FUZIJA SENZORA I KALMAN FILTAR

Celio Cega, Petar

Master's thesis / Specijalistički diplomski stručni

2021

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Split / Sveučilište u Splitu**

Permanent link / Trajna poveznica: https://urn.nsk.hr/urn:nbn:hr:228:907289

Rights / Prava: In copyright / Zaštićeno autorskim pravom.

Download date / Datum preuzimanja: **2024-11-30**

Repository / Repozitorij:

Repository of University Department of Professional Studies

zir.nsk.hr

UNIVERSITY OF SPLIT

dabar
DIGITALNI AKADEMSKI ARHIVI I REPOZITORIJI

SVEUČILIŠTE U SPLITU

SVEUČILIŠNI ODJEL ZA STRUČNE STUDIJE

Specijalistički diplomski stručnij studij Elektrotehnike

PETAR CELIO CEGA

Z A V R Š N I R A D

FUZIJA SENZORA I KALMAN FILTAR

Split, rujan 2021.

SVEUČILIŠTE U SPLITU

SVEUČILIŠNI ODJEL ZA STRUČNE STUDIJE

Specijalistički diplomski stručnij studij Elektrotehnike

Predmet: Digitalna obrada signala

Z A V R Š N I R A D

Kandidat: Petar Celio Cega

Naslov rada: Fuzija senzora i Kalman filtar

Mentor: dr.sc.Barbara Džaja

Komentor: dr.sc.Tonko Kovačević

Split, rujan 2021.

SADRŽAJ

SAŽETAK 3
Fuzija senzora i Kalman filtar 3
SUMMARY 3
Sensor fusion and Kalman filter 3
1. UVOD 4
2. FUZIJA SENZORA 6
2.1. Osnovni procesi spajanja podataka 6
2.2. Kategorije procesa fuzije senzora 8
2.3. Low-level fuzija i Kalman filtar 8
3. KALMAN FILTAR 9
3.1. Algoritam i dijagram toka 9
3.2. Iterativan proces kroz tri jednadžbe 11
3.2.1. Kalman dobitak 11
3.2.2. Trenutna procjena 12
3.2.3. Trenutačna greška u procjeni 12
4. KALMAN FILTAR – VIŠEDIMENZIJSKI MODEL 14
4.1. Dijagram toka procesa 14
4.2. Parametarske jednadžbe 15
4.3. Definiranje procesa 16
4.4. Jednodimenzijski sustav 17
4.5. Dvodimenzijski sustav 20
4.5.1. Matrica procesne kovarijance 23
4.5.2. Matrica kovarijance stanja 25
4.6. Primjer u dvodimenzionalnom prostoru 26
4.6.1. Parametri 27
4.6.2. Prvi ciklus 28
4.6.3. Drugi ciklus 31
5. PRAKTIČNA IMPLEMENTACIJA KALMAN FILTRA U JEDNOJ DIMENZIJI 34
5.1. Sklopovlje 34
5.2. Sklopovsko i programsko rješenje 35

5.3. Programski kod 38
5.3.1. Glavni program 38
5.3.2. Biblioteka (JednodimenzijskiKalmanFilter.cpp) 40
5.3.3. Header dokument za biblioteku (JednodimenzijskiKalmanFilter.h) 41
6. ZAKLJUČAK 42
7. POPIS SLIKA I TABLICA 43
8. LITERATURA 44

SAŽETAK

Fuzija senzora i Kalman filtar

Kroz ovaj rad objašnjeno je na koji način moderan elektronički sustav koristeći proces zvan *Kalman filtar* obrađuje podatke iz više različitih osjetnika u svrhu poboljšanja kvalitete informacija o prostoru u kojem se nalazi. Dan je uvod u procesni model *Kalman filtra*, detaljno pojašnjen način na koji predviđa buduće stanje sustava te je kroz dvodimenzionalni primjer *Kalman filtra* prikazano kako dva potpuno različita osjetnika mogu raditi zajedno u cilju unaprijeđenja preciznosti, robusnosti i kvalitete dobivenih vrijednosti.

Ključne riječi: Kalman filtar, fuzija senzora, arduino

SUMMARY

Sensor fusion and Kalman filter

Throughout this thesis it is explained how a modern electronics system uses Kalman filtering to process the data from multi-sensor systems in a bid to improve the quality of information about its surroundings. The details of Kalman filtering process are demonstrated on predictive state model. Comprehensive two-dimensional example demonstrates in which way two heterogeneous sensors can be paired to work together and deliver data which is more precise, robust and of higher quality.

Keywords: Kalman filter, sensor fusion, arduino

1. UVOD

Sav život na planetu oslanja se na mogućnost obrade podataka iz više osjetnih izvora. Spoj različitih osjetila i ispravna interpretacija osjetnih podataka usko je vezana uz proces prirodne selekcije. Kombiniranjem osjetila sluha i njuha (npr. šuškanje trave i nepoznat miris u zraku) mogu upozoriti na opasnost s većom preciznošću nego pojedini osjet zasebno.

Informacije koje osjetila primaju prenose se do jedinice za obradu podataka, kao što je primjerice ljudski mozak, gdje se obrađuju. Obrađeni podaci nam mogu poslužiti u svrhu predviđanja buduće situacije, ponekad i sa većom preciznošću nego "sirovi" podaci koje prihvaćamo u točno određenom trenutku.

Slično biološkim sustavima, elektronički sustavi koriste osjetnike ili senzore kako bi primili informacije iz okoline. U mnogim sustavima, senzori su prva instanca na osnovu čijih se očitanja donose odluke koje djeluju na cijeli sustav. To je jedan od razloga zbog kojeg je od preteče važnosti da informacije koje senzori prikupljaju i prosljeđuju budu što točnije.

Kombinacijom više senzora istog tipa obično se dolazi do preciznijih mjerenja. Međutim, senzori istog tipa gotovo uvijek imaju ista ili slična ograničenja.

Kao primjer može se navesti GPS senzor koji prima podatke frekvencijom od 1Hz, s točnošću od ±4m. U slučaju da imamo dva senzora takvog tipa, podaci koje nam takvi senzori dostavljaju dostatni su za primjerice procjenu pozicije broda na otvorenom moru. Zbog velikog otklona (±4m) te spore frekvencije osvježavanja (*1Hz*) takvi senzori, ma koliki njihov ukupan broj u sustavu bio, nisu dovoljni za vožnju autonomnim vozilom kroz gusti promet. Ako bi se GPS sustavu senzora pridružili senzori za akceleraciju i orijentaciju, takav sustav bio bi točniji, robusniji i daleko manje podložan greškama.

Podaci dobiveni iz više različitih tipova senzora mogu se "spojiti" procesom zvanim **fuzija senzora** (*eng. Sensor fusion*). Najčešće korišten proces implementacije *fuzije senzora* izveden je koristeći višedimenzionalni **Kalman filtar**.

Kako je spajanje podataka iz različitih senzora opsegom zahtjevniji proces, zbog višedimenzionalnosti rezultata i kombinacije istih, ovaj rad obuhvatiti će teorijski dio i samu izvedbu jednostavnog *Kalman filtera* na jednodimenzionalnom skupu podataka.

Podaci dobiveni sa senzora za akceleraciju i otklon na jednoj osi biti će prihvaćeni i obrađeni, te će se dobiveni rezultati obrade predstaviti zajedno sa izmjerenim rezultatima kako bi ih mogli usporediti. Rezultati dobiveni koristeći *Kalman filtar* biti će bliže stvarnoj vrijednosti i s manjim odstupanjima od stvarne vrijednosti nego direktno dobiveni podaci sa senzora.

Skup obrađenih podataka dobivenih *Kalman filtrom* ostvariti će manje kaotično ponašanje sustava te minimalizirati utjecaj šuma na sustav.

2. FUZIJA SENZORA

2.1. Osnovni procesi spajanja podataka

Senzori su uređaji koji su izravno u interakciji s okolinom i koji su izvor svih ulaznih podataka u sustavu za *višesenzorsku fuziju podataka*. Fizički element koji stupa u interakciju s okolinom poznat je kao senzorski element i može biti bilo koji uređaj koji može opaziti fizičko svojstvo ili svojstvo okoliša, poput topline, svjetla, zvuka, tlaka, magnetizma ili gibanja. Da bi bio koristan, senzor mora preslikati vrijednost svojstva ili atributa u kvantitativno mjerenje na dosljedan i predvidljiv način. Senzori ne moraju mjeriti samo fizičko svojstvo, već moraju obavljati i dodatne funkcije. Ove se funkcije mogu opisati u smislu *kompenzacije, obrade informacija, komunikacije i integracije*:

- **Naknada** (*eng. Compensation*) se odnosi na sposobnost senzora da detektira i reagira na promjene u okolini kroz samo-dijagnostičke testove, samo-kalibraciju i prilagodbu.
- **Obrada informacija** (*eng. Information processing*) se odnosi na procese kao što su kondicioniranje signala, smanjenje podataka, otkrivanje događaja i donošenje odluka, koji poboljšavaju informacijski sadržaj sirovih mjerenja senzora.
- **Komunikacija** se odnosi na upotrebu standardiziranog sučelja i standardiziranog komunikacijskog protokola za prijenos informacija između senzora i vanjskog svijeta.
- **Integracija** se odnosi na spajanje procesa "osjeta" i računanja na istom silicijskom čipu. Često se provodi tehnologijom mikroelektromehaničkih sustava (MEMS). Praktična implementacija takvog senzora poznata je kao pametan ili inteligentan senzor

Spajanje podataka (*eng. Data fusion*) ili kombiniranje podataka iz različitih izvora pomaže u kreiranju modela, te procjeni i predviđanju ciljanog objekta ili okoliša.

U procesima prikupljanja i obrade podataka razlikuju se dva temeljna pojma:

1. **Informacijska fuzija** (*eng. Information fusion*) koja se koristi za opisivanje spajanja bilo kakvih podatkovnih informacija, te informacija izvora podataka.
2. **Senzorska fuzija** (*eng. Sensor fusion*) predstavlja kombinaciju senzorskih podataka ili informacija izvedenih iz senzorskih podataka. Krajnja informacija dobivena na takav

način je korisnija nego što bi bila da je izvedena od svakog senzora zasebno. Informacija dobivena procesom *senzorske fuzije* može biti iz homogenog ili heterogenog sustava senzora, tj. senzorskih informacija.

Fuzija senzora može se smjestiti pod pojam **direktne fuzije**. *Direktna fuzija* obuhvaća podatke iz skupa homogenih ili heterogenih senzora, te prethodnih vrijednosti senzorskih podataka. Treba napomenuti da *fuzija senzora* nije univerzalna metoda. Niti će mnogo loših podataka dati dobre podatke, niti će mnogo senzora niske kvalitete moći zamjeniti senzore visoke kvalitete.

2.2. Kategorije procesa fuzije senzora

Procesi *fuzije senzora* su obično smješteni u jednu od tri kategorije:

1. **Low-level (ili raw data) fuzija** - kombinira nekoliko izvora istog tipa informacija (*raw preprocessed data* - RPP) kako bi se dobio novi set podataka za koji se očekuje da bi mogao biti korisniji i informativniji nego sami senzorski podaci.
2. **Intermediate-level (mid level ili feature level) fuzija** kombinira različite značajke, kao što su primjerice rubovi, linije, uglovi, teksture i pozicije u mapu značajki. Takva mapa se koristi za segmentaciju slika, detekciju objekata itd.
3. **High-level (ili decision) fuzija** kombinira odluke iz nekoliko eksperata. Njene metode su izglasavanje, fuzzy logic, te određene statističke metode.

2.3. Low-level fuzija i Kalman filtar

Za sustave koji se gibaju, dakle dinamičke sustave, najčešće se koristi *low-level fuzija*. Za obradu i predviđanje podataka u dinamičkom sustavu (kinetičkih i statičkih) jedan od važnijih algoritama u primjeni naziva se **Kalman filtar**. *Kalman filtar* je rekurzivni algoritam kojem je svrha procjena stanja u sustavu. To je iterativan matematički proces koji koristi set jednadžbi i niz podatkovnih ulaza kako bi brzo procijenili pravu vrijednost, poziciju, brzinu itd. nekog objekta čiji parametri se mjere. I to u slučaju kada izmjerene vrijednosti sadrže nepredviđene i/ili nasumične pogreške, nesigurnosti ili varijacije.

Rezultantna vrijednost Kalman filtera ovisi primarno o dvije informacije:

1. **Mjernim rezultatima relevantnih senzora**
2. **Matematičkom modelu sistema** opisanog kako različita stanja ovise međusobno te kako mjerni rezultati ovise o stanjima.

U procesu Kalman filtera točnost mjernih rezultata i model uvijek trebaju biti specificirani.

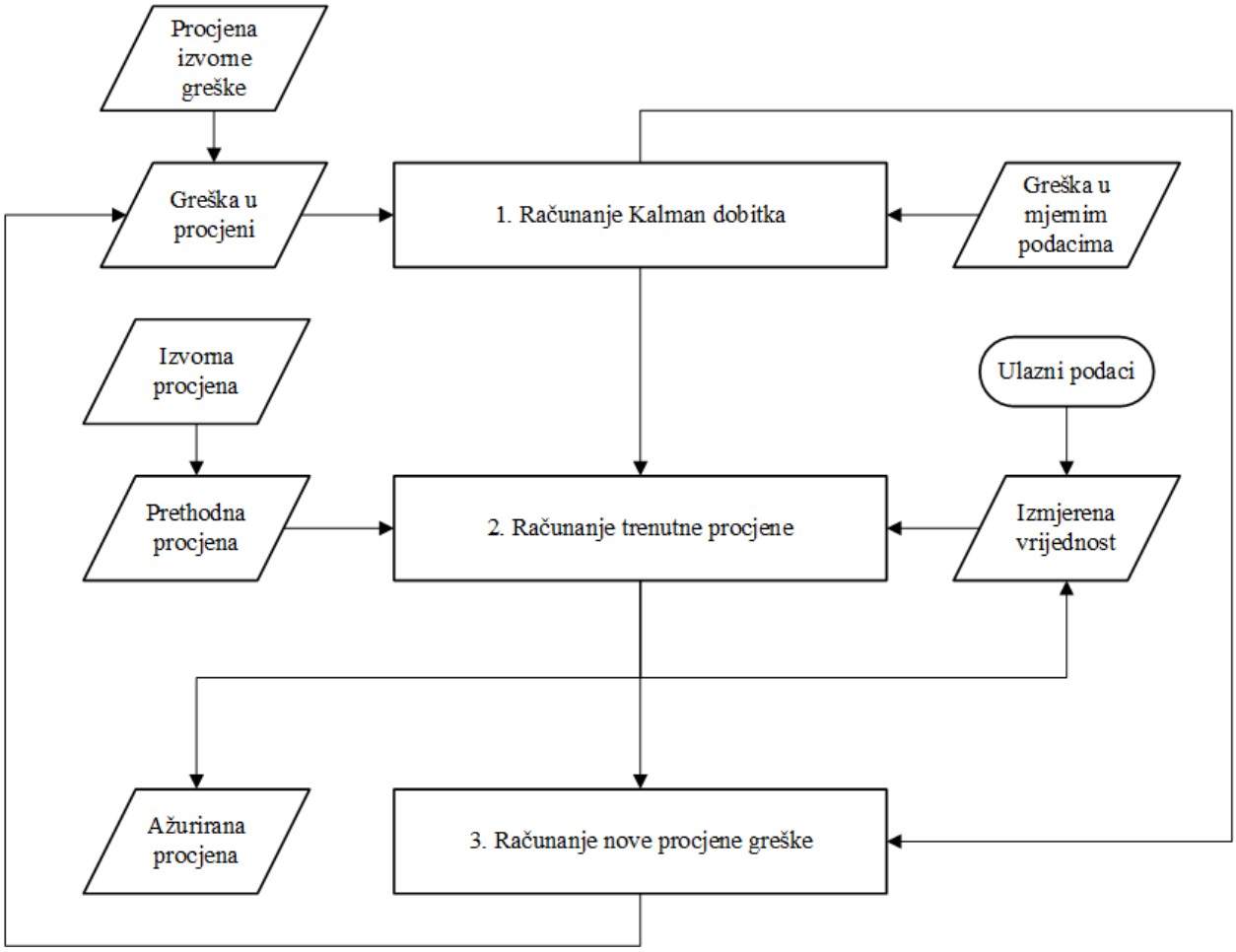
3. KALMAN FILTAR

3.1. Algoritam i dijagram toka

Rekurzivni Kalman filtar opisan je od stanja t_0, te prikazan slijedom:

1. U vremenu t_0 Kalman filtaru je pridružena inicijalna ili početna procjena.
2. Bazirano na matematičkom modelu i inicijalnoj procjeni, predviđa se nova procjena u vremenu t_1. Netočnost predviđene procjene je izračunata uzevši u obzir početnu netočnost i točnost modela (*eng. process noise*).
3. Ispravna mjerenja u vremenu t_1 nam daju novu informaciju o stanjima. Bazirana na točnosti mjerenja (*eng. measurement noise*), te neizvjesnost predviđene procjene, prosuđuju se dva izvora informacija i izračunava se novi t_1 kao novo ažurirana, valjana procjena.
4. U vremenu t_2 nova procjena je predviđena na isti način kao i u koraku 2, ali sada je bazirana na ažuriranoj procjeni iz vremena t_1.

Ako su modeli i prethodne pretpostavke ispravne, Kalman filtar će dostaviti nove optimalne pretpostavke.



Slika 3.1 Iterativan proces jednostavnog Kalman filtra.

3.2. Iterativan proces kroz tri jednadžbe

3.2.1. Kalman dobitak

Kalman dobitak (*eng. Kalman gain*) koristi se da bi odredili u kojoj ćemo mjeri koristiti nove rezultate mjerenja da ažuriramo novu procjenu.

$$KG = \frac{E_{EST}}{E_{EST} + E_{MEA}} \qquad (3.1.)$$

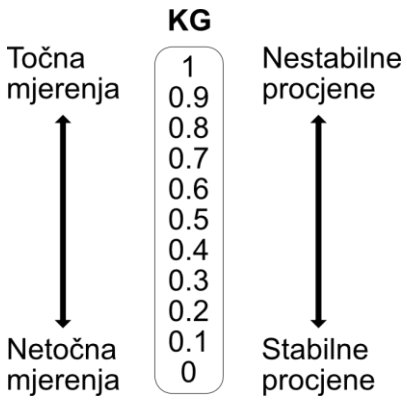
Gdje su:

$KG = Kalman\ dobitak$

$E_{EST} = Greška\ u\ procjeni$

$E_{MEA} = Greška\ u\ mjernim\ podacima$

Kalman dobitak je odnos između greške u procjeni (E_{EST}), te zbroja greške u procjeni i greške u mjerenju. Ako je KG velikog iznosa to znači da je greška u procjeni (E_{EST}) veća od greške u mjerenju (E_{MEA}).



Slika 3.2 Iznos Kalman dobitka i njegov utjecaj na procjenu stanja

3.2.2. Trenutna procjena

Druga jednadžba nam prikazuje *trenutnu procjenu*.

$$EST_t = EST_{t-1} + KG[MEA - EST_{t-1}] \qquad (3.2.)$$

Gdje su:

EST_t – trenutna procjena

EST_{t-1} – protekla procjena

KG – Kalman dobitak

MEA - rezultati mjerenja

Trenutna procjena ovisi o *prethodnoj procjeni* (EST_{t-1}) i umnošku *Kalman dobitka* (KG) sa razlikom *trenutnog mjerenja* (MEA) i *protekle procjene*, tj. procjene prethodnog koraka (EST_{t-1}).

3.2.3. Trenutačna greška u procjeni

Treća jednadžba služi za računanje *trenutačne greške u procjeni*.

$$E_{EST_t} = \frac{(E_{MEA}) \cdot (E_{EST_{t-1}})}{(E_{MEA}) + (E_{EST_{t-1}})} \Rightarrow E_{EST_t} = [1 - KG] \cdot (E_{EST_{t-1}}) \qquad (3.3)$$

Gdje su:

$KG = Kalman\ dobitak$

$E_{EST_{t-1}} = Greška\ protekle\ procjene$

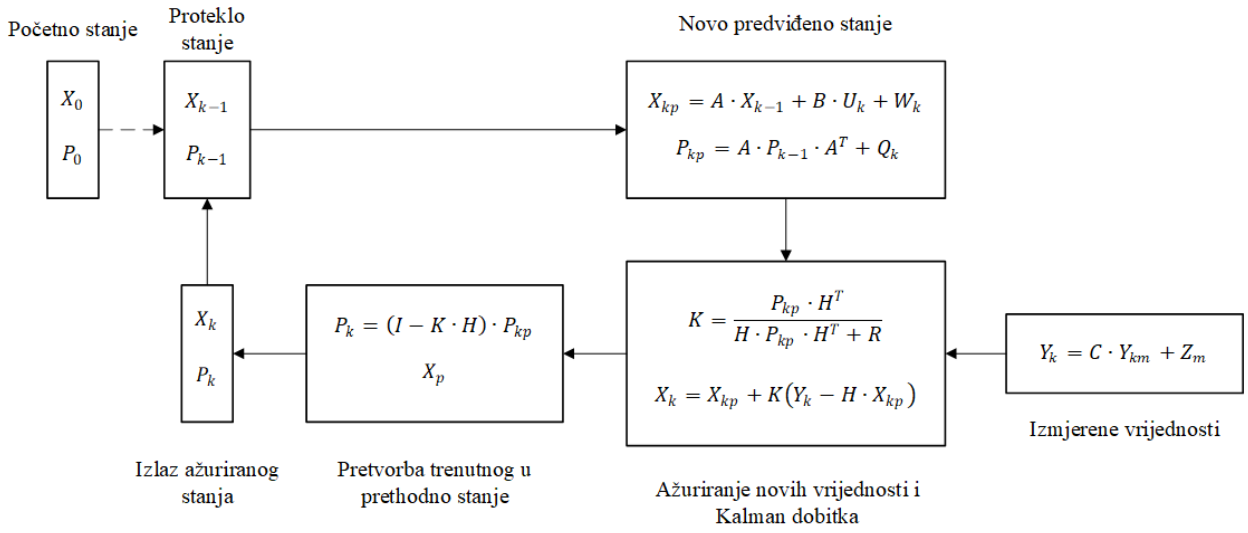
Trenutačna greška u procjeni dobije se umnoškom razlike između vrijednosti broja jedan (*1*) i iznosa *Kalman dobitka* (*1-KG*) te *greške u procjeni* iz prethodne iteracije ($E_{EST_{t-1}}$). Kako je

skup ovih jednadžbi iterativan proces, greška iz ovog koraka će se dalje koristiti za izračunavanje *Kalman dobitka* (*KG*) u sljedećoj iteraciji.

4. KALMAN FILTAR – VIŠEDIMENZIJSKI MODEL

Višedimenzijski model *Kalman filtra* ne razlikuje se procesno od jednodimenzijskog modela. Međutim, kako se uvode dodatne vrijednosti i dimenzije u sam proces potrebno je u model uvesti matrice i matrične kalkulacije kako bi se omogućilo što brže i preglednije računanje parametara.

4.1. Dijagram toka procesa



Slika 4.1 Dijagram toka procesa višedimenzijskog modela Kalman filtra

Gdje su:
$X - \ Matrica\ stanja$
$P - Matrica\ procesne\ kovarijance$
$U - Matrica\ kontrolne\ varijable$
$W - Matrica\ predviđenog\ šuma\ stanja$
$Q - Matrica\ kovarijance\ procesnog\ šuma$
$I - Matrica\ identiteta$
$K - Kalman\ dobitak$
$R - Matrica\ kovarijance\ osjetničkog\ šuma$
$Y - Matrica\ mjernog\ stanja$
$A, B, H, C - Transformacijske\ matrice$

4.2. Parametarske jednadžbe

Početno stanje:

$$X_0, P_0 \qquad (4.1.)$$

Prethodno stanje:

$$X_{k-1}, P_{k-1} \qquad (4.2.)$$

Novo predviđeno stanje:

$$X_{kp} = A \cdot X_{k-1} + B \cdot U_k + W_k \qquad (4.3.)$$

$$P_{kp} = A \cdot P_{k-1} \cdot A^T + Q_k \qquad (4.4.)$$

Izmjerene vrijednosti:

$$Y_k = C \cdot Y_{km} + Z_m \qquad (4.5.)$$

Ažurirano stanje sa novim izmjerenim vrijednostima i Kalman dobitak:

$$K = \frac{P_{kp} \cdot H^T}{H \cdot P_{kp} \cdot H^T + R} \qquad (4.6.)$$

$$X_k = X_{kp} + K\left(Y_k - H \cdot X_{kp}\right) \qquad (4.7.)$$

Pretvorba iz trenutnog stanja u prethodno stanje:

$$P_k = (I - K \cdot H) \cdot P_{kp} \qquad (4.8.)$$

$$X_p \qquad (4.9.)$$

Izlazne vrijednosti ažuriranog stanja:

$$X_k, P_k \qquad (4.10.)$$

4.3. Definiranje procesa

Matrica stanja

$$X_{kp} = A \cdot X_{k-1} + B \cdot U_k + W_k \qquad (4.3.)$$

Gdje su:

$X - Matrica\ stanja$

$U - Matrica\ kontrolne\ varijable$

$W - Šum\ u\ procesu$

Matrica stanja sadrži vrijednosti koji prikazuju parametre objekta koji se prati.
Ako se radi o promatranju dvaju parametara (npr. položaj i brzina) na jednoj osi, matrica stanja ima sljedeći oblik:

$$X = \begin{bmatrix} x \\ \dot{x} \end{bmatrix}$$

Gdje su:

$x - Položaj\ na\ x - osi$

$\dot{x} - Brzina\ na\ x - osi$

U slučaju da pratimo ista dva parametra na dvije osi (x i y), matrica stanja bi imala sljedeći oblik:

$$X = \begin{bmatrix} x \\ \dot{x} \\ y \\ \dot{y} \end{bmatrix}$$

Gdje su:

$x - Položaj\ na\ x - osi$

$\dot{x} - Brzina\ na\ x - osi$

$y - Položaj\ na\ y - osi$

$\dot{y} - Brzina\ na\ y - osi$

Matrica stanja biti će ažurirana na osnovu *prethodnog stanja* (X_{k-1}) i *transformacijske matrice* (*A*) zbrojenih sa umnoškom *matrice kontrolne varijable* (U_k), te njene *transformacijske matrice* (*B*). *Transformacijske matrice A* i *B* pomažu nam da dovedemo krajnji izgled matričnog sustava u odgovarajući format za daljnju obradu bez gubitka relevantnih podataka. *Matrica stanja* analogna je *Newton-ovom zakonu gibanja*:

$$s = v \cdot t + \frac{1}{2} a \cdot t^2 \qquad (4.11.)$$

Razradom *matrice stanja* mogu se opisati njeni pojedini djelovi:

$$X_k = A \cdot X_{k-1} + B \cdot U_k + W_k$$

Prvi dio formule za *matricu stanja* (4.3.) koji glasi ($A \cdot X_{k-1}$) predstavlja promjenu baziranu na poziciji i brzini. Ovaj dio će se ažurirati ovisno o prethodnoj poziciji i brzini kako bi se dobilo trenutno stanje. Analogan je prvom dijelu formule (4.11.) za *Newton-ov zakon gibanja sa ubrzanjem* ($v \cdot t$). Drugio dio formule (4.11.) bavi se ubrzanjem objekta. Analogan je ($\frac{1}{2} x t^2$) *Newton-ovoj formuli za gibanje s ubrzanjem* (4.11.). *Kontrolna varijabla* je ubrzanje ostvareno primjerice utjecajem gravitacije. Doprinos ovog djela formule je vezan uz ubrzanje objekta po nekoj osi. U slučaju *Kalman filtera* vrše se proračuni koristeći malu promjenu u proteklom vremenu označenu sa Δt. Takva promjena u vremenu će predstavljati jedan vremenski ciklus, tj. vrijeme između mjerenja. *Matrica stanja* (4.3.) predstavlja poziciju i brzinu objekta koji se želi pratiti, bilo na jednoj, dvije ili tri osi.

4.4. Jednodimenzijski sustav

Jednostavnim primjerom objekta u slobodnom padu prikazati će se kako se ažuriraju parametri *matrice stanja* (4.3.) u jednodimenzionalnom prostoru. Primjer padajućeg objekta je idealan jer osim pozicije i brzine također sadrži ubrzanje zbog utjecaja gravitacije.

Neka su početne vrijednosti:

$y_{k-1} = 20\ m$

$\Delta T = 1\ s$

$\dot{y}_{k-1} = 0\ m/s$

Gdje su:

$y_{k-1} - Početna\ pozicija\ na\ osi\ y$

$\Delta T - Vremenski\ ciklus$

$\dot{y}_{k-1} - Početna\ brzina\ na\ osi\ y$

$$X_{kp} = A \cdot X_{k-1} + B \cdot U_k + W_k$$

U ovom slučaju koristila se nulta vrijednost (0) za iznos *matrice predviđenog šuma stanja* kako bi primjer bio što jednostavniji.

$$X_{kp} = \begin{bmatrix} 1 & \Delta T \\ 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} y_{k-1} \\ \dot{y}_{k-1} \end{bmatrix} + \begin{bmatrix} \frac{1}{2}\Delta T^2 \\ \Delta T \end{bmatrix} \cdot [g] + 0$$

$$X_{kp} = \begin{bmatrix} y_{k-1} + \Delta T \cdot \dot{y}_{k-1} \\ 0 + \dot{y}_{k-1} \end{bmatrix} + \begin{bmatrix} \frac{1}{2}\Delta T^2 \cdot g \\ \Delta T \cdot g \end{bmatrix}$$

$$X_{kp} = \begin{bmatrix} y_{k-1} + \Delta T \cdot \dot{y}_{k-1} + \frac{1}{2}\Delta T^2 \cdot g \\ 0 + \dot{y}_{k-1} + \Delta T \cdot g \end{bmatrix}$$

$$X_{kp} = \begin{bmatrix} 20 + (1 \cdot 0) + \frac{1}{2} \cdot 1^2 \cdot (-9.8) \\ 0 + 0 + 1 \cdot (-9.8) \end{bmatrix} = \begin{bmatrix} 15.1 \\ -9.8 \end{bmatrix}$$

Rezultat su nove vrijednosti za poziciju i brzinu objekta u matričnom obliku. Prvi red matrice predstavlja poziciju nakon jedne sekunde slobodnog pada, te iznosi 15.1 metar, dok drugi red matrice predstavlja trenutnu brzinu objekta koja sada iznosi 9.8m/s u suprotnom smjeru od promatrane osi, zbog toga je vrijednost iskazana sa predznakom minus (-).

Sada se može ažurirati *matrica mjernog stanja* (4.5.) sa stanjima izmjerenih vrijednosti:

$$Y_k = C \cdot Y_{km} + Z_m$$

Gdje je:

$Y_k - Pozicija\ ili\ izmjerena\ vrijednost\ koja\ predstavlja\ matricu\ stanja$

$C - Transformacijska\ matrica$

$Z_k - Šum\ u\ mjerenju$

Transformacijska matrica (*C*) poprima oblik ovisno o parametrima sustava. Kako se u ovom slučaju promatraju dva različita parametra, pozicija i brzina, transformacijska matrica poprima sljedeći oblik.

$$C = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

Slijedom toga formula za *matricu mjernog stanja* (4.5.) izgleda:

$$Y_k = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ \dot{x} \end{bmatrix} + Z_m$$

Da bi primjer bio prikazan na jednostavniji način zanemariti će se utjecaj *šuma u mjerenju* (Z_m) te ga se može izjednačiti sa nulom (0).

4.5. Dvodimenzijski sustav

Za malo složeniji primjer, poput onih u realnom svijetu, prikazati će se kako izgleda proračun *matrice stanja* (4.3) u dvije dimenzije. Kako se u sustav unosi nova dimenzija, osim *matrice parametara*, *transformacijske matrice* također mijenjaju oblik. Koristi se ista formula iz koje će se dobiti *matrica stanja* (4.3.).

$$X_{kp} = A \cdot X_{k-1} + B \cdot U_k + W_k$$

Kako je uvedena dodatna dimenzija, *matrica vrijednosti* poprima sljedeći oblik:

$$X = \begin{bmatrix} x \\ y \\ \dot{x} \\ \dot{y} \end{bmatrix}$$

Gdje su:

$x, y - Pozicija\ i\ brzina\ na\ x - osi$

$\dot{x}, \dot{y} - Pozicija\ i\ brzina\ na\ y - osi$

Transformacijska matrica (*A*) u dvodimenzionalnom prostoru poprima sljedeći oblik:

$$A = \begin{bmatrix} 1 & 0 & \Delta T & 0 \\ 0 & 1 & 0 & \Delta T \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Iz toga slijedi:

$$AX = \begin{bmatrix} 1 & 0 & \Delta T & 0 \\ 0 & 1 & 0 & \Delta T \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ \dot{x} \\ \dot{y} \end{bmatrix} = \begin{bmatrix} x + \Delta T \cdot \dot{x} \\ y + \Delta T \cdot \dot{y} \\ \dot{x} \\ \dot{y} \end{bmatrix}$$

Iz ovog je vidljivo da će se pozicija ažurirati koristeći prethodnu poziciju (*x*,*y*) dok se brzina ($\dot{x}, \dot{y}$) neće ažurirati jer u ovom dijelu formule nije prisutno ubrzanje.

U sljedećem koraku u sustav se uvode *kontrolne varijable* (a_x, a_y). Zovu se *kontrolne* jer kontroliraju promjenu u položaju i brzini koristeći parametar ubrzanja. Kako bi se rezultantna matrica prilagodila formuli koristi se *transformacijska matrica B* koja je predstavljena na sljedeći način:

$$B = \begin{bmatrix} \frac{1}{2}\Delta T^2 & 0 \\ 0 & \frac{1}{2}\Delta T^2 \\ \Delta T & 0 \\ 0 & \Delta T \end{bmatrix}$$

Sada se može uvesti ubrzanje u sustav:

$$B \cdot U = \begin{bmatrix} \frac{1}{2}\Delta T^2 & 0 \\ 0 & \frac{1}{2}\Delta T^2 \\ \Delta T & 0 \\ 0 & \Delta T \end{bmatrix} \cdot \begin{bmatrix} a_x \\ a_y \end{bmatrix} = \begin{bmatrix} \frac{1}{2}\Delta T^2 \cdot a_x \\ \frac{1}{2}\Delta T^2 \cdot a_y \\ \Delta T \cdot a_x \\ \Delta T \cdot a_y \end{bmatrix}$$

Na kraju, prisutno je sve potrebno kako bi se prikazao krajnji izgled *matrice stanja* za dvije dimenzije.

$$X_k = \begin{bmatrix} x_{k-1} + \Delta T \cdot \dot{x} \\ y_{k-1} + \Delta T \cdot \dot{y} \\ \dot{x}_{k-1} \\ \dot{y}_{k-1} \end{bmatrix} + \begin{bmatrix} \frac{1}{2}\Delta T^2 \cdot a_x \\ \frac{1}{2}\Delta T^2 \cdot a_y \\ \Delta T \cdot a_x \\ \Delta T \cdot a_y \end{bmatrix} = \begin{bmatrix} x_{k-1} + \dot{x}_{k-1}\Delta T + \frac{1}{2}\Delta T^2 \cdot a_x \\ y_{k-1} + \dot{y}_{k-1}\Delta T \cdot + \frac{1}{2}\Delta T^2 \cdot a_y \\ \dot{x}_{k-1} + \Delta T \cdot a_x \\ \dot{y}_{k-1} + \Delta T \cdot a_y \end{bmatrix}$$

Gdje su:

$x_{k-1}, y_{k-1} - Prethodna\ pozicija\ u\ smjeru\ x\ i\ y\ osi$

$\dot{x}_{k-1}\Delta T, \dot{y}_{k-1}\Delta T - Ispravak\ uzrokovan\ prethodnom\ brzinom\ u\ x\ i\ y\ osi$

$\frac{1}{2}\Delta T^2 \cdot a_x, \frac{1}{2}\Delta T^2 \cdot a_y - Doprinos\ ubrzanja\ na\ poziciju\ u\ x\ i\ y\ osi$

$\dot{x}_{k-1}, \dot{y}_{k-1} - Prethodna\ brzina\ u\ smjeru\ x\ i\ y\ osi$

$\Delta T \cdot a_x, \Delta T \cdot a_y - Doprinos\ ubrzanja\ na\ brzinu\ u\ smjeru\ x\ i\ y\ osi$

4.5.1. Matrica procesne kovarijance

Kako bi se bolje razumjela **matrica procesne kovarijance**, treba se prvo definirati kovarijanca u sustavu. Neka je dostupno nekoliko mjernih vrijednosti duljine nekog objekta te su dobiveni rezultati iznosa:

x = 2, 5, 4, 5, 3.

Kao što je vidljivo postoji pet mjernih vrijednosti u isto toliko mjerenja. Svaka od izmjerenih vrijednosti može biti promatrana kao x_i. Dakle, x_i predstavlja vrijednost 2 ili 5 ili 4… **Srednja vrijednost** dobije se na način da se zbroje sve izmjerene vrijednosti te rezultat podjeli sa brojem mjerenja. Tako dobivena srednja vrijednost može se predstaviti sa $\bar{x}$.

$$\bar{x} = \frac{\sum_{i=1}^{N} x_i}{N}$$

Ako se od srednje vrijednosti $\bar{x}$ oduzme svaki od izmjerenih rezultata dobije se razlika koja se naziva **standardna devijacija** od srednje vrijednosti $(\bar{x} - x_i)$

ili

$$\delta x = \sqrt{\frac{\sum_{i=1}^{N} (\bar{x} - x_i)^2}{N}}$$

Ako se ta ista devijacija kvadrira dobije se **kvadrat devijacije**.

$$(\bar{x} - x_i)^2$$

Sada, ako bi se uzeli svi kvadrati devijacija te se zbrojili pa taj rezultat podjelio sa brojem izmjerenih rezultata (*N*), dobiveni broj naziva se **varijanca**.

$$\delta x^2 = \frac{\sum_{i=1}^{N} (\bar{x} - x_i)^2}{N}$$

Kvadriranjem standardne devijacije rezultat je varijanca. Dakle, *varijanca* je veći broj od *standardne devijacije*. Oko dvije trećine svih izmjerenih vrijednosti će upasti u raspon jedne standardne devijacije. Ako bi se uzela *varijanca*, onda se očekuje da će sve izmjerene vrijednosti upasti u taj raspon. *Varijanca* ima puno širi raspon od *standardne devijacije*. Kako se kod *Kalman filtera* koristi skup vrijednosti, poželjna je sigurnost da će sve dobivene vrijednosti koje se očekuju upasti u taj raspon *varijance*. *Varijanca* daje ideju koliko široko su neke vrijednosti rasprostranjene. Kako bi se definirala **kovarijanca**, uz duljinu objekta mjerena je i njegova širina te su dobiveni sljedeći rezultati:

x = 2, 5, 4, 5, 3

y = 4, 3, 2, 4, 5

Ako se usporedi varijacija jedne i druge varijable, rezultat takve usporedbe naziva se *kovarijanca*. Dakle, *standardna devijacija* duljine i *standardna devijacija* širine pomnoži se pa je rezultat tog umnoška iznos *kovarijance*:

$$\delta x \delta y = \frac{\sum_{i=1}^{N} (\bar{x} - x_i)(\bar{y} - y_i)}{N}$$

Matrica procesne kovarijance koristi se nekoliko puta u Kalman filteru.

