

**SLOVENSKÁ TECHNICKÁ UNIVERZITA V
BRATISLAVE
STROJNÍCKA FAKULTA**

**BoBSHIELD: MINIATURIZOVANÝ EXPERIMENT
„GUĽÔČKA NA TYČI“**

Bakalárska práca

SjF-13432-104853

**SLOVENSKÁ TECHNICKÁ UNIVERZITA V
BRATISLAVE
STROJNÍCKA FAKULTA**

**BoBSHIELD: MINIATURIZOVANÝ EXPERIMENT
„GUĽÔČKA NA TYČI“**

Bakalárska práca

SjF-13432-104853

Študijný odbor:	Automatizácia a informatizácia strojov a procesov
Študijný program:	5.2.14 automatizácia
Školiace pracovisko:	Ústav automatizácie, merania a aplikovanej informatiky
Vedúci záverečnej práce:	Ing. Mgr. Anna Vargová
Konzultant:	Ing. Erik Mikuláš

Bratislava, 2022

Michal Bíro

Tu je zaviazané zadanie záverečnej práce (v jednom odovzdanom výtlačku originál zadania, v ďalších výtlačkoch kópie zadania). Nezabudnite nahradiť túto stranu so zadáním. Zaviazať zadanie do práce je povinné.

Čestné prehlásenie

Vyhlasujem, že predloženú záverečnú prácu som vypracoval samostatne pod vedením vedúceho záverečnej práce, s použitím odbornej literatúry a ďalších informačných zdrojov, ktoré sú citované v práci a uvedené v zozname použitej literatúry. Ako autor záverečnej práce ďalej prehlasujem, že som v súvislosti s jej vytvorením neporušil autorské práva tretích osôb.

Bratislava, 20. mája 2022

.....
Vlastnoručný podpis

Na tomto mieste môže byť poďakovania napr. vedúcemu diplomovej práce, resp. konzultantom, za pripomienky a odbornú pomoc pri vypracovaní diplomovej práce. Poďakovanie je nepovinnou ale obvyklou súčasťou záverečných prác.

Vzor: Ďakujem vedúcemu diplomovej práce, doc. Ing. Jozefovi Jazvečíkovi, PhD., za odbornú pomoc pri vypracovaní diplomovej práce. Chcem poďakovať aj konzultantovi diplomovej práce, Ing. Jánovi Čerešničkovi, za pomoc a pripomienky pri spracovaní nameraných hodnôt.

Bratislava, 20. mája 2018

Bc. Jožko Mrkvička

Názov práce: Miniaturizovaný experiment "gulôčka na tyči"

Kľúčové slová: - Arduino, ToF, servomotor gulôčka na tyči, riadenie

Abstrakt: Cieľom bakalárskej práce je vylepšenie existujúceho zariadenia BOBShield, ktoré reprezentuje experiment s názvom „gulôčka na tyči“ a je súčasťou projektu AutomationShield. V prvej kapitole rozoberá samotný experiment a opisuje pôvodné verzie zariadenia. V druhej kapitole hovorí o zmenách, ktoré sme na zariadení vykonali, vysvetľuje dôvody, prečo sme sa pre dané zmeny rozhodli a ukazuje výsledný model nášho zariadenia. V poslednej kapitole sa venuje softvérovej časti zariadenia, kde hovorí o API v programoch Simulink a Arduino IDE, ktoré sme pre zariadenie upravili alebo vytvorili a príkladoch, ktoré sme testovali na novom hardvéry. V závere hovorí o priestore na zlepšenie daného zariadenia v budúcnosti.

Title: Force measurement by strain gauges

Keywords: - Arduino, ToF, servomotor, ball on beam, control

Abstract: The aim of this bachelor thesis is to improve the existing BOBShield device, which represents an experiment called “ball on beam” and is part of the AutomationShield project. In the first chapter analyze the experiment itself and describes older versions of the device. In next chapter talks about the changes we have made to the device, explains the reasons for their implementation and shows the final model of our device. The last chapter deals with the software of device, which consists of APIs made in programming environments Simulink and Arduino IDE and examples made in these environments, which were tested on the new hardware. At the end it talks about possible improvements that can be made to the device in the future and outcomes of this thesis.

Obsah

Úvod	1
1 BOBShield	2
1.1 AutomationShield	2
1.2 Experiment - Gulôčka na tyči	3
1.3 Bobshield R2	4
1.3.1 R2 softvér	4
1.3.2 R2 hardvér	5
2 R3 hardvér	6
2.1 Schéma zapojenia	6
2.2 Komponenty	8
2.2.1 Servo	10
2.2.2 ToF snímač - VL6180X	12
2.2.3 Gyroskop - MPU 6050	14
2.2.4 Lineárny regulátor napätia - LDO	15
2.3 Teleso	15
2.4 Prevod rotačného pohybu - servo, trubička	18
2.5 3D prvky	20
3 BoBshield API	24
3.1 API pre Arduino IDE	24
3.1.1 Filtrácia	24
3.1.2 Knižnica	27
3.2 API pre Simulink	29
3.2.1 Knižnica	29
3.2.2 PID príklad	31
4 Záver	33
Literatúra	34

Úvod

Oblasť automatizácie a automatického riadenia sa rýchlo rozvíja a v modernom svete má veľké zastúpenie v oblasti výroby. Vo svete sú čoraz viac využívané automatické výrobné linky, prevádzky a závody, práve kvôli zvyšovaniu produktivity a redukciou vznikajúcich chýb vo výrobe. Aby mohol rozvoj napredovať je potrebné pre danú oblasť podporovať vzdelávanie budúcich technikov a vedcov. Práve kvalita vzdelávania sa totiž výrazne odrazí v nasledujúcich rokoch ich práce.

Hoci dôležitou časťou vzdelávania je práve teória riadenia, s ktorou sa študenti univerzít stretávajú na prednáškach, platí, že človek si najviac zapamätá, keď si to môže aj vyskúšať v praxi. Nie každý má však túto možnosť a nie každá univerzita disponuje rovnako vybavenými laboratóriami. Preto je pre mnohých niekedy takmer nemožné stretnúť sa počas štúdia s praxou a to hlavne v časoch, kedy sa študenti nemôžu zúčastniť prezenčnej vzdelávacej formy. Absencia praktických ukážok môže mať za následok neskoršie problémy pri prechode študentov na pracovný trh.

AutomationShield so svojimi zariadeniami prináša jednoduché riešenie tohto problému, kedy tvorbou zariadení minimálnych rozmerov a nákladov, poskytuje študentom možnosť vyskúšať si riadenie systému v praxi. Vďaka svojej cene nie sú pre školu tieto zariadenia finančne náročné a svojimi šikovnými rozmermi poskytujú študentom možnosť pracovať na nich aj z domova., čo je viac ako vítané pri dištančnej forme štúdia. Momentálne sa v projekte nachádza 11 zariadení a BOBShield je jedným z nich. Zariadenie už prešlo určitým vývojom a momentálne existujú 2 verzie. Hoci pri druhom zariadení prišlo ku viacerým zmenám a vylepšeniam oproti prvej verzii, stále poskytuje priestor na vylepšenie. Cieľom práce bolo opäť urobiť analýzu zariadenia a prísť s nápadmi na vylepšenie, či už hardvéru alebo softvéru. V oblasti softvéru išlo o vývoj API v programovacom prostredí Simulink a tvorba príkladu na riadenie systému. Pri hardvéru sme sa zamerali na použité komponenty, kde sme hľadali ich varianty, nachádzajúce sa na trhu a analyzovali sme ako by ich zmena vplývala na chod zariadenia.

Túto tému som si vybral hlavne z dôvodu . Tiež považujem celú myšlienku AutomationShieldu veľmi efektívnu a praktickú. Páči sa mi snaha poskytnúť študentom priestor pre rozvoj a vzdelávanie nie len v teoretickej rovine ale aj v praxi. Hoci ide o v princípe jednoduché zariadenia, každé má veľký potenciál a proces vývoja a vylepšovania jednotlivých verzií a prevedení so sebou prináša taktiež drahocenné skúsenosti a vedomosti.

1 BOBShield

BOBShield je zariadenie, ktoré vzniklo ako súčasť väčšieho projektu AutomationShield. Predstavuje nízko nákladový miniaturizovaný systém - Gulička na tyči, ktorý slúži ako učebná pomôcka pri predmetoch zaoberajúcich sa riadením systémov. Zariadenie používa servomotor pre otáčanie trubičkou, v ktorej sa pohybuje gulička. Cieľom experimentu je nastaviť naklonenie trubičky tak aby bolo sa gulička nachádzala v nami požadovanej pozícii alebo aby sledovala nami určenú referenčnú trajektóriu. Zariadenie svojimi malými rozmermi a nízkymi nákladmi na výrobu poskytuje možnosť pre študentov vyskúšať si riadenie v praxi mimo zariadení v školských laboratóriách, ktorých dostupnosť výrazne náročnejšia.

1.1 AutomationShield

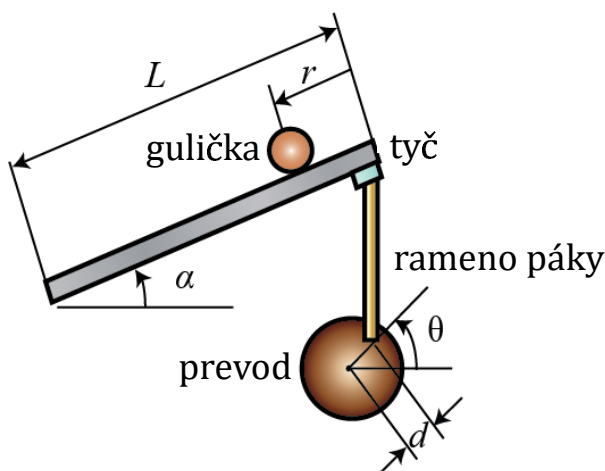
AutomationShield je projekt zameraný na tvorbu jednoduchých a cenovo dostupných zariadení, ktoré slúžia ako učebné pomôcky pri výučbe riadenia systémov. Keďže cieľom je práve dostupnosť pre všetkých, ktorý sa zaujímajú o riadenie a regulovanie systémov, ide o voľne dostupný hardvér a softvér. Základom celého projektu je práve mikropočítač, ktorý na riadenie využívame. Ide o Arduino, taktiež opensource hardvér a softvér, ktorý má medzi študentami pomerne veľkú základňu a je veľmi obľúbenou voľbou pre výučbu mikroprocesorov techniky na univerzitách a školách technického zamerania. Na tvorbu jednotlivých zariadení sa používajú dosky rozširujúce funkcionality samotnej arduino dosky, vo svete všeobecne známe ako „shields“. V podstate nejde o nič iné než o dosku plošných spojov (PCB), ktorá sa stará o elektrické spojenie medzi jednotlivými komponentami na zariadení a arduinom. Keďže väčšina typov arduino dosiek dodržiava zaužívané rozloženie pinov na doske, nie je používanie shields obmedzené len je jednu z nich. Z dôvodu dostupnosti, je podstatnou časťou tohto projektu práve nízka cena výsledného zariadenia. V priemere dosahujú jednotlivé shields cenu okolo 20 eur. Ďalšou časťou sú práve minimálne rozmery zariadení, čo poskytuje jednoduchú manipuláciu a možnosť využívať ich naozaj kdekoľvek, čo samo o sebe robí tento projekt veľmi sympatickým a pre používateľa vytvára dojem jednoduchosti a prístupnosti.

V projekte existuje niekoľko rôznych typov zariadení, ktoré simulujú riadenia rozličných systémov, no či už ide o regulovanie teploty, tlaku alebo pozície telesa v systéme, jedná sa o miniaturizované verzie väčších systémov. Okrem hardvéru je v AutomationShielde vyvíjaný aj softvér pre riadenie týchto systé-

mov. Medzi programy využívané pre tvorbu API - rozhraní pre programovanie aplikácií patrí v základe Arduino IDE, MATLAB a Simulink.

1.2 Experiment - Gulôčka na tyči

Gulička na tyči – po anglicky Ball on Beam, odkiaľ pochádza aj názov nášho zariadenie, je dynamický systém, ktorý pozostáva z dlhej tyče pripojenej k servomotoru, ktorá sa jeho pôsobením nakláňa a guličky, ktorá sa pohybuje po tyči dopredu a dozadu, v závislosti od jej naklonenia. Ide o veľmi populárny príklad používaný pri výučbe teórie riadenia. Gulička na tyči je teda nestabilný, nelineárny systém s otvorenou slučkou. Aj pri malej výchylke trubičky sa gulička skotúľa do jednej z krajných polôh tyče. Stabilizáciu guličky v určitej pozícii je potrebný riadiaci systém, ktorý meria polohu guličky na tyči a na základe získaných hodnôt vyhodnocuje nastavenie polohy tyče. Na obrázku 1.1 môžeme vidieť parametre systému, kde zmena uhla θ pôsobí cez rameno páky na tyč a spôsobí jej otočenia o uhol α . Keď sa zmení uhol z horizontálnej pozície, gravitácia zapríčiní pohyb guličky po tyči. Úlohou riadiaceho systému je teda aby sme vedeli ovládať pozíciu guličky.

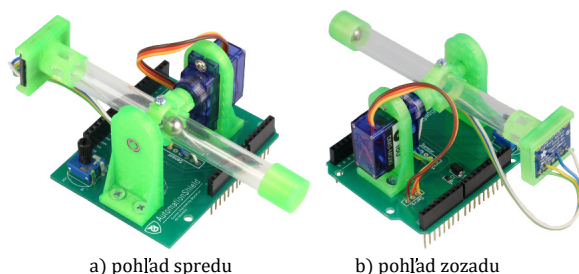


Obr. 1.1: Systém gulička na tyči [5]

Náš systém sa od systému na obrázku mierne líši. Namiesto tyče sa gulička pohybuje uprostred trubičky a jej stred otáčania sa nenachádza na konci trubičky ako to môžeme vidieť na obr. 1.1. Uložená je v držiaku, ktorý sa nachádza v jej strede a práve pomocou neho je na ňu prenášaný krútiaci moment zo servomotora. V pôvodnej verzii sa teda nenachádzal ani prevod spomenutý na obrázku, držiak bol priamo pripevnený k servomotoru. Prevod sme však pri vylepšovaní hardvéru zahrnuli do jeho zmien, no oproti klasickému riešeniu uvedenému na obrázku sme zvolil náš vlastný originálny prístup. Výhodnou tohto spôsobu zostavenia systému sú jeho rozmery, sú dostatočne malé aby sme vedeli celý model uložiť na Arduino shield.

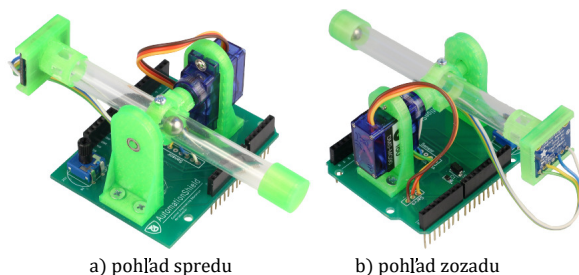
1.3 Bobshield R2

Vývoj zariadenia je náročný proces vyžadujúci si určitý čas, skúšanie nových možností a vylepšení, či už sa jedná o hardvérovú alebo softvérovú časť. Používaním zariadenia postupne dochádzame k novým nápadom na vylepšenie a tie postupne implementujeme a doladujeme, čím sa blížime k ideálnemu riešeniu. Tento stav však takmer nikdy nedosiahneme vzhľadom k neustálemu vývoju a preto je vždy priestor na zmeny a aktualizáciu projektu. Aj BOBShield si už prešiel mnohými zmenami, čo sa použitých komponentov, dizajnu alebo API týka. Prvá verzia BOBShield R1 (obr.1.2) vznikla ako výsledok práce študentov na predmete mikroprocesorová technika 2, na ústave Automatizácie a merania STU v Bratislave.



Obr. 1.2: BOBShield R1 [4]

Poslednou verziou je BOBShield R2 (obr. 1.3) a bola vytvorená ako súčasť diplomovej práce na Strojníckej fakulte STU v Bratislave. Oproti prvej verzii prišlo k zmenám hardvéru, kde bolo zmenené rozloženie komponentov na shielde, došlo k vylepšeniu 3D prvkov a zmene veľkosti trubičky a guľičky. Tiež boli vytvorené nové príklady pre zariadenie v protredí Arduino IDE ako napríklad identifikácia systému.



Obr. 1.3: BOBShield R2 [4]

1.3.1 R2 softvér

Z programových prostredí, ktoré sa v projekte AutomationShield využívajú k riadeniu systému, konkrétne ide o Arduino IDE, MATLAB a Simulink bolo

vytvorené API (aplikačné programovacie prostredie) len v prostredí Arduino IDE. Pozostáva z knižnice BOBShield.h a niekoľkých príkladov ako je regulácia alebo identifikácia systému. To ponúka priestor pre tvorbu nových API v už spomenutých programovacích prostrediach.

1.3.2 R2 hardvér

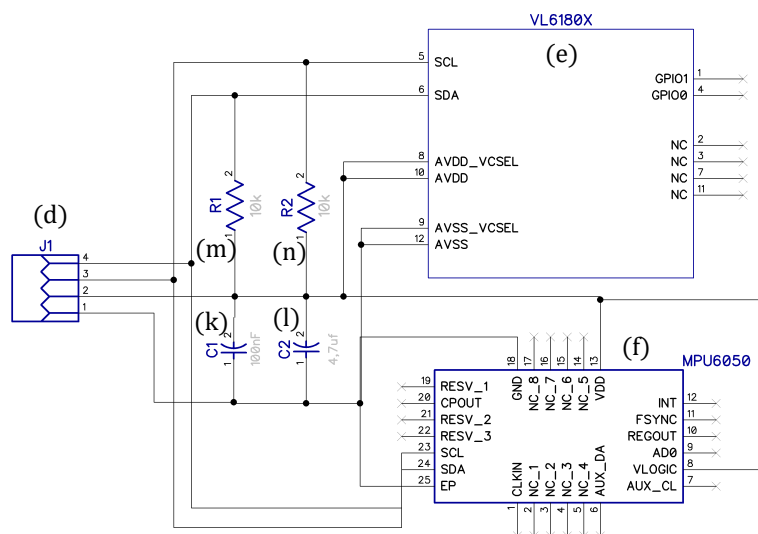
Z hľadiska hardvéru ponúka verzia R2 priestor pre analýzu použitých komponentov a ich porovnanie s novými možnosťami ponúkanými na trhu. Tiež je možné zvýšenie presnosti otáčania trubičky o implementáciou už spomínaného prevodu medzi servomotorom a držiakom trubičky. Z toho následne vyplýva aj zmena 3D prvkov a celkového dizajnu zariadenia.

2 R3 hardvér

Nasledujúca kapitola je venovaná vyhotoveniu zariadenia z pohľadu hardvéru. Hovorí o častiach, z ktorých sa zariadenie skladá, o ich parametroch, funkciách a vlastnostiach. Taktiež porovnáva poslednú verziu vyhotovenia – verzia R3, so staršími verziami zariadenia, vysvetľuje, prečo sme sa pre dané zmeny rozhodli a ako vplývajú na celkové fungovanie zariadenia. Medzi výraznejšie zmeny patrí napríklad implementácia nového snímača - gyroskopa alebo nový typ prevodového systému. Kapitola najprv vysvetľuje schému zapojenia nášho zariadenia, neskôr detailnejšie opisuje jednotlivé komponenty, z ktorých sa zariadenie skladá. Tiež opisuje výber vhodného telesa pre náš systém, implementáciu nového spôsobu prevodu rotačného pohybu zo servomotora na trubičku a tvorbu nových 3D prvkov. V závere sa nachádza zoznam všetkých komponentov a súčiastok, z ktorých sa zariadenie skladá a hovorí o jeho výslednej cene.

2.1 Schéma zapojenia

Schéma zapojenia nášho zariadenia prešla od poslednej verzie niekoľkými zmenami. Celkovo by sa dala rozdeliť na 2 oddelené schémy, ktoré sú vzájomne prepojené pomocou FFC (flat flexible cable) káblu. Oddelenie schém je potrebné z dôvodu, že senzory na snímanie polohy guľičky v trubičke a pootočenia ramena musia byť umiestnené na trubičke. Z toho vyplýva že schéma celého nášeho zariadenia musí byť oddelená aby mohli byť vytvorené 2 PCB dosky, ktoré v celku tvoria jednu schému. Prvou je časť zariadenia nachádzajúca sa na hlavnej PCB doske (obr. 2.1), ktorú priamo zapájame do Arduina. Obsahuje servomotor (a) , ktorý je napájaný 5 V a je pripojený k arduinu () prostredníctvom pinu 9, cez ktorý prijíma od arduina PWM signál, potrebný na riadenie polohy otočenia jeho hriadeľa. K servomotoru je pripojený aj kondenzátor (g) a dióda (j). Úlohou kondenzátora je vyhladiť prechodové javy, ktoré by mohli ovplyvniť napájanie serva a dióda slúži ako ochrana pred poškodením obvodov spätným EMF – elektromotorickým napätím. Potenciometer (c) je napájaný napätím 3,3 V a k arduinu je pripojený pomocou analógového pinu A0, ktorý slúži na získavanie hodnôt, ktoré neskôr transformujeme do formy, v ktorej môžu byť použité ako referenčná hodnota pre riadenie polohy. Zmenou v schéme zapojenia oproti poslednej verzii je implementácia lineárneho regulátora napätia – LDO (b), potrebného pre zmenu hladiny napätia pre napájanie snímačov. Je pripojený k zdroju napätia o veľkosti 5V, ktorého hladinu upravuje a už zmenenú posiela na konektor pre FFC kábel (d). Ten slúži na



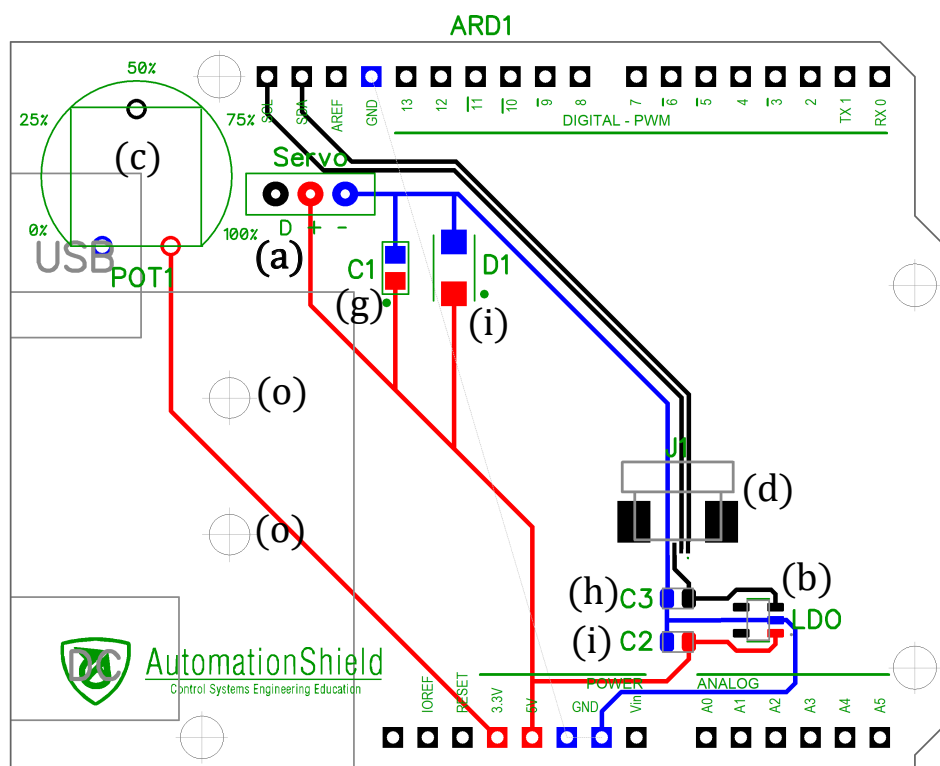
Obr. 2.2: Schéma PCB dosky so snímačmi

a rukou užívateľa pri nastavovaní referenčnej hodnoty. Umiestnenie kondenzátora (g) a diódy (i) je tiež podľa pôvodnej verzii. Umiestnenie konektora pre FFC káble (d) sme zvolili tak, aby sa po montáži kompletného zariadenia nachádzal pod trubičkou a tak nedochádzalo k jeho zahnutiu a spĺňalo to aj estetickú funkciu. Na základe jeho pozície došlo k umiestneniu spoja pre LDO (b), keďže jeho úlohou je zmena hladiny napätia do rozsahu napájacích napätí snímačov, nachádza sa medzi konektorom FFC kábla (d) a zdrojom napätia z arduina. Kondenzátory (h) (i), na základe informácií z datasheetu [1] potrebné pre fungovanie LDO, sú umiestnené v jeho tesnej blízkosti ako je odporúčané.

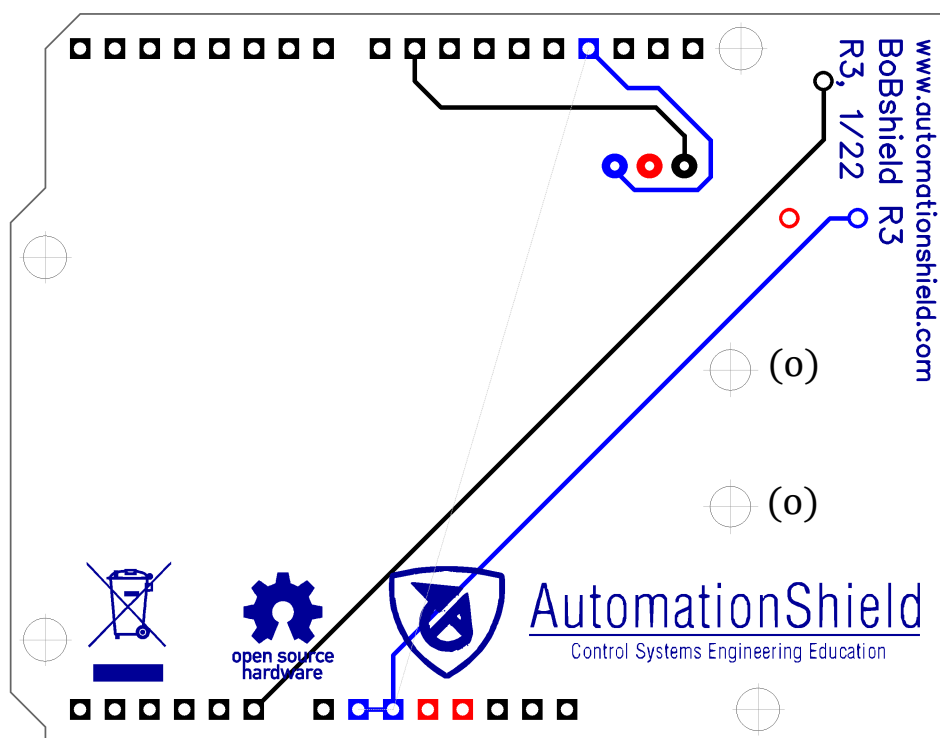
Pri návrhu PCB dosky určenej pre snímače (obr. 2.5), sme neboli obmedzení žiadnym konkrétnym tvarom dosky, preto bolo našim cieľom dosiahnuť rozloženie jednotlivých komponentov optimálnym spôsobom aby sme dosiahli čo najmenšie rozmery. Na hornej časti dosky môžeme nájsť plošný spoj pre umiestnenie snímača MPU-6050 (f), konektor pre pripojenie FFC kábla (d) a jednotlivé komponenty (g) (h) (i) (j), ktorých implementácia vyplýva zo schémy zapojenia jednotlivých snímačov, ktorú je možné nájsť v datasheetoch [1]. Na spodnej strane PCB dosky sa nachádza spoj pre umiestnenie snímača VL6180X (e). Umiestnenie snímača na spodnej strane je z dôvodu aby smeroval do vnútra trubičky, v ktorej sa nachádza guľička. Po umiestnení PCB dosky do držiaka na konci trubičky, snímač VL6180X smeruje do vnútra trubičky a z naopak FFC kábel smeruje von z trubičky, čo poskytuje jeho jednoduchú montáž.

2.2 Komponenty

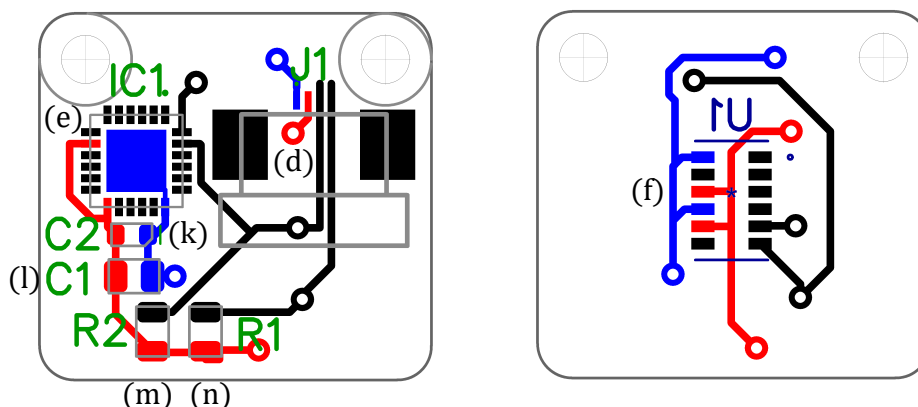
Nasledujúca podkapitola detailnejšie opisuje vlastnosti komponentov, s ktorými sme sa už stretli pri schéme zapojenia. Hovorí o ich parametroch, funkciách, ktoré v zariadení zastrešujú a tiež opisuje proces ich výberu a dôvod



Obr. 2.3: PCB doska - pohľad z hora



Obr. 2.4: PCB doska - pohľad z dola



Obr. 2.5: PCB doska pre snímače

prečo sme si pre naše zariadenie zvolili práve tieto komponenty.

Pre riadenie a regulovanie akéhokoľvek systému je potrebné mať vstup do systému, ktorým nastavujeme zásah do systému. O túto časť riadiacej slučky sa starajú aktuátory. Výstup zo systému zase prezentujú hodnoty, ktoré získavame pomocou snímačov a na ich základe môžeme vyhodnocovať aktuálny stav systému.

V našom zariadení sa o jednotlivé funkcie starajú nasledujúce komponenty:

- snímače – Tof snímač vzdialenosti, Gyroskop
- aktuátory – Servo motor

2.2.1 Servo

Servomotor patrí spolu so snímačmi ku komponentom, ktorých vlastnosti výrazne vplyvajú na kvalitu riadenia systému. V našom systéme plní funkciu aktuátora, teda sa stará o akčný zásah do systému a tak ovplyvňuje náklon trubičky a následne pohyb guľičky, ktorej polohu riadime. Na základe dôležitosti tohto komponentu sme sa rozhodli pre analýzu ponúkaných možností na trhu a ich následné porovnanie so servomotorom v pôvodnej verzii. Vo verzii zariadenia R2 bol použitý analógový servomotor s kovovými prevodmi MG-90. Ide o veľmi rozšírený servomotor bežne sa nachádzajúci v začiatočníckych arduino kitchoch, určených pre domáce projekty. Z parametrov môžeme vidieť, že spĺňa požiadavku na malé rozmery, taktiež podľa rozsahu pracovného napätie odpadá potreba jeho úpravy, a servo môže byť napájané priamo z arduina. Vďaka kovovým prevodom je jeho životnosť vyššia a dosahuje aj lepšie presnosti oproti plastovým verziám. Čím ale servo najviac vyniká oproti iným typom a verziám je jeho nízka cena, ktorá sa pohybuje okolo 4 eur.

Parametre:

- Hmotnosť: 12 g
- Rozmery: 22.3 x 11.8 x 26.3 mm

- Krútiaci moment: 1,8 kg (4,8 V)
- Rýchlosť: 0,12 sekundy / 60 ° (4,8 V)
- Prevádzkové napätie: 4.8V - 6.0V
- Typ: Analógové
- prevody: Kovové
- cena: 6 eur

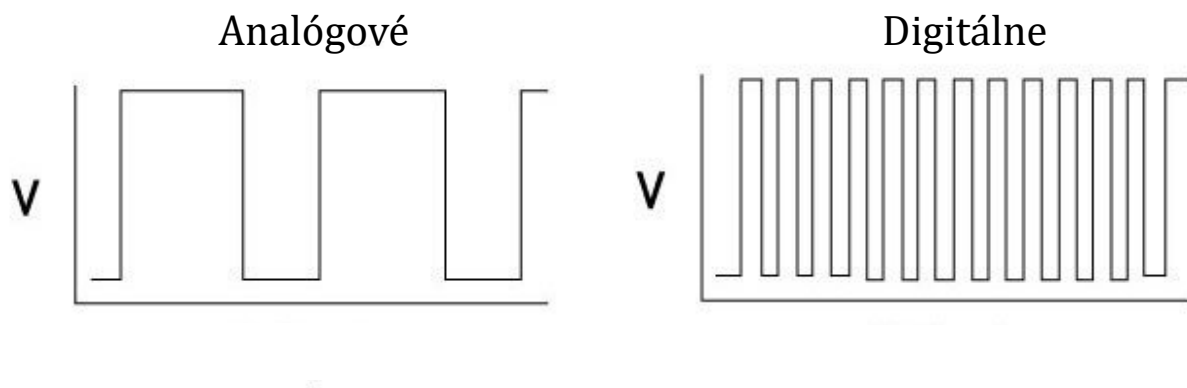
Zmenu, ktorú sme teda zvažovali bolo zvoliť namiesto analógového servomotora digitálny. Ide o dilemu, s ktorou sa pri výbere servomotora stretáme veľmi často. Každý typ má svoje výhody a nevýhody no najväčším rozdielovým parametrom, od ktorého závisí výber typu serva je frekvencia PWM signálu, s ktorým dané servo pracuje. Zároveň od toho závisia aj ostatné výhody a nevýhody, no podstata je ukrytá práve za týmto faktorom.

Princíp fungovania servomotorov je založený na prijímaní napäťového signálu, ktorý prichádza v podobe PWM (pulse width modulation – impulzová šírková modulácia) signálu. PWM signál posiela motoru elektrický impulz rôznej šírky, na základe ktorej sa otočí hriadeľ do polohy, ktorú požadujeme. Analógové servomotory pracujú so signálom o frekvencii 50 Hz, čo znamená že perióda jedného cyklu je 0,02 s, čo sa môže zdať ako dostatočne malý čas no pri riadení systému, tak dynamickeho ako je ten náš, ide o pomerne veľký časový interval. Pri digitálnych servách dosahuje PWM signál výrazne vyššie frekvencie – pohybujeme sa v stovkách hertzov. Ak si zoberieme PWM signál o frekvencii 500 Hz, perióda jedného cyklu je 0,002 s, čo je 10 krát viac cyklov za rovnaký čas ako pri analógovom servomotore. To má za následok zníženie času kedy do serva nie je posiela-ný žiadny elektrický signál. Graficky znázornený rozdiel medzi dĺžkami cyklov PWM signálu pre digitálne a analógové servomotory, môžeme vidieť na obrázku 2.1. Zvýšením frekvencii teda digitálne servá dosahujú oveľa rýchlejšie reakcie na zmeny požadovanej polohy, lepšie rozlíšenie a väčší krútiaci moment. Na druhú stranu však s väčšou frekvenciou prichádza aj väčšia spotreba energie, vyššia cena a hluk pri prevádzke.

My sme sa rozhodli pre zmenu z analógového na digitálne, z dôvodu potreby zrýchlenia reakcie aktuátora na zmenu polohy guľičky a tiež lepšieho rozlíšenia pulzu. Za servomotor sme si preto zvolili SH-0253, digitálne servo značky SAVOX. Parametrami je veľmi podobný pôvodnému servomotoru MG-90.

Parametre:

- Hmotnosť: 13,6 g
- Rozmery: 23,8 x 12 x 25,4 mm
- Krútiaci moment: 1,8 kg (4,8 V) – 2,2 (6 V)
- Rýchlosť: 0,12 s/60 ° (4,8 V) – 0,9 s/60 ° (6 V)
- Prevádzkové napätie: 4.8V - 6.0V



Obr. 2.6: Dĺžka cyklu PWM signálu pre analógové a digitálne servo

- Typ: Digitálne
- prevody: Kovové
- cena: 16 eur

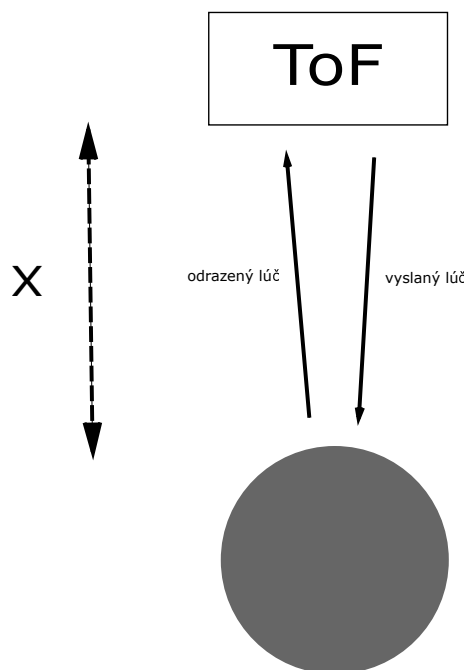
2.2.2 ToF snímač - VL6180X

Úlohou snímača je snímať polohu guľičky v trubičke a dáta posielat' mikropočítaču. Poskytuje nám aktuálnu hodnotu vzdialenosti, ktorá sa porovnáva s požadovanou na základe čoho regulátor vypočíta hodnotu akčného zásahu do systému.

Pri snímači vzdialenosti sme sa rozhodli ostať pri pôvodnej voľbe typu a modelu, ktorým je VL6180X (obr. 2.8). Ide o ToF snímač, ktorý meria vzdialenosť na základe vysielania, odrazu a následného prijímania svetelného lúča. Princíp fungovania (obr. 2.7) je založený na meraní času, za ktorý svetelný lúč vyslaný zo snímača doletí k telesu, ktorého pozíciu meriame, odrazí sa od neho a vráti sa späť. Na základe rovnice (rov. 2.1), kde c - predstavuje rýchlosť svetla, ΔT - čas, ktorý trvá lúču odraz a návrat k snímaču, samotný snímač vypočíta vzdialenosť - x , v ktorej sa nachádza teleso od snímača. Komunikácia s arduinom prebieha prostredníctvom I2C protokolu. Rozsah napájacieho napätia je 2,6V - 3,0V.

$$2x = c * \Delta T \quad (2.1)$$

Hoci sme zvažovali aj iné typy snímačov vzdialenosti ako napríklad ultrazvukové alebo IR (infračervené) snímače, prišli sme k záveru, že najvhodnejším typom snímača bude práve ToF vďaka jeho malým rozmerom. Z možností ToF snímačov ponúkaných na trhu je pre naše zadanie VL6180X momentálne ideálnou možnosťou z dôvodu:



Obr. 2.7: Princíp fungovania ToF senzora

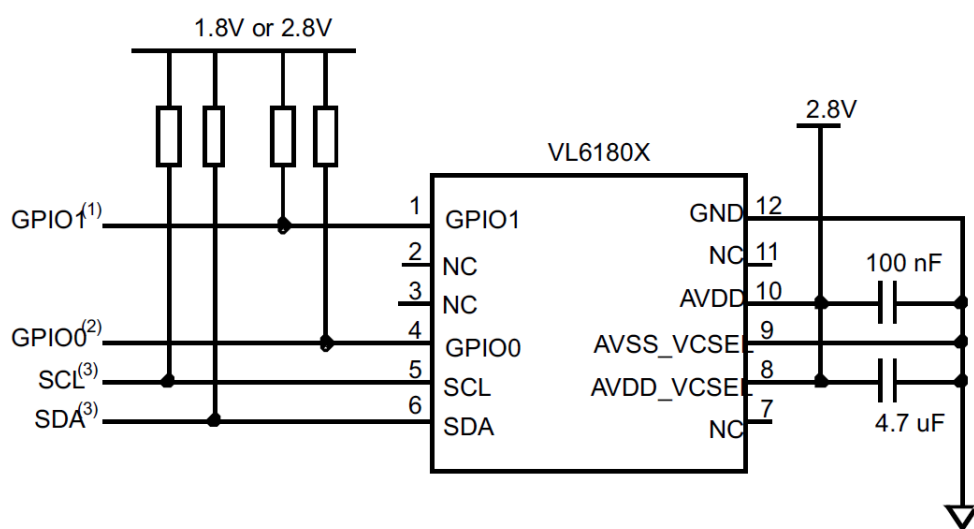
- merateľného rozsahu - 100 mm
- presnosti merania - na 1 mm
- malým rozmerom – 4.8 x 2.8 x 1.0 mm [1]

Merateľný rozsah snímača je pre nás úplne postačujúci, keďže guľička sa pohybuje v trubičke práve o dĺžke 100 mm, s tým že v koncových bodoch sa nachádzajú ešte uzavretia trubičky, takže rozsah pohybu guľičky je približne 90 mm, čo nám dáva ešte určitú rezervu. Dokonca veľa snímačov nespĺňa naše požiadavky práve z dôvodu že ich merateľný rozsah nezačína v 0 mm ale až pri vyšších hodnotách. Hoci presnosť meranie na 1 mm nie je pre riadenie systému ideálna, z ponúkaných senzorov sa nám nepodarilo v danej kategórii nájsť žiadny s lepšou presnosťou. Rozmery sú pre nás dôležité z hľadiska potreby umiestnenia snímača do jednej z koncových polôh trubičky aby sme mohli merať pozíciu guľičky. Taktiež príliš veľké rozmery ako napríklad pri ultrazvukovom snímači by pre minimálne rozmery nášho zariadenia neboli vhodné z hľadiska estetickej. Vďaka minimálnym rozmerom snímača sa nám oproti pôvodnej verzii podarilo zmenšiť diel upevnenia snímača na trubičke, čím sme zlepšili aj estetickú zložku zariadenia.

Na obrázku 2.9 môžeme vidieť schému zapojenia snímača uvedenú v datasheete. V našej aplikácii sme piny 1 a 4 (GPIO1 a GPIO0) nepoužívali a nechali sme ich voľné. Piny 5 a 6 – SCL a SDA, slúžiace na I2C komunikáciu snímača s mikropočítačom sú pomocou pull up rezistorov pripojené k napätiu. Za hodnoty pull up rezistorov sme zvolili odpor 10 k Ω . Ako môžeme vidieť v schéme piny 9 a 12 sú uzemnené a piny 10 a 8 sú zase pripojené k napájaniu, pričom sú k nim paralelne pripojené 2 kondenzátory, ktorých hodnota je uvedená



Obr. 2.8: ToF snímač - VL6180X[1]



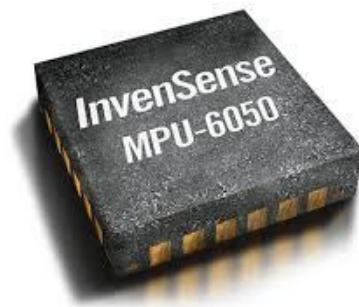
Obr. 2.9: Schéma zapojenia snímača [1]

pre daný snímač v datasheete. Taktiež je spomenuté aby ich pozícia bola čo možno najbližšia k daným pinom.

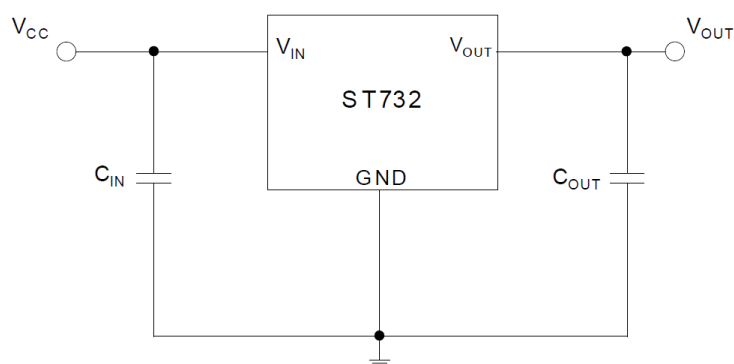
2.2.3 Gyroskop - MPU 6050

Snímač MPU 6050 (obr. 2.10) je oproti pôvodným verziám zariadenia úplnou novinkou. Ide o 6 osový snímač pohybu – 3 osový gyroskop a 3 osový akcelerometer zostavený v malom balení o rozmeroch 4 x 4 x 0,9 mm. Jeho využitie je veľmi široké a stretnúť sa s ním môžeme pri smartfónoch alebo tabletoch. Poskytuje funkcie využívané pri aplikáciách ako navigácia, rozšírená realita, monitorovanie zdravia a pohybu a mnoho ďalších. Snímač užívateľovi poskytuje možnosť nastavenia meraného rozsahu v závislosti od aplikácie, pre ktorú bude využitý, teda pre sledovanie jednak rýchlych aj pomalých pohybov. Pri gyroskope si užívateľ môže vybrať z nasledujúcich rozsahov - ± 250 , ± 500 , ± 1000 a $\pm 2000^\circ/\text{s}$. U akcelerometra ide o rozsahy $\pm 2g$, $\pm 4g$, $\pm 8g$ a $\pm 16g$. Komunikácia s arduinom prebieha rovnako ako pri ToF snímači prostredníctvom I2C protokolu a rozsah napájacieho napätia 2,375V – 3,46 V [2].

Jeho úlohou je merať uhol natočenia trubičky, na základe čoho vieme nastaviť trubičku do vodorovnej polohy.



Obr. 2.10: Schéma zapojenia snímača



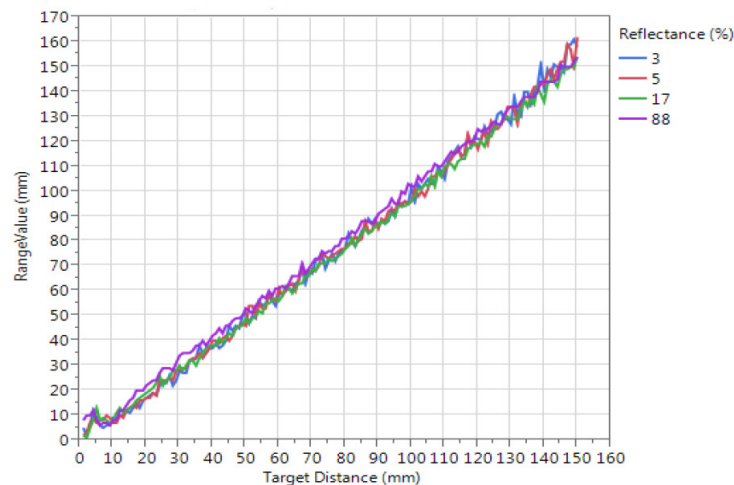
Obr. 2.11: Schéma zapojenia snímača[3]

2.2.4 Lineárny regulátor napätia - LDO

Z dôvodu tvorby vlastného braekoutu, na ktorom sú umiestnené oba naše snímače bolo potrebné nájsť vhodné napájacie napätie aby vyhovovalo obom snímačom a spadalo do ich rozsahov napätí. Pre snímač VL6180X je vstupné napätia v rozmedzí 2,6 V - 3,0 V [1] a pri MPU6050 ide o rozsah 2,375 – 3,46 V[2]. Arduino nám ponúka len 2 úrovne napájania a to 3,3 V a 5 V. Ak by sme využívali len snímač MPU6050 mohli by sme priamo využiť napájanie 3,3 V, ktoré spadá do jeho pracovného rozsahu, no pre snímač polohy je táto možnosť nevyhovujúca. Preto je potrebné upraviť ponúkané úrovne napätia z arduino dosky do rozsahu 2,6 - 3,0 V, vyhovujúcemu obom snímačom. O zmenu napätia sa stará lineárny regulátor napätia – STM732M28R od firmy STMicroelectronics , ktorý vstupné napätie v rozsahu 2,5 – 28 V prevádza na úroveň 2,8 V. Táto hladina je vyhovujúca pre oba snímače.

2.3 Teleso

Dôležitou časťou celého systému je práve guľička, pohybujúca sa v trubičke, ktorej polohu sledujeme a riadime. Na náš systém vplýva ako jej hmotnosť a tvar tak aj kvalita a farba jej povrchu. Keďže na meranie polohy guľičky používame ToF snímač, ktorý meria vzdialenosť na základe času, za ktorý sa



Obr. 2.12: Závislosť merania od odrazivosti telesa[1]

svetelný lúč vyslaný zo senzora odrazí od telesa a vráti naspäť, na kvalitu merania vplývajú aj tieto parametre.

Výrobca v datasheete [1] uvádza nezávislosť merania od farby alebo kvality povrchu telesa. Deklarované je to aj grafom na obrázku 2.12, kde môžeme vidieť, že hodnoty pre telesá rozdielnej odrazivosti sa navzájom nelíšia a nachádzajú sa v tom istom rozsahu.

Aj napriek tomu sme v našich meraniach sme mohli sledovať odlišnosti v presnosti pre rôzne typy guľôčok. Tento fakt môže byť ovplyvnený práve tvarom meraného telesa, ktorý pre funkčnosť snímača nie je ideálny, no pre potreby nášho zadania nevