Аутоматика

Automation

Projektovanje klizne površi za klizne režime višeg reda u linearnim sistemima sa više ulaza

Boban Veselić, *Member, IEEE,* Čedomir Milosavljević, Branislava Draženović, *Senior Member, IEEE,* Senad Huseinbegović, *Member, IEEE*

***Apstrakt*—Rad ispituje mogućnost projektovanja klizne površi za realizaciju kliznih režima višeg reda (KRVR) u linearnim sistemima upravljanja sa više ulaza. U posmatranom slučaju klizna površ mora ispuniti dva zahteva: da obezbedi željenu dinamiku sistema u KRVR i da ostvari odgovarajući relativni red klizne promenjive u zavisnosti od željenog reda KR. Pokazano je da takva klizna površ postoji samo u sistemima sa specifičnom strukturom, i ne dozvoljava proizvoljni izbor dinamike sistema. Teorijsko dobijeni rezultati su verifikovani na numeričkom primeru i ilustrovani simulacionim rezultatima.**

***Ključne reči*—Klizni režimi višeg reda, projektovanje klizne površi, dinamika sistema u kliznom režimu, podešavanje polova.**

I. Uvod

Jedna od značajnih robusnih tehnika upravljanja su sistemi upravljanja promenljive strukture (SUPS) sa kliznim režimom (KR) [1], zbog teorijske invarijantnosti na poremećaje koji deluju u vektorskom prostoru upravljanja [2], tj. ispunjavaju uslove poklapanja. Ova osobina idealnih KR se u realnosti svodi na veliku robusnost sistema. Međutim, usled konačne prekidačke frekvencije i postojanja nemodelovane dinamike indukuju se visoko frekvencijske oscilacije u sistemu (chattering), koje predstavljaju glavnu prepreku široke primene ove tehnike upravljanja. KR višeg reda (KRVR) se javljaju u nastojanju da se redukuje pojava četeringa [3]. Najpre se primenjuju u sistemima sa jednim ulazom [4-6], a kasnije i u multivarijabilnim sistemima [7].

Određivanjem klizne površi u prostoru stanja po kojoj se odvija KR, definiše se dinamika sistema u KR. Postoje nekoliko metoda projektovanja klizne površi za konvencionalne KR (prvoga reda) u slučaju linearnih sistema. Najčešće se one baziraju na transformaciji modela sistema u prostoru stanja u tzv. regularnu formu [1], gde je redukovana dinamika sistema u KR jasno uočljiva. U sistemima sa jednim ulazom, primenom Akermanove formule [8] je moguće projektovati kliznu površ bez transformacije sistema. Takođe, predložen je i metod projektovanja klizne površi bez transformacije za sisteme sa jednim i više ulaza [9,10]. Projektovanje klizne površi se može vršiti i sa ciljem minimizacije uticaja poremećaja koji ne zadovoljavaju uslove poklapanja [2], što je predloženo u radovima [11,12].

Organizacija KRVR zahteva da relativni red vektora klizne promenljive, koji je vezan za kliznu površ, bude jednak željenom redu KR. Dakle, klizna površ u slučaju KRVR mora da ispuni dvojaki zahtev: (i) da obezbedi željenu redukovanu dinamiku sistema u KRVR i (ii) da ostvari potreban relativni red kliznih promenljivih u odnosu na upravljanje. Mali broj radova je tretirao problematiku projektovanja klizne površi u slučaju KRVR i to samo za sisteme sa jednim ulazom. Generalizacija Akermanove formule [8] za projektovanje klizne površi proizvoljnog relativnog reda, koja obezbeđuje željenu dinamiku u sistemu sa jednim ulazom je data u [13,14]. U [15] je takođe data procedura projektovanja klizne površi za KRVR u sistemima sa jednim ulazom, koja se bazira na analogiji sa projektivanjem konvencionalne povratne sprege po stanju.

U ovom radu se ispituje mogućnost projektovanja klizne površi kod linearnih sistema n-tog reda sa m ulaza sa ciljem organizovanja KR proizvoljnog reda r, uz ostvarenje željene dinamike sistema u KRVR. Takva klizna površ treba da bude relativnog reda r u odnosu na vektor upravljanja i da obezbedi željenu redukovanu dinamika $(n-rm)$-tog reda u KRVR. Pokazano je da takvu kliznu površ je moguće naći samo u sistemima koji ispunjavaju specifične strukturne zahteve, tj. u sistemima gde su indeksi kontrolabilnosti međusobno jednaki i jednaki redu KR r. Takođe, predložena je i jednostavna procedura za nalaženje te klizne površi. Validnost ove metode za projektovanje klizne površi je matematički dokazana i verifikovana simulacionim rezultatima numeričkog primera.

II. Opis problema

Posmatra se linearni vremenski invarijantni sistem upravljanja, opisan modelom u prostoru stanja

$$\dot{x} = Ax + B(u + d), \qquad (1)$$

gde $x \in \mathbb{R}^n$ je vektor stanja i $u, d \in \mathbb{R}^m$ su vektori upravljanja i nepoznatog ograničenog poremećaja, respektivno. Treba primetiti da su zadovoljeni uslovi poklapanja [2], te je sistem u idealnom KR invarijantan na poremećaj d. Matrice A i B su konstantne i imaju odgovarajuće dimenzije. Takođe, matrica B je punog ranga $(\text{rank}(B) = m)$ i sistem je potpuno kontrolabilan, pa je matrica kontrolabilnosti $Q_c = [B \quad AB \quad \cdots \quad A^{n-1}B]$ punog ranga $(\text{rank}Q_c = n)$. Sve

Boban Veselić – Elektronski fakultet, Univerzitet u Nišu, Niš, Srbija (e-mail: boban.veselic@elfak.ni.ac.rs).

Čedomir Milosavljević – Elektrotehnički fakultet, Univerzitet u Istočnom Sarajevu, Istočno Sarajevo, Bosna i Hercegovina (e-mail: cedomir.milosavljevic@elfak.ni.ac.rs).

Branislava Draženović i Senad Huseinbegović – Elektrotehnički fakultet, Univerzitet u Sarajevu, Sarajevo, Bosna i Hercegovina (e-mails: brana_p@hotmail.com, shuseinbegovic@etf.unsa.ba).

Ovaj rad je podržan od strane Ministarstva prosvete, nauke i tehnološkog razvoja Republike Srbije.

promenljive u (1) su funkcije vremena, ali zbog jednostavnije notacije argumenti promenljivih su izostavljeni u (1) i nadalje.

Zadatak upravljanja je organizovati KR r-tog reda u sistemu (1) duž klizne površi koja obezbeđuje željenu dinamiku sistema u KR. Neka je vektor klizne promenljive $g \in \mathbb{R}^m$ definisan sa

$$g = Cx, C \in \mathbb{R}^{m\times n}. \quad (2)$$

Kretanje sitema (1), (2) u potprostoru datog jednačinama

$$g = \dot{g} = \ddot{g} = \cdots = g^{(r-1)} = 0 \quad (3)$$

se naziva KR r-tog reda, [3]. Da bi upravljanje ostvarilo uslov (3) za konačno vreme u sistemu (1), (2), ono mora biti diskontinualno, makar na skupu (3), [5], uz preduslov da je relativni red klizne promenljive jednak r u odnosu na upravljanje. To znači da se upravljanje po prvi put eksplicitno pojavljuje u r-tom izvodu po vremenu klizne promenljive g, tj. u $g^{(r)}$. r-ti izvod klizne promenljive se dobija kao

$$g^{(r)} = CA^r x + \sum_{j=0}^{r-1} CA^{r-1-j}B(u^{(j)} + d^{(j)}). \quad (4)$$

Na osnovu (4), klizna promenljiva će imati traženi relativni red ukoliko važi sledeći uslov

$$C \cdot [B \quad AB \quad \cdots \quad A^{r-2}B] = 0_{m\times m(r-1)}, \quad (5)$$

$$CA^{r-1}B \neq 0_m. \quad (6)$$

Tada će r-ti izvod klizne promenljive (4) biti

$$g^{(r)} = CA^r x + CA^{r-1}B(u + d). \quad (7)$$

Dinamika sistema u KR r-tog reda je redukovanog reda, koji je jednak $(n - mr)$. Ekvivalentno upravljanje u_{eq} se određuje iz (7) na osnovu uslova

$$g^{(r)}\big|_{u=u_{eq}} = 0, \quad (8)$$

te se dobija

$$u_{eq} = -(CA^{r-1}B)^{-1}CA^r x - d. \quad (9)$$

Smenom ekvivalentnog upravljanja u (1) dobija se opis sistema u KR r-tog reda

$$\dot{x} = Ax + B(u + d)|_{u=u_{eq}} = [I - B(CA^{r-1}B)^{-1}CA^{r-1}]Ax = PAx = A_{eq}x. \quad (10)$$

Dinamika sistema u idealnom KR r-tog reda (10) pokazuje da je sistem neosetljiv na poremećaj d, koji zadovoljava uslove poklapanja. Kako je poremećaj nepoznat, ekvivalentno upravljanje (9) se u praksi ne može ostvariti jer zahteva poznavanje d.

Lako se pokazuje da za matricu $P = I - B(CA^{r-1}B)^{-1}CA^{r-1}$ važi $P^2 = P$, tj. P je idempotentna matrica. Tada je matrica P projektor. Na osnovu osobina projektorskih matrica [16], može se utvrditi da je $\mathrm{rank}(P) = n - m$. Tada je $\mathrm{rank}(A_{eq}) = \mathrm{rank}(PA) = n - m$, što daje $\det(A_{eq}) = 0$. Dakle, A_{eq} je singularna matrica čije nenulte sopstvene vrednosti definišu dinamiku sistema koja je $(n - mr)$-tog reda nakon nastanka KR r-tog reda. Matrica A_{eq} treba imati $(n - mr)$ stabilnih sopstvenih vrednosti, dok preostalih mr sopstvenih vrednosti treba biti jednake nuli. Iz (10) se vidi da se sopstvene vrednosti matrice A_{eq} mogu podešavati jedino izborom matrice C koja definiše kliznu površ (2). Dakle, matrica C ima dvojaku ulogu u sistemima upravljanja sa KRVR. Da ostvari željenu dinamiku u KR r-tog reda (10) i da obezbedi da relativni red klizne promenljive (2) bude r.

Kako je sistem (1) kontrolabilan, matrica kontrolabilnosti Q_c ima puni rang što znači da ova matrica ima n linearno nezavisnih kolona. Te kolone se mogu izvući iz Q_c da obrazuju kvadratnu matricu $H \in \mathbb{R}^{n\times n}$ koja je zapravo redukovana matrica kontrolabilnasti. Neka su b_i, $i = \overline{1,m}$ kolone matrice B. Tada se matrica H može predstaviti u obliku

$$H = [b_1 \cdots b_m \quad Ab_1 \cdots A^{r_1-1}b_1 \quad \cdots \quad Ab_m \cdots A^{r_m-1}b_m]. \quad (11)$$

Indeksi r_i, $i = \overline{1,m}$ iz matrice (11) su indeksi kontrolabilnosti [17], i važi $\sum_{i=1}^{m} r_i = n$.

Korišćenjem indeksa kontrolabilnosti moguće je dekomponovati sistem (1) u skup od m podsistema uvođenjem vektora koji se sastoji od m pomoćnih izlaza y_i $(i = \overline{1,m})$ i koji se definiše kao

$$y = Fx, F = \begin{bmatrix} f_1 \\ \vdots \\ f_m \end{bmatrix}, F \in \mathbb{R}^{m\times n}. \quad (12)$$

Da bi se sprovela dekompozicija, svaki pomoćni izlaz y_i treba imati relativni red koji je jednak indeksu kontrolabilnosti r_i, [18]. Zato treba naći odgovarajuću matricu F koja ostvaruje taj zahtev. Jedno rešenje matrice F je dato u [19] kao

$$f_i = [0_{1\times m\cdot(r_i-1)} \quad 0_{1\times(i-1)} \quad 1]H_i^{\dagger} \ (i = \overline{1,m}), \quad (13)$$

$$H_i = [B \quad AB \quad \cdots \quad A^{r_i-2}B \quad A^{r_i-1}B_i], \quad (14)$$

gde operator † označava pseudoinverziju i $B_i = [b_1 \quad b_2 \quad \cdots \quad b_i]$ je deo matrice B.

Izvodi po vremenu pomoćnih izlaza koji imaju tražene relativne redove su

$$y_i^{(j)} = f_i A^j x \ (j = \overline{0, r_i - 1}), \quad (15)$$

$$y_i^{(r_i)} = f_i A^{r_i} x + f_i A^{r_i-1}B(u + d). \quad (16)$$

Sistem (1) je dekomponovan u m podsistema, kao što je pokazano u [20], ako i samo ako kvadratna matrica $W \in \mathbb{R}^{m\times m}$

$$W = \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_m \end{bmatrix} = \begin{bmatrix} f_1 A^{r_1-1}B \\ f_2 A^{r_2-1}B \\ \vdots \\ f_m A^{r_m-1}B \end{bmatrix} \quad (17)$$

je punog ranga. Ovo je trougaono rasprezanje [21] pošto važi

$$w_i = f_i A^{r_i-1}B = [0_{1\times(i-1)} \quad 1 \quad \omega_{1\times(m-i)}], \quad (18)$$

gde je $\omega_{1\times(m-i)} = [\omega_{i,i+1} \quad \omega_{i,i+2} \quad \cdots \quad \omega_{i,m}]$ $(\forall \omega_{i,j} \in \mathbb{R};\ i = \overline{1,m};\ i < j \leq m)$. Matrica W je gornje-trougaona matrica, pa je $\det(W) = 1$.

Očigledno je da se ekvivalentno upravljanje (9) može sagledati kao tradicionalna povratna sprega po stanju, te se model (10), koji opisuje dinamiku sistema u KRVR, može napisati kao

$$\dot{x} = A_{eq}x = (A - B(CA^{r-1}B)^{-1}CA^r)x = (A - BK)x, \quad (19)$$

gde je $K \in \mathbb{R}^{m\times n}$ matrica pojačanja povratne sprege po stanju $u = -Kx$. Vidi se da postoji izvesna korelacija između matrice klizne površi C i matrice pojačanja K. Dakle, nalaženje matrice C, koja obezbeđuje željenu dinamiku u KRVR (iskazanu sopstvenim vrednostima matrice A_{eq}), je ekvivalentno određivanju matrice K povratne sprege po stanju koja obezbeđuje željenu lokaciju polova spregnutog sistema.

Mogućnost redukcije reda dinamike u KRVR je veća u odnosu na kovencionalni KR. Kod KR r-tog reda, matrica A_{eq} će imati mr nultih sopstvenih vrednosti. To znači da će se pojaviti slučaj da je red višestrukosti polova spregnutog sistema veći od broja ulaza u sistem, što predstavlja problem u korišćenju nekih standardnih metoda projektovanja povratne

sprege po stanju multivarijabilnih sistema.

Efikasna metoda projektovanja povratne sprege po stanju koja rešava nevedeni problem višestrukosti polova je predložena u [19], i zasniva se na dekompoziciju sistema (15), (16). Dinamika svakog od podsistema je reda r_i. Neka je željena dinamika sistema opisana sledećim skupom diferencijalnih jednačina

$$\delta^{r_i} y_i + \sum_{j=1}^{r_i} \alpha_{i,j}\, \delta^{r_i - j} y_i = P_i(\delta) y_i = 0, i = \overline{1,m}, \quad (20)$$

gde δ označava operator diferenciranja d/dt, a $\alpha_{i,j}$ su realni koeficijenti. U [19] je pokazano da matrica pojačanja K koja obezbeđuje željenu dinamiku spregnutog sistema se može izračunati kao

$$K = W^{-1} M, \quad (21)$$

$$M = \begin{bmatrix} m_1 \\ m_2 \\ \vdots \\ m_m \end{bmatrix}, m_i = f_i P_i(A), i = \overline{1,m}. \quad (22)$$

III. Projektovanje klizne površi za KRVR

Klizna površ je potprostor u prostoru stanja duž koje se organizuje KR pri kretanju stanja sistema do koordinatnog početka. Klizna površ se najčešće bira u linearnom obliku (2), koja definiše hiprravan

$$Cx = 0 \quad (23)$$

u prostoru stanja. Projektovanje klizne površi podrazumeva određivanje matrice C koja dozvoljava nastanak KRVR i ostvaruje željenu dinamiku sistema.

Uslov ostvarenja željene dinamike se može izvesti iz (19), gde se uočava jednakost $(CA^{r-1}B)^{-1} CA^r = K$, pri čemu se pretpostavlja da je lako naći matricu K koja obezbeđuje željenu dinamiku sistema sa povratnom spregom po stanju. Dobijeni uslov se može napisati u obliku

$$CA^{r-1}(A - BK) = 0_{m \times n}. \quad (24)$$

Takođe, matrica C treba ispuniti neophodne uslove relativnog reda, date sa (5) i (6). Uslov (6) se bez gubitka opštosti može usvojiti kao $CA^{r-1}B = I_m$. Sve ove jednačine koje mora da zadovolji matrica C se mogu združiti u matričnu formu

$$CL = D, \quad (25)$$

$$L = [A^{r-1}(A - BK) \quad B \quad AB \quad \cdots \quad A^{r-2}B \quad A^{r-1}B], \quad (26)$$

$$D = [0_{m \times n} \quad 0_{m \times m(r-1)} \quad I_m]. \quad (27)$$

Dakle, matricu C treba naći kao rešenje jednačine (25), koja se sastaji od m sistema jednačina, pri čemu svaki sistem čine $n + rm$ jednačina sa n nepoznatih, koje su elementi vrsta matrice C. Svaki sistem će imati rešenje ukoliko je broj linearno nezavisnih jednačina manji ili jednak broju nepoznatih, što je u u ovom slučaju n. Razmatrajući (25), bar jedno rešenje za C postoji ukoliko je rang proširene matrice jednak rangu matrice L, to jest

$$\operatorname{rank}\left(\begin{bmatrix} L \\ D \end{bmatrix}\right) = \operatorname{rank}(L), L \in \mathbb{R}^{n \times (n+rm)}, D \in \mathbb{R}^{m \times (n+rm)}. \quad (28)$$

Ukoliko je u (28) ovaj rang jednak n, rešenje je jedinstveno.

U cilju sagledavanja mogućnosti postojanja rešenja jednačina (25) potrebno je inicijalno proceniti broj linearno nezavisnih jednačina. Zato će se nezavisno analizirati jednačine koje matrica C mora da zadovolji. Kada je $\operatorname{rank}(A) = n$, onda je $\operatorname{rank}\left(A^{r-1}(A - BK)\right) = \operatorname{rank}\left(A^{r-1}A_{eq}\right) = n - m$, pošto je $\operatorname{rank}\left(A_{eq}\right) = n - m$. Ovo ukazuje da (24) ima $n - m$ linearno nezavisnih jednačina. Jednačina (6) doprinosi još dodatnih m linearno nezavisnih jednačina. Na osnovu ovoga je lako zaključiti da linearno nezavisne jednačine iz (5) u odnosu na celokupni sistem čine da broj ukupnih linearno nezavisnih jednačina prevazilazi n. To znači da u opštem slučaju treba očekivati da će pridodavanje matrice D matrici L u (28) povećati rang proširene matrice. Odatle sledi da

$$\operatorname{rank}\left(\begin{bmatrix} L \\ D \end{bmatrix}\right) > \operatorname{rank}(L) = n, \quad (29)$$

i sistem u opštem slučaju nema rešenje. Međutim, potrebno je ispitati da li postoji specifični slučaj kada sistem jednačina ima rešenje.

Posmatraće se slučaj kada su u sistemu (1) indeksi kontrolabilnosti međusobno jednaki i jednaki redu željenog KRVR, tj. kada važi

$$r = r_1 = r_2 = \cdots = r_m, \quad (30)$$

Kako je $\sum_{i=1}^{m} r_i = n$, iz (30) sledi $\sum_{i=1}^{m} r_i = mr = n$. Uslov (30) se onda može razložiti u dva zahteva

$$r_{\mathrm{i}} = n/m\,, i = \overline{1,m}, \quad mr = n. \quad (31)$$

Red sistema n treba biti deljiv brojem ulaza m. Takođe, pošto ima mr nultih sopstvenih vrednosti matrice A_{eq} za KR r-tog reda, u ovom slučaju svih n sopstvenih vrednosti matrice A_{eq} moraju biti jednake nuli. Redukovana matrica kontrolabilnosti H u slučaju (30) postaje

$$H = [B \quad AB \quad \cdots \quad A^{r-1}B]. \quad (32)$$

Polazeći od (24), prvi korak je odrediti matricu pojačanja K koja daje potrebne sopstvene vrednosti matrice A_{eq}, korišćenjem prethodno opisane procedure. U slučaju (30), trougaona matrica W (17) postaje

$$W = \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_m \end{bmatrix} = \begin{bmatrix} f_1 \\ f_2 \\ \vdots \\ f_m \end{bmatrix} A^{r-1}B = FA^{r-1}B. \quad (33)$$

Potrebna dinamika spregnutog sistema u ovom slučaju je da su sve sopstvene vrednosti jednake nuli, što na osnovu (20) daje

$$P_i(A) = A^r, i = \overline{1,m}. \quad (34)$$

Tada se matrica M iz (22) može napisati kao

$$M = \begin{bmatrix} m_1 \\ m_2 \\ \vdots \\ m_m \end{bmatrix} = \begin{bmatrix} f_1 \\ f_2 \\ \vdots \\ f_m \end{bmatrix} A^r = FA^r. \quad (35)$$

Vrste f_i, $i = \overline{1,m}$ matrice F se mogu naći na osnovu (13), (14). Svaki sistem jednačina vezan za (13), (14)

$$f_i[B \quad AB \quad \cdots \quad A^{r_i-2}B \quad A^{r_i-1}B_i] = [0_{1 \times m \cdot (r_i - 1)} \quad 0_{1 \times (i-1)} \quad 1], (i = \overline{1,m}) \quad (36)$$

se sastoji od $n - m + i$ linearno nezavisnih jednačina koji je manji od broja nezavisnih n za $i = \overline{1,m-1}$. To znači da postoji beskonačno mnogo rešenja za F, a samim tim i za K. Bilo koje od dobijenih rešenja za K će obezbediti željenu dinamiku. Zato je moguće sistem jednaćina (36) dopuniti sa $m - i$ jednačina koje neće uticati na korektnost rešenja. Adekvatana dopuna jednačina (36) je

$$f_i[A^{r-1}b_{i+1} \quad \cdots \quad A^{r-1}b_m] = 0_{1 \times (\mathrm{m-i})}, (i = \overline{1,m}). \quad (37)$$

Tada se sistem (36) proširuje u oblik

$$f_i[B \quad AB \quad \cdots \quad A^{r-1}B] =$$
$$[0_{1\times(n-m)} \quad 0_{1\times(i-1)} \quad 1 \quad 0_{1\times(m-i)}], (i = \overline{1,m}). \quad (38)$$

U proširenom sistemu (38) broj jednačina je jednak broju nepoznatih.

Objedinjavajući svih m sistema jednačina, dobija se sledeći matrični zapis

$$F[B \quad AB \quad \cdots \quad A^{r-1}B] = FH = [0_{m\times(n-m)} \quad I_m]. \quad (39)$$

Posmatrajući jednačinu $FA^{r-1}B = I_m$ iz zapisa (39), zaključuje se da će (33) biti jedinična matrica, tj. $W = I_m$. Iz (21) sledi $K = M$ što na osnovu (35) daje

$$K = FA^r. \quad (40)$$

Poređenjem (39) sa sistemom jednačina (5) i $CA^{r-1}B = I_m$, može se primetiti da su ova dva sistema identična. Odatle se može zaključiti da je $F = C$ ukoliko nađeno K iz (40) zadovoljava (24). Zaista, ukoliko se $K = CA^r$ zameni u (24) dobija se $CA^{r-1}(A - BCA^r)$, što je nula matrica pod uslovom $CA^{r-1}B = I_m$. To znači da je C identično sa F i da je jednačina (24) sadržana u preostalim jednačinama sistema (25)-(27). Onda se (24) može zanemariti u sistemu (25)-(27), što daje

$$CH = [0_{m\times(n-m)} \quad I_m]. \quad (41)$$

Kako je redukovana matrica kontrolabilnosti H punog ranga n, njena inverzna matrica postoji i C se lako može naći kao

$$C = [0_{m\times(n-m)} \quad I_m]H^{-1}. \quad (42)$$

Ovaj rezultat pokazuje da matrica klizne površi C, koja istovremeno zadovoljava željenu dinamiku sistema u KRVR i neophodan relativni red, se može naći samo u veoma specifičnom slučaju. Ograničenja (30) ili (31) uslovljavaju da je matrica spregnutog sistema A_{eq} nilpotentna, te KRVR obezbeđuje dinamiku nultog reda [19], što rezultuje konačnim vremenom dolaska u ravnotežno stanje. U svim ostalim slučajevima linearnih sistema, matrica C koja obezbeđuje proizvoljen red KRVR i željenu dinamiku se ne može naći.

IV. IZBOR ALGORITMA UPRAVLJANJA

Nakon konstrukcije klizne površi potrebno je naći upravljanje koje organizuje KR r-tog reda duž te površi. Primenjeni upravljački algoritam treba da obezbedi uslov (3) klizne promenljive g. Pošto nađeno C zadovoljava uslove relativnog reda (5), (6) i $CA^{r-1}B = I_m$, izvodi po vremenu klizne promenljive su dati sa

$$g^{(j)} = CA^j x, j = \overline{0, r-1}. \quad (43)$$

$$g^{(r)} = CA^r x + u + d = Kx + u + d. \quad (44)$$

Iz (44) je očigledno da svaka komponenta vektora $g^{(r)}$ zavisi isključivo od odgovarajuće komponente vektora upravljanja u. U ovako raspregnutom sistemu svaka komponenta klizne promenljive se može posmatrati odvojeno, to jest

$$g_i^{(r)} = c_i A^r x + u_i + d_i = k_i x + u_i + d_i, i = \overline{1,m}, \quad (45)$$

gde su c_i i k_i vrsta vektori matrica C i K, respektivno. Dakle, sa stanovišta projektovanja regulatora, sistem (1) se može tretirati kao m sistema sa jednom izlazom. Tkođe, na osnovu jednakosti $F = C$ u slučaju (30) može se zaključiti da je pomoćni izlaz y ekvivalentan kliznoj promenljivi g.

Ostvareno rasprezanje (45) omogućava da se u realizaciji regulatora primene algoritmi upravljanja KRVR razvijeni za sisteme sa jednim ulazom. Na primer, u [5] je predloženo diskontinualno upravljanje koje u sistemu sa jednim ulazom ostvaruje uslov (3) za konačno vreme. Za isprojektovanu stabilnu dinamiku u KRVR, trajektorija sistema će asimptotski konvergirati duž (3) u koordinatni početak ($x \to 0$ for $t \to \infty$) uprkos dejstvu poremećaja koji zadovoljava uslove poklapanja.

Jedna mogućnost je primeniti upravljanje [6] u obliku

$$u_i = -c_i A^r x - \gamma_i(\xi_i),\ \xi_i = \left(g_i, \dot{g}_i, \dots, g_i^{(r-1)}\right). \quad (46)$$

Tada (45) postaje

$$g_i^{(r)} = -\gamma_i(\xi_i) + d_i. \quad (47)$$

Levant je u radu [5] dao skup kvazi-kontinualnih funkcija $\gamma_i(\xi_i)$ koje uspostavljaju KR r-tog reda u nelinearnim sistemima sa skalarnim upravljanjem. Kompleksnost ovih funkcija raste sa porastom reda KR, dok se četering smanjuje. Ove funkcije se lako mogu primeniti i na linearni slučaj (47).

Funkcije $\gamma_i(\xi_i)$ koje garantuju nastanak KRVR za konačno vreme se mogu izabrati kao nelinearne funkcije date u [4,5]. Na primer, $\gamma_i(\xi_i)$ za $r = 1,2,3$ su date respektivno kao

$$\gamma_i(\xi_i) = \alpha_i \frac{g_i}{|g_i|}, \quad (48)$$

$$\gamma_i(\xi_i) = \alpha_i \frac{\dot{g}_i + \beta_{1,i}|g_i|^{\frac{1}{2}}\mathrm{sign}(g_i)}{|\dot{g}_i| + \beta_{1,i}|g_i|^{\frac{1}{2}}}, \quad (49)$$

$$\gamma_i(\xi_i) = \alpha_i \frac{\ddot{g}_i + \beta_{2,i}\left(|\dot{g}_i| + \beta_{1,i}|g_i|^{\frac{2}{3}}\right)^{-\frac{1}{2}}\left(\dot{g}_i + \beta_{1,i}|g_i|^{\frac{2}{3}}\mathrm{sign}(g_i)\right)}{|\ddot{g}_i| + \beta_{2,i}\left(|\dot{g}_i| + \beta_{1,i}|g_i|^{\frac{2}{3}}\right)^{\frac{1}{2}}}, \quad (50)$$

gde su parametri $\alpha_i, \beta_{1,i}, \beta_{2,i} > 0$ i izabrani dovoljno veliki.

Treba uočiti da izračunavanje upravljanja zahteva poznavanje sukcesivnih izvoda po vremenu klizne promenljive $g_i^{(j)}$, $j = \overline{0, r-1}$. Međutim, u sistemima na koje deluju poremećaji koji zadovoljavaju uslove poklapanja, ovi izvodi se mogu naći iz (43) kao

$$g_i^{(j)} = c_i A^j x, j = \overline{0, r-1}. \quad (51)$$

Uključujući sve komponente upravljanja (46), konačni vektor upravljanja se može formirati kao

$$u = -CA^r x - \gamma(\xi), \gamma(\xi) = \begin{bmatrix} \gamma_1(\xi_1) \\ \vdots \\ \gamma_m(\xi_m) \end{bmatrix}, \quad (52)$$

gde su $\xi_i = \left(g_i, \dot{g}_i, \dots, g_i^{(r-1)}\right), i = \overline{1,m}$.

V. NUMERIČKI PRIMER I SIMULACIONI REZULTATI

Ispravnost predložene procedure konstrukcije klizne površi za KRVR u linearnim sistemima sa više ulaza ispitana je na numeričkom primeru i ilustrovana simulacionim rezultatima.

Kao što je pokazano prethodnom analizom, da bi se u linearnom sistemu sa više ulaza organizovao KRVR duž odgovarajuće klizne površi potrebno je da objekat upravljanja zadovoljava strukturalne zahteve (31). S toga, neka je proizvoljni dinamički sistem šestog reda (1) opisan matricama

$$A = \begin{bmatrix} 1 & -1 & 0 & -0.5 & 4 & 7 \\ 2 & -2 & 0 & -1 & 8 & 14 \\ 1 & 3 & 5 & 0.5 & 1 & -3 \\ -2 & -3 & -4 & 0.5 & 3 & -2 \\ 4 & 3 & -1 & 2 & -3 & 5 \\ 0.5 & 1 & 3 & 5 & -6 & 7 \end{bmatrix}, B = \begin{bmatrix} 1 & 0 \\ 2 & -2 \\ 3 & 1 \\ 4 & 4 \\ -1 & 1 \\ 3 & 5 \end{bmatrix}.$$

Sistem je potpuno kontrolabilan, matrica A je nestabilna i

singularna i $\text{rank}(B) = m = 2$. Iz redukovane matrice kontrolabilnosti (11) se mogu odrediti indeksi kontrolabilnosti kao $r_1 = r_2 = 3$, te u slučaju KR trećeg reda ($r = 3$) uslov (30) je zadovoljen. To znači da se može naći klizna površ koja obezbeđuje traženi relativni red $r = 3$ i odgovarajuću dinamiku sistema. Kako je $n - rm = 0$, sve sopstvene vrednosti spregnutog sistema treba biti jednake nuli. Matrica klizne površi C se može naći korišćenjem (42) kao

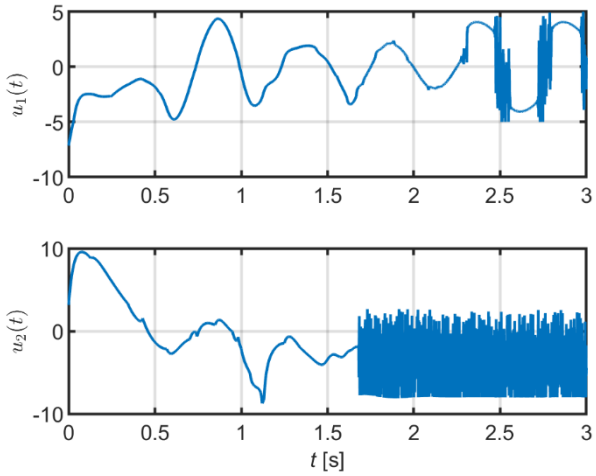
$$C = \begin{bmatrix} 0.0148 & -0.0061 & 0.0015 & -0.0023 & -0.003 & -0.0003 \\ -0.0074 & 0.0033 & -0.0001 & 0.0020 & 0.0038 & -0.001 \end{bmatrix}.$$

Lako se može verifikovati da ovako dobijeno C zadovoljava jednačine (25)-(27).

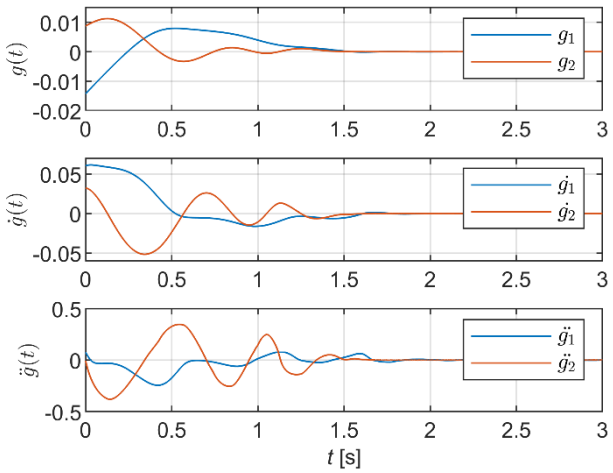
Upravljanje koje uspostavlja KR trećeg reda u posmatranom sistemu je realizovano kao (52), gde su funkcije $\gamma_i(g_i, \dot{g}_i, \ddot{g}_i)$, $i = 1{,}2$ date sa (50). Parametri regulatora su izabrani kao $\alpha_{1,1} = 5$, $\beta_{1,1} = 0.35$, $\beta_{2,1} = 0.65$, $\alpha_{1,2} = 8$, $\beta_{1,2} = 1$ i $\beta_{2,2} = 2$.

U ispitivanju performansi projektovanog regulatora, posmatraće se kretanje sistema iz početnog stanja $x(0) = [1 \quad 5 \quad -2 \quad -6 \quad 3 \quad 0]^{\mathrm{T}}$ pri dejstvu vektora spoljnog poremećaja

$$d(t) = \begin{bmatrix} 2\sin 4\pi t \\ 2h(t-1) \end{bmatrix}.$$

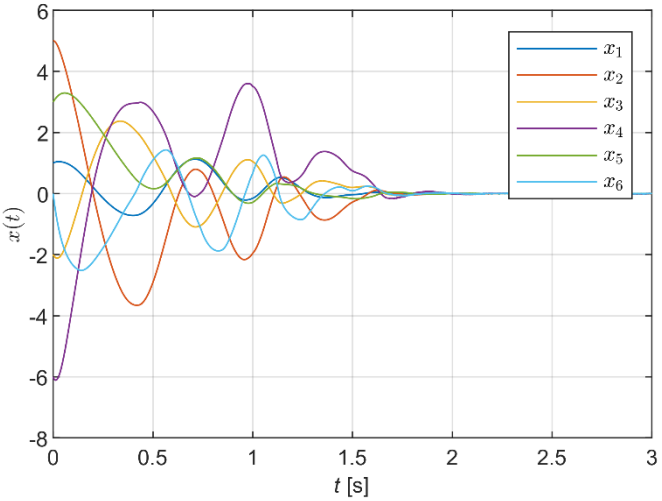


Sl. 1. Vektor upravljanja.



Sl. 2. Vektor klizne promenljive i njegovi izvodi.

Komponente dobijenog vektora upravljanja su prikazane na Sl. 1. Na Sl. 2 su dati vektor klizne promenljive i njegovi prvi i drugi izvodi po vremenu. Očigledno je da svi ovi signali postaju jednaki nuli za konačno vreme, što ukazuje da nastaje KR trećeg reda za konačno vreme duž površi $g = \dot{g} = \ddot{g} = 0$. Kako primenjeno upravljanje obezbeđuje sve nulte sopstvene vrednosti sistema u KRVR, ostvarena je konvergencija promenljivih stanja u koordinatni početak za konačno vreme takođe, što se može videti na Sl. 3. Ovakvo upravljanje redukuje red dinamike u KR na nulu, zbog čega se naziva i upravljanje u konačnom vremenu.



Sl. 3 Promenljive stanja

VI. Zaključak

U radu se ispituje mogućnost konstrukcije klizne površi za realizaciju KRVR kod linearnih sistema sa više ulaza. Projektovana klizna površ treba da obezbedi traženi relativni red, koji je uslovljen redom KRVR, kao i željenu dinamiku sistema. Pokazano je da kod sistema sa više ulaza tražena klizna površ postoji samo u slučajevima kada su indeksi kontrolabilnosti međusobno jednaki i jednaki redu KRVR. Ostvarena dinamika sistema na toj kliznoj površi je nultog reda, što obezbeđuje dolazak u ravnotežno stanje u konačnom vremenu. Dakle, da bi se ostvario KRVR kod linearnih sistema sa više ulaza potrebno je da sistem zadovolji tražene strukturne preduslove.

Za ovaj slučaj je predložena jednostavna procedura za izračunavanje matrice klizne površi C, i sugerisan je algoritam upravljanja koji ostvaruje KRVR. Razvijena metoda je ispitana na numeričkom primeru kroz simulacije, čiji su rezultati potvrdili analitički predviđeno ponašanje sistema.

Literatura

[1] V. I. Utkin, Sliding modes in control and optimization, Berlin, Germany: Springer-Verlag, 1992.

[2] B. Draženović, "The invariance conditions in variable structure systems," *Automatica*, vol. 5, no. 3, pp. 287-295, 1969.

[3] A. Levant, "Sliding order and sliding accuracy in sliding mode control," *Int. J. Contr.*, vol. 58, no. 6, pp. 1247-1163, 1993.

[4] A. Levant, "Homogeneity approach to higher-order sliding mode design" *Automatica*, vol. 41, pp. 823-830, 2005.

[5] A. Levant, “Quasi-continuous high-order sliding-mode controllers,” *IEEE Trans. on Automatic Control*, vol. 50, no. 11, pp. 1812-1816. 2005.
[6] V.I. Utkin, “Discussion aspects of higher-order sliding modes,” *IEEE Trans. on Automatic Control*, vol. 61, no. 3, pp. 829-833, 2016.
[7] A. Levant, M. Livne, “Uncertain disturbances’ attenuation by homogeneous MIMO sliding mode control and its discretization,” *IET Control Theory & Applications*, vol. 9, no. 4, pp. 515-525, 2015.
[8] J. Ackermann, V. Utkin, “Sliding mode control design based on Ackermann’s formula,” *IEEE Trans. on Automatic Control*, vol. 43, no. 2, pp. 234-237, 1998.
[9] B. Peruničić, Č. Milosavljević, B. Veselić, V. Gligić, “Comprehensive approach to sliding subspace design in linear time invariant systems,” Proc. of IEEE 12th Int. Workshop on Variable Structure Systems (VSS 2012), pp. 473-478, Mumbai, India, 2012.
[10] B. Draženović, Č. Milosavljević, B. Veselić, “Comprehensive Approach to Sliding Mode Design and Analysis in Linear Systems”, in B. Bandyopadhyay, S. Janardhanan and S.K. Spurgeon (Eds.), *Advances in Sliding Mode Control: Concept, Theory and Implementation*, Lecture Notes in Control and Information Sciences, Vol. 440, Ch. 1, pp 1-19, Springer Berlin Heidelberg, 2013.
[11] B. Veselić, B. Draženović, Č. Milosavljević, “Sliding manifold design for linear systems with unmatched disturbances,” *Journal of the Franklin Institute*, Vol. 351, No. 4, pp. 1920-1938, 2014.
[12] B. Veselić, B. Draženović, Č. Milosavljević, “Integral sliding manifold design for linear systems with additive unmatched disturbances,” *IEEE Transactions on Automatic Control*, Vol. 68, No. 9, pp. 2544-2549, 2016.
[13] D. Hernandez, F. Castanos, L. Fridman, “Pole-placement in higher-order sliding-mode control,” Proc. 19th IFAC Word Congres, pp. 1386-1391, 2014.
[14] I. Castillo, F. Castaños, L Fridman, “Sliding Surface Design for Higher-Order Sliding Modes,” in L. Fridman, J.P. Barbot, F. Plestan (eds.), *Recent Trends in Sliding Mode Control*, IET, 2016
[15] B. Veselić, Č. Milosavljević, B. Draženović, S. Huseinbegović, "Podešavanje dinamike kliznih režima višeg reda kod linearnih sistema sa jednim ulazom", Zbornik radova 63. Konferencije za ETRAN, str. 219-223, 2019.
[16] O.M.E. El-Ghezawi, A.S.I. Zinober, S.A. Billings, “Analysis and design of variable structure systems using a geometric approach,” *Int. J. Control*, Vol. 38, No. 3, pp. 657–671, 1983.
[17] R.E. Kalman, Kronecker invariants and feedback, Stanford University California, Department of Operations Research, 1971.
[18] M. Mueller, “Normal form for linear systems with respect to its vector relative degree,” *Linear Algebra and Applications*, Vol. 430, No. 4, pp. 1292-1312, 2009.
[19] B. Peruničić-Draženović, B. Veselić, S. Huseinbegović, Č. Milosavljević, “Higher order sliding mode control design with desired dynamics for multi-input LTI systems,” Proc. of 18th Europian Control Conference (ECC 2019), pp. 3589-3594, Napoli, Italy, 2019.
[20] P.L. Falb, W.A. Wolovich, “Decoupling in the design and synthesis of multivariable control systems,” *IEEE Transactions on Automatic Control*, Vol. 12, pp.651-659, 1967.
[21] A. Morse, W. Wonham, “Triangular decoupling of linear multivariable systems,” *IEEE Transactions on Automatic Control*, Vol. 15, No. 4, pp. 447-449, 1970.

ABSTRACT

The paper investigates possibilities of sliding manifold design for realization of high-order sliding mode (HOSM) in linear multi-input control systems. For this case, the sliding manifold must meet two requirements: to achieve the desired dynamics in HOSM and to provide the appropriate relative degree of the sliding variable depending on the SM order. It is shown that such sliding manifold exists only in systems with specific structural constraints and does not allow arbitrary SM dynamics selection. Theoretically obtained results are validated through a numerical example and illustrated by digital simulations.

Sliding Manifold Design for Higher-Order Sliding Mode in Multi-Input Linear Systems

B. Veselić, Č. Milosavljević, B. Draženović, S. Huseinbegović

Concept of System for Surveillance and Monitoring of IoT HFSWR Network

Nikola Stojkovic, *Member, IEEE*, Vladimir Orlic, *Member, IEEE,* Miroslav Peric, *Member, IEEE,* Dejan Drajic, *Senior Member, IEEE,* Aleksandar Rakic, *Member, IEEE*

***Abstract*—The IoT (Internet of Things) concepts for maritime surveillance systems represent an interesting, but rather unexplored area. This paper presents the IoT architecture for the High Frequency Surface Wave Radar (HFSWR) network within the well-known Integrated Maritime Surveillance (IMS) concept. An overview of the topology of a typical HFSWR network is given, and IoT architecture layers and distributed middleware functionality are defined. The architecture is implemented and tested in the Gulf of Guinea, Africa, where an aggregated surveillance and monitoring Web application operates in the private cloud, supported by the Web REST services and SNMP. Effectiveness of the solution is demonstrated in both network monitoring and surveillance aspects by giving details of a SNMP agent testing and the system-level insight to the network operation from the application layer.**

***Index Terms*—HFSW Radar, OTH Radar, IoT concept, HFSWR network, Integrated Maritime Surveillance.**

I. Introduction

Monitoring of remote sea areas inside EEZ (Exclusive Economic Zone) of maritime nations could be performed via satellite and aviation surveillance or by the deployment of HF-OTHR (High Frequency Over-the-Horizon Radar). Application of HF-OTHR network, without doubts, provides significant advantages in terms of deployment price and availability of sensor data over the aforementioned solutions. There are many possible technological implementations of HF-OTHR and one the most common types is HFSWR (High Frequency Surface Wave Radar).

HFSWR network is a Integrated Maritime Surveillance (IMS) subsystem, therefore conformed with HFSWR based IMS concept, defined in [1]-[2]. From this conformity certain assumptions could be made about HFSWR network's topology and disposition of its nodes. First of all, IMS concept assumes aggregate data processing node and arbitrary number of remote nodes, from where surveillance data originates. This certainly indicates a star-shaped topology of the network. Yet, some specifics of the topology need to be properly defined. Every remote node of this subsystem has potential problem with its communication channel, since remote nodes are installed on locations where is a lack of desirable communication infrastructure to support the work of HFSWR network. Besides measurement data, there are other secondary sensor data related to infrastructure state, like device calibration results or diagnostic information, that need to be transferred to processing nodes. Besides main sensors, there are other sensors and controllable devices on site nodes, e.g. ambient, power measurement sensors, routers, power distribution units (PDU), power amplifiers etc. All these facts define HFSWR network as a complex system for distributed measurement and control, whose elements are often resource-limited either by communication channel or by the construction of sensors itself, or even by both of these factors. This is the reason for creating a detailed conception of mechanism for control and monitoring, which should provide functional and uninterrupted flow of data and monitoring reports. For this purpose, the Internet of Things (IoT) conceptual scheme will be used, mainly based on SOA (Service Oriented Architecture) paradigm. It will, at some extent, follow principles of interaction with resource-limited sensors, given in [3], good practices of infrastructure monitoring, e.g. [4] and [5]. Note that HFSWR is a sensor, which resource limitation is predominantly regarding its communication ability, rather then computational capability. This fact is taken into account when building IoT infrastructure for HFSWR based naval surveillance systems.

Sensors and devices in the system have various built-in interfaces and an unified external access should be provided via Web service implementations. Additionally, HFSWR network can be controlled via external NMS (Network Management System) applications. These monitoring systems usually deploy SNMP protocol, and, in order to comply, middleware layer should contain one or more SNMP agents [6], for those components which do not own such interface.

To formulate the concept, topology of HFSWR network will be explained in Section II, together with network requirements in terms of security and planning and IoT architecture layers and its elements. Then, distributed middleware details will be presented. In Section III details of implementation are described, where details about Web

Nikola Stojkovic is with the School of Electrical Engineering, University of Belgrade, 73 Bulevar kralja Aleksandra, 11020 Belgrade, Serbia and the Vlatacom Institute, Belgrade, Bulevar Milutina Milankovića 5, 11070 Novi Beograd, Serbia, (e-mail: nikola.stojkovic@vlatacom.com)

Vladimir Orlic is with the Vlatacom Institute, Belgrade, Bulevar Milutina Milankovića 5, 11070 Novi Beograd, Serbia, (e-mail: vladimir.orlic@vlatacom.com)

Miroslav Peric is with the Vlatacom Institute, Belgrade, Bulevar Milutina Milankovića 5, 11070 Novi Beograd, Serbia, (e-mail: miroslav.peric@vlatacom.com)

Dejan Drajic is with the School of Electrical Engineering, University of Belgrade, Bul. Kralja Aleksandara 73, 11120 Belgrade, Serbia, (e-mail: ddrajic@etf.rs)

Aleksandar Rakic is with the School of Electrical Engineering, University of Belgrade, Bul. Kralja Aleksandara 73, 11120 Belgrade, Serbia, (e-mail: rakic@etf.rs)

service specification, SNMP agent solution and CC (Cloud Computing) platform utilization are given. Finally, fully functional IoT architecture, based on deployed HFSWR network in Gulf of Guinea, will be evaluated in Section IV via demonstration of applications from IoT application layer. Section V concludes the paper and provides details about future work.

II. HFSWR NETWORK IOT ARCHITECTURE

A. *HFSWR Network Topology*

HFSWR network, typically, has a star-shaped topology, in which external nodes represent individual remote sensor installations or operator nodes, with central node collection, representing the location of a Command and Control (C2) center, as presented in Fig.1. Remote sensor nodes (nodes of remote sites) represent installations, in general, of one or more homogeneous or heterogeneous sensors. Hence, by its nature, nodes of remote sensor sites can be elemental or mixed. Mixed remote sites usually contain combination of two, or even all, elemental sites. Types of elemental sites are:

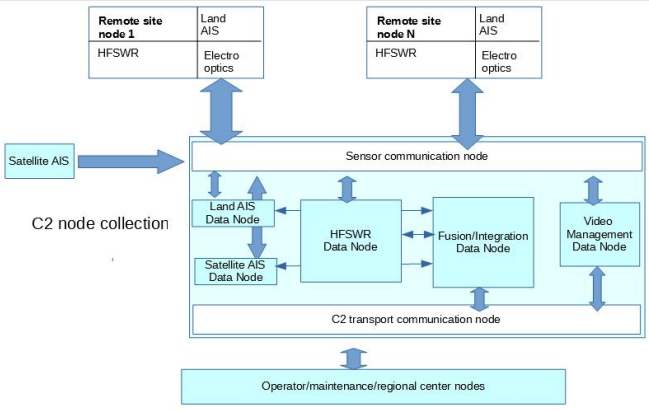
Satellite AIS reception site. This site contains satellite receiver equipment, necessary to connect to the satellite AIS (Automated Identification System) provider services. Note that satellite AIS node could be installed near C2 center facilities.

HFSWR site. This node contains HFSWR sensor, namely, the equipment that enables its function, other sensors for ambient and power measurements and server equipment for in-node primary processing of sensor data.

Land AIS base station site. Node contains AIS transceiver base station and network equipment for further data exchange.

EO surveillance site. On this node video surveillance cameras of different type (thermal, low-light etc.) are installed, which allow detection and identification of sea vessels on relatively large distances.

Another group of external nodes is dedicated for organized user installations, in forms of central or regional operation centers, which can be divided in three groups, maritime surveillance operator centers, supervision and maintenance operator centers and regional center node collections for surveillance and supervision.



.Figure 1. Typical star-shaped HFSWR network topology.

All nodes of the HFSWR network, located at C2 center installation, are represented in Fig. 1 as a collection of C2 center nodes, which form central point of topology. . Described nodes can be classified in following categories:

Sensor communication nodes. These nodes contain communication link and part of the C2 center infrastructure, with the task of reception of all sensor data in the network.

C2 transport nodes. Final results of integral data processing, including the system state related data is transmitted to this node, to which external user node communication link is joined, usually in the form of ethernet network. Final conditioning of data and its transport to end-users is performed in this node.

Aggregation AIS nodes. All LAIS (Land AIS) and SAIS (Satellite AIS) links are connected over sensor communication nodes to this node, where all data is processed in concentrated manner.

HFSWR data reception node. Data from all HFSWR sensors is concentrated in these nodes, filtered, processed and delivered to later stages, usually integration and monitoring nodes.

Sensor fusion / integration nodes. These process nodes perform fusion processes of different sensor data, HFSWR and AIS at the first place.

Video management nodes. Their main role is reception and resource management of video images from remote sensors for maritime and security surveillance.