- U proračunu novog predviđenog stanja (4.4.):

$$P_{kp} = A \cdot P_{k-1} \cdot A^T + Q_k$$

- U proračunu Kalman dobitka (4.6.):

$$K = \frac{P_{kp} \cdot H^T}{H \cdot P_{kp} \cdot H^T + R}$$

- U formuli za pretvorbu trenutnog u prethodno stanje (4.8.):

$$P_k = (I - K \cdot H) \cdot P_{kp}$$

Gdje je:

$P - Matrica\ kovarijance\ stanja$

$Q - Matrica\ kovarijance\ procesnog$ šuma

$R - Matrica\ kovarijance\ osjetničkog\ šuma$

$K - Kalman\ dobitak$

Matrica procesne kovarijance je zapravo matrica *varijance* i *kovarijance*. Međutim, zbog kraće notacije većinom se naziva samo *matrica procesne kovarijance*. Izgled *matrice procesne kovarijance* za dvodimenzijski sustav je sljedeći:

$$P = \begin{bmatrix} \delta x^2 & \delta x \delta y \\ \delta y \delta x & \delta y^2 \end{bmatrix}$$

Matrica procesne kovarijance koristi se kao važan dio za računanje nekoliko stanja u Kalman filteru, među ostalim i samog *Kalman dobitka*.

4.5.2. Matrica kovarijance stanja

Matrica kovarijance stanja u Kalman filteru govori koliki se naglasak treba staviti u *Kalman dobitku* na predviđenu vrijednost. Odnosno, koliki se naglasak treba staviti na vrijednost dobivenu mjerenjem. Također pomaže označiti tip greške kojeg se može očekivati prilikom obrade podataka. Ako je greška u predviđanju za neku varijablu x (npr. pozicija) potpuno neovisna od druge varijable $\dot{x}$ (npr. brzina), kovarijanca elemenata će biti nultog iznosa (0). Nikakva podešavanja neće biti urađena na predviđanjima jedne varijable zbog procesnih grešaka druge varijable. Primjerice, ako matrica procesne kovarijance izgleda na sljedeći način:

$$P = \begin{bmatrix} 0.25 & 0 \\ 0 & 0.04 \end{bmatrix}$$

Neka je iznos od 0.04 varijanca u brzini zbog samog procesa. Iz tog razloga nazivaju se *greške procesa* ili *greške predviđanja*. Greška uzrokovana obradom jedne varijable nema nikakav utjecaj na grešku uzrokovanu obradom druge varijable. Nule (0) u matrici su prikazane jer ne postoji kovarijanca iz razloga što su varijable potpuno neovisne jedna o drugoj.

4.6. Primjer u dvodimenzionalnom prostoru

Nakon što su se definirali svi potrebni parametri može se prikazati praktična implementacija *Kalman filtera* na primjeru praćenja objekta (npr. zrakoplova) u dvodimenzionalnom prostoru.

Sam proces definiran je sljedećim koracima:

1. Definirane vrijednosti za početno/prethodno stanje
2. Novo predviđeno stanje
3. Proračun procesne kovarijance
4. Kalman dobitak
5. Prihvat izmjerenih vrijednosti
6. Ažuriranje novih vrijednosti
7. Pretvorba novih vrijednosti u protekle (za sljedeći ciklus)
8. Ažuriranje vrijednosti za prethodno stanje

4.6.1. Parametri

Početna pozicija i brzina:

$x_0 = 4000\ m$

$y_0 = 3000\ m$

$v_{0x} = 280\ m/s$

$v_{0y} = 120\ m/s$

Izmjerene vrijednosti:

$x_0 = 4000\ m$

$x_1 = 4260\ m$

$x_2 = 4550\ m$

$v_{0x} = 280\ m/s$

$v_{1x} = 282\ m/s$

$v_{2x} = 285\ m/s$

Početne vrijednosti:

$a_x = 2\ m/s^2$

$v_x = 280\ m/s$

$\Delta T = 1\ s$

$\Delta x = 25\ m$

Procesne greške u matrici kovarijance procesa:

$\Delta Px = 20\ m$

$\Delta Pv_x = 5\ m/s$

Greške u promatranju:

$\Delta x = 25\ m$

$\Delta v_x = 6\ m/s$

4.6.2. Prvi ciklus

1.) Prva iteracija kroz *Kalman filtar* započinje prije prihvata izmjerenih vrijednosti te zahtjeva korištenje proizvoljnih početnih vrijednosti (x_0, P_0).

2.) Proračun predviđenog stanja (4.3.)

Nakon što su početne vrijednosti postavljene kreće se u računanje predviđenog stanja. Rezultat jednadžbe predstavlja novo ili trenutno predviđeno stanje.

$$X_{kp} = A \cdot X_{k-1} + B \cdot U_k + W_k = \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x_0 \\ v_{0x} \end{bmatrix} + \begin{bmatrix} \frac{1}{2}\Delta t^2 \\ \Delta t \end{bmatrix} \cdot [a \cdot x_0] + 0$$

$$= \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 4000 \\ 280 \end{bmatrix} + \begin{bmatrix} \frac{1}{2} \\ 1 \end{bmatrix} \cdot [2] = \begin{bmatrix} 4280 \\ 280 \end{bmatrix} + \begin{bmatrix} 1 \\ 2 \end{bmatrix} = \begin{bmatrix} 4281 \\ 282 \end{bmatrix}$$

3.) Proračun procesne kovarijance

Ovaj korak potrebno je napraviti samo jednom. U našem slučaju kako nema povezanosti među greškama u brzini i poziciji, mogu se postaviti dijagonalne vrijednosti na nulu.

$$P_{k-1} = \begin{bmatrix} \Delta x^2 & \Delta x \Delta v \\ \Delta v \Delta x & \Delta v^2 \end{bmatrix} = \begin{bmatrix} 400 & 100 \\ 100 & 25 \end{bmatrix} = \begin{bmatrix} 400 & 0 \\ 0 & 25 \end{bmatrix}$$

$$P_{kp} = A \cdot P_{k-1} \cdot A^T + Q_k = \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 400 & 0 \\ 0 & 25 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 \\ \Delta t & 1 \end{bmatrix} + 0$$

$$= \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 400 & 0 \\ 0 & 25 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix} = \begin{bmatrix} 400 & 25 \\ 0 & 25 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix} = \begin{bmatrix} 425 & 25 \\ 25 & 25 \end{bmatrix}$$

$$= \begin{bmatrix} 400 & 0 \\ 0 & 25 \end{bmatrix}$$

Kako su u zadnjem koraku zanemarene dijagonalne vrijednosti, tj. iznose nula, može se isto napraviti i u ovom slučaju.

4.) Proračun Kalman dobitka (4.6.)

$$K = \frac{P_{kp} \cdot H^T}{H \cdot P_{kp} \cdot H^T + R} = \frac{\begin{bmatrix} 425 & 0 \\ 0 & 25 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}}{\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 425 & 0 \\ 0 & 25 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} + \begin{bmatrix} 625 & 0 \\ 0 & 36 \end{bmatrix}}$$

U ovom koraku može se primjetiti kako posljednja matrica u nazivniku sadrži kvadrate grešaka u promatranju. Vrijednosti grešaka u promatranju potrebno je kvadrirati kako bi bili u istom formatu kao i P_{kp}.

$$K = \frac{\begin{bmatrix} 425 & 0 \\ 0 & 25 \end{bmatrix}}{\begin{bmatrix} 425 & 0 \\ 0 & 25 \end{bmatrix} + \begin{bmatrix} 625 & 0 \\ 0 & 36 \end{bmatrix}} = \frac{\begin{bmatrix} 425 & 0 \\ 0 & 25 \end{bmatrix}}{\begin{bmatrix} 1050 & 0 \\ 0 & 61 \end{bmatrix}} = \begin{bmatrix} 0.405 & 0 \\ 0 & 0.410 \end{bmatrix}$$

Što je *Kalman dobitak* veći to je greška u mjerenju manja, dok u slučaju da je *Kalman dobitak* manji, greška u mjerenju je veća. Dobiveni rezultati govore da će se u ovoj iteraciji staviti veći naglasak na predviđena ažuriranja nego na prihvaćene izmjerene vrijednosti. U nastavku će se koristiti ovako dobivena matrica *Kalman dobitka* kako bi se ažuriralo predviđeno stanje, te kako bi se mogla procjeniti pozicija zrakoplova u sljedećoj iteraciji.

5.) Sada, kada je proračunata matrica *Kalman dobitka* može se uključiti nove rezultate promatranja. Početne vrijednosti u prvom ciklusu su bile ($x_0,\ v_{0x}$). U ovom ciklusu imamo nove izmjerene vrijednosti (x_1, v_{1x}).

$$Y_k = C \cdot Y_{km} + Z_m$$

Gdje je Y_k *matrica izmjerenih vrijednosti* (4.5.) promatranog stanja zrakoplova. Potrebno je da je ova matrica istog tipa kao *matrica stanja X*.

$$Y_k = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 4260 \\ 282 \end{bmatrix} + 0 = \begin{bmatrix} 4260 \\ 282 \end{bmatrix}$$

Nakon što su se dobile vrijednosti Y_k mogu se uključiti u proračun *Kalman dobitka*

(4.6.) kako bi se dobilo novo predviđeno stanje.

6.) Proračun trenutnog stanja (4.7.)

$$X_k = X_{kp} + K\left(Y_k - H \cdot X_{kp}\right)$$

Sve potrebne podatke može se uvesti u sustav te proračunati novo, trenutno stanje. Koristiti će se kombinacija predviđenog stanja, izmjerenih vrijednosti, *matrica kovarijance stanja* i *Kalman dobitka* kako bi se odredilo koliko će se dati na težini izmjerenim vrijednostima ili predviđenim vrijednostima u ovoj iteraciji.

$$X_k = \begin{bmatrix} 4281 \\ 282 \end{bmatrix} + \begin{bmatrix} 0.405 & 0 \\ 0 & 0.410 \end{bmatrix} \cdot \left(\begin{bmatrix} 4260 \\ 282 \end{bmatrix} - \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 4281 \\ 282 \end{bmatrix} \right)$$
$$= \begin{bmatrix} 4281 \\ 282 \end{bmatrix} + \begin{bmatrix} 0.405 & 0 \\ 0 & 0.410 \end{bmatrix} \cdot \begin{bmatrix} -21 \\ 0 \end{bmatrix} = \begin{bmatrix} 4281 \\ 282 \end{bmatrix} + \begin{bmatrix} -8.5 \\ 0 \end{bmatrix} = \begin{bmatrix} 4272.5 \\ 282 \end{bmatrix}$$

7.) Ažuriranje matrice procesne kovarijance (4.8.)

$$P_k = (I - K \cdot H) \cdot P_{kp}$$

Kao što je vidljivo iz jednadžbe, trenutna procesna kovarijanca (P_k) jednaka je umnošku *matrice predviđene procesne varijance* (P_{kp}) i razlici jedinične matrice (I) sa umnoškom *Kalman dobitka* (K) i *transformacijske matrice* (H).

$$P_k = \left\{ \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} - \left(\begin{bmatrix} 0.405 & 0 \\ 0 & 0.410 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \right) \right\} \cdot \begin{bmatrix} 425 & 0 \\ 0 & 25 \end{bmatrix}$$
$$= \left\{ \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} - \begin{bmatrix} 0.405 & 0 \\ 0 & 0.410 \end{bmatrix} \right\} \cdot \begin{bmatrix} 425 & 0 \\ 0 & 25 \end{bmatrix}$$
$$= \begin{bmatrix} 0.595 & 0 \\ 0 & 0.590 \end{bmatrix} \cdot \begin{bmatrix} 425 & 0 \\ 0 & 25 \end{bmatrix} = \begin{bmatrix} 253 & 0 \\ 0 & 14.8 \end{bmatrix}$$

Rješenje ove jednadžbe je nova *matrica procesne kovarijance*. Može se primjetiti kako su iznosi manji od prijašnjih, što znači da se koristeći Kalman filtar približava

predviđenom stanju gdje će se zrakoplov nalaziti.

8.) Nakon što smo dobili sve potrebne podatke, za kraj prvog ciklusa obaviti će se pretvorba stanja vrijednosti. Trenutne vrijednosti koje su uključene u sljedeći ciklus iteracije postaju prethodne vrijednosti za taj ciklus.

$$X_k = \begin{bmatrix} 4272.5 \\ 282 \end{bmatrix} \rightarrow X_{k-1}$$

$$P_k = \begin{bmatrix} 253 & 0 \\ 0 & 14.8 \end{bmatrix} \rightarrow P_{k-1}$$

4.6.3. Drugi ciklus

Kalman filtar je iterativan proces koji osim novih izmjerenih vrijednosti također koristi rezultate prethodnog ciklusa, što se može uočiti odmah na prvom koraku u *matrici prethodnog stanja* (X_{k-1}).

1.) Potrebno je izračunati (4.3.) nove predviđene vrijednosti pozicije i brzine zrakoplova

$$X_{kp} = A \cdot X_{k-1} + B \cdot U_k + W_k = \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 4272.5 \\ 282 \end{bmatrix} + \begin{bmatrix} \frac{1}{2}\Delta t^2 \\ \Delta t \end{bmatrix} \cdot [2] + 0$$

$$= \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 4272.5 \\ 282 \end{bmatrix} + \begin{bmatrix} \frac{1}{2} \\ 1 \end{bmatrix} \cdot [2] + 0 = \begin{bmatrix} 4554.5 \\ 282 \end{bmatrix} + \begin{bmatrix} 1 \\ 2 \end{bmatrix} = \begin{bmatrix} 4555.5 \\ 284 \end{bmatrix}$$

Te novu *matricu procesne kovarijance* (4.4.)

$$P_{kp} = A \cdot P_{k-1} \cdot A^T + Q_k = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 253 & 0 \\ 0 & 14.8 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix} + 0$$

$$= \begin{bmatrix} 253 & 14.8 \\ 0 & 14.8 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix} = \begin{bmatrix} 267.8 & 14.8 \\ 14.8 & 14.8 \end{bmatrix} \cong \begin{bmatrix} 267.8 & 0 \\ 0 & 14.8 \end{bmatrix}$$

2.) Kod proračuna novog *Kalman dobitka* može se primjetiti da i dalje uključuje greške tj. neizvjesnosti u promatranju (*R*). Greške se vremenom mogu smanjiti, pa shodno tome se mogu ažurirati. U tom slučaju potrebno je mnogo više iteracija kroz Kalman filtar kako bi postojala sigurnost da su greške manje od predviđenih. Kako se *matrica procesne kovarijance* smanjuje, tako se smanjuje i utjecaj grešaka na sustav, kao što se može vidjeti iz nazivnika sljedeće jednadžbe (4.6.).

$$K = \frac{P_{kp} \cdot H^T}{H \cdot P_{kp} \cdot H^T + R} = \frac{\begin{bmatrix} 267.8 & 0 \\ 0 & 14.8 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}}{\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 267.8 & 0 \\ 0 & 14.8 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} + \begin{bmatrix} 625 & 0 \\ 0 & 36 \end{bmatrix}}$$

$$= \frac{\begin{bmatrix} 267.8 & 0 \\ 0 & 14.8 \end{bmatrix}}{\begin{bmatrix} 892.8 & 0 \\ 0 & 50.8 \end{bmatrix}} = \begin{bmatrix} 0.300 & 0 \\ 0 & 0.291 \end{bmatrix}$$

Izračunata je vrijednost novog *Kalman dobitka*. Može se primjetiti da se vrijednosti elemenata *Kalman dobitka* smanjuju usporedno s vrijednostima *Kalman dobitka* iz prve iteracije. Što znači da se u ovom ciklusu veći (otprilike za jednu trećinu) naglasak stavlja na predviđene vrijednosti nego na vrijednosti dobivene mjerenjem. Ipak koriste se obe vrijednosti (predviđene i izmjerene) kako bi se približilo točnoj poziciji i brzini zrakoplova koji se prati.

3.) Rezultati promatranja (4.5.)

U ovom ciklusu uvode se novi *rezultati promatranja*.

$$Y_k = C \cdot Y_{km} + Z_m = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 4550 \\ 285 \end{bmatrix} + 0 = \begin{bmatrix} 4550 \\ 285 \end{bmatrix}$$

4.) Proračun trenutnog stanja (4.7.)

$$X_k = X_{kp} + K\left(Y_k - H \cdot X_{kp}\right)$$
$$= \begin{bmatrix}4555.5\\284\end{bmatrix} + \begin{bmatrix}0.300 & 0\\0 & 0.291\end{bmatrix} \cdot \left(\begin{bmatrix}4550\\285\end{bmatrix} - \begin{bmatrix}1 & 0\\0 & 1\end{bmatrix} \cdot \begin{bmatrix}4555.5\\284\end{bmatrix}\right)$$
$$= \begin{bmatrix}4555.5\\284\end{bmatrix} + \begin{bmatrix}0.300 & 0\\0 & 0.291\end{bmatrix} \cdot \begin{bmatrix}-5.5\\1\end{bmatrix} = \begin{bmatrix}4555.5\\284\end{bmatrix} + \begin{bmatrix}-1.7\\0.3\end{bmatrix}$$
$$= \begin{bmatrix}4553.8\\284.3\end{bmatrix}$$

5.) Ažuriranje trenutne *procesne kovarijance* (4.8.)

$$P_k = (I - K \cdot H) \cdot P_{kp} = \left(\begin{bmatrix}1 & 0\\0 & 1\end{bmatrix} - \begin{bmatrix}0.300 & 0\\0 & 0.291\end{bmatrix} \cdot \begin{bmatrix}1 & 0\\0 & 1\end{bmatrix}\right) \cdot \begin{bmatrix}267.8 & 0\\0 & 14.8\end{bmatrix}$$
$$= \begin{bmatrix}0.700 & 0\\0 & 0.709\end{bmatrix} \cdot \begin{bmatrix}267.8 & 0\\0 & 14.8\end{bmatrix} = \begin{bmatrix}187.5 & 0\\0 & 10.5\end{bmatrix}$$

6.) U posljednjem koraku ovog ciklusa, kao što se napravilo na kraju prethodnog ciklusa, opet će se obaviti pretvorba stanja vrijednosti. Trenutne vrijednosti koje se uključuju u sljedeći ciklus iteracije postaju prethodne vrijednosti za taj ciklus.

$$X_k = \begin{bmatrix}4553.8\\284.3\end{bmatrix} \rightarrow X_{k-1}$$

$$P_k = \begin{bmatrix}187.5 & 0\\0 & 10.5\end{bmatrix} \rightarrow P_{k-1}$$

Kroz ovaj kratak primjer može se vidjeti cijeli iterativni proces *Kalman filtera* kroz nekoliko jednostavnih koraka. U slučaju uključivanja više dimenzija ili pridruživanjem novih varijabli, primjerice ubrzanja, iterativni proces se ne mjenja. Promjene se u tom slučaju dešavaju u dimenziji matrica te njihovom sastavu i položaju varijabli unutar matrica.

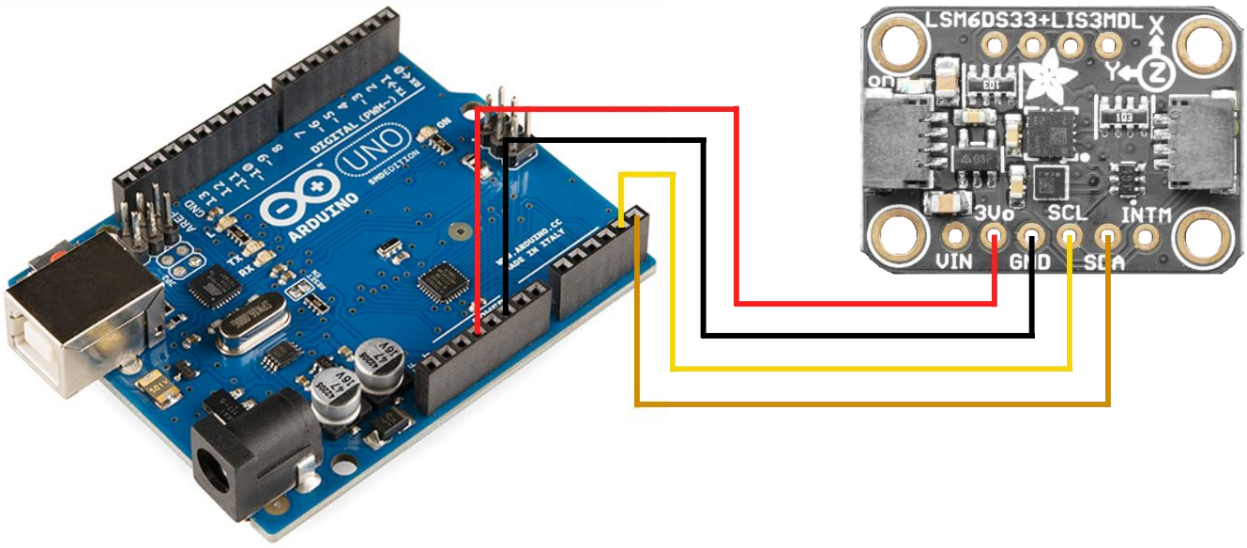
Iz primjera je namjerno izostavljen sav šum u sustavu jer bi uveo dodatnu kompleksnost koja u svrhu demonstracije procesa nije potrebna.

5. PRAKTIČNA IMPLEMENTACIJA KALMAN FILTRA U JEDNOJ DIMENZIJI

5.1. Sklopovlje

Kako bi se na jednostavan način predočio iterativan proces Kalman filtera, korišteni su **Arduino UNO** platforma i **LSM6DS33** senzor za mjerenje akceleracije, žiroskop te magnetometar.

Prikupljeni podaci jedne osi akcelerometra i žiroskopa obraditi će se iterativnim procesom *Kalman filtera* i prikazati kako su vrijednosti koje daje *Kalman filtar* manje kaotične te bliže stvarnim vrijednostima nego očitanja koja se dobivaju direktno sa senzora.



Slika 5.1 Arduino UNO i LSM6DS33 senzor povezani koristeći I^2C protokol

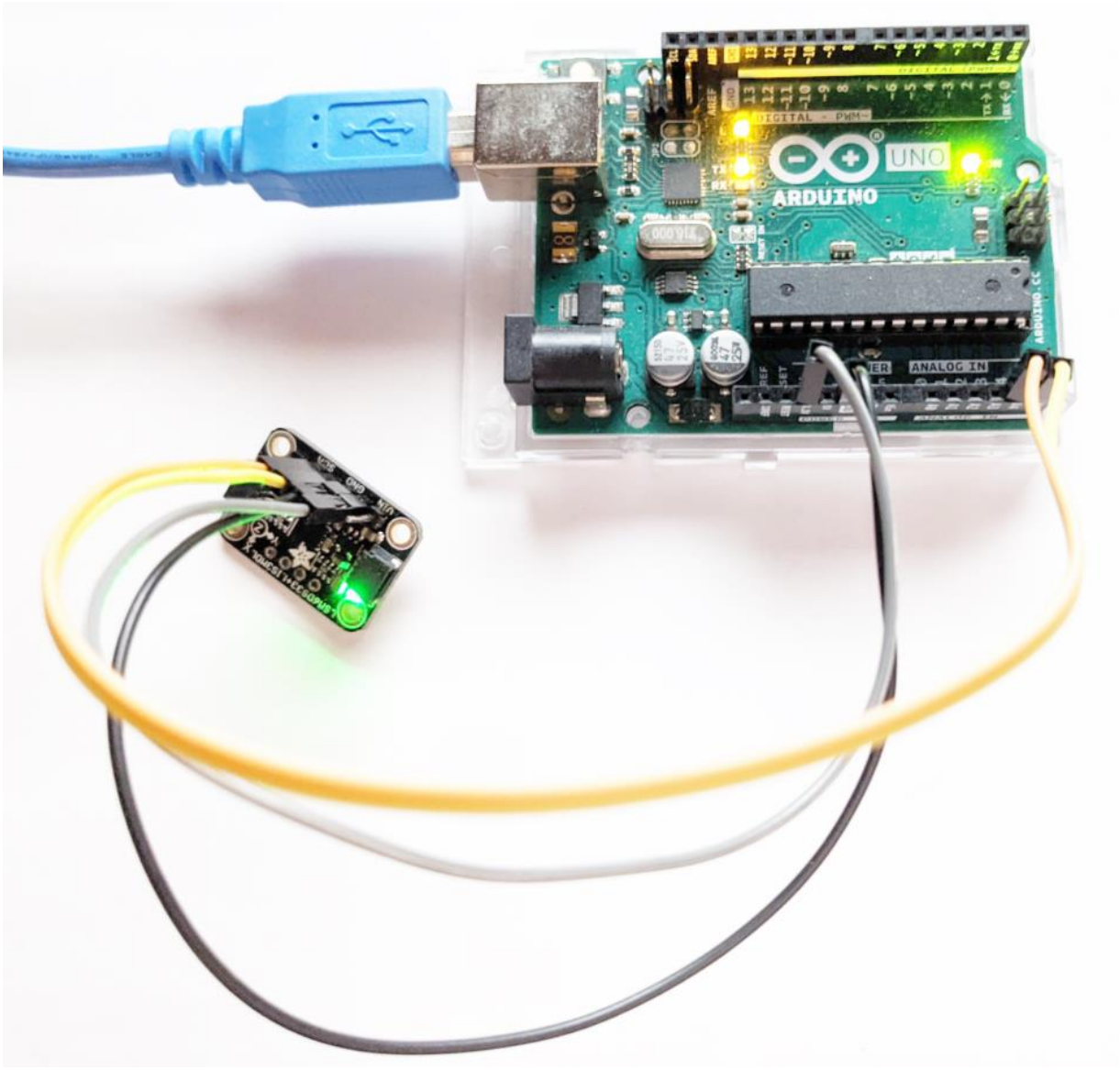
Senzor je napajan naponom od 3.3V direktno sa *Arduina*, a komunikacija je ostvarena preko *I^2C* sučelja - A4 i A5 pinovi na *Arduino UNO* povezani su sa SCL i SDA pinovima na *LSM6DS33* senzoru.

5.2. Sklopovsko i programsko rješenje

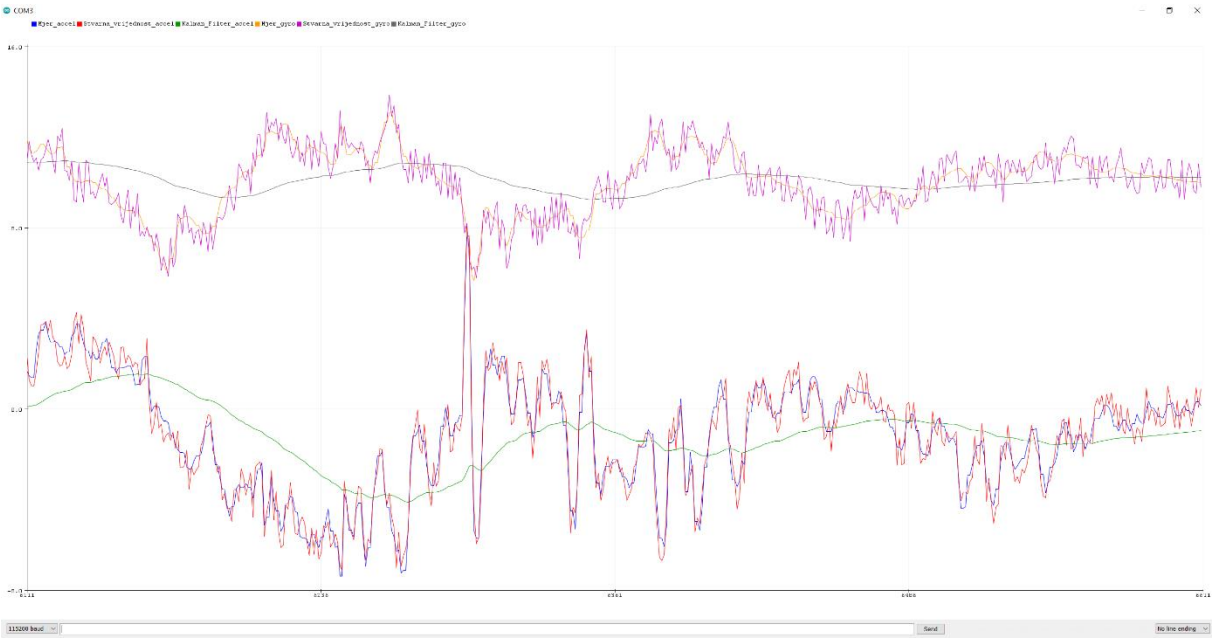
Arduino platforma programirana je koristeći *Arduino IDE*. Kako bi se ostvarila komunikacija sa *LSM6DS33* senzorom korištena je gotova biblioteka iz repozitorija unutar Arduino IDE imena **Adafruit_LSM6DS.h**.
Također je korištena biblioteka **JednodimenzijskiKalmanFilter.h** gdje je smješten dio računalnog koda za računanje parametara *Kalman filtera*. Komunikacija između *Arduino* platforme i *LSM6DS33* senzora ostvarena je koristeći I^2C protokol.

Senzor akceleracije postavljen je na parametar za mjerenje sile u rasponu od +-4G te na frekvenciju prikupljanja podataka od 104Hz, dok je žiroskop također postavljen na frekvenciju od 104Hz. U programu se prikupljaju podaci o akceleraciji i otklonu na X-osi te se kao takvi prikazuju na grafu. Isti podaci se prosljeđuju u jednadžbe *Kalman filtera* kako bi se dobio željeni rezultat te se također iscrtavaju na grafu kako bi prezentirali usporedbu među direktno izmjerenim i obrađenim podacima. Kako bi se ostvario što je moguće točniji prikaz realnih uvjeta, na mjerne vrijednosti sa senzora dodan je simulirani šum mjerenja.

Jednadžbe *Kalman filtera* implementirane su u zasebnu programsku klasu zbog jednostavnosti pregleda te se pozivaju u glavnom programu. Prikupljeni i obrađeni podaci prosljeđuju se na serijski port te se kao takvi mogu vidjeti prikazom *Serial monitor* ili *Serial plotter* funkcije unutar *Arduino IDE-a*.



Slika 5.2 Povezano sklopovlje pod naponom



Slika 5.3 Očitanja akcelerometra i žiroskopa sa rezultatom

Može se primjetiti da su vrijednosti mjerenja senzora poprilično kaotična te kao takva unose nestabilnost u sustav. Vrijednosti dobivene koristeći Kalman filtar su stabilnije i ne odstupaju značajno od izmjerenih vrijednosti.

Iako je frekvencija prikupljanja podataka na senzoru postavljena na 104Hz, sam sustav prihvaća i obrađuje mjerne podatke 10 puta u sekundi (10Hz), što se može vidjeti u programskom kodu.

5.3. Programski kod

5.3.1. Glavni program

```
#include <JednodimenzijskiKalmanFilter.h>
#include <Adafruit_LSM6DS33.h>

/* PROGRAMSKA FUNKCIONALNOST
 *  Prihvat i ispis direktnih podataka sa LSM6DS33 senzora, te izracun Kalman
filtar vrijednosti koristeci klasu
 *  Prilagodjeno za Serial Plotter pregled.
 *
 *  VARIJABLE
 *  real_value_accel: Ocitanje akcelerometra
 *  measured_value_accel: Ocitanje akcelerometra sa simuliranim sumom
 *  real_value_gyro: Ocitanje ziroskopa
 *  measured_value_gyro: Ocitanje ziroskopa sa simuliranim sumom
 *  estimated_value_accel: Obradjeni podaci akcelerometra koristeci Kalman filter
 *  estimated_value_gyro: Obradjeni podaci ziroskopa koristeci Kalman filter
 *  q: Process Noise
 */

Adafruit_LSM6DS33 lsm6ds33;
JednodimenzijskiKalmanFilter jednodimenzijskiKalmanFilterAcc(1, 1, 0.01);
JednodimenzijskiKalmanFilter jednodimenzijskiKalmanFilterGyro(1, 1, 0.01);

// Vrijeme osvjezenja podataka na serijskom portu
const long SERIAL_REFRESH_TIME = 100;
long refresh_time;

void setup() {
  Serial.begin(115200);
  while (!Serial)
    delay(10); // Pauza za Arduino dok se serijsko sucelje ne otvori

  if (!lsm6ds33.begin_I2C()) {
    Serial.println("Pronalazak LSM6DS33 nije uspjesan");
    while (1) {
      delay(10);
    }
  }

  lsm6ds33.configInt1(false, false, true); // akcelerometar DRDY na INT1
  lsm6ds33.configInt2(false, true, false); // ziroskop DRDY na INT2
```

```
// Legenda na Serial Plotter grafu
  Serial.print("Mjer_accel, Stvarna_vrijednost_accel, Kalman_Filter_accel,
Mjer_gyro, Stvarna_vrijednost_gyro, Kalman_Filter_gyro \n");
}

void loop() {
  // Prihvati novi normalizirani sensor event
  sensors_event_t accel;
  sensors_event_t gyro;
  sensors_event_t temp;
  lsm6ds33.getEvent(&accel, &gyro, &temp);

  // Ocitavanje senzorskih vrijednosti akcelerometra i ziroskopa
  float real_value_accel = accel.acceleration.x;
  float real_value_gyro = gyro.gyro.x;

  // Dodan simulirani sum na vrijednosti ocitane sa senzora
  float measured_value_accel = real_value_accel + random(-100,100)/100.0;
  float measured_value_gyro = real_value_gyro + random(-10,10)/10.0;

  // Izracun predvidjene vrijednosti Kalman filtera
  float estimated_value_accel =
jednodimenzijskiKalmanFilterAcc.azurirajProcjenu(measured_value_accel);
  float estimated_value_gyro =
jednodimenzijskiKalmanFilterGyro.azurirajProcjenu(measured_value_gyro);

  // Ispis na serijsko sucelje u formatu pogodnom za iscrtavanje grafa
  if (millis() > refresh_time) {
    Serial.print(real_value_accel,4);
    Serial.print(",");
    Serial.print(measured_value_accel,4);
    Serial.print(",");
    Serial.print(estimated_value_accel,4);
    Serial.print(",");
    Serial.print(real_value_gyro,4);
    Serial.print(",");
    Serial.print(measured_value_gyro,4);
    Serial.print(",");
    Serial.print(estimated_value_gyro,4);
    Serial.println();

    refresh_time = (millis()/10) + SERIAL_REFRESH_TIME;
  }

}
```

5.3.2. Biblioteka (JednodimenzijskiKalmanFilter.cpp)

```
/*
 * JednodimenzijskiKalmanFilter - Implementacija Kalman filtera s jednom
varijablom.
 * Petar Celio Cega, 27.9.2021.
 */
#include "Arduino.h"
#include <math.h>
#include "JednodimenzijskiKalmanFilter.h"

JednodimenzijskiKalmanFilter::JednodimenzijskiKalmanFilter(float mjerenje_err,
float procjena_err, float q)
{
  _greska_mjerenja=mjerenje_err;
  _greska_procjene=procjena_err;
  _q = q;
}

float JednodimenzijskiKalmanFilter::azurirajProcjenu(float mjerenje)
{
  _kalman_dobitak = _greska_procjene/(_greska_procjene + _greska_mjerenja);
  _trenutna_procjena = _protekla_procjena + _kalman_dobitak * (mjerenje -
_protekla_procjena);
  _greska_procjene =  (1.0 - _kalman_dobitak)*_greska_procjene +
fabs(_protekla_procjena-_trenutna_procjena)*_q;
  _protekla_procjena=_trenutna_procjena;

  return _trenutna_procjena;
}
void JednodimenzijskiKalmanFilter::postaviGreskuMjerenja(float mjerenje_err)
{
  _greska_mjerenja=mjerenje_err;
}

void JednodimenzijskiKalmanFilter::postaviGreskuProcjene(float procjena_err)
{
  _greska_procjene=procjena_err;
}
void JednodimenzijskiKalmanFilter::postaviSumProcesa(float q)
{
  _q=q;
}
float JednodimenzijskiKalmanFilter::prihvatiKalmanDobitak() {
  return _kalman_dobitak;
}
```

5.3.3. Header dokument za biblioteku (JednodimenzijskiKalmanFilter.h)

```
/*
 * JednodimenzijskiKalmanFilter - Implementacija Kalman filtera za modele sa
jednom varijablom.
 * Petar Celio Cega, 27.9.2021.
 */

#ifndef JednodimenzijskiKalmanFilter_h
#define JednodimenzijskiKalmanFilter_h

class JednodimenzijskiKalmanFilter
{

public:
  JednodimenzijskiKalmanFilter(float mjerenje_err, float procjena_err, float q);
  float azurirajProcjenu(float mjerenje);
  void postaviGreskuMjerenja(float mjerenje_err);
  void postaviGreskuProcjene(float procjena_err);
  void postaviSumProcesa(float q);
  float prihvatiKalmanDobitak();

private:
  float _greska_mjerenja;
  float _greska_procjene;
  float _q;
  float _trenutna_procjena;
  float _protekla_procjena;
  float _kalman_dobitak;

};

#endif
```

6. ZAKLJUČAK

Kroz ovaj rad pošnjene su osnove fuzije senzora i najčešće korišten proces koji se veže uz *low-level* fuziju senzora, a to je *Kalman filter*. Osim *Fourier-ove transformacije, Kalman filtar* je trenutno najčešće korišten algoritam za obradu signala.

Uspješno je demonstrirana primjena digitalne obrade signala sa senzora za akceleraciju i žiroskopa koristeći jednodimenzionalan *Kalman filter*. Kao što se može vidjeti iz rezultata, mjerenja koja su obrađena koristeći *Kalman filtar* su mnogo manje kaotična, s manjim utjecajem šuma, te bliže stvarnim vrijednostima od mjerenja dobivenih direktno sa senzora za akceleraciju.

Isti proces može se koristiti za bilo kakav drugi tip senzora, primjerice senzora za temperaturu, gdje će se iteriranjem po jednadžbama *Kalman filtera* već nakon nekoliko mjerenja jako brzo približiti stvarnoj vrijednosti.

Demonstracija u ovom radu koristi dvije vrijednosti, akceleraciju i otklon na osi x te je samo uvod u najčešće korištenu svrhu *Kalman filtera*, a to je *fuzija senzora*. U *fuziji senzora* se kroz proces implementacije višedimenzionalnog *Kalman filtera* dolazi do preciznog mjerenja koristeći više različitih homogenih ili heterogenih senzora. Što je u kompleksnim elektroničkim sustavima s velikim brojem senzora vrlo česta pojava. Danas gotovo neizbježna.

7. POPIS SLIKA I TABLICA

Slika 3.1 Iterativan proces jednostavnog Kalman filtra..... 10
Slika 3.2 Iznos Kalman dobitka i njegov utjecaj na procjenu stanja 11
Slika 4.1 Dijagram toka procesa višedimenzijskog modela Kalman filtra 14
Slika 5.1 Arduino UNO i LSM6DS33 senzor povezani koristeći I^2C protokol 34
Slika 5.2 Povezano sklopovlje pod naponom 36
Slika 5.3 Očitanja akcelerometra i žiroskopa sa rezultatom 37

8. LITERATURA

[1] H. B. Mitchell, HBM. *Multi-Sensor Data Fusion, First edition.* Berlin, Heidelberg, New York: Springer, 2007.

[2] Lawrence A. Klein, LAK. *Sensor and Data Fusion.* Bellingham, Washington: Spie press, 2010.

[3] Geir Evensen, GE. *Data Assimilation, Second edition*. Springer, 2009.

[4] Michel van Biezen, MvB. *Special Topics: Kalman Filter,* YouTube, 2015.

[5] Arduino - https://www.arduino.cc/

[6] Adafruit LSM6DS33 akcelerometar- https://www.adafruit.com/

[7] I^2C protokol - https://www.circuitbasics.com