparenom s Xilinxovom programabilnom logikom. Ostale karakteristike sustava su:

- Memorija: 512 MB DDR3, 256 MB quad-SPI Flash, 4 GB SD kartica
- Ugrađen USB-JTAG programator
- 10/100/1000 Ethernet priključak
- USB OTG 2.0 i USB-UART adapter
- PS i PL I/O proširenje
- HDMI i VGA priključci, ugrađen OLED prikaznik
- I<sup>2</sup>S Audio kodek

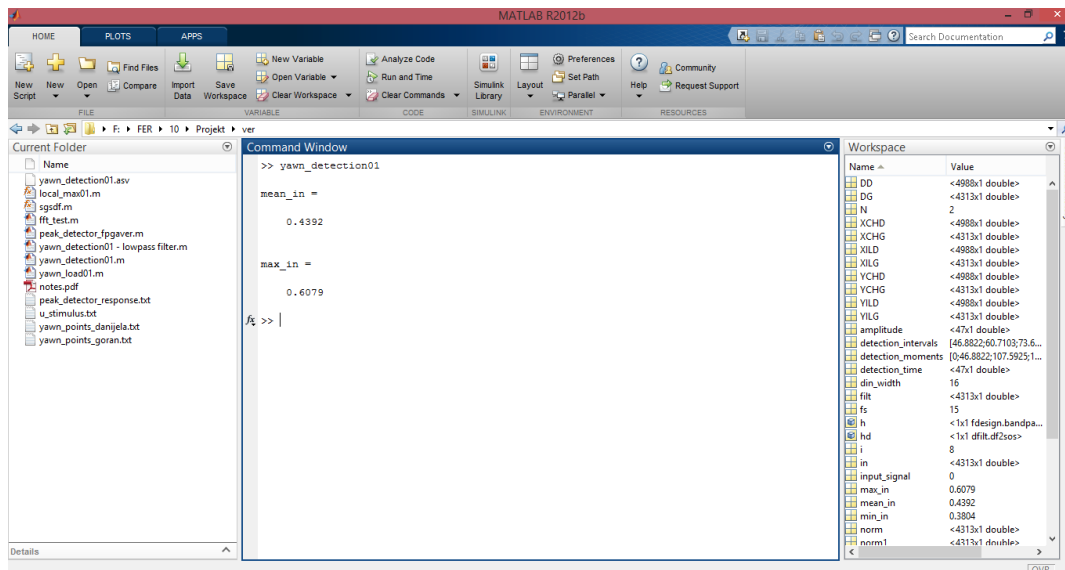
Za sustav je razvijen poseban operacijski sustav Xilinx. To je sustav s grafičkim sučeljem baziran na Ubuntu 12.04 LTS distribuciji za Zync-7000 seriju čipova. Primarna namjena mu je platforma za razvijanje miješanih softversko-logičkih aplikacija.



Slika 15: Zedboard razvojni sustav

## 5.2. Matlab

Matlab je programski jezik visoke razine i razvojno okruženje primjenjivo u raznim područjima poput obrade signala, obrade slike, razvijanje algoritama za komunikacijske sustave, upravljačke sustave... Zbog brojnih gotovih funkcija za obradu signala koristi se u ovom radu za izradu modela sustava. Osim gotovih funkcija, Matlab dolazi sa uključenim specijaliziranim okruženjima za obradu signala poput *Signal Processing Toolboxa* i *DSP System Toolboxa* koji uključuju alate *Filter Builder* za izradu filtera i *Filter Design and Analysis* za analizu kreiranih filtera što znatno olakšava i ubrzava razvoj projekta.



Slika 16: Matlab razvojno okruženje

### 5.3. ModelSim

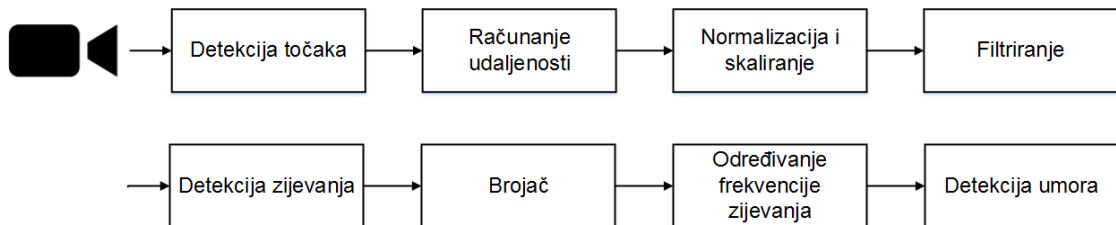
ModelSim je alat HDL simulaciju kojeg je razvila tvrtka Mentor Graphics. Koristi se za simulaciju sklopova opisanih VHDL-om, Verilogom ili SystemC-om. Može se koristiti uz druge alate za razvoj digitalnih sklopova poput Xilinx ISE ili Altere Quartus ili samostalno. U ovom radu koristit će se za *debugging* i simulaciju sklopova opisanih u VHDL-u.

### 5.4. Xilinx Vivado

*Vivado Design Suite* je Xilinxovo besplatno razvojno okruženje za razvoj FPGA sklopova. Ima podršku za Zedboard stoga će se koristiti za izradu ovog rada. *Vivado Design Suite* čine programi *Vivado*, *IP Integrator*, *Vivado High Level Synthesis* i *Software Development Kit*. *Vivado* je centralni program u kojem se obavlja RTL dizajn sklopova te sinteza, verifikacija i implementacija tih sklopova. On uključuje i *IP Integrator* koji objedinjuje Xilinxovu bazu IP jezgri te *Software Development Kit* za razvijanje programa u višim programskim jezicima. U ovom radu će se koristiti za izradu IP jezgre detektora koja će se AXI sabirnicom spojiti na ARM procesor.

## 6. Implementacija

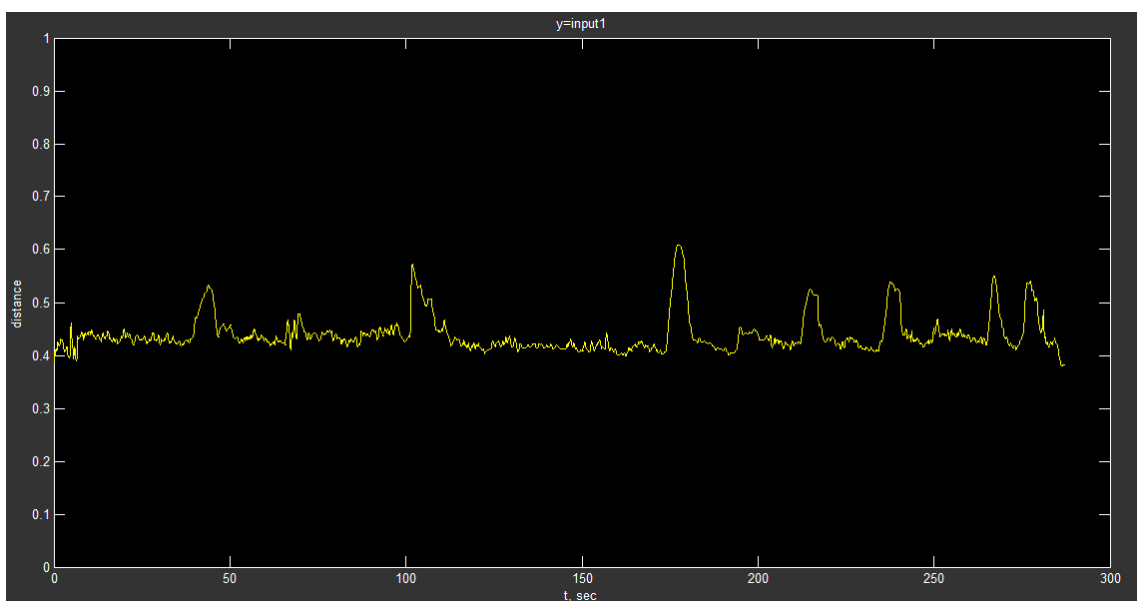
U sklopu rada implementiran je sustav za detekciju umora vozača baziran na analizi zijevanja. Shema implementacije nalazi se na 17.



Slika 17: Shema sustava za detekciju zijevanja

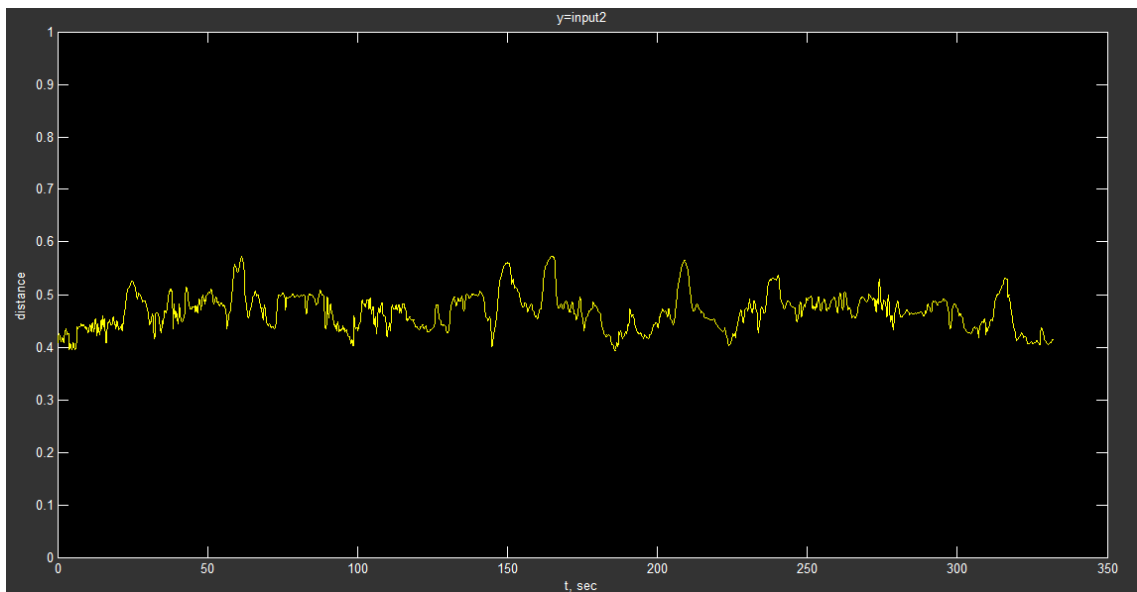
Podaci se dobivaju iz video signala frekvencijom od 15 Hz. Točke na licu koje se detektiraju i koriste su gornji vrh usne i dno brade. Korištenja širina riječi za ulazne podatke je 16 bita. Iz tih točaka računa se njihova međusobna udaljenost. Nakon toga dobiveni signal potrebno je normalizirati i skalirati na raspon od 0 do 1 zbog korištenja frakcionalne aritmetike navedenog raspona u VHDL implementaciji. Dobiveni signal provlači se kroz filter koji je projektiran kao pojasno propusni filter desetog reda. Određivanje praga radi se u svrhu detekcije zijevanja. Brojač se koristi za mjerenje vremena između dva zijevanja. Izlaz iz brojača koristi se za određivanje frekvencije zijevanja. Postavljanjem praga frekvencije zijevanja ostvaruje se detekcija umora.

Za testiranje će se koristiti dva ulazna signala prikazana na slikama 18 i 19.



Slika 18: Ulazni signal 1





*Slika 19: Ulazni signal 2*

U oba signala ispitanici su okrenuti prema kameri. Tijekom snimanja su pričali, smijali se, pjevali, šutili i zijevali. Prvi ispitanik je zijevao u periodima od 32. do 41. sekunde, od 168. do 176. sekunde, od 208. do 213., od 231. do 236., 260. do 265 i od 271. do 275. sekunde. Ti periodi se manifestiraju kao „šiljci“ odnosno povećanje razmaka promatranih točaka u ulaznom signalu što je naočigled vidljivo. Uzrok ostalim „šiljcima“ koji se pojavljuju je smijanje i pričanje ispitanika. Drugi ispitanik ima značajno nepravilniji signal zbog manje udaljenosti između promatranih točaka dok zijeva, smije se i priča. On zijeva u periodima od 57. do 63. sekunde, od 145. do 155., od 160. do 170., od 205. do 211., od 235. do 241., od 312. do 317. sekunde.

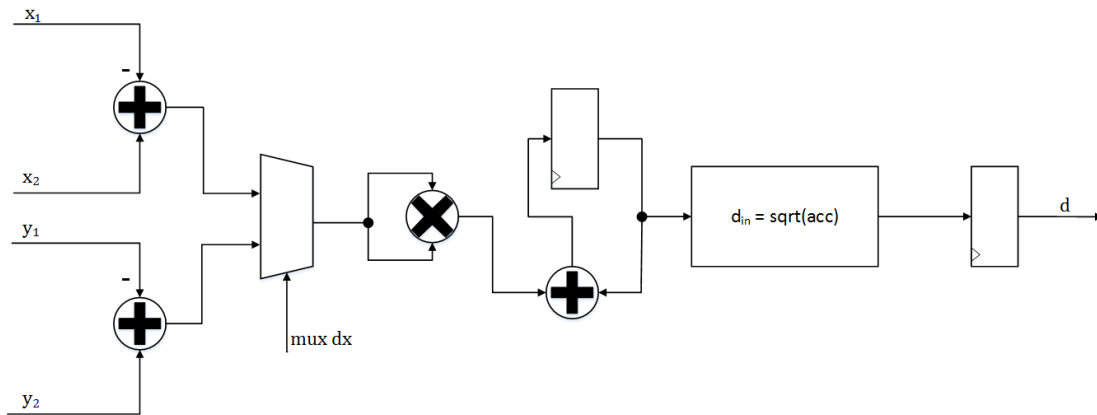
U sljedećim podpoglavljima dodatno će se razraditi svaki element sustava i prikazati testiranja i međurezultat svake komponente.

## 6.1. Računanje udaljenosti

Udaljenost se računa po standardnoj formuli za geometrijsku udaljenost:

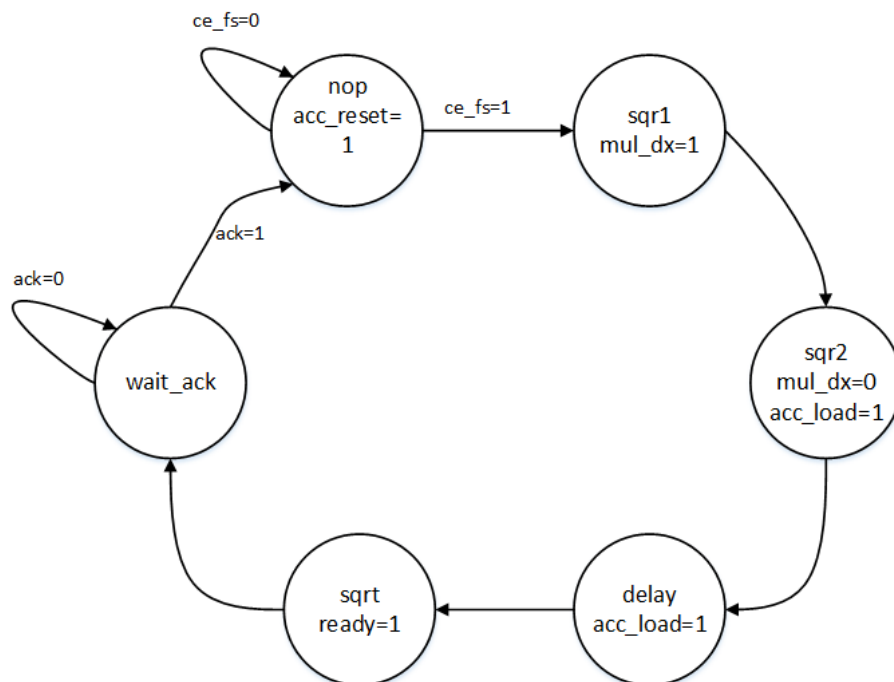
$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \quad (7)$$

Za računanje udaljenosti razvijen je modul koji na ulaz prima dva para koordinata  $(x_1, y_1)$  i  $(x_2, y_2)$  a na izlazu daje udaljenost  $d$ . Shema sklopa nalazi se na slici 20.



Slika 20: Sklop za računanje udaljenosti

Za sklop je bilo potrebno koristiti samo jedno množilo. Stoga je napravljen multipleksor ulaza u množilo. Signalom `mux_dx` odabire se hoće li na ulazu biti razlika ulaza `x` ili `y`. Izlaz množila spojen je na zbrajalo akumulatora koji je spojen na ulaz sklopa za računanje korijena. Radom sklopa upravlja se Mooreovim automatom stanja. Graf stanja nalazi se na slici 21.

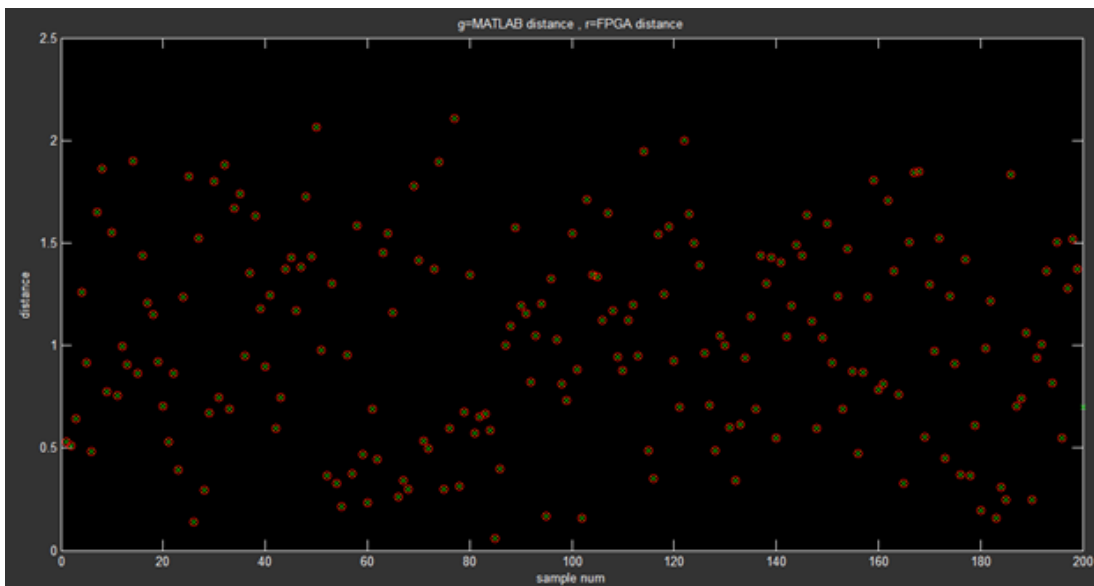


Slika 21: Automat stanja sklopa za računanje udaljenosti

U početnom stanju `nop`, automat drži signal `acc_reset` u jedinici kako bi akumulator bio spreman za novi upis vrijednosti. U stanju `sqr1` signalom `mul_dx` na množilo se kroz multipleksor dovodi razlika `x` koordinata, a u stanju `sqr2` razlika `y` koordinata. Signalom `acc_load` u stanjima `sqr2` i `delay` omogućuje se upis vrijednosti sa izlaza iz množila u akumulator. U stanju `sqrt_ready` se postavlja

u jedan i signalizira se sklopu za računanje korijena da je ulaz spreman. Zatim se prelazi u stanje čekanja završetka operacije korjenovanja wait\_ack u kojem se ostaje dok signal ack ne bude jedan. Prijelazi bez uvjeta događaju se na rastući brid takta.

Ulaz modula su frakcionalni brojevi u rasponu od minus jedan do jedan. Izlaz je frakcionalan broj u rasponu od nula do korijena iz osam. Modul koristi prethodno razvijen sklop za računanje korijena. Za simulaciju napisan je modul distance2d\_tb. Simulacija se izvršava u programu Modelsim. Napisane su i simulacijske datoteke tb.do,compile.do i wave.do. Izlaz simulacije je tekstualna datoteka s vrijednostima udaljenosti parova točaka. Izlazni podaci učitavaju se u Matlab skriptu gdje se iscrtavaju rezultati i provjerava točnost. Primjer iscrtavanja nalazi se na slici 22. Crvene oznake predstavljaju udaljenosti izračunate u VHDL-u a zelene oznake udaljenosti izračunate u Matlabu.



Slika 22: Usporedni rezultati modula u VHDL-u i Matlabu

Izlaz iz VHDL-a kasni za ulazom za (*širina ulazne riječi+4*) . Kašnjenje je prihvatljivo jer je frekvencija uzorkovanja mala i brzina izračuna nije prioritet. VHDL modul u usporedbi s Matlabom odstupa u točnosti na bitu najmanje težine, što uzeći u obzir korištenu širinu riječi od 16 bita, nije značajna greška i može se zanemariti.

## 6.2. Normalizacija i skaliranje

Cilj normalizacije i skaliranja je smještanje svih vrijednosti signala unutar prozora od 0 do 1 uz što veću dinamiku. U tu svrhu u Matlabu su razvijena dva modula. Prvi modelira idealno rješenje normalizacije i skaliranja i radi po formuli:

$$in_{norm} = \frac{in - \min(in)}{\max(in) - \min(in)} \quad (8)$$

Drugi je eksperimentalno modeliran prema prosječnim vrijednostima koji se pojavljuju u ulaznim testnim signalima. Postupak se obavlja u sljedeća četiri koraka:

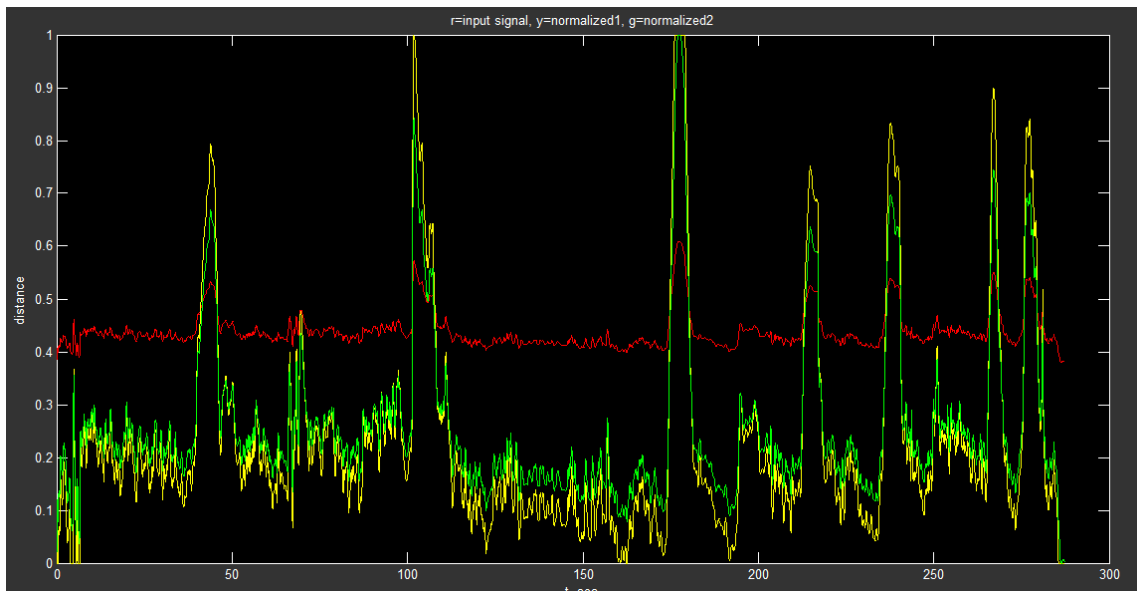
$$in_1 = in - 0.40$$

$$in_2 = \begin{cases} 0, & in_1 < 0 \\ in_1, & \text{inače} \end{cases}$$

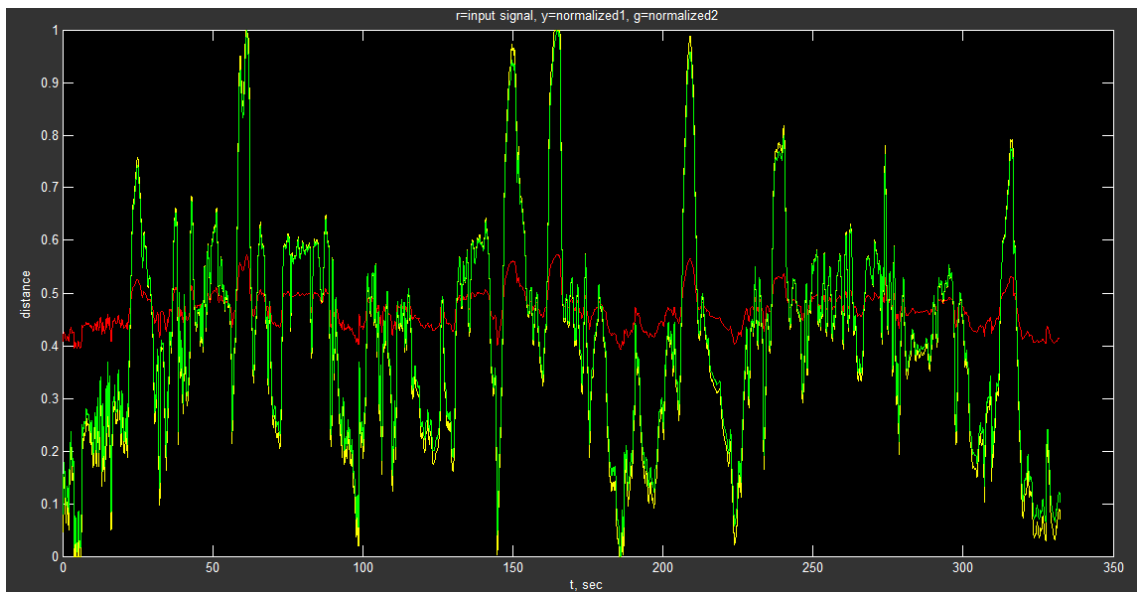
$$in_3 = in_2 \cdot 6$$

$$in_{norm} = \begin{cases} 1, & in_3 > 1 \\ in_3, & \text{inače} \end{cases}$$

Drugi postupak koristi aritmetiku sa zasićenjem koja osigurava da signal bude u odgovarajućem rasponu. Usporedni graf oba rješenja nalazi se na slici 23 i 24.



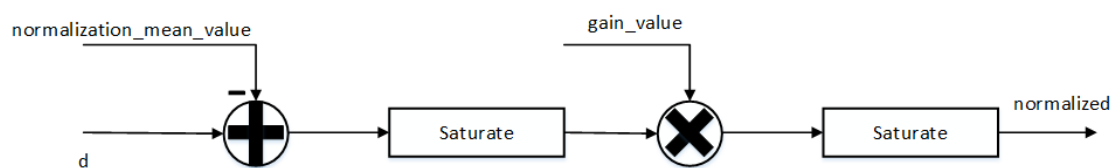
Slika 23: Prikaz izvornog i normaliziranog ulaznog signala 1



Slika 24: Prikaz izvornog i normaliziranog ulaznog signala 2

Na grafu se može vidjeti ulazni signal crvene boje, signal normaliziran prvim postupkom (žuta boja) i signal normaliziran drugim postupkom (zelena boja). Zbog korištenja aritmetike sa zasićenjem drugim postupkom ostvaruju se bolji rezultati u dinamici signala što se može vidjeti na većim vrijednostima lokalnih maksimuma.

U VHDL-u je implementiran drugi postupak normalizacije i skaliranja. Ulaz u sklop je udaljenost izračunata u modulu distance\_2d a izlaz je normalizirani i skalirani signal. Shema implementiranog sklopa nalazi se na slici 20.



Slika 25: Sklop za normalizaciju

Normalization\_mean\_value iznosi 0,40 a gain\_value iznosi 6. Konstante su definirane u paketu yawndetector\_pkg.

### 6.3. Filtar

FIR filtri (engl. *Finite Impulse Response*) i IIR filtri (engl. *Infinite Impulse Response*) čine dvije glavne skupine digitalnih filtara koji se koriste kod obrade

informacija. Glavna razlika FIR i IIR filtra je ta što FIR filter ima konačan impulsni odziv što mu daje prednost nad IIR filtrom. Općenita jednadžba prijenosne funkcije digitalnog filtra je:

$$H(z) = \frac{B(z)}{A(z)} = \frac{b_0 + b_1 z^{-1} + b_2 z^{-2} + \dots + b_N z^{-N}}{1 + a_1 z^{-1} + a_2 z^{-2} + \dots + a_M z^{-M}} \quad (9)$$

Red filtra je veći broj između N i M.

Filtar koji će se koristiti oblikovan je pomoću Matlabovog alata za dizajn filtera *fdesign*<sup>5</sup> uz korištenje *Signal Processing Toolboxa*. Za dizajn filtera također može se koristiti *DSP System Toolbox* koji ima grafičko sučelje za specifikaciju parametara filtra.

*Fdesign* je objekt za specifikaciju filtra. Dizajn filtra radi se u četiri koraka. Prvi korak je pomoću *fdesign.response* metode konstruirati specifikaciju filtra. U drugom koraku potrebno je odabrati metodu dizajna za specificirani filter. U trećem koraku koristi se metoda *design* koja iz specifikacije konstruira objekt filtra. U četvrtom koraku može se koristiti *FVTool* za provjeru i analizu dobivenih rezultata.