Note that there are other possible topologies, e.g. a regionally grouped variant, with multiple regional processing nodes, but this topology is the most common and with significant advantages in terms of availability and implementability. HFSWR network, installed in Gulf of Guinea, also follows the star-shaped topology, presented in Fig. 1.

B. *HFSWR network IoT architecture details*

When establishing IoT architecture, one can start with the best IoT framework examples from [7] and radar IoT applications [8]. Block scheme of IoT architecture is consisted of sensor, network and application layer as shown in Fig. 2.

Sensor layer consists of already mentioned main measurement devices and services: HFSWR, LAIS, SAIS, video, GPS, ambient and power measurement sensors.

Network layer represents a bridge between sensor and application layer [4]. It is consisted of following blocks:

VPN (Virtual Private Network). End-user of the system is, most likely, military navy or a specialized state organization, which implies the need for security of exchanged data in HFSWR network.

Satellite network. Quite often, satellite link is the only way of communication between remote sites and the C2 center. Details about the role of satellite network in IoT concept of HFSWR network, deployed in Bay of Guinea can be found in [9].

GSM/3G/4G network. A convenient way to alternatively address the issue of communication link is the available cellular network, and, at the same time, the economic cost is much lower than the satellite network. The problem is that the number of cellular network access points on remote sites is usually one or none at all, and the quality of the connection could be quite low.

Integration of heterogeneous networks. In general, there

are many different networking solutions present in the HFSWR network. From the use of VPNs in some domains of, or in the whole network, through satellite or cellular network data link. These are all factors of a heterogeneous network and the integration represents a complex task.

Remote access. Remote access is the principle of monitoring the HFSWR network, in which sensors and various devices are controlled and monitored over the Internet by client applications or processes, managed by people or expert systems.

M2M (Machine to Machine) wireless access. Certain parts of the HFSWR network on the remote sites may have a wireless interface or have provided access to the HFSWR network through fixed or even mobile access points.

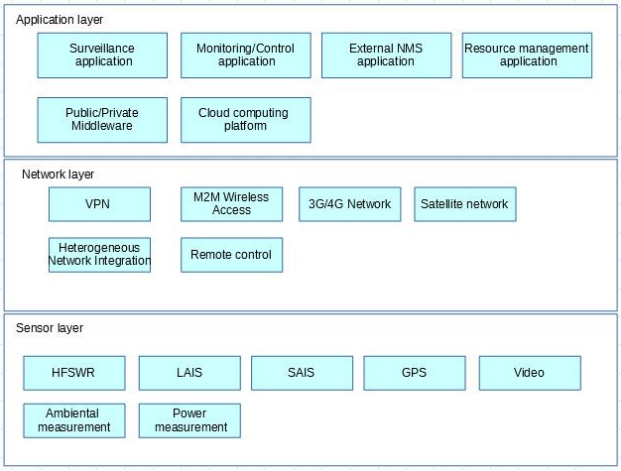


Figure 2. Basic IoT architecture concept of HFSWR based IMS system

The application layer consists of the following blocks:

A surveillance application. This is an application for observation of sensor coverage surfaces, which will conveniently display the maritime situation.

HFSWR network monitoring / control application. This application includes an overview of all monitor points, which should be hierarchically classified.

External NMS application. Due to the fact that the HFSWR network can be a subsystem of a more complex network and because of the popularity of SNMP-based NMS, there is a need to provide an additional SNMP-based interface that would use external NMS applications, such as Centerity Monitor [10], to represent a functional "mirror" of the underlying Web service interface.

Resource management application. This application should be able to configure the software part related to the postprocessing of sensor data, configure the computer hardware and take care of correct device configurations and access parameters.

Cloud Computing Platform. The Cloud Computing (CC) platform is tasked with enabling the implementation of systems in a private cloud. Its physical base consists of several physical servers, data storage, routers and switches.

Private and public middleware. Distribution middleware is software, installed on remote sites and in the C2 center, which interacts with another part installed on the CC platform. This segment represents the private middleware. The second part of the distribution middleware interacts with active, external users of the system. This middleware segment is named public. The more details are provided in the next subsection.

C. Distributed middleware

A generalized distributed middleware scheme is given in Fig. 3, on the example of HFSWR site. Distributed private middleware on remote sites and C2 center nodes is a set of proxy gateway components that contain mappers and data handlers and allow access to controllable parameters, monitoring points (probes) and alarm definitions. These components have a role of translating, mapping and packing inputs and outputs according to the communication channel needs The CC platform has the task of implementing the main SOA-based IoT infrastructure, composed of a stack of Web services. It implements a private and public middleware, whose functionality can be divided into 4 groups, and all of these components have their share in both the private and public part of the distributed middleware:

Agentware. All software components that transform the SNMP polling or control requests of external NMS applications, or that customize internal information by transforming the interface of the controlled components into information that is customized to the SNMP interface, are collectively referred to as the agentware.

Device managers. Device managers perform configuration, polling, startup, shutdown, and direct control of individual devices in the HFSWR network.

Notification managers. In the case of one-way sensors that have a limited interface and send measurements and eventual status messages, notification managers are responsible for receiving such messages, processing and responding, in coordination with the agent middleware.

Data managers. They are primarily responsible for implementing interfaces to various data storage. This specifically refers to maritime surveillance databases, system user databases and system monitoring databases, which store information related to the HFSWR network diagnostics.

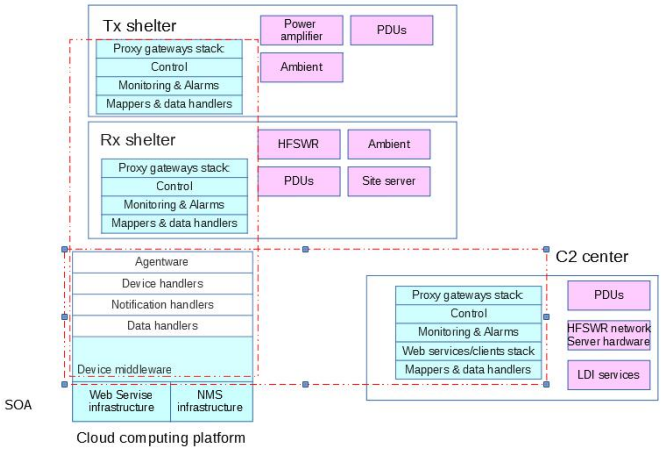


Figure 3. Distributed middleware structure of HFSWR network's IoT architecture

III. IMPLEMENTATION OF IOT ARCHITECTURE SOFTWARE INFRASTRUCTURE

The implementation of the IoT architecture software on the CC platform begins with a description of the hardware and software platform configuration. In the logical

presentation of the CC platform, it is clear that an application server is required, which will be the carrier of Web service implementations and much of the private and public middleware. Due to database needs, one of the logical units has to be a database server. These two logical servers (without counting their redundancies) form the basis of the CC platform's logical architecture. There are 3 databases in the system:

System user database. The database stores user information, roles, allocated resources and specific user tasks.

Naval observation database. This database stores common operational picture (COP) data in each iteration. In addition to the COP, tables for entering AIS data as well as individual HFSWR outputs were implemented.

Network system database. This database records system configuration and error logs in the middleware, analogous to the role of the middleware historian model in [11].

The application server implements a Web service stack, a major SNMP agent for communication with external NMS infrastructure and part of a private and public middleware, which refers to device managers and data controllers. The software is organized as one multi-component Web application, hosted on Microsoft IIS (Internet Information Services), with the direct connection of SNMP agent and Web service stack. External requests to the SNMP agent are translated by agentware either into calls to the appropriate Web Service stack methods, where further execution takes place, or passed through device handlers to specific devices in network. Notification mechanism delivers response through message queue, where notification handler generates response via initial calling interface. Within the C2 center, the data processing nodes house the servers where the software that performs these tasks is installed. The fusion integration server, which contains a central process for processing sensor data and aggregation monitor component, accesses the web services of the application server mostly through its Web clients. Most of monitoring data originates from sensors with one-way communication style and is collected on this server. Besides monitoring purposes, this data is also used in sensor processing and it is a direct advantage of star-shaped HFSWR network topology. User communication is based on the HTTP REST software architecture style. The data is transmitted in JSON format. This applies to both internal and external nodes, which need to exchange data with the application server. The exception is, of course, the external NMS application. The web service stack was implemented in C# programming language and hosted on IIS. There are 3 main groups of services: control, monitoring and surveillance data services. Control services, represented in Fig. 4 with light green color and described through their service contract names, are dedicated for control of power amplifiers, power distribution units and configuration and process state control. Monitoring services (yellow color on Fig. 4) are dedicated for monitoring of equipment, alarms and current process states data flow, along with error logging, diagnostics and access to middleware historian. Surveillance data services (dark green color on Fig. 4) are dedicated to operational surveillance data exchange between inner and outer HFSWR network nodes.

Typical information flow will be demonstrated on one scenario with one HFSWR data scan, presented on Fig. 5. On new HFSWR data available, proxy gateway component activates its data handler, which packs sensor readings and dispatches them through communication channel, via file transfer protocol and ethernet network. Upon reception of data packet in HFSWR data node, data is processed and passed through fusion/integration node routines, where COP output, possible alarms and other information are generated, via notification mechanism. Notification managers of aggregate monitor component and web client from C2 transport node process and pack messages into JSON format and send it via its Web clients to application server data and monitoring services. Monitoring data is further translated via agentware component into format suitable for SNMP OID data storage, where SNMP traps mechanism generates trap notifications, if necessary. The data from monitoring, surveillance and SNMP OID repositories is then available for external clients, who read it via appropriate service or SNMP calls.

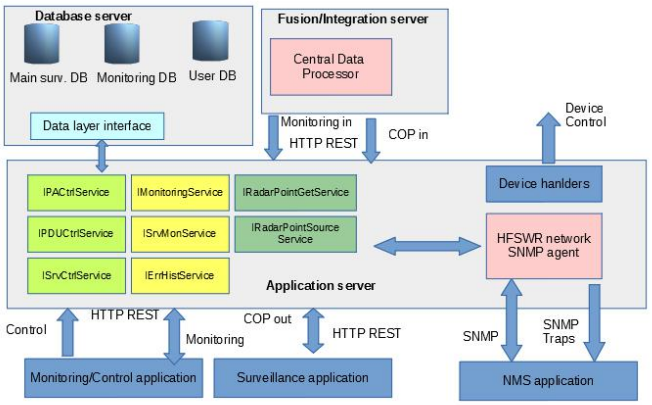


Figure 4. CC platform logical architecture

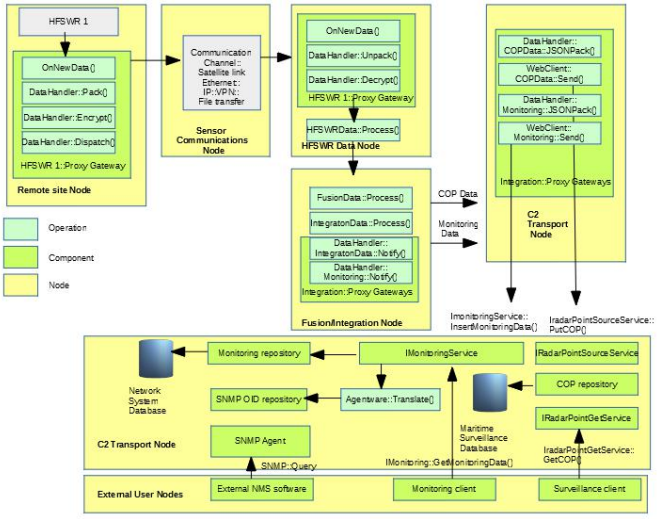


Figure 5. Information flow diagram for one HFSWR data scan scenario

IV. Demonstration

Current implementation on the CC platform takes up the resources of one physical server with 2 CPUs and a total of 24 CPU cores. IoT architecture of the HFSWR network is demonstrated with screen-shoots from application layer

utilities. First, on Fig. 6, a typical view from the maritime surveillance application client is presented. The client presents all possible view layers, including integrated views, and presents selected target details, including integration information. For example, selected target details (white hexagon marker) show that it is a target from HFSWR fusion view layer, integrated with AIS MMSI 27321110, followed for more than 5 hours. Other useful details, including velocity, course and estimated coordinates are presented as well. Monitoring of the equipment and remote site node measurements are read and displayed in appropriate monitoring application. In Fig. 7, one of its windows is presented. On the left side of screen there are general alarms, in charge for general HFSWR network state. On the right side, there are particular alarms and measurements from remote sites, special areas of interest (marked as yellow polygons on Fig. 6) and ionospheric interference zone alarms overview. SNMP interface is tested via simple SNMPB application [12], installed on application server. This application is used as Management Information Base (MIB) file browser and SNMP agent tester. MIB file for the HFSWR network is loaded and main OIDs (Object Identifiers) are displayed on Fig. 8, left. SNMP query is tested on marked OID, named vMsMonitor, and results are shown on the right side of Fig. 8.

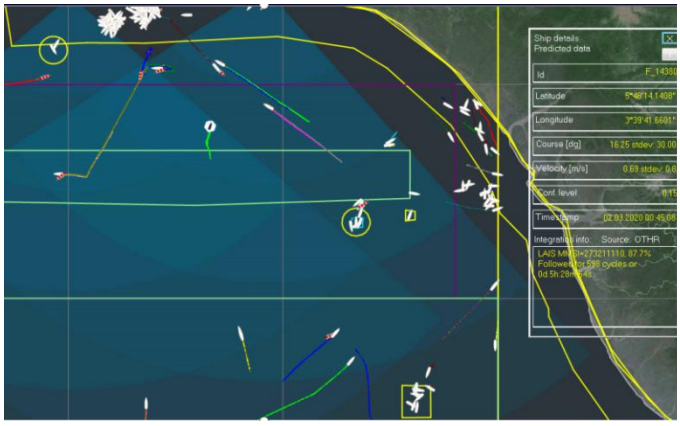


Figure 6. Maritime surveillance client application screen.

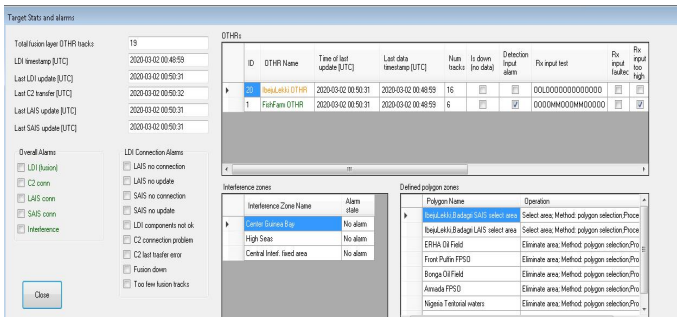


Figure 7. A window with monitoring alarms from maintenance/monitoring client application

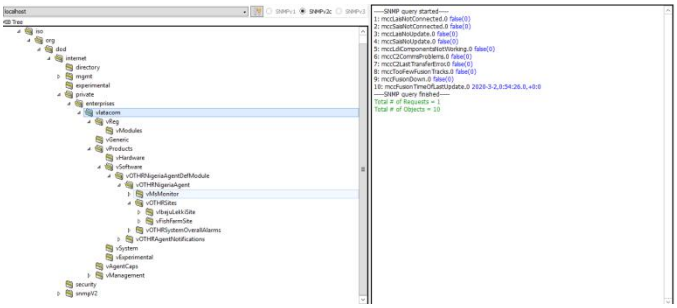


Figure 8. SNMP query test of HFSWR network's main SNMP agent in SNMPB utility

V. Conclusion

This paper presents the basis for building the IoT architecture of the HFSWR network in the IMS concept. Exploiting network's topology features, an aggregated surveillance and monitoring solution via Web REST services and popular SNMP protocol has been developed as a Web application, working in private cloud. Such solution allowed direct monitoring and collection of data from one set of nodes, located in C2 center, greatly simplifying distributed middleware, which contains minimal number of intermediate components, thus increasing the reliability and availability of system components. Finally, it can be concluded that an efficient system for remote monitoring of the HFSWR network has been developed, with a rapid response of both the system and the user to unforeseen events, while providing complete insight into all the functionality of the network to its operators. For the future work, further expansion of presented IoT architecture is planned. At first glance, the system topology seems to have a small number of process nodes, but the whole presented architecture is scalable and designed in IoT sense to provide flexibility in the integration of additional process nodes and future more modern and secure solutions, such as more reliable private cloud technology stacks based on the Linux operating system.Smart, autonomous, run-time configurable software agent concepts that will manage remote site-C2 center data exchange will be introduced. Besides on-site sensor and communication channel interfacing, their role will also be the management of uploads of large amount of data, accumulated during remote site's communication offline stages, via narrowband and noisy communication channel, thus supporting implementations of scenarios for communication in harsh environments.

References

[1] L.D. Sevgi, A.M. Ponsford, H.C. Chan, "An integrated maritime surveillance system based on high-frequency surface-wave radars Part 1: Theoretical background and numerical simulations", IEEE Antennas and Propagation Magazine, vol. 43, no. 4, pp. 28-43, Aug. 2001.

[2] L.D. Sevgi, A.M. Ponsford, H.C. Chan, "An integrated maritime surveillance system based on high-frequency surface-wave radars Part 2: Operational status and system performance", IEEE Antennas and Propagation Magazine, vol. 43, no. 5, pp. 52-63, Oct. 2001.

[3] Buckl, C., Sommer, S., Scholz, A., Knoll, A., Kemper, A., Heuer, J., Schmitt, A., 2009. "Services to the field: an approach for resource constrained sensor/actor networks". In: Proceedings of WAINA, Bradford, United Kingdom.

[4] Enji, S., Zhanga, X., Lib, Z., 2012. "The internet of things (IOT) and cloud computing (CC) based tailings dam monitoring and prealarm system in mines". Saf. Sci. 50 (4), 811–815.

[5] Drenoyanis, A.; Raad, R.; Wady, I.; Krogh, C. Implementation of an IoT Based Radar Sensor Network for Wastewater Management. Sensors 2019, 19, 254.
[6] H. Nwana, “Software agents: An overview,” Knowl. Eng. Rev. J., vol. 11, no. 3, 1996.
[7] A. Ouaddah, H. Mousannif, A. Abou Elkalam and A. Ait Ouahman, "Access control in IoT: Survey & state of the art," 2016 5th International Conference on Multimedia Computing and Systems (ICMCS), Marrakech, 2016, pp. 272-277, doi: 10.1109/ICMCS.2016.7905662
[8] Gameiro, A., Castanheira, D., Sanson, J. et al. Research Challenges, Trends and Applications for Future Joint Radar Communications Systems. Wireless Pers Commun 100, 81–96 (2018).
[9] Petrovic R., Simic D., Drajic D., Cica Z., Nikolic D., Peric M. “Designing Laboratory for IoT Communication Infrastructure Environment for Remote Maritime Surveillance in Equatorial Areas Based on the Gulf of Guinea Field Experiences”. Sensors. 2020; 20(5):1349.
[10] Centerity Monitor information page, https://www.centerity.com, available online 17.04.2020
[11] Spiess, P., Karnouskos, S., Guinard, D., Savio, D., Baecker, O., Souza, L., Trifa, V., 2009. “SOA-based integration of the internet of things in enterprise services”. In: Proceedings of IEEE ICWS, Los Angeles, Ca, USA.
[12] SNMPB application download page, https://sourceforge.net/projects/snmpb, available online 27.04.2020

Implementation of the Monitoring System for HFSWR-based Maritime Surveillance Networks

Nikola Stojkovic, *Member, IEEE*, Dejan Nikolic, *Member, IEEE,* Vladimir Orlic, *Member, IEEE,* Bojan Dzolic, Nikola Lekic, *Member, IEEE*

***Abstract*—Designing surveillance and monitoring applications for HFSWR (High Frequency Surface Wave Radar) networks faces with significant challenges, where it is mandatory to involve literally thousands of factors in order to create a comprehensive monitoring system, and respond to the needs of efficient maritime surveillance at the same time. All these factors make HFSWR network a complex system for distributed measurement and control. Described structure of sensor processing software and surveillance utility features allow system operators to make insight into specific surveillance situations, active protection of restricted naval areas and resolving missing vessel identifications, which represents huge problem in Exclusive Economic Zone (EEZ) monitoring of underdeveloped regions. Implemented solution for surveillance and monitoring system is presented through demonstration of its practical functionality, obtained from operational HFSWR network, installed in Gulf of Guinea, Africa.**

***Index Terms*—HFSWR network, Distributed sensor network, Integrated Maritime Surveillance, HF-OTH radar, Maritime security**

I. Introduction

HFSWR is a sensor used for over the horizon surveillance in maritime applications, whose principles of operation are described in [1]. These sensors are often used in conjunction with AIS (Automated Identification System) [2] - [3], forming HFSWR networks, a well-known IMS (Integrated Maritime Surveillance) concept, based on HFSWR sensor (eventually supported with other low-range sensors) and described in [4] - [5]. One such network based on HFSWRs, presented in [6], has been installed in Gulf of Guinea.

Nikola Stojković is with the School of Electrical Engineering, University of Belgrade, 73 Bulevarkralja Aleksandra, 11020 Belgrade, Serbia and the Vlatacom Institute, Belgrade, Bulevar Milutina Milankovića 5, 11070 Novi Beograd, Serbia, (e-mail: nikola.stojkovic@vlatacom.com)

Dejan Nikolic is with the Vlatacom Institute, Belgrade, Bulevar Milutina Milankovića 5, 11070 Novi Beograd, Serbia, (e-mail: dejan.nikolic@vlatacom.com)

Vladimir Orlic is with the Vlatacom Institute, Belgrade, Bulevar Milutina Milankovića 5, 11070 Novi Beograd, Serbia, (e-mail: vladimir.orlic@vlatacom.com)

Bojan Dzolic is with the Vlatacom Institute, Belgrade, Bulevar Milutina Milankovića 5, 11070 Novi Beograd, Serbia, (e-mail: bojan.dzolic@vlatacom.com)

Nikola Lekic is with the Vlatacom Institute, Belgrade, Bulevar Milutina Milankovića 5, 11070 Novi Beograd, Serbia, (e-mail: nikola.lekic@vlatacom.com)

Since HFSWR based maritime network is a complex system, it consists of hundreds of subsystems, and thus equal number of possible monitoring and control probes. Just to name a few, there are: power amplifiers (PA), controllable power distribution units (PDU), ambient sensors, power measurement devices and many other sensor and controllable devices. Almost any of them could cause problems, that can jeopardize basic functionality of individual subsystems or even entire network. In [7] a typical topology of HFSWR networks is presented, from which certain design considerations about entire measurement / monitoring system can be deduced. First of all, access to measurement points could be organized from a single node and allow deployment of one aggregated monitoring module, which will ease the distribution and processing of monitoring data. Purpose of this paper is to give insight into HFSWR network, as a complex system for distributed measurement and control [8], to give insight into structure of data processing software and to give guidelines for the design of monitoring and surveillance applications on a practical example, by providing demonstration from working system.

In Section II an overview of HFSWR network sensors is made. Sensors, formats of their measurement data and equipment monitoring messages have been reviewed. After that, in Section III, monitoring and surveillance software structure has been described. Surveillance and monitoring applications design considerations and demonstration is presented in Section IV. The paper is concluded in Section V.

II. HFSWR network sensors overview

In order to identify monitoring and control probes in the HFSWR network, a brief description of sensor outputs and format of both sensor and monitoring data is necessary. Since it is not possible to present every specific detail, an overview is made from the system level point of view.

A. HFSWR as a sensor in the HFSWR network

The output result of the HFSWR as a sensor is, of course, its measurement data set. The HFSWR, after a scan period of approximately 33 seconds is completed, creates so-called Constant False Alarm Rate (CFAR) registration data, i.e. a file which contains data regarding potential target detections. Each data entry contains information such as distance from HFSWR, bearing relative to the true north position, radial velocity,

corresponding standard deviations of measurement errors, CFAR Signal-to-Noise Ratio (SNR) and total cell SNR. It is important to note that this is textual file, which is backed up to a server on the remote HFSWR site. The file is transferred to the appropriate server on HFSWR data node in the C2 center via file transfer protocol or IP socket connection. An example of a listing of one such file is given in Fig. 1. In addition to its sensor function, the radar also performs regular calibration adjustments, which play a major role in the diagnosis of correct signal reception and transmission. Also, system possesses a notification mechanism concerning of critical events and hardware failures.

```
118.669  177.020    5.714   1.433   2.486   0.864   35.654   35.772
290.665  177.499    9.754   1.293   2.271   1.768   13.506    8.249
108.271  181.191    5.577   1.281   2.438   0.879   37.794   33.468
252.134  182.024   -2.889   1.313   2.528   2.799    9.395    5.243
111.143  183.019   -6.013   1.344   2.514   1.210   20.761   18.098
275.041  183.905   17.544   1.195   2.019   2.116   11.294    7.080
102.843  184.115   -6.078   1.380   2.439   0.873   24.191   19.100
120.134  184.027   -5.932   1.115   2.500   1.191   21.648   16.687
136.795  183.966   -5.893   1.337   2.501   1.224   18.174   14.369
149.499  184.048    6.015   1.017   2.447   0.885   25.786   28.901
156.265  184.150    6.331   1.100   2.467   0.918   26.778   23.903
162.483  183.878   -5.216   1.427   2.348   1.057   19.663   13.704
234.151  184.476   27.184   1.102   2.322   1.662   13.158    7.822
 15.167  185.097    7.832   1.017   2.503   2.093   14.820   33.055
```

Figure 1. Listing of a CFAR registration file with radar detection. Columns in the file represent range in km, bearing in degrees, radial velocity in m/s, standard error deviations of respective measurements, relative to resolution cell size, CFAR SNR in dB and total cell SNR in dB, respectively.

Calibration measurements are carried out every hour, during so-called Direct Path Test DPT) procedure [9]. These measurements are required to be stored for regular inspection, indication of interference and analysis of radar behavior. In order to do that, the amplitudes of the spectrum in a simple binary format are transferred to the HFSWR data node at the C2 center, where the DPT results are analyzed and alarms are generated as needed. This method allows not only on-site HFSWR self-calibration, but also provide possibility for network-aided, remotely controlled calibration scheme and better traceability of problems [10].

HFSWR has a notification mechanism, which is linked to the monitoring software with its own mechanisms for responding to notifications. The original notifications are textual with tab separated fields and the format is defined in Table 1. It specifies the remote HFSWR site ID that sent the message, the module that generated the message (message source), the message priority (values 1 to 4, with increasing priority), the destination to which the message is intended (binary coded two digits, the first describes the maintenance operator and the second describes supervisors of the entire network), the descriptive text of the error message and the message code. This mechanism transmits messages that are vital to the functioning of the equipment: temperature reports, voltage states on components (mains voltage and DC voltage references), failures in calibration tests on individual channels, states of individual components, etc. This is the reason why these messages are treated intensively in the software alarm system. It is convenient to transfer them from the site by file transfer or via socket connections to the servers in the C2 center. Due to their importance in radar diagnostics, they are backed up twice, both on the site server and on the server in the C2 center, and represent the basic diagnostic level of monitoring of HFSWR equipment.

TABLE 1.

NOTIFICATION MESSAGE FORMAT AND EXAMPLES

Remote site ID	Message source	Prior.	Dest.	Message description
1	Remote Control-FCR	1	10	Frequency Control unit seems to be off, although it should be running. Resetting the unit...
1	Remote Control-PSR	1	10	All Voltages and all temperatures in Power Supply unit are wrong ! No status cable connected.
20	DPT_IQ_3	3	10	I/Q-Balance of channel 3 out of range, according to DPT test. Check DPT results.

B. AIS as a sensor in the HFSWR network

AIS sensor data is available via terrestrial and satellite feed links (Land AIS and Satellite AIS feed), to which the corresponding client components can connect. These components are part of the fusion and integration nodes in the C2 center infrastructure. LAIS messages are most often transmitted by UDP or TCP/IP protocol to a specific destination - a collection endpoint, where a special software module - AIS concentration process, is installed, which has the role of filtering messages and eliminating duplicated and obsolete messages. The LAIS message format is a standard NMEA AIVDM [11], while the SAIS uses a slightly expanded, vendor specific format, which consists of concatenated header with fields separated usually by a semicolon character, in which the time of message reception, message specifics, and sender identification are specified. After the header, message continues with the original AIVDM message. The result of the AIS concentration process is a uniform stream of AIS messages, delivered to the connected clients with the knowledge of the access parameters (IP address, port) via IP socket communication.

III. MONITORING AND SURVEILLANCE SOFTWARE STRUCTURE

The structure of surveillance and monitoring software is composed of one or more modules, dedicated to tracking, fusion and sensor integration tasks, and monitoring components. Basically, highest level of software structure can be represented with one central processing module, which contains definitions of alarms, within its monitoring rule-based

engine. Short description of main software components follows in next subsections.

A. Long Distance Integrator module

The software module that will perform the functions of sensor fusion and integration in the network's corresponding nodes is called the Long Distance Integrator (LDI) module. It has a primary role of target integration over longer distances. The reason for this name lies in the fact that closer targets are usually subject to consideration by Microwave (MW) radars, provided that aforementioned radars exist in the IMS. The structural diagram of the module is shown in Fig. 2.

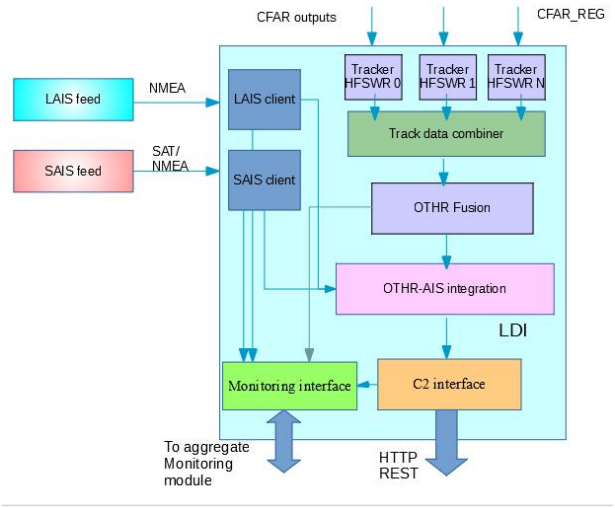


Figure 2. LDI module structure and input and output formats

The LDI module inputs are derived from HFSWR and AIS sensors, and handled by dedicated components, whose performance needs to be monitored in some way. First of all, it is necessary to describe the operation of each component individually, and define notifications and alarms which are suitable for monitoring tasks. The LDI module consists of the following components:

- LAIS and SAIS clients. The components connect to the LAIS or SAIS feed on the basis of predefined access parameters (IP address, port). Once the connection is established, the type of communication is one-way, in the direction of the client. The interface type is based on IP protocol and standard socket communication.

- **Tracker components**. One Tracker component is assigned to each particular input connection towards the HFSWR sensor. Its input interface accepts CFAR registrations from an associated sensor and executes an target tracking algorithm [12]. The component output product is a list of track registrations with a given time stamp, inherited from the received set of CFAR registrations.

- **Track Data Combiner**. This component accepts track registrations from Tracker components and performs time-pairing of the data from all HFSWR sensors. The data in this component arrives, in general case, marked by different timestamps, while being arbitrarily late, depending on the state of the communication channel and processing. The component pairs data from all Tracker components according to the specific time criteria.

- OTHR Fusion component. This component performs fusion of the combined information of the Track Data Combiner component [13]. It has its own time-stamping mechanism, relies on input reference information and provides an unified HFSWR picture (UHP), i.e. operating image for the associated set of HFSWR sensors.

- **OTHR - AIS Integration component**. The component fuses an UHP data and all AIS sensor data. This component provides a common operational picture (COP) [14] for a given set of associated HFSWR sensors.

-C2 interface component. This component is used to format the output. It transforms data into JSON format. Then the HTTP client, as its integral part, connects to the appropriate service in the C2 center and submits the data using HTTP REST methods.

- **The monitoring interface** serves to collect notifications and alarms from each individual component of the LDI module. This data is then conveniently presented in an internally used format and passed on to a higher level of processing.

B. Central process

Central process (CP) is a process that has a function analogous to the one given in [5] by a multi-sensor association processor (MSAP). CP associates HFSWR output detections to create multi-layer target associations and combines HFSWR and AIS sensor data to form a partial or complete operating images. The role of CP is to provide centralized treatment of sensor processing and monitoring of the entire HFSWR network, which is made possible due to the specific star-shaped topology of the HFSWR network [7]. In the general case of a widely distributed HFSWR network, it is possible to adopt the principle that individual LDI modules collect data from only those sets of HFSWR sensors whose coverage zones overlap, so fused image is required for these separated areas, since they can be viewed as separate network subsystems. Individual measurements from other sensors, calibration results, etc. can be delivered to a specific set of endpoints, different interfaces, suitable for the scenario and communication conditions in the field. The presented generalized approach is depicted on Fig. 3. The implementation of the CP is based on hosting the process on classic windows services, and the number of processes and separation by functionality (for example, LDI modules in one process, monitoring of sensors in a network in another) is a matter of preference. In the simplest case, the topology with fewer branches, the single-process version represents the optimal solution. In such implementations, it is possible to fully aggregate the monitoring information and create a unique reporting system, before the final, application layer.

The aggregation monitoring module provides a complete monitoring image of the HFSWR network in the case described above. The only exception is the system monitoring of the central process itself, which is either the task of a special component or can be left to, for example, the Windows management interface. The information accepted by the CP through its monitor endpoints is generally one-way in nature and of different types, from simple ambient sensor temperature measurements, hard disk drive occupancy on site servers to signal calibration measurements on each of the HFSWR receiving antenna arrays. The notifications created by the HFSWR are received in central process and its event handler triggers certain alarms. Temperature events, events related to failed DPT calibrations on individual channels, as well as all those with higher priority, without referring to the normalization of the condition, raise the level of maintenance alarms for the HFSWR.

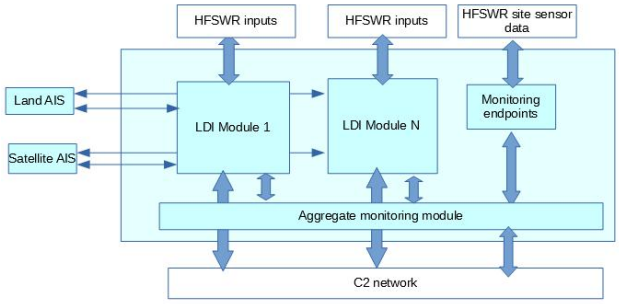


Figure 3. Structure of the central processing and monitoring process - single-process realization

C. *Measurement/control summary and alarm system build-up*

All measurements and controls within the HFSWR network, for monitoring purposes, can be classified into 9 groups, see Table 2.

TABLE 2.

MEASUREMENT AND CONTROL GROUPS IN HFSWR NETWORK

Group	Description
1	Ambient measurements
2	PDU Control and measurement of power consumption
3	Measurements of reflected and radiated power
4	Rx input validation (DPT) tests
5	Multi-server cluster info and fail-overs
6	Control and status of Power amplifiers
7	Control, notifications and monitoring in C2 center
8	Monitoring/control of servers/PC machines
9	HFSWR acquisition status

In the system, besides the sensors described that perform the main function in the HFSWR network, there are a number of others, whose role is also important, only in other aspects. Monitoring system owns an alarm mechanism to invoke user reaction on certain critical failures. The nature of alarms is predominantly related to sensors, but there are also operational surveillance alarms, related to zones of special interest, maritime rules of engagement and rules of surveillance. Not every alarm in the system produces the same response from both software and maintenance operators. There is a need to define different levels of severity of alarms, i.e. the states they produce within the system. Because of this, alarms are often followed with a new feature, which characterizes them as two-static, three-static or multi-static.

IV. SURVEILLANCE AND MONITORING SYSTEM DEMONSTRATION

Surveillance and monitoring system will be demonstrated with application layer utility screen-shots from operational HFSWR network, installed in Golf of Guinea, Africa. Purpose of this demonstration is to present functionality of surveillance application, that reflects actions of different software blocks in LDI, and monitoring application, that should provide insight into main monitoring features and alarm mechanism. In Fig. 4, surveillance application elements are presented, alongside with all-layer views. Protected maritime areas (1), including the polygon of HFSWR exclusions in territorial waters, reserved for MW radar operations (2), and interference alarm zones (3) are marked, to indicate zones with special rules of operator and alarming engagement.

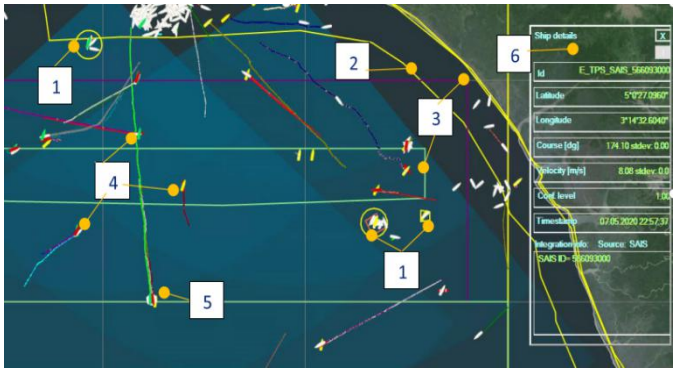


Figure 4. Surveillance screen elements overview

Note that in protected areas, UHP picture elements are excluded from surveillance. The reason for this rule lies in the fact that HFSWR is used to detect vessels entering or leaving such zones, while zones within MW range are better covered with aforementioned sensors due to its better resolution. Individual targets (4) are represented with ship markers of different colors. Green and red markers are reserved for particular HFSWR targets. Red markers represent UHP, while white markers are reserved for AIS targets. Details of the selected target, white hexagon (5), are presented on target details view, on the right (6). Another important aspect of surveillance application is its layered view. It has proven useful to have the ability to distinguish target origin in COP view and to visualize its tracking data by different sensor sources, which

will be demonstrated in an example, later in the text. On Fig. 4, an all-layer view is shown, which means that multiple markers exist for the same target, originating from different sources, both sensors and intermediate processing layers. There are 4 types of basic view layers: tracking, UHP, AIS and COP view layer and each corresponds to one processing stage in LDI components or sensor source.

Primary, lower tracking layer view is the result of Tracker components in LDI, associated with every HFSWR, see Fig. 5.

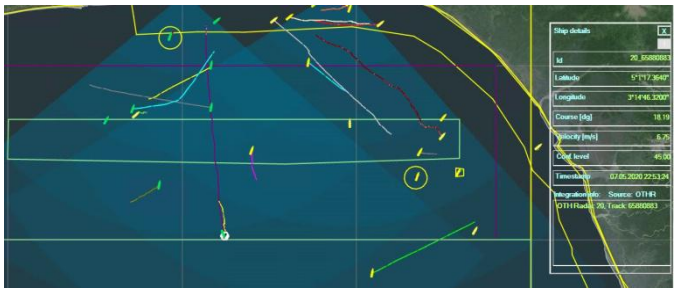


Figure 5. Lower tracking layer view.

After the fusion process, UHP is formed in the fusion component of LDI. This result is shown on Fig. 6, where UHP targets are presented with red markers. Comparing the picture from Fig. 6 with the one on Fig. 5, proper grouping of targets can be easily noted.

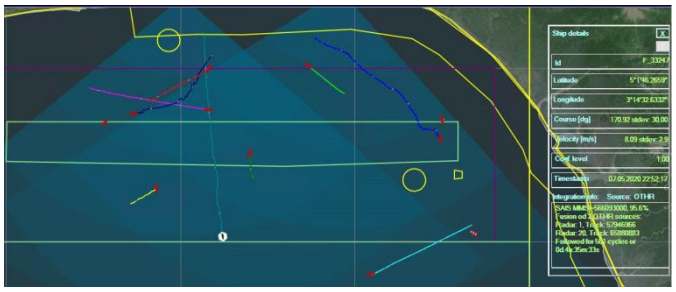


Figure 6. UHP layer view

Fig. 7 displays combined UHP and AIS view, where AIS targets are displayed with white markers. This figure is shown only to be compared with COP layer view, on Fig. 8. At this point, all particular UHP and AIS targets are completely integrated, forming a reliable operational view. Color of markers indicates origin of the target, giving advantage to HFSWR (red markers), even if target has multiple origins. As an example, a selected target on Fig. 7 has multiple sensor origins. It is tracked by 2 HFSWRs, out of which one UHP target is formed, then by AIS, and, at the end, on COP layer, a single target is formed. On Fig 7, one AIS target, marked with red circle, stopped emitting its AIS positions. On the other hand, it was clearly followed with HFSWR, which was marked with white circle. If we switch to COP view on Fig. 8, it can be noted that AIS target from previous view is missing. Reason for that lies in HFSWR-AIS integration rules [14], where integration link remains active for up to 6 hours after last emitted AIS position report. This example not only emphasizes the importance of layered view in surveillance applications, but also indicates huge problems for surveillance operators on the field [14].

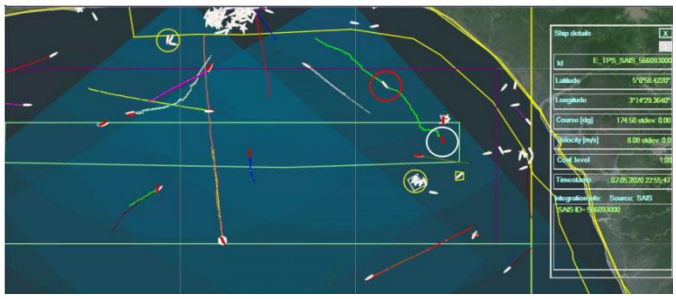


Figure 7. Combined AIS and UHP layers view

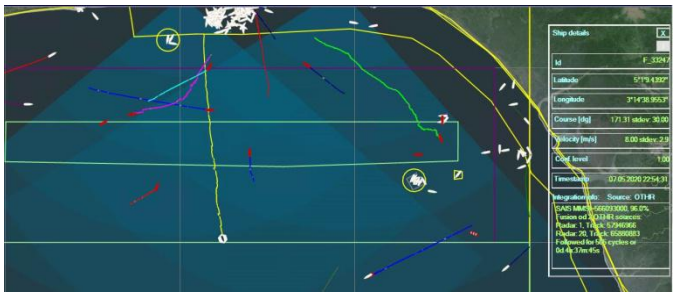


Figure 8. COP layer view

Monitoring diagnostics starts with a alarm overview panel, Fig. 9, where main monitoring groups are labeled. Overall HFSWR network alarms group (1) provides quick information about the health of the network. This group of alarms, combined with overall remote site states, is usually placed on graphical user interfaces of surveillance operators, for informational purposes. LDI connection alarms (2) inform about general state of LDI components. Interference zonal alarms (3) inform the users about possibility of very high ionospheric interference in defined zones. Alarms in these areas usually lead to lower detection probabilities, tracking problems and higher false alarm rate.

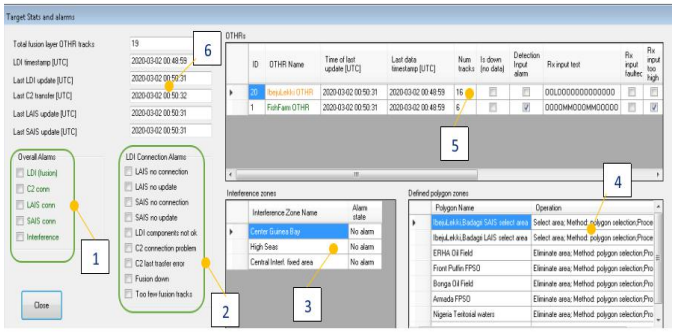


Figure 9. Main alarm and monitoring panel

Special polygonal areas (4) include protected areas, e.g. oil rigs and HFSWR exclusion zones. Protected areas, if configured, provide intrusion alarms, when unidentified vessels approach or leave such zones, based on defined surveillance rules. Group of remote sites (5) provides general state and overview of the most important equipment and

measurement alarms and notifications, e.g. quick overview of HFSWR calibration (DPT) tests, active maintenance alarms etc. Main notifications group (6) provides timing information inside network's CP. By clicking certain fields, e.g. one of the remote site rows, another panel opens with lower level alarms, related to that feature and possibilities to control and monitor other components and sensors. On this way, monitoring of the HFSWR network allows maintenance operators easy and fast diagnostics of possible problems.

V. CONCLUSION

In this paper a comprehensive network's alarm system has been presented, through which operators can achieve fast and descriptive insight into the entire HFSWR network parameters and ongoing surveillance operation state. Described solution was made feasible through aggregate monitoring principle, where practically all monitoring data is accessed or gathered from one node in C2 center, due to specific star-shaped topology of HFSWR network's IoT infrastructure. Specific sensor data processing structure and surveillance client utility implementation allows layered view of surveillance coverage area, which helps operators resolving critical situations, missing vessel identifications and special area protection. Along with detailed explanation of the adopted structure for implementation, its practical application was presented as well, through various examples of real – life data obtained from one operational IMS based on HFSWR network. Engagement in both surveillance functionality and main network parameters monitoring was presented from the perspective of the end user, in order to demonstrate achieved functionality at the top – level of application at its current version. While solutions of this kind are commonly tailor – made for particular system and the user, general flexibility of proposed structures makes it quite adaptable and thus convenient for easy reshaping in data representation at the top – level stage. For the future work, further optimization of network's alarm system and development of more functionally reach application utilities is planned, in order to further alleviate monitoring of this complex system for distributed measurement and control.

REFERENCES

[1] G. Fabrizio, High Frequency Over-the-Horizon Radar: Fundamental Principles Signal Processing and Practical Applications, New York, NY, USA:McGraw-Hill, 2013.

[2] ITU, „Recommendation M.585-7 Assignment and Use of Identities in the Maritime Mobile Service," ITU, Geneva, Switzerland, 2015.

[3] International Maritime Organization, Resolution MSC.74 Annex 3 Recommendation on Performance Standards AIS, London: IMO, 1998.

[4] D. Sevgi, A. M. Ponsford, H. C. Chan, "An integrated maritime surveillance system based on high-frequency surface-wave radars Part 1: Theoretical background and numerical simulations", IEEE Antennas and Propagation Magazine, vol. 43, no. 4, pp. 28-43, Aug. 2001.

[5] L. D. Sevgi, A. M. Ponsford, H. C. Chan, "An integrated maritime surveillance system based on high-frequency surface-wave radars Part 2: Operational status and system performance", IEEE Antennas and Propagation Magazine, vol. 43, no. 5, pp. 52-63, Oct. 2001.

[6] D. Nikolic, N. Stojkovic,P. Petrovic, N. Tosic,N. Lekic,Z. Stankovic, "The high frequency surface waveradar solution for vessel tracking beyond the horizon".Facta Univ. Electron. Energetics, 2020,33, 37–59.

[7] N. Stojkovic, V. Orlic, M. Peric, D. Drajic,A. Rakic "Concept of System for Surveillance and Monitoring of IoT HFSWR Network", Proc. of IcETRAN, Sept. 2020, under review

[8] H.F. Rashvand, J.M. Calero "Distributed sensor systems: practise and pplications", 2012, John Wiley & Sons Ltd.

[9] P. Petrovic, N. Grbic, B. Dzolic, N. Lekic, M. Peric, "Software for Monitoring of Direct Path Test Data for HFSW Over the Horizon Radar," Proc. of IcETRAN 2018, Palic, SR, June, 2018.

[10] A. Carullo, F. Ferraris, M. Parvis, "Traceability issues in distributed measuring systems", XVIII IMEKO World Congress, 2006, Brasil

[11] NMEA 0183 standard overview, web page: https://www.nmea.org/content/STANDARDS/NMEA_0183_Standard available online: 10.05.2020

[12] N. Stojkovic, D. Nikolic and S. Puzović, "Density Based Clustering Data Association Procedure for Real–Time HFSWRs Tracking at OTH Distances," in IEEE Access, vol. 8, pp. 39907-39919, 2020, doi: 10.1109/ACCESS.2020.2976481.

[13] D. Nikolic, N. Stojkovic, Z. Popovic, N. Tosic, N. Lekic, Z. Stankovic, et al., "Maritime over the horizon sensor integration: HFSWR data fusion algorithm", Remote Sens., vol. 11, no. 7, pp. 852, Apr. 2019.

[14] D. Nikolic, N. Stojkovic and N. Lekic, "Maritime over the horizon sensor integration: High frequency Surface-Wave-Radar and automatic identification system data integration algorithm", Sensors, vol. 18, no. 4, pp. 1147, Apr. 2018

Vowel recognition using formant analysis and neural networks

Emilija Kisić, Goran Dikić and Vera Petrović

***Abstract*—In this paper, a system for vowel recognition using formant analysis and neural network is described. Complete procedure for vowel recognition which consists of historical dataset forming, dataset preprocessing, power spectral density estimation, formant extraction and neural network training and testing is given. Finally, gain results are discussed and it is shown that with first three formant frequencies and with appropriate neural network architecture vowels can be classified and recognized with big accuracy.**

***Index Terms*—formant frequencies; vowel recognition; neural network.**

I. Introduction

Automatic speech recognition (ASR) is scientific field which attracts scientist and researches for more than 60 years. Full development of this scientific field has happened with the transition from analog to digital systems. Recently, with global technological development, ASR has gained application in large number of applications that can be found in everyday life [1].

In this paper automatic vowel recognition using formant analysis and neural network is described. It is known from acoustic theory of speech production, that every uttered vowel has three main resonant frequencies which are called formant frequencies or just formants [2, 3]. In this paper is described complete procedure for automatic vowel recognition. First step was forming of dataset which consists of recorded vowels uttered by male speakers. After that, dataset was processed for noise cleaning and for extraction of useful part of every recorded vowel. After dataset preprocessing, power spectral density estimation of signals is performed, so formants could be extracted [4]. After power spectral estimation, first three formant frequencies were extracted from each vowel using adaptive algorithm. Extracted formants were used for neural network training [5, 6]. After training, neural network was tested on new vowels that were not in historical dataset. Very good results were gained during training and during neural network testing. System for automatic vowel recognition using formant analysis and neural network gave good results and did recognition with big accuracy. It is shown that first three formant frequencies can make good classification between vowels and with good neural network design system can make vowel recognition with big accuracy. On Fig.1 algorithm for vowel recognition is shown.

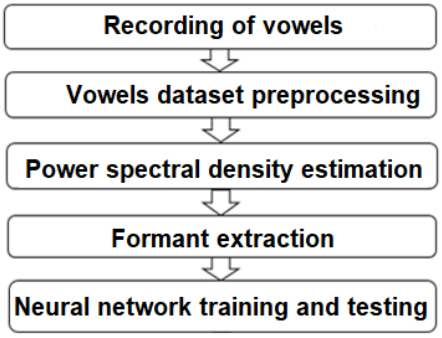


Fig.1. Algorithm for vowel recognition.

First four steps of the algorithm were performed in programming language *Matlab*, while neural network training and testing was performed in programming language *Python.*

II. Acoustic Theory of Speech Production

In acoustic theory term sound refers to vibration. Vibrations are the cause of sound waves production, which propagate due to particle oscillating in the medium through they travel. Because of that, basic principles of physics must describe production and propagation of sound in vocal tract. For vocal tract modeling, detailed acoustic theory must take into account next factors: time variability of the shape of vocal tract, losses due to thermal conductivity and viscous friction on the walls of the vocal tract, softness of the walls of the vocal tract, radiation of sound on the lips, acoustic relation between oral and nasal cavities and sound source in the vocal tract [3]. In this paper simplified mathematical model is taken which neglects the above factors. The simplest model which can describe the process of speech production is shown on Fig.2. Vocal tract is modeled as a tube of uneven, time-varying cross-section.

Frequency characteristic of vocal tract is defined as ratio between the complex amplitude of the volumetric air flow at the end and the beginning of the tube. In the field of ASR resonant frequencies of vocal tract tube are called formant frequencies or just formants.

Emilija Kisić is with the School of Electrical and Computer Engineering of Applied Studies, 283 Vojvode Stepe, 11000 Belgrade, Serbia (e-mail: emilija.kisic@ viser.edu.rs).

Goran Dikić is with the School of Electrical and Computer Engineering of Applied Studies, 283 Vojvode Stepe, 11000 Belgrade, Serbia (e-mail: gdikic@ viser.edu.rs).

Vera Petrović is with the School of Electrical and Computer Engineering of Applied Studies, 283 Vojvode Stepe, 11000 Belgrade, Serbia (e-mail: verap@ viser.edu.rs).

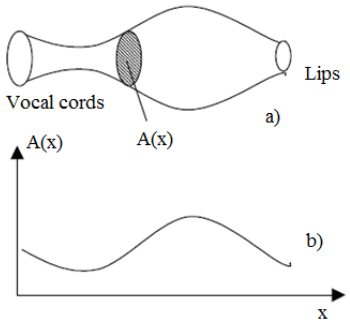


Fig.2. a) Vocal tract model b) Transfer function of vocal tract cross-section, $A(x,t)$.

Formant frequencies depend on the shape and dimensions of the vocal tract. Therefore, the spectral properties of the speech signal change over time with the change in the shape of the vocal tract. The bandwidth of the first formant (first formant frequency) is mainly determined by the vibration of the walls. The bandwidths of the second and third formant frequencies are determined with the combination of the effects of wall vibrations and radiation on the lips. At high frequencies, the influence of radiation on the lips is dominant, which at these frequencies overcomes the influence of wall vibrations, friction and thermal conductivity. Information about the content of the speech signal is hidden in the first two formants, while the third and fourth formant refers to the color of the voice [3].

III. Recording of Vowels

The first step in making a system for vowel recognition was vowel recording. Vowels are voices that arise with quasi-periodic excitation, where the function of cross-section of the vocal tract is stationary. The way in which the cross section of the vocal tract changes determines the resonant frequencies of the tract (formants) and thus the sound is produced. Each vowel can be characterized by the function of cross-section of the vocal tract used in its production. It is obvious that this is very imprecise characterization due to the natural differences that exist between the vocal tracts of different speakers. The second representation is trough the resonant frequencies of the vocal tract. Also, in this case, there are numerous variations that are expected for the same vowel that is uttered by a large number of different speakers. The period of the basic frequency of oscillation of the vocal cords, i.e. the period of the glottal wave is called the pitch period. It ranges from $(7 \div 8)ms$ for men, about $(4 \div 5)ms$ for women and about $(2.5 \div 3.5)ms$ for children. In other words, the fundamental frequency of vocal cord is about $(100 \div 120)Hz$ for men, about $(200 \div 250)Hz$ for woman and about $(300 \div 350)Hz$ for children. For this reason, the formant frequencies for female speakers and children are shifted relative to the formant frequencies for male speakers. In Table I are shown first three average formant frequencies for vowels uttered by male (native English) speakers [3].

TABLE I
Formant Frequencies for Male Speakers

First three formant frequencies for vowels pronounced by male speakers			
Vowel	F_1	F_2	F_3
A	730	1090	2440
E	530	1840	2480
I	390	1990	2550
O	570	840	2410
U	300	870	2240

Because of different values of formant frequencies for male and female speakers and children it would be a very difficult task to make a unique algorithm for vowel recognition for vowels uttered by men, women and children. For this reason we decided to make an algorithm for vowel recognition uttered only by male speakers. Original base of vowels consisted of 500 uttered vowels. Ten male speakers uttered each vowel for 10 times. Recording of each vowel lasted 2 seconds with sample frequency of 8 kHz. The sample frequency was chosen in order to satisfy Shannon's sampling theorem [7].

IV. Dataset Preprocessing

After recording of vowels, next step was dataset preprocessing. All vowels were filtrated for removing of noise and high frequencies that are not of interest for first three formant frequencies seeking. On Fig.3 speech signal which represents uttered vowel "a" after filtering is shown.

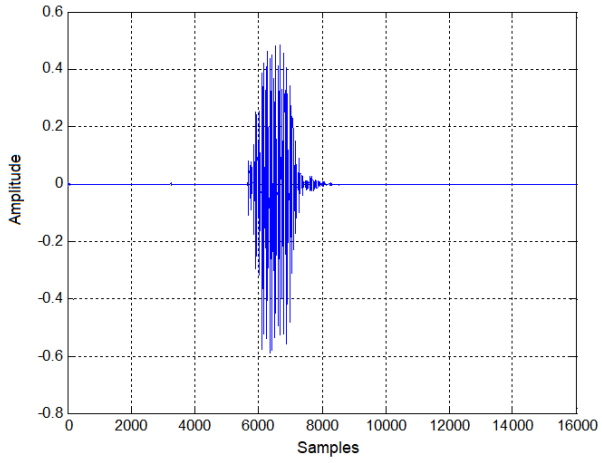


Fig.3. Uttered vowel "a" after filtering.

After filtering, it was necessary to extract only useful part from recorded signals. As we can see from Fig.3 beginning and the end of recorded signal is useless for further analysis, because it is part of a signal which represents silence. Good solution for extraction of useful part of the speech signal is calculating the square of amplitude of signal and extracting the part which is above some threshold [3]. The original

signal with squared amplitude can be defined with equation:

$$X = x(n)^2, \tag{1}$$

where n is number of samples. Part of each recorded signal that was above the threshold which is 25% of the maximum of signal with squared amplitude was extracted. Threshold was chosen empirically. In this way, only useful part of the signals was extracted and useless part was removed (part at the beginning and the end of the signal which represents silence). On Fig.4 speech signal which represents uttered vowel "a" with squared amplitude and threshold are shown.

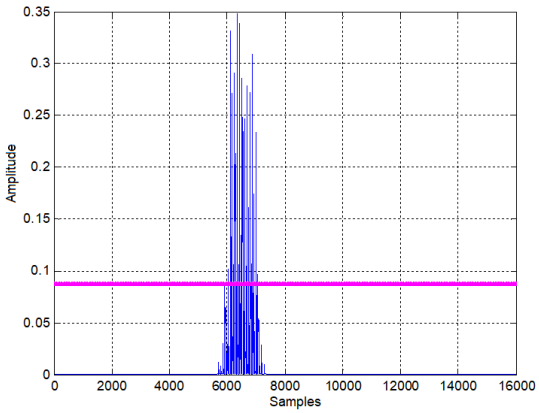


Fig.4. Speech signal which represents uttered vowel "a" with squared amplitude and threshold.

The useful part of each recorded vowel was divided into "packets" of 800 samples and thus, after preprocessing a dataset consisting of 1080 signals is obtained. Dividing of useful part of the signals is performed for increasing of dataset of uttered vowels.

V. Power Spectral Density Estimation of Speech Signals

In order to extract formant frequencies, it was necessary to perform power spectral density estimation of signals. Power spectral density (which is noted as $P_{xx}(f)$) of complex, wide stationary random process $x[n]$ is defined as:

$$P_{xx}(f) = \sum_{k=-\infty}^{\infty} r_{xx}[k]\exp(-j2\pi fk), \quad -\frac{1}{2} \le f \le \frac{1}{2}. \tag{2}$$

With $r_{xx}[k]$ is noted autocorrelation function of $x[n]$ defined as:

$$r_{xx}[k] = \varepsilon(x^*[n]x[n-k]), \tag{3}$$

where ε represents mathematical expectation operator.

The power spectral density function actually represents the distribution of power over the frequency of a random process. Since the power spectral density is a function of an infinite number of values of the autocorrelation function, the task of estimating the power spectral density based on a finite set of data is almost impossible. There are various models for power spectral density that can be assumed in order to minimize the problem of spectral estimation. The choice of a model may depend on which model best finds the required spectral characteristics. An example of this is the search for formant frequencies in speech signal. Spectral estimation via Welch [8] gave the best results comparing with other two methods-Periodogram and Blackman-Tukey [4].

Basically, Welch's method is based on time averaged periodogram which is defined as:

$$\hat{P}_{PER}(f) = \frac{1}{N}\left|\sum_{n=0}^{N-1} x[n]\exp(-j2\pi fn)\right|^2, \tag{4}$$

where N is number of samples. According to Welch's method, the number of samples N in which we perform the estimation should be divided into L intervals of M samples and each of these intervals should be multiplied by a window. For each interval multiplied by a window, a periodogram should be calculated and at the end averaging of periodograms should be performed. Multiplying with a window solves the problem of "spectrum leakage", i.e. prevents the occurrence of signals at lower levels to be masked by the side lobes of signals at higher levels, if the signals are close in frequency. Multiplying with the window reduces the signal level on the side lobes, at the cost of increasing the width of the main lobe. Welch's method solves the problem of making compromises between spectral resolution, variance and bias by allowing data intervals to overlap. As the N increases, variance decreases, the estimation is not shifted, and since the intervals overlap, we did not lose much when it comes to resolution. In this paper, Hamming window [7] was chosen with length of 128 samples, and the overlap between data intervals was 50%. On Fig.5 is presented power spectral density estimation of uttered vowel "a" via Welch.

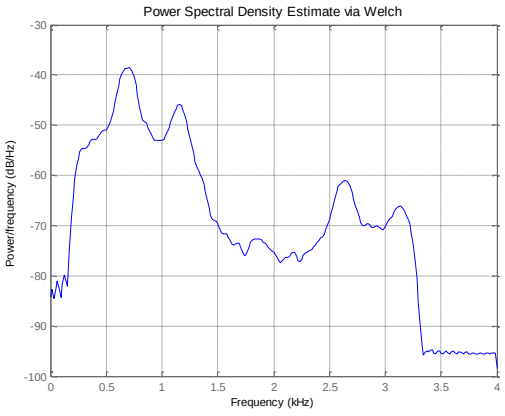


Fig.5. Power spectral density estimate via Welch for uttered vowel "a".

From Fig.5 can be noticed that formant frequencies are very clear represented as peaks in the signal, so they can be easy extracted. First formant is always global maximum,

while second and third formants are local maxima.

VI. Formant Extraction

After dataset preprocessing all recorded vowels were filtered, useful part of the signals was extracted and divided in smaller "packets" and for dataset consisting of 1080 preprocessed signals power spectral density estimation was performed. In this way dataset was prepared for formant extraction.

An adaptive algorithm was developed, based on Table I, depending on the range in which the formant frequencies are expected to appear. The first formant was found as the global maximum, and the remaining two formants were found based on where we expected them to appear, depending on the vowel. Only for the vowel "a" the range for the first formant does not overlap with the ranges for the first formant of other vowels, so it was easiest for the vowel "a" to create a separable class. The overlap of the first formant frequencies occurs with the vowels "e" and "i", but they can be classified quite successfully based on the position of the second and third formant frequencies. Also, the vowels "o" and "u" overlap by the second formant, but they can be classified based on the position of the first and third formant frequencies. Another difficulty that occurred during creating an adaptive algorithm was that formant frequencies do not always appear in expected ranges. Formant frequencies can be shifted higher or lower than expected, depending on the uttering that can vary for different speakers. In Table II average values of formant frequencies extracted from the recorded vowels are shown. There are some differences between Table I and Table II in average values of formant frequencies. Table I is formed according to male speakers which are English native speakers, and Table II is formed according to male speakers from Serbia. Another cause of these differences is uttering of vowels. Speakers which utter the vowels are from different age groups, some of them may be smokers, speakers are in different mood during recording, etc. All this affect on formant frequencies positions. Even a same speaker can utter the same vowel differently. This is the reason why ASR is very difficult task.

TABLE II
AVERAGE VALUES OF FORMANT FREQUENCIES

Average values of formant frequencies extracted from recorded vowels pronounced by male speakers			
Vowel	F_1	F_2	F_3
A	744	1177	2591
E	442	1889	2583
I	335	1790	2681
O	468	840	2518
U	354	865	2442

After formant extraction, five classes were obtained, one class for each vowel, but their overlapping could not be avoided. Classification in five separate classes based on first three formant frequencies is difficult because of formant frequencies overlap between some vowels, but the adaptive algorithm that was applied to extract the formants gives quite good results which are shown on Fig.6.

After formant extraction, for each vowel about 200 formant frequencies (first three formant frequencies) are obtained. This is now prepared historical dataset which will be used for neural network training and testing.

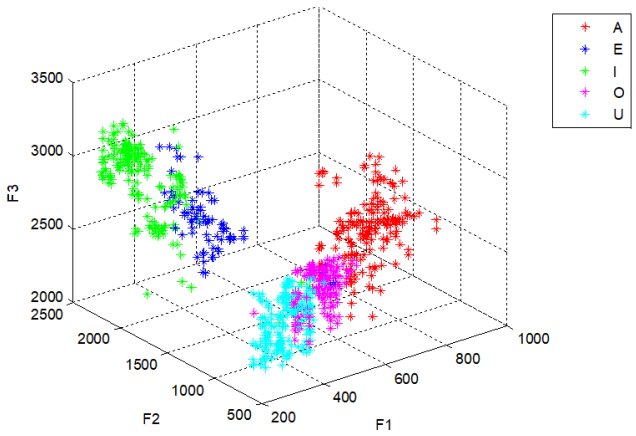


Fig.6. Distribution of the vowels in F_1 F_2 F_3 space.

VII. Neural Network for Vowel Recognition and Gain Results

After formant extraction final step was neural network design. For vowel recognition problem multilayer feed-forward neural network was chosen [9]. Supervised learning technique is performed because neural network has target values for input data. Supervised learning implies that neural network has output values for all input values. On these input/output pairs neural network is trained and tested [10].

Neural network has three inputs, one for each formant frequency. Neural network has five outputs, one for each class that represents one vowel. We decided to split our historical dataset that has 1080 formant frequencies (about 200 instances for first three formant frequencies for each vowel), so 90% of data was used for training and 10% was used for testing. For better algorithm performances k-fold cross validation was performed [5]. Entire dataset was split on 10 folds. For model metrics we used accuracy which was counted on testing dataset in each iteration during k-fold cross validation. Finally, average accuracy of the model was counted for all ten iterations.

Number of hidden layers and nodes was chosen by trial and error method. First, small number of hidden layers and nodes was chosen. This number was increased in order to increase the accuracy of the model, but we took care about overfitting and stopped with increasing when accuracy started to decrease [9]. We also tried network training with different optimization algorithms, activation functions and values of learning rate. Number of epochs was also changed.

The best result was achieved with neural network architecture that consists of six hidden layers with 100 nodes.

Optimization algorithm that was used is Adam [11] with sparse categorical cross entropy loss function, with learning rate one and with RELU activation function [9]. Number of epochs was 2000. With this neural network architecture average accuracy of the model on testing dataset was 96.3%.

When optimal neural network architecture was chosen, neural network was trained on entire historical dataset. Finally, neural network testing was performed on uttered vowels that did not were in original base of vowels. Three new male speakers uttered each vowel five times. For new vowels confusion matrix is shown in Table III.

TABLE III
CONFUSION MATRIX FOR NEW VOWELS

	A	E	I	O	U
A	15	0	0	0	0
E	0	12	3	0	0
I	0	0	15	0	0
O	0	0	0	15	0
U	0	0	0	0	15

New vowels were recognized with accuracy of 96%. We can see that all vowels except vowel “e” were recognized with 100% accuracy. Vowel “e” was three times recognized as vowel “i”. This error is not surprising because classes of vowels for neural network training were not separable, and there was overlapping between classes “e” and “i”. These results could be different for some other speakers and that is the reason why vowel recognition is very complex task.

VIII. CONCLUSION

Considering that historical dataset for neural network training was not big, gain results are very satisfying. It is shown that with first three formant frequencies and with adequate choice of neural network, vowels can be recognized with big accuracy.

First problem during vowel recognition system making was different uttering of vowels. Since this is a system that is independent of the speaker, the way in which vowels are uttered is very important for their recognition. For this reason, it is very difficult to create a single algorithm that classifies vowels, regardless of which speaker uttered the vowel. Second problem was overlapping between classes that represent vowels. This overlapping was unavoidable and it affected on model accuracy.

Dataset with formant frequencies for neural network training was not very big. Increasing of dataset with recorded vowels for neural network training would probably increase model accuracy. Also, more uttered vowels for neural network testing would give more reliable results about model accuracy. Despite the above limitations and problems encountered in designing a vowel recognition system, the designed system gives very good results.

Due to their properties, neural networks are nowadays very attractive for scientists and researchers when it comes to speech recognition [12]. In some future work other types of neural network can be applied for solving vowels recognition problem and comparative analysis can be made with gain results.

REFERENCES

[1] J. Li, L. Deng, R. Haeb-Umbach, Y. Gong, *Robust Automatic Speech recognition-A Bridge to Practical Application*, Oxford, UK: Academic Press, 2016.

[2] L. R. Rabiner, R. W. Schafer, *Digital Processing of Speech Signals*, Prentice-Hall, Inc., Englewood Cliffs, New Jersey 07632, 1978.

[3] L. Rabiner, B.-H. Juang, B. Yegnarayana,, *Fundamentals of Speech Recognition*, Pearson, India, 2010.

[4] S. Kay, *Modern Spectral Estimation: Theory & Application*, Prentice-Hall, New Jersey, 1988.

[5] C. T. Lin, C. S. G. Lee, *Neural Fuzzy Systems: A Neuro-Fuzzy Synergism to Intelligent Systems*, Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1996.

[6] J. Tebelskis, “Speech recognition using neural networks”, Ph.D. dissertation, Carnegie Mellon University Pittsburgh, 1995.

[7] S. K. Mitra, *Digital Signal Processing*, A *Computer Based Approach*, Wcb/McGraw-Hill, 4th ed., USA, 2011.

[8] P. D. Welch, “The use of fast Fourier transforms for the estimation of power spectra: A method based on time averaging over short modified periodograms,” *IEEE Transactions on Audio and Electroacoustics*, vol. 15, pp. 70-73, 1967.

[9] I. Goodfellow, Y. Bengio, A. Courville, *Deep Learning (Adaptive Computation and Machine Learning Series)*, MIT, USA, 2016.

[10] K. Fukunaga, *Introduction to Statistical Pattern Recognition*, Academic Press, Boston, 1980.

[11] D. P. Kingma, J.L. Ba, “Adam: A method for stochastic optimization,” *arXiv*: 1412.6980v9, 2014.

[12] A. Graves, A. Mohamed, A. G. E. Hinton, “Speech recognition with deep recurrent neural networks” *IEEE International Conference on Acoustics, Speech and Signal Processing*, Vancouver, BC, Canada 6645-6649, 2013.

Multi-resonant observer PLL with estimation of grid unbalances

Aleksandra Mitrović, Mirna N. Kapetina, Milan R. Rapaić

***Abstract*—In this paper, a novel discrete Phase Locked Loop (PLL) algorithm is introduced to obtain an optimal grid synchronization. One of the major problems in the grid synchronization process, which occurs due to grid imperfections, is higher order harmonics which are treated as disturbances in the electric power system. The proposed control design of PLL is implemented with a reduced order observer of periodic disturbances and with RST controller structure. The design is performed completely in the discrete time domain, and two cases were considered: the first case is with one higher order harmonic, and the second case is with two higher order harmonics in the system. The theoretical findings are substantiated by extensive simulation examples.**

Keywords: Grid synchronization, PLL algorithm, Reduced order observer, RST controller

I. Introduction

One of the important aspects of designing grid connected power electronics systems is the synchronization of the inverter output with the grid itself. Synchronization with the grid means matching the phase and the magnitude of the inverter output voltage with grid voltage signal and implicitly control of power factor, active and reactive power flow in the system. If the grid is unbalanced, side effects such as short circuits, failures and power outages can occur. Also, due to the grid imperfections, higher order harmonics appear in the electric power system. They are especially significant and represent unwanted spectral components of a distorted signal whose frequencies are equal to an integer product of the base frequency. These polluting harmonics are treated as disturbances in the system and consequently, it is necessary to neutralize their influence in order of optimal functioning of the electric power system.

Some modern solutions for efficient and reliable integration with grid of renewable energy sources are grid side converters (GSC). These power electronics devices enable the injecting or receiving of electric energy from the utility grid and they must be precisely synchronized with grid voltage [1,2]. There are several different algorithms for grid synchronization and one of the most commonly used of them is the Phase Locked Loop or PLL algorithm [3]. Among existing PLL techniques, the Synchronous Reference Frame (SRF-PLL) [4] is commonly utilized in power engineering applications. Also, there are other solutions such as Decoupled Double Synchronous Reference Frame PLL (DDSRF-PLL) [5] or Double Second-Order Generalized Integrator PLL (DSOGI-PLL) [6]. These algorithms perform excellent with negligible grid imperfections and they can accurately extract voltage synchronization signal when the grid is unbalanced. In situations when the grid voltage contains harmonics of higher order, large oscillations may appear in the synchronization signal and then Multi-SOGI method proposed in [7] could be used.

Achieving phase synchronization despite voltage sags and higher order harmonics and identifying amplitudes and phase angles of the polluting harmonics are essential for real-time control of power systems. An effective PLL scheme for efficient simultaneous identification and quantification of grid unbalance and higher harmonics content is proposed in [8]. However, the proposed method is described in continuous time domain and the reliability of a discretized system depends upon the approximation made to their continuous equations. Some methods, as the Forward Euler, the Backward Euler and the Tustin (Trapezoidal) numerical integration offer a good performance when used for continuous filter discretization, but those methods, could be inadequate under certain conditions, due to necessity of additional sample delays. Therefore, in this paper procedure for multi resonant PLL is completely implemented in the discrete domain without using numerical discretization method. The main reason is to obtain expressions that can be directly applied in the implementation, i.e. to avoid the discretization procedure that would otherwise be necessary.

The organization of this paper is such that in section 2 the grid voltage signal is characterized and vector notation is introduced. The basic aspects of the PLL algorithm as well as the new proposed modification of the algorithm applicable in the discrete domain are given in section 3. Finally, the results of the numerical simulation are presented in section 4.

II. Grid Characterization

A non-ideal 3-phase utility grid voltage in steady state can be represented as [3]

$$u_i = U^p \cos(\omega_0 t - k_i \frac{2\pi}{3} + \varphi^p) + U^n \cos(-\omega_0 t - k_i \frac{2\pi}{3} + \varphi^n) + \sum_h U_{ih} \cos(h\omega_0 t - k_i \frac{2\pi}{3} + \varphi_{ih}) \,, \quad (1)$$

where $k_{i\in\{a,b,c\}} = \{0, 1, -1\}$. Variables U^p, U^n are amplitudes and φ^p, φ^n are phases of the Fundamental Frequency

A. Mitrović(aleksandra.mitrovic@uns.ac.rs), M. N. Kapetina (mirna.kapetina@uns.ac.rs), M. R. Rapaić(rapaja@uns.ac.rs) University of Novi Sad, Faculty of Technical Sciences, Department of Computing and Control Engineering, Trg Dositeja Obradovića 6, 21000 Novi Sad, Serbia.

Positive Sequence (FFPS) and Fundamental Frequency Negative Sequence (FFPN), respectively [4,10]. Magnitude ω_0 is the fundamental frequency which value is typically $2\pi \cdot 50\frac{\text{rad}}{s}$ or $2\pi \cdot 60\frac{\text{rad}}{s}$. The third term in (1) represents higher order harmonics, i.e. U_{ih} is amplitude and φ_{ih} is phase of h-th harmonic for phase i. If the grid voltage is balanced, without asymmetrical voltage sags and higher order harmonics, it is clear that $U^n = 0$ and $U_{ih} = 0$ [8].

In order to simplify the notation, (1) can be rewritten in the phasor form

$$\underline{U} = \underline{U}_1 + \sum_h \underline{U}_h \ , \tag{2}$$

where $\underline{U}$ is the phasor corresponding to signal u. Phasors in (2), $\underline{U}_1$ and $\underline{U}_h$, can be represented by their positive and negative sequence components

$$\begin{aligned} \underline{U}_1 &= U^p e^{j\omega_0 t} + U^n e^{-j(\omega_0 t+\varphi_n)} \ , \\ \underline{U}_h &= U_h^p e^{j(h\omega_0 t+\varphi_h^p)} + U_h^n e^{-j(h\omega_0 t+\varphi_h^n)} \ , \end{aligned} \tag{3}$$

where it was assumed that FFPS voltage component is the referent one with zero phase ($\varphi^p = 0$). It is common for 3-phase phenomena analysis to use rotating reference frame (dq) instead of staionary (abc), so Park transformation [9] is applied on (3) and one obtains

$$\begin{aligned} \underline{U}^{dq} = \underline{U} e^{-j\omega_0 t} = U^p + U^n e^{-j(2\omega_0 t+\varphi_n)} + \\ + \sum_h U_h^p e^{\left[j(h-1)\omega_0 t+\varphi_h^p\right]} + U_h^n e^{\left[-j(h+1)\omega_0 t+\varphi_h^n\right]} \ . \end{aligned} \tag{4}$$

We can write $\underline{U}^{dq} = u_d + ju_q$ so (4) can be decoupled to d and q components

$$\begin{aligned} u_d &= U^p + U^n \cos(2\omega_0 t + \varphi_n) \\ &+ \sum_h U_h^p \cos\left[(h-1)\omega_0 t + \varphi_h^p\right] + U_h^n \cos\left[(h+1)\omega_0 t + \varphi_h^n\right], \\ u_q &= -U^n \sin(2\omega_0 t + \varphi_n) \\ &+ \sum_h U_h^p \sin\left[(h-1)\omega_0 t + \varphi_h^p\right] - U_h^n \sin\left[(h+1)\omega_0 t + \varphi_h^n\right] \end{aligned} \tag{5}$$

Finally, from (5) can be concluded if the grid voltage is balanced and without polluting harmonics, d and q components in the steady-state appears as DC values, $u_d = U^p$ and $u_q = 0$.

III. The proposed PLL algorithm

Ideal PLL structure is resistant to disturbances and asymmetries and quickly and accurately determines the phase angle of the grid voltage. PLL control loop is presented in Fig. 1 where we have phase comparator that determines the difference between the phase angle of the input quantity and the estimated angle of the output quantity. This signal is fed to the controller and the output signal from the controller excites the signal generator which generates an estimated value of the grid phase angle that follows the phase angle of the input quantity. In the digital form of PLL algorithm, the most common controller structure is the PI controller [10].

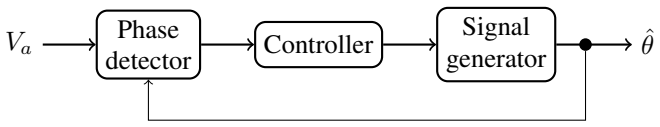


Figure 1: *Generalized PLL control loop*

Nowdays there are various modifications of the PLL algorithm caused by the development of different implementation techniques and different reference coordinate systems in which the estimation of the grid phase angle is performed. One of the most commonly used modifications is Synchronous Frame PLL algorithm or SF-PLL algorithm [11].

SF-PLL algorithm is adapted synchronization algorithm to the specifics of 3-phase systems, which conventional control loop is presented in Fig. 2. By applying this variation of the algorithm, the estimation is performed by setting one component of the grid voltage to zero value and in that way the estimated angle is connected to the vector representation of the grid voltage.

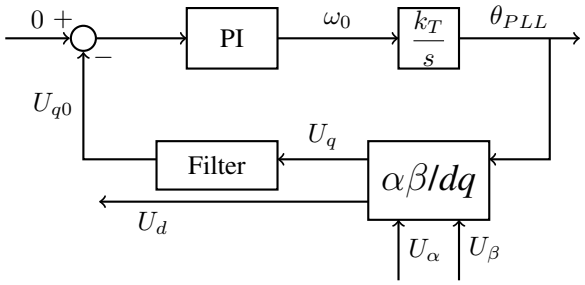


Figure 2: *PLL control loop*

The plant in PLL algorithm can be modelled by transfer function $G_p(s) = \frac{k_T}{s}$, where k_T is linearization gain equal to the grid voltage amplitude, $k_T = U$. Filter in Fig. 2 is not needed if utility voltage is balanced and without higher order harmonics, but without it, PI controller can't cope with oscillations induced by imperfection of grid voltage [8]. Accordingly, the main purpose of all advanced PLL algorithms is to neutralize (filter) oscillatory components.

In this paper, SF-PLL algorithm with multi-resonant disturbance observer was implemented in discrete time domain, as illustrated in Fig. 3. Input variables in control loop shown in Fig. 3 are measured phase voltages u_a, u_b and u_c. After applying Park transformation, 3-phase voltages u_a, u_b, u_c are transformed into u_d and u_q in a synchronously rotating 2-phase system. For these transformed signals, observers were designed to estimate value of higher order harmonics. Since observers used for d and q axis are identical, only the q-axis observer will be described in sequel. Finally, obtained estimated grid frequency and angle are denote $\hat{\omega}$ and $\hat{\theta}$.

A. Observer design

Estimation is the process of evaluating the value of an immeasurable quantity on the basis of available data [2]. In the field of control, it is of interest to estimate values of disturbance signals and states of process. The algorithms

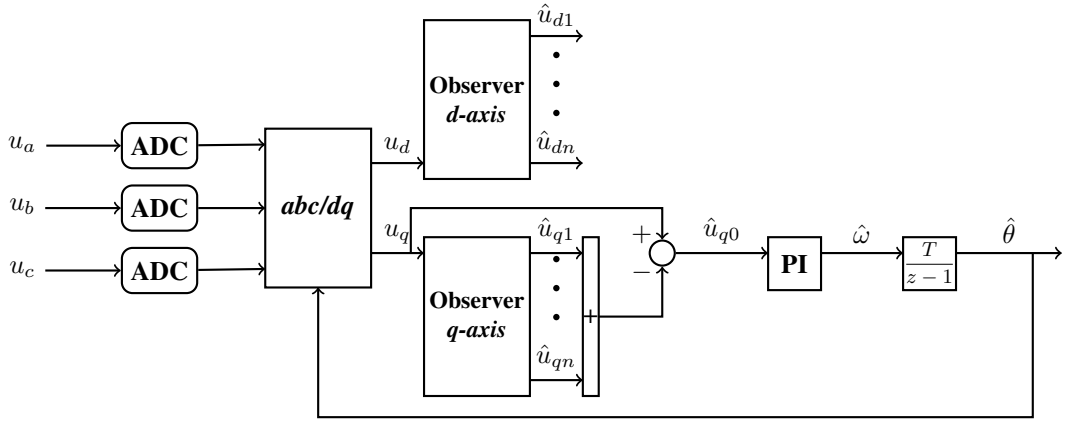


Figure 3: *PLL algorithm with multi-resonant observer (blocks **ADC** represent analog to digital converters)*

used to perform this estimation are observers and in this paper, multi-resonant observer was designed in discrete time domain.

Discrete-time observer

After applying Park transformation and considering the fact that a superposition of any two harmonic signals of equal frequencies will give another harmonic signal of the same frequency, (5) can be written as

$$\begin{aligned} u_q &= u_{q0} + \sum_{h=1}^{n} u_{qh} \ , \\ u_{qh} &= U_{qhm} \sin(\omega_h k T - \varphi_{qh}) = U_{qhm} \sin(\theta_h k - \varphi_{qh}) \ , \end{aligned} \tag{6}$$

where U_{qhm} is total amplitude and φ_{qh} is the total phase angle of h-th harmonic in dq reference frame, $\theta_h = \omega_h T$ is discrete (digital) frequency, T is sample time and k is ordinal number of samples in the discrete time domain. Higher order harmonics are treated as disturbances in system and reduced-order observer was designed to estimate values of additional harmonics from measured signal. Estimation is performed by imitating a process that generates magnitude of interest. Hence, a linear system was constructed with impulse response equal to the (6) [12]

$$u_{qh}(k+2) - 2\cos(\theta_h) u_{qh}(k+1) + u_{qh}(k) = 0 \ . \tag{7}$$

In order to develop an reduced-order observer, a selection of variable states was made [13]

$$\begin{aligned} x_1(k) &= u_q(k) \ , \\ x_{2h}(k) &= u_{qh}(k) \ , \\ x_{2h+1}(k) &= u_{qh}(k+1) \ . \end{aligned} \tag{8}$$

In shifted discrete time domain variables are

$$\begin{aligned} x_1(k+1) &= x_1(k) - \sum_{h=1}^{n} x_{2h}(k) + \sum_{h=1}^{n} x_{2h+1}(k) \ , \\ x_{2h}(k+1) &= x_{2h+1}(k) \ , \\ x_{2h+1}(k+1) &= -x_{2h}(k) + 2\cos(\theta_h) x_{2h+1}(k) \ , \end{aligned} \tag{9}$$

or in matrix form

$$x(k+1) = \begin{bmatrix} 1 & -1 & 1 & -1 & 1 & \dots \\ 0 & 0 & 1 & 0 & 0 & \dots \\ 0 & -1 & 2\cos(\theta_1) & 0 & 0 & \dots \\ 0 & 0 & 0 & 0 & 1 & \dots \\ 0 & 0 & 0 & -1 & 2\cos(\theta_2) & \dots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix} x(k) \ . \tag{10}$$

After selecting state vector $x(k)$, it is convenient to split state vector into to parts: a directly measurable variables $x_1(k)$ and the remaining states $x_{2:}(k) = \begin{bmatrix} x_2(k) & x_3(k) & \dots & x_{2h+1}(k) \end{bmatrix}^T$ [13]. Subsequently, (10) can be rewritten as

$$\begin{aligned} x_1(k+1) &= A_{11} x_1(k) + A_{12} x_2(k) \ , \\ x_{2:}(k+1) &= A_{21} x_1(k) + A_{22} x_2(k) \ , \end{aligned} \tag{11}$$

where

$$A_{11} = 1 \ ,$$

$$A_{12} = \begin{bmatrix} -1 & 1 & -1 & \dots & 1 \end{bmatrix}_{2n} \ ,$$

$$A_{21} = \begin{bmatrix} 0 & 0 & 0 & \dots \end{bmatrix}_{2n}^{T} \ ,$$

$$A_{22} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & \dots \\ -1 & 2\cos(\theta_1) & 0 & 0 & 0 & 0 & \dots \\ 0 & 0 & 0 & 1 & 0 & 0 & \dots \\ 0 & 0 & -1 & 2\cos(\theta_2) & 0 & 0 & \dots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix}_{2n \times 2n} \ .$$

Mathematical model of reduce-order observer is

$$\begin{aligned} z(k+1) &= A_o z(k) + G u_q(k) \ , \\ \hat{x}_{2:}(k) &= z(k) + L u_q(k) \ . \end{aligned} \tag{12}$$

where z is $2n \times 1$ vector of internal observers states and A_o, G and L are constant parameter matrices of dimension $2n \times 2$, $2n \times 1$ and $2n \times 1$, respectively. Vector L is observer gain matrix [13]. Parameter matrices A_o and G are determined according to

$$\begin{aligned} A_o &= A_{22} - L A_{12} \ , \\ G &= A_{21} - L A_{11} + A_o L \ . \end{aligned} \tag{13}$$

The vector L is selected in such a way to place eigenvalues of A_o at desired locations [8]. Observer characteristic polynomial is calculated on the basis of desired poles

$$f_o(z) = z^{2n} + p_{2n-1}z^{2n-1} + \cdots + p_1 z + p_0 \ . \tag{14}$$

Accordingly, elements of the observer gain vector L can be calculated from

$$f_o(z) = det(zI - A_{22} + LA_{12}) \ . \tag{15}$$

After the observer gain vector L is determined, matrices A_o and G can be calculated from (13). Finally, DC component or utility signal component for synchronization can be reconstructed by subtracting estimated harmonic signals from the measured voltage

$$\hat{u}_{q0} = x_1 - \sum_{h=1}^{n} \hat{x}_{2h} \ , \tag{16}$$

where magnitude $\hat{u}_{q0}$ is used as the input of PLL controller [8].

B. Controller design

Controller was implemented in discrete time domain by means of pole-placement procedure and RST synthesis [14]. In both time domains, desired closed loop characteristic polynomial, i.e. poles of the system in closed loop are selected. These poles completely determine stability and behavior of the system in closed loop. Finally, parameters of observer and controller are selected so to ensure that the actual characteristic polynomial of the closed loop system matches the desired one.

Discrete-time controller

Magnitude $\hat{u}_{q0}$, calculated from (16), is input signal of PLL algorithm, so observer transfer function relating u_q and u_{q0} is [8]

$$G_o(z) = \frac{U_{q0}(z)}{U_q(z)} = k_o \frac{\prod_h (z^2 - 2z\cos(\theta_h) + 1)}{f_o(z)} \ , \tag{17}$$

where z is complex variable in discrete time domain, $f_o(z)$ is the characteristic polynomial of the observer and k_o is its gain. Observer transfer function must have zeros at frequencies ω_h and additional zeros are not allowed, since this can make transfer function $G_o(z)$ improper (degree of the polynomial in the numerator would be greater than the degree of the polynomial in the denominator). Value of the gain k_o can be calculated from the fact that static gain of (17) must be 1

$$\lim_{z\to 1} G_o(z) = 1 \Rightarrow k_o = \frac{f_o(1)}{\prod_h (2 - 2\cos(\theta_h))} \ . \tag{18}$$

RST controllers are a modern management solution in whose structure there are two degrees of freedom of movement and three polynomials - R, S and T. Namely, classic PID controllers have one degree of freedom of movement and can be effective in monitoring setpoints, but not so good in eliminating disturbances. Accordingly, it is desirable to use controllers that have a component of direct control and feedback control such as RST controllers. Typical structure of RST controller is $R(z)U(z) = -S(z)Y(z) + T(z)R_{ref}(z)$, where $R(z)$, $S(z)$ and $T(z)$ are polynomials in RST controller structure, $R_{ref}(z)$ is reference signal and $Y(z)$ is output signal.

As denote on Fig. 3 input signal to the controller is q component of the grid voltage and the output is the estimated grid frequency $\hat{\omega}$. Due to the reference signal is equal to zero, as shown in Fig. 2, which implies that $T(z) = 0$, controller form is

$$R(z)U(z) = -S(z)Y(z) \Rightarrow R(z)U_q(z) = -S(z)\hat{\Omega}(z) \ , \tag{19}$$

where $R(z)$ is denominator, $S(z)$ is numerator of the controller transfer function, $\hat{\Omega}(z)$ and $U_q(z)$ are $\mathcal{Z}$ transforms of the estimated frequency and q component of the grid voltages, respectively. Observer is part of the controller so structure of polynomials $R(z)$ and $S(z)$ should be [8]

$$R(z) = (z-1)f_o(z)R_1(z) \ ,$$
$$S(z) = k_o \prod_h (z^2 - 2zcos(\theta_h))S_1(z) \ ,$$

where $R_1(z)$ and $S_1(z)$ contain remaining factors to be determined in the sequel. Additional integrator is paramount for the ability of the control loop to follow grid angle signal, which has ramp-like behavior [8]. As two integrators are required to follow ramp signal without steady state error, second integrator is in the plant transfer function $G_p(z) = \frac{k_T}{z-1}$.

Characteristic polynomial of the closed loop system is

$$f_c(z) = (z-1)^2 f_o(z)R_1(z) + \\ + k_o k_T S_1(z) \prod_h (z^2 - 2z\cos(\theta_h) + 1) \ . \tag{20}$$

Unknown parameters in (20) can be calculated from the fact that actual characteristic polynomial of the closed loop system must be equal the desired one. This means that number of variables must be equal to the degree of the characteristic polynomial and solution of minimal order is

$$R_1(z) = 1, \ S_1(z) = k_p(z + \sigma) \ ,$$

where k_p and σ are real constants. In fact, kp is the proportional gain and $\sigma = \frac{1}{T_i}$ is the reciprocal of the integral time constant. Finally, obtained characteristic polynomial of the closed loop system is

$$f_c(z) = (z-1)^2 f_o(z) + \\ + k_o k_T k_p (z + \sigma) \prod_h (z^2 - 2z\cos(\theta_h) + 1) \ . \tag{21}$$

In section IV, all previous consideration will be shown in two different simulations: first is case with one harmonic in system and other is case with two harmonics in system.

IV. Simulation results

A. The case with one harmonic

The proposed PLL algorithm has been implemented and tested through simulation in case when grid is symmetrical until $t = 0.1$s, when fifth harmonic amplitude 0.2 [p.u.] appears. In that case there is one polluting harmonic in system ($h = 1$) and characteristic polynomial of closed loop system given by (21), can be written as

$$f_c(z) = (z-1)^2(z^2 + p_1 z + p_0) + \\ + k_o k_T k_p (z+\sigma)(z^2 - 2z\cos(\theta_1) + 1) \ . \tag{22}$$

Desired characteristic polynomial is formed based on the poles a_i that determine stability and performance of the system. Choosing a poles we make a compromise between response speed, disturbance rejection properties, sensitivity to measurement noise, robustness and other characteristics. In this paper, a pair of dominant poles have natural frequency equal to ω_0 ($\omega_0 = 2\pi \cdot 50 \frac{\text{rad}}{s}$) and high damping ($\xi \geq 0.7$), while the remaining poles are located deeper within the unit circle. Poles are

$$a_{1,2} = e^{-\frac{\omega_0 \xi}{\sqrt{1-\xi^2}} T \pm j\omega_0 T} \ , \\ a_3 = e^{-2\omega_0 T} \ , \\ a_4 = e^{-4\omega_0 T} \ .$$

where T is sampling time chosen to satisfy Nyquist theorem. In this case, polluting harmonic is at frequency ω_1, so sampling frequency is chosen so that it is equal $\omega_s = 5\omega_1$, where $T = \frac{2\pi}{\omega_s}$ and $\theta_1 = \omega_1 T$. Desired characteristic polynomial is

$$f_c = (z-a_1)(z-a_2)(z-a_3)(z-a_4) \ . \tag{23}$$

Higher order harmonic of interest is fifth harmonic which is seen as harmonic at the frequency $\omega_1 = 4 \cdot \omega_0$ in dq reference frame.

By equating desired characteristic polynomial (23) with characteristic polynomial of closed loop system (22), observer gain matrix and controller parameters are obtained:

$$L = \begin{bmatrix} 0.3982 & 0.5676 \end{bmatrix}^T \ , \\ k_p = 0.3866 \ , \sigma = -0.8524 \ .$$

Phase voltages are shown in Fig. 4, while on Fig. 5 is shown estimated frequency of grid voltage and on Fig. 6 is shown estimation error which represents the difference between actual and estimated value of the phase angle.

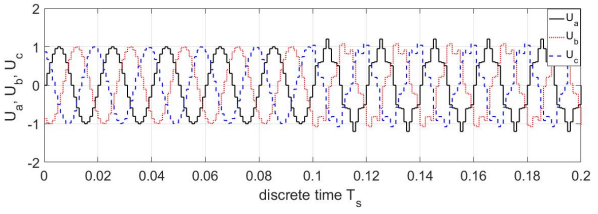


Figure 4: Phase voltages

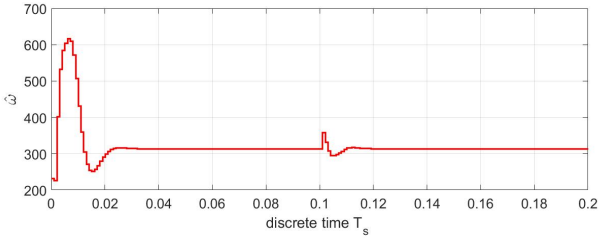


Figure 5: Estimated grid frequency

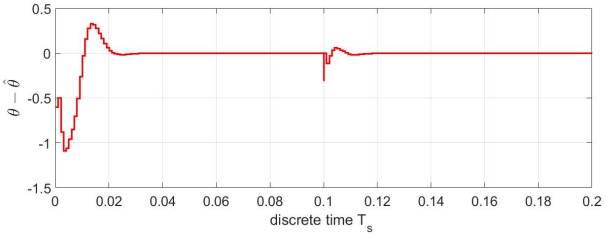


Figure 6: Phase estimation error

B. The case with two harmonics

The proposed PLL algorithm has been implemented and tested through simulation in case when grid is symmetrical until $t = 0.1$s, when fifth harmonic amplitude 0.2 [p.u.] appears and then in $t = 0.2$s seventh harmonic amplitude 0.5 [p.u.] appears. In that case there are two polluting harmonics in system ($h = 2$) and characteristic polynomial of closed loop system given by (21), can be written as

$$f_c(z) = (z-1)^2(z^4 + p_3 z^3 + p_2 z^2 + p_1 z + p_0) + \\ + k_o k_T k_p (z+\sigma)(z^2 - 2z\cos(\theta_1) + 1)(z^2 - 2z\cos(\theta_2) + 1) \ . \tag{24}$$

As in the first case, desired characteristic polynomial is formed based on the poles a_i that determine stability and performances of the system. A pair od dominant poles have natural frequency $\omega_0 = 2\pi \cdot 50 \frac{\text{rad}}{s}$ and damping ratio $\xi = 0.7$, while the remaining poles are located deeper within the unit circle. Poles are

$$a_{1,2} = e^{-\frac{\omega_0 \xi}{\sqrt{1-\xi^2}} T \pm j\omega_0 T} \ , \\ a_{3,4} = e^{-2\omega_0 T} \ , \\ a_{5,6} = e^{-4\omega_0 T} \ .$$

where T is sampling time chosen to satisfy Nyquist theorem. In this case, polluting harmonics are at frequencies ω_1 and ω_2 ($\omega_1 < \omega_2$), so sampling frequency is chosen so that it is equal $\omega_s = 5\omega_2$, where $T = \frac{2\pi}{\omega_s}$, $\theta_1 = \omega_1 T$ and $\theta_2 = \omega_2 T$. Desired characteristic polynomial is

$$f_c = (z-a_1)(z-a_2)(z-a_3)(z-a_4)(z-a_5)(z-a_6) \ . \tag{25}$$

Higher order harmonics of interest are fifth and seventh harmonics which are seen as harmonics at the frequencies $\omega_1 = 4 \cdot \omega_0$ and $\omega_2 = 6 \cdot \omega_0$ in dq reference frame, respectively.

By equating desired characteristic polynomial (25) with characteristic polynomial of closed loop system (24), observer gain matrix and controller parameters are obtained:

$$L = \begin{bmatrix} 0.4898 & 0.8004 & -0.5634 & 0.2361 \end{bmatrix}^T ,$$
$$k_p = 551.391 \, , \sigma = -0.865 \, .$$

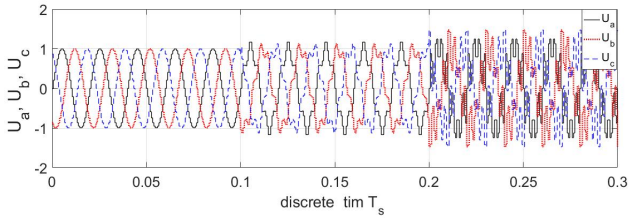


Figure 7: Phase voltages

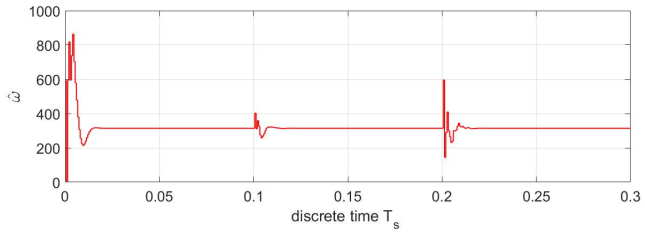


Figure 8: Estimated grid frequency

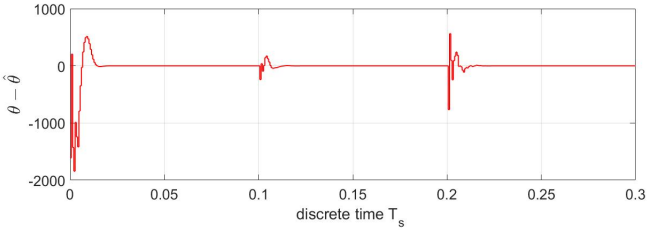


Figure 9: Phase estimation error

Phase voltages are shown in Fig. 7, while on Fig. 8 is shown estimated frequency of grid voltage and on Fig. 9 is shown estimation error which represents the difference between actual and estimated value of the phase angle.

V. Conclusion

Most modern devices are connected to the grid via power electronics devices and it is very important to achieve synchronization between them. Magnitudes of interest for grid synchronization are frequency and phase angle of grid voltage. In this paper, we proposed PLL algorithm for grid synchronization based on multi-resonant observer. Algorithm is implemented with observer which estimate unknown values of polluting higher order harmonics which occur due to the grid imperfection and we treat them as disturbances in system. The proposed PLL algorithm designing process is established as a simple four steps algorithm:

1) Choose the frequencies ω_h of higher order harmonics of interest.
2) Specify the desired closed loop polynomial $f_c(d)$.
3) Compute unknown parameters of controller (k_p, σ) and observer polynomial (p_0, p_1, $\cdots$, p_{2n-1}).
4) Compute observer matrix gain L.

Algorithm control loop is implemented in discrete time domain. After algorithm designing, simulations were done in case when fifth and seventh harmonics appear as disturbances in the system. The results demonstrated the efficiency of the proposed PLL algorithm which is capable of achieving phase synchronization despite voltage unbalances and higher order harmonics.

VI. Acknowledgment

This research (paper) has been supported by the Ministry of Education, Science and Technological Development through the project no. 451-03-68/2020-14/200156: "Innovative scientific and artistic research from the FTN (activity) domain" (M.N.K., M.R.R., A.M.).

References

[1] Benysek G, Kaźmierkowski MP, Popczyk J, Strzelecki R. *Power electronic systems as crucial part of a smart grid infrastructure – a survey*. Bull Pol Ac.: Tech 2011;59(4,).

[2] Gude, S. , Chu, C.-C. *Three-Phase PLLs by Using Frequency Adaptive Multiple Delayed Signal Cancellation Pre-filters under Adverse Grid Conditions*. IEEE Trans. Ind. Applications (Early Access), http://doi.org/ 10.1109/TIA.2018.2823263.

[3] Longwei Zhang, Shuhai Quan, Liang Huang, Kang Yang, Ying Xiong, Jin Quan. *Research on frequency conversion PLL for three-phase unbalanced and harmonic power system*. 32nd Youth Academic Annual Conference of Chinese Association of Automation (YAC), 2017.

[4] Fehmi Sevilmiş, Hulusi Karaca. *Performance analysis of SRF-PLL and DDSRF-PLL algorithms for grid interactive inverters*. International Advanced Researches and Engineering Journal, Vol. 03, pages 116-122, August 2019.

[5] Awad H, Svensson J, Bollen M. *Tuning software phase-locked loop for series-connected converters*. IEEE Trans. Power Del. Jan. 2005;20(1):300–8.

[6] Rodríguez P, Teodorescu R, Candela I, Timbus AV, Liserre M, Blaabjerg F, et al. *Positive-sequence voltage detector for grid synchronization of power converters under faulty grid conditions*. Proc IEEE Power Electron Specialists Conf 2006:1–7.

[7] Rodríguez P, Luna A, Etxeberria I, Hermoso JR, Teodorescu R. *Multiple second order generalized integrators for harmonic synchronization of power converters.*. Proc IEEE Energy Convers Congr Expo 2009:2239–46.

[8] M. Vekić, M. R. Rapaić, T. B. Šekara, S. Grabić, E. Adžić. *Multi-Resonant observer PLL with real-time estimation of grid unbalances*. International Journal of Electrical Power & Energy Systems, Vol. 108, pages 52-60, June 2019.

[9] Yazdani A, Iravani R. *Voltage-sourced converters in power systems*. Hoboken, New Yersey:IEEE-Wiley, 2010.

[10] Bane Popadić. *Napredno upravljanje pretvaračem povezanim na mrežu pri nesimetričnim naponskim prilikama u elektroenergetskom sistemu*. PhD thesis, University of Novi Sad, Faculty of Technical Sciences, 2018.

[11] Xiao-Qiang Guo, Wei-Yang Wu, He-Rong Gu. *Phase locked loop and synhronization methods for grid-interfaced converters: a review*. Yanshan University, 2011.

[12] Karl J. Astrom, Bjorn Wittenmark. *Computer-Controlled Systems - Theory and Design*. Dover Publications Inc. 3rd edition, New York, USA, 2011.

[13] Milan R. Rapaić, Zoran D. Jeličić. *Projektovanje linearnih regulatora i estimatora u prostoru stanja*. Univerzitet u Novom Sadu, Fakultet tehničkih nauka, Novi Sad, 2014.

[14] Atröm KJ, Murray RM. *Feedback systems: an introduction for scientists and engineers*. Princeton University Press; 2008

Decision Support System for Traffic Jams by using Artificial Intelligence

Luka Bjelica, Svetozar Vulin, Anja Buljević

***Abstract*—This paper presents one possible decision support system as partial solution of traffic jam problem using autonomous vehicles. Proposed solution is based on deep learning algorithms and artificial neural network (ANN) models which task is to make a decision whether to move the car to a side lane, or stay in the current lane. The optimality criteria is maximization of the elapsed distance in traffic by training ANN, using a supervised learning paradigm from labeled data, to control the car. Gradient descent algorithm is used for the network's parameters estimation. Verification, testing and simulation application is also presented in this paper.**

Keywords: self-driving, artificial neural networks, deep learning, gradient descent, traffic jam.

I. Introduction

Traffic congestion has become one of the biggest modern life problems. Time spent in traffic jams is irreversibly wasted. Moreover, some researches have also shown that traffic congestion represents negatively impacts the Earth's climate, leading to global warming [1]. One solution to this problem is to minimize the time spent in traffic using autonomous vehicles. This solution not only reduces air pollution and global warming rate, but also saves time for each commuter. Autonomous cars are vehicles which are driven by digital technologies without any, or little, human intervention. They are capable of driving and navigating themselves on the roads by sensing the environmental impacts. They are designed to occupy less space on the road in order to avoid traffic jams and reduce the likelihood of accidents. Autonomous cars are one of the biggest challenges in industry nowadays, so various solutions solving the automation of control of cars on the road have been widely studied in the literature [2], [3].

This paper concerns the automation of the car movement in a highway traffic jam, based on deep learning algorithms, and ANN models, like Multilayer Perceptron (MLP) [4], in order to make decision whether to move the car to a side lane, or stay in the current lane. The idea is to maximize the average speed of the car or to maximize the distance that car would elapse in certain amount of time. The car is controlled by trained ANN. Gradient descent algorithm is used for the network's parameters estimation. Simulation application could be found in [5].

L. Bjelica (bjelicaluka@uns.ac.rs), S. Vulin (svetozar.vulin@uns.ac.rs), A. Buljević (anjabuljevic@uns.ac.rs), University of Novi Sad, Faculty of Technical Sciences, Department of Computing and Control Engineering, Trg Dositeja Obradovića 6, 21000 Novi Sad, Serbia.

After Introduction, the paper is organized in the following manner: description of self-driving component, model of the environment, data collecting procedure and the agent controlling the car are explained in Section II, explanation of the custom deep learning library can be found in Section III, technical details about the implementation are introduced in Section IV, simulation results are shown in Section V and the concluding remarks are given in the final Section VI.

II. Self-Driving Component

As previously mentioned, the topic of this paper is the automation of car movement through a dense highway traffic environment. The real-world environment is represented in a computer simulation and due to its complexity, only main dynamic characteristics are emphasised. For the need of this paper, it is expected that the vehicle already has automatic acceleration and braking systems implemented so the emphasis is on the component responsible for highway traffic jams. That component is called self-driving component. The purpose of this component is to make sure that the agent gets out of highway traffic jams as fast as possible (maximum distance optimization problem). It is important to know that self-driving component does not ensure that the vehicle is completely autonomous, thus it only provides a solution to dense traffic.

Environment (2D traffic simulation)

The model of the environment is a simplified representation of the real world (Fig. 1). It is represented in two dimensional (2D) space. The environment consists of N traffic lanes, passing cars and the vehicle that is being controlled (agent).

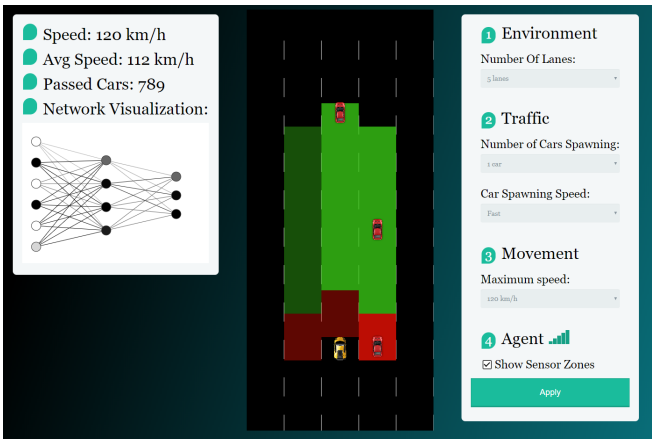


Fig. 1. Environment screenshot

Cars can be created in any of N lanes in random order and are moving in constant speed relative to each other. The agent has three sensors: front, left-side and right-side sensors, each consisting of a red and green zone. The green zone provides a distance between the car controlled by the agent and the nearest facing car in that lane. The red zone, or the safety zone, activates the braking system that manages to stop the controlled car when the car in front is detected and prohibits sidetracking when the car is detected on either left or right side. If the front sensor has not detected any cars in the red zone, the controlled car automatically accelerates until it reaches the maximum speed. This ensures that the car has a built-in safety system and that the agent has the ability to accelerate automatically when there are no cars in front of it. The environment implementation details are explained in the Section IV.

Data (collecting and labeling)

Data is collected and labeled while the user is controlling the car. Each time the user makes a decision (turns left, right or continues to move forward) a snapshot of each sensor data is collected. A snapshot is a list of data containing the following parameters:

- Left-side sensor green zone distance
- Left-side sensor red zone active status
- Front sensor green zone distance
- Front sensor red zone active status
- Right-side sensor green zone distance
- Right-side sensor red zone active status.

The label represents the user's decision (-1 for left, 0 for forward and 1 for right). After the user has provided enough training samples (the number of training samples is 28), the dataset is created which can then be used for training the neural network model.

Agent Controlling the Car

On a fixed time interval the agent queries the trained network with the current state of sensors. Agent expects to get a response containing the prediction about the action it should make. Based on the network's decision, the agent parses the response and either moves the car to a side lane, or stays in the current lane. If the car gets stuck because the network is continually making bad decisions, the user can correct the decision of a network. The corrections are saved and the user can put them in the dataset and retrain the model.

III. Deep Learning Library

For need of this paper, a deep learning library is implemented from scratch. Its model is based on the most popular deep learning framework Keras [6]. The library consists of a Model that represents the main component, Neural Network that acts as a container of parameters (weights and biases), Initializers, Operators, Losses, Optimizers, and Regularizers. The model is implemented as a Feed-Forward MLP [4], which means that each node (neuron) from every layer is connected to each node in the previous and following layer. It supports only the Supervised Learning paradigm [4], in which the desired outputs are known and the model is trained to predict future outcomes (output) depending on given (input) data.

Initializers

Initializers provide the initial values for the model parameters at the start of training. Initialization plays an important role in training deep neural networks, because bad parameter initialization can lead to slow or no convergence. Parameters of the network are initialized as small random weights drawn from the normal distribution [7].

Operators

Operators are the basic building blocks of any neural network. They are vector-valued functions that transform the data. Some commonly used operators are:

- *layers* (linear, convolution, and pooling)
- *activation functions* (Rectified Linear Unit (ReLU), Sigmoid, SoftMax, Tanh and etc.)

The only type of operators that are implemented for the need of this paper are activation functions. They are used to normalize the output of each neuron in the desired range.

Losses

Losses are differentiable mathematical expressions given in closed-form that are used as surrogates for the optimization objective of the problem at hand. Loss functions provide feedback on the training progression. They map a vector of values to a number that represents the quality of the network at a certain moment of training.

The cost or loss function has an important job in that it must faithfully distill all aspects of the model down into a single number in such a way that improvements in that number are a sign of a better model. [8]

Regularizers

Regularizers provide the necessary control mechanism to avoid overfitting and promote generalization. L2 weight regularization [9] is implemented that makes the weights sparser and uniform, respectively.

Optimizers

Optimizers provide the iterative update of model parameters with respect to the optimization objective. In this context, the optimization objective is to minimize the loss function that depends on the network's parameters. The parameters are considered optimal when the loss function reaches its global minimum. Loss function determines the error between the expected value and output value. Gradient Descent optimizer [8], [10] is used for fitting the model. Gradient Descent is an iterative optimization algorithm used for finding the minimum of a differentiable function. With a goal of minimizing the loss function, this algorithm takes steps of the steepest descent which is calculated by the negative value of the gradient of the function. Iteratively moving to the minimum, the size of each step is determined

by the learning rate. For network training using Gradient Descent [4], the following steps are taken:

1) **Creating a mini-batch**
   In this step, the portion of data is taken from the dataset, which is used in the present iteration. The size of the portion is equal to the batch size option that the user provides
2) **Forward pass and Calculating Loss**
   The feed-forward function is used to calculate the output of each layer. Outputs are stored in a list and are later used for calculating the gradients. Next, the value of loss function is calculated based on the output of the network.
3) **Backward pass**
   In this step, the error is being calculated and passed backward through the network. After the backward pass, deltas (the portion of error) have been calculated for each layer. It is important to apply the derivative of activation function to each delta in order to deactivate it.
4) **Backpropagation**
   This is the main step in which the gradients, based on small changes in a gradient direction, are calculated and the outputs of the layer from the feed-forward pass. In simple terms, after each *forward pass* through the network, error is calculated and *backward pass* is being performed while adjusting the network's **parameters** (*weights and biases*).
   The goal is to adjust each parameter in proportion to how much it contributes to the overall error.
   Partial derivative, of the loss function, concerning each parameter, represents the value of gradient.
   To calculate derivatives with respect to any nested variable in the equation ($\frac{\partial L}{\partial w_j}$ or $\frac{\partial L}{\partial b}$) we use the method called *chain rule* [7]. The *chain rule* shows that for the given:
   a) *preactivation* z of input x and bias b with respect to weight matrix transposed w^T is:
   $$z = w^T x + b$$
   b) value of the *preactivation* passed through *activation function* σ:
   $$\hat{y} = \sigma(z)$$
   c) *loss function (in this case cross entropy)* determines the error between the target value y and the output value of the network $\hat{y}$:
   $$L(y, \hat{y}) = -y \log(\hat{y}) - (1 - y) \log(1 - \hat{y}),$$
   the derivatives of the *loss function* with respect to w and b are
   $$\frac{\partial L}{\partial w_j} = \frac{\partial L}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial z} \frac{\partial z}{\partial w_j}$$
   $$\frac{\partial L}{\partial b} = \frac{\partial L}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial z} \frac{\partial z}{\partial b}.$$
   After the partial derivatives of nested equations have been calculated and substituted into the chain rule the obtained result are following
   $$\begin{aligned} \frac{\partial L}{\partial w_j} &= \frac{\partial L}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial z} \frac{\partial z}{\partial w_j} \\ &= \frac{\hat{y} - y}{\hat{y}(1 - \hat{y})} \hat{y}(1 - \hat{y}) w_j \\ &= (\hat{y} - y) w_j \end{aligned}$$
   $$\begin{aligned} \frac{\partial L}{\partial b} &= \frac{\partial L}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial z} \frac{\partial z}{\partial b} \\ &= \frac{\hat{y} - y}{\hat{y}(1 - \hat{y})} \hat{y}(1 - \hat{y}) \\ &= (\hat{y} - y). \end{aligned}$$
   In vectorized form with m training examples (when using Batch or Mini-Batch GD) the following equations are obtained
   $$\frac{\partial L}{\partial w} = \frac{1}{m} X (\hat{y} - y)^T$$
   $$\frac{\partial L}{\partial b} = \frac{1}{m} \sum_{i=1}^{m} (\hat{y}^{(i)} - y^{(i)}).$$
   After the gradients have been calculated, the parameters are ready to be adjusted (updated) accordingly
   $$w^{(i+1)} = w^{(i)} - \gamma \frac{\partial L}{\partial w}$$
   $$b^{(i+1)} = b^{(i)} - \gamma \frac{\partial L}{\partial b}.$$
   γ (*learning rate*) determines how big a step the adjustment is making in the negative direction of the gradient.
5) **Adjusting Weights and Biases**
   In this step, network's parameters are updated according to the calculated gradients. If γ is too big, algorithm would diverge and never find the minimum. On the other hand, if γ is too small, it would take too much time to find the minimum. So it is necessary to find optimal value for γ in order to find minimum in reasonable time. For more details about choosing optimal γ, see [11].

Model

Model is the main component used to bind all of the above-explained components together. It represents a public Application Programming Interface (API) through which users can interact with the library, define the network's architecture and train the network.

Knowing that representability and trainability are two main attributes that describe the network, layer structure has a huge impact on the network's performance. The network is "representable" if it can represent a solution to the problem with a certain level of complexity. The higher the number of

layers, the more complex problems a network can represent. Trainability sets a threshold to the number of hidden layers and the overall complexity of the network. Too many layers and a number of nodes in each layer can lead to slow convergence or high computational power demands.

The task of the network is to make a decision, whether the car should turn left, right or stay in the same lane. That decision is represented with a column vector consisting of three numbers that represent probabilities for each action. Softmax is our activation function which helps the network achieve this by squashing the outputs of each unit to be in the interval $[0, 1]$, and also dividing each output such that the total sum of the outputs is equal to 1.

IV. System Implementation

The system is distributed in two main components: the React.js application that implements the environment and a Flask web application that implements the neural network model. They communicate over HyperText Transfer Protocol (HTTP).

Network Architecture

The architecture of a network [4] used for training the self-driving agent is determined by a numerical experiment through a large number of simulations and it is shown in Fig. 2. The implemented network consists of three layers: an input layer with 6 neurons, a hidden layer with 4 neurons and an output layer with 3 neurons. The input layer does not have an activation function or more precisely, the activation function of the input layer is the identity function. The activation function used in the hidden layer is the ReLU.

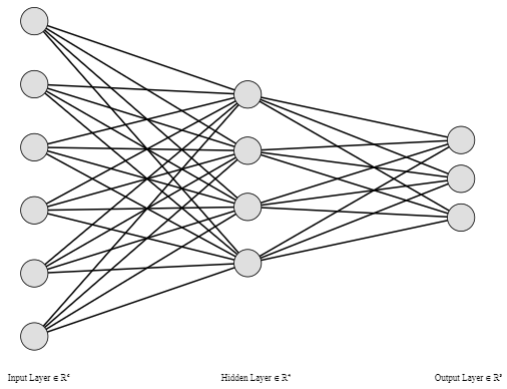


Fig. 2. Network architecture

Environment

The environment (Fig. 1) is represented through the React.js application. It is divided into three main components: highway environment, settings panel, and results pane.

The highway environment is represented through HyperText Markup Language (HTML) canvas. It consists of traffic lanes, passing cars and the controlled car. Traffic movement and canvas dynamics are managed using the p5.js library. Passing cars are spawned at the top of the screen and are moving with the constant speed relative to each other. The controlled car's sensor zones are displayed around him with a transparent background color. If a passing car is detected in any of the zones, the transparency is reduced and the color becomes more solid which indicates that the sensor is active. This is useful when collecting training data because the user can see the boundaries of the agent's "vision" and drive accordingly. The settings panel is used for changing the parameters of the environment. It is divided into three sections: environment, traffic, movement and agent settings. In the environment section, the user can change the number of traffic lanes in range of 4 to 10. In the traffic section, the car creation interval and the number of cars spawned on each interval can be modified. The maximum speed of the controlled car can be changed in the movement section and visibility of the sensor zones can be toggled in the agent subsection. Next to the agent label, the connection status with the backend application is displayed. Ability to change the parameters enables the option to change the "difficulty", or the complexity, of the environment.

In the results pane, the current speed of the car, average speed and the number of passing cars are displayed. That is the direct indicator of how well the agent is controlling the car. Below the results pane, there is a canvas that holds the figure of the neural network model. According to the outputs of each layer calculated in the forward pass, the nodes and the connection lines are shaded.

Collecting the Data: In manual mode, the user can control the car using the arrows keys and data is collected that way. The data is represented as a list of 6 numbers, a snapshot that contains information gathered from the sensors. Three of those six numbers represent a distance from the nearest car that is in the green zone of one of three sensors. In order to normalize input data, each distance is divided by the maximum distance, the green zone length, which ensures that those numbers are in interval $[0, 1]$. The other three numbers are boolean values, 0 or 1, which only indicate the presence of a car in the red zone of that sensor. Apart from the snapshot, a label that represents the user's decision is also collected in the form of a whole number between -1 and 1 (-1 for left, 0 for forward and 1 for right) and is pushed to the list. So the final result is a list of 7 numbers. The result is then added to the list of collected data that represents the dataset for training the model.

Training the Model: The model is created using the following code:

```
model = Model([[6], [4, "relu"], [3,
    ↪ "softmax"]],
    ↪ loss_function='cross_entropy',
    ↪ learning_rate=0.001)
```

After the model is initialized, training examples and labels are extracted from the saved .csv dataset and passed as arguments for training the model.

```
model.train("gradient_descent",
    inputs=np.array(inputs),
    labels=np.array(labels), epochs=500,
    batch_size=1)
```

Each input argument, such as learning rate, batch size or a number of epochs, has an impact on the time spent training and the quality of the results. The main goal is to find the best proportion of those two factors.

One epoch represents only one forward and backward pass of the dataset through the network. In every epoch, weights and biases are changed and adjusted. One pass leads to underfitting the curve in gradient descent algorithm, so more epochs are needed in order to achieve better results. On the other hand, too many epochs lead to a big increase in time spent training and overfitting the curve. It has been determined experimentally that the most satisfying number of epochs for this problem is 1000.

In case that the dataset is too big, data is separated into batches. Using this technique, the data can be passed in smaller groups, achieving faster training time, and having more training examples in each update of parameters. Because the self-driving dataset is not too big, the chosen batch size is 1, which means that only one training example is used for each parameter adjustment.

Agent Controlling the Car: After training the model and starting the Flask web service holding the network, the user can turn on the auto mode. In auto mode, the car is controlled by the agent. The current state of sensors and the name of the trained model are sent to the Flask web service every 500ms through an HTTP request. The input data is passed to the trained model. The model makes a prediction based on the passed data and the response is generated and sent back to the agent. The response comes in the form of a probability distribution vector. The index of the element in the vector that has the highest probability represents the action that the agent should make. The agent then moves the car based on the parsed prediction.

Each time the agent gets stuck in the traffic, the user is able to move the controlled car and correct the agent's decision. Correcting the agent means switching to manual mode. Each correction is saved in a list. Then, the user is able to download saved corrections in the form of a new dataset, add them to the initial dataset and to refine the agent's decision making by retraining the model.

Because it is expected that the controlled car has a built-in safety system, the acceleration and braking are performed automatically. When the front sensor detects that a car has entered the safety zone, it triggers the braking system and the controlled car starts to decelerate in proportion to the distance from the facing car and the current speed. If there are no detected cars in the facing line, the controlled car starts to accelerate in proportion to the current speed until it reaches the maximum speed of N km/h. N takes the maximum speed out of a set $80, 120, 160$ km/h.

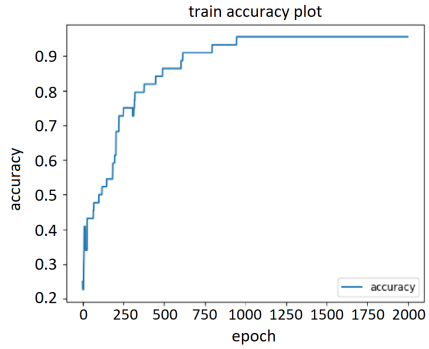
V. Simulation Results

This section presents the results of training the model, testing the agent's behaviour and comparison to the user's behaviour.

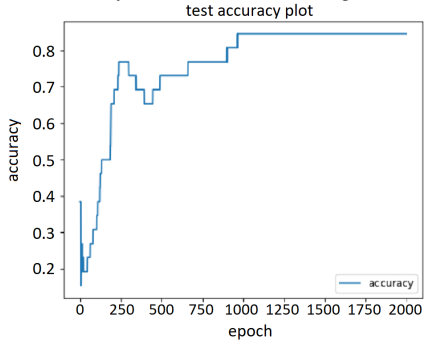
Model Training

The accuracy of the model represents how good the predictions that the model makes according to the given dataset are. It is calculated by comparing the received results from the model and expected results from the dataset represented in percentages.

Using the network architecture and training parameters explained in Section IV, the calculated accuracy over the training dataset is 95% and the calculated accuracy over the test dataset is 85%. Each training parameter affects the accuracy of the model. With an increased number of epochs, the accuracy may stay the same, but the time spent training increases. After the successful training, the obtained results are following.



(a) Accuracy of the model for training dataset



(b) Accuracy of the model for test dataset

Fig. 3. Accuracy of the model

Fig. 3a presents the accuracy of the model for training dataset and it can be noticed that the accuracy score rises from 20% to 95% within first 1000 epochs, and stays stable until the end of the training. Fig. 3b presents the accuracy of the model for test dataset and it can be noticed that the accuracy score rises from 10% to 85% within first 1000 epochs, and stays stable until the end of the training.

It is expected that the accuracy for test dataset is not as good as the accuracy for training dataset, but still test results are very satisfying.

Agent's behaviour based on a quality of the dataset

Clear information about lane changing, means positioning the agent in a good spot for evading the incoming car without hitting the safety (red) zone. A good dataset does not need to have a lot of examples, it needs to have a clear set of instructions and cover most typical scenarios making sure that it contains a balanced number of labeled decisions. It is not good if any of the three decisions is dominant in the dataset.

Training is considered not adequate if the user does not give clear information about his actions and many decisions with similar, or same, inputs are in contradiction, which makes the model disorientated and not able to move in the right direction. Even though the accuracy of the trained model is at a high percentage, the agent still acts disoriented and makes wrong turns, often getting stuck in the traffic.

Users vs Agent driving

Simulation experiment is based on three main drivers categories. First experienced user, then inexperienced user and finally, trained agent. First two categories covers manual driving with ten different participants in both categories and averaging their results. Different scenarios were tested with various maximal speeds of car, number of lanes, number of cars spawning and car spawning speed. Results of simulation are presented by total distance that car could pass in 30 minutes. Three representative results are presented in Fig. 4. Maximal speed was $120km/h$, five lanes, one spawning car with all spawning speeds modes. From this figure, it is clear that in most cases after 30 minutes of simulation, agent approach shows the best behaviour that is maximal elapsed distance.

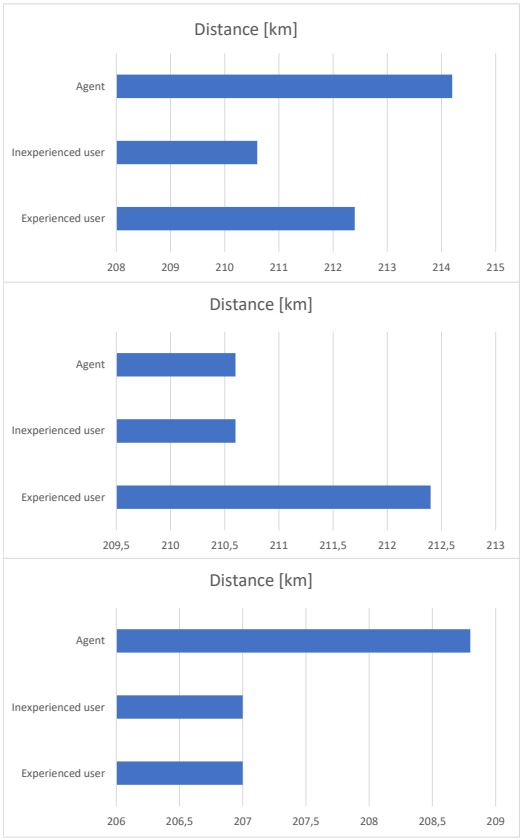


Fig. 4. Simulation results for three different scenarios. Maximal speed $120km/h$, five lanes, one spawning car with spawning speeds modes: slow (upper figure), normal (middle figure) and fast (bottom figure).

VI. Conclusion

Multilayer Perceptron Model and supervised learning paradigm used in this paper present one possible approach for training the model to autonomously drive itself trough traffic jams. The aim of this project was to achieve successful autonomous driving in a highway traffic jam in terms of maximizing elapsed distance in traffic by maximizing the average speed of the car.

After network's parameters estimation and training the model, agent's behaviour was tested in simulations. Although this solution does not ensure that the vehicle is completely autonomous, obtained training results have proved satisfying on a tested system in terms of getting out of the traffic jam as fast as possible.

In the future work, we will be focused on a new approach that will allow us to generate realistic camera image for simulation directly using sensor data collected by a self-driving car.

VII. Acknowledgment

This paper has been supported by the Ministry of Education, Science and Technological Development through the project no. 451-03-68/2020-14/200156: "Innovative scientific and artistic research from the FTS domain". (A.B.)

References

[1] J. Baron, "Thinking about global warming," *Climatic Change 77, 137–150 (2006). DOI: 10.1007/s10584-006-9049-y*, 2006.

[2] A. Takacs, I. J. Rudas, D. Bosl, and T. Haidegger, "Highly automated vehicles and self-driving cars.," *IEEE Robotics Automation Magazine 25(4):106-112, DOI: 10.1109/MRA.2018.2874301*, 2018.

[3] S. Grigorescu, B. Trasnea, T. Cocias, and G. Macesanu, "A survey of deep learning techniques for autonomous driving.," *Journal of Field Robotics, Online ISSN:1556-4967, DOI: 10.1002/rob.21918*, 2019.

[4] A. P. Engelbrecht, "Computational intelligence: An introduction," *Wiley Publishing, ISBN:978-0-470-03561-0*, 2007.

[5] L. Bjelica, S. Vulin, and A. Buljević. http://self-driving.bjelicaluka.live/, 2020.

[6] "Keras documentation." https://keras.io/.

[7] D. Kriesel, "A brief introduction on neural networks." http://www.dkriesel.com/_media/science/neuronalenetze-en-zeta2-2col-dkrieselcom.pdf.

[8] J. Nocedal and S. J. Wright, "Numerical optimization, second edition," *Springer, New York, ISBN: 978-0-387-30303-1, DOI: 10.1007/978-0-387-40065-5*, 2006.

[9] A. Nagpal, "L1 and l2 regularization methods." https://towardsdatascience.com/l1-and-l2-regularization-methods-ce25e7fc831c, 2017.

[10] M. J. Kochenderfer and T. A. Wheeler, "Algorithms for optimization," *The MIT Press. ISBN:978-0-262-03942-0*, 2019.

[11] Y. xiang Yuan, "A new stepsize for the steepest descent method," *Journal of Computational Mathematics Vol. 24, No. 2 (MARCH 2006), pp. 149-156*, 2006.

AR Speech Model Parameter Estimation Using a Robust Non-Recursive Estimator

Ana Lazović, Mihailo Bjekić and Aleksandra Marjanović

***Abstract*—In this paper a robust block linear prediction (RBLP) method for autoregression (AR) model parameter estimation of speech signal is considered. The considered method consists of two separate iterative steps which are described in detail. The method is tested on both synthesized and natural human speech in the presence of outliers and additive measurement noise. A comparative analysis of the RBLP method and the conventional linear prediction method shows that the robust method gives results which are less biased and have a smaller variance.**

***Index Terms*—AR model, outliers, RBLP, estimation, speech.**

I. Introduction

Statistical signal processing is frequently based on the assumption of a priori known probability data distribution, stationarity, linearity as well as independence of stochastic processes [1]. In most engineering problems, real system model parameter estimation methods assume that stochastic processes have Gaussian distribution, which is, in most cases, justified. The assumption of Gaussian distribution allows for a straightforward implementation of optimal estimators. On the other hand, in the case of large realizations of stochastic perturbations, as well as non-Gaussian additive measurement noise, the assumption is not justified. Optimal estimation methods based on the assumption of Gaussian distribution are extremely sensitive to deviations from the assumed Gaussian distribution, where, in some cases, even a small deviation from the assumed distribution results in significant deterioration in estimator performance.

Even though in many cases, the Gaussian distribution is justified, there are many situations where it has been determined that the actual data distribution is far from Gaussian. One of the main reasons for the distribution deviations are the impulse perturbations, i.e. outliers.

As it can be seen in [2], outliers can be found in various areas, such as image processing, speech signal processing, sensor networks, medicine, industry, etc. Because of the prevalence of outliers in different areas, outlier detection and elimination are of great importance.

Based on the fact that the conventional linear prediction (CLP) methods assume the Gaussian distribution of the residuals, CLP methods assign equal weights to all residuals [3]. If the residual distribution varies from the Gaussian distribution, the results of the CLP methods may be inaccurate with both a large bias and variance. Due to the unsatisfactory results, robustification must be performed.

There are two different approaches to estimator robustification, the diagnostic approach and statistically robust approach [4]. Diagnostic approach detects outliers and replaces them using classical parameter estimation, as discussed in [5]. The statistically robust approach uses the entire dataset and bounds the influence of outliers using influence functions.

In this paper, robust estimation of autoregression (AR) model of speech signal is discussed. Due to the impulse-type quasiperiodic excitation of the vocal tract, distribution of the residuals varies from the Gaussian distribution. This variation from the Gaussian distribution is considered to be a result of the presence of outliers, so the performance of CLP methods decreases.

Robust parameter estimation techniques are efficient even without the a priori known statistical characteristics of the perturbations in the system [6]. Robust estimation techniques are less sensitive to the presence of impulse perturbations in the system, that is, robust estimation techniques enable outlier detection in the signal and its elimination.

Specific application of outlier detection in AR parameter estimation of speech signal can be found in [7], where a voice-based Parkinson's disease detection is considered.

In this paper, robust block linear prediction (RBLP) method for an AR parameter estimation of speech signal, described in detail in [8], is analyzed. The algorithm in [8] is improved, especially the second step of the algorithm. The outliers considered in [8] originate only from the impulse-type quasiperiodic excitation, whereas, in this paper, outliers originating from the measurement errors, also referred to as additive outliers (AO) [4], will be analyzed. The effects of measurement noise, as well as the effects of different types of nonlinearities are considered. The RBLP method will be tested on both synthesized and natural voiced signals in the presence of additive measurement noise.

II. Estimation problem

The AR model of the speech signal [9], considered in this paper, shown in the Fig. 1, is given in the form:

$$E(z)G(z)V(z)L(z) = \frac{E(z)}{A(z)} = S(z), \tag{1}$$

Ana Lazović is a PhD student at the School of Electrical Engineering, University of Belgrade, 73 Bulevar kralja Aleksandra, 11020 Belgrade, Serbia (e-mail: analazovic1995@gmail.com).

Mihailo Bjekić is a PhD student at the School of Electrical Engineering, University of Belgrade, 73 Bulevar kralja Aleksandra, 11020 Belgrade, Serbia (e-mail: mihailobjekic@gmail.com).

Aleksandra Marjanović is with the School of Electrical Engineering, University of Belgrade, 73 Bulevar kralja Aleksandra, 11020 Belgrade, Serbia (e-mail: amarjanovic@etf.bg.ac.rs).

where $E(z)$ is the z transform of the excitation of the glottal tract, $G(z)$ is the transfer function of the glottal tract, $V(z)$ is the transfer function of the vocal tract and $L(z)$ is the transfer function of the radiation on the lips. Combined transfer function of $G(z)V(z)L(z)$ can be represented as an inverse filter given by:

$$A(z) = \frac{1}{G(z)V(z)L(z)}. \tag{2}$$

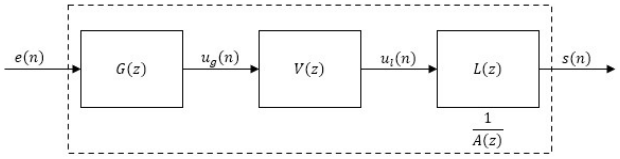


Fig. 1. A linear model of the speech production system.

In this paper, two impulse-type quasiperiodic excitations of the AR model, shown in the Fig. 2, are considered. First type of excitation is the train of Dirac pulses, while the second type of excitation is the twice differentiated Strube's glottal wave [10], which is a more accurate representation of the speech signal excitation. Both excitations are normalized and periodic with the period equal to the fundamental period of speech.

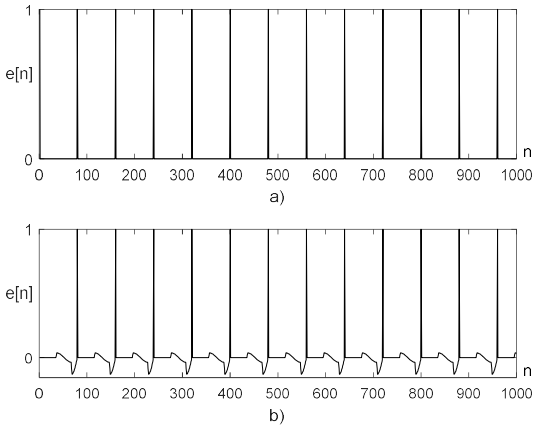


Fig. 2. Impulse-type quasiperiodic excitation of the AR speech model with the fundamental period of 8 ms with the sampling frequency of 10 kHz: a) Dirac pulse train; b) twice differentiated Strube's glottal wave.

AR model of the speech signal of the p-th order, in the time domain, is given by (3).

$$s(n) + \sum_{i=1}^{p} a_i s(n-i) = e(n) \tag{3}$$

In the case of the $N + p$ known speech signal samples $s(n)$, (3) can be rewritten in the matrix form:

$$S = H\theta + E, \tag{4}$$

where S is a vector of the speech signal samples with the length of N, θ is a vector of coefficients a_i of the AR model with the length of p, E is a vector of excitation samples $e(k)$ with the length of N and H is a matrix of speech signal observations with the dimensions $N \times p$.

$$S^{\mathrm{T}} = [s(k+1)\, s(k+2) \ldots s(k+N)] \tag{5}$$

$$\theta^{\mathrm{T}} = [a_1\, a_2 \ldots a_p] \tag{6}$$

$$E^{\mathrm{T}} = [e(k+1)\, e(k+2) \ldots e(k+N)] \tag{7}$$

$$H = \begin{bmatrix} -s(k) & \cdots & -s(k+1-p) \\ -s(k+1) & \cdots & -s(k+2-p) \\ \vdots & \ddots & \vdots \\ -s(k+N-1) & \cdots & -s(k+N-p) \end{bmatrix} \tag{8}$$

One of the CLP methods for speech parameter estimation is the least-squares (LSQ) method [8]. In the presence of outliers, LSQ method does not give satisfactory results. The reason for decreased performance of the LSQ method lies in the fact that the LSQ method is based on the assumption of the Gaussian data distribution and assigns equal weights to all samples including outliers. For the purpose of overcoming the problems that the LSQ method encounters, robustification is proposed.

III. RBLP METHOD

One of the robust estimators used for estimating AR model parameter coefficients is the approximate maximum likelihood estimator, shortly M-estimator [10]. M-estimators are based on the approximation of the unknown probability density function, and its corresponding nonlinear score function, which reduces the effects of outliers.

The minimization problem whose solution is a robust M-estimation of the parameter vector θ is defined by:

$$J_N(\hat{\theta}) = \sum_{i=1}^{N} \rho\left[\frac{s_i - h_i^T \hat{\theta}}{d}\right], \tag{9}$$

where s_i is the i-th element of S, h_i is the i-th row of H, N is the dimension of S, d is the scaling factor and ρ is the nonlinear score function [8].

The minimization problem can be reformulated into:

$$\sum_{i=1}^{N} \frac{1}{d} h_{ij}\, \psi\left[\frac{s_i - h_i^T \hat{\theta}}{d}\right] = 0, \qquad j = 1, \ldots, p, \tag{10}$$

where h_{ij} is the element in the i-th row and the j-th column of H, and ψ is the influence function $\psi(\cdot) = \rho'(\cdot)$.

In this paper, a non-recursive RBLP method for AR parameter estimation is considered [8]. The proposed RBLP method starts from the estimates obtained using one of the CLP methods, such as the LSQ method, and further consists of two separate iterative steps.

A. Dutter algorithm

The first step of the non-recursive RBLP is based on M-estimation [10]. The used nonlinear score function is Huber's score function:

$$\rho(x) = \begin{cases} \frac{x^2}{2}, & |x| \le k \\ k|x| - \frac{k^2}{2}, & |x| > k \end{cases}, \tag{11}$$

where k ensures the desired efficiency in the case of nominal Gaussian distribution. The appropriate influence function is given by:

$$\psi(\mathrm{x}) = \begin{cases} x, & |x| \le k \\ k\ \mathrm{sign}(x), & |x| > k \end{cases}. \tag{12}$$

Because the solution of the system (10) cannot be obtained in the closed form, Dutter iterative procedure is used [10]. For the initial guess θ_0 of the unknown vector θ, result of the LSQ algorithm is used, while the initial scaling factor d_0 is obtained by (13).

$$d_0 = \mathrm{median}\left(\frac{|s_i - \mathrm{median}(s_i)|}{0.6745}\right) \tag{13}$$

Dutter iterative method consists of several steps:

Step 1: Calculation of the nonnormalized residual:

$$n_i(\hat{\theta}_0) = s(i) - h_i^T \hat{\theta}_0. \tag{14}$$

Step 2: Calculation of a new estimation of the scaling factor d_1:

$$d_1^2 = \frac{1}{(N-p)E\{\psi^2(z)\}} \sum_{i=1}^{N} d_0^2\, \psi^2[\varepsilon_i(\hat{\theta}_0)], \tag{15}$$

where the normalized residual is given by:

$$\varepsilon_i(\hat{\theta}_0) = \frac{n_i(\hat{\theta}_0)}{d_0}, \tag{16}$$

and $E\{\psi^2(z)\}$ is the mathematical expectation for the standard normal random variable:

$$E\{\psi^2(z)\} = \int_{-\infty}^{\infty} \psi^2(z) p(z) dz, \tag{17}$$

where $p(z)$ is the probability density function of the standard Gaussian random variable z $(N(0,1))$. For the adopted parameter $k = 1.5$, (17) results in 0.7785.

Step 3: Residual Windsorization:

$$\Delta_i = \begin{cases} n_i(\hat{\theta}_0), & |\varepsilon_i(\hat{\theta}_0)| \le k \\ k d_1, & \varepsilon_i(\hat{\theta}_0) > k \\ -k d_1, & \varepsilon_i(\hat{\theta}_0) < k \end{cases}. \tag{18}$$

Step 4: Calculation of the regression coefficients θ using LSQ estimation:

$$\Delta\theta = [H^{\mathrm{T}}H]^{-1} H^{\mathrm{T}} \psi_v;\ \psi_v = \{\Delta_1, \dots, \Delta_N\}. \tag{19}$$

Step 5: Update of the estimation of the parameter θ:

$$\hat{\theta}_1 = \hat{\theta}_0 + q\Delta\theta, \tag{20}$$

where q is the correction factor defined by:

$$q = \min\left[\frac{1}{2\, erf(k)}, 1.9\right]. \tag{21}$$

Step 6: Repetition of the steps 1-5 with the new estimations $\hat{\theta}_1$ i d_1 as the initial guesses until the fulfillment of the termination conditions:

$$\left|\hat{\theta}_1^{(m)} - \hat{\theta}_0^{(m)}\right| < \eta \left|\hat{\theta}_0^{(m)}\right|, \tag{22}$$

$$|d_1 - d_0| < \eta \left|\hat{\theta}_0^{(m)}\right|, \tag{23}$$

where $\eta > 0$ is a conveniently chosen small number and $\hat{\theta}^{(m)}$ is the m-th element of the vector $\hat{\theta}$.

B. Weighted Least Squares Algorithm

The second step of the RBLP method, considered in [8], is the Weighted Least Squares Algorithm (WLSQ). The estimations obtained by the Dutter algorithm are used as the initial guesses for the iterative WLSQ algorithm, which further reduces the effects of outliers.

In this paper an improvement of the WLSQ algorithm, given in [8], is presented.

The minimization problem (10), in the case of the WLSQ algorithm, is given in the form:

$$\sum_{i=1}^{N} h_{ij} w_{i0} \left(s(i) - h_i^{\mathrm{T}} \hat{\theta}\right) \approx 0; j = 1,2, \dots, p\ , \tag{24}$$

where coefficients w_{i0} are given by:

$$w_{i0} = \begin{cases} \dfrac{\psi\left(\frac{s(i) - h_i^{\mathrm{T}} \hat{\theta}_0}{d_0}\right)}{\frac{s(i) - h_i^{\mathrm{T}} \hat{\theta}_0}{d_0}}, & s(i) \ne h_i^{\mathrm{T}} \hat{\theta}_0 \\ 1, & s(i) = h_i^{\mathrm{T}} \hat{\theta}_0 \end{cases}, \tag{25}$$

where $\hat{\theta}_0$ and d_0 are the initial estimations obtained using the Dutter algorithm.

Matrix notation of the system is given by:

$$W_0 = diag\{w_{10}, \dots, w_{N0}\};\ S^{\mathrm{T}} = \{s(1), \dots, s(N)\}. \tag{26}$$

The new estimation of the vector of parameters $\hat{\theta}$ is calculated using:

$$\hat{\theta} = (H^{\mathrm{T}} W_0 H)^{-1} H^{\mathrm{T}} W_0 S. \tag{27}$$

The WLSQ algorithm is repeated using the new estimation $\hat{\theta}$ as the initial guess for the vector parameter θ. The scaling factor d_0 changes with each iteration, and its new value is calculated using (15) with an appropriate $E\{\psi^2(z)\}$, which is the main improvement of the algorithm given in [8] The algorithm is repeated until the terminal condition (22) is satisfied.

The influence functions that can be used in this step of the RBLP algorithm are either Andrews (28) or Tukey (29) nonlinearity [10]. The influence functions are shown in Fig. 3.

$$\psi(x) = \begin{cases} \sin(x/a), & |x| < a\pi \\ 0, & |x| > a\pi \end{cases}, \; a \in [0.45\ 0.65] \qquad (28)$$

$$\psi(x) = \begin{cases} x(1-(x/a)^2), & |x| < a \\ 0, & |x| > a \end{cases}, \; a \in [1.5\ 2] \qquad (29)$$

The influence functions (28) and (29) can lead to divergence of the algorithm because of their non-convexity. By limiting the number of the iterations of the WLSQ algorithm to only a few, the problem of the non-convex influence functions can be overcome, while the quality of the estimation does not deteriorate.

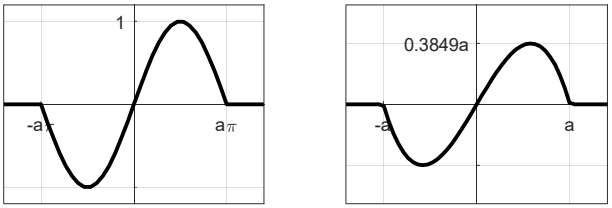


Fig. 3. Nonlinear influence functions $\psi(\cdot)$: a) Andrews nonlinearity; b) Tukey nonlinearity.

IV. Results and Discussion

In this paper, the RBLP method is tested on both synthesized and natural voiced signal "a" in the presence of additive measurement noise. For the purpose of testing the RBLP method, estimations of the AR model parameters are obtained by performing both a CLP method and the RBLP method at the sliding rectangular window with the length of 256 samples.

In the second step of the RBLP method, Andrews nonlinearity, with the parameter $a = 0.5$ and $E\{\psi^2(z)\} = 0.5617$, is used, while the number of iterations is limited to four iterations.

A. Synthesized vowel "a"

The considered AR model of the order $p = 8$ is the same as in [8], with the parameters: $a_1 = -2.22$, $a_2 = 2.89$, $a_3 = -3.08$, $a_4 = 3.27$, $a_5 = -2.77$, $a_6 = 2.35$, $a_7 = -1.7$ and $a_8 = 0.75$. The excitation of the AR model is described in detail in the second section. The output of the AR model is noised by additive Gaussian noise $N(0,0.0002)$ and contaminated with outliers. One of the most common types of outliers in time series, such as the speech signal, are the AO [11], which are the type of outliers used in this model. The outliers are generated as a random variable with the Gaussian distribution $N(0,0.02)$ with the probability of occurrence of 5% [4].

In the Fig. 4, results of the RBLP method for parameter a_1, performed on noised and non-noised synthesized vowel "a", in the case of train of Dirac pulses excitation (Fig. 2(a)), are shown.

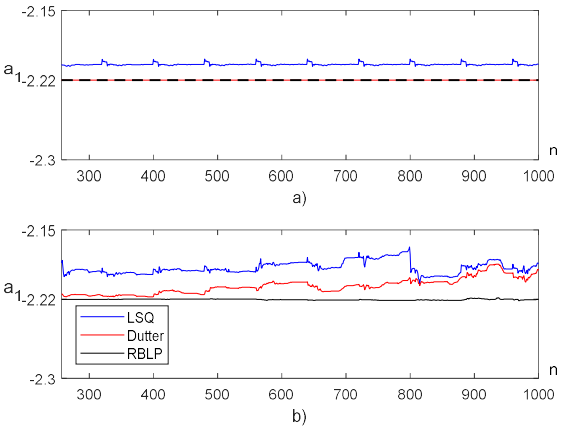


Fig. 4. Comparative analysis of the CLP and the RBLP methods performed on synthesized vowel "a", parameter a_1 estimates for: a) non-noised b) noised train of Dirac pulses excitation.

In the Fig. 4(a) the results of the RBLP method without both additive Gaussian noise and outliers, are shown. For signals as simple as this, it can be seen that the first step of the RBLP method gives satisfactory results that are both unbiased and insensitive to the position of the sliding window, unlike the results of the LSQ algorithm. In the Fig. 4(b) the same type of excitation is used, but in the presence of both additive Gaussian noise and outliers. In the presence of noise and outliers, the importance of the second step of the RBLP method can be seen. The WLSQ algorithm ensures an unbiased estimation with a smaller variance than the first step of the RBLP method.

In the Fig. 5 the results of the RBLP method for the parameter a_1, in the case of an excitation that is twice differentiated Strube's glottal wave (Fig. 2(b)), are shown.

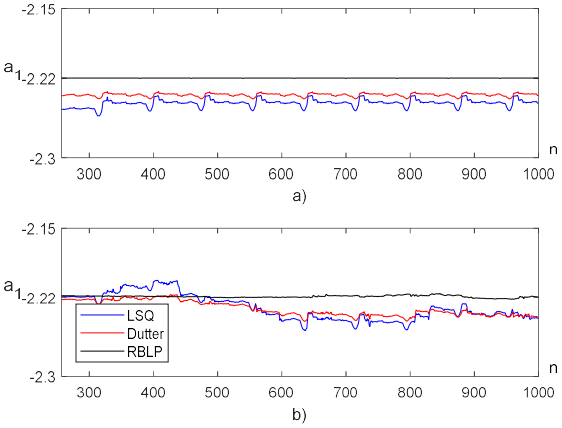


Fig. 5. Comparative analysis of the CLP and the RBLP methods performed on synthesized vowel "a", parameter a_1 estimates for: a) non-noised b) noised twice differentiated Strube's glottal wave excitation.

When there is no noise nor outliers (Fig. 5(a)), in the contrast to the case when the excitation is the train of Dirac pulses (Fig. 4(a)), the first step of the RBLP method does not give good enough results. The unsatisfactory results of the Dutter method, which reflect in the presence of a large bias in the parameter estimation, are improved using the WLSQ algorithm, whose results are the same as in (Fig. 4(a)). In the presence of noise and outliers (Fig. 5(b)) the variance of the

estimation of the parameters is bigger than in the previous case, while the estimation is still unbiased.

In the Fig. 6 the results of the RBLP method, for the twice differentiated Strube's glottal wave excitation, with noise, for both constant and variable scaling factor d in the WLSQ algorithm, are shown. It can be seen that the result of the RBLP algorithm with variable scaling factor has both smaller variance and bias, which is the main improvement of the algorithm in [8].

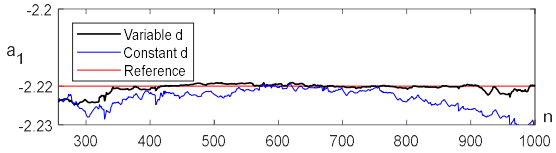


Fig. 6. Comparative analysis of the RBLP with constant and variable scaling factor in the second step of the algorithm.

B. *Natural human spoken vowel "a"*

The algorithm was tested on the natural human spoken vowel "a" which was recorded with the sampling frequency of 10 kHz and filtered with a low-pass filter with the cut-off frequency of 4 kHz. Because the exact values of the parameters of the AR model are not a priori known, for the reference values an estimation provided by the CLP method, performed on the sliding window with the length shorter than the fundamental period, is adopted [8]. When the sliding window for the CLP method does not contain the part of the signal originating from the excitation, the estimation is considered unbiased. For the length of the sliding window of the CLP method, 37 samples are adopted, as the fundamental period of the test signal is 47 samples. The adopted order of the AR model is $p = 8$.

The filtered signal is given in Fig. 7(a), while the estimates for the parameter a_1 are given in Fig. 7(b). For the reference values of the parameter a_1, results of the CLP method, with a shorter window length, when there is no effect of the impulse type excitation, are adopted. In the Fig. 7(b), there is no effect of the impulse type excitation when the CLP estimates are at their higher values.

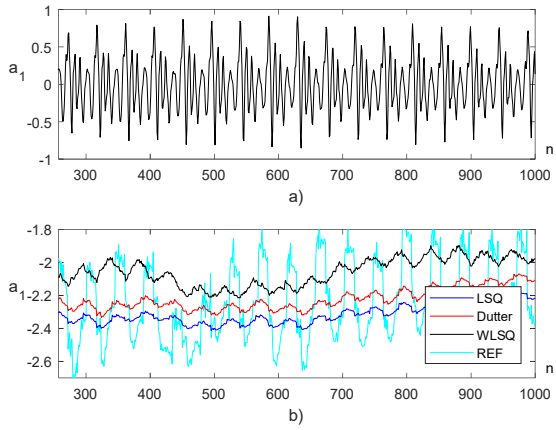


Fig. 7. Comparative analysis of the CLP and the RBLP methods performed on natural human spoken vowel "a": a) Filtered spoken vowel "a"; b) a_1 parameter estimates for the CLP and the RBLP methods.

It can be seen that the estimates provided by the Dutter algorithm are less biased than the starting LSQ estimates, while the WLSQ algorithm gives significantly less biased estimates.

V. Conclusion

In this paper, an improvement to the RBLP algorithm, described in detail in [8], has been proposed. The main improvement of the algorithm is in the second step, the WLSQ algorithm, where it is proposed that the scaling factor changes with each iteration. The experimental results given in [8] have been confirmed. The algorithm has also been tested to the presence of the additive measurement noise as well as outliers, and it has been shown that the RBLP method is much more robust than the CLP methods. In the case of the natural human speech, the results of the RBLP method have a significantly smaller bias than the results of the CLP methods. Overall, the RBLP method proves to be much better for estimating the AR model parameters of the speech signal for both synthetized and natural human speech in the presence of outliers.

Acknowledgment

This research was partially supported by the Ministry of Education, Science and Technological Development of the Republic of Serbia, Contracts No. TR32038.

References

[1] A. M. Zoubir, V. Koivunen, Y. Chakhchoukh and M. Muma, "Robust Estimation in Signal Processing: A tutorial style treatment of fundamental concepts", *IEEE Signal Processing Magazine*, vol. 29, no. 4, pp. 61–80, July, 2012.

[2] K. Singh and S. Upadhyaya, "Outlier detection: applications and techniques", *International Journal of Computer Science Issues*, vol. 9(1), no. 3, pp.307–323, January, 2012.

[3] K.Y. Lee, B. G. Lee, I. Song and S. Ann, "Robust Estimation of AR Parameters and Its Application for Speech Enhancement," in IEEE International Conference on Acoustics, Speech and Signal Processing, San Francisco, USA, vol. 1, pp. 309–312, March, 1992.

[4] M. Muma and A. M. Zoubir, "Bounded Influence Propagation τ-Estimation: A New Robust Method for ARMA Model Estimation", *IEEE Transactions on Signal Processing*, vol. 65, no. 7, pp. 1712–1727, April, 2017.

[5] A. Azad and L. Mili, "Robust Speech Filter And Voice Encoder Parameter Estimation using the Phase-Phase Correlator", *IEEE Transactions/ACM on Audio, Speech, and Language Processing*, vol. 28, pp. 592–604, 2020.

[6] B. Kovačević, Z. Banjac i Ž. Đurović, *Filtracija stohastičkih signala – Optimalni, adaptivni i robusni estimatori parametara i stanja*, Beograd, Srbija: Akademska misao, 2017.

[7] A. H. Poorjam, M. S. Kavalekalam, L. Shi, Y. P. Raykov, J. R. Jensen, M. A. Little and M. G. Christensen, "Automatic Quality Control and Enhancement for Voice-Based Remote Parkinson's Disease Detection", IEEE Access, 2020.

[8] M. Veinović, B. Kovačević and M. Milosavljević, Robust Non-recursive AR Speech Analysis, *Signal Processing*, vol. 37, pp. 189–201. 1994.

[9] D. Bajić, M. Veinović, B. Kovačević i M. Milosavljević, „Praćenje formantnih trajektorija primenom robusnih metoda linearne predikcije", u ETRAN, Zlatibor, Srbija, str. 493–496, Jun, 1995.

[10] B. Kovačević, M. Milosavljević, M. Veinović and M. Marković, *Robust Digital Processing of Speech Signals*, Berlin, Germany: Springer, 2017.

[11] A. S. Ahmar, S. Guritno, Abdurakhman, A. Rahman, Awi, Alimuddin, I. Minggi, M. A. Tiro, M. K. Aidid, S. Annas, D. U. Sutiksno, D. S. Ahmar, K. H. Ahmar, A. A. Ahmar, A. Zaki, D. Abdullah, R. Rahim, H. Nurdiyanto, R. Hidayat, D. Napitupulu, J. Simarmata, N. Kurniasih, L. A. Abdillah, A. Pranolo, Haviluddin, W. Albra and A. N. M. Arifin, "Modeling Data Containing Outliers using ARIMA Additive Outlier (ARIMA-AO)", *Journal of Physics: Conference Series*, vol. 954, March, 2018.

Modelling and Control of a Series Direct Current (DC) Machines Using Feedback Linearization Approach

Mitra Vesović, Radiša Jovanović, Lara Laban, Vladmir Zarić

***Abstract*—The nonlinear feedback control system applied to the direct current - DC motor is proposed in this research. Nonlinear mathematical model has been obtained using dead zone, Coulomb and viscous friction. The system stability has been analyzed using Lyapunov stability theory. The effectiveness and the comparison of the performance between linear and nonlinear control algorithm have been validated using Matlab/Simulink software. From the conclusions, based on the simulation and experimental results that have been provided, it is easy to see that nonlinear control systems are more suitable and have a better reach for controlling position. The validity of using feedback linearization in DC motors has been proven.**

***Index Terms*—feedback linearization; nonlinear systems; nonlinear control; identification**

I. Introduction

Position control in a direct current motor (DC) has been one of the most fundamental and challenging tasks, that has been largely studied for decades. Many studies have been done to model electrical machines. For example, serial DC motor has often been modeled as linear object. On the other hand, models in which motor current or flux are found as essential parameters are considered to be nonlinear [1]. This paper presents the design and implementation concerning both, linear and nonlinear models for the system. They are obtained for identification and control purposes. The major nonlinearities in the system, such as Coulomb friction and dead zone, are investigated and integrated in the nonlinear model [2].

In the different types of application accurate control of position in DC machines is a great challenge for engineers. Disparate controllers have been proposed to lead the position of DC machines into the desired value. For example Proportional-Integral-Derivative (PID) controller is a popular controller in industries due to simple structure, low cost and easy to implement. It provides reliable performance for the system if PID parameter is identified properly. But it suffers due to lack of robustness [1]. The linear approximation, of the nonlinear state space representation of the series DC motor, around the equilibrium point and PI controller design the tracking performance is deteriorated in the periods in which the speed is reduced. This is due to the fact that the input signal $u(t)$ is limited to a minimum of 0 [V]. That is, in this condition the motor is actually operating in open loop [3].

Besides linear, there are plenty of nonlinear controllers: the fuzzy logic and genetic – based new fuzzy models [4], artificial neural networks [5], adaptive control technique [6], and others.

It is important to make this comparison to find out under what conditions a technique presents a superior performance over the other one and thus have the certainty when it is useful to implement nonlinear controllers, which have greater complexity [7].

Modelling a nonlinearity is often a very complicated challenge. One of the first steps in the synthesis of a control system is to create a mathematical model, because it saves time and it brings the cost-effectiveness.

The main objective of this research is the development and later implementation of a nonlinear control system, by the feedback linearization method, for a laboratory installed DC motor, SRV02 Rotary Servo Base Unit, which has been considered as a single-input-single-output (SISO) system.

Feedback linearization is an approach to nonlinear control design which has attracted a great deal of research interest in recent years. By a combination of a nonlinear transformation and state feedback (feedback linearization), the nonlinear control design is reduced to designing a linear control law [8]. The central idea of the approach is to algebraically transform a nonlinear system dynamics into a (fully or partly) linear one, so that linear control techniques can be applied. This differs entirely from conventional linearization in that feedback linearization is achieved by exact state transformations and feedback, rather than by linear approximations of the dynamics [9]. This technique has been successfully implemented in many applications of control, such as industrial robots, high performance aircraft, helicopters and biomedical dispositifs, more tasks used the methodology are being now well advanced in industry [10].

II. Linear Model of System Dynamics

Constructing an accurate model is a pivotal stage in practical control problems. An appropriately developed system model is essential for reliability of the designed control. A DC series motor is an example of a simple, controlled process that can serve as a vehicle for the evaluation of the performance of the various controllers [4].

A schematic diagram of the DC motor is given in Fig. 1.

Mitra Vesović is with the Faculty of Mechanical Engineering, University of Belgrade, Kraljice Marije 16, 11120 Belgrade, Serbia (e-mail: rjovanovic@mas.bg.ac.rs)

Radiša Jovanović is with the Faculty of Mechanical Engineering, University of Belgrade, Kraljice Marije 16, 11120 Belgrade, Serbia (e-mail: mvesovic@mas.bg.ac.rs)

Lara Laban is with the Faculty of Mechanical Engineering, University of Belgrade, Kraljice Marije 16, 11120 Belgrade, Serbia (e-mail: llaban@mas.bg.ac.rs)

Vladimir Zarić is with the Faculty of Mechanical Engineering, University of Belgrade, Kraljice Marije 16, 11120 Belgrade, Serbia (e-mail: vzaric@mas.bg.ac.rs)

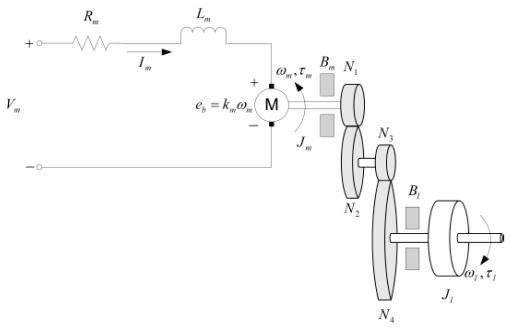


Fig. 1. SRV02 DC motor armature circuit and gain train [11]

The equations that describe the motor electrical components are as follows:

$$V_m(t) = R_m I_m(t) + L_m \frac{dI_m(t)}{dt} + e_b(t) \tag{1}$$

$$e_b(t) = k_m \omega_m(t) \tag{2}$$

where V_m, e_b, k_m and ω_m are motor voltage, back electromotive voltage, back electromotive voltage constant and speed of the motor shaft, respectively. Since the motor inductance L_m is much less than its resistance R_m, it can be ignored [11]. Solving the system of equations for motor current I_m, we get an electrical equation of DC motor:

$$I_m(t) = \frac{V_m(t) - k_m \omega_m(t)}{R_m}. \tag{3}$$

The linear model can be obtained using the Second Newton's Law of Motion and connection between moment of inertia of the load J_l and of the motor shaft J_m, speed of the load shaft ω_l, viscous friction acting on the motor shaft B_m and on the load shaft B_l, total torque applied on the load τ_l and on the motor τ_m, with resulting torque acting on the motor shaft from the load torque denoted as τ_{ml}:

$$J_l \frac{d\omega_l(t)}{dt} + B_l \omega_l(t) = \tau_l(t) \tag{4}$$

$$J_m \frac{d\omega_m(t)}{dt} + B_m \omega_m(t) + \tau_{ml}(t) = \tau_m(t) \tag{5}$$

so the mechanical equation is:

$$J_{eq} \frac{d\omega_l(t)}{dt} + B_{eq} \omega_l(t) = \eta_g K_g \tau_m(t) \tag{6}$$

where J_{eq} and B_{eq} are total moment of inertia and damping term. η_g and K_g are, respectively, the gearbox efficiency and the total gear ratio.

Combining electrical and mechanical equations, assuming that motor torque is proportional to the voltage, the final equation becomes:

$$\left(\frac{d}{dt}\omega_l(t)\right) J_{eq} + B_{eq,v} \omega_l(t) = A_m V_m(t) \tag{7}$$

where the equivalent damping term is given by:

$$B_{eq,v} = \frac{\eta_g K_g^2 \eta_m k_t k_m + B_{eq} R_m}{R_m} \tag{8}$$

where the η_m and k_t are the motor efficiency and the current-torque constant respectively. The actuator gain equals:

$$A_m = \frac{\eta_g K_g \eta_m k_t}{R_m} \tag{9}$$

Linear mathematical model that defines the relationship between voltage and angular position of the load shaft θ_l is:

$$J_{eq} \ddot{\theta}_l(t) + B_{eq,v} \dot{\theta}_l(t) = A_m V_m(t). \tag{10}$$

Choosing $y = \theta_l$ as output variable and $u = V_m$ as input signal, state equation of the system is obtained as follows:

$$J_{eq} \ddot{y}(t) + B_{eq,v} \dot{y}(t) = A_m u(t). \tag{11}$$

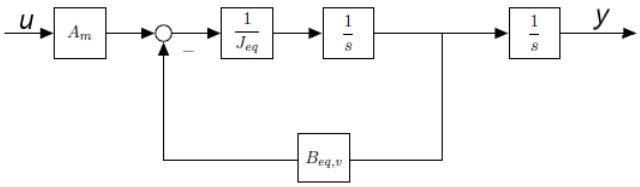


Fig. 2. Block diagram of a linear system

III. EXPERIMENTAL VERIFICATION OF THE OBTAINED LINEAR MATHEMATICAL MODEL

Responses of the system represented with the block diagram in the Fig. 2 are shown in the Fig. 3 and Fig. 4. After recording the responses of the object, comparisons were made with the responses obtained by simulations of the linear model, for step and sinusoidal inputs.

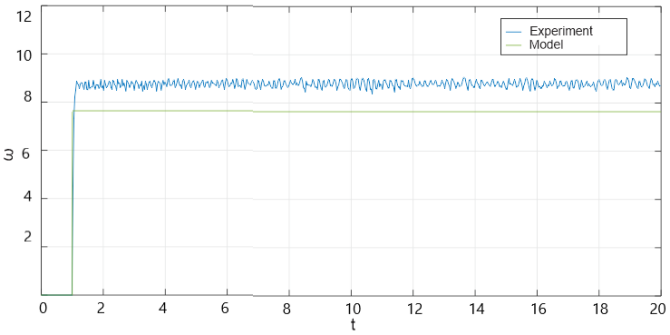


Fig. 3. Experimental results: comparison between real and model data for step input

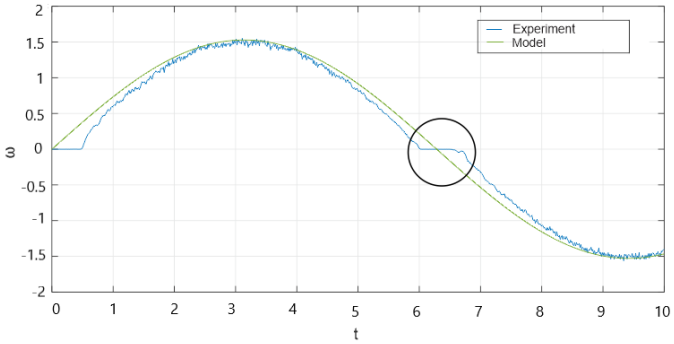


Fig. 4. Experimental results: comparison between real and model data for sinusoidal input

From this simulated example, an important conclusion can be drawn. Simulated linear model of the plant does not match well response of the real system. It is obvious that mathematical model of the series DC motor is nonlinear.

IV. Feedback Linearization

In this section, the conditions for the linearizing transformation and nonlinear feedback allowing the DC motor to be controlled are outlined. Of particular interest will be the coordinate transformation also known as diffeomorphism, and the feedback law which will allow it to be accomplished.

Feedback linearization approach differs from the classical linearization (about the desired equilibrium point) in that no approximation is used; it is exact. Exactness, however, assumes perfect knowledge of the state equation and uses that knowledge to cancel the nonlinearities of the system. Since perfect knowledge of the state equation and exact mathematical cancellation of terms are almost impossible, the implementation of this approach will almost always result in a close-loop system, which is a perturbation of a nominal system whose origin is exponential stable. The validity of the method draws upon Lyapunov theory for perturbed systems [12] (that can be further studied in Chapter 9 of literature [12]).

Consider the single – input – single – output nonlinear SISO system [12]:

$$\begin{aligned} \dot{\boldsymbol{x}} &= \boldsymbol{f}(\boldsymbol{x}) + g(\boldsymbol{x})u \\ y &= h(\boldsymbol{x}) \end{aligned} \tag{12}$$

where $\boldsymbol{f(x)}$, $g(\boldsymbol{x})$ and $h(\boldsymbol{x})$ are sufficiently smooth in a domain $D \subset R^n$ (the mapping $f : D \longrightarrow R^n$, $g : D \longrightarrow R^n$ are vector fields on D) and $\dot{\boldsymbol{x}} = [x_1\ x_2\ \dots\ x_n\ ]^T$ is a state vector. It is necessary to find a state feedback control u, that transforms the nonlinear system into an equivalent linear system. Clearly, generalization of this idea is not possible in every nonlinear system: there must be a certain structural property that allows performing in such a manner of cancellation.

Using feedback to cancel nonlinearities requires the nonlinear state equation to have a structure:

Definition [12]:

$$\dot{\boldsymbol{x}} = \boldsymbol{A}\boldsymbol{x} + B\gamma(\boldsymbol{x})[u - \alpha(\boldsymbol{x})] \tag{13}$$

where $\boldsymbol{A}$ is $n \times n$ and B is $n \times p$ matrix, the functions $\alpha : R^n \longrightarrow R^p$, $\gamma : R^n \longrightarrow R^{p \times p}$ are defined on domain $D \subset R^n$ that contains the origin. Furthermore, two conditions must be satisfied. The first one is that the pair (A, B) must be controllable. The second one is that $\gamma(\boldsymbol{x})$ must be nonsingular for all $\boldsymbol{x} \in D$. This is consequence of the control law form: $u = \alpha(\boldsymbol{x}) + \frac{1}{\gamma(\boldsymbol{x})} v$ that provides a new control signal v.

Even if the state equation does not have the structure (13), sometimes it is possible to execute feedback linearization for another choice of variables. Therefore, a more comprehensive definition is given [12]:

A nonlinear system:

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}) + G(\boldsymbol{x})\, u \tag{14}$$

where $f : D \longrightarrow R^n$ and $G : D \longrightarrow R^{n \times p}$ are sufficiently smooth on a domain $D \subset R^n$, is said to be feedback linearizable (or input – state linearizable) if there exist a diffeomorphism $T : D \longrightarrow R^n$ such that $D_z = T(D)$ contains the origin and the change of variables $\boldsymbol{z} = T(\boldsymbol{x})$ transforms the system (14) into the form:

$$\dot{\boldsymbol{z}} = A\boldsymbol{z} + B\gamma(\boldsymbol{x})[u - \alpha(x)] \tag{15}$$

with (A,B) controllable and $\gamma(\boldsymbol{x})$ nonsingular for all $\boldsymbol{x} \in D$.

V. Determination of Relative Degree

The relative degree of a linear system is defined as the difference between the poles (degree of the transfer function's denominator polynomial number) and zeros (degree of its numerator polynomial). To extend this concept to nonlinear systems more mathematical treatment will be needed. The following definition is given and repeated here for completeness:

Definition [13]: Given the Single Input – Single Output System, SISO, outlined in (12), it is said to have relative degree r at a point x_0 if:

i) $L_{\boldsymbol{g}}\, L_{\boldsymbol{f}}^{k}\, h(\boldsymbol{x}) = 0$ for all x in a neighborhood of x_0 and all $k < r - 1$

ii) $L_{\boldsymbol{g}}\, L_{\boldsymbol{f}}^{r-1}\, h(\boldsymbol{x}) \neq 0$

The terms $L_{\boldsymbol{g}}$ and $L_{\boldsymbol{f}}^{k}$ represent the Lie derivative of $h(x)$ taken along $\boldsymbol{g}(x)$ and $k - times$ along $\boldsymbol{f}(x)$, respectively.

VI. Nonlinear Mathematical Model

The nonlinear mathematical model of the DC motor was obtained considering the speed dependent friction nonlinearity. Many models of friction, widely studied in the literature, differ mainly in the description of the moment of friction. These models, generally, describe the friction torque as a static and/or dynamic function of angular velocity [14]. Here, as well as in [14], Tustins friction model was adopted as follows: $T_{frict} = T_{stribeck} + T_{viscous} = T_c\ \mathrm{sgn}(\dot{\theta}_l) + (T_s - T_c)e^{\left(\frac{\dot{\theta}_l}{\dot{\theta}_s}\right)}\mathrm{sgn}(\dot{\theta}_l) + B\,\dot{\theta}_l$,

where B is the viscous friction coefficient and $\dot{\theta}_s$ is Stribeck velocity. It includes viscous friction part $T_{viscousv}$ and Stribeck function $T_{stribeck}$ that is a decreasing function in relation to the velocity increase and with upper bound equal to the static friction torque T_s, at zero velocity, and lower bound equal to

the Coulomb friction torque T_c. In this approach, the constant portion of the Coulomb model is replaced by Stribeck function. The viscous component of friction torque is a linear function, and friction curve of Stribeck model is nonlinear function, and they will be considered separately. Therefore, the nonlinear mathematical model of the DC motor is addopted as follows:

$$J_{eq}\ddot{\theta}_l + T_{st}(\dot{\theta}_l) + B_{eq,v}\,\dot{\theta}_l = A_m V_m \qquad (16)$$

TABLE I
THE NUMERICAL VALUES OF THE PLANT PARAMETERS

Parameters	Values and units
Jeq	0.0021 kgm^2
Rm	2.6 Ω
kt	0.0077 Nm/A
ηm	0.69
ηg	0.9
Kg	70

In order to identify friction for the above described DC motor, two distinct experiments, cited here, were performed in paper [14]. In the first, the control voltage is increased gradually at the rate of 0.05 V/s and at the instant when the motor shaft starts to rotate, control voltage is recorded. Ten measurements were done and by these values averaging, the static friction torque was obtained. From (16), it can be seen that if the velocity is kept constant, the friction torque is proportional to the control signal V_m. In the second experiment, a linear PI control algorithm was used to stabilise angular velocity to various constant values, and after transient time, the average values of the control voltage and the angular velocity are calculated and recorded. The part of the obtained friction curve $T_{st}(\dot{\theta}_l)$, for low angular velocity values, where the Stribeck effect is dominant, is shown in Fig. 5. It is assumed that friction characteristics are symmetrical, for negative and positive values of angular velocity. Applying standard optimization techniques with Matlab, the friction parameters were obtained, as follows:

$$T_{st} = 0.0174 sgn(\dot{\theta}_l) + 0.0087 e^{-\frac{\dot{\theta}_l}{0.064}} sgn(\dot{\theta}_l),\ B_{eq,v} = 0.0721 \qquad (17)$$

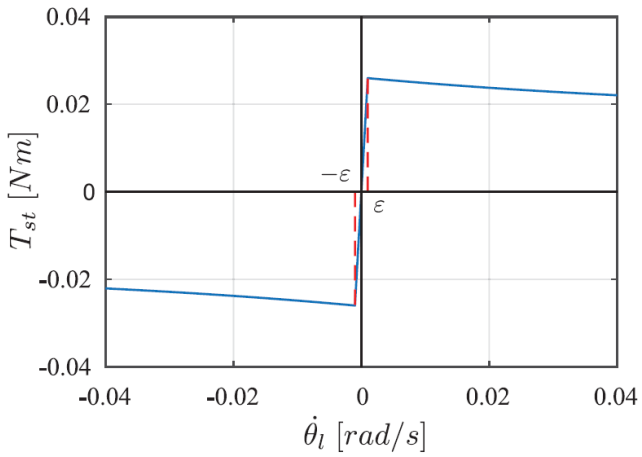


Fig. 5. Friction characteristics of DC motor [14]

In order to overcome the jump discontinuity of the proposed friction model, at $\dot{\theta}_l = 0$, that jump is replaced by a line of finite slope, up to a very small threshold ε, as is shown in Fig. 5 [14]. The slope is bounded by red dashed lines defined by this threshold.

This the line of finite slope will be used only for comparison with the hyperbolic tangent function (Fig. 6), because method of feedback linearization requires differentiable functions (as can be seen from the given definitions in the previous section). In this way only Coulomb and viscous friction is modeled and static friction is neglected. Choosing $x_1 = \theta_l\,, x_2 = \dot{\theta}_l$ as state variables, $y = \theta_l$ as measured variable and $u = V_m$ as control variable and denoting nonlinearity by $f(\boldsymbol{x})$, state equation of the system was obtained as follows:

$$\dot{x} = \begin{bmatrix}\dot{x}_1\\ \dot{x}_2\end{bmatrix} = \begin{bmatrix}0 & 1\\ 0 & \frac{-B_{eq,n}}{J_{eq}}\end{bmatrix}\boldsymbol{x} + \begin{bmatrix}0\\ -1\end{bmatrix} f(\boldsymbol{x}) + \begin{bmatrix}0\\ \frac{A_m}{J_{eq}}\end{bmatrix} u \qquad (18)$$

$$y = [1 \quad 0]\boldsymbol{x} \qquad (19)$$

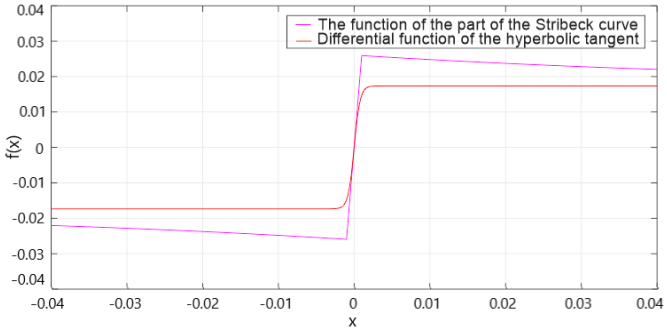


Fig. 6. Differential function of the hyperbolic tangent

To ensure that this model is an equivalent representation of the original system, an experiment was performed, with the results shown below on Fig. 7 for step and Fig. 8 for sinusoidal response.

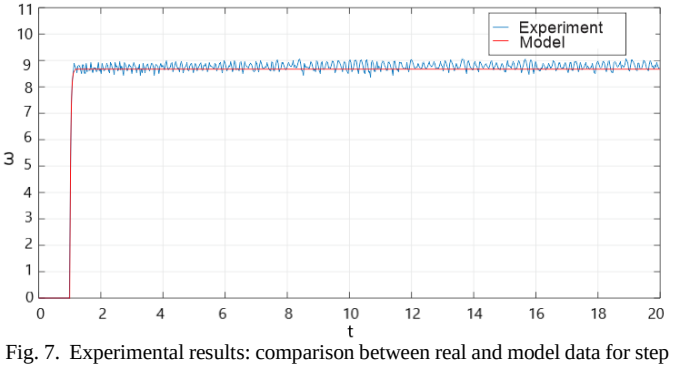


Fig. 7. Experimental results: comparison between real and model data for step input

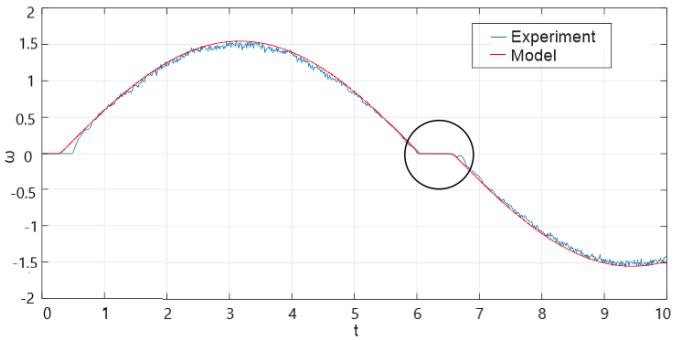


Fig. 8. Experimental results: comparison between real and model data for sinusoidal input

VII. EXPERIMENTAL RESULTS

Applying Definition [12] to the system (18) – (19) yields:

$$A = \begin{bmatrix} 0 & 1 \\ 0 & -\frac{B_{eq,n}}{J_{eq}} \end{bmatrix} \tag{20}$$

$$B = \begin{bmatrix} 0 \\ \frac{A_m}{J_{eq}} \end{bmatrix} \tag{21}$$

$$\alpha(x) = \frac{J_{eq}}{A_m} f(x) \tag{22}$$

$$\gamma(x) = 1. \tag{23}$$

First condition is met:

$$U = [B \quad AB \quad A^2B \quad \dots \quad A^{n-1}B]. \tag{24}$$

Order of system is $n = 2$ and, because *rank U =n*, the pair *(A, B)* is controllable:

$$U = [B \quad AB] = \begin{bmatrix} 0 & \frac{A_m}{J_{eq}} \\ \frac{A_m}{J_{eq}} & -\frac{B_{eq,n}}{J_{eq}^2} \end{bmatrix}. \tag{25}$$

System transformation is not required and all functions are smooth and differentiable. γ(x) is not equal to zero, so the second condition is also met. With both conditions fulfilled feedback linearization is allowed.

The first derivative of the system (18) – (19) output does not depend on the control signal, which means that the relative degree of the system is not 1:

$$\dot{y} = L_f\, h(\boldsymbol{x}) + L_g h(\boldsymbol{x})\, u. \tag{26}$$

$$L_g\, h(\boldsymbol{x}) = 0 \text{ and } L_f\, h(\boldsymbol{x}) = x_2 \tag{27}$$

$$\ddot{y} = \dot{x}_2 = -\frac{B_{eq,n}}{J_{eq}} x_2 - f(\boldsymbol{x}) + \frac{A_m}{J_{eq}}\, u \tag{28}$$

$$L_f^2 h(\boldsymbol{x}) = -\frac{B_{eq,n}}{J_{eq}} x_2 - f(\boldsymbol{x}) \tag{29}$$

$$L_g\, L_f\, h(\boldsymbol{x}) = \frac{A_m}{J_{eq}} \tag{30}$$

Conclusion is that relative degree of this system is equal to the system order $r = 2$. The desired time – domain specifications for controlling the position of the load shaft are: overshoot: $PO = 0\%$ and settling time: $t_s \leqslant 2.3\ s$. Choosing the control signal u in the following form:

$$\begin{aligned} u &= \frac{1}{L_g\, L_f\, h(\boldsymbol{x})}\left[-L_f^2 h(\boldsymbol{x}) + v\right] \\ &= \frac{J_{eq}}{A_m}\left[\frac{B_{eq,n}}{J_{eq}} x_2 + f(\boldsymbol{x}) + v\right] \end{aligned} \tag{31}$$

with $v = -K_0 x_1 - K_1 x_2 + K_0 x_{ref}$, where $K_0 = 400$, $K_1 = 40$ were obtained by calculating the minimum damping ratio and natural frequency, which were required to meet the specifications; x_{ref} is desired output or reference. Linear control is obtained in the same way, with the same coefficients, but without canceling the nonlinearity:

$$u_l = -K_0 x_1 - K_1 x_2 + K_0 x_{ref} \tag{32}$$

The experiments were performed with Quanser rotary servo motor, SRV02. This model is equipped with the optical encoder and tachometer, for motor position and speed measuring, respectively [14].

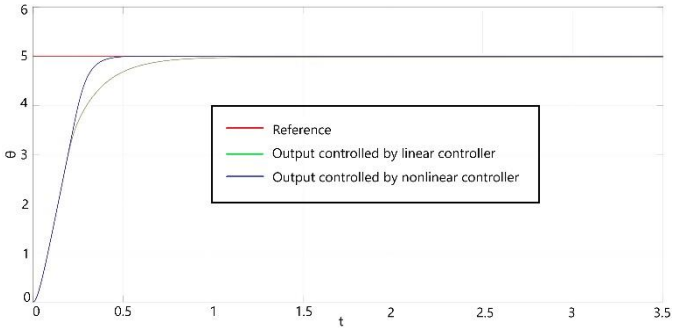


Fig. 9. Experimental results: position tracking of step signal for the linear and nonlinear controller

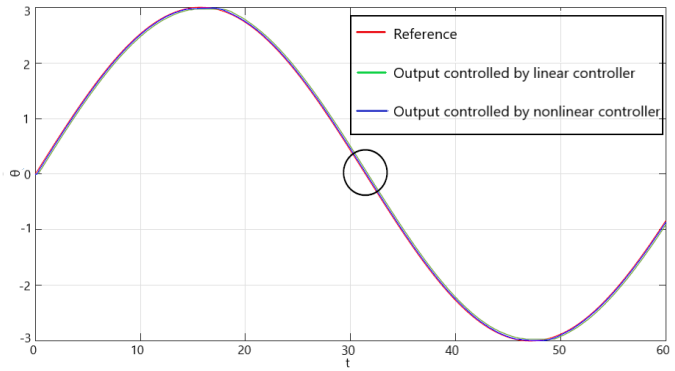


Fig. 10. Experimental results: position tracking of sine signal for the linear and nonlinear controller

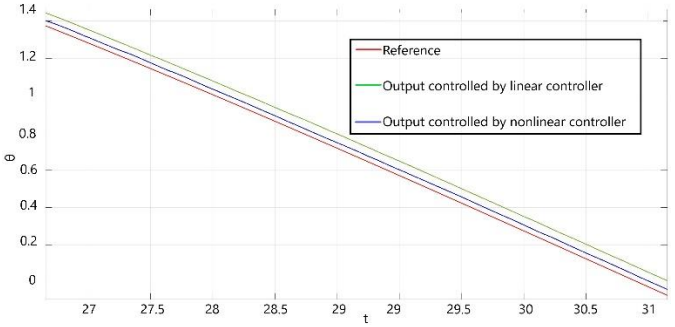


Fig. 11. Detail from Fig. 10.

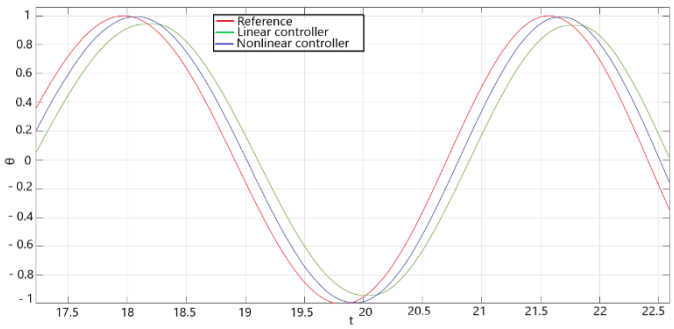


Fig. 12. Experimental results: position tracking of chirp signal for the linear and nonlinear controller

It can be observed, from the Fig. 9, Fig. 10, Fig. 11 and Fig. 12 that the specific requirements are met. The overshoot and the settling time are in the domain of desired values. Furthermore, it is observed that the nonlinear controller is more convenient and has better achievements for position management.

VIII. Conclusion

The feedback linearization technique was used for controlling the nonlinear system. The primary aim was to corroborate this method for controlling position of DC motor. First, the modelling of an object has been obtained.

After it has been experimentally confirmed that linear equations did not describe this object well enough, the nonlinear model was presented by including Stribeck model of the friction. Using the concise presentation of the feedback theory the conditions for accomplishing this technique were considered. In order to satisfy those conditions an approximation of the function, which represent nonlinearity, was found as hyperbolic tangent. Then the fulfillment of the conditions for the synthesis of the control law was proven.

At the end it could be observed, through the experiment and analysis results, that the desired response (output signal of the model reference) was tracked by the plant response. The comparison of the linear and nonlinear controller is given. The results show that the controllers, synthesized in this way, are able to satisfy desired position, but that nonlinear controller gives better outcome.

Acknowledgment

This research was supported by the Science Fund of the Republic of Serbia, grant No. 6523109, AI- MISSION4.0, 2020-2022

This paper was conceived within the research on the project: “Integrated research in the field of macro, micro and nano mechanical engineering - Deep machine learning of intelligent technological systems in production engineering”, The Ministry of Education, Science and Technological Development of the Republic of Serbia (contract no. 451-03 - 68 / 2020-14 / 200105), 2020.

This work was financially supported by the Ministry of Education, Science and Technological Development of the Serbian Government, Grant TR-35029 (2018-2020).

References

[1] S. K. Sarkar, S. K. Das, “High Performance Nonlinear Controller Design for AC and DC Machines: Partial Feedback Linearization Approach”, International Journal of Dynamics and Control, vol. 6, pp. 679 – 693, May, 2017.

[2] T. Kara, I. Eker, “Nonlinear modeling and identification of a DC motor for bidirectional operation with real time experiments,” Energy Conversion and Management, vol. 45, pp. 1087–1106, 2004.

[3] J. U. Liceaga-Castro, I. I. Siller-Alcalá, J. Jaimes-Ponce, R. A. Alcántara-Ramírez, E. A. Zamudio, “Identification and Real Time Speed Control of a Series DC Motor,” Mathematical Problems in Engineering, vol. 2017, March 2017.

[4] D. Park, A. Kandel, G. Langholz Author, “Genetic-Based New Fuzzy Reasoning Models with Application to Fuzzy Control,” IEEE Transactions on Systems, Man, and Cybernetics, vol. 24, no. 1. January 1994.

[5] S. Valluru, N. Singh, M. Kumar, “Implementation of NARMA-L2 neuro controller for speed regulation of series connected DC motor,” IEEE 5th India International Conference on Power Electronics (IICPE), Delhi, India, pp. 1 – 7, 2012.

[6] T. Furuhashi, S. Sangwongwanich, S. Okuma, “A Position and Velocity Sensorless Control for Brushless DC Motors Using an Adaptive Sliding Mode Observer,” IEEE Transactions on Industrial electronics, vol. 39, no. 2, pp. 89 – 95, 1992.

[7] C. Rengifo, N. Casas, D. Bravo, “A performance comparison of nonlinear and linear control for a DC series motor,” Revista Ciencia en Desarrollo, vol. 8, May 2017.

[8] S. Metha, J. Chiasson, “Nonlinear Control of a Series DC Motor: Theory and Experiment,” Proceedings of the American Control Conference, Albuquerque, New Mexico, June, 1997.

[9] J. E. Slotine, W. Li, “Feedback linearization,” in Applied Nonlinear Control, Englewood Cliffs, New Jersey, United States: P.H, 1991, ch. 6, sec. x, p. 207.

[10] W. Ghozlane, J. Knani, “Nonlinear Control via Input-Output Feedback Linearization of a Robot Manipulator,” Advances in Science, Technology and Engineering Systems Journal, vol. 3, October, 2018.

[11] J. Apkarian, M. Levis, H. Gurocak, Student Workbook, SRV02 Base Unit Experiment For Matlab/Simulink Users

[12] H. K. Khalil, Nonlinear Systems; 3rd ed. New Jersey, United States: Prentience Hall, 2002

[13] A. Isidori, Nonlinear Control Systems, 3rd ed. London, United Kingdom: Springer, 2002.

[14] L. T. Gruyitch, Z. M. Bučevac, R. Ž. Jovanović & Z. B. Ribar (2019): Structurally variable control of Lurie systems, International Journal of Control

Modeling and Control of a Liquid Level System Based on the Takagi-Sugeno Fuzzy Model Using the Whale Optimization Algorithm

Radiša Jovanović, Vladimir Zarić, Mitra Vesović and Lara Laban

Abstract—**The liquid level control remains an important task for research and is used by process control engineers. Firstly, the linear models for the tank system are obtained for the three different operating points. From these identified linear models a Takagi-Sugeno (TS) model is obtained using triangular membership functions in the premises of the rules. Furthermore, the whale optimization algorithm is implemented to fine tune parameters of Takagi-Sugeno fuzzy model, according to the chosen objective function. By using the parallel distributed compensation (PDC), a fuzzy controller is created by the fuzzy blending of three PI controllers designed for each of the operating points. In order to evaluate performance of the PDC based fuzzy controllers, the comparison is made between several local linear PI controllers and the PDC. Moreover, the PDC controllers from the optimized and original TS plant model are compared. The experimental results and the comparison results verificate efficiency of the proposed method.**

Index Terms—**Takagi-Sugeno; liquid level control; parallel distributed compensation; whale optimization algorithm.**

I. Introduction

The liquid level control has a wide range of applications in the process industries such as petro-chemical, waste water treatment and purification, biochemical, spray coating, beverages and pharmaceutical industries.

In [1] authors have conveyed and stressed the issue of performance analysis of three control schemes for couple tank system, PI (based on pole placement, Ziegler Nichols and Ciancone correlation tuning methods), PI-plus-feedforward and model predictive control. Moreover, paper [2] addresses the nonlinear control design problem for a liquid level system. A model-based backstepping controller and an adaptive backstepping controller are developed for the liquid level system. Following, the article [3] dabbles with the fuzzy-PID controller applied to the nonlinear dynamic model of the liquid level of the coupled tank system, all the while taking into account the effects of noise. The fuzzy model proposed by Takagi and Sugeno [4] is described by fuzzy IF-THEN rules which depict local linear input-output relations of a nonlinear system. Fuzzy logic has many varieties that can be implemented for control purposes. For instance, one of them is parallel distributed compensation (PDC). The PDC offers a chance to use a technique to design a fuzzy controller from a given TS fuzzy model. In paper [5], a fuzzy controller is constructed based on a PDC method and it is implemented in an experimental tank level control system. The paper [6] suggests a procedure used to make two-variable fuzzy logic controllers (FLCs) set for the levels in a laboratory coupled-tank system. The plant input and output experimental data are then used for derivation via genetic algorithms optimization of a Takagi-Sugeno-Kang (TSK) plant model needed for FLC improvements. The TSK model is validated on a different set of experimental data and used in designing of two variable linear proportional-plus-integral PI controller and PDC with local linear PI controllers. In [7] a novel modification to the original PDC method is submitted, so that, besides the stability issue, the closed-loop performance of the system can be considered at the design stage. The strong point is that, for example, a faster response can be obtained, for a given bound on the control input. The following paper [8] gave a unified approach to a nonlinear model following control that contains the regulation and servo control problems as distinctive cases. A PDC for fuzzy reference models was proposed. As a result of the following paper [9] a captivating method that improves the quality of robust control by interpolating a robust and optimal controller is presented. That paper introduces a new method called advanced robust parallel distributed compensation (ARPDC) for automatic control of nonlinear systems.

The fuzzy design can be considered as an optimization problem, where the structure, antecedent, and consequent parameters are required to be identified. Metaheuristic methods as global optimization algorithms can deal with non-convex, nonlinear, and multimodal problems subjected to linear or nonlinear constraints with continuous or discrete decision variables. The synergy of fuzzy models and nature-inspired optimization algorithms belongs to the actual trends in soft computing, where all individual contributing technologies are seamlessly structured together. Attractive points of view on this combination are treated in the literature [10]. Recently, several metaheuristic methods have been proposed. Some of them include the genetic algorithm (GA) [11], particle swarm optimization (PSO) [12]-[15], gray wolf optimization (GWO) [16], whale optimization algorithm (WOA) [17] and ant colony optimization algorithm (ACO) [18]. Notwithstanding, the paper [19] explores the potentiality of a bat algorithm for tuning the

Radiša Jovanović is with the University of Belgrade, Faculty of Mechanical Engineering, Kraljice Marije 16, 11020 Belgrade, Serbia (e-mail: rjovanovic@ mas.bg.ac.rs).

Vladimir Zarić is with the University of Belgrade, Faculty of Mechanical Engineering, Kraljice Marije 16, 11020 Belgrade, Serbia (e-mail: vzaric@ mas.bg.ac.rs).

Mitra Vesović is with the University of Belgrade, Faculty of Mechanical Engineering, Kraljice Marije 16, 11020 Belgrade, Serbia (e-mail: mvesovic@ mas.bg.ac.rs).

Lara Laban is with the University of Belgrade, Faculty of Mechanical Engineering, Kraljice Marije 16, 11020 Belgrade, Serbia (e-mail: llaban@ mas.bg.ac.rs).

PID controllers. A modified WOA (MWOA) is used to tune the AFPID (adaptive fuzzy logic PID) parameters and showed improved performance when compared with conventional PID [14].

In this study, the structure and consequent parameters are known (number of rules, shapes of input membership functions and linear models in the consequent part of the rules), and antecedent parameters are determined using the whale optimizing algorithm.

II. System Model

The plant consisting of a pump integrated with a water basin and the tank as shown on Fig. 1. A practical industrial applications of such plant can be found in the processing system of petro-chemical, paper making, and/or water treatment plants, to name a few.

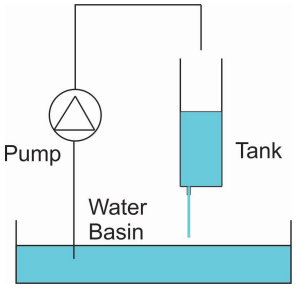


Fig. 1. The liquid level system.

A. Mathematical modeling

The input into the process is the voltage to the pump V_p and its output is the water level in tank, H. The volumetric inflow rate to tank, Q_i, is supposed to be directly proportional to the applied pump voltage, Q_i=KV_p. Applying Bernoulli's equation for small orifices, the outflow velocity from tank, V_o, can be expressed by a succeeding relationship

$$V_O = \sqrt{2gH}\,, \quad Q_O = A_O V_O\,, \tag{1}$$

where A_o is an area of the outlet orifice, while Q_o is the outflow rate. In acquiring the tank equation of motion the mass balance principle can be applied to the water level in tank, i.e.

$$A_t \dot{H} = Q_i - Q_O = KV_p - A_O V_O = KV_p - A_O\sqrt{2gH}\,, \tag{2}$$

where A_t is the area of tank. The nonlinear differential equation that describes the change in level in tank is

$$\dot{H} = \frac{K}{A_t}V_p - \frac{A_O}{A_t}\sqrt{2gH}\,. \tag{3}$$

B. Takagi Sugeno fuzzy model and identification

The main idea of the TS fuzzy modeling method is to partition the nonlinear system dynamics into several locally linearized subsystems, so that the overall nonlinear behavior of the system could be captured by fuzzy blending of such subsystems. Thus, a fuzzy model and identification of a liquid level system will be implemented in accordance with the TS model containing three rules. The fuzzy rule associated with the i-th linear subsystem, can then be defined as i-th rule:

IF $x(t)$ is M_i THEN

$$\dot{x}(t) = a_i x(t) + b_i u(t), \quad i = 1,2,3, \tag{4}$$

$$y(t) = c_i x(t)\,, \quad i = 1,2,3. \tag{5}$$

Here M_i is the fuzzy set, $x(t)\in\mathbb{R}$ is the state variable, $u(t)\in\mathbb{R}$ is the input, $y(t)\in\mathbb{R}$ is the output variable, $a_i,b_i,c_i\in\mathbb{R}$. In our case, the selected state space variable is equal to the output variable $x(t)$=$y(t)$=$H(t)$.

The overall output, using the fuzzy blend of the linear subsystems, will then be as follows:

$$\dot{x}(t) = \frac{\sum_{i=1}^{3} w_i(x(t))\{a_i x(t) + b_i u(t)\}}{\sum_{i=1}^{3} w_i(x(t))}, \tag{6}$$

$$h_i(x(t)) = \frac{w_i(x(t))}{\sum_{i=1}^{3} w_i(x(t))}, \tag{7}$$

$$\dot{x}(t) = \sum_{i=1}^{3} h_i(x(t))\{a_i x(t) + b_i u(t)\}, \tag{8}$$

$$y(t) = \frac{\sum_{i=1}^{3} w_i(x(t))c_i x(t)}{\sum_{i=1}^{3} w_i(x(t))} = \sum_{i=1}^{3} h_i(x(t))c_i x(t)\,, \tag{9}$$

where $w_i(x(t))$=$M_i(x(t))$ is the grade of membership of $x(t)$ in M_i and $h_i(x(t))$ is normalized weight. The linear models in the consequent rules (4) can be obtained by utilizing an analytical linearization of a non-linear equation. Besides that, another approach is to apply the methods of identification in accordance with the measured input output data. The identification methods were used based on the step response. Since models obtained by identification experimentally turned out to be more of an adequate approximation, in comparison with the analytically obtained linearized models, they were used. Linear models can be represented by following transfer function,

$$G(s) = \frac{H(s)}{V_p(s)} = \frac{K}{\tau s + 1}, \tag{10}$$

where K and τ are tank's gain and time constant, respectively. Nominal levels in the tank H_{Ni}, nominal voltages V_{pNi} and corresponding identified transfer functions are given in Table 1. Voltage deviation represent control deviations so we can write $u(t)$=$v_p(t)$. Constants for the state space plant model a_i and b_i are given in Table 2.

TABLE I
NOMINAL VALUES AND LINEAR MODELS

i	H_{Ni} [m]	V_{pNi} [V]	G_i(s)
1	0.077	4.4	$\frac{0.002313}{s+0.04758}$
2	0.1665	6	$\frac{0.002627}{s+0.04235}$
3	0.2415	7.1	$\frac{0.002642}{s+0.03469}$

TABLE II
CONSTANTS FOR THE STATE SPACE SYSTEM MODEL

i	1	2	3
a_i	-0.04758	-0.04235	-0.03469
b_i	0.002313	0.002627	0.002642

In this article a nonlinear model is obtained by combining three linear models around 0.08 m, 0.16 m and 0.24 m. The membership functions have a triangular shape and are depicted in Fig. 2. Moreover, the predefined parameters are arbitrary function parameters, and it is assumed that they are symmetric.

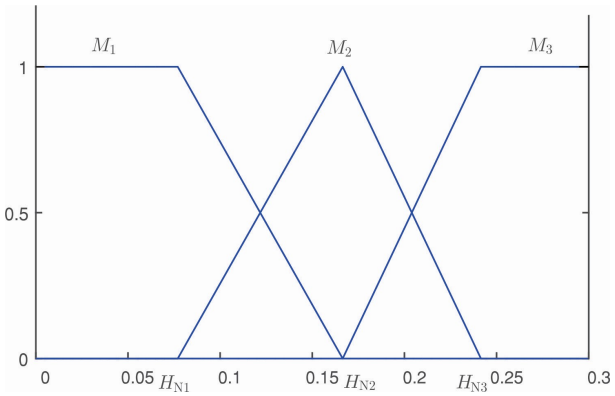


Fig. 2. Membership functions.

III. THE WHALE OPTIMIZER

Whale Optimization Algorithm has proven to be outstanding at resolving a variety of modes, multimodal and problems that are not linear. The foremost supremacies of this algorithm, and all metaheuristic algorithms in general, are that it has random distribution, which avoids getting stuck in the local minimum. WOA was first suggested by Seyedali Mirijalili and Andrew Lewis in [17]. The paper was inspired by a dozen whales, working together in a sophisticated way to harvest the krill. A curtain of bubbles and the hunting horn hold a secret to an indigenous ways of fishing - the bubble net feeding. The leader whale (bubble blower) dives a couple of meters deep into the ocean. It's his job to find the fish. The rest follow information. Each takes exactly the same position in every lunch. Once the leader has located the fish, he blows a net of bubbles in a spiral shape, which completely encircles the prey. Another whale calls to synchronize the group. Panicked by the hearing sound of the blinding bubbles barrier the fish herds will be captured, allowing whales to swim up toward them. The hunt contains three phases. The first one is encircling the prey by defining the best search agent and updating the position of others. The mathematical model of this phase is proposed using the distance vector **D** and vector **X** which is used to update the position:

$$\mathbf{D} = \left| \mathbf{C}\mathbf{X}'(t) - \mathbf{X}(t) \right|, \tag{11}$$

$$\mathbf{X}(t+1) = \mathbf{X}'(t) - \mathbf{A}\mathbf{D}, \tag{12}$$

$$\boldsymbol{A} = 2\boldsymbol{a}\boldsymbol{r} - \boldsymbol{a}, \ \boldsymbol{C} = 2\boldsymbol{r}, \tag{13}$$

where t indicates the current iteration, **A** and **C** indicate coefficient vectors. Adjusting those values improves positions around the best agent ***a***, where ***a*** is linearly decreased from 2 to 0 over the course of iterations and ***r*** is a random vector in [0,1]. **X′** is the position vector of the best solution obtained so far and **X** is the position vector. The second phase-exploration is given either with shrinking encircling mechanism (defining the new position of the searching agent using **A**), or with spiral updating position (first calculation distance between whale and prey using helix-based movement. The new position of the agent is located between the current best agent and the original position. The function for this approach is:

$$\mathbf{X}(t+1) = \begin{cases} \mathbf{X}'(t) - \mathbf{A}\mathbf{D} & \text{if} \quad p < 0.5 \\ \mathbf{D}' - e^{bl}\cos(2\pi l) + \mathbf{X}'(t) & \text{if} \quad p \geq 0.5 \end{cases}, \tag{14}$$

where p is a random number in [0,1], b is a constant for defining the shape of the logarithmic spiral, l is a random number in [-1,1] and $\boldsymbol{D}'$ indicates the distance of the i-th whale from the prey [17]. The third one, exploration phase, is based on a random search, that provides a good balance between the last two phases. This is called adoptive variation that depends of the value search vector **A.**

IV. TAKAGI-SUGENO MODEL OPTIMIZATION

In the Fig. 2 we observe the beforehand mentioned TS model which was obtained based on the symmetric shape of the membership functions. The configuration of the functions is triangular and the centers of the membership functions are located in the selected nominal points in which the linear models are identified. However, in order to achieve a better approximation of the non-linear characteristics and overall behavior of the plant, a more adequate approximation of the non-linear model is presented by adjusting the parameters of the membership functions. We can view the parameters as the width of the membership functions. So in conclusion, in this case we only optimized the parameters that were located in the rule premise. Moreover, the mentioned TS parameters are all coded into one whale, per say one agent, that is presented with a vector which contains the premise parameters, in our case it has four parameters. In the proposed WOA algorithm the population is set to 20, while the total number of iterations is set to 30. The population size and the number of iterations, viewed as a criteria of stopping, are determined based on a series of experiments with different values, all the while taking in account the specificity of our problem which is

that the dimensionality of the problem is small (only 4 unknown parameters). Furthermore, in this optimization method, one agent represents one potential optimal fuzzy model. The mean square error (MSE) is taken as an objective function and it can be calculated as

$$\text{MSE}=\frac{1}{n}\sum_{i=1}^{n}\left(y(i)-y_m(i)\right)^2, \qquad (15)$$

where n is the number of data points, $y(i)$ is the measured output of the plant, $y_m(i)$ is the output of the model.

A dataset for the learning process of the WOA algorithm, in other words for the optimization of the TS model, is obtained from the plant operation in 1600 seconds. All of the parameter values that were used in the implementation of the WOA were taken from the original paper [17]. In the aim of identification we bring the input voltage which has a shape as depicted in Fig. 3.

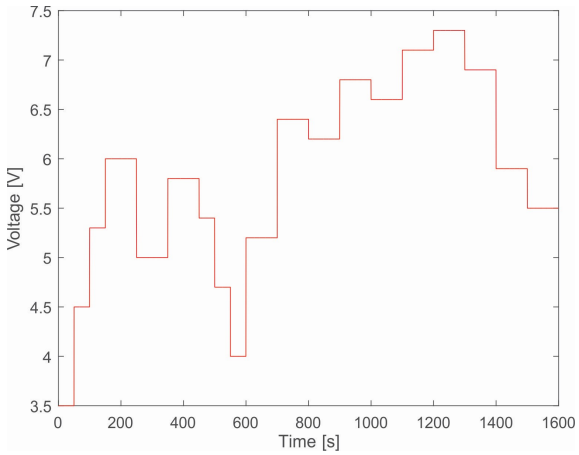


Fig. 3. Voltages used for model optimization.

There it should be observed that the values are between the nominal voltages, this is done in order to cover the range of interest. Optimized membership functions are shown on Fig. 4.

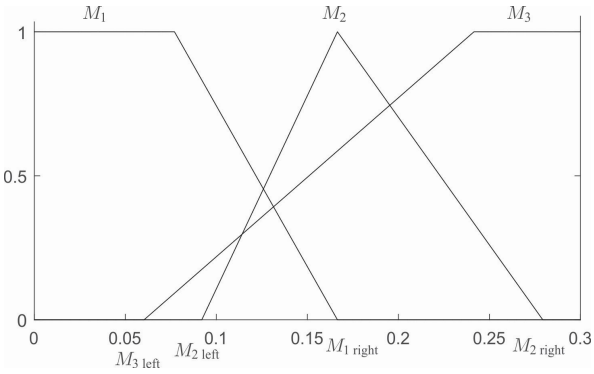


Fig. 4. Optimized membership functions.

where $M_{2\text{left}}$=0.092, $M_{1\text{right}}$=0.1665, $M_{3\text{left}}$=0.0603, $M_{2\text{right}}$=0.2793. Comparison of the TS model based on initial membership functions and the TS model based on optimized membership functions is showed on Fig. 5.

V. CONTROL SYSTEMS DESIGN

The main control objective is to maintain the liquid level in tank at a desired level by adjusting the pump flow rate. The requirement is that the control systems for all three operating points should satisfy the following specifications: the steady state error should be zero; the percentage overshot in tank has to be less than 5%, the PO ≤ 5%; the settling time for the tank should be less than 30 seconds, T_s ≤ 30 sec.

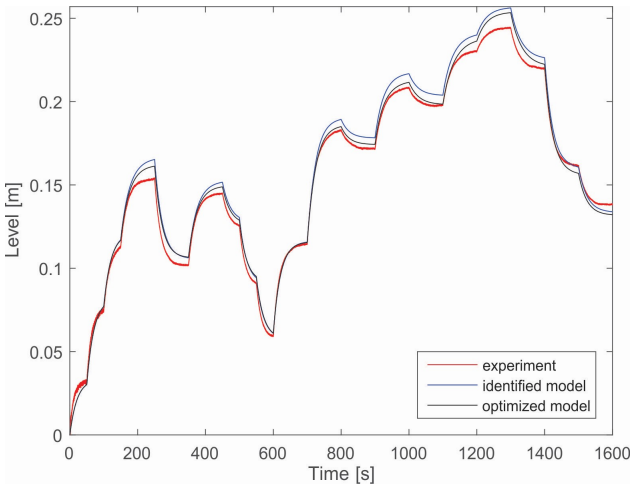


Fig. 5. Comparison of identified and optimized TS model.

The history of the purported PDC was set in motion with a model-based design procedure proposed by Kang and Sugeno, [20]. The PDC proposes a procedure to design a fuzzy controller from a given TS fuzzy model. Furthermore, each control rule is designed from the corresponding rule of a TS fuzzy model during the construction of a PDC design. As a consequence, the designed fuzzy controller shares the same fuzzy sets as the fuzzy model in the premise parts. For our concrete model in this paper we have defined for each of the linearized models a linear PI controller. The control rule i of the fuzzy controller via the PDC is:

IF $x(t)$ is about H_{Ni}, THEN the controller is C_i.

The overall fuzzy controller is represented by

$$C=\frac{\sum_{i=1}^{3}w_i\left(x(t)\right)C_i}{\sum_{i=1}^{3}w_i\left(x(t)\right)}=\sum_{i=1}^{3}h_i\left(x(t)\right)C_i, \qquad (16)$$

where C_i are PI controllers defined in a complex domen as $C_i = K_{Pi} + K_{Ii}/s$, i=1,2,3. Parameters for all three controllers C_i, were obtained based on the linear theory and according to the control objective as can be seen in [2]. Percent overshoot and settling time requirements for the closed-loop system responses can be transformed into the desired natural frequency ω_{ni} and damping coefficient ζ_i. If the i-th plant model is represented by $G_i(s)=\beta_i/(s-\alpha_i)$ then:

$$b_i=\left|\ln\left(\frac{\text{PO}_i}{100}\right)\right|,\ \zeta_i=\frac{b_i}{\sqrt{b_i^2+\pi^2}},\ \omega_{ni}=\frac{4}{T_{si}\zeta_i}, \qquad (17)$$

$$K_{P_i} = \frac{1}{\beta_i}\left(2\zeta_i \omega_{ni} + \alpha_i\right),\ K_{I_i} = \frac{\omega_{ni}^2}{\beta_i},\ i = 1,2,3. \quad (18)$$

Meanwhile, proportional and integral gains for all of the above stated controllers are given in Table 3.

TABLE III
PARAMETERS OF CONTROLLERS

i	1	2	3
$K_{\mathrm{P}i}$	94.72	85.389	87.803
$K_{\mathrm{I}i}$	16.139	14.21	14.129

VI. FURTHER EXPERIMENTAL RESULTS

In order to display the effectiveness of the utilized methods we performed a couple of experiments and verified the efficiency of the optimization and identification, subsequently. Thus, the Fig. 6 depicts the difference between the plant response that is controlled by a local linear PI controller, which is designed to work around 0.08m, and the plant response that is controlled by the PDC. As can be seen in Fig. 6 the PDC achieves a better performance than the local PI because when we are operating in the range of 0.08 m to 0.12 m, both controllers that are designed to operate around 0.08 m and 0.16 m are active, see Fig. 2.

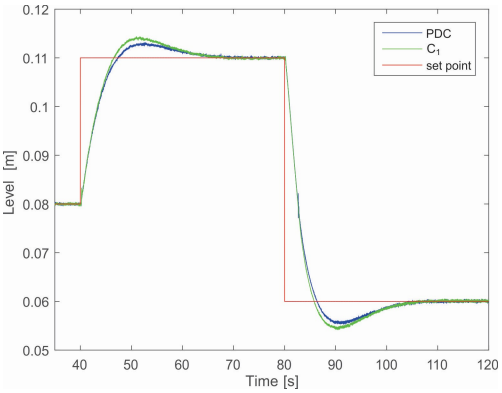


Fig. 6. Comparison of PDC and PI control system around 0.08 m.

The same analysis applies to the operation of the plant around 0.16 m, which is shown in the Fig. 7. In this case with the PDC, all three local linear controllers that are designed to operate around 0.08 m, 0.16 m and 0.24 m are active, as can be seen from the Fig. 2.

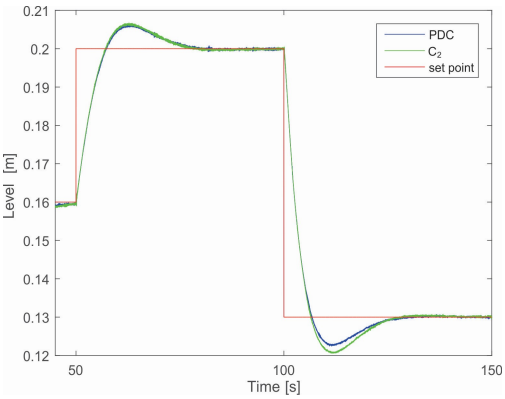


Fig. 7. Comparison of PDC and PI control system around 0.16 m.

The PDC controller was compared with the specifically designed controller for the nominal point 0.12 m. As to say, that the most onerous challenge for the PDC is precisely this, because that point is the most further from the operation points of local linear controllers, which are designed to operate around 0.08 m and 0.16 m.

The requirements for this local linear PI controller are the same. In the same way we obtained parameters $K_{\mathrm{P}i}$=67.384 and $K_{\mathrm{I}i}$=12.081. A juxtapose of the operation of this local PI controller with the PDC is shown in Fig. 8.

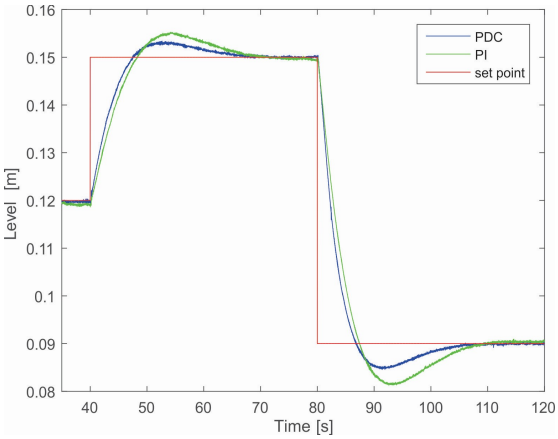


Fig. 8. Comparison of PDC and PI control system around 0.12 m.

A smaller overshoot and settling time, were obtained when the plant was controlled using a PDC that contains information about the optimized model, than when the plant was controlled by a PDC with initial membership functions. In order for our results to be observed better, the filtered responses are shown in Fig. 9. The same moving average filter with a span of 30 data points has been used for both of the signals.

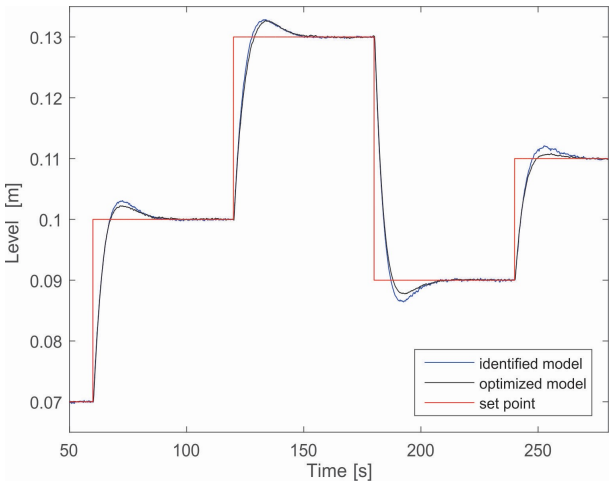


Fig. 9. Comparison of PDCs with an identified and optimized plant model (experiment with moving average filter).

Comparison of control signals of PDCs with an identified and optimized plant model is shown on the Fig.10.

Comparisons of system response percentage overshoots and settling times of system response, for the identified and optimized TS model, are shown in Table 4, Table 5, respectively.

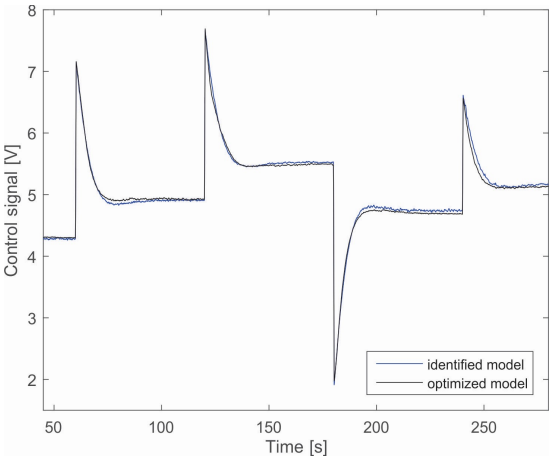


Fig. 10. Comparison of control signals of PDCs with an identified and optimized plant model.

TABLE IV
PERCENTAGE OVERSHOOT FOR DIFFERENT STEP RESPONSES

Step [m]	0.09-0.11	0.13-0.09	0.1-0.13	0.07-0.1
Id. [%]	10	8.75	9.3	10
Op. [%]	4	5.5	8.7	7.3

TABLE V
SETTLING TIME FOR DIFFERENT STEP RESPONSES

Step [m]	0.09-0.11	0.13-0.09	0.1-0.13	0.07-0.1
Id. [s]	26	22.7	25.3	24
Op. [s]	21	21	25	23.6

VII. CONCLUSION

Initially, in this paper, the mathematical model of the liquid level system was obtained experimentally. Further, TS fuzzy model was obtained based on three identified local linear models. Regardless of the superiority in "catching" the nonlinear behavior of the plant, TS model was optimized using WOA metaheuristic and verificated by comparing it with the original. Consequently, based on the given requirements three local linear PI controllers were designed. Then, by using the PDC method, two fuzzy controllers were designed based on identified and optimized TS model. The designed controllers, which were based on the PDC, were implemented in an experimental setup in order to prove their performance. Given the very satisfying results, the developed TS models are tremendously simple and consists only of three fuzzy rules. Using the whale optimization method the accuracy of TS model is improved, which enlarges the efficiency of TS based PDC controller. Future research will focus on exploiting these possibilities in terms of using more fuzzy rules.

ACKNOWLEDGMENT

This research was supported by the Science Fund of the Republic of Serbia, grant No. 6523109, AI-MISSION4.0, 2020-2022. This paper was conceived within the research on the project: "*Integrated research in the field of macro, micro and nano mechanical engineering - Deep machine learning of intelligent technological systems in production engineering*", The Ministry of Education, Science and Technological Development of the Republic of Serbia (contract no. 451-03 - 68 / 2020-14 / 200105), 2020. This work was financially supported by the Ministry of Education, Science and Technological Development of the Serbian Government, Grant TR-35029 (2018-2020).

REFERENCES

[1] A. S. Tijjani, M. A. Shehu, A. Alsabari, Y. A. Sambo, N. L. Tanko, "Performance analysis for coupled-tank system liquid level control using MPC, PI and PI-plus-feedforward control scheme," *Journal of Robotics and Automation*, vol. 1, no. 1, *pp.* 42-53, June, 2017.

[2] H. Pan, H. Wong, V. Kapila, M.S. de Queiroz, "Experimental validation of a nonlinear backstepping liquid level controller for a state coupled two tank system," *Control Engineering Practice*, vol. 13, no. 1, pp. 27-40, January, 2005.

[3] T. L. Mien, "Liquid level control of coupled-tank system using fuzzy-PID controller," *International Journal of Engineering Research & Technology*, vol. 6, no. 11, pp. 459-464, November, 2017.

[4] T. Takagi, M. Sugeno, "Fuzzy identification of systems and its applications to modeling and control," IEEE Transactions on Systems, Man, Cybernetics, New York, New York, vol. 15, pp. 116-132, 1985.

[5] M. S. Sadeghi, B. Safarinejadian, A. Farughian, "Parallel distributed compensator design of tank level control based on fuzzy Takagi-Sugeno model," *Applied Soft Computing*, vol. 21, pp. 280-285, August, 2014.

[6] S. Yordanova, "Fuzzy logic approach to coupled level control," *Systems Science & Control Engineering*, vol. 4, no. 1, pp. 2015-222, September, 2016.

[7] A. D. Marka, M. Seidi, "Performance-oriented parallel distributed compensation," *Journal of the Franklin Institute*, vol. 348, pp.1231-1244, 2011.

[8] T. Taniguchi, K. Tanaka, K. Yamafuji, H. O. Wang, "Nonlinear model following control via Takagi-Sugeno fuzzy model," Proceedings of the American Control Conference, San Diego, California, vol. 3, pp. 1837-1841, 1999.

[9] M. Polanský, "Advanced robust PDC fuzzy control of nonlinear systems," *International Journal of Computer and Information Engineering*, vol. 1, no.11, pp. 3715-3720, 2007.

[10] F. S. Gharehchopogh, H. Gholizadeh, "A comprehensive survey: Whale optimization algorithm and its applications," *Swarm and Evolutionary Computation*, vol. 48, pp. 1-24, August, 2019.

[11] O. Cordón, F. Herrera, "A two-stage evolutionary process for designing TSK fuzzy rule-based systems," *IEEE Transactions on Systems, Man, and Cybernetics - Part B: Cybernetics*, vol. 29, no. 6, pp. 703–715, December, 1999.

[12] S. H. Tsai, Y. W. Chen, "A novel identification method for Takagi-Sugeno fuzzy model," *Fuzzy Sets and Systems*, vol. 338, pp. 117-135, May, 2018.

[13] R. E. Precupa, M. C. Sabaua, E. M. Petriub, "Nature-inspired optimal tuning of input membership functions of Takagi-Sugeno-Kang fuzzy models for Anti-lock Braking Systems," *Applied Soft Computing*, vol. 27, pp. 575-589, 2015.

[14] L. Lin, F. Guo, X. Xie, B. Luo, "Novel adaptive hybrid rule network based on TS fuzzy rules using an improved quantum-behaved particle swarm optimization", *Neurocomputing*, vol. 149, pp. 1003-1013, February, 2015.

[15] S. Rastegar, R. Araújo, J. Mendes, "Online identification of Takagi–Sugeno fuzzy models based on self-adaptive hierarchical particle swarm optimization algorithm," *Applied Mathematical Modelling*, vol. 45, pp. 606-620, May, 2017.

[16] M. Ghanamijaber, "A hybrid fuzzy-PID controller based on gray wolf optimization algorithm in power system," *Evolving Systems*, vol. 10, pp. 273-284, 2018.

[17] S. Mirjalili, A. Lewis, "The whale optimization algorithm," *Advances in Engineering Software*, vol. 95, pp. 51-67, May, 2016.

[18] M. Z. Kamali, K. Nallasamy, K. Ratnavelu, "Takagi–Sugeno fuzzy modelling of some nonlinear problems using ant colony programming," *Applied Mathematical Modelling*, vol. 48, pp. 635-654, August, 2017.

[19] N. Katal, P. Kumar, S. Narayan, "Optimal PID controller for coupled-tank liquid-level control system using bat algorithm," International Conference on Power, Control and Embedded Systems, Allahabad, pp. 1-4, 2014.

[20] M. Sugeno, G.T. Kang, "Structure identification of fuzzy model," Fuzzy sets and systems, vol. 28, pp. 15-33, 1988.

The Interacting Multiple Model GM PHD for Single Target Tracking

Branko Kovačević, Zvonko Radosavljević and Dejan Ivković

***Abstract*—The recursive algorithm based on non switching Interacting Multiple Model (IMM) with Gaussian Mixture Probability Hypothesis Density (GM PHD) filtering is proposed for estimating and tracking maneuvering target. The approach involves modeling the complex movements with maneuver of target and measurements and applying the probability hypothesis density (PHD) recursion to propagate the posterior intensity, a first order statistics of target in time. We present there closed-form solution to the algorithm recursion. The proposed algorithm combines these recursions with a strategy for managing the number of Gaussian components to increase efficiency. At begin, a single target tracking is performed and tested. Proposed algorithm is shown better performance, in relation with standard GM PHD.**

***Index Terms*— Target tracking, Random finite set, GM PHD algorithm, Interacting multiple model.**

I. Introduction

Basic problem in multi-target tracking is the unknown association of measurements with appropriate targets [1]. Moreover, the data association problem makes up the growth of the computational load in multi target tracking algorithms. Recently, multi-target tracking formulations involve explicit associations between measurements and targets [2]. Multiple Hypotheses Tracking (MHT) and its variations concern the propagation of association hypothesis in time [3].

The random finite set (RFS) approach to multi-target tracking is an alternative association-based methods and comparative discussion between the RFS approach and traditional multi target tracking methods has been given in [4], [5]. In the RFS formulation, the collection of individual targets is treated as a set-valued state, and the collection of individual observations is treated as a set-valued observation. Modeling set valued states and set-valued observations as RFSs allows the problem of dynamically estimating multiple targets in the presence of clutter and association uncertainty to be cast in a Bayesian filtering framework [6], [7]. Here, the states of objects are represented as random sets. Using this model, the birth and death of objects can be described in the tracking algorithm. Moreover, measurements and false alarms are also represented as random sets in the observation model. Mahler [9] employed the random set framework to propose a probability hypothesis density (PHD) filter. This method can avoid the data association between observations and objects. Some implementations of PHD filter are proposed by using the sequential Monte Carlo (SMC) method [8-11. Especially, the implementation in [12] has the convergence proof, and it is called particle PHD filter. In these implementations, the state estimates are extracted from particles representing the posterior intensity by using clustering techniques. In [13] proposed a close-form for PHD filter with assumptions on linear Gaussian system. It is called GMPHD filter. This method reduced a lot computation compared with particle PHD filter. For multi-sensor multi-object tracking, there are some methods to fuse data from multi-sensor in random set approaches such as multiplication likelihood function from sensors [13] or sequential sensor updating [14]. These methods can track varying number of objects with multi-sensor. However, they are implemented based on sequential Monte Carlo, so they need a lot computation.

The Gaussian mixture, consisting of a weighted sum of Gaussian probability density function (*pdf*), each with different means and covariance's, is the natural form of the *pdf* of target state. Using such a structure, a mixture component is created for every possible association, using every possible pairing of target and measurements with the mean and covariance calculated assuming that the particular hypothesis is true, and the weight calculated to represent the probability that the particular hypothesis is true.

In this paper, we proposed a method for multi-sensor multiobject tracking based on GMPHD filter. We extended the GMPHD filter from one sensor to multi-sensor. Therefore, the IMM estimator provides significant noise reduction and a fast response. Previous investigations have found that the IMM estimator is a cost-effective technique for tracking a maneuvering target [4]. In the IMM estimator, several Kalman filters are used in parallel. Each Kalman filter uses a different dynamic model. In order to gain possible improvement on the tracking performance, this paper combines the Interaction Multiple Model (IMM) estimator and GMPHD filter to create an IMM GMPHD filter. Our method can collaborate information from multiple sensors and avoid the data association between observations and objects.

Paper is organized as follows. After the introducing preamble in the Chapter II, formulation of problem is given. Chapter addresses description of proposed algorithm IMM and GMPHD, independent and than together. Chapter IV

Branko Kovačević is with the School of Electrical Engineering, University of Belgrade, 73 Bulevar kralja Aleksandra, 11020 Belgrade, Serbia (e-mail: kovacevic_b@ etf.rs).

Zvonko Radosavljević (corresponding author) is with the Military Technical Institute, Ratka Resanovica 1, 11030 Belgrade, Serbia (e-mail: zvonko.radosavljevic@gmail.com).

Dejan Ivković is with the Military Technical Institute, Ratka Resanovica 1, 11030 Belgrade, Serbia (e-mail: divkovic555@gmail.com).

commences by the testing problem and results of simulations providing a graphical demonstration of their operation. Finally, concluding remarks is given in the Chapter V.

II. PROBLEM STATEMENT

A jump Markov system (JMS) [8] can be described by a set of parameterized state space models whose underlying parameters evolve with time according to a finite state Markov chain. Let $\xi_k \in R^n$ and $y(k) \in R^m$ denote the kinematics state (e.g. target coordinates and velocity) and observation, respectively, at time *k*. Suppose that $\mu^k \in \mathbf{M}$ is the label of the model in effect at time *k*, where M denotes the discrete set of all model labels. Then, the state dynamics and observation are described by the following state transition density $\tilde{f}_{k|k-1}[\xi_k | \xi_{k-1}, \mu^k]$ and measurement likelihood $g_k[y(k), \xi_k, \mu^k]$ In addition, the modes follow a discrete Markov chain with transition probability $\pi(k|k-1)[\mu^k | \mu^{k-1}]$ and the transition of the state vector x_k is governed by

$$f_{k|k-1}[x(k)|x(k-1)] = \tilde{f}_{k|k-1}[x(k)|x(k-1), \mu^k]\pi(k|k-1)[\mu^k | \mu^{k|k-1}] \quad (x)$$

A linear Gaussian JMS (LGJMS) is a JMS with linear Gaussian models, i.e. conditioned on mode r_k the state transition density and observation likelihood are given by [16]

$$\tilde{f}[k|k-1, \xi_k | \xi_{k-1}, \mu(k)] = N[\xi_k; F_{k-1}(r_k)\xi_{k-1}, Q_{k-1}(\mu(k))] \quad (1)$$

$$g_k[y(k), \xi_k, \mu(k)] = N[y(k); H_k(\mu(k))\xi_k, R_k(\mu(k))] \quad (2)$$

where $N(¢; m; Q)$ denotes a Gaussian density with mean *m* and covariance *Q*, $F_{k|1}(r_k)$ and $H_k(r_k)$ denote the transition and observation matrices of model r_k respectively, $Q_{k|1}(r_k)$ and $R_k(r_k)$ denote covariance matrices of the process noise and measurement noise, respectively.

III. IMM GMPHD FILTER

A. Interacting Multiple Model Approach

The basic idea of the multiple-model estimation approach is to assume a set of models M for the hybrid system; run a bank of filters, each based on a unique model in the set; and the overall estimate is given by a certain combination of the estimates from these filters. The IMM algorithm is a suboptimal method of solving the problem of state estimation of a Markov Jump-Linear System (MJLS). A *Markov* chain transition matrix is used to specify the probability that the target is one of the models of operations. IMM algorithm runs filters in parallel, each with an appropriately weighted combination of state estimates as mixed initial conditions [2]. At begin, the model-conditioned re-initialization is perform. The predicted mode probability is calculated by the following [3]:

$$\mu_i^{k|k-1} = \sum_{j=1}^{N} \pi_{ij} \mu_j^{k-1} \quad (3)$$

Likewise, the mixing weight of probability is given by:

$$\mu_{i|j}^{k-1} = \frac{\pi_{ij} \mu_j^{k-1}}{\sum_{j=1}^{N} \pi_{ij} \mu_j^{k-1}} \quad (4)$$

Assuming that the measurement history Y^k is well modeled by the N estimates from the previous processing cycle is then approximated by a single *Gaussian* density $\Lambda_i(k) \approx N\{\mathbf{x}(k); \hat{\mathbf{x}}^i(k), \mathbf{P}^i(k)\}$, where the mean and covariance of the Gaussian are given [17]:

$$\hat{\mathbf{x}}_{k-1|k-1}^{j} = \sum_{i=1}^{N} \hat{\mathbf{x}}_{j,k-1|k-1} \mu_{i|j}^{k-1|k-1}) \quad (5)$$

$$P_{k-1|k-1}^{j} = \sum_{j=1}^{N} \{P_{j,k-1|k-1} + [x_{j,k-1|k-1} - x_{k-1|k-1}^{j}][x_{j,k-1|k-1} - x_{k-1|k-1}^{j}]^T\} \mu_{i|j,k-1} \quad (6)$$

Here, $\hat{\mathbf{x}}_j(k), \mathbf{P}_j(k)$ refer the filter estimate at the output of the previous processing cycle, while $\hat{\mathbf{x}}^i(k), \mathbf{P}^i(k)$ represents the mixed estimates to be provided at the input to the next processing cycle. After the model conditioning filtering step, the mode probability update is performed. A mode probability is given by the:

$$\mu_i(k) = \frac{\mu_i(k|k-1)\Lambda_i(k)}{\sum_{j=1}^{N} \mu_j(k|k-1)\Lambda_j(k)} \quad (7)$$

Finally, the overall estimate and covariance are given by the:

$$\hat{\mathbf{x}}(k|k) = \sum_{i=1}^{N} \hat{\mathbf{x}}^i(k|k) \mu_i(k) \quad (8)$$

$$P(k|k) = \sum_{i=1}^{N} \{P^j(k|k) + [\hat{\mathbf{x}}^j(k|k) - \hat{\mathbf{x}}(k|k)] \cdot [\hat{\mathbf{x}}^j(k|k) - \hat{\mathbf{x}}(k|k)]'\} \cdot \mu_i(k) \quad (9)$$

The basic form of a Gaussian mixture containing *N* components is:

$$p[x(k)] = \sum_{i=1}^{N} \mu_i(k) N\{x(k); \hat{x}_i(k), P^{(i)}(k)\} \quad (10)$$

where $\mu_i(k)$ are the relative weights of each Gaussian component, are the means of each component, and $P^{(i)}$ are the covariance s. As will be seen in the following sections, Gaussian mixture models arise naturally as the solution to several problems in target tracking, including maneuvering target tracking and data association. In the coming sections it will often be necessary to calculate the overall mean and overall covariance of a Gaussian mixture.

B. Gaussian Mixture Probability Hypothesis Density Filter

In this section, we describe the linear-Gaussian multiple target model and the recently developed Gaussian Mixture

PHD filter. The multiple target models for the PHD recursion is described here. Each target follows a linear Gaussian dynamical model:

$$f(k|k-1,x|\xi) = N[x; F(k-1)\xi, Q(k-1)] \quad (11)$$

$$g(k,y|x) = N[y; H(k,x), R(k)] \quad (12)$$

where $N(.;m,P)$ denotes a Gaussian density with mean m and covariance P, $F(k-1)$ is the state transition matrix, $Q(k-1)$ is the process noise covariance, $H(k)$ is the observation matrix and $R(k)$ is the observation noise covariance. The survival and detection probabilities are state independent, $p_S(k,x) = p_S(k)$ and $p_D(k,x) = p_D(k)$.

The intensities of the spontaneous birth and spawned targets are Gaussian mixtures :

$$\gamma(k,x) = \sum_{i=1}^{J_\gamma(k)} w_\gamma^i(k) N[x; m_\gamma^i(k), P_\gamma^i(k)] \quad (13)$$

$$\beta(k|k-1,x|\varsigma) = \\ = \sum_{j=1}^{J_\beta(k)} w_\beta^j(k) N[x; F_\beta^j(k-1)\varsigma + d_\beta^j(k-1), Q_\beta^j(k-1)] \quad (14)$$

where $J_\gamma(k), w_\gamma^i(k), m_\gamma^i(k), P_\gamma^i(k), \ i = 1,...,J_\gamma(k)$ are given model parameters that determine the shape of the birth intensity, similarly, $J_\beta(k), w_\beta^j(k), F_\beta^j(k-1), d_\beta^j(k-1)$ and $Q_\beta^j(k-1) \quad j = 1,...,J_\beta(k)$ determine the shape of the spawning intensity of a target with previous state.

Prediction step:

Under the assumptions that each target follows a linear Gaussian dynamical model, the survival and detection probabilities are constant, the intensities of the birth and spawned targets are Gaussian mixtures, and that the posterior intensity at time *k-1* is a Gaussian mixture of the form

$$D(k-1|k-1,x) = \\ \sum_{i=1}^{J(k-1)} w^i(k-1) N[x; m^i(k-1), P^i(k-1)] \quad (xx) \quad (15)$$

Then the predicted intensity to time *k* is also a Gaussian mixture, and is given by

$$D_{k|k-1,x} = D_{S,k|k-1,x} + D_{\beta,k|k-1,x} + \gamma_{k,x} \quad (16)$$

where $D_S(k|k-1,x)$ is the PHD of existing targets, $D_\beta(k|k-1,x)$ is the PHD for spawned targets, and $\gamma(k,x)$ is the PHD of spontaneous birth targets. The density for existing targets, $D_S(k|k-1,x)$, is determined from the linear Gaussian model using the Kalman prediction equations:

$$D_S(k|k-1,x) = \\ p_S(k) \sum_{j=1}^{J(k-1)} w^j(k-1) N[x; m_S^j(k|k-1), P_S^j(k|k-1)] \quad (17)$$

where

$$m_S^j(k|k-1) = F(k-1) m^j(k-1) \quad (18)$$

$$P_S^j(k|k-1) = Q(k-1) + F(k-1) P^j(k-1) F^T(k-1) \quad (19)$$

and similarly for the spawned target density

$$D_\beta(k|k-1,x) = \\ \sum_{j=1}^{J(k-1)} \sum_{l=1}^{J_\beta(k)} w^j(k-1) w_\beta^l N[x; m_\beta^{j,l}(k|k-1), P_\beta^{j,l}(k|k-1)] \quad (20)$$

where

$$m_\beta^{j,l}(k|k-1) = F_\beta^l(k-1) m^j(k-1) + d_\beta^l(k-1) \quad (21)$$

$$P_\beta^{j,l}(k|k-1) = Q_\beta^l(k-1) + F_\beta^l(k-1) P^j(k-1) (F_\beta^l(k-1))^T \quad (22)$$

Update step:

Under the above assumptions, and that the predicted intensity to time *t* is a Gaussian mixture of the form

$$D(k|k-1,x) = \\ \sum_{i=1}^{J(k|k-1)} w^i(k|k-1) N[x; m^i(k|k-1), P^i(k|k-1)] \quad (23)$$

Then the posterior intensity at time *k* is also a Gaussian mixture, and is given by

$$D(k|k,x) = [1 - p_D(k)] D(k|k-1,x) + \\ + \sum_{y \in Y^k} \sum_{j=1}^{J(k|k-1)} w^j(k,y) N[x; m^j(k|k,y), P^j(k|k)] \quad (24)$$

where the weights are calculated according to the closed form PHD update equation,

$$w^j(k,y) = \\ \frac{p_D(k) w^i(k|k-1) N[y; H(k) m^j(k|k-1), R(k) + H(k) P^j(k|k-1) H^T(k)]}{K(k,y) + p_D(k) \sum_{l=1}^{J(k|k-1)} w^l(k|k-1) N[y; H(k) m^l(k|k-1), R(k) + H(k) P^l(k|k-1) H^T(k)]} \quad (25)$$

and the mean and covariance are updated with the Kalman filter update equations,

$$m^j(k|k,y) = m^j(k|k-1,y) + K^j[k, y - H(k)m^j(k|k-1,y)] \quad (26)$$

$$P^j{}_{k|k} = [I - K^j{}_k H_k]P^j{}_{k|k-1} \quad (27)$$

$$K^j{}_{k|k} = P^j{}_{k|k-1}H^T{}_k[H_k P^j{}_{k|k-1}H^T{}_k + R_k]^{-1} \quad (28)$$

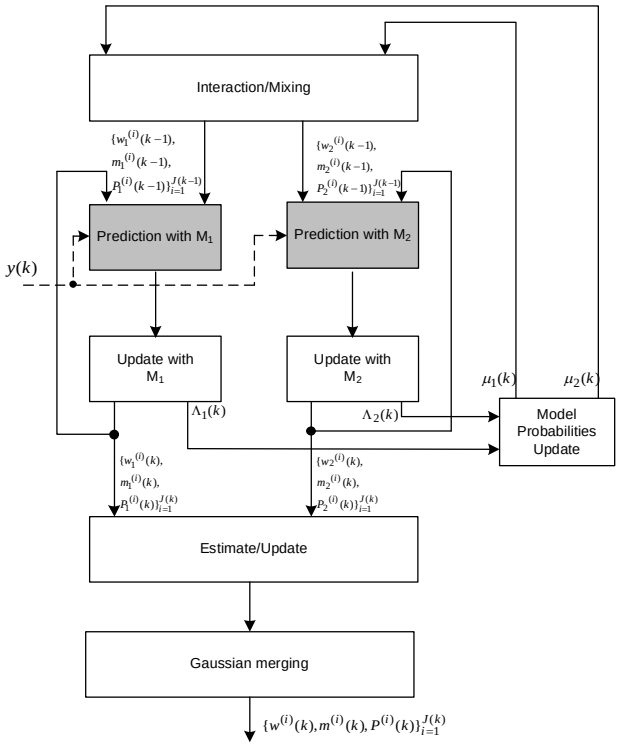


Fig.1 Block diagram of one cycle IMM-GMPHD algorithm

IV. RESULTS OF SIMULATIONS

The implemented IMM-GM-PHD is evaluated by Monte Carlo (MC) simulations over representative 2-dimensional test scenario. A target motion scenario (Fig.2) includes series of non-maneuvering and maneuvering flights modes. Dimension of terrain surveillance is x=1000m and y=1000m. The single target scenario is perform. The speed is constant and equal to 311 m/s. The sampling period of radar sensor is *T=1.5s*. Duration of the scenario is 60 scans. Very important aspects to the implementation of IMM-GM-PHDalgorithm is the selection of the model structures and their parameters.

The following two models, constant velocity (CV) model and constant acceleration (CA) model provide an adequate and self contained model set for tracking purposes. Hence, the model set for IMM filters has been selected as follows:M_1=*CV*, M_2=*CA*. Both models have the same state variables and this greatly simplifies the re-initialization operation of the IMM-GM-PHD filter. The system input is modeled as follows: vector state $\mathbf{x}(k) = [x\ \dot{x}\ y\ \dot{y}]^T$ where x, y are the Cartesian coordinates of the target position, and $\dot{x}, \dot{y}$ are the appropriate velocities, initial

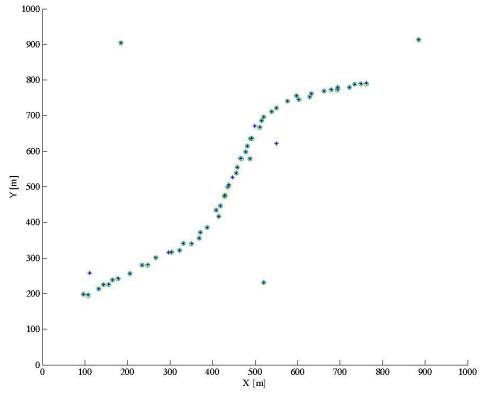


Fig. 2. Simulation scenario (true target and measurements).

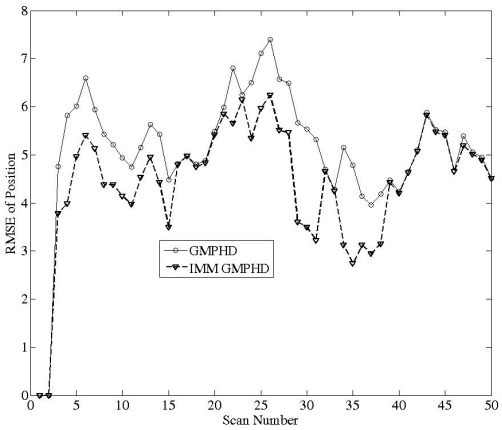


Fig. 3. RMSE position error diagram.

target state. Transition matrix and process noise matrix are given by:

$$\mathbf{F}_1 = \begin{bmatrix} 1 & T & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & T \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad \mathbf{F}_2 = \begin{bmatrix} 1 & \sin wT/w & 0 & \frac{\cos wT - 1}{w} \\ 0 & \cos wT & 0 & -\sin wT \\ 0 & \frac{1-\cos wT}{w} & 1 & \frac{\sin wT}{w} \\ 0 & \sin wT & 0 & \cos wT \end{bmatrix} \quad (29)$$

$$\mathbf{Q}(k) = q\begin{bmatrix} T^3/3 & T^2/2 & 0 & 0 \\ T^2/2 & T & 0 & 0 \\ 0 & 0 & T^3/3 & T^2/2 \\ 0 & 0 & T^2/2 & T \end{bmatrix} \quad (30)$$

respectively, where q=q1=0.0052 is a maneuver coefficient for model M1 and q=q2=0.052 is a maneuver coefficient for model M2. The measurement model is given by the matrices and. The simulation process is governed by a Markov chain with TPM. The calculations are based on MC simulations using $N_{MC} = 100$ realizations. Root mean square error is given by the Fig. 3.

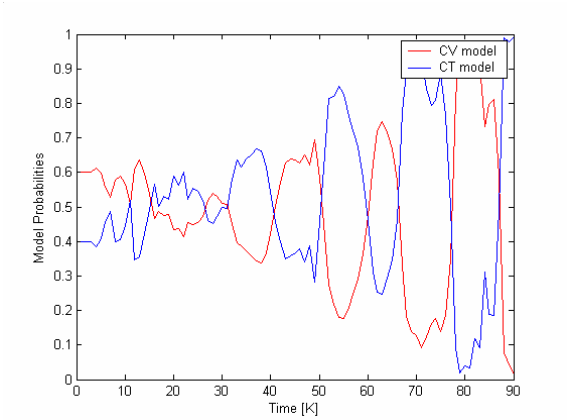


Fig.4. Diagram of associative model probabilities for CV and CT model.

Diagram of associative model probabilities for CV and CT model is shown by the Fig. 4.

V. CONCLUSION

A Interacting Multiple Model method combined with the PHD approach based on Gaussian Mixture implementation for single target tracking, is proposed in this paper. The interaction between IMM and GMPHD filter was illustrated: use the result of GMPHD filter to get the estimated and locations of the target for IMM; use the result of IMM for PHD peak extraction. A novel GMPHD representation defined in a resolution cell was also presented. Simulation results with two-dimensional scenario showed that the proposed algorithm ends up with better performance and less computational load than the standard PHD filter and with better performance than the traditional MHT/assignment algorithms.

Proposed algorithm, which has been presented for tracking target in clutter, have the ability to estimate the target, track the trajectory of the target over time, operate with missed detections and give the trajectories of the targets in the past once a target.

At begin, a single target tracking is performed and tested. Proposed algorithm is shown better performance, in relation with standard GM PHD. In the future work, we have tested multi target scenario, with proposed methodology.

REFERENCES

[1] D. Reid, An algorithm for tracking multiple targets" IEEE Trans. AC, vol. AC-24, no. 6, pp. 843–854, 1979.

[2] Y. Bar-Shalom and T. E. Fortmann, Tracking and Data Association. Academic Press, San Diego, 1988.

[3] Ristic, B., Arulampamm, S., Gordon, N. Beyond the Kalman Filter.Dedham, MA: Artech House, 2004.

[4] Musicki, D., Evans, R. J., and Stankovic, S. Integrated probabilistic data association. IEEE Transactions on Automatic Control, 39, 6 (June 1994), 1237—1241.

[5] S. Blackman, Multiple hypothesis tracking for multiple target tracking, IEEE A & E Systems Magazine, vol. 19, no. 1, part 2, pp. 5–18, 2004.

[6] I. Goodman, R. Mahler, and H. Nguyen, Mathematics of Data Fusion. Kluwer Academic Publishers, 1997.

[7] R. Mahler, Multi-target Bayes filtering via first-order multi-target moments, IEEE Trans. AES, vol. 39, no. 4, pp. 1152–1178, 2003.

[8] R. Mahler, "An introduction to multisource-multitarget statistics and applications," Lockheed Martin Technical Monograph, 2000.

[9] B. Vo, S. Singh, and A. Doucet, "Sequential Monte Carlo methods for multi-target filtering with random finite sets," to appear in IEEE Trans. AES, 2005.

[10] K. Panta, B. Vo, and S. Singh. Improved probability hypothesis density filter (PHD) for multitarget tracking. Proceedings ICISIP, Bangalore 12th-15th December 2005.

[11] B. N. Vo, S. Singh, and A. Doucet, "Sequential Monte Carlo methods for Bayesian multi-target ┐ltering with random ┐nite sets," IEEE Trans.Aerospace and Electronic Systems, vol. 41, no. 4, pp. 1224–1245, Oct 2005.

[12] T. L. Song, D. Mušicki, D.S. Kim and Z.Radosavljević, Gaussian mixtures in multi-target tracking: a look at Gaussian mixture probability hypothesis density and integrated track splitting, IET proceedings on Radar Sonar and Navigation, Vol 6, no 5, pp. 359-364, June 2012.

[13] Z. Radosavljević,,IPDA Filters in the Sense of Gaussian Mixture PHD Algorithm, Scientific Technical Review, 2016,Vol.66,No.3, pp.34-40, Belgrade 2016, UDK : 681:5.017:623:746.3, COSATI:17-09, 12-01.

[14] T. Zajic and R. Mahler, "A particle system implement of the PHD multitarget tracking ┐lter," in Signal Processing, Sensor Fusion and target Recognition XII, SPIE Proc, 2003, pp. 291–299.

[15] B. N. Vo and W. K. Ma, "The Gaussian mixture probability hypothesis density ┐lter," IEEE Transaction Signal Processing, vol. 54, no. 11, 2006.

[16] W. K. Ma, B. Vo, S. Singh, and A. Baddeley, "Tracking an unknown time-varying number of speakers using TDOA measurements: a random finite set approach," IEEE Trans Signal Processing, vol. 54, no. 9, 2006.

[17] B. N. Vo, S. Singh, and W. K. Ma, "Tracking multiple speakers with random sets," in ICASSP, Montreal, Canada, 2004.

[18] O. Erdinc, P. Willet, and Y. Bar-Shalom. Probability Hypothesis Density Filter for Multitarget Multisensor Tracking. Proc. FUSION 2005.

Raspberry PI Based Sound Acquisition Platform for Machine State Estimation

Petar Jandrić, Uroš Rakonjac, Željko Đurović, Aleksandra Marjanović, Sanja Vujnović

Abstract — **Scheduled maintenance in thermal plants is one of the essential tasks in efficient planning and could significantly decrease unwanted expenses due to unnecessary maintenance. Traditional systems currently in use in industry are composed of multiple different subsystems with highly specialized components which makes them unnecessarily expensive and complicated. The purpose of this paper is to propose a single, compact, cost efficient device to perform the acquisition of acoustic signals, which would then be used by a state detection algorithm. The algorithm uses a combination of wavelet transform and neural networks and is computationally inexpensive, so it can be implemented on a simple microcontroller. The testing has been done on real acoustic signals recorded in thermal power plant Kostolac in Serbia.**

Index Terms – **Industry 4.0, Compact platform, Wavelet transform, Sound acquisition**

I. Introduction

It is estimated that up to one third of the maintenance cost in power plants is due to inefficient maintenance [1]. This usually causes machines function in less than optimal environment which can be financially exhausting. While early maintenance of those machines represents an unnecessary cost, unexpected failure of the system due to the lack of maintenance could cause larger scale problems such as unstable power supply and terminate generation of power temporarily. It is widely known that most of the machines in industrial plants consist of rotary actuators whose main point of failure is due to wear of particular components [2]. Being able to estimate the state condition of those parts would increase efficiency and stability of the plant.

The central point of this paper is to compose an efficient, low cost device with enough computational power to perform data acquisition of acoustics signals well enough to be used by the state estimation algorithm. In order to do this, it has to be placed in vicinity of the machine with a rotary actuator. The most important advantage of a system like this is that it is a single, integrated device that has a potential to replace multiple devices used for data acquisition, signal processing and state estimation [3]. This would significantly reduce the complexity of the system and the overall cost as well as the possible communication issues and maintenance needs. Also, since the entire system is represented by a single compact device, which has its own power supply, it can easily be transferred to different machines, as opposed to being connected only to a single one. The data acquired by this device should be informative enough to be used to determine the condition of the actuator, using the already developed algorithm. Thus far this state detection algorithm based on wavelet transform and neural networks has relied on signal acquisition using expensive professional equipment, so the switch from professional microphone to inexpensive one can significantly increase the applicability of the device [4]. Discrete form of wavelet transform and the neural network for classification and state detection are two computationally inexpensive processes, so they can be executed on a microcomputer or a microcontroller. The discrete wavelet transform is used because it is able to extract the necessary features for condition monitoring and state detection from acoustic signals, while keeping low computational requirements [5]. These features are then fed to a neural network which classifies the current state of the machine's rotary actuator.

To implement the data acquisition part of the process, a Raspberry Pi microcomputer was used, as one of the most popular commercially available platforms. Some of its main characteristics are expandability, high potential, configurability and easy programming which contribute to its advantages over traditional data acquisition equipment. USB microphone is used to obtain recordings of an acoustics signals.

This paper is structured as follows. Section 2 gives a detailed description of hardware used to build the device. Section 3 describes the acquisition of acoustic signals using a professional microphone compared to an inexpensive, commercial one. Section 4 compares the quality of acoustic recordings obtained with both microphones bearing in mind the features used by the state estimation algorithm. Final remarks and conclusions are given in section 5 of this paper.

II. Mobile Raspberry Pi Platform

Industrial sensors and data acquisition systems for condition monitoring in power plants are usually very costly and complicated. They are also dependent on multiple hardware and software components to work correctly which can result in communication issues and additional costs and maintenance needs. The platform described in this paper can overcome all of the raised issues, by using commercially available components and integrating them to make a single, compact device. This would decrease the manufacturing,

Petar Jandrić is with the School of Electrical Engineering, University of Belgrade, 11120 Belgrade, Serbia (e-mail: petar.jandric@etf.bg.ac.rs).

Uroš Rakonjac is with the School of Electrical Engineering, University of Belgrade, 11120 Belgrade, Serbia (e-mail: uros.rakonjac@etf.bg.ac.rs).

Željko Đurović is with the School of Electrical Engineering, University of Belgrade, 11120 Belgrade, Serbia (e-mail: zdjurovic@etf.bg.ac.rs).

Aleksandra Marjanović is with the School of Electrical Engineering, University of Belgrade, 11120 Belgrade, Serbia (e-mail: amarjanovic@etf.bg.ac.rs).

Sanja Vujnović is with the School of Electrical Engineering, University of Belgrade, 11120 Belgrade, Serbia (e-mail: svujnovic@etf.bg.ac.rs).

operating and maintenance costs of the condition monitoring system significantly, as well as transition the concept of using multiple different subsystems to using one simple system which contains all the necessary functionalities. The device consists of four main components as shown in Figure 1: Raspberry Pi platform, USB microphone, LCD display and a battery for the power supply.

Fig. 1. System structure.

Using a battery instead of an external power supply enables the platform to be portable and used with less constrictions, which is only possible because of the low power consumption of the other components in the system. The capacity of the battery and low operating times of the system allow the device to be used a large number of times before the need to recharge. The characteristics of the battery are shown in Table I.

TABLE I
BATTERY CHARACTERISTICS

Capacity (nominal)	10000mAh – 3.7V; (6.500mAh - 5.1V); 37Wh
Input connector	Micro USB
Output connector	USB-A
Input voltage / current	5V/2A, 9V/2A, 12V/1.5A
Output voltage / current	5.1V/2.4A, 9V/1.6A max, 12V/1.2A max

Another important advantage of the described system over the traditional industrial data acquisition systems is that it is much more flexible and mobile [3]. This means that the same device could be used for measurements on different machines with rotary actuators as it is easily transportable and reconfigurable, as opposed to systems which are programmed for a highly specific purpose and connected to only one machine.

The base of the device is a Raspberry Pi platform, which is highly used and commercially available, making it a less expensive and a more versatile alternative to industrial computers and controllers. The specifications of the Raspberry Pi used in this paper are given in Table II. Because it is not designed for a single purpose specifically, it can be integrated with a variety of sensors and actuators as a single mobile device. Also, it is easily configurable and programmable which is important if the system requires to be upgraded or adapted for a different task or a different machine.

TABLE II
RASPBERRY PI 3 MODEL B+ SPECIFICATIONS

Processor	1.4GHz 64-bit quad-core
Connectivity	2.4GHz and 5GHz IEEE 802.11.b/g/n/ac wireless LAN, Bluetooth 4.2, BLE Gigabit Ethernet over USB 2.0 (maximum throughput 300Mbps) 4 × USB 2.0 ports
Input power	5VDC via micro USB connector 5V DC via GPIO header
Operating temperature	0–50°C

Data acquisition is performed using a commercial USB microphone. The conditions in a power plant under which the system should operate are such that there is a lot of background noise. The purpose of this choice of microphone is to show that the data recorded by a simple omnidirectional microphone can be used by the state detection algorithm just as well as the data recorded by a professional directional microphone. Choosing this microphone decreases the complexity and connectivity issues, as well as the cost, by trading a not so important level of sound quality. The specifications of the used "*Gyvazla"* USB microphone are given in Table III.

TABLE III
GYVAZLA USB MICROPHONE SPECIFICATIONS

Connection	Standard USB
Polar pattern	Omnidirectional
Impedance	≤ 2.2kΩ
Sensitivity	-30dB ± 3dB
Operating voltage	5V
Frequency response	20Hz – 16kHz
Signal/Noise ratio	84dB
Sampling rates supported	8kHz, 11kHz, 44kHz, 48kHz, 16-bit stereo

The interface of the device consists of two buttons and an LCD screen connected to the Raspberry Pi using I2C communication. The screen displays the current state of data acquisition process, which can be started or stopped by pushing the corresponding button. After the recording has finished, the data is saved and ready to be prepared and used by the detection algorithm.

Operation of the device is very simple, and it contributes to the advantages presented over the systems used broadly in industry. On system power up, Raspberry Pi runs a python

code which integrates the communication with the display, commands from the buttons and the data acquisition. Messages on the display follow the recording process and give the information which buttons can be pressed in order to start or stop the recording. Once the recording is over, the data is saved, and the process can start over if needed.

III. Case Study

The sound acquisition platform was tested on a fan mill in thermal power plant Kostolac A2 in Serbia. Fan mills are used in power plants to grind coal into fine powder using the friction between coal chunks and impact plates within the mill. The impact plates themselves rotate around the center of the mill, and after around 70 working days they get worn and maintenance needs to be conducted in order to change them. The plates in block A2 rotate with the frequency of 12.5 Hz and there is 8 of them placed around the center of the mill.

Recording of the acoustic signals is conducted in such way as to enable comparative analysis of the high-quality professional microphone (AKG C451) and inexpensive USB microphone connected to Raspberry PI platform. The recording lasted for 5 minutes, sampling frequency was 48 kHz and both of the microphones were placed in the vicinity of the mill. In the preprocessing stage the entire 5-minute-long signal has been downsampled to 4.8 kHz and divided into shorter signals 45 seconds long. The main idea is to verify that the acoustic recordings obtained with a cheaper microphone are informative enough for the existing state detection algorithm described in [5].

Features which are relevant for these purposes are discrete wavelet transform (DWT) coefficients which are used to calculate the power of the signal in specific frequency ranges. Namely, the signal has been decomposed into 6-level DWT, and since this transform is implemented as a series of high pass and low pass filters, the obtained frequency ranges after this decomposition are given in Table 4.

TABLE IV
Coefficient Names and Frequency Ranges of a 6-Level DWT Decomposition

Coefficient names	Abbreviation	Frequency range (Hz)
Detailed coefficients, level 1	cd1	1200-2400
Detailed coefficients, level 2	cd2	600-1200
Detailed coefficients, level 3	cd3	300-600
Detailed coefficients, level 4	cd4	150-300
Detailed coefficients, level 5	cd5	75-150
Detailed coefficients, level 6	cd6	37.5-75
Approximation coefficient, level 6	ca6	0-37.5

In order to properly test how different states of the mill reflect on different features, two recording sessions were conducted, one in January and the other in February of 2020. In January the impellers of the recorded mill had 1742 working hours which means that the maintenance was probably well overdo. In February the recording was done after the maintenance was conducted, so the mill in question had only 290 working hours. This implies that the extracted features are expected to significantly differ between the two recordings. However, the features obtained from different microphones on the same dates are expected to be similar.

IV. Results

In order to verify that the recordings obtained from the cheaper microphone are as informative as the ones obtained from the professional equipment, the behavior of the signal in frequency domain needs to be analyzed. The first check is to establish whether all dominant frequency components are present in signals obtained with both microphones. Figure 2 shows the power spectral density (PSD) of both signals in January and in February. There is a significant match between these two signals and the dominant frequency components on 100, 200, 300 and 400 Hz are noticeable with both microphones.

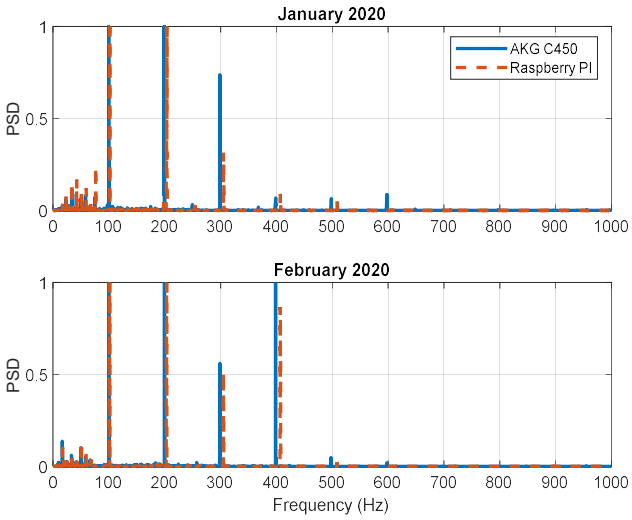


Fig. 2. Power Spectral Density of the signal obtained from the professional microphone (AKG C450) and USB microphone (Raspberry PI) in two recording sessions.

Since the results described in [5] are obtained using DWT coefficient, it would be useful to compare the two microphones in wavelet domain. Discrete wavelet transform is difficult to interpret visually, so Figure 3 shows scalogram of the signal (absolute value of the continuous wavelet transform, CWT) for the session conducted in February, for both microphones. The scalogram is obtained using Morse wavelet with parameters $\gamma = 10$ and $D = 400$ [6]. The dominant frequencies remain the same, and the periodic change of the amplitude of specific frequencies is visible in both cases.

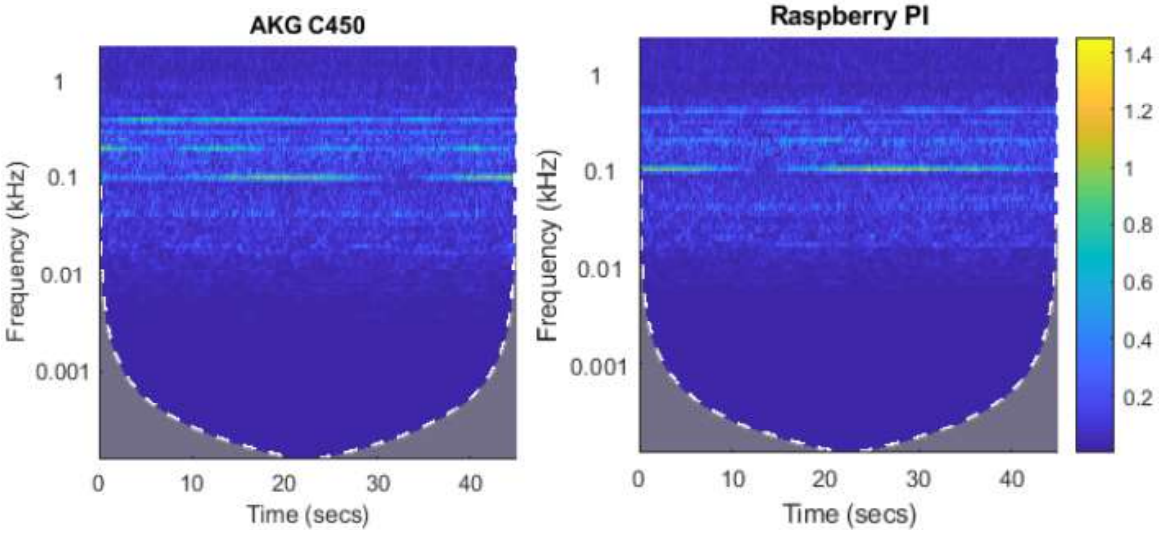


Fig. 3. Scalogram of the signals recorded in February 2020 using professional microphone (left) and USB microphone (right).

After verifying that the signal obtained with the cheaper microphone has all the same frequency components and time variability as the one obtained with the professional microphone, the final test is to check how the features extracted in [5] behave. There are 7 features which represent the power of each of the DWT coefficients from Table 4. Recording sessions both in January and in February are around 5 minutes long, so after preprocessing these signals into segments of 45 seconds separated 15 seconds from one another, there are around 20 recording segments in each session for each of the microphones. The behavior of the features is given in Figure 4 using box plot. The red line represents the median, and the blue box represents the range from 25th to 75th percentile.

There are two things that are interesting to note from Figure 4. First of all, in each recording sessions features from both microphones are quite similar. The only slightly different feature is the one associated with level 6 detailed coefficients in January 2020, when the impact plates were very worn out. This demonstrates that there is similar time-frequency content of the signals obtained with professional and USB microphones. Not only that, but the features themselves are quite similar as well and they change in the same way as the impact plates get worn, thus showing that for the purposes of state detection algorithm presented in [5] the acoustic signals obtained with the Raspberry PI mobile platform can be used with the same success.

The other interesting thing is to examine the behavior of the features between the 2 recordings. The lower frequency components (cd6 and ca6) as well as the higher frequency components (cd1, cd2 and cd3) remain similar when the impact plates are worn (January) and when the impact plates are healthy (February). On the other hand, the coefficient associated with the frequency range between 150 Hz and 300 Hz (cd4) are lower when the plates are new, while the coefficients associated with the frequency range between 75 Hz and 150 Hz are significantly higher, indicating that these two features are probably the most informative for state detection of this particular fan mill.

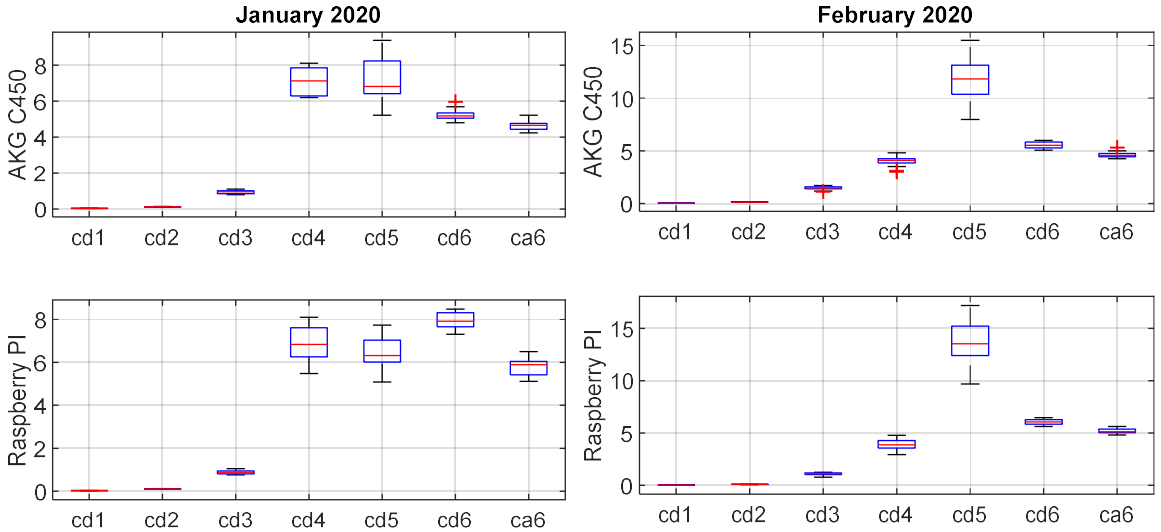


Fig. 4. Power of DWT coefficients of the signal obtained using professional microphone (left) and USB microphone (right) during both recording sessions.

V. CONCLUSION

In this paper a microcontroller platform was analyzed for developing a compact and mobile device which would replace complicated industrial equipment for the acquisition of acoustic signals. A platform based on a Raspberry Pi microcomputer and a commercial USB microphone is proposed. The platform was placed in the vicinity of a mill and was used to record sounds generated by the rotary actuator. The data was then compared to the data obtained with the professional recording equipment in order to estimate whether the acoustic signals are informative enough to be used by the state detection algorithm developed in [5].

The results show that the average power consumption of the device is 300 mA, which enables it to record a large number of recording sequences. Also, the data obtained when using a simple microphone is comparable to the data acquired by a professional microphone when observed in both frequency and wavelet domain. This indicates that the state detection algorithm will give similar results when using signals provided by this device, compared to signals recorded by a professional microphone and a computer. Therefore, the acoustic signals recorded by the described device are informative enough for state detection and that there is no need for expensive professional recording equipment.

The aim of this research was to show that some of the traditional industrial equipment in power plants used for state detection and condition monitoring can be replaced with simple and affordable components without sacrificing performance. In addition to this, using microcomputers or microcontrollers instead of highly specialized computers enables the system to be easily expandable and configurable. Further research will be focused on implementing the state detection algorithm using a discrete wavelet transform and neural networks on the same platform used for data acquisition.

ACKNOWLEDGMENT

This paper is a result of activities within Eureka project E!13084 and the projects supported by Serbian Ministry of Education and Science III42007 and TR32038

REFERENCES

[1] R. K. Mobley, “An introduction to predictive maintenance,” 2nd ed. Amsterdam, Netherlands: Butterworth-Heinemann, 2002.

[2] Y. Lei, J. Lin, Z. He, and M. J. Zuo, “A review on empirical mode decomposition in fault diagnosis of rotating machinery,” Mech Syst Signal Pr, vol. 35, no. 1, pp. 108 – 126, 2013.

[3] E. Uhlmann, A. Laghmouchi, E. Hohwieler, C. Geisert, “Condition monitoring in the cloud,” Procedia CIRP, vol. 38, pp. 53 – 57, 2015.

[4] N. Baydar, A. Ball, “Detection of gear failures via vibration and acoustic signals using wavelet transform,” Mech Syst Signal Pr, vol.17, no. 4, pp. 787 – 804, 2003.

[5] S. Vujnovic, Z. Djurovic, A. Marjanovic, Z. Zarkovic, M. Micovic, “State Detection of Rotary Actuators Using Wavelet Transform and Neural Networks,” XXIII međunarodna naučno-stručna konferencija Informacione Tehnologije (IT’20), Žabljak, Crna Gora, 2020, ISBN 978-9940-8707-0-6.

[6] S. C. Olhede, A. T. Walden, “Generalized Morse wavelets,” IEEE Transactions on Signal Processing, Vol. 50, No. 11, 2002, pp. 2661-2670.

Distributed Multi-Agent Reinforcement Learning Algorithm based on Gradient Correction

Miloš S. Stanković, Marko Beko, Nemanja Ilić and Srdjan S. Stanković

***Abstract*—In this paper we propose a decentralized gradient correction based temporal difference algorithm for multi-agent learning of linear approximation of the value function in Markov decision processes. The algorithm is composed of local parameter updates based on the single-agent off-policy temporal difference with gradient correction algorithm TDC(λ), and a linear dynamic consensus scheme. The overall scheme allows applications in which the agents may have different behavior policies, while evaluating a single target policy. Starting from the properties of the underlying Feller-Markov processes, we show that under nonrestrictive assumptions on the time-varying network topology and the individual state-visiting distributions of the agents the algorithm weakly converges to consensus. A discussion is given on the asymptotic parameter values at consensus, network design and variance reduction. The algorithm's properties are illustrated by simulation results.**

I. Introduction

The problems of decision making by multiple agents operating in uncertain and dynamically changing environments have recently received much attention; these algorithms have fundamentally important role of in the cutting edge technologies and concepts such as Cyber-Physical Systems, Internet of Things (IoT), Industry 4.0, etc. (e.g. [1]–[9] and references therein). Reinforcement learning (RL) is a powerful paradigm for decision making and control in uncertain environments, typically using Markov Decision Process (MDP) models and providing (approximate) solutions to optimal control problems [10], [11]. An important idea in RL is the *temporal difference* (TD) learning, often used to learn approximations to the value function of an MDP. This problem is especially important under complex prctical conditions, such as very large state space and off-policy learning (e.g. [12]). In [13]–[18] several *fast gradient-based algorithms* for TD learning have been proposed which can successfully handle these situations.

Distributed and multi-agent RL methods represent an important element for solving essential problems in the areas dealing with complex, intelligent and networked systems (see e.g. [19], [20] and references therein). In this paper, we propose a new decentralized and distributed algorithm for *multi-agent off-policy gradient temporal difference learning* of linear approximation of the value function in MDPs, based on a recently proposed single-agent off-policy algorithm denoted as *temporal difference algorithm with gradient correction* (TDC(λ)) [13]–[15], [17], [21]. The main idea is to start from the TDC algorithm and to incorporate linear distributed dynamic *consensus* iterations over the underlying network of agents. Off-policy nature of the algorithm allows applications in which the agents may have different behavior policies while evaluating a single target policy. The consensus convexification steps are applied either only to the *parameter vector* in the linear approximation function or to both parameter and auxiliary parameter vectors (introduced by the single-agent algorithm construction) [4], [22], [23]. Under nonrestrictive assumptions on the time-varying network topology and the individual behavior policies, we prove that the parameter estimates generated by the proposed algorithm *weakly converge* to consensus. The proof is the main theoretical contrinution of the paper; it is derived using the results from [4], [8], [17], [24] with the rigorous treatment of the stochastic nature of the underlying MDP's using ergodicity properties of the random processes [13], [22], [25]. We also discuss the possible sets of consensus points and their properties. Finally, simulation results illustrate the main properties and demonstrate that the practical efficiency of the proposed algorithms.

The rest of the paper is organized as follows. In Section II we formulate the problem and define the novel algorithm. The first part of Section III treats some preliminary results, including basic properties of the Feller-Markov state-trace processes. In the second part of Section III, a weak convergence analysis is presented for the proposed algorithm. In Section IV we discuss several important issues, such as introduction of constraints, the impact of consensus to the convergence point, the communication network design and the variance reduction effect. In Section V the results of simulations are shown.

II. Problem Formulation and Definition of the Algorithms

Consider N autonomous agents collaborating to learn linear approximation to the state value function for a given policy in a MDP, denoted by $\mathrm{MDP}^{(0)}$, using observations of sample transitions in additional N *independent MDPs*, denoted as $\mathrm{MDP}^{(i)}$, $i = 1, \ldots, N$. We consider all $N + 1$ processes on a *finite state space* $\mathcal{S} = \{s_1, \ldots, s_M\}$, so that $\mathrm{MDP}^{(0)}$ has the transition matrix P, and $\mathrm{MDP}^{(i)}$ the transition matrices $P^{(i)}$, $i = 1, \ldots, N$; the Markov chains are induced by π

M. S. Stanković is with Singidunum University, Belgrade, Serbia; Vlatacom Institute, Belgrade, Serbia; and Innovation Center, School of Electrical Engineering, University of Belgrade, Serbia; e-mail: milstank@gmail.com

M. Beko is with Faculty of Information Technology and Engineering, University Union Nikola Tesla (FITI), Belgrade, Serbia; and COPELABS, Universidade Lusófona de Humanidades e Tecnologias, Lisboa, Portugal; mbeko@uninova.pt

N. Ilić is with the College of Applied Technical Sciences, Kruševac, Serbia; and Vlatacom Institute, Belgrade, Serbia; e-mail: nemili@etf.rs

S. S. Stanković is with School of Electrical Engineering, University of Belgrade, Serbia; e-mail: stankovic@etf.rs

This work was supported by the Science Fund of the Republic of Serbia under project DECIDE.

and $\pi^{(i)}$, referred to as *target policy* and *behavior policies*, respectively. We are, therefore, dealing with a *cooperative off-policy learning* problem [10], [12], [25].

We consider the one-stage *reward function* $r_\pi : \mathcal{S} \to \mathcal{R}$, specifying the *expected reward* at each state $s \in \mathcal{S}$ [10], [17]. The associated *discounted total reward criterion* (*value function*), with state dependent discount factors $\gamma(s) \in [0,1]$, $s \in \mathcal{S}$, is given by $v_\pi(s) = E_s^\pi\{r_\pi(S_0(0)) + \sum_{n=1}^{\infty} \gamma(S_0(1)) \cdots \gamma(S_0(n)) \cdot r_\pi(S_0(n))\}$, where $E_s^\pi\{\cdot\}$ indicates the expectation w.r.t. to the Markov chain $\{S_0(n)\}_{n>0}$ induced by π, with the initial state $S_0(0) = s$. Denote by Γ the $M \times M$ diagonal matrix with $\gamma(s)$ as diagonal entries.

We start from the basic assumptions on *target and behavior policies* [17]:

(A1) a) P is such that $I - P\Gamma$ is nonsingular;

b) $P^{(i)}$ are *irreducible* and such that for all $s, s' \in \mathcal{S}$, $P_{ss'}^{(i)} = 0 \Rightarrow P_{ss'} = 0$, $i = 1, \ldots, N$.

By the MDP theory [10], [17], [26], v_π (represented as a vector) uniquely satisfies the *Bellman equation*

$$v_\pi = r_\pi + P\Gamma v_\pi, \tag{1}$$

where r_π also denotes (with some abuse of notation) the vector of one-stage expected rewards. Besides (1), v_π also satisfies a broad family of *generalized Bellman equations* (see e.g. [17], [26]). Within the framework of the *temporal-difference* (TD) algorithms, it is usual to consider the Bellman equation depending on the so-called λ-*parameters*, procedurally introduced by the *eligibility traces* (see more technical details below). In this sense, $v_\pi = T^{(\lambda)}v_\pi$ represents a *generalized Bellman equation*, where $T^{(\lambda)}$ is an affine operator on $\mathcal{R}^M$, such that $T^{(\lambda)}v = r_\pi^\lambda + P^{(\lambda)}v$, $\forall v \in R^{|S|}$, for a vector $r_\pi^{(\lambda)}$ and a substochastic matrix $P^{(\lambda)}$.

Let $\phi : S \to R^p$ be a function that maps each state to a p-dimensional feature vector $\phi = [\phi_1 \cdots \phi_p]^T$; let the subspace spanned by feature vectors ϕ be $\mathcal{L}_\phi$. TD algorithms look for a vector $v \in \mathcal{L}_\phi$ that satisfies $v \approx T^{(\lambda)}v$. We assume that $v(s) = \phi(s)^T\theta$, $s \in S$, where $\theta \in R^p$ is a parameter vector, so that the algorithms learn the vector θ. If Φ is an $M \times p$ matrix composed of p-vectors $\phi(s)$ as row vectors, we have the representation $v_\theta = \Phi\theta$.

In order to construct a consensus based distributed algorithm for collaborative finding an approximation function $v_\theta \in \mathcal{L}_\phi$ by using observations from MDP$^{(i)}$, $i = 1, \ldots, N$, we introduce the global Np-dimensional parameter vector $\Theta = [\theta_1^T \cdots \theta_N^T]^T$ ($\dim\{\theta_i\} = p$) and define the following *constrained optimization problem*

$$\begin{gathered} J(\Theta) = \textstyle\sum_{i=1}^N q_i J_i(\theta_i) \\ \text{Subject to } \ \theta_1 = \cdots = \theta_N = \theta, \end{gathered} \tag{2}$$

where $J_i(\theta_i) = \frac{1}{2}\sum_{i=1}^N q_i \|\Pi_{\xi_i}\{T^{(\lambda_i)}v_{\theta_i} - v_{\theta_i}\}\|_{\xi_i}^2$ are the *local objective functions*, $q_i > 0$ *a priori* defined weighting coefficients, λ_i the local λ-parameter and $\Pi_{\xi_i}\{\cdot\}$ the projection onto the subspace $\mathcal{L}_\phi$ w.r.t. the weighted Euclidean norm $\|v\|_{\xi_i}^2 = \sum_{s\in\mathcal{S}} \xi_{i;s} v(s)^2$ for a positive M-dimensional vector ξ_i with components $\xi_{i;s}$, $i = 1, \ldots, M$ (see [17], [25]). In accordance with [17], [26], we take ξ_i to be the invariant probability distribution for the local Markov chain MDP$^{(i)}$, with the transition matrix $P^{(i)}$ induced by $\pi^{(i)}$ ($\xi_i^T P^{(i)} = \xi_i^T$). It follows that

$$\nabla J(\theta) = \sum_{i=1}^N q_i[\Phi^T\Xi_i(T^{(\lambda_i)}v_\theta - v_\theta) + (\Phi^T\Xi_i P^{(\lambda_i)})^T w_i(\theta)], \tag{3}$$

where $\nabla J(\theta) = \nabla J(\Theta)|_{\theta_1=\cdots=\theta_N=\theta}$, Ξ_i is the $M \times M$ diagonal matrix with the components of ξ_i on the diagonal, and $w_i(\theta)$ represents the unique solution (in w_i) of the equation $\Phi w_i = \Pi_{\xi_i}\{T^{(\lambda_i)}v_\theta - v_\theta\}$, assuming that $w_i \in \text{span}\{\phi(s)\}$; it is possible to show that this equation is equivalent to $\Phi^T\Xi_i\Phi w_i = \Phi^T\Xi_i(T^{(\lambda_i)}v_\theta - v_\theta)$ [17].

We define the "local" gradient temporal-difference updates of the local parameter vectors using samples from the expression (3). However, before constructing *distributed gradient based temporal difference algorithms* for learning θ, we need some additional definitions. Recalling that agent i can observe only the Markov chain induced by the behavior policy $\pi^{(i)}$, the local *importance sampling ratios* $\rho_i(s, s') = P_{ss'}^i / P_{ss'}$ for $s, s' \in \mathcal{S}$, are introduced to compensate for the difference in the dynamics of all the corresponding pairs of Markov chains, $i = 1, \ldots, N$; denote $\rho_i(n) = \rho_i(S_i(n), S_i(n+1))$, as well as $\gamma_i(n) = \gamma(S_i(n))$ [17], [26]. The local *temporal-difference term* based on the observed transitions $(S_i(n), S_i(n+1))$ and the reward $R_i(n+1)$ is denoted as

$$\begin{aligned} \delta_i(v_\theta; n) = \rho_i(n)(R_i(n+1) \\ + \gamma_i(n+1)v_\theta(S_i(n+1)) - v_\theta(S_i(n))), \end{aligned} \tag{4}$$

where $v_\theta(S_i(n))$ is the approximation of the value function obtained by the i-th agent using the parameter vector θ. The sequences of local *eligibility trace vectors* $\{e_i(n)\}$ defined by

$$e_i(n) = \lambda_i(n)\gamma_i(n)\rho_i(n-1)e_i(n-1) + \phi(S_i(n)), \tag{5}$$

where $\lambda_i(n) \in [0,1]$ is the local λ-*parameter*, $i = 1, \ldots, N$ [17], [26].

We propose an algorithm composed of two main parts: 1) *local parameter updates*, based on the *gradient descent* methodology using local observations, and 2) interchange of the current estimates aimed at achieving *consensus* between the agents.

The algorithm, denoted as D1-TDC(λ), contains, as its first part, the algorithm TDC from [13]:

$$\begin{aligned} \theta_i'(n) = \theta_i(n) + \alpha_i(n)q_i[e_i(n)\delta_i(v_{\theta_i(n)}; n) - \rho_i(n) \\ \times(1 - \lambda_i(n+1))\gamma_i(n+1)\phi(S_i(n+1))e_i(n)^T w_i(n)], \end{aligned} \tag{6}$$

$$\begin{aligned} w_i'(n) = w_i(n) + \beta_i(n)(e_i(n)\delta_i(v_{\theta_i(n)}; n) \\ -\phi(S_i(n))\phi(S_i(n))^T w_i(n)) \end{aligned} \tag{7}$$

where $w_i(n)$ are auxiliary p-vectors and $v_{\theta_i(n)} = \Phi\theta_i(n)$. The initial values $\theta_i(0)$ are chosen arbitrarily; however, $w_i(0)$, as well as $e_i(0)$, are assumed to satisfy $w_i(0)$, $e_i(0) \in \text{span}\{\phi(S)\}$, in order to achieve the desired convergence properties. Sequences $\{\alpha_i(n)\}$ and $\{\beta_i(n)\}$ are positive step sizes, satisfying $\alpha_i(n) << \beta_i(n)$ (two-time-scale algorithm) [13], [17]. The second part of the algorithm aimed at achieving consensus is given by

$$\theta_i(n+1) = \sum_{j=1}^N a_{ij}(n)\theta_j'(n), \quad w_i(n+1) = w_i'(n), \tag{8}$$

where $a_{ij}(n)$ are random variables, elements of a random matrix $A(n) = [a_{ij}(n)]$ [8], [25], [27]. If one adopts that the available N MDP's are connected by communication links in accordance with a directed graph $\mathcal{G} = (\mathcal{N}, \mathcal{E})$, where $\mathcal{N}$ is the set of nodes and $\mathcal{E}$ the set of arcs, then matrix $A(n)$ has zeros at the same places as the graph adjacency matrix A_G and is *row-stochastic,* i.e. $\sum_{j=1}^{N} a_{ij}(n) = 1$, $i = 1, \dots, N$, $\forall n \geq 0$ (properties of $A(n)$ will be used in the convergence proof).

If we apply convexification to both θ_i and w_i, $i = 1, \dots, N$ we obtain a second algorithm denoted as D2-TDC(λ). It is obtained by keeping the first part of D1-TDC(λ) and the first relation in (8) the same, while the second equation in (8) becomes $w_i(n+1) = \sum_{j=1}^{N} a_{ij}(n) w'_j(n)$.

A global model of the whole system, necessary for the overall analysis, can be constructed in a straightforward way. Let $X(n) = [\Theta(n)^T \vdots W(n)^T]^T$, $\Theta(n) = [\theta_1(n)^T \cdots \theta_N(n)^T]^T$, $W(n) = [w_1(n)^T \cdots w_N(n)^T]^T$; similarly, $X'(n) = [\Theta'(n)^T \vdots W'(n)^T]^T$, together with the corresponding vector components. Then, we have

$$\begin{aligned} X'(n) &= X(n) + \Gamma(n) F(X(n), n), \\ X(n+1) &= \mathrm{diag}\{(A(n) \otimes I_p), (A(n) \otimes I_p)\} X'(n), \end{aligned} \quad (9)$$

where $\otimes$ denotes the Kronecker's product, while $\Gamma(n) = \mathrm{diag}\{\alpha_1(n), \dots, \alpha_N(n), \beta_1(n), \dots, \beta_N(n)\} \otimes I_p$, $F(X(n), n) = [F^{\theta}(X(n), n)^T, F^{w}(X(n), n)^T]^T$, $F^{\theta}(X(n), n) = [q_1 g_1(\theta_1(n), w_1(n), Z_1(n))^T + e_i(n)^T \omega_i(n+1) \vdots \cdots \vdots q_N g_N(\theta_N(n), w_N(n), Z_N(n))^T + e_i(n)^T \omega_i(n+1)]^T$ and $F^{w}(X(n), n) = [k_1(\theta_1(n), w_1(n), Z_1(n))^T + e_1(n)^T \omega_1(n+1) \vdots \cdots \vdots k_N(\theta_N(n), w_N(n), Z_N(n)) + e_N(n)^T \omega_N(n+1)]^T$,

$$\begin{aligned} g_i(\theta, w, z) =& e \bar{\delta}_i(s, s', v_\theta) - \rho_i(s, s') \\ & \times (1 - \lambda_i(s')) \gamma(s') \phi(s') e^T w, \end{aligned} \quad (10)$$

and

$$k_i(\theta, w, z) = e \bar{\delta}_i(s, s', v_\theta) - \phi(s) \phi(s)^T w, \quad (11)$$

with $Z_i(n) = (S_i(n), e_i(n), S_i(n+1))$, $\bar{\delta}_i(s, s', v_\theta) = \rho_i(s, s')(r_i(s, s') + \gamma(s') v_\theta(s') - v_\theta(s))$, and $r_i(s, s')$ is one-step expected reward following policy $\pi^{(i)}$ when transitioning from state s to s'.

Hence, the multi-agent network can be taken as a tool for exploitation of complementary capabilities of different agents, for improvement of performance by obtaining better final value function approximation by adequate selection of the behavior policies and their proper weighting, and for reduction of the variance of the estimates by averaging over the network nodes. It is important to preserve system decentralization by avoiding any centralized fusion center for data processing and decision making, ensuring in such a way to its scalability and robustness.

III. Convergence Analysis

A. Preliminaries

1) Properties of the State-Trace Processes: Under the behavior policies $\pi^{(i)}$ and state-dependent λ, the *state-trace processes* are defined as $\{S_i(n), e_i(n)\}$. It has been shown that these processes are Markov chains with the weak Feller property [17], [26]. In the context of our analysis, we have the following important result [17], [26]:

a) the state-trace weak Feller-Markov chain process has a unique invariant probability measure ζ_i; for each initial condition the *occupation probability measure* converges weakly to ζ_i [17, Theorem 2.1(i)];

b) if $E_{\zeta_i}\{\cdot\}$ denotes the expectation for the stationary state-trace process with initial distribution ζ_i, then $E_{\zeta_i}\{\|f(Z_0)\|\} < \infty$ and $\frac{1}{n}\sum_{j=0}^{n-1} f(Z_i(j))$ converges to $E_{\zeta_i}\{f(Z_i(0))\}$ in mean and a.s., where $f(\cdot)$ is a Lipschitz continuous function [17, Theorem 2.1(ii)].

The results a) and b) are used to prove the following: 1) $E_{\zeta_i}\{\phi(S_i(0))\phi(S_i(0))^T\} = \Phi^T \Xi_i \Phi$; 2) $E_{\zeta_i}\{e_i(0)\delta_i(v;0)\} = \Phi^T \Xi_i (T^{(\lambda_i)} v - v)$, $\forall v \in \mathcal{R}^M$ 3) $E_{\zeta_i}\{e_i(0)\rho_i(0)(\phi^j(S_i(0)) - \gamma(1)\phi^j(S_i(1)))\} = \Phi^T \Xi_i (I - P^{(\lambda_i)})\Phi_j$, $1 \leq j \leq p$; 4) $E_{\zeta_i}\{e_i(0)\rho_i(0)(1 - \lambda_i(1))\gamma(1)\phi^j(S_i(1))\} = \Phi^T \Xi_i P^{(\lambda_i)} \Phi_j$, $1 \leq j \leq p$; where $\phi^j(\cdot)$ is the j-th component of $\phi(\cdot)$ and Φ_j the j-th column vector of Φ [17, Proposition 2.1].

Based on the above results, we have

$$\begin{aligned} \bar{g}_i(\theta_i, w_i) =& E_{\zeta_i}\{g_i(\theta_i, w_i, Z_i(0))\} \\ =& \Phi^T \Xi_i (T^{(\lambda_i)} v_{\theta_i} - v_{\theta_i}) - (\Phi^T \Xi_i P^{(\lambda_i)} \Phi)^T, \end{aligned} \quad (12)$$

$$\begin{aligned} \bar{k}_i(\theta_i, w_i) =& E_{\zeta_i}\{k_i(\theta_i, w_i, Z_i(0))\} \\ =& \Phi^T \Xi_i (T^{(\lambda_i)} v_{\theta_i} - v_{\theta_i}) - \Phi^T \Xi_i \Phi w_i, \end{aligned} \quad (13)$$

Recalling that for any given θ_i there is a unique solution $w_i(\theta_i)$ to the linear equation $\bar{k}_i(\theta_i, w_i) = 0$, $w_i \in \mathrm{span}\{\phi(S)\}$, we obtain that $\bar{g}_i(\theta_i, w_i(\theta_i)) = -\nabla J_i(\theta_i)$.

Based on the above definitions, we have:

Lemma 1 ([17]): Under (A1), the following holds for each θ_i and w_i and each compact set $D_i \subset \mathcal{Z}_i$:

a) $\lim_{m,n\to\infty} \frac{1}{m} \sum_{s=n}^{n+m-1} E_n\{k_i(\theta_i, w_i, Z_i(s+1)) - \bar{k}_i(\theta_i, w_i)\} I(Z_i(n) \in D_i) = 0$ in mean,

b) $\lim_{m,n\to\infty} \frac{1}{m} \sum_{s=n}^{n+m-1} E_n\{g_i(\theta_i, w_i, Z_i(s+1)) - \bar{g}_i(\theta_i, w_i)\} I(Z_i(n) \in D_i) = 0$ in mean,

where $E_n\{\cdot\}$ denotes the conditional expectation given $(Z_i(0), \cdots, Z_i(n))$ and $I(\cdot)$ is the indicator function.

B. Convergence Proof

Define $\Psi(n|k) = A(n) \cdots A(k)$ for $n \geq k$, $\Psi(n|n+1) = I_N$. Let $\mathcal{F}_n$ be an increasing sequence of σ-algebras such that $\mathcal{F}_n$ measures $\{X(k), k \leq n, A(k), k < n\}$.

We introduce the following assumptions:

(A2) There is a scalar $\alpha_0 > 0$ such that $a_{ii}(n) \geq \alpha_0$, and, for $i \neq j$, either $a_{ij}(n) = 0$ or $a_{ij}(n) \geq \alpha_0$.

(A3) For all n there are a scalar $p_0 > 0$ and an integer n_0 such that $P_{\mathcal{F}_n}\{\text{agent } j \text{ communicates to agent } i \text{ on the interval } [n, n+n_0]\} \geq p_0$, $i, j = 1, \dots N$.

(A4) The digraph $\mathcal{G}$ is strongly connected.

(A5) Sequence $\{A(n)\}$ is independent of the processes in MDP$^{(i)}$, $i = 1, \dots, N$.

(A6) There is a $N \times N$ matrix $\bar{\Psi}$ such that $E\{|E_k\{\Psi(n)\} - \bar{\Psi}|\} \to 0$ as $n - k \to \infty$, which has the form $\bar{\Psi} = [\hat{\Psi} \cdots \hat{\Psi}]^T$, where $\hat{\Psi} = [\bar{\psi}_1 \cdots \bar{\psi}_N]^T$ ($|\cdot|$ denotes the infinity norm).

(A7) Sequence $\{X(n)\}$ is tight.

Theorem 1: Let (A1)–(A7) hold. Let $X^{\alpha,\beta}(n)$ be generated by D1-TDC(λ) with $\alpha(n)=\alpha$ and $\beta(n)=\beta$, and let both $w_i^{\alpha,\beta}(0)=w_{i,0}^{\alpha,\beta}$ and $e_i(0)=e_{i,0}\in \text{span}\{\phi(S)\}$. Define $X^{\alpha,\beta}(0)$ by $\lim_{\beta\to 0,\alpha/\beta\to 0} X_0^{\alpha,\beta}=[\theta_0^T\cdots\theta_0^T w_{1,0}^T\cdots w_{N,0}^T]^T$, according to [25], [28]. Then $X^{\alpha,\beta}(\cdot)$ is tight and converges weakly at the fast time-scale to a process $W(\cdot)=[w_1(\cdot)^T\cdots w_N(\cdot)^T]^T$ generated by

$$\dot{w}_i=\bar{k}_i(\theta_i,w_i), \tag{14}$$

for $i=1,\ldots,N$, and any given $\theta_1,\ldots,\theta_N$, with $w_{i,0}\in\text{span}\{\phi(S)\}$, $i=1,\ldots,N$, and at the slow time-scale to $\Theta(\cdot)=[\theta(\cdot)^T\cdots\theta(\cdot)^T]^T$, where

$$\dot{\theta}=\sum_{i=1}^N \bar{\psi}_i q_i \bar{g}_i(\theta,w_i(\theta)), \tag{15}$$

with the initial condition θ_0, where $w_i(\theta)$ is the unique solution (w.r.t. w_i) of the equation

$$\bar{k}_i(\theta,w_i)=G_i\theta+b_i-H_i w_i=0. \tag{16}$$

Moreover, for any integers $n'_{\alpha,\beta}$ such that $\alpha n'_{\alpha,\beta}\to\infty$ as $\alpha\to 0$, there exist positive numbers $\{T_{\alpha,\beta}\}$ with $T_{\alpha,\beta}\to\infty$ as $(\beta,\alpha/\beta)\to 0$ such that for any $\epsilon>0$

$$\lim \sup_{\beta\to 0,\alpha/\beta\to 0} P\{(\theta_i(n'_\alpha+k))\notin N_\epsilon(\bar{\Sigma}_\theta)\}=0 \tag{17}$$

for some $k\in[0,T_{\alpha,\beta}/\alpha]$, where $N_\epsilon(\cdot)$ denotes the ϵ-neighborhood, while $\bar{\Sigma}_\theta$ is the set of points $\bar{\theta}\in\mathcal{R}^p$ defined by $\sum_{i=1}^N \bar{\psi}_i q_i G_i^T w_i(\bar{\theta})=0$.

Proof: The proof can be derived from the general results related to weak convergence of two time-scale stochastic approximation algorithms from [24] (paragraph 8.6) and [4] (Section 3.1), as well as from the specific results presented in [28]. We will not pursue here complex technical details related to the first part of the theorem, since they do not contain any additional aspect interesting from the point of view of the multi-agent system as a whole. We will indicate only some characteristic details related to the attached mean ODE, describing the algorithm asymptotics. Namely, for the fast time-scale we have (14) as a consequence of the fact that $(\alpha/\beta)\bar{g}_i(\theta,w)$ is negligible when $\beta,\alpha/\beta\to 0$. Therefore, for any given θ there is a unique solution $w_i(\theta))$ (w.r.t. w_i) to the linear equation $\bar{k}_i(\theta,w_i)=0$, $w_i\in\text{span}\{\phi(S)\}$. Therefore, at the slow time scale we have the mean ODE (14).

In order to prove (17), we study the limit set $E=\cap_{t\geq 0}\text{cl}\{\theta(t)|t\geq\bar{t}\}$ of (14). We introduce the Lyapunov function

$$V(\theta)=\sum_{i=1}^N \bar{\psi}_i q_i J_i(\theta), \tag{18}$$

according to (2). We have further

$$\begin{aligned}\dot{V}(\theta)&=\langle\sum_{i=1}^N \bar{\psi}_i q_i\nabla J_i(\theta),-\sum_{i=1}^N \bar{\psi}_i q_i\bar{g}_i(\theta,w_i(\theta))\rangle\\&=-\|\sum_{i=1}^N \bar{\psi}_i q_i\bar{g}_i(\theta,w_i(\theta))\|^2.\end{aligned} \tag{19}$$

It follows that $\dot{V}(\theta)=0$ if $\theta\in\bar{\Sigma}_\theta$; if $\theta\notin\bar{\Sigma}_\theta$, then, $\sum_{i=1}^N \bar{\psi}_i q_i\bar{g}_i(\theta,w_i(\theta))\neq 0$ and hence $\dot{V}(\theta)<0$, from which the results follows. ■

Algorithm D2-TDC(λ) has similar properties as the algorithm D1-TDC(λ); however, its convergence points at consensus are different, having in mind that it contains implicitly the additional constraint that all the values of w_i at consensus have to be equal. Obviously, this implies differences in the convergence points at the slow time-scale. We shall not provide a complete presentation of the convergence properties of the algorithm in this case for the lack of space. We shall only specify the limit set of the corresponding ODE: it is defined by $\bar{\Sigma}=\bar{\Sigma}_\theta\times\bar{\Sigma}_w$, the set of points $\bar{x}=[\bar{\theta}^T\bar{w}^T]^T\in\mathcal{R}^{2p}$ satisfying

$$\bar{G}\bar{\theta}+\bar{g}-\bar{H}\bar{w}=0,\qquad \bar{G}^T\bar{w}=0, \tag{20}$$

where $\bar{G}=\sum_{i=1}^N \bar{\psi}_i q_i\Phi^T\Xi_i(P^{(\lambda_i)}-I)\Phi$, $\bar{b}=\Phi^T\sum_{i=1}^N \bar{\psi}_i q_i\Xi_i r_\pi^{(\lambda_i)}$, $r_\pi^{(\lambda_i)}$ is a constant M-vector in the affine function $T^{(\lambda_i)}(\cdot)$, while $\bar{H}=\sum_{i=1}^N \bar{\psi}_i q_i\Phi^T\Xi_i\Phi$. We introduce in this case the Lyapunov function

$$V(\theta,w_1,\ldots w_N)=\frac{1}{2}\|\theta-\bar{\theta}\|^2+\frac{1}{2}\|w-\bar{w}\|^2, \tag{21}$$

where $\bar{\theta}$ and $\bar{w}$ are given by (20). We have directly

$$\begin{aligned}\dot{V}(\theta,w)&=\langle\theta-\bar{\theta},\bar{G}^T w\rangle+\langle w-\bar{w},\bar{G}\theta+\bar{b}-\bar{H}w\rangle\\&=-\langle w-\bar{w},\bar{H}(w_-\bar{w})\rangle.\end{aligned} \tag{22}$$

Therefore, $\dot{V}(\theta,w)<0$ for $w_i\in\text{span}\{\phi(S)\}$ and $w\neq\bar{w}$, showing that $\hat{w}=\bar{w}$ if $[\hat{\theta}^T\hat{w}^T]^T\in E$ and $\hat{w}\in\text{span}\{\phi(S)\}$. Similarly, we can demonstrate that if $[\hat{\theta}^T\bar{w}^T]^T\in E$, then $\hat{\theta}=\bar{\theta}$. In such a way we infer that for initial conditions $w_i(0)\in\text{span}\{\phi(S)\}$ the limit set is the set $\bar{\Sigma}$.

IV. Discussion

A. Following [17], it is easily possible to construct a constrained version of the proposed algorithm and to prove their weak convergence, introducing projections on the constraint sets in the form of closed balls in $\mathcal{R}^p$ centered at the origin and with sufficiently large radii. We shall only mention at this point that assumption (A7) can be removed in the case of the constrained algorithms. Also, one has to take into account that the asymptotic ODE's contain now the boundary reflection terms, influencing, in general, the definition of their limit sets [17]. Notice that [4] contains also an analysis devoted to the constrained algorithms, starting from different formulation of the projection sets.

B. The role of convexification of w_i, introduced in D2-TDC(λ) remains to be additionally clarified. Obviously, D1-TDC(λ) follows from the basic local relations $\Phi w_i=\Pi_{\xi_i}\{T^{(\lambda_i)}v_\theta-v_\theta\}$ providing the unique solutions $w_i(\theta)$ for all θ, $i=1,\ldots,N$. The algorithm D2-TDC(λ) is based on the additional constraint $w_1(\theta)=\cdots=w_N(\theta)=w(\theta)$, where $w(\theta)$ is the unique solution of

$$\Phi^T(\sum_{i=1}^N \bar{\psi}_i q_i\Xi_i)\Phi w(\theta)=\sum_{i=1}^N \bar{\psi}_i q_i\Pi_{\xi_i}\{T^{(\lambda_i)}v_\theta-v_\theta\}. \tag{23}$$

It is straightforward to show that we have

$$\Phi^T(\sum_{i=1}^{N}\bar{\psi}_i q_i \Xi_i)\Phi w(\theta) = \Phi^T \sum_{i=1}^{N}\bar{\psi}_i q_i \Xi_i \Phi w_i(\theta), \qquad (24)$$

for any given θ. Consequently, convergence points for θ are, accordingly, different for D1-TDC(λ) and D2-TDC(λ). Only in the case of equal λ-parameters and equal behavior policies for all the agents both algorithms provide the same solution.

C. The proposed multi-agent algorithm can be considered from two viewpoints: 1) as a tool for organizing coordinated actions of multiple agents contributing to a common goal and 2) as a parallelization tool, allowing faster convergence. In the first case, the agents can have: 1) different behavior policies and 2) different ways of defining local λ-parameters. The choice of weighting factors $\bar{\psi}_i q_i$ can help, in this sense, to place more emphasis on those agents that can provide greater contribution to the overall goal. It is an open question how the nature of the given set of MDPs influences the choice of q_i, and *vice versa*, how to choose local behavior policies, having in mind the type of coordination offered by the communication network. Practically, it is obvious that complementary actions with adequate weights, covering possibly overlapping subsets of MDP states, can contribute significantly to the overall rate of convergence.

D. All the above theorems show that the proposed algorithm converges weakly to the defined fix points; clearly, these points depend on the products $\bar{\psi}q_i$. Therefore, one of the prerequisites for final planning of the whole system is the network design, including the network topology and the numerical values of the convexification coefficients defining matrix $A(n)$. If one adopts a time invariant network, the problem reduces to the definition of the elements of an $N \times N$ matrix A providing $\bar{\psi}_i = \frac{1}{N}$ (having in mind the initial freedom in selecting the weights q_i). Then, one has to solve the standard equation $\mathbf{1}^T A = \mathbf{1}^T$, where $\mathbf{1}^T = [1 \cdots 1]^T$, which always has a solution satisfying the given constraints $(a_{ij} \geq 0, \sum_j a_{ij} = 1)$ [8].

E. One of the main problems related to the applications of temporal difference learning algorithms is large variance of eligibility traces and, consequently, of the parameter estimates in the value function approximation problems. In this sense, introduction of the consensus based multi-agent methodology provides a possibility for variance reduction, based on the intrinsic noise averaging over the set of network nodes. This statement holds for the distributed TDC algorithms proposed in this paper. More detailed formal analysis of the asymptotic variance can be based on the methodology proposed in [4] using asymptotic stochastic differential equations attached to the algorithm.

V. Simulation Results

In this section we illustrate the discussed properties of the proposed algorithm by applying it to a version of the Boyan's chain shown in Fig. 1 [13], [23], [25], [28].

We assume that the discount factor is $\gamma = 0.85$. The policy which can chosen at each state is the probability of selecting the exit action a^{exit} at state s. The reward for exiting is

Fig. 1. Diagram of the simulated MDP

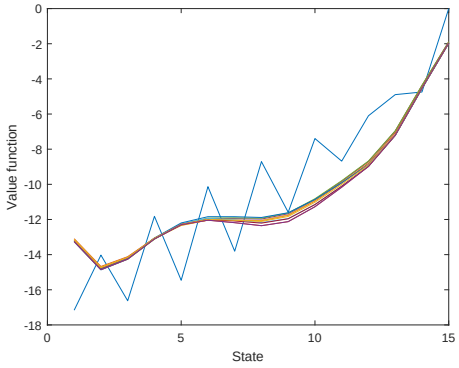


Fig. 2. Value function approximation obtained using D1-TDC(λ) in which the agents have behavior policies such that they can individually visit only a subset of the states. True value function is shown using blue line.

$r(s, a^{\text{exit}}, s') = -4$ for all s and s', but the probability of staying in the same state (jammed) is fixed to 0.2. If we choose action a^h the reward is $r(s, a^h, s') = -1$ for all s and s', but the probability of staying in the same state grows with the state number as $1 - \frac{1}{s}$, where s it the state number. The *target policy* is the stationary policy $\pi(s, a^{\text{exit}}) = 0.8$. We assume 10 agents with time-invariant communication network connected only to a few randomly chosen neighbors. The agents can only obtain 7-features Gaussian radial basis representations of the state vector as functions of distances to the states 1, 3, 5, 7, 9, 11 and 13. Since the chain has an absorbing state we run the algorithms in multiple episodes.

In the first experiment, we assume that the behavior policies are such that the agents can individually visit only a subset of the states; hence, they are not able to estimate the value function without collaboration. While visiting the allowed subsets of the states the agents have different behavior policies. In Fig. 2 the value function approximation obtained for this case is shown for the D1-TDC(λ) algorithm, after 20 episodes, where each agent also uses different λ parameter.

In the second experiment we demonstrate the denoising effect of the introduced distributed algorithms. We assume that the agents have the same stationary behavior policies as above, but that they all start in state 1 and are able to advance to the final state 15. In Fig. 3, the value function approximation obtained after 8 episodes using D2-TDC(λ) is represented for step sizes $\alpha = 0.01$, $\beta = 0.5$ (two time scales). The true value function is depicted using the dashed line. It can be seen that the approximation is better for the states $z_i \in \{1, 3, 5, 7, 9, 11, 13\}$, since these are references for the radial basis representation (it is not possible to converge to the true value function because of the radial basis functions approximation). As can be seen from the figure, the agents have practically achieved consensus. Fig. 4 shows the param-

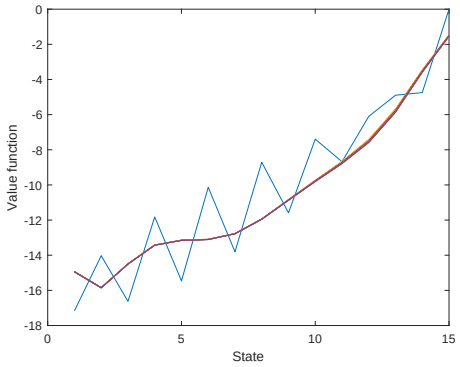


Fig. 3. Value function approximation obtained using D2-TDC(λ) in which all the agents have behavior policies such that they can visit all the states. True value function is shown using blue line.

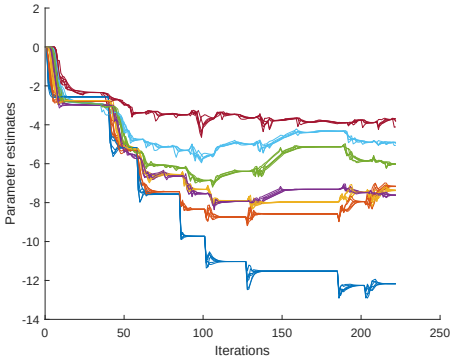


Fig. 4. Parameter estimates for all the agents using D2-TDC(λ) in the second experiment.

eter estimates $\theta_i(n)$ as functions of the number of iterations n which obviously have lower variance compared to the single agent case (Fig. 5).

REFERENCES

[1] S. S. Stanković, M. S. Stanković, and D. M. Stipanović, "Consensus based overlapping decentralized estimator," *IEEE Trans. Autom. Control*, vol. 54, pp. 410–415, 2009.

[2] ——, "Consensus based overlapping decentralized estimation with missing observations and communication faults," *Automatica*, vol. 45, pp. 1397–1406, 2009.

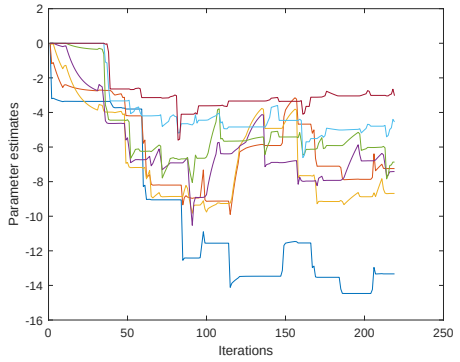


Fig. 5. Parameter estimates using single-agent TDC(λ) algorithm.

[3] J. N. Tsitsiklis, D. P. Bertsekas, and M. Athans, "Distributed asynchronous deterministic and stochastic gradient optimization algorithms," *IEEE Trans. Autom. Control*, vol. 31, pp. 803–812, 1986.

[4] H. J. Kushner and G. Yin, "Asymptotic properties of distributed and communicating stochastic approximation algorithms," *SIAM J. Control Optim.*, vol. 25, pp. 1266–1290, 1987.

[5] A. Nedić and A. Ozdaglar, "Distributed subgradient methods for multi-agent optimization," *IEEE Trans. Autom. Control*, vol. 54, pp. 48–61, 2009.

[6] P. Bianchi, G. Fort, and W. Hachem, "Performance of a distributed stochastic approximation algorithm," *IEEE Trans. Inf. Theory*, vol. 59, pp. 7405–7418, 2013.

[7] M. S. Stanković, S. S. Stanković, and K. H. Johansson, "Distributed time synchronization for networks with random delays and measurement noise," *Automatica*, vol. 93, pp. 126 – 137, 2018.

[8] M. S. Stanković, N. Ilić, and S. S. Stanković, "Distributed stochastic approximation: Weak convergence and network design," *IEEE Trans. Autom. Control*, vol. 61, no. 12, pp. 4069–4074, 2016.

[9] M. S. Stanković, K. H. Johansson, and D. M. Stipanović, "Distributed seeking of Nash equilibria with applications to mobile sensor networks," *IEEE Trans. Autom. Control*, vol. 57, no. 4, pp. 904–919, 2012.

[10] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. MIT press Cambridge, 1998.

[11] D. P. Bertsekas and J. Tsitsiklis, *Neuro-Dynamic Programming*. Athena Scientific, 1996.

[12] D. Precup, R. S. Sutton, and S. Dasgupta, "Off-policy temporal-difference learning with function approximation," in *Proc. 18th Int. Conf. on Machine Learning*, 2001, pp. 417–424.

[13] R. S. Sutton, H. R. Maei, D. Precup, S. Bhatnagar, D. Silver, C. Szepesvári, and E. Wiewiora, "Fast gradient-descent methods for temporal-difference learning with linear function approximation," in *Proc. 26th Int. Conf. on Machine Learning*, 2009, pp. 993–1000.

[14] M. Geist and B. Scherrer, "Off-policy learning with eligibility traces: A survey," *Journal of Machine Learning Research*, vol. 15, pp. 289–333, 2014.

[15] H. R. Maei, "Gradient temporal difference learning algorithms," Ph.D. dissertation, University of Alberta, 2011.

[16] C. Dann, G. Neumann, and J. Peters, "Policy evaluation with temporal differences: a survey and comparisons," *Journal of Machine Learning Research*, vol. 15, pp. 809–883, 2014.

[17] H. Yu, "On convergence of some gradient-based temporal-differences algorithms for off-policy learning," *arXiv:1712.09652*, 2017.

[18] B. Dai, A. Shaw, L. Li, L. Xiao, N. He, J. Chen, and L. Song, "SBEED: Convergent reinforcement learning with nonlinear function approximation," *arXiv:1712.10285*, 2017.

[19] J. K. Gupta, M. Egorov, and M. Kochenderfer, "Cooperative multi-agent control using deep reinforcement learning," in *Autonomous Agents and Multiagent Systems*, G. Sukthankar and J. A. Rodriguez-Aguilar, Eds. Cham: Springer International Publishing, 2017, pp. 66–83.

[20] A. OroojlooyJadid and D. Hajinezhad, "A review of cooperative multi-agent deep reinforcement learning," *arXiv:1908.03963*, 2019.

[21] H. R. Maei and R. S. Sutton, "A general gradient algorithm for temporat difference prediction learning with eligibitity traces," in *Proc. 3rd Conf. Artificial General Intelligence*, 2010, pp. 91–96.

[22] S. V. Macua, J. Chen, S. Zazo, and A. H. Sayed, "Distributed policy evaluation under multiple behavior strategies," *IEEE Trans. Autom. Control*, vol. 60, no. 5, pp. 1260–1274, 2015.

[23] ——, "Cooperative off-policy prediction of markov decision processes in adaptive networks," in *Proc. Int. Conf. Acoustics, Speech and Signal Processing (ICASSP)*, 2013, pp. 4539–4543.

[24] H. J. Kushner and G. Yin, *Stochastic Approximation and Recursive Algorithms and Applications*. Springer, 2003.

[25] M. S. Stanković and S. S. Stanković, "Multi-agent temporal-difference learning with linear function approximation: Weak convergence under time-varying network topologies," in *2016 American Control Conference (ACC)*, 2016, pp. 167–172.

[26] H. Yu, A. Mahmood, and R. Sutton, "On generalized Bellman equations and temporal-difference learning," *Journal of Machine Learning Research*, vol. 19, pp. 1–49, 2019.

[27] S. S. Stanković, M. S. Stanković, and D. M. Stipanović, "Decentralized parameter estimation by consensus based stochastic approximation," *IEEE Trans. Autom. Control*, vol. 47, pp. 531–543, 2011.

[28] M. S. Stanković, M. Beko, and S. S. Stanković, "Distributed gradient temporal difference off-policy learning with eligibility traces: Weak convergence," in *Proc. IFAC World Congress*, 2020.

Optimizing Convergence Speed in Adaptive Consensus Algorithms

Nemanja Ilić, Miloš S. Stanković and Srdjan S. Stanković

***Abstract*—In this paper multi-step adaptive consensus algorithms for distributed information processing via networks of intelligent sensors are considered. Network nodes are assumed to have limited resources in view of the limited connectivity range and the local availability of measurements (limited sensing range). Consensus schemes aimed at achieving the weighted average of node states (local information processing results) by utilizing the locally available communication channels are discussed. The setting where the adaptive asymptotic consensus weights reflect assertions about the individual node quality as well as the real-time measurement availability is adopted. Within this context, the paper proposes practically convenient procedure for designing the adaptive communication weights exhibiting the fastest convergence speed of the resulting consensus scheme, and proves the claimed optimality. This issue is of great importance for reducing sensor network communication requirements and increasing its energy efficiency. A number of numerical examples are given, illustrating the algorithm behind the proposed procedure and demonstrating its effectiveness.**

***Index Terms*—Sensor networks, Adaptive consensus, Convergence speed, Optimization.**

I. Introduction

The problem of signal and information processing via sensor networks has attracted a great deal of scientific interest over the past few decades, *e.g.*, [1], [2]. Distributed algorithms have been in the focus of many researchers, due to their high potential for increasing the robustness and fault tolerance of the resulting schemes. Among various types of distributed schemes, consensus algorithms have emerged as one of the dominantly used protocols for reaching the agreement of different nodes in the network regarding variables connected to the global processing task [3], [4].

The application domain of consensus algorithms has been very wide. Consensus has been used in distributed state estimation [2], [5], [6], [7], fault diagnosis [8], [9], change detection [10], [11], [12], [13], optimization [14], target tracking [15], [16], [17], [18], [19], [20], etc. All of these solutions inherently adopt the local connectivity assumption, *i.e.*, nodes in the network are connected only to their neighboring nodes; all-to-all connections are usually not considered since they correspond to the centralized schemes.

One of the major challenges for the successful application of consensus schemes has been to appropriately address the problem of some sensors not receiving measurements connected to the global processing task at a given time instant, *i.e.*, the problem of local observability [21], [22]. This represents the special case of a more general setting with discrepant local processing results - different nodes in the network obtain local variables with different "quality", which can be related to both *a priori* assertions about the nodes (*e.g.*, connected to local measurement error covariances [9], [12], [13]), as well as to real-time assertions (*e.g.*, reflecting the fact that a node does or does not receive measurements at a given time instant [15], [17], or receives them with high or low probability [16], [23]). In any case, this issue has to be taken into account when designing the consensus communication scheme, so that the achieved agreement is dominantly influenced by the "high quality" nodes. The so-called adaptive consensus schemes [15], [16] have demonstrated their effectiveness in solving the problems with discrepant local processing results. The proposed algorithms in [23], [17] solve the problem by using multiple consensus iterations at each time instant and by adopting the desirable asymptotic behavior of the consensus scheme, resembling the two parallel passes scheme from [24].

When applying any communication procedure in the context of sensor networks, and, in particular, when applying the multi-step consensus schemes, it is very important to reduce the needed communication burden. This represents an issue that must be taken into account, having in mind the imposed energy efficiency imperatives when working with sensor networks. In this paper we address the problem of designing the adaptive consensus weights exhibiting the fastest convergence speed, and propose a practically convenient procedure as a solution. We also prove the optimality of the proposed solution, extending the classical results from [25]. The adopted problem setting is general and can be applied to different application scenarios, such as distributed state estimation, target tracking, change detection, optimization, etc. A number of numerical examples are given, illustrating the underlying steps behind the proposed procedure and demonstrating its effectiveness.

The outline of the paper is as follows. Problem setting and the used adaptive multi-step consensus algorithm are described in Section II. Section III deals with the problem of optimizing the consensus scheme convergence speed. Characteristic numerical simulations are given Section IV, while concluding remarks are stated in Section V.

II. Adaptive Consensus Algorithm

A. Problem Definition

Assume that we have a network of N intelligent sensors, represented by a directed graph $\mathcal{G} = (\mathcal{N}, \mathcal{E})$, where

N. Ilić, who is the corresponding author, is with the College of Applied Technical Sciences, 37000 Kruševac, Serbia and Vlatacom Institute, 11000 Belgrade, Serbia; E-mail: nemili@etf.rs

M. S. Stanković is with the Innovation Center, Faculty of Electrical Engineering, University of Belgrade, 11000 Belgrade, Serbia; Vlatacom Institute, 11000 Belgrade, Serbia; COPELABS, Universidade Lusófona, Lisboa, Portugal and Singidunum University, 11000 Belgrade, Serbia; E-mail: milsta@kth.se

S. S. Stanković is with the Faculty of Electrical Engineering, University of Belgrade, 11000 Belgrade, Serbia; E-mail: stankovic@etf.rs

$\mathcal{N} = \{1, \ldots, N\}$ is the set of nodes and $\mathcal{E} = \{(i,j)\}$ is the set of directed links from node i to node j. Let A be the adjacency matrix of $\mathcal{G}$. We shall assume that the network is aimed at performing a global information processing task in a distributed manner, based on the locally available information and real-time communication via the available communication links (local connectivity assumption). In order to accomplish the adopted global task, we assume that each node locally processes its state vector:

$$\xi_i(t|t) = \xi_i(t|t-1) + F_i\left(z_i(t)\right), \tag{1}$$

where ξ_i represents the state vector (size $n \times 1$) of the node i, F_i its local filtering function, and z_i locally available measurements. This setting encompasses typical distributed estimation and tracking problems, as well as distributed change detection, optimization and calibration tasks. Importantly, we shall assume that different nodes in the network can be in different real-time situations regarding the availability of measurements (local observability assumption), so that the quality of the individual filtering processes can differ significantly in time. In this respect, we shall further assume that each node can estimate the quality of its local filtering process, resulting in the scalar local weight $\gamma_i(t) \in (0,1]$, reflecting quality of the estimate $\xi_i(t|t)$. These weights $\gamma_i(t)$ typically directly reflect the availability of measurements (as in target tracking scenarios where only part of the nodes observe the target).

The nodes subsequently exchange information on their states and perform local predictions:

$$\xi_i(t+1|t) = P_i\left(\sum_{j \in \mathcal{J}_i} c_{ij}(t)\xi_j(t|t)\right), \tag{2}$$

where P_i is the local prediction function, and $c_{ij}(t)$, $i,j = 1,\ldots,N$, are time varying weights reflecting the real-time communication weights between the nodes, such that $N \times N$ matrix $C(t) = [c_{ij}(t)]$ (consensus matrix) is row-stochastic for all t, $\mathcal{J}_i = \mathcal{N}_i \cup \{i\}$, where $\mathcal{N}_i$ is the set of in-neighbors of the i-th node, *e.g.*, [1], [26], [27].

We see that the proposed setting requires the exchange of node states only (size $n \times 1$) between the nodes (compare with [22], [19]). It contains two parts: 1) the filtering part, in which the local measurements are processed, and 2) the prediction part, in which the agreement between the nodes is enforced by forming a convex combination of the communicated local states, which are then included in the prediction step, *e.g.*, [1], [4], [26].

B. Multi-Step Scheme

The main idea behind the adaptive consensus scheme discussed in the paper is to use the locally available weights $\gamma_i(t)$ in the design process of the consensus weights $c_{ij}(t)$, so that the corresponding consensus scheme potentiates the states with high $\gamma_i(t)$. To this end, we shall apply the so-called multi-step consensus, and design such a scheme where the asymptotic consensus weight connected to $\xi_i(t|t)$ will be proportional to $\gamma_i(t)$. Moreover, in order to propose a general solution, we shall assume that each node i is associated with the weight w_i, reflecting *a priori* assertions about the quality of that node, so that the desired asymptotic consensus weights will be

$$c_{ij}^{\infty}(t) = \frac{w_j\gamma_j(t)}{\sum_{j'=1}^{N} w_{j'}\gamma_{j'}(t)}. \tag{3}$$

Consequently, after the communication scheme is applied, all the nodes will have estimations that are mostly influenced by the "high quality" nodes, both in terms of *a priori* as well as real-time quality assertions. Of course, in the case of equal *a priori* weights, we will have $c_{ij}^{\infty}(t) = \gamma_j(t) / \sum_{j'=1}^{N}\gamma_{j'}(t)$.

In order to accomplish the set task, we shall apply K consensus steps at each time instant t. In each consensus step, the nodes will exchange the variables based on $\gamma_i(t)$ as well as the local states. The variables will be used for weighting the communicated states. Before the consensus scheme is applied, we need to obtain matrix B, which represents a consensus matrix based on A used in communicating local weights $\gamma_i(t)$ between the nodes, such that

$$\lim_{K\to\infty} B^K = \mathbf{1}w^T, \tag{4}$$

where $\mathbf{1}$ represents a vector of ones of appropriate size, and w *a priori* weight vector $w = [w_1 \cdots w_N]^T$. Such a matrix satisfies $w^T B = w^T$ [4], [1], [14]. It can be shown that this equation has, in principle, infinitely many solutions, using the procedure analogous to the one from [14]. Solving it requires the knowledge of network topology; it is performed only once for a fixed topology (see the following section).

Let $\gamma_i^{[\kappa]}(t)$ and $\xi_i^{[\kappa]}(t|t)$, $\kappa = 1,\ldots,K$, be the weight and the state of the i-th node connected to the κ-th consensus step, respectively. We shall start from

$$\gamma_i^{[1]}(t) = \gamma_i(t), \quad \xi_i^{[1]}(t|t) = \xi_i(t|t).$$

In this first consensus step, the nodes exchange $\gamma_i^{[1]}(t)$ and their states $\xi_i^{[1]}(t|t)$. The weights $\gamma_i^{[1]}(t)$ are being exchanged through B, so that the corresponding consensus matrix that defines the weights for the communicated states is

$$C^{[\kappa]}(t) = \left(B \cdot diag\left(\gamma_1^{[\kappa]}(t), \ldots, \gamma_N^{[\kappa]}(t)\right)\right)_{rs}, \tag{5}$$

$\kappa = 1$, where $(\cdot)_{rs}$ denotes an operator making the resulting matrix row-stochastic - it divides elements of each row of the argument matrix by the corresponding row sums.

Now that we have the consensus matrix $C^{[\kappa]}(t) = [c_{ij}^{[\kappa]}(t)]$, $i,j = 1,\ldots,N$, the local states are obtained by

$$\xi_i^{*[\kappa]}(t|t) = \sum_{j\in\mathcal{J}_i} c_{ij}^{[\kappa]}(t)\xi_j^{[\kappa]}(t|t), \tag{6}$$

where star denotes the states obtained after the consensus step is applied.

For the next consensus step, our design requires that the weight of each node corresponds to the sum of the weights it received previously, so that $\gamma^{[\kappa+1]}(t) = [\gamma_1^{[\kappa+1]}(t) \quad \cdots \quad \gamma_N^{[\kappa+1]}(t)]^T$ becomes:

$$\gamma^{[\kappa+1]}(t) = B \cdot diag\left(\gamma_1^{[\kappa]}(t), \ldots, \gamma_N^{[\kappa]}(t)\right) \cdot \mathbf{1} = B \cdot \gamma^{[\kappa]}(t). \tag{7}$$

Also,

$$\xi_i^{[\kappa+1]}(t|t) = \xi_i^{*[\kappa]}(t|t). \tag{8}$$

At this point one can proceed with (5) and (6) for $\kappa+1$ and likewise repeat the described procedure for the total of K consensus steps.

It is straightforward to show that the consensus parameters $c_{ij}(t)$ in (2) obtained by applying K consensus steps are:

$$[c_{ij}(t)] = C(t) = C^{[K]}(t)\cdot C^{[K-1]}(t)\cdot \ldots \cdot C^{[1]}(t) = \left[\frac{b_{ij}^K \gamma_j^{[1]}(t)}{\sum_{j=1}^n b_{ij}^K \gamma_j^{[1]}(t)}\right], \quad (9)$$

where $[b_{ij}^K] = B^K$ (B to the power of K). Under the usual assumption that $\mathcal{G}$ is strongly connected [4], [14], we select B satisfying (4), and readily obtain:

$$\lim_{K\to\infty} \xi_i^{*[K]}(t|t) = \sum_{j\in\mathcal{J}_i} c_{ij}^{\infty}(t)\xi_j(t|t), \quad (10)$$

where

$$[c_{ij}^{\infty}(t)] = \begin{bmatrix} \frac{w_1\gamma_1(t)}{w_1\gamma_1(t)+\cdots+w_N\gamma_N(t)} & \cdots & \frac{w_N\gamma_N(t)}{w_1\gamma_1(t)+\cdots+w_N\gamma_N(t)} \\ \vdots & \ddots & \vdots \\ \frac{w_1\gamma_1(t)}{w_1\gamma_1(t)+\cdots+w_N\gamma_N(t)} & \cdots & \frac{w_N\gamma_N(t)}{w_1\gamma_1(t)+\cdots+w_N\gamma_N(t)} \end{bmatrix},$$

which represents the desired result from (3)

III. Optimizing Convergence Speed

In order to have a complete algorithm, we have to define a constant matrix B in (5) and (7) according to (4), for a given network topology and choice of w. Since our concern is to achieve *the fastest convergence* to consensus (maximally reducing the communication efforts), the formal problem to be solved is [25]:

$$\begin{aligned} &\text{minimize } \ r(B-\mathbf{1}w^T) \ \text{ with respect to } B \\ &\text{subject to } w^T B = w^T \text{ and } B\mathbf{1}=\mathbf{1}, \end{aligned} \quad (11)$$

where $r(\cdot)$ is equal to either the spectral radius $\rho(\cdot)$ or the spectral norm $\|\cdot\|_S$ (see [25]). It is possible to reduce the number of degrees of freedom by introducing additional, practically justifiable constraints on A.

Let B have equal non-diagonal non-zero elements in each column (an analogous row-wise assumption can be used as well), *i.e.*, $B =$

$$\begin{bmatrix} \sum_{k,k\neq 1}(1-\alpha c_k a_{1k}) & \alpha c_2 a_{12} & \cdots & \alpha c_n a_{1n} \\ \alpha c_1 a_{21} & \sum_{k,k\neq 2}(1-\alpha c_k a_{2k}) & \cdots & \alpha c_n a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ \alpha c_1 a_{n1} & \alpha c_2 a_{n2} & \cdots & \sum_{k,k\neq n}(1-\alpha c_k a_{nk}) \end{bmatrix},$$

where α represents a scaling factor, c_j are the elements of the normalized vector of column values

$$c = [c_1 \quad \cdots \quad c_N]^T,$$

so that $\mathbf{1}^T c = 1$, and a_{ij} are the elements of the adjacency matrix A of the digraph $\mathcal{G}$.

Formally, first we have to solve the standard equation

$$w^T B = w^T, \quad (12)$$

for the unknown parameters in B under the given topology constraints. It is straightforward to show that (12) reduces to

$$\tilde{L}_w^T c = 0, \quad (13)$$

where $\tilde{L}_w$ is in the form of a weighted Laplacian matrix of the underlying graph $\mathcal{G}$ [28], *i.e.*, $\tilde{L}_w =$

$$\begin{bmatrix} \sum_{k,k\neq 1} w_k a_{1k} & -w_2 a_{12} & \cdots & -w_n a_{1n} \\ -w_1 a_{21} & \sum_{k,k\neq 2} w_k a_{2k} & \cdots & -w_n a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ -w_1 a_{n1} & -w_2 a_{n2} & \cdots & \sum_{k,k\neq n} w_k a_{nk} \end{bmatrix}.$$

If $\mathcal{G}$ is strongly connected (which is a standard topology assumption within similar contexts [14]), we have $\mathrm{rank}(\tilde{L}_w) = N-1$ [4], [14]. Thus, (13) represents a system of $N-1$ linearly independent equations with N unknowns, which, combined with $\mathbf{1}^T c = 1$, yields a unique solution for c.

On the other hand, it is easy to see that

$$B = I - \alpha \tilde{L}_c, \quad (14)$$

where $\tilde{L}_c$ is in the form of another weighted Laplacian matrix, *i.e.*, $\tilde{L}_c =$

$$\begin{bmatrix} \sum_{k,k\neq 1} c_k a_{1k} & -c_2 a_{12} & \cdots & -c_n a_{1n} \\ -c_1 a_{21} & \sum_{k,k\neq 2} c_k a_{2k} & \cdots & -c_n a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ -c_1 a_{n1} & -c_2 a_{n2} & \cdots & \sum_{k,k\neq n} c_k a_{nk} \end{bmatrix}.$$

Now, we can directly generalize the results from [25] to conclude that (4) is achieved for

$$0 < \alpha < \frac{2}{\lambda_1(\tilde{L}_c)}, \quad (15)$$

where $\lambda_i(\cdot)$ denotes the i-th largest eigenvalue of the argument matrix. Further, the optimal choice of α in view of (11) is

$$\alpha = \frac{2}{\lambda_1(\tilde{L}_c) + \lambda_{N-1}(\tilde{L}_c)}. \quad (16)$$

It is to be noted that simple bounds that give choices that do not require exact knowledge of the Laplacian spectrum can be used as well [25].

IV. Simulation Examples

We shall consider a sensor network with $N = 10$ nodes, randomly dispersed within a square area, and connected if the corresponding inter-distance is less than half of the side of the square (the so-called Geometric Random Graph topology). In order to show that our proposed methodology is applicable to the general case of digraphs, approximately 25% of two-way connections is randomly made one-way. One obtained realization of such a network, together with its communication topology, is shown in Fig 1. The corresponding adjacency matrix is

$$A = \begin{bmatrix} 0&0&1&1&0&1&1&0&0&1 \\ 1&0&0&1&0&0&1&0&0&0 \\ 1&1&0&0&0&0&1&0&0&0 \\ 1&1&0&0&0&0&1&0&0&0 \\ 0&0&0&0&0&0&0&1&0&1 \\ 0&0&1&0&0&0&1&1&0&1 \\ 1&1&0&0&0&1&0&0&0&1 \\ 0&0&0&0&1&1&0&0&0&0 \\ 1&0&0&0&0&0&0&0&0&1 \\ 1&0&0&0&1&1&1&0&1&0 \end{bmatrix}.$$

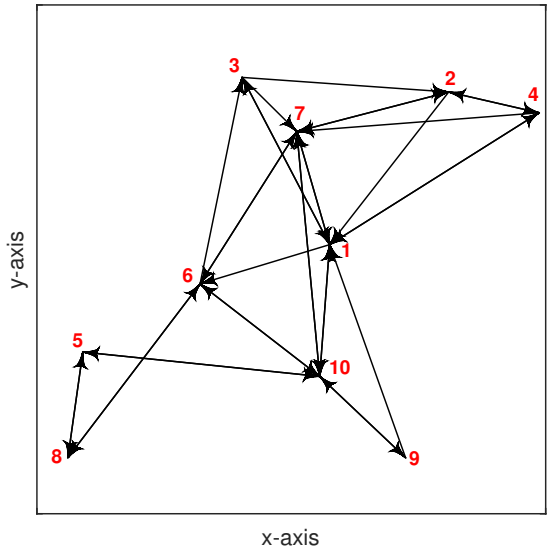


Fig. 1. Sensor network with its communication topology.

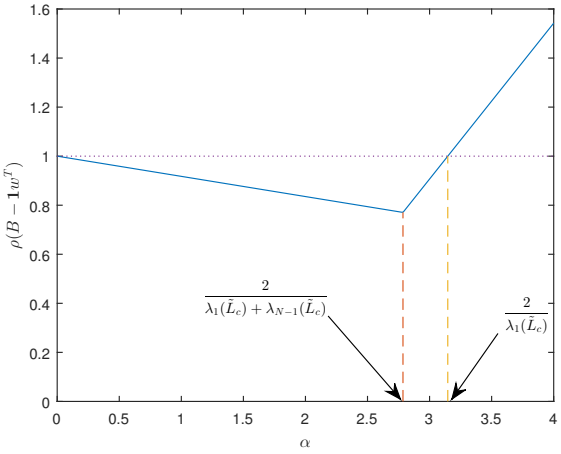


Fig. 2. Spectral radius of $B-\mathbf{1}w^T$ for different values of α.

First, we shall consider the average consensus problem, *i.e.*, the case when we have

$$w = [\,0.1\ 0.1\ 0.1\ 0.1\ 0.1\ 0.1\ 0.1\ 0.1\ 0.1\ 0.1\,]^T .$$

The corresponding weighted Laplacian matrix is

$$\tilde{L}_w = \begin{bmatrix} 0.6 & 0 & -0.1 & -0.1 & 0 & -0.1 & -0.1 & 0 & 0 & -0.1 \\ -0.1 & 0.3 & 0 & -0.1 & 0 & 0 & -0.1 & 0 & 0 & 0 \\ -0.1 & -0.1 & 0.2 & 0 & 0 & 0 & -0.1 & 0 & 0 & 0 \\ -0.1 & -0.1 & 0 & 0.2 & 0 & 0 & -0.1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.2 & 0 & 0 & -0.1 & 0 & -0.1 \\ 0 & 0 & -0.1 & 0 & 0 & 0.4 & -0.1 & -0.1 & 0 & -0.1 \\ -0.1 & -0.1 & 0 & 0 & 0 & -0.1 & 0.6 & 0 & 0 & -0.1 \\ 0 & 0 & 0 & 0 & -0.1 & -0.1 & 0 & 0.2 & 0 & 0 \\ -0.1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.1 & -0.1 \\ -0.1 & 0 & 0 & 0 & -0.1 & -0.1 & -0.1 & 0 & -0.1 & 0.5 \end{bmatrix} .$$

By coupling (13) (for example, taking the first $N-1$ independent rows) with $\mathbf{1}^T c = 1$, we come to

$$\begin{bmatrix} 0.6 & -0.1 & -0.1 & -0.1 & 0 & 0 & -0.1 & 0 & -0.1 & -0.1 \\ 0 & 0.3 & -0.1 & -0.1 & 0 & 0 & -0.1 & 0 & 0 & 0 \\ -0.1 & 0 & 0.2 & 0 & 0 & -0.1 & 0 & 0 & 0 & 0 \\ -0.1 & -0.1 & 0 & 0.2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.2 & 0 & 0 & -0.1 & 0 & -0.1 \\ -0.1 & 0 & 0 & 0 & 0 & 0.4 & -0.1 & -0.1 & 0 & -0.1 \\ -0.1 & -0.1 & -0.1 & -0.1 & 0 & -0.1 & 0.6 & 0 & 0 & -0.1 \\ 0 & 0 & 0 & 0 & -0.1 & -0.1 & 0 & 0.2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.1 & -0.1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} \cdot c = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} ,$$

and readily obtain

$$c = [\,0.08\ 0.08\ 0.11\ 0.11\ 0.10\ 0.09\ 0.06\ 0.09\ 0.18\ 0.10\,]^T .$$

The corresponding weighted Laplacian matrix is given by $\tilde{L}_c =$

$$\begin{bmatrix} 0.47 & 0 & -0.11 & -0.11 & 0 & -0.09 & -0.06 & 0 & 0 & -0.10 \\ -0.08 & 0.25 & 0 & -0.11 & 0 & 0 & -0.06 & 0 & 0 & 0 \\ -0.08 & -0.08 & 0.22 & 0 & 0 & 0 & -0.06 & 0 & 0 & 0 \\ -0.08 & -0.08 & 0 & 0.22 & 0 & 0 & -0.06 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.19 & 0 & 0 & -0.09 & 0 & -0.10 \\ 0 & 0 & -0.11 & 0 & 0 & 0.36 & -0.06 & -0.09 & 0 & -0.10 \\ -0.08 & -0.08 & 0 & 0 & 0 & -0.09 & 0.35 & 0 & 0 & -0.10 \\ 0 & 0 & 0 & 0 & -0.10 & -0.09 & 0 & 0.19 & 0 & 0 \\ -0.08 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.18 & -0.10 \\ -0.08 & 0 & 0 & 0 & -0.10 & -0.09 & -0.06 & 0 & -0.18 & 0.50 \end{bmatrix} .$$

Now that we have $\tilde{L}_c$, we can design the matrix B from (14) by choosing the scaling factor α. Plotting the spectral radius of $B-\mathbf{1}w^T$ for different values of α (Fig. 2), we see that the minimal value and fastest convergence is obtained exactly for $\alpha = 2/(\lambda_1(\tilde{L}_c)+\lambda_{N-1}(\tilde{L}_c)) = 2.79$, while the stability boundary is reached for $\alpha = 2/\lambda_1(\tilde{L}_c) = 3.15$. The resulting matrix B providing the fastest consensus is

$$B = \begin{bmatrix} -0.31 & 0 & 0.30 & 0.30 & 0 & 0.25 & 0.16 & 0 & 0 & 0.28 \\ 0.22 & 0.31 & 0 & 0.30 & 0 & 0 & 0.16 & 0 & 0 & 0 \\ 0.22 & 0.23 & 0.39 & 0 & 0 & 0 & 0.16 & 0 & 0 & 0 \\ 0.22 & 0.23 & 0 & 0.39 & 0 & 0 & 0.16 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.46 & 0 & 0 & 0.26 & 0 & 0.28 \\ 0 & 0 & 0.30 & 0 & 0 & -0.01 & 0.16 & 0.26 & 0 & 0.28 \\ 0.22 & 0.23 & 0 & 0 & 0 & 0.25 & 0.02 & 0 & 0 & 0.28 \\ 0 & 0 & 0 & 0 & 0.27 & 0.25 & 0 & 0.48 & 0 & 0 \\ 0.22 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.50 & 0.28 \\ 0.22 & 0 & 0 & 0 & 0.27 & 0.25 & 0.16 & 0 & 0.50 & -0.40 \end{bmatrix} .$$

We shall further consider the weighted consensus case. For example, let the weight vector be

$$w = [\,0.05\ 0.16\ 0.08\ 0.12\ 0.22\ 0.06\ 0.09\ 0.06\ 0.06\ 0.11\,]^T .$$

Similarly as above, we obtain

$$c = [\,0.04\ 0.11\ 0.16\ 0.12\ 0.21\ 0.07\ 0.05\ 0.07\ 0.08\ 0.10\,]^T ,$$

and the fastest convergence is achieved for $\alpha = 3.1$, while the stability boundary is reached for $\alpha = 3.35$. The resulting matrix B providing the fastest consensus is

$$B = \begin{bmatrix} -0.52 & 0 & 0.49 & 0.36 & 0 & 0.21 & 0.15 & 0 & 0 & 0.31 \\ 0.11 & 0.38 & 0 & 0.36 & 0 & 0 & 0.15 & 0 & 0 & 0 \\ 0.11 & 0.33 & 0.40 & 0 & 0 & 0 & 0.15 & 0 & 0 & 0 \\ 0.11 & 0.33 & 0 & 0.40 & 0 & 0 & 0.15 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.48 & 0 & 0 & 0.21 & 0 & 0.31 \\ 0 & 0 & 0.49 & 0 & 0 & -0.16 & 0.15 & 0.21 & 0 & 0.31 \\ 0.11 & 0.33 & 0 & 0 & 0 & 0.21 & 0.04 & 0 & 0 & 0.31 \\ 0 & 0 & 0 & 0 & 0.65 & 0.21 & 0 & 0.14 & 0 & 0 \\ 0.11 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.57 & 0.31 \\ 0.11 & 0 & 0 & 0 & 0.65 & 0.21 & 0.15 & 0 & 0.25 & -0.38 \end{bmatrix} .$$

In order to show the effectiveness of our solution, we shall compare the aggregated consensus results (powers of the matrix B) in the optimized convergence speed case (with $\alpha = 2/(\lambda_1(\tilde{L}_c)+\lambda_{N-1}(\tilde{L}_c))$, shown in Fig. 3), to one example of the non-optimized case (with $\alpha = 1$, shown in Fig. 4). It can be clearly seen that the desired asymptotic weights are reached much faster with the optimized solution.

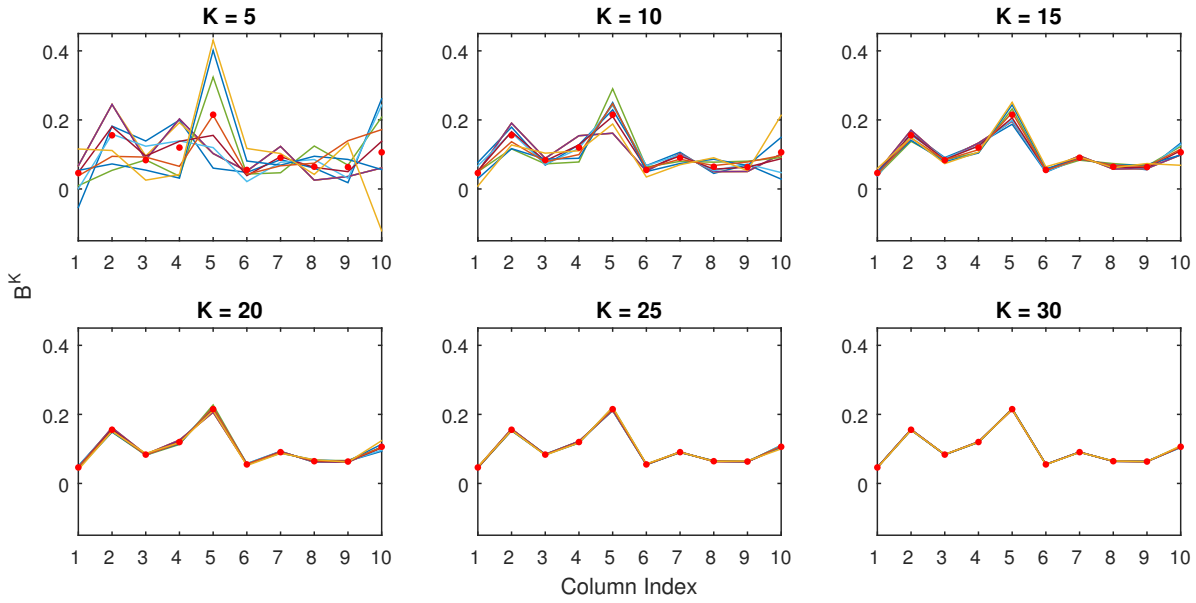


Fig. 3. Values of all rows of B^K when B is optimized; asymptotic weights w are illustrated with red dots.

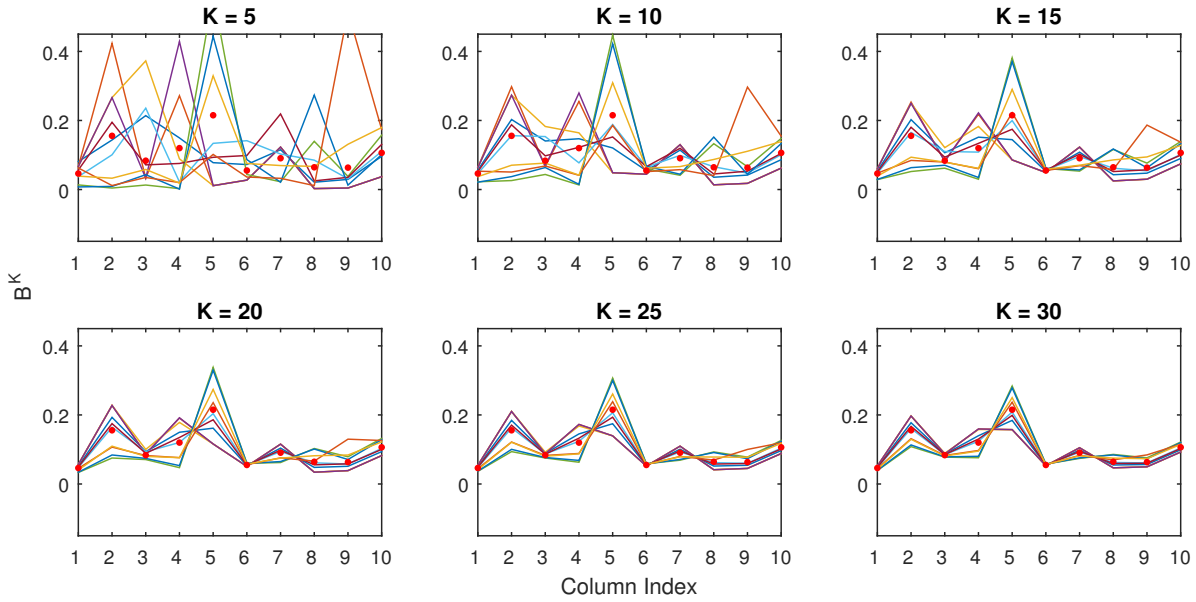


Fig. 4. Values of all rows of B^K when B is not optimized; asymptotic weights w are illustrated with red dots.

Finally, as a small proof-of-concept example, we shall assume that the nodes states directly reflect the received scalar measurements; for all the nodes except nodes 5 and 8 these are randomly sampled from the Gaussian distribution with $\mu = 100$ and $\sigma = 10$ (we assume that these are low quality measurements with corresponding weights $\gamma_i = 0.001$), while for nodes 5 and 8 the measurements are randomly sampled from the Gaussian distribution with $\mu = 10$ and $\sigma = 1$ (high quality measurements with corresponding weights $\gamma_i = 0.99$). We adopt equal *a priori* weights w_i for all i. Therefore, we expect our scheme to achieve consensus value around 10, influenced primarily by nodes 5 and 8. We see in Figs. (5) and (6) that this indeed happens. Also, it can be seen that the optimized case offers a satisfying solution already for a number of consensus steps of 3, while similar amount of disagreement between the nodes in the non-optimized case is reached only just for a total number of consensus steps of 10.

V. Conclusion

In this paper a procedure for designing the adaptive multi-step consensus communication scheme in sensor networks is

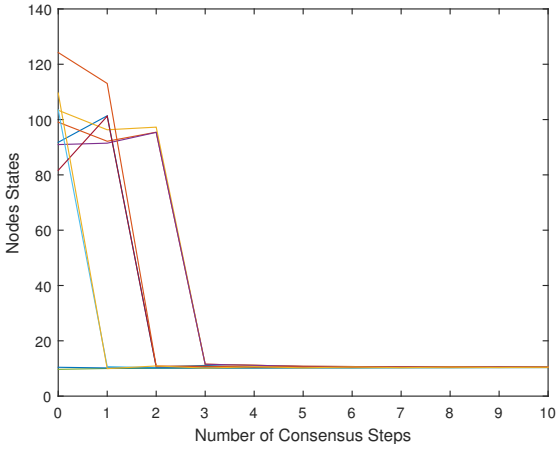


Fig. 5. Nodes' states for different number of consensus steps - optimized case.

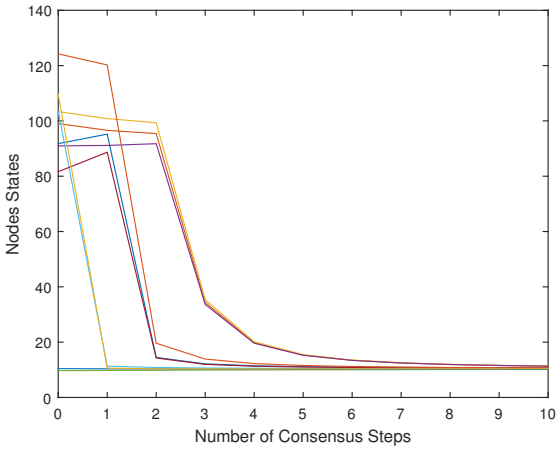


Fig. 6. Nodes' states for different number of consensus steps - non-optimized case.

proposed, exhibiting the fastest convergence to the desirable asymptotic values, and thus minimizing the resulting communication burden. The scheme allows both *a priori* and real-time weightings of the local processing results connected to different nodes in the network, influencing the asymptotic behavior of the consensus protocol in appropriate manner.

Further work can be oriented towards exploring the design possibilities of analogous randomized schemes, similarly as in [29], [12], [14].

REFERENCES

[1] R. Olfati-Saber, A. Fax, and R. Murray, "Consensus and cooperation in networked multi-agent systems," *Proceedings of the IEEE*, vol. 95, no. 1, pp. 215–233, 2007.

[2] R. Olfati-Saber, "Distributed Kalman filtering for sensor networks," in *46th IEEE Conference on Decision and Control*, 2007, pp. 5492–5498.

[3] W. Ren and R. W. Beard, *Distributed Consensus in Multi-vehicle Cooperative Control: Theory and Applications*. Springer Publishing Company, Incorporated, 2007.

[4] W. Ren and R. Beard, "Consensus seeking in multi-agent systems using dynamically changing interaction topologies," *IEEE Transactions on Automatic Control*, vol. 50, no. 5, pp. 655–661, 2005.

[5] R. Olfati-Saber and N. F. Sandell, "Distributed tracking in sensor networks with limited sensing range," in *American Control Conference*, 2008, pp. 3157–3162.

[6] S. S. Stanković, M. S. Stanković, and D. M. Stipanović, "Consensus based overlapping decentralized estimator," *IEEE Trans. Autom. Control*, vol. 54, pp. 410–415, 2009.

[7] ——, "Consensus based overlapping decentralized estimation with missing observations and communication faults," *Automatica*, vol. 45, no. 6, pp. 1397–1406, 2009.

[8] R. M. G. Ferrari, T. Parisini, and M. L. Polycarpou, "Distributed fault diagnosis with overlapping decompositions and consensus filters," in *Proc. Amer. Contr. Conf.*, 2007.

[9] N. Ilić, M. Stanković, and S. Stanković, "Consensus based overlapping decentralized observer for fault detection and isolation," in *Proc. Melecon 2010 Conf.*, 2010.

[10] P. Braca, S. Marano, and V. Matta, "Enforcing consensus while monitoring the environment in wireless sensor networks," *IEEE Trans. Signal Processing*, vol. 56, pp. 3375–3380, 2008.

[11] P. Braca, S. Marano, V. Matta, and P. Willett, "Asymptotic optimality of running consensus in testing binary hypotheses," *IEEE Trans. Signal Processing*, vol. 58, pp. 814–825, 2010.

[12] S. S. Stanković, N. Ilić, M. Stanković, and K. H. Johansson, "Distributed change detection based on a consensus algorithm," *IEEE Transactions on Signal Processing*, vol. 59, pp. 5686–5697, 12 2011.

[13] N. Ilić, S. S. Stanković, M. Stanković, and K. H. Johansson, "Consensus based distributed change detection using generalized likelihood ratio methodology," *Signal Processing*, vol. 92, 06 2011.

[14] M. S. Stanković, N. Ilić, and S. S. Stanković, "Distributed stochastic approximation: Weak convergence and network design," *IEEE Transactions on Automatic Control*, vol. 61, no. 12, pp. 4069–4074, 2016.

[15] N. Ilić, M. S. Stanković, and S. S. Stanković, "Adaptive consensus-based distributed target tracking in sensor networks with limited sensing range," *IEEE Transactions on Control Systems Technology*, vol. 22, no. 2, pp. 778–785, 2014.

[16] K. O. A. Ali, N. Ilić, M. S. Stanković, and S. S. Stanković, "Consensus-based distributed adaptive target tracking in camera networks using integrated probabilistic data association," *EURASIP Journal on Advances in Signal Processing*, vol. 2018, no. 1, p. 13, 2018. [Online]. Available: https://doi.org/10.1186/s13634-018-0534-z

[17] ——, "Distributed target tracking in sensor networks using multi-step consensus," *IET Radar, Sonar Navigation*, vol. 12, no. 9, pp. 998–1004, 2018.

[18] A. T. Kamal, J. A. Farrell, and A. K. Roy-Chowdhury, "Information weighted consensus filters and their application in distributed camera networks," *IEEE Transactions on Automatic Control*, vol. 58, no. 12, pp. 3112–3125, 2013.

[19] A. T. Kamal, J. H. Bappy, J. A. Farrell, and A. K. Roy-Chowdhury, "Distributed multi target tracking and data association in vision networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, no. 7, pp. 1397–1410, 2016.

[20] G. Battistelli, L. Chisci, C. Fantacci, A. Farina, and A. Graziano, "Consensus CPHD filter for distributed multitarget tracking," *IEEE Journal of Selected Topics in Signal Processing*, vol. 7, no. 3, pp. 508–520, 2013.

[21] B. Sinopoli, L. Schenato, M. Franceschetti, K. Poolla, M. I. Jordan, and S. S. Sastry, "Kalman filtering with intermittent observations," *IEEE Transactions on Automatic Control*, vol. 49, no. 9, pp. 1453–1464, Sept 2004.

[22] N. F. Sandell and R. Olfati-Saber, "Distributed data association for multi-target tracking in sensor networks," in *47th IEEE Conference on Decision and Control*, 2008, pp. 1085–1090.

[23] N. Ilić, K. O. Al-Ali, M. Stanković, and S. Stanković, "Distributed multi-target tracking in camera networks using multi-step consensus," in *Proc. 4th IcETRAN Conf.*, 2017.

[24] A. Olshevsky and J. N. Tsitsiklis, "Convergence speed in distributed consensus and averaging," *SIAM Review*, vol. 53, no. 4, pp. 747–772, 2011. [Online]. Available: https://doi.org/10.1137/110837462

[25] L. Xiao and S. Boyd, "Fast linear iterations for distributed averaging," *Systems and Control Letters*, vol. 53, no. 1, pp. 65–78, 2004.

[26] J. N. Tsitsiklis, D. P. Bertsekas, and M. Athans, "Distributed asynchronous deterministic and stochastic gradient optimization algorithms," *IEEE Trans. Autom. Control*, vol. 31, pp. 803–812, 1986.

[27] R. A. Horn and C. A. Johnson, *Matrix Analysis*. Cambridge, England: Cambridge University Press, 1985.

[28] C. Godsil and G. Royle, *Algebraic Graph Theory*. New York: Springer Verlag, 2001.

[29] N. Ilić and S. S. Stanković, "Communication gains design in a consensus based distributed change detection algorithm," in *Proc. 8th European Workshop on Advanced Control and Diagnosis*, 2010.