# Konstrukcija ultrazvučnog anemometra baziranog na Arduinu

**Orešković, Augustin**

**Master's thesis / Diplomski rad**

**2020**

*Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj:* **University of Zagreb, Faculty of Science / Sveučilište u Zagrebu, Prirodoslovno-matematički fakultet**

*Permanent link / Trajna poveznica:* https://urn.nsk.hr/urn:nbn:hr:217:894223

*Rights / Prava:* In copyright / Zaštićeno autorskim pravom.

*Download date / Datum preuzimanja:* **2024-04-25**

*Repository / Repozitorij:*

Repository of the Faculty of Science - University of Zagreb

SVEUČILIŠTE U ZAGREBU
PRIRODOSLOVNO-MATEMATIČKI FAKULTET
FIZIČKI ODSJEK

Augustin Orešković

# KONSTRUKCIJA ULTRAZVUČNOG ANEMOMETRA BAZIRANOG NA ARDUINU

Diplomski rad

Zagreb, 2020.

SVEUČILIŠTE U ZAGREBU
PRIRODOSLOVNO-MATEMATIČKI FAKULTET
FIZIČKI ODSJEK

INTEGRIRANI PREDDIPLOMSKI I DIPLOMSKI SVEUČILIŠNI STUDIJ
FIZIKA; SMJER ISTRAŽIVAČKI

**Augustin Orešković**

Diplomski rad

# Konstrukcija ultrazvučnog anemometra baziranog na Arduinu

Voditelj diplomskog rada: doc. dr. sc. Dalibor Paar

Ocjena diplomskog rada: _____

Povjerenstvo: 1. _____

2. _____

3. _____

Datum polaganja: _____

Zagreb, 2020.

*Prijateljima i poznanicima, precima i potomcima.*

# Sažetak

Istraživanja u kontekstu praćenja klimatskih promjena, procesa u okolišu i druge primjene pored ostalog imaju potrebu sustavnih mjerenja brzine i smjera strujanja zraka. U ovom radu dan je prikaz postupka konstrukcije ultrazvučnog anemometra namijenjenog takvoj primjeni; za izradu anemometra odabrana je Arduino platforma. Za kalibraciju anemometra korišten je jednostavan zračni tunel koji je omogućavao brzine do $2.7\,\mathrm{m/s}$. Izrađeni uređaj pokazao se funkcionalan, premda preciznost mjerenja nije adekvatna za praktičnu primjenu. Objašnjena su poboljšanja u metodi detekcije, kao i u kalibraciji, kojima bi se moglo poboljšati preciznost uređaja.

Ključne riječi: mjerni instrument, ultrazvučni anemometar

# Ultrasonic Anemometer Construction Based on Arduino


## Abstract

Research in the context of monitoring climate change, environmental processes and other applications, among other things, needs systematic measurements of air velocity and direction. In this paper, the construction procedure of an ultrasonic anemometer intended for such use is presented; the Arduino platform has been chosen for the construction purposes. A simple wind tunnel providing wind speeds up to $2.7\,\mathrm{m/s}$ has been used to calibrate the anemometer. The designed device proved to be functional, although its measurement precision is inadequate for practical use. Improvements in detection method, as well as those in calibration, that would improve the device precision are explained.



Keywords: measuring instrument, ultrasonic anemometer

# Sadržaj

**1 Uvod** **1**

1.1 Teorijska podloga . . . . . 1

1.2 Arduino . . . . . 2

1.3 Senzori . . . . . 3

**2 Dizajn** **5**

2.1 Programski kod . . . . . 5

2.2 Odabir senzora . . . . . 8

**3 Kalibracija** **11**

3.1 Prvi zračni tunel . . . . . 11

3.2 Drugi zračni tunel . . . . . 14

**4 Zaključak** **18**

**Dodaci** **19**

**A Ispis programskog koda** **19**

A.1 Globalne varijable . . . . . 19

A.2 Početne postavke . . . . . 20

A.3 Glavna petlja . . . . . 22

A.4 Mjerenje vremena putovanja signala . . . . . 23

A.5 Transformacija koordinata . . . . . 24

A.6 Ispis na računalo . . . . . 25

A.7 Ispis na LCD zaslon . . . . . 26

# 1 Uvod

Mjerni instrument ključan je alat u znanstvenoj metodi. Kvaliteta i opseg znanstvenog istraživanja često će ovisiti o dostupnosti i mogućnostima mjernog instrumenta. U 21. stoljeću cijene senzora značajno padaju te su sve dostupnije različite izvedbe mikroračunala. To otvara mogućnost razvoja mjernih instrumenata za potrebe istraživanja, posebno ako se radi o terenskim istraživanjima gdje je potreban veći broj instrumenata.

Razna klimatološka i meteorološka istraživanja zahtijevaju precizno mjerenje brzine i smjera strujanja zraka; uz primjenu u znanosti, anemometri se koriste u aeronautici [1], hobističkom letenju bespilotnim letjelicama [2], moreplovstvu [3], jedrenju na dasci [4], streljaštvu [5], rukovanju ventilacijskim uređajima [6] i šire. Radi ovako široke primjene očita je potreba za jeftinim i lako dostupnim anemometrima. Primarna je motivacija ovog rada upravo dizajn jeftinog općenamjenskog anemometra. Za tu je potrebu odabrana ultrazvučna tehnika radi elegancije, prenosivosti i male osjetljivosti na meteorološke uvjete. Ultrazvučna tehnika omogućava i trodimenzionalnu detekciju brzine i smjera vjetra kao i detekcije turbulencija.

Za izvedbu odabrana je Arduino platforma radi dostupnosti, jednostavnosti i male cijene.

## *1.1 Teorijska podloga*

Određivanje brzine vjetra ultrazvukom temelji se na mjerenju razlike u vremenu puta ultrazvučnog impulsa između dva senzora. Nazovimo senzore *A* i *B* te neka postoji komponenta brzine vjetra $v$ usmjerena od senzora *A* prema senzoru *B*. Brzina zvuka može ovisiti o faktorima poput temperature, tlaka i vlažnosti zraka, međutim pokazat će se da možemo izmjeriti brzinu vjetra bez određivanja tih veličina, uz uvjet da se ne mijenjaju značajno tijekom jednog mjerenja; s obzirom na to da će jedno mjerenje trajati znatno kraće od jedne sekunde ova je pretpostavka opravdana. Označimo brzinu zvuka s $c$ i ravnolinijsku udaljenost među senzorima s $d$; neka je $t_{AB}$ vrijeme puta ultrazvučnog impulsa od senzora $A$ do senzora $B$. Sada će za vremena puta vrijediti jednadžba 1.1.

$$t_{AB} = \frac{d}{c+v}, \qquad t_{BA} = \frac{d}{c-v} \tag{1.1}$$

Eliminacijom brzine zvuka iz sustava preostaje izraz za brzinu vjetra dan jednadžbom 1.2. Ovo potvrđuje raniju tvrdnju da ne moramo znati vrijednosti parametara o kojima ovisi brzina zvuka (pa tako ni brzinu zvuka uopće) dok god se oni ne mijenjaju u tijeku jednog mjerenja.

$$v = \frac{d}{2}\left(\frac{1}{t_{AB}} - \frac{1}{t_{BA}}\right) \tag{1.2}$$

U slučaju da je potrebno mjeriti brzinu i smjer vjetra u više dimenzija moguće je koristiti više međusobno ortogonalnih postava tako da svaki mjeri jednu komponentu ukupne brzine. U ovom radu korištena su četiri senzora za dobivanje dvije komponente vjetra, što je dovoljno za većinu primjena jer nas najčešće zanima vjetar paralelan površini Zemlje.

Vrijeme puta moguće je mjeriti izravno, što je metoda korištena u ovom radu zbog svojstava korištenih senzora. Ovakva mjerenja vrše se emisijom kratkog ultrazvučnog pulsa na jednom senzoru te mjerenjem vremena do njegove detekcije na drugom.