*response* metoda u *fdesign.response* naredbi označava jednu od opcija za definiranje odziva filtra. Ona može biti: *bandpass*, *bandstop*, *lowpass*, *highpass*, *hilbert*, *interpolator*... Za potrebe rada koristit će se *bandpass* metoda za dizajn filtra. Metoda prima dva argumenta: specifikaciju i vrijednosti parametara. Specifikacije za tu metodu su:

1. 'Fst1, Fp1, Fp2, Fst2, Ast1, Ap, Ast2'
2. 'N, F3dB1, F3dB2'
3. 'N, F3dB1, F3dB2, Ap'
4. 'N, F3dB1, F3dB2, Ast'
5. 'N, F3dB1, F3dB2, Ast1, Ap, Ast2'
6. 'N, F3dB1, F3dB2, BWp'
7. 'N, F3dB1, F3dB2, BWst'
8. 'N, Fc1, Fc2'
9. 'N, Fc1, Fc2, Ast1, Ap, Ast2'
10. 'N, Fp1, Fp2, Ap'

<sup>5</sup> <http://www.mathworks.com/help/dsp/ref/fdesign.html>

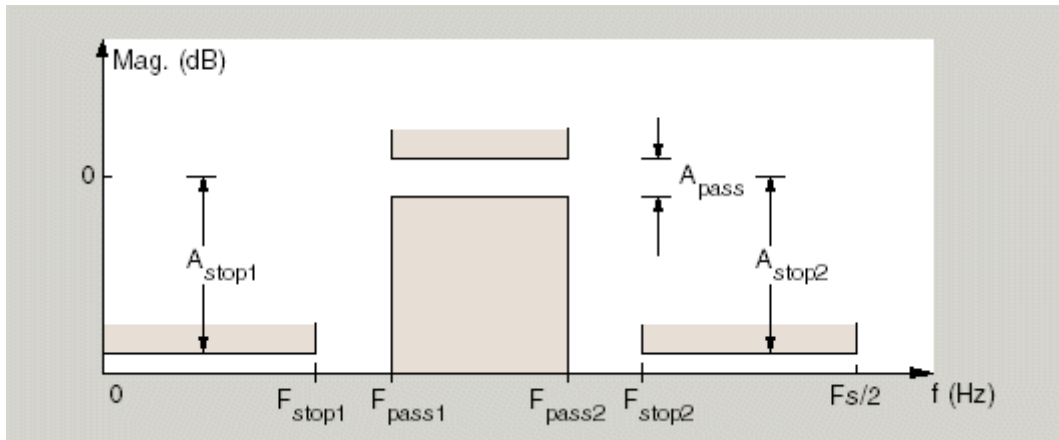
11. 'N, Fp1, Fp2, Ast1, Ap, Ast2 '
12. 'N, Fst1, Fp1, Fp2, Fst2 '
13. 'N, Fst1, Fp1, Fp2, Fst2, C '
14. 'N, Fst1, Fp1, Fp2, Fst2, Ap '
15. 'N, Fst1, Fst2, Ast '
16. 'Nb, Na, Fst1, Fp1, Fp2, Fst2 '

Značenje argumenata je sljedeće:

- Ap — dozvoljena valovitost izlaznog signala
- Ast1 — gušenje prvog pojasa u decibelima
- Ast2 — gušenje drugog pojasa u decibelima
- BWp — propusnost središnjeg pojasa filtra zadana pomoću normalizirane frekvencije
- BWst — propusnost dva pojasa gušenja zadana pomoću normalizirane
- C — Zastavica koja omogućuje postavljanje valovitosti propusnog pojasa ili gušenja u prvom ili drugom nepropusnom pojasu
- F3dB1 — prva granična frekvencija gušenja zadana 3 dB ispod frekvencije propusta izražena normaliziranom frekvencijom (IIR filtri)
- F3dB2 — druga granična frekvencija gušenja zadana 3 dB iznad frekvencije propusta izražena normaliziranom frekvencijom (IIR filtri)
- Fc1 — prva granična frekvencija gušenja zadana 6 dB ispod frekvencije propusta izražena normaliziranom frekvencijom (FIR filtri)
- Fc2 — druga granična frekvencija gušenja zadana 3 dB iznad frekvencije propusta izražena normaliziranom frekvencijom (FIR filtri)
- Fp1 — frekvencija na rubu početka pojasa propusta izražena u normaliziranoj frekvenciji
- Fp2 — frekvencija na rubu kraja pojasa propusta izražena u normaliziranoj frekvenciji
- Fst1 — frekvencija na početnom rubu pojasa propusta (unutar propusta) izražena u normaliziranoj frekvenciji
- Fst2 — frekvencija na krajnjem rubu pojasa propusta (unutar propusta) izražena u normaliziranoj frekvenciji
- N — red FIR filtra ili red brojnika i nazivnika IIR filtra
- Na — red nazivnika IIR filtra

- Nb — red brojnika IIR filtra

Grafički prikaz parametara nalazi se na slici 24.



Slika 26: Grafički prikaz parametara dizajna filtra. Slika preuzeta sa <http://www.mathworks.com/help/signal/ref/fdesign.bandpass.html>

Regije između  $F_{stop1}$  i  $F_{pass1}$  te  $F_{pass2}$  i  $F_{stop2}$  su prijelazne regije gdje odziv filtra nije eksplicitno definiran.

Pomoću metode `designmethods` mogu se vidjeti dostupne metode dizajna filtra. Metodom `design` objekt specifikacije filtra prevodi se u objekt filtra. Metoda može primiti više različitih argumenata. Ako se metodi proslijedi samo specifikacija filtra, ona će optimizacijskim metodama zaključiti koji filter je prikladan.

Naredbom `fdesign.bandpass('N,Fst1,Fp1,Fp2,Fst2,Ap',10,0.000009,0.0002,0.02,0.021,5)`

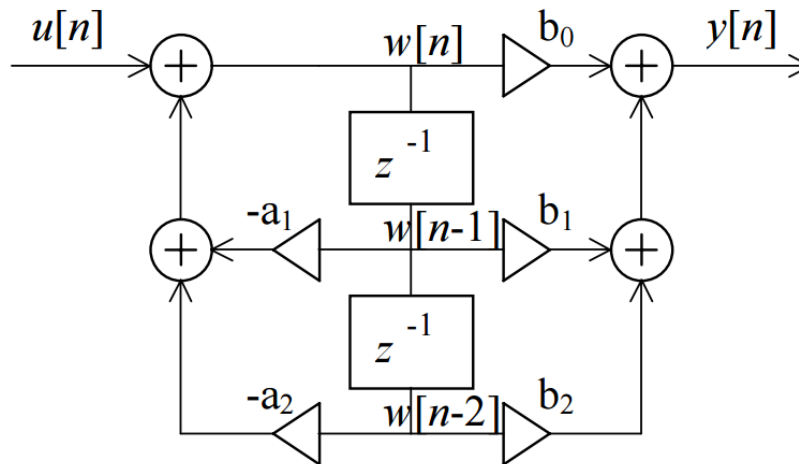
alatu su zadani parametri pojasno propusnog filtra. Red filtra postavljen je na deset. Okvirni iznos više frekvencije gušenja dobiven je analizom grafova ulaznih signala. Za najvišu frekvenciju zijevanja uzima se 0.083Hz (period 12 sekundi). Normalizirani iznos te frekvencije računa se prema formuli

$$\omega_c = \frac{2\pi f_c}{f_s} \quad (10)$$

gdje su  $f_c$  frekvencija gušenja a  $f_s$  frekvencija uzorkovanja (15 Hz). Iznos normalizirane frekvencije je  $0.0347 \pi \text{ rad/uzorak}$ . Uz taj početni podatak, metodom pokušaja i pogrešaka građen je pojasno propusni filter. Uz zadane parametre, alat za dizajn filtra može realizirati IIR filter. Rezultat je bikvadratni

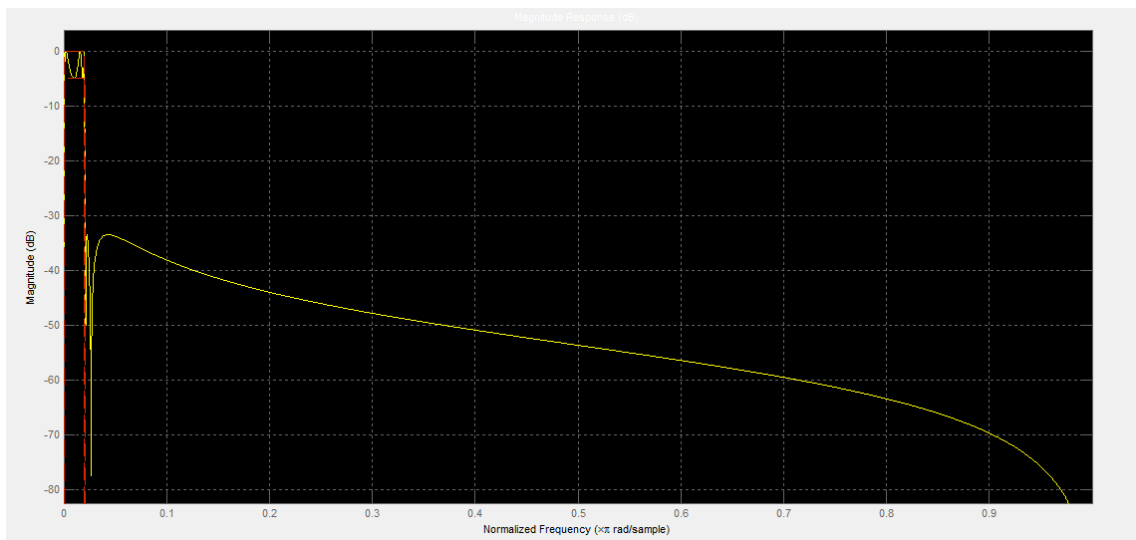


filtrar realiziran u direktnoj formi II. Shema jednog bloka bikvadratnog filtra nalazi se na slici 27.



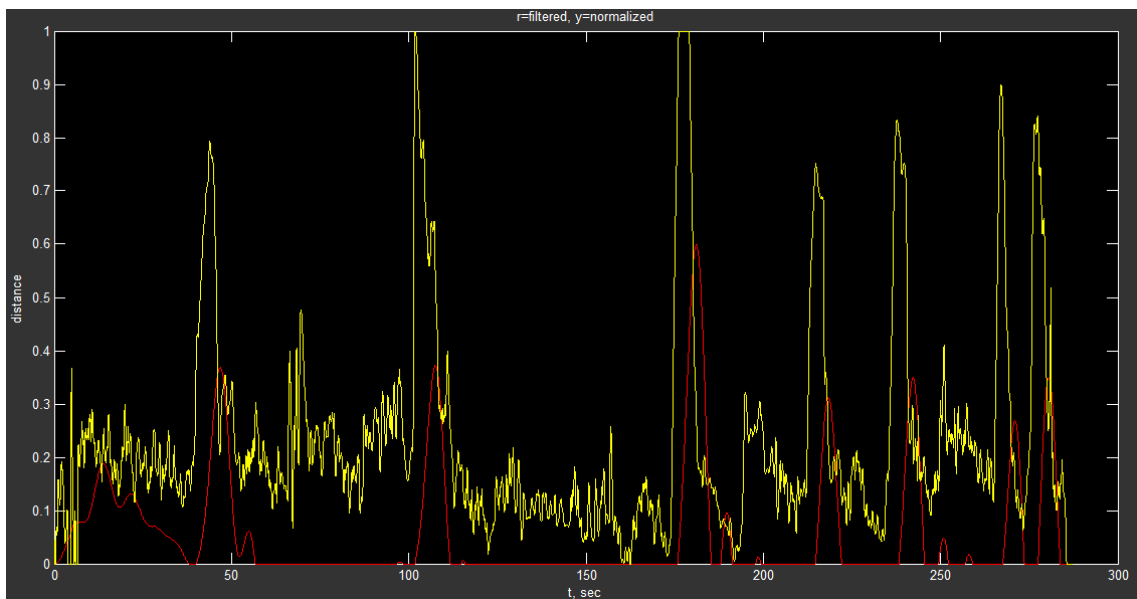
Slika 27: Shema bikvadratnog filtra[5]

Prijenosna funkcija filtra nalazi se na slici 28.

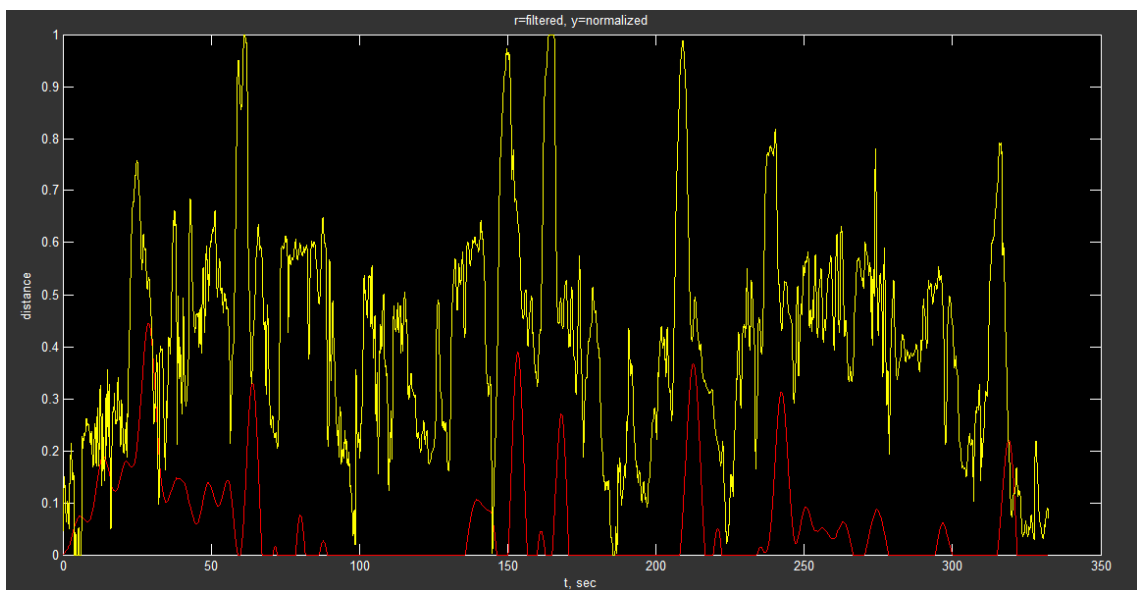


Slika 28: Prijenosna funkcija realiziranog filtra

Na slici 29 i 30 mogu se vidjeti ulazni i filtrirani signali za oba testna primjera. Filtrirani signal još uvijek sadrži informacije koje ne označavaju zijevanje tako da će se u daljnjoj obradi morati dodatno izbaciti nepotrebne informacije.



Slika 29: Ulazni i filtrirani signal ulaznog signala 1



Slika 30: Ulazni i filtrirani signal ulaznog signala 2

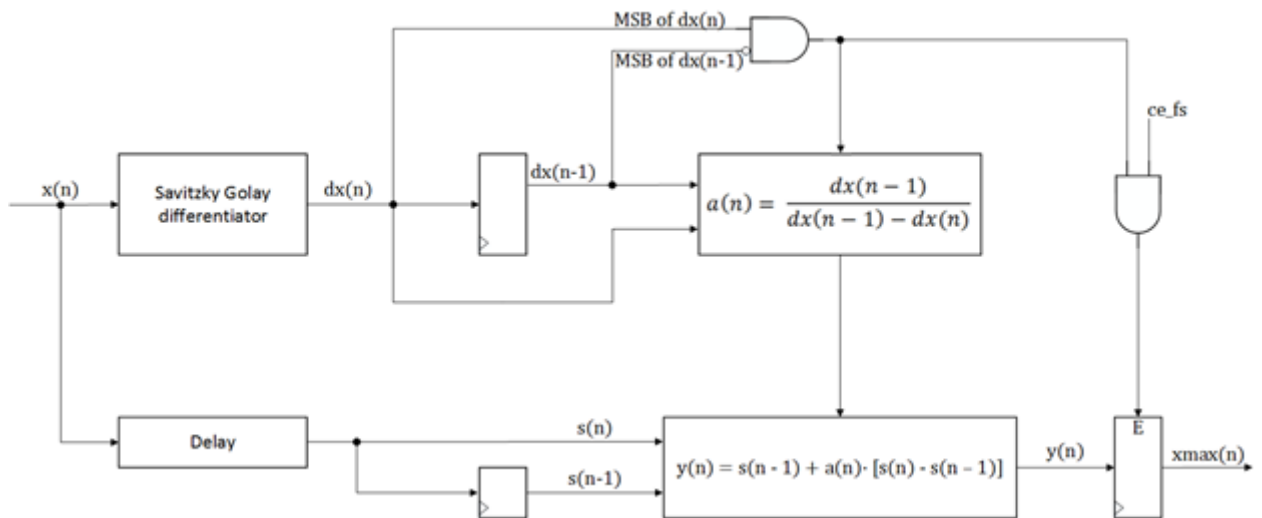
#### 6.4. Detekcija zijevanja

Detekcija zijevanja ostvaruje se pomoću dva sklopa: detektor lokalnog maksimuma i sklop za detekciju zijevanja. Detektor lokalnog maksimuma će izdvojiti vršne vrijednosti ulaznog signala a sklop za detekciju zijevanja će

pomoću zadanog praga odrediti koje od tih vršnih vrijednosti predstavljaju zijevanje. U sljedeća dva podpoglavlja opisani su navedeni sklopovi.

### 6.4.1. Detektor lokalnog maksimuma

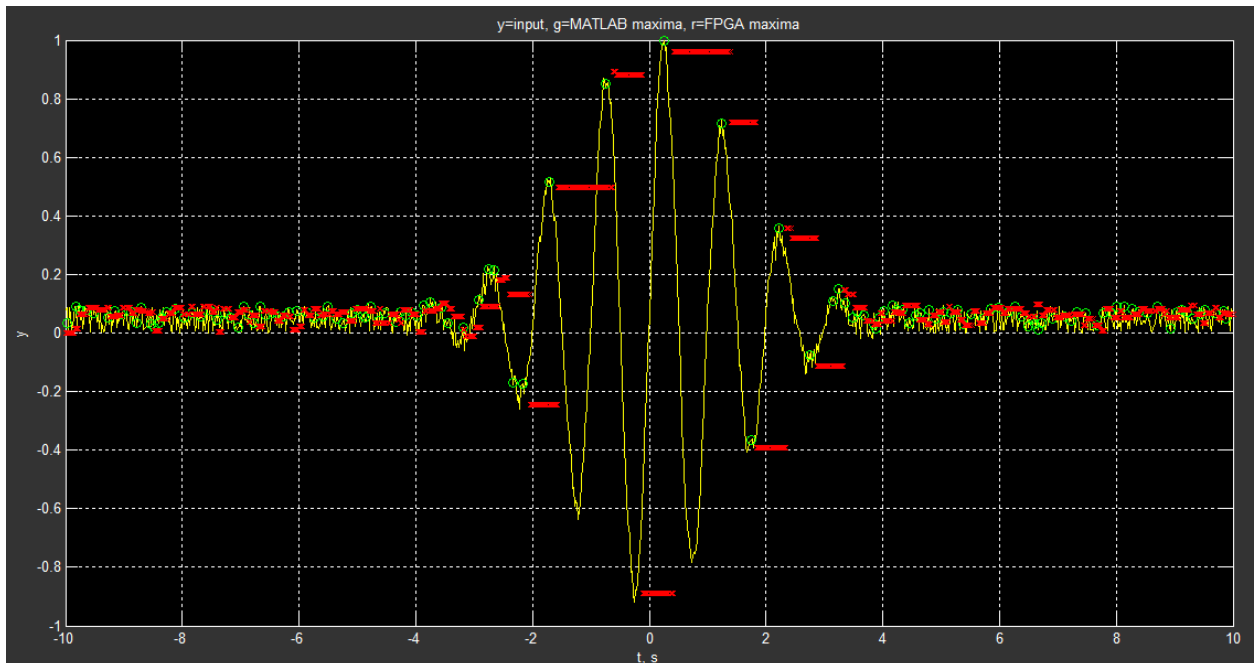
Sklop za detekciju lokalnog maksimuma na ulazu prima vrijednosti funkcije i signal da je ulaz spreman a na izlazu daje vrijednost zadnjeg detektiranog maksimuma. Shema sklopa nalazi se na slici 31.



Slika 31: Shema detektora maksimuma

Sama detekcija maksimuma obavlja se preko logičkog „I“ sklopa koji uspoređuje predznake derivacije trenutnog uzorka s derivacijom prošlog uzorka koja je zapisana u registru. Ako je došlo do promjene predznaka iz pozitivnog u negativni, na tom mjestu se nalazi maksimum. Derivacija ulaznih uzoraka obavlja se Savitzky-Golay filtrom. *Delay* sklop služi za kašnjenje ulaznih uzoraka zbog kašnjenja izračuna derivacije. Kada se detektira maksimum, sklop aktivira dio za interpolaciju vrijednosti maksimuma. Pomoću izraza  $a(n)$  i razlike uzastopnih uzoraka moguće je izračunati vrijednost maksimuma između ta dva uzorka.

Razvijen je modul `peak_detector` s gore opisanom funkcionalnosti. Ulaz i izlaz modula je frakcionalan broj u rasponu od minus jedan do jedan. Modul koristi prethodno razvijeni sklop za dijeljenje, sklop za računanje derivacije i paket u kojem su definirani potrebni tipovi i u kojem su nabrojani koeficijenti filtra. Za simulaciju napisan je modul `peak_detector_tb`. Simulacija se izvršava u programu `Modelsim`. Napisane su i simulacijske datoteke `tb.do`, `compile.do` i `wave.do`. Izlaz simulacije je tekstualna datoteka s vrijednostima detektiranih maksimuma. Izlazni podaci učitavaju se u Matlab skriptu gdje se iscrtavaju rezultati i provjerava točnost. Primjer iscrtavanja nalazi se na slici 32. Ulazna funkcija označena je žutom bojom, crvene oznake predstavljaju detektirane maksimume iz VHDL-a a zelene oznake su pravi maksimumi izračunati u Matlabu.



Slika 32: Primjer provjere rješenja

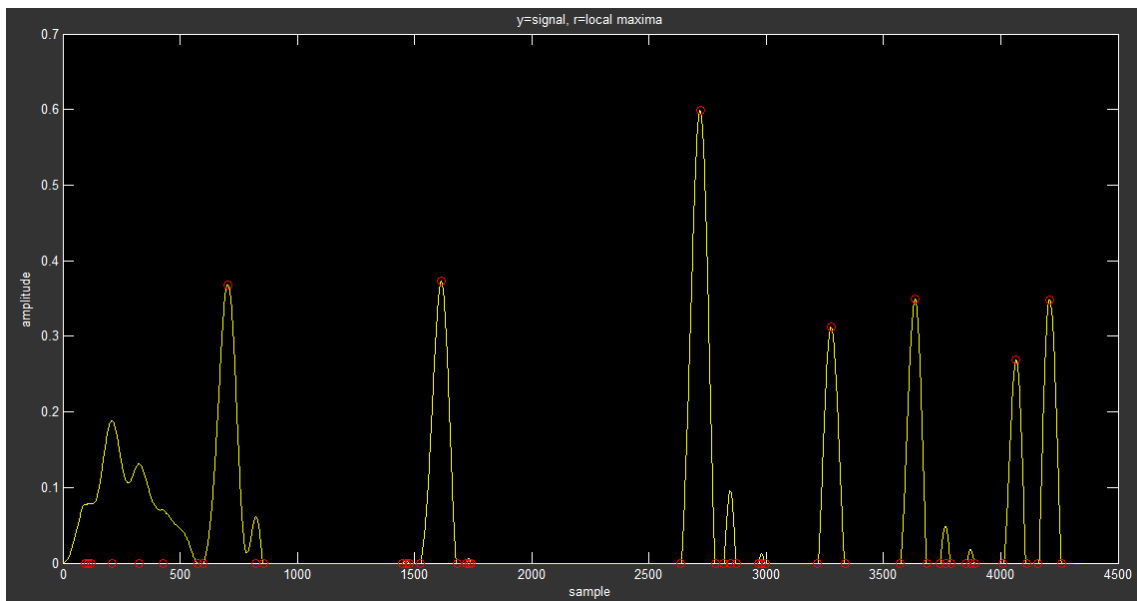
Izlaz iz VHDL-a kasni za ulazom za sedam uzoraka (na slici 32 se vidi kašnjenje detekcije za malo vrijeme). VHDL modul u usporedbi s Matlabom odstupa u točnosti na bitu najmanje težine, što uzeći u obzir korištenu širinu riječi od 16 bita, nije značajna greška i može se zanemariti. Greška je veća što ulazni signal ima veći dinamički raspon. Zbog toga se signal ne može dobro očitati i interpolator ne može izračunati točnu vrijednost maksimuma. Ta greška se može vidjeti na slici 32 gdje crvene točke odstupaju u vrijednosti od zelenih. Ona se može smanjiti povećanjem frekvencije uzorkovanja. Realni ulazni signal neće imati velik dinamički raspon tako da dodatno povećanje uzorkovanja nije potrebno.

#### 6.4.2. Detektor zijevanja

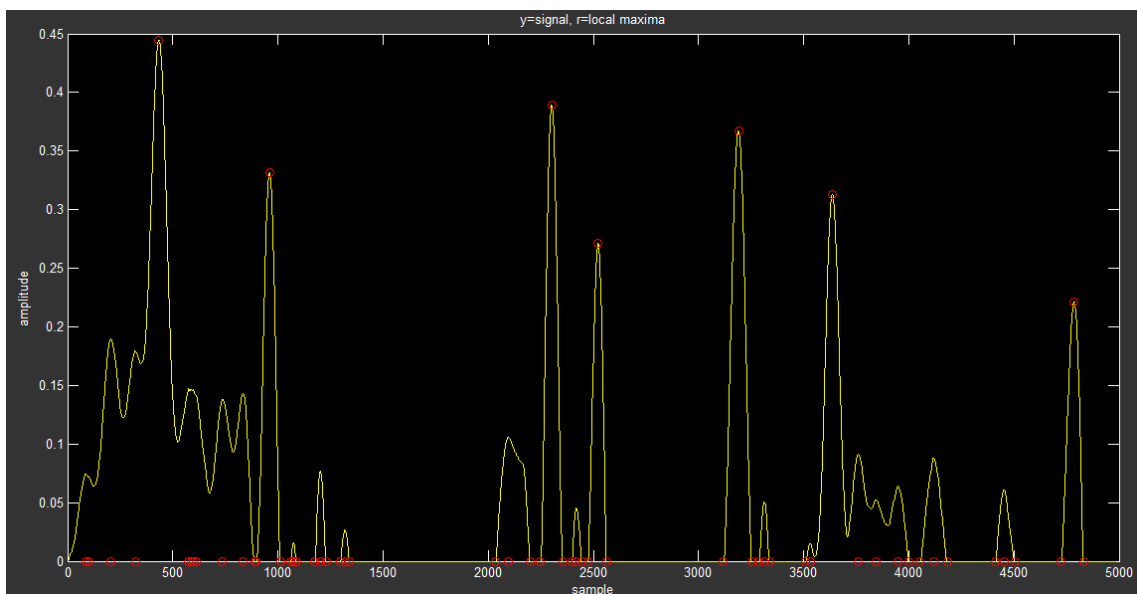
Detektor zijevanja uzima vrijednosti izlaza iz sklopa za detekciju lokalnog maksimuma, uspoređuje ih s preddefiniranim pragom i uklanja one vrijednosti koje su ispod njega. U Matlabu je to implementirano tako da sve vrijednosti signala ispod praga budu nula a sve iznad čuvaju svoju vrijednosti. Prag je eksperimentalno određen i iznosi 0,20.

U VHDL-u je korišten jednostavni komparator koji uspoređuje vrijednost praga s ulaznim signalom. Vrijednost praga definirana je u paketu yawndetector\_pkg. Ulaz u VHDL sklop je detektirani lokalni maksimum iz prethodnog sklopa u lancu a izlaz je signal da je zijev detektiran.

Na slikama 33 i 34 mogu se vidjeti rezultati rada oba sklopa. Vršne vrijednosti signala su izdvojene i one vrijednosti koje nisu iznad praga izjednačene s nulom. U prvom ulaznom signalu detektirano je sedam zjevova iako je ispitanik zijevao šest puta. Pogrešno detektirani zijev je drugi po redu. U tom trenutku ispitanik je glasno pričao i izračunata udaljenost između promatranih točaka lica i brzina promjene odgovarale su frekvenciji zijevanja što se može vidjeti i u ulaznom signalu. U drugom signalu ispitanik je zijevao šest puta dok je detektirano sedam zjevanja. Pogrešno detektirani zijev je prvi zijev na slici 34. U ulaznom i filtriranom signalu može se vidjeti velika prisutnost informacija frekvencijom sličnih zijevanju što uzrokuje navedenu grešku. U tom intervalu promatrana osoba se smijala i pričala.



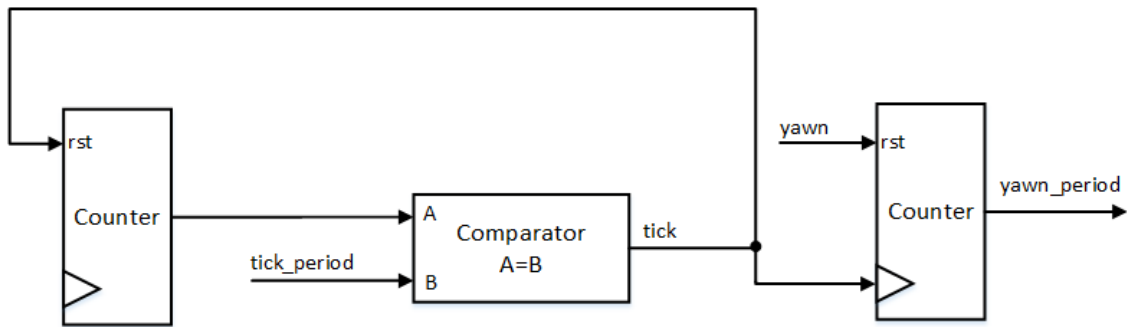
*Slika 33: Prikaz detektiranih zjevova prvog signala*



*Slika 34: Prikaz detektiranih zjevova drugog signala*

## 6.5. Brojač

Kako bi se odredila frekvencija zijevanja potrebno je napraviti sklop koji će mjeriti vrijeme između dva detektirana zijeva. U tu svrhu razvijen je VHDL sklop koji mjeri vrijeme između dva signala detekcije dobivena iz prethodnog sklopa. Shema sklopa nalazi se na slici 35.



Slika 35: Sklop za mjerenje perioda zijevanja

Sklop se sastoji od dva brojača. Prvi brojač radi na signal takta i njegov zadatak je da broji do 10 milisekundi. Brojanje do zadanog vremena ostvaruje se korištenjem komparatora koji sa zadanim uspoređuje izlaz brojača brojem taktova. Broj taktova zadan je sljedećom linijom koda:

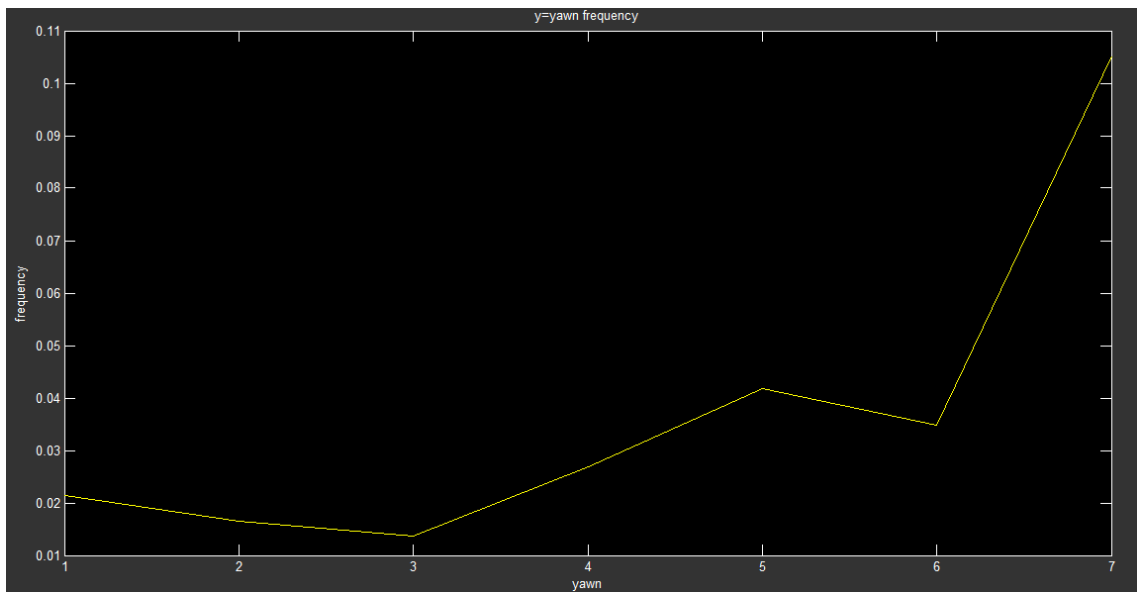
```
constant tick_period:positive:=(10*1000000/clock_period);
```

Tako zadana konstanta će označavati broj taktova u 10 milisekundi neovisno o periodu takta. Izlaz iz komparatora koristi se za resetiranje brojača i kao signal takta za drugi brojač. Zadaća drugog brojača je mjerenje broja signala iz komparatora između dva signala za detektirano zijevanje. Brojač se resetira signalom iz prethodnog sklopa (detektirano zijevanje). Izlaz je period zijevanja izražen u broju perioda od 10 milisekundi.

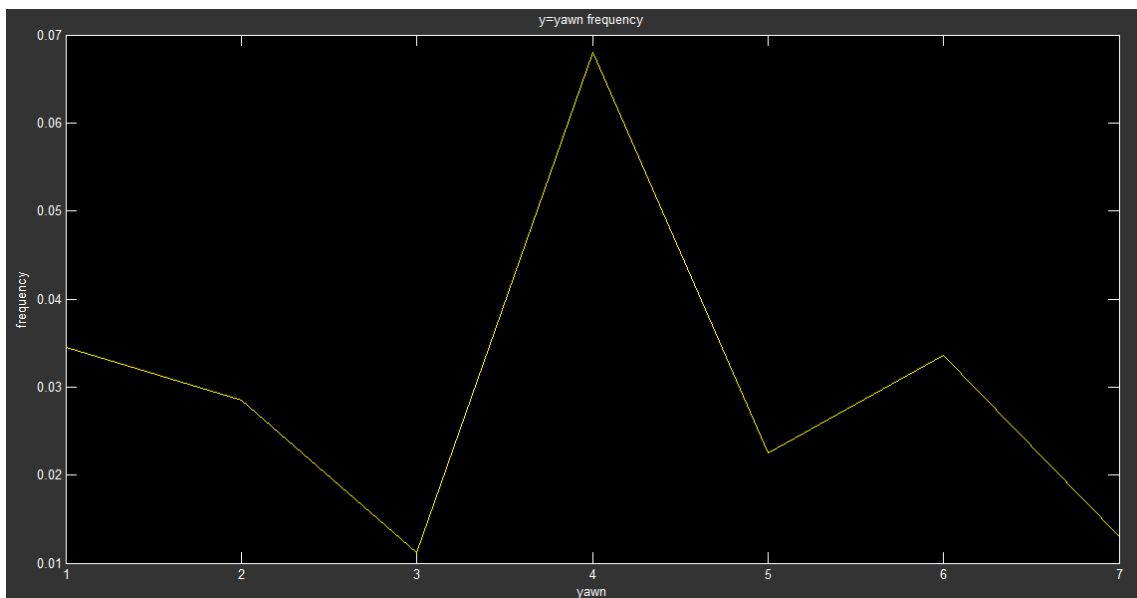
## 6.6. Frekvencija zijevanja i detekcija umora

Frekvencija zijevanja određuje se direktno iz prethodno izračunatog perioda. Zbog jednostavnosti postupka ona se računa između svakog detektiranog zijeva. Napredniji način bio bi da se pri izračunu koristi više perioda između detektiranih zijevanja te da frekvencija opada ako nakon određenog vremena zijevanje ne bude detektirano.

Problem koji se pojavljuje kod određivanja umora je postavljanje praga frekvencije iznad kojeg bi se smatralo da je promatrani vozač umoran. Za točno postavljanje praga potrebno je imati više uzoraka signala i po mogućnosti uzoraka snimljenih u stvarnim uvjetima. Testni uzorci korišteni u ovom radu su od osoba koji nisu stvarno bili umorni ni stvarno zijevali nego su to odglumili. Na slikama 36 i 37 nalaze se grafovi frekvencije zijevanja.



*Slika 36: Frekvencija zijevanja prvog vozača*



*Slika 37: Frekvencija zijevanja drugog vozača*

Može se vidjeti da oba ispitanika u jednom trenutku naglo krenu povećavati frekvenciju zijevanja. Kako u ulaznom signalu nije bilo puno detektiranih zijevanja, nije moguće točno odrediti srednju vrijednost frekvencije koja je prihvatljiva ispod koje bi se smatralo da vozač nije umoran.

Frekvencija koja je određena kao prag umora je 0.05 Hz, odnosno period zijevanja ispod kojeg se vozač smatra umornim je 20 sekundi. Ako se uzme u obzir da je prosječno trajanje zijevanja od četiri do šest sekundi, može se zaključiti da zijevanje navedenom frekvencijom uistinu znači da je vozač umoran. S takvim pragom, kod oba vozača detektira se umor. Kod prvog vozača alarm

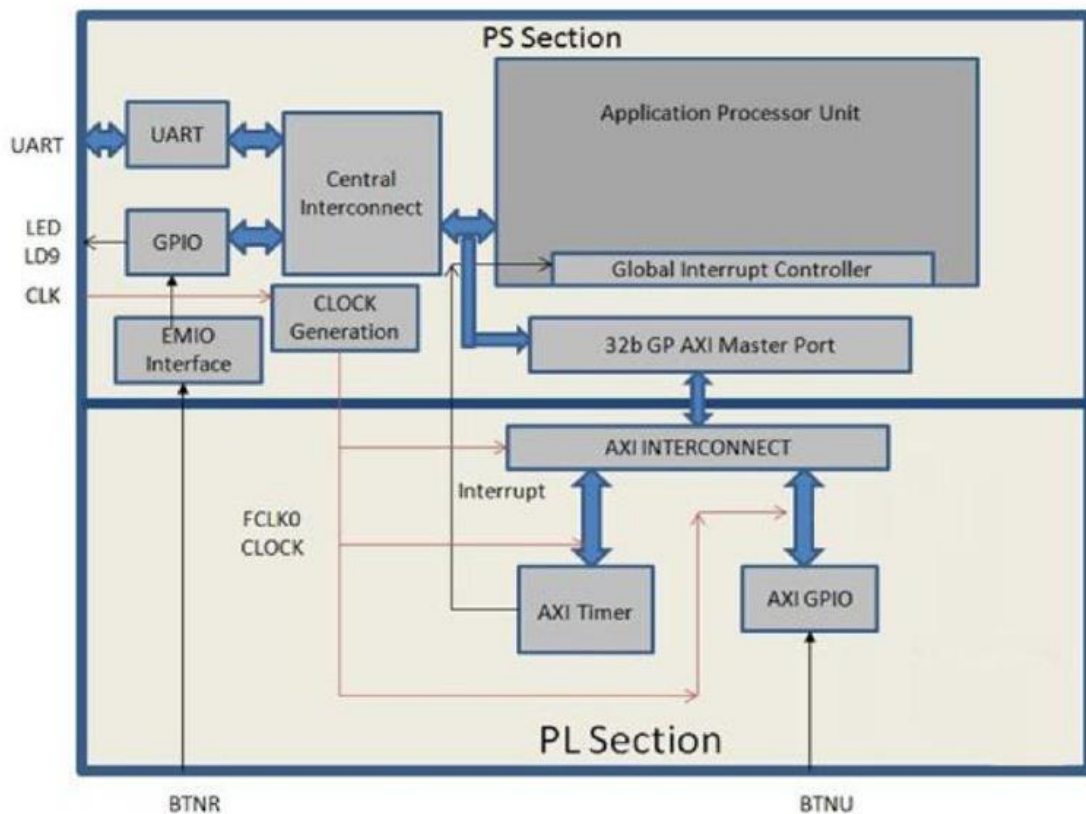


detekcije se aktivira nakon posljednjeg detektiranog zijeva (od 271. do 275. sekunde) a kod drugog vozača nakon trećeg (od 160. do 170. sekunde). U tim trenutcima ispitanici su glumili umor i u stvarnim uvjetima alarm detekcije bio bi pravovremen i ispravan. Lažno detektirana zijevanja u ovom slučaju nisu utjecala na konačni rezultat, no bez veće baze ulaznih signala snimljenih u laboratorijskim i stvarnim uvjetima ne može se točno znati kako bi se detektor ponašao.

## 6.7. Integracija u IP jezgru

ZedBoard razvojna ploča podržava razvoj na konceptu sustava *Zync-7000 All Programmable SoC*. Koncept kombinira mogućnosti razvoja temeljenog na dvojezrenom procesoru ARM Cortex-A9 i Xilinxove programabilne logike. ZedBoard sadrži FPGA uređaj iz porodice Artix-7.

Aplikacija se razvija u Xilinx Vivado razvojnom okruženju. On nudi mogućnosti razvoja za ARM procesor i programabilnu logiku. IP jezgre (engl. *Intellectual Property*) su korisničke komponente razvijene u nekom od jezika za opis hardvera zapakirane u cjelinu pogodnu za ugradnju u programabilnu logiku i komunikaciju s aplikacijskim procesorom. Na slici 38 može se vidjeti skica projekta za ZedBoard.



Slika 38: Skica projekta za ZedBoard

Programabilna logika komunicira s aplikacijskim procesorom putem AXI sabirnice. AXI (engl. *Advanced eXtensible Interface*) je dio ARM AMBA sabirnica za mikrokontrolere. Današnja verzija AXI sabirnice je AXI4 i koristi se od 2010. godine. Postoje tri tipa AXI sabirnice:

1. AXI4 – memorijski mapiran pristup visokih performansi
2. AXI4-Lite – memorijski mapiran pristup niskih performansi
3. AXI4-Stream – brzi prijenos toka podataka

AXI4-Lite i AXI4 odgovaraju zahtjevima za komunikaciju sa sklopom detektora. Kako će u stvarnom sustavu podaci pristizati 15 puta u sekundi, visoke performanse AXI4 sabirnice nisu potrebne stoga će se koristiti AXI4-Lite. Dodatna prednost te sabirnice je to što njena funkcionalnost zauzima manje prostora u programabilnoj logici.

Daljnji razvoj i korištenje integrirane komponente obavlja se u Vivado SDK alatu. U njemu se u programskom jeziku C izrađuje aplikacija za ARM procesor koja unutar svog koda poziva funkcije za komunikaciju s IP jezgrom. Kompletna aplikacija se iz SDK-a prenosi na ZedBoard razvojni sustav.

## 7. Zaključak

Razvijeni sustav za detekciju umora vozača funkcionira dobro i moguće ga je koristiti u stvarnom sustavu. Nedostatak je nedovoljna količina ulaznih testnih podataka zbog čega nije moguće bolje odrediti parametre za odluke u raznim dijelovima sustava. Konkretno, veći broj različitih ispitanika dao bi bolju procjenu srednje vrijednosti udaljenosti između promatranih točaka lica što bi dalo bolje rezultate kod normalizacije i skaliranja. Više snimljenih zijevanja dalo bi bolje podatke o frekvenciji promjena pri zijevanju iz čega bi se mogle točnije odrediti frekvencije gušenja kod filtra. Dodatna poboljšanja moguća su i kod određivanja praga detekcije zijeva i praga umora za frekvenciju zijevanja.

Cjelokupna detekcija umora zamišljena je tako da se bazira na analizi više fizioloških pojava na licu. Korištenjem više različitih izvora informacija o vozaču moguće je poništiti nedostatke svakog pojedinog dijela sustava. Dodatni izvori informacija mogu biti položaj glave, brzina pomicanja zjenica, brzina i učestalost zatvaranja kapaka, pritisak vozačeve ruke na volan ili najtočnija ali najmanje praktična metoda detekcije iz EEG signala.

Sustav je namijenjen za ugradnju u FPGA sklopove. Parametri sklopa su promjenljivi i mogu se proizvoljno zadati u bilo kojem trenutku. Cjelokupni sklop zapakiran je u IP jezgra i predviđen je za korištenje na Xilinxovoj Zync-7000 arhitekturi na kojoj bi se s ostalim komponentama spajao preko AXI sabirnice.

Primarna namjena sustava je detekcija umora kod vozača. Uz to sustav se može koristiti i za nadzor profesionalnih vozača koji se često sami radi većeg profita dovode do granica izdržljivosti. Sustavi za detekciju umora već postoje i implementirani su u raznim tipovima vozila u raznim oblicima (Volkswagenov „*Fatigue detection system*“, Audijev „*Rest Recommendation System*“, BMW-ov „*Active Driving Assistant*“, ...). Takvi sustavi znatno smanjuju rizik od nesreća uzrokovanih umorom i u skoroj budućnosti očekuje se povećanje broja prodanih automobila s ugrađenim sustavom detekcije umora.

## 8. Literatura

[1] Bryan van Hal: Real-time Stage 1 Sleep Detection and Warning System Using a Low-Cost EEG Headset, Grand State Valley University, travnja 2013.

[2] Qiang Ji, Zhiwei Zhu, Peilin Jan: Real-Time Nonintrusive Monitoring and Prediction of Driver Fatigue, IEEE, srpnja 2004.

[3] Thum Chia Chieh, Mohd Marzuki Mustafa, Aini Hussain, Edmond Zahedi, Burhanuddin Yeop Majlis: Driver Fatigue using Steering Grip Force, Universiti Kebangsaan Malaysia, 2003.

[4] Magdlena Krbot, Ana Branka Šefer: FPGA implementacija FIR filtra, FER, ožujka 2008.

[5] Davor Petrinović, Branko Jeren: Predavanja iz Digitalne obradbe signala, listopada 2014.

[6] Iva Anić: Vozački umor,  
<http://www.istrazime.com/prometna-psihologija/vozacki-umor/>, datum pristupa: 21.4.2015.

[7] Audi: Rest Recommendation System, <https://www.audi-mediacycenter.com/en/third-generation-dynamics-the-new-audi-tt-and-the-tts-3027/driver-assistance-systems-3079>, datum pristupa: 23.4.2015.

[8] BMW: Driving Assistant,  
[https://www.press.bmwgroup.com/global/pressDetail.html?title=bmw-model-upgrade-measures-taking-effect-from-the-summer-of-2013&outputChannelId=6&id=T0141144EN&left\\_menu\\_item=node\\_\\_4101](https://www.press.bmwgroup.com/global/pressDetail.html?title=bmw-model-upgrade-measures-taking-effect-from-the-summer-of-2013&outputChannelId=6&id=T0141144EN&left_menu_item=node__4101), datum pristupa: 23.4.2015.

- [9] Volkswagen: Fatigue Detection, [http://www.volkswagen.com.au/en/technology\\_and\\_service/technical-glossary/fatigue-detection.html](http://www.volkswagen.com.au/en/technology_and_service/technical-glossary/fatigue-detection.html), datum pristupa: 23.4.2015.
- [10] Bosch: Driving Safety Systems, [http://products.bosch-mobility-solutions.com/en/de/driving\\_safety/driving\\_safety\\_systems\\_for\\_passenger\\_cars\\_1/driver\\_assistance\\_systems/driver\\_assistance\\_systems\\_2.html#](http://products.bosch-mobility-solutions.com/en/de/driving_safety/driving_safety_systems_for_passenger_cars_1/driver_assistance_systems/driver_assistance_systems_2.html#), datum pristupa 23.4.2015.
- [11] Daimler: Attention Plus, <http://media.daimler.com/dcmmedia/0-921-658892-1-1147698-1-0-0-1147922-0-1-11702-0-0-1-0-0-0-0-0.html?TS=1266506682902>, datum pristupa: 23.4.2015.
- [12] Xilinx: ZedBoard Specification, <http://zedboard.org/product/zedboard>, datum pristupa: 10.6.2015.
- [13] FER, ZEMRIS: Laboratorij računalnog inženjerstva 2: ZedBoard razvojni sustav, svibanj 2014.
- [14] Xilinx: AXI Reference Guide, [http://www.xilinx.com/support/documentation/ip\\_documentation/axi\\_ref\\_guide/v13\\_4/ug761\\_axi\\_reference\\_guide.pdf](http://www.xilinx.com/support/documentation/ip_documentation/axi_ref_guide/v13_4/ug761_axi_reference_guide.pdf), datum pristupa 20.6.2015.

# Naslov

Algoritmi za detekciju umora vozača

## Sažetak

U radu je opisan sustav za detekciju umora baziran na analizi zijevanja. Sustav uzima koordinate vrha gornje usne i dna brade iz čega računa njihovu udaljenost. Ta udaljenost se zatim provlači kroz sklop za normalizaciju i skaliranje u interval od 0 do 1 nakon čega se iz signala filtrom uklanjaju nepotrebne informacije. Iz filtriranog signala sklopom za detekciju lokalnog maksimuma izdvajaju se vršne vrijednosti. Detektor zijeva zatim uzima te vršne vrijednosti, uspoređuje ih sa zadanim pragom te signalizira ukoliko je zijev detektiran. Brojač vremena pamti trenutke kada je zijevanje detektirano te na izlaz daje intervale između svakog zijeva. Pomoću tih intervala određuje se frekvencija zijevanja. Zadavanjem praga frekvencije zijevanja kod umorne osobe ostvaruje se detekcija umora vozača. Detekcija bazirana na analizom zijevanja je funkcionalna i može se koristiti u stvarnim uvjetima. No, takav sustav predviđen je kao dio većeg sustava koji će donositi odluke bazirane na više različitih izvora informacija.

Osim praktične implementacije, u radu su opisani i istraživački radovi na temu detekcije umora vozača. U tim radovima detekcija se bazira na EEG-u, analizi pritiska na volan i analizi karakteristika lica čemu je dana posebna pažnja. U tim radovima pokazano je da je EEG najtočnija metoda za detekciju umora. U radu su ukratko opisani i već postojeći stvarni sustavi za detekciju umora koji se koriste u modernim automobilima.

## Ključne riječi

Detekcija umora, obrada signala, obrada videa, aritmetika sa zasićenjem, digitalni filter, VHDL, Matlab, Modelsim, Xilinx, digitalni dizajn, Zync, Zedboard, FPGA, EEG, prepoznavanje karakterističnih točaka lica

# Title

Driver Drowsiness Detection Algorithms

## Abstract

In this paper, the system for drowsiness detection based on yawn analysis is described. The system takes upper lips and lower chin coordinates and calculates their distance. The calculated distance is then normalized and scaled in 0 to 1 interval. After that, the signal is brought to FIR filter where nonessential information is removed. From the filtered signal peak detector detects local maxima (signal peaks). The yawn detector takes these peaks, compares them to the threshold, and signals if yawn is detected. The timer takes signals of detected yawns and measures time between two yawn detection intervals. By using these intervals, yawn frequency is calculated. In the end, driver drowsiness is detected by setting yawn frequency threshold. The developed drowsiness detection is functional and can be used in real systems. However, this system is intended to be used as a part of a bigger one which uses more different information sources to make a valid decision.

Besides practical part, in this paper the various scientific research papers which are focused on drowsiness detection and monitoring are shown. In these papers, the detection is based on EEG analysis, steering grip force analysis and face analysis. The EEG method is considered most accurate and it is used as a reference for others. The systems for drowsiness detection which are used in modern cars today are also described.

## Keywords

Drowsiness detection, signal processing, video processing, saturation arithmetics, digital filter, VHDL, Matlab, ModelSim, Xilinx, digital design, Zync, ZedBoard, FPGA, EEG, face characteristics analysis