Puno preciznije mjerenje vremena moglo bi se postići metodom faznog pomaka. Ako bismo emitirali kontinuiran ton te usporedili funkciju tog tona s funkcijom tona kojeg primamo na drugom senzoru, u idealnim uvjetima vidjeli bismo da su jednake do na fazni pomak proporcionalan vremenu puta. Mjerenjem vremena između prolaska kroz nulu ovih signala izravno bismo mjerili vrijeme puta do na cijeli broj perioda signala.

Uprosječivanjem niza mjerenja bilo bi moguće izravnim mjerenjem ostvariti preciznost unutar jednog perioda signala, a metodom faznog pomaka mjeriti vrijeme još preciznije. Ovakav postupak omogućio bi detekciju izuzetno malih strujanja zraka, no nažalost, zbog tehničkih ograničenja senzora korištenih u ovom radu korištenje metode faznog pomaka nije bilo moguće.

Bitno je napomenuti kako se smjer vjetra uobičajeno izražava kao smjer iz kojega vjetar puše, a ta je konvencija korištena i ovdje.

## *1.2 Arduino*

Arduino [7] je tvrtka koja se bavi dizajnom i distribucijom sklopovlja otvorenog dizajna. Otvorenost dizajna, dostupnost razvojnih alata i vrlo kvalitetna dokumentacija privlače brojne korisnike, a veličina zajednice korisnika i već navedeno čine Ardu-

ino najboljom platformom za prototipizaciju danas. Kao takav predstavlja idealnu platformu za razvoj obrazovnih programa te predstavlja izvrstan alat za problemski orijentiranu nastavu u STEM područjima.

U ovom projektu korišten je mikrokontroler Arduino Uno [8], ultrazvučni senzori HY-SRF05 [9] i VMA306 [10] (koji su sami jednaki senzorima HY-SRF05) te LCD1602 zaslon i ostali osnovni elementi sadržani u *Arduino Starter Kit-u* [11].

Premda bi za izradu anemometra bili prikladniji općenamjenski ultrazvučni transduktori, zbog aktualnih okolnosti[1] korišteni su lakše dostupni senzori namijenjeni detekciji prepreka u navigaciji. Posljedica ovoga je nemogućnost korištenja metode faznog pomaka u određivanju vremena putovanja signala zbog ograničenja u dizajnu senzora.

Za sve programiranje mikrokontrolera korišten je programski paket *Arduino-CLI 0.10.0* [12]. Za prijem podataka od strane računala tijekom mjerenja izvršenih u sklopu ovog rada, te za obradu i prikaz podataka korišten je programski jezik *Python 3.8.2*.

U svrhu kalibracije, kao referentni anemometar korišten je tvornički kalibriran uređaj Testo 410i. Ovaj mehanički anemometar omogućava mjerenje brzine vjetra u rasponu od $0.4\,\mathrm{m/s}$ do $30\,\mathrm{m/s}$, uz rezoluciju od $0.1\,\mathrm{m/s}$. Preciznost mu je $\pm(0.2\,\mathrm{m/s} + 2\,\%)$ [13].

## *1.3 Senzori*

HY-SRF05 i VMA306 senzori imaju pet izvoda: `VCC`, `Trig`, `Echo`, `OUT` i `GND` [9] [10]. Za rad senzora, `VCC` se spaja na konstantan izvor napona $5\,\mathrm{V}$, a `GND` na uzemljenje. Senzor očekuje puls trajanja $10\,\mu\mathrm{s}$ na izvod `Trig`, te po primanju istog emitira ultrazvučni puls na jednom od transduktora i odašilje na izvodu `Echo` puls do zaprimanja povratnog ultrazvučnog pulsa. Mjerenjem trajanja emitiranog pulsa na izvodu `Echo` neposredno je moguće mjeriti vrijeme putovanja ultrazvučnog pulsa do mjesta refleksije i nazad.

Za potrebe mjerenja vremena putovanja ultrazvučnog pulsa između dva senzora u samo jednom smjeru, puls se emitira s oba senzora istovremeno, a rezultantni signal na izvodu `Echo` mjeri se samo na jednom od senzora. Kako senzor ne zna razlikovati

[1] Za čitatelje u dalekoj budućnosti: ovaj je diplomski rad napisan tijekom pandemije bolesti COVID-19.

vlastiti puls od pulsa drugog senzora, detektira upadni puls drugog senzora i daje vrijeme između emisije svog pulsa i primljenog pulsa drugog senzora. Kako su oba pulsa emitirana istovremeno, ovo vrijeme odgovara vremenu putovanja pulsa drugog senzora u jednom smjeru. Za dimenzije korištenog postava ova se metoda pokazala dobrom.

Izvod `OUT` služi promjeni moda funkcioniranja senzora te korištenje jednog izvoda za funkcionalnost i `Trig` i `Echo` izvoda, no u ovom slučaju nije korišten.

# 2 Dizajn

Uređaj je dizajniran s ciljem mogućnosti zamjene senzora kako bi se omogućilo testiranje pojedinih među dostupnim senzorima (bilo je dostupno ukupno sedam senzora). Gotov uređaj može se vidjeti na slici 2.1. Postav je izrađen od drva, a napravljen je prema shemi vidljivoj na slici 2.2. Korišteni su čavli čiji promjer odgovara promjeru postojećih rupa na senzorima te su postavljeni tako da se senzori mogu mijenjati bez potrebe da se rastavlja cijeli uređaj. Primjer senzora nataknutog na spomenute čavle može se vidjeti na slici 2.1d. Ovakav je dizajn znatno olakšao odabir senzora prikazan u potpoglavlju 2.2. Dimenzije uređaja birane su tako da budu dovoljne za funkcionalnost anemometra bez da budu suviše velike za praktičnu primjenu uređaja.

Elektronička shema vidljiva je na slici 2.3. `Trig` izvodi na parovima senzora spojeni su s istim izvodom mikrokontrolera kako bi se omogućilo istovremeno odašiljanje signala na pojedinom paru; `Echo` izvodi spojeni su na različite izvode mikrokontrolera kako bi detekcija na pojedinim senzorima bila moguća. Dostupno je dovoljno izvoda na mikrokontroleru da se uključe još dva senzora za treću os ako za to ima potrebe, što iz praktičnih razloga ovdje nije učinjeno. LCD zaslon spojen je po uzoru na *Arduino Project Book* [14].

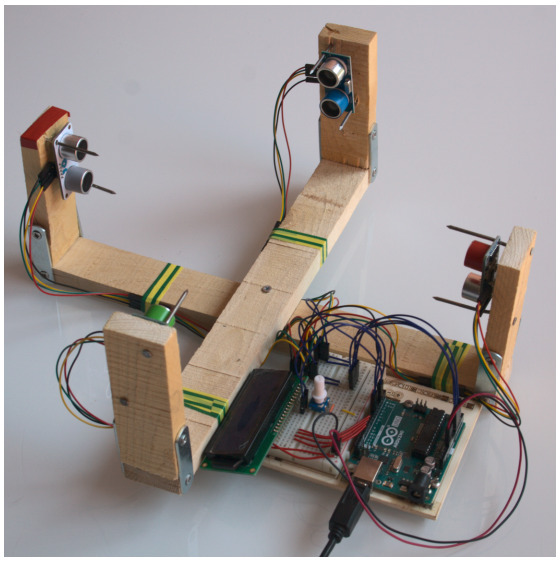
U konačnom postavu svakom je senzoru dodijeljena strana svijeta, tako da pri ispravnoj orijentaciji kut $\phi$ odgovara azimutu[2].

## *2.1 Programski kod*

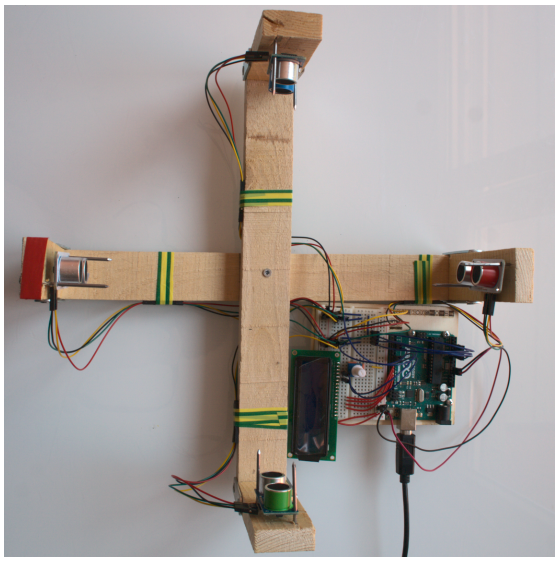
Potpuni programski kod može se vidjeti u dodatku A.

Arduino Uno mikrokontroler zahtijeva prisutnost dvije glavne funkcije u kodu: `setup` i `loop`. Prilikom spajanja na napajanje (ili pritiska tipke `RESET`) jednom se izvrši funkcija `setup`, koja se ovdje koristi za inicijalizaciju komunikacije s računalom i za inicijalizaciju korištenih izvoda na mikrokontroleru, kako za mjerenja tako i za ispis rezultata na LCD zaslon. Nakon inicijalizacije, mikrokontroler će izvršavati funkciju `loop` poput beskonačne petlje sve do isključivanja napajanja (ili pritiska tipke `RESET`). Unutar ove funkcije odvijaju se sva mjerenja i izračuni.

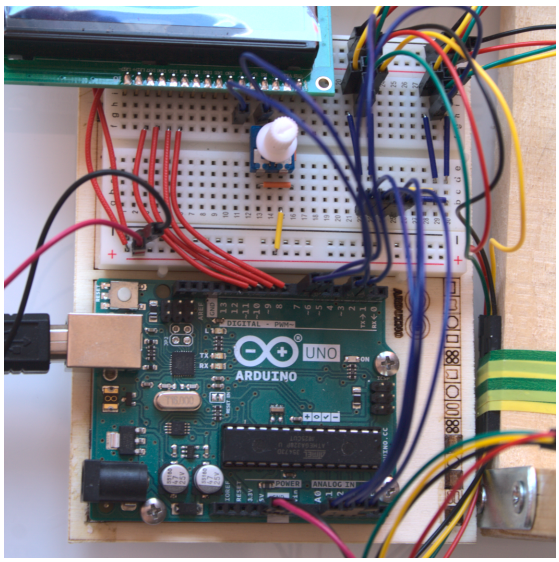
[2] Azimutu smjera iz kojeg vjetar puše, što je suprotno smjeru vektora brzine, kako je objašnjeno u poglavlju 1.1.



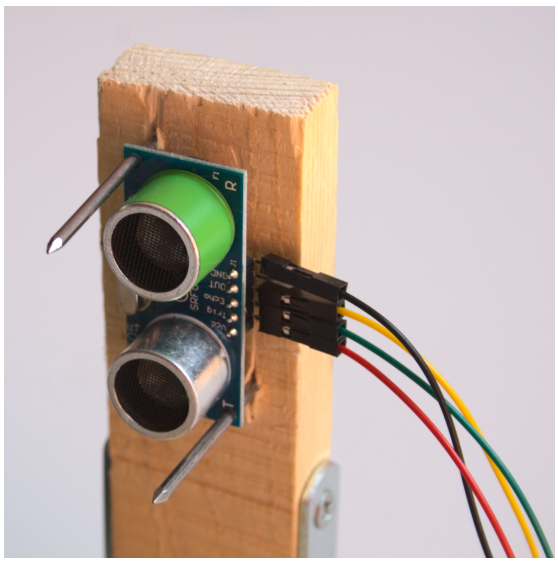
(a)



(b)

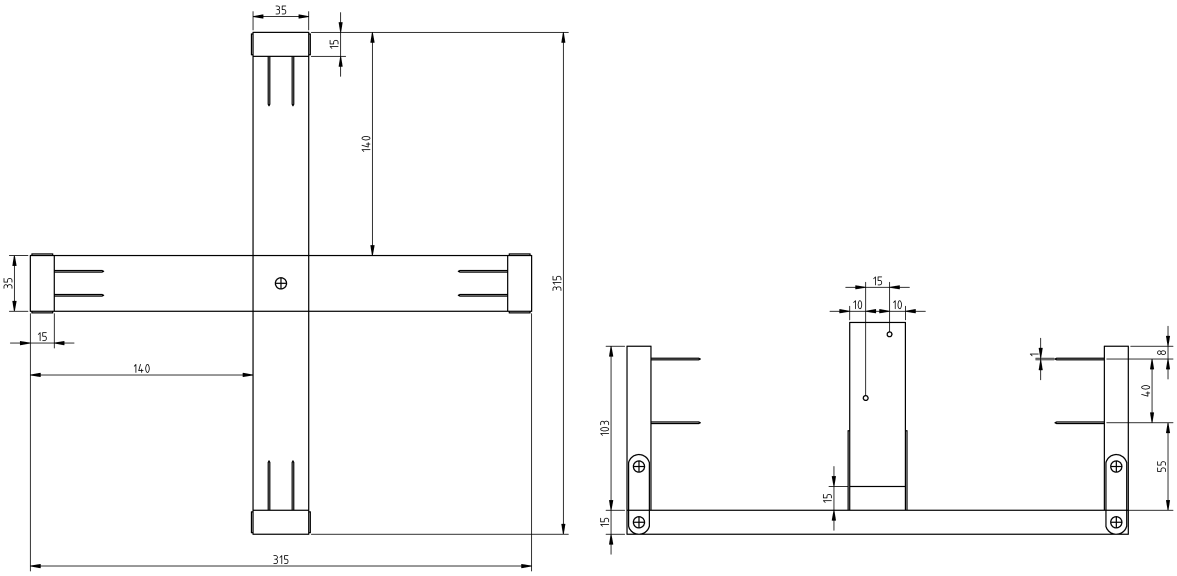



(c)

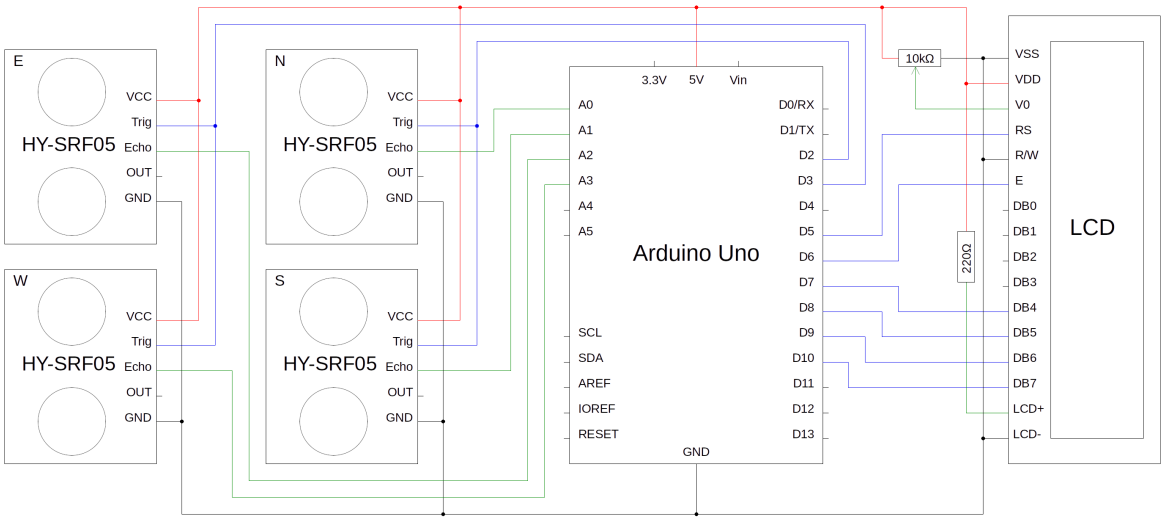


(d)

Slika 2.1: Izrađeni anemometar, te neki njegovi detalji.




Slika 2.2: Shema prema kojoj je izrađen postav. Postav je izrađen od drvenih letvica kroz koje su probijeni čavli na koje se senzori mogu nataknuti. Senzori nisu prikazani na shemi. Sve prikazane mjere dane su u milimetrima.




Slika 2.3: Skica električnih spojeva u konačnom uređaju. Žice su označene različitim bojama radi preglednosti. Senzori su označeni engleskim početnim slovima strana svijeta kojima pripadaju u konačnom postavu.

Glavni dio programa počinje pozivom funkcije `timeMeasure` koja vraća vrijeme putovanja signala između dva senzora u sekundama, a to vrijeme dobiva komunikacijom sa senzorima skladno njihovoj funkcionalnosti kako je objašnjeno u potpoglavlju 1.3. Navedena funkcija kao argument prima redni broj osi te redni broj senzora na osi, pri čemu redni broj osi određuje koja će se dva senzora koristiti u mjerenju (svakoj osi pripadaju točno dva senzora), a redni broj senzora koji će se od senzora koristiti kao detektor u mjerenju (odnosno, u kojem smjeru duž osi će se mjeriti vrijeme putovanja).

Iz vremena puta duž iste osi u dva smjera, program računa komponentu brzine vjetra paralelnu toj osi koristeći formulu 1.2 uz zamjenu udaljenosti i bezdimenzionalnih faktora u formuli multiplikativnim kalibracijskim faktorom te dodavanjem aditivnog kalibracijskog faktora. Time dobivamo formulu 2.1, gdje je $m$ multiplikativni, a $a$ aditivni kalibracijski faktor.

$$v = m\left(\frac{1}{t_{AB}} - \frac{1}{t_{BA}}\right) + a \tag{2.1}$$

Prvi je faktor dimenzije udaljenosti, a drugi dimenzije brzine. Ovi su faktori dobiveni kalibracijom uređaja kako je prikazano u poglavlju 3. Ovakva mjerenja i računi brzine ponavljaju se određeni broj puta, zadan varijablom `cycleLength`, a rezultati se zbrajaju te se naposljetku računa prosjek. Ovakav postupak potreban je radi povećanja preciznosti.

Dobivena prosječna brzina transformira se u cilindrični i sferni sustav u skladu s formulama poznatim iz matematike [15], a potom se ispisuju u željenom koordinatnom sustavu na LCD zaslonu i na računalu. U slučaju da se koriste samo dvije osi, kao što je u ovdje izloženoj aparaturi, cilindrični i sferni sustav svode se na polarni, a u slučaju da se koristi samo jedna os svi su sustavi ekvivalentni.

## *2.2 Odabir senzora*

Za izradu konačnog uređaja bilo je dostupno pet senzora modela HY-SRF05 te dva modela VMA306, no bila su potrebna svega četiri senzora (dva para po dva) za konačan uređaj. Kako bi se među dostupnim senzorima odabrali najpogodniji, testiran je svaki par senzora.

Cijeli mjerni uređaj stavljen je u zatvorenu kutiju, a razmak među pojedinim mje-

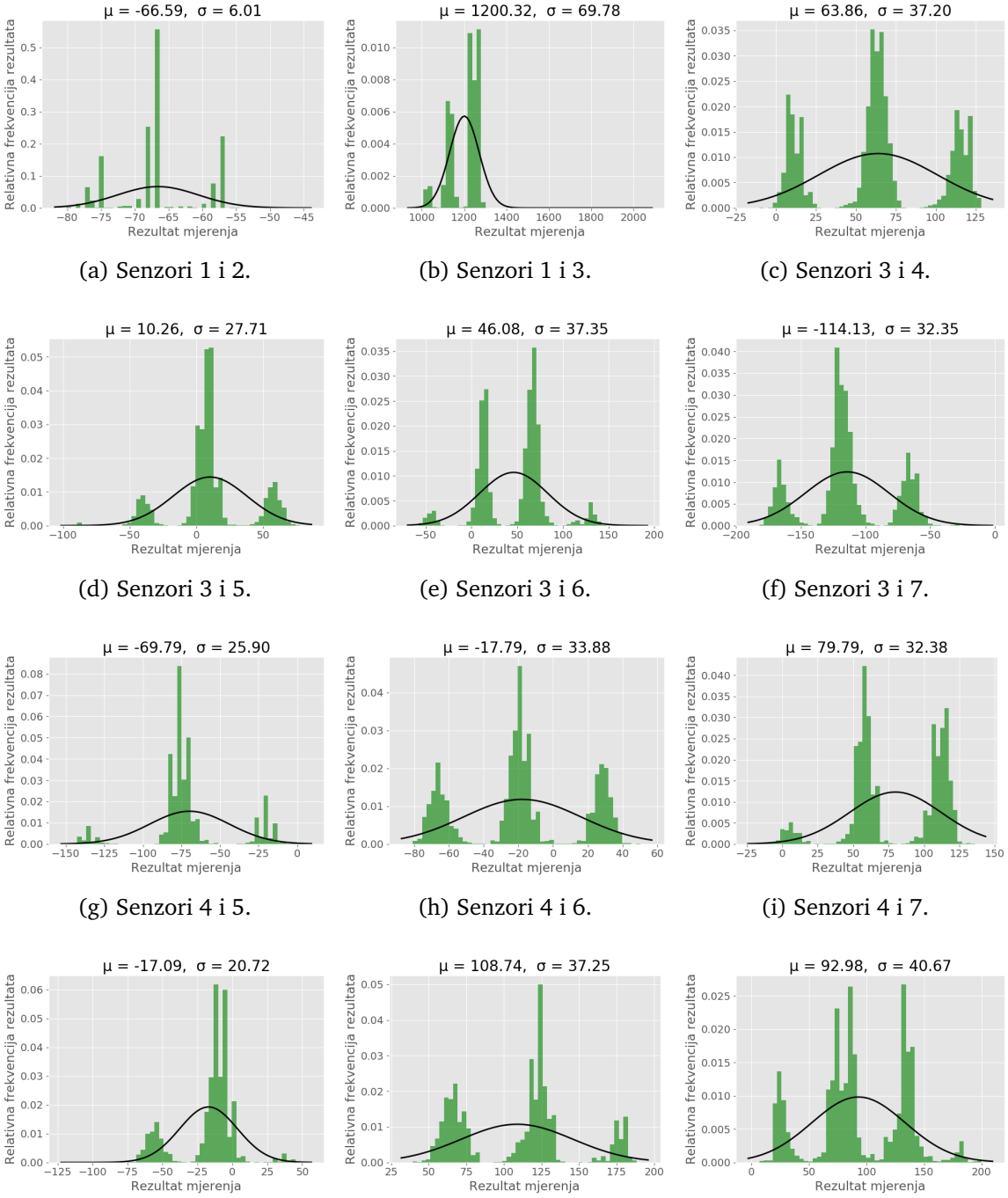
renjima povećan je na pedeset milisekundi, kako bi se osiguralo dovoljno dugo vrijeme da intenzitet pulsa padne ispod osjetljivosti senzora. Napravljeno je i nekoliko kontrolnih mjerenja s mnogo većim vremenom među mjerenjima i bez kutije, ali u zatvorenoj prostoriji, međutim oboje se pokazalo nepraktičnim za ostvarivanje željenog broja mjerenja. Pokazalo se, doduše, da je korištena metoda zadovoljavajuća jer su kontrolni rezultati odgovarali konačnim rezultatima.

Kako je ovo rađeno prije kalibracije uređaja, bilo je nemoguće rezultate mjerenja prikazati kao brzine. Ipak, moglo se očekivati da će rezultati biti približno nula s obzirom na to da su mjerenja rađena u odsustvu vjetra, što znači da bi vrijeme putovanja signala trebalo biti jednako u oba smjera te bi prema formuli 2.1 rezultat trebao biti nula.

Za svaki par senzora napravljeno je deset tisuća mjerenja te je pronađena normalna distribucija koja ih opisuje, a rezultati su vidljivi na slici 2.4. Na apscisi se nalaze rezultati mjerenja, koji su u ovom stadiju bili samo brojevi, a na ordinati relativna frekvencija pojedine skupine rezultata. Pokazalo se da, iako nominalno jednaki, senzori modela VMA306 (senzori 1 i 2) i HY-SRF05 (senzori 3, 4, 5, 6 i 7) ne rade dobro u tandemu; na slici je prikazan samo jedan takav primjer (1 i 3), no svi su imali slične srednje vrijednosti i standardne devijacije. VMA306 senzori pokazali su se precizniji od senzora HY-SRF05.

Naposljetku su odabrani parovi 1 i 2 (VMA306) te 5 i 6 (HY-SRF05), na osnovu najmanjih standardnih devijacija. Pritom su senzori 1 i 2 korišteni za os koja u daljnjem tekstu nosi oznaku `NS`, a 5 i 6 za os koja u daljnjem tekstu nosi oznaku `EW`.[3]

[3]Oznake potječu od engleskih naziva za strane svijeta.




(a) Senzori 1 i 2. (b) Senzori 1 i 3. (c) Senzori 3 i 4.

(d) Senzori 3 i 5. (e) Senzori 3 i 6. (f) Senzori 3 i 7.

(g) Senzori 4 i 5. (h) Senzori 4 i 6. (i) Senzori 4 i 7.

(j) Senzori 5 i 6. (k) Senzori 5 i 7. (l) Senzori 6 i 7.

Slika 2.4: Rezultati testiranja u parovima. Za svaki skup podataka prikazana je normalna distribucija koja ga najbolje opisuje, a mjereni podatci prikazani su u obliku histograma s pedeset stupaca. Srednja vrijednost i standardna devijacija prikazane normalne distribucije vidljive su iznad grafa, a označene su grčkim slovima $\mu$ i $\sigma$. S obzirom da su ova mjerenja rađena prije kalibracije uređaja, rezultate je nemoguće poistovjetiti s fizikalnom veličinom brzine strujanja zraka te su dani kao bezdimenzionalni brojevi.

# 3 Kalibracija

Kako bi se anemometar mogao koristiti, bilo je potrebno pronaći kalibracijske faktore $m$ i $a$ iz jednadžbe 2.1. U tu svrhu izrađena su dva zračna tunela, pri čemu se prvi pokazao neuspješan, a drugi je izrađen kako bi riješio uočene probleme prvoga. Uspješno su određeni kalibracijski koeficijenti, no pokazalo se kako je preciznost uređaja mala.

## *3.1 Prvi zračni tunel*

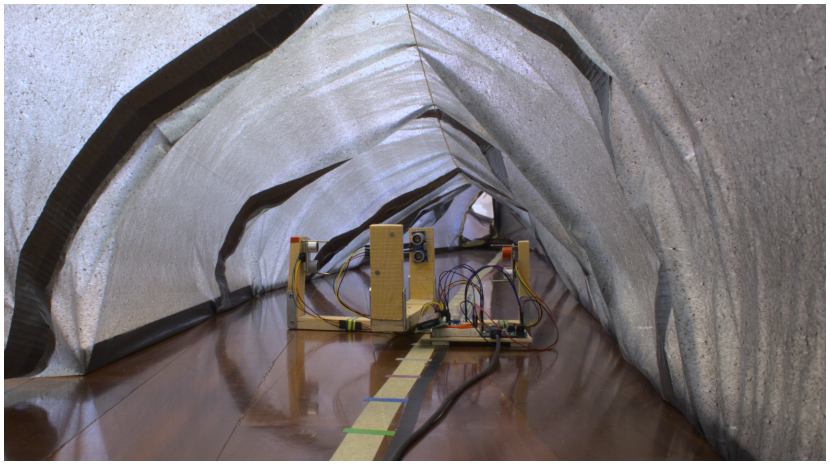
Prvi zračni tunel izrađen za pokušaj kalibracije može se vidjeti na slici 3.1. Tunel je izrađen od plastičnih vreća obješenih o špagu te učvršćenih uz pod ljepljivom trakom i drvenim daskama. Za tjeranje zraka kroz tunel korišten je kućni ventilator. Tunel je bio dug približno četiri metra, a cilj je bio varijabilnim presjekom tunela dobiti velik spektar brzina bez obzira na svega tri različite jačine koje je dostupni ventilator dopuštao.

Referentnim anemometrom načinjena su mjerenja svakih dvadeset centimetara duž cijelog tunela. Rezultati ovih mjerenja prikazani su na slici 3.2. Apscisa prikazuje udaljenost od kraja tunela; na samom kraju brzina je vrlo velika jer je izlazni otvor uži od ostatka tunela, a na početku brzina išćezava jer zrak koji ventilator puše u tunel prolazi iznad anemometra. Uz protok zraka duž tunela na svakom od tih mjesta mjerio se i protok zraka okomito na smjer tunela. Premda bi svako ovakvo mjerenje načelno trebalo kao prosječnu brzinu dati nula, s obzirom na to da korišteni referentni anemometar ne razlikuje orijentacije vjetra, zapravo su mjerene apsolutne vrijednosti trenutnog okomitog toka. Dobiveni prosjeci apsolutnih vrijednosti okomitog toka korištena su kao izravan indikator turbulentnosti toka na pojedinom mjestu. Naime, u slučaju da je na određenoj poziciji struja zraka više turbulentna, okomita komponenta će više oscilirati oko nule, te će se to u konačnom rezultatu vidjeti kao izražena okomita komponenta. Na mjestima gdje postoje velike deformacije oblika tunela moguća je i prosječna struja u smjeru koji nije potpuno duž tunela.

Sa slike 3.2 je vidljivo kako je vjetar duž većine tunela turbulentan. Za potrebe kalibracije korištena su ona mjesta za koja vrijedi da je i na njima, i na udaljenosti od dvadeset centimetara u oba smjera, mjereni okomiti tok jednak nula. Za ta su mjesta mjerene brzine u smjeru duž tunela prikazane na slici.

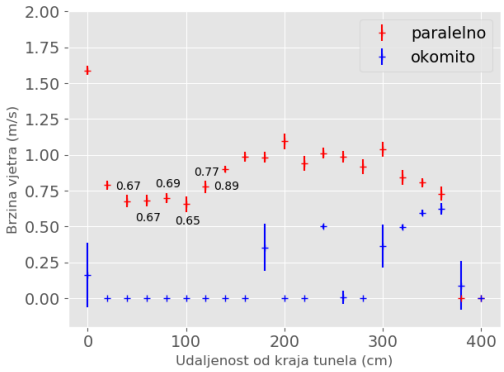
(a)

(b)

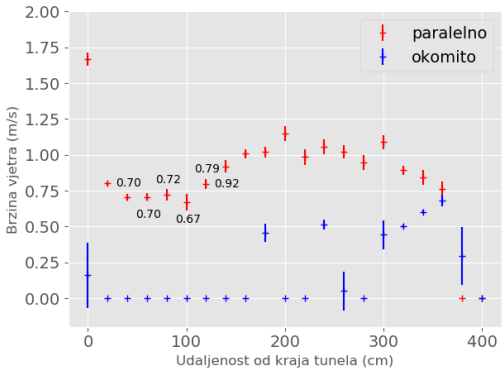


(c)

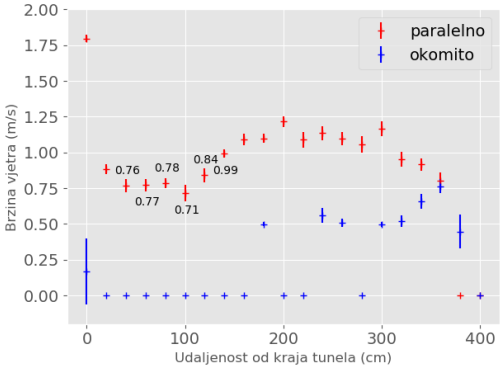
Slika 3.1: Pogled na prvi zračni tunel iz dva kuta te na anemometar unutar tunela za vrijeme mjerenja.




(a) Snaga ventilatora: 1

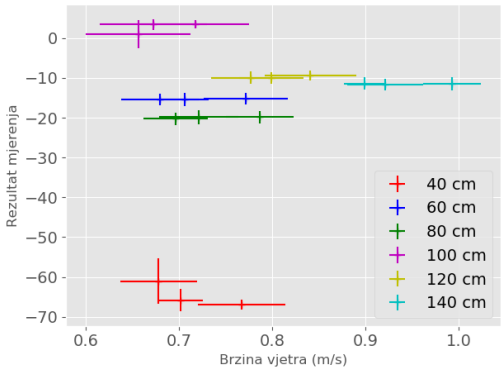



(b) Snaga ventilatora: 2

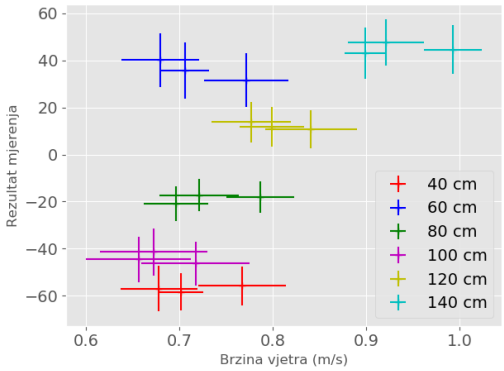



(c) Snaga ventilatora: 3

Slika 3.2: Mjerenja brzine unutar prvog tunela načinjena referentnim anemometrom. Prikazana su mjerenja načinjena paralelno smjeru protoka zraka kroz tunel (tj. duž tunela) i okomito na taj smjer. Mjerenja načinjena okomito smjeru strujanja zraka ne odgovaraju stvarnom stanju kako je objašnjeno u tekstu. Dane su numeričke vrijednosti brzine na mjestima koja su korištena za kalibracijska mjerenja.




(a) Smjer: NS.




(b) Smjer: EW.

Slika 3.3: Kalibracijska mjerenja odrađena u prvom tunelu. Brzine vjetra mjerene referentnim anemometrom nalaze se na apscisi, a kalibracijska mjerenja načinjena izrađenim anemometrom na ordinati (dana bezdimenzionalno). Boje označavaju mjerenja napravljena na različitim udaljenostima od kraja tunela.

Mjerenja brzine vjetra na istim mjestima načinjene za potrebe kalibracije, napravljena izrađenim anemometrom moguće je vidjeti na slici 3.3. Nažalost, pokazalo se da rezultati mjerenja više ovise o mjestu mjerenja nego o nominalnoj brzini. Ovo je vjerojatno posljedica činjenice da je tunel sagrađen od savitljivog materijala te upućuje na promjene u brzini toka koje ovise o drugim faktorima, a ne samo o jačini ventilatora. Najvjerojatnije je da su sitne deformacije u obliku vreća, nastale tijekom pomicanja mjerne aparature, uzrokovale značajne promjene u lokalnoj brzini vjetra.

S obzirom na to da se iz dobivenih rezultata nije moglo ništa zaključiti o preciznosti izrađenog anemometra, izrađen je drugi zračni tunel. U njegovoj izradi korišteno je iskustvo dobiveno u izradi prvog tunela, te se pokazao uspješnijim.

## *3.2 Drugi zračni tunel*

Drugi tunel izrađen je konstantnog presjeka, kako bi se oba anemometra mogla koristiti istovremeno. Glavna prednost ovakve metode jest da se nije potrebno oslanjati na konstantnu lokalnu brzinu vjetra u tunelu (što se pokazalo problematičnim kod prvog tunela), već se trenutna brzina mjeri i s referentnim i s izrađenim anemometrom. Pritom konstantan presjek tunela osigurava da su ove brzine jednake.

Tunel je izrađen većinski od stiropora, a može se vidjeti na slici 3.4. Uz promjenjivu jačinu ventilatora, korištena je pomična stiroporna ploča za mijenjanje oblika cijevi koja je spajala ventilator i ulaz u tunel. Korištenjem ove ploče moglo se osigurati da će ulazni dio tunela pružati različiti otpor ventilatoru ovisno o poziciji ploče, tako postižući veći raspon brzina vjetra u tunelu. Ovaj je mehanizam potvrdio i ranije sumnje o nedostatku funkcionalnosti prvog tunela, odnosno to da brzina vjetra značajno ovisi o malim promjenama u obliku cijevi. Ipak, drugi tunel nije patio od ovog problema zahvaljujući činjenici da je mjerni dio nepromjenjivog oblika, te da oba anemometra rade istovremeno. Tunel je također moguće otvoriti, kako je vidljivo na slici 3.4e, kako bi se anemometri mogli pomicati bez straha da će se pritom oštetiti tunel.

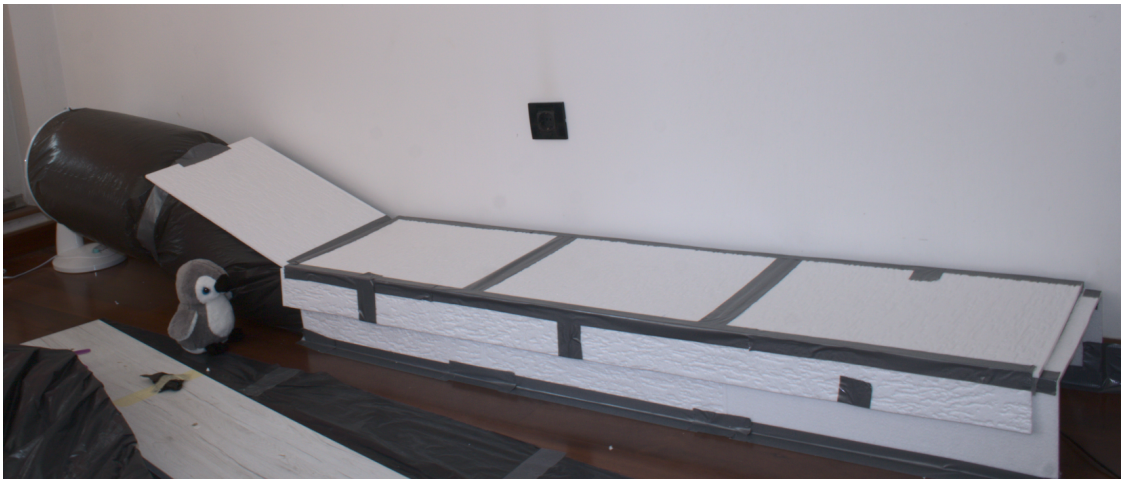
Za dodatno povećanje dostupnog raspona brzina, korištena je ljepljiva traka na poleđini ventilatora kojom se ograničavala količina zraka koju ventilator može povući iz okoline. Ovo je na prvom tunelu bilo nemoguće jer je radi veće površine presjeka i uz vrlo malo ljepljive trake ventilator postajao suviše slab da bi postigao brzinu veću

od $0.4\,\mathrm{m/s}$, što je minimalna brzina koju referentni anemometar može izmjeriti.

Podatci dobiveni mjerenjima u drugom tunelu vidljivi su na slici 3.5. Uz podatke, prikazani su i pravci dobiveni linearnom regresijom kojom su pronađeni kalibracijski faktori $m$ i $a$.[4] Važno je naglasiti kako se na svakom od grafova nalazi gotovo deset tisuća točaka; pritom se mnoge međusobno prekrivaju. Zbog ovoga, može se činiti kako linije dobivene linearnom regresijom ne odgovaraju podatcima. Koeficijenti dobiveni ovom kalibracijom korišteni su u konačnom kodu mikrokontrolera, a iznose $m_{\texttt{NS}} = -0.045\,928\,\mathrm{m}$, $a_{\texttt{NS}} = 0.701\,016\,\mathrm{m/s}$, $m_{\texttt{EW}} = 0.030\,570\,\mathrm{m}$ i $a_{\texttt{EW}} = -0.067\,870\,\mathrm{m/s}$.

Vidljivo je kako je za svaku danu brzinu vjetra raspodjela rezultata koje daje izrađeni anemometar vrlo velika, što potvrđuje sumnju da korišteni senzori nisu dovoljno precizni za željene rezultate. Ovo je vidljivo i iz dobivenih koeficijenata determinacije (vidljivih na slici 3.5). Ipak, ovisnost mjerenja o stvarnosti jest vidljiva, što potvrđuje koncept, ujedno i cilj ovog rada.

[4] U stvari su kalibracijski faktori očitani iz inverzne linearne regresije zbog oblika ovisnosti.



(a)

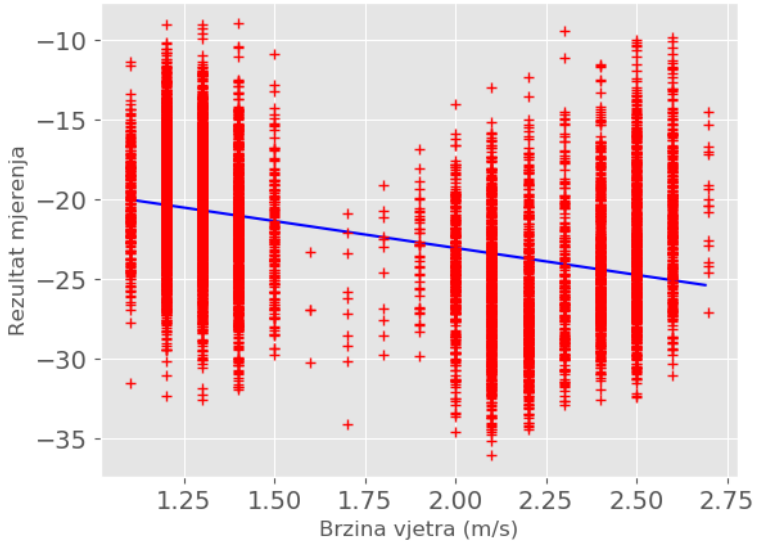
(b)

(c)

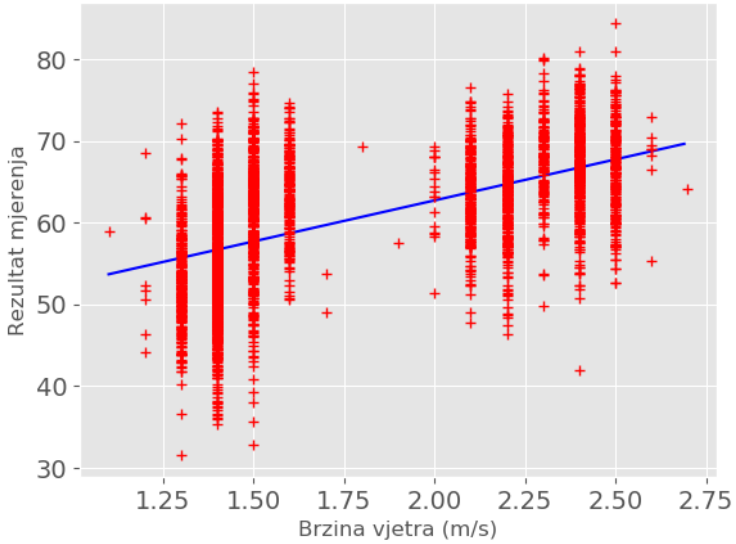
(d)

(e)

Slika 3.4: Drugi zračni tunel. Na detaljima vidljiva je ploča korištena za postizanje većeg raspona brzina vjetra u različitim položajima. Vidljiv je i otvoren tunel, te oba anemometra u njemu.




(a) Smjer: NS. $R^2 = 0.154988$




(b) Smjer: EW. $R^2 = 0.306750$

Slika 3.5: Mjerenja provedena u drugom tunelu. Prikazani su rezultati svih mjerenja, kao i pravci dobiveni linearnom regresijom. Brzine vjetra mjerene referentnim anemometrom nalaze se na apscisi, a kalibracijska mjerenja načinjena izrađenim anemometrom na ordinati (dana bezdimenzionalno). Ispod grafova prikazan je koeficijent determinacije $R^2$.

# 4 Zaključak

Uspješno je izrađen ultrazvučni anemometar, premda male preciznosti. Ovime je potvrđen koncept, no potrebna su poboljšanja kako bi uređaj bio koristan za stvarne potrebe.

Prvenstveno poboljšanje bila bi zamjena senzora za transduktore kojima bi bilo moguće koristiti metodu faznog pomaka, kako je opisano u poglavlju 1.1. Također, veliko poboljšanje bila bi kalibracija korištenjem zračnog tunela koji može postići brzine vjetra od nekoliko desetaka metara u sekundi. Takav raspon brzina omogućio bi značajno bolju kalibraciju već izrađenog anemometra, bez obzira na izostanak metode faznog pomaka.

Arduino platforma pokazala se zadovoljavajućim izborom.

# Dodaci

## Dodatak A Ispis programskog koda

### *A.1 Globalne varijable*

Datoteka koja sadrži konstante i varijable koje se koriste kroz program, a nisu specifične za pojedinu instancu glavne petlje.

```
#ifndef GLOBAL_VARIABLES
  #define GLOBAL_VARIABLES
```

Broj osi: može biti 1, 2 ili 3.

```
  const int numberOfAxis = 2;
```

Broj senzora po osi: uvijek jednak 2.

```
  const int sensorsPerAxis = 2;
  const int numberOfSensors = numberOfAxis * sensorsPerAxis;
```

Broj mjerenja koji se uprosječuje i ispisuje.

```
  const int cycleLength = 10;
```

Kalibracijski faktori dimenzije duljine.

```
  const double mulFactor[3] = {-45928.0, 30570.0, 1000000.0};
  const double addFactor[3] = {0.701016, -0.067870, 0.0};
```

Parametri koji oblikuju puls prva dva su u mikrosekundama zadnji je u milisekundama.

```
  const int prePulseDelay = 245;
  const int pulseDuration = 10;
  const int postPulseDelay = 10;
```

Izlazi odgovaraju pojedinim osima.

```
  int triggerPin[numberOfAxis];
```

Ulazi odgovaraju pojedinim senzorima na pojedinim osima.

```
int echoPin[numberOfAxis][sensorsPerAxis];
```

Izlaz i ulaz kojima kreće inicijalizacija.

```
const int initTriggerPin = 2;
const int initEchoPin = A0;
```

Uprosječena brzina u prethodnom ciklusu; prvi indeks određuje koordinatni sustav, a drugi os.

```
double averageVelocity[3][numberOfAxis] = {0};
```

Oznake komponenti koordinatnih sustava.

```
String coordinateSystemMarking[3][3] ={
  {"x", "y", "z"},
  {"rho", "phi", "z"},
  {"r","phi","theta"}
};
```

Brzina komunikacije s računalom.

```
const int baudRate = 9600;
```

Izlazi koji se koriste za komunikaciju s LCD zaslonom.

```
const int lcdRs = 6;
const int lcdE = 7;
const int lcdD4 = 8;
const int lcdD5 = 9;
const int lcdD6 = 10;
const int lcdD7 = 11;
```

```
#endif
```

## *A.2 Početne postavke*

Funkcija koja se poziva jednom pri pokretanju mikrokontrolera. Koristi se za inicija-

lizaciju komunikacije s računalom i za inicijalizaciju izlaza i ulaza.
Prima: ništa
Vraća: ništa

```
#include "globalVariables.h"
#include <LiquidCrystal.h>
#include <math.h>

LiquidCrystal lcd(lcdRs, lcdE, lcdD4, lcdD5, lcdD6, lcdD7);

void setup() {
```

Inicijalizacija komunikacije s računalom.

```
  Serial.begin(baudRate);
```

Inicijalizacija LCD zaslona.

```
  lcd.begin(16,2);
```

Izlaz za prvu os nalazi se na initTriggerPin, a za svaku sljedeću jedan dalje. Prvi ulaz prve osi nalazi se na initEchoPin, drugi jedan dalje, a za svaku iduću os dodaju se još dva ulaza.

```
  for (int i = 0; i < numberOfAxis; i ++) {

    triggerPin[i] = initTriggerPin + i;

    pinMode(triggerPin[i],OUTPUT);

    for (int j = 0; j < sensorsPerAxis; j ++) {

      echoPin[i][j] = initEchoPin + (2 * i) + j;

      pinMode(echoPin[i][j],INPUT);
```

```
    }
  }
}
```

## A.3 Glavna petlja

Funkcija koja se poziva u krug pri radu mikrokontrolera. Poziva funkciju timeMeasure za svaki senzor te računa brzine iz vremena putovanja i zbraja rezultate. Nakon broja mjerenja određenog varijablom cyclelength računa prosječnu brzinu te poziva generateCoordinateTransformations kako bi dobila prosječnu brzinu iskazanu u cilindričnom i sfernom sustavu. Naposljetku, poziva funkcije za ispis rezultata na LCD zaslon i računalo.
Prima: ništa
Vraća: ništa

```
void loop() {

  double time[2];
  double velocity;
  double sumOfVelocities[numberOfAxis] = {0};

  for (int n = 0; n < cycleLength; n++) {

    for (int i = 0; i < numberOfAxis; i++) {

      for (int j = 0; j < sensorsPerAxis; j++) {

        time[j] = timeMeasure(i, j);
      }
```

Formula za brzinu poznata iz fizike.

```
      timeFactor = (1.00 / time[1]) - (1.00 / time[0])
      velocity = mulFactor[i] * timeFactor + addFactor[i];
      sumOfVelocities[i] += velocity;
```

```
    }
  }

  for (int i = 0; i < numberOfAxis; i++) {
```

Osi na kojima su senzori već opisuju kartezijev sustav.

```
    averageVelocity[0][i] = sumOfVelocities[i] / cycleLength;
    sumOfVelocities[i] = 0;
  }

  generateCoordinateTransformations();
```

0 označava ispis u kartezijevom sustavu.

```
  serialPrintVelocity(0);
```

1 označava ispis u cilindričnom sustavu.

```
  LCDPrintVelocity(1);
}
```

## *A.4 Mjerenje vremena putovanja signala*

Funkcija odašilje impuls na svim senzorima na danoj osi, a impuls prima samo danim senzorom.
Prima: redni broj osi i senzora
Vraća: vrijeme puta impulsa u mikrosekundama

```
double timeMeasure(int axis, int pin) {

  double duration;
```

Kratki niski puls osigurava precizan početak signala.

```
  digitalWrite(triggerPin[axis], LOW);
  delayMicroseconds(prePulseDelay);
```

```
  digitalWrite(triggerPin[axis], HIGH);
  delayMicroseconds(pulseDuration);
  digitalWrite(triggerPin[axis], LOW);

  duration = pulseIn(echoPin[axis][pin], HIGH);
```

Kratka stanka osigurava da mjerenja ne interferiraju.

```
  delay(postPulseDelay);

  return duration;
}
```

## *A.5 Transformacija koordinata*

Funkcija koja iz prosječne brzine u Kartezijevom sustavu računa njenu reprezentaciju u cilindričnom i sfernom sustavu.
Prima: ništa
Vraća: ništa

```
void generateCoordinateTransformations () {

  double cartesian[3] = {0};
  double cylindrical[3] = {0};
  double spherical[3] = {0};

  double rhoSquared;

  for (int i = 0; i < numberOfAxis; i++) {

    cartesian[i] = averageVelocity[0][i];
  }

  rhoSquared = square(cartesian[0]) + square(cartesian[1]);
```

```
  cylindrical[0] = sqrt(rhoSquared);
  cylindrical[1] = atan2(cartesian[1], cartesian[0]);
  cylindrical[2] = cartesian[2];

  spherical[0] = sqrt(rhoSquared + square(cartesian[2]));
  spherical[1] = cylindrical[1];
  spherical[2] = atan2(sqrt(rhoSquared), cartesian[2]);

  for (int i = 0; i < numberOfAxis; i++) {

    averageVelocity[1][i] = cylindrical[i];
    averageVelocity[2][i] = spherical[i];
  }
}
```

## *A.6 Ispis na računalo*

Funkcija ispisuje brzinu na računalo. Pritom se osi odvajaju razmakom, a mjerenja novim redom.
Prima: indeks željenog koordinatnog sustava
Vraća: ništa

```
void serialPrintVelocity(int coordinateSystem) {

  for (int i = 0; i < numberOfAxis; i++) {

    Serial.print(averageVelocity[coordinateSystem][i]);

    Serial.print(",");
  }

  Serial.println();
}
```

## *A.7 Ispis na LCD zaslon*

Funkcija ispisuje brzinu na LCD zaslon. Pojedine komponente ispisuju se u zasebne redove. Broj ispisanih komponenti ograničen je na dva zbog dimenzija korištenog zaslona.
Prima: indeks željenog koordinatnog sustava
Vraća: ništa

```
void LCDPrintVelocity (int coordinateSystem) {

  lcd.clear();

  for (int i = 0; i < min(numberOfAxis, 2); i++) {

    lcd.setCursor(0, i);

    lcd.print(coordinateSystemMarking[coordinateSystem][i]);

    lcd.print(": ");

    lcd.print(averageVelocity[coordinateSystem][i]);
  }
}
```

# Literatura


[1] Anemometer, National Geographic Encyclopedia, https://www.nationalgeographic.org/encyclopedia/anemometer/, 11. rujna 2020.

[2] Drone Safety: Here's a Preflight Checklist Before Flying A Drone, Ennologic, https://ennologic.com/drone-safety/, 11. rujna 2020.

[3] 10 Best Boat Anemometers (Reviews Updated 2020), Solar Sailor, https://www.solarsailor.com/best-boat-anemometers/, 11. rujna 2020.

[4] The best anemometers in the world, Surfer Today, https://www.surfertoday.com/windsurfing/the-best-anemometers-in-the-world, 11. rujna 2020.

[5] The best wind meters for long range shooting, Wind and Weather Tools, https://windandweathertools.com/the-best-wind-meters-for-long-range-shooting/, 11. rujna 2020.

[6] Anemometer, HVAC Troubleshooting, https://www.hvacspecialists.info/service/anemometer.html, 11. rujna 2020.

[7] Arduino web stranica, https://www.arduino.cc, 11. rujna 2020.

[8] Arduino Uno, https://www.tindie.com/products/upgradeindustries/hc-sr05-hy-srf05-precision-ultrasonic-sensor/, 11. rujna 2020.

[9] HY-SRF05, https://www.tindie.com/products/upgradeindustries/hc-sr05-hy-srf05-precision-ultrasonic-sensor/, 11. rujna 2020.

[10] VMA306, https://www.velleman.eu/products/view/?id=435526, 11. rujna 2020.

[11] Arduino Starter Kit, https://store.arduino.cc/genuino-starter-kit, 11. rujna 2020.

[12] Arduino-CLI programski paket, https://arduino.github.io/arduino-cli/latest/, 11. rujna 2020.

[13] Testo 410i anemometar, tehnički podatci, https://www.testo.com/en-US/testo-410i/p/0560-1410, 11. rujna 2020.

[14] The Arduino Projects Book, 1st ed., Arduino, 2015, str. 116

[15] Bronštejn, I. N. et. al., Matematički priručnik, četvrto izdanje. Zagreb : Golden marketing - Tehnička knjiga, 2004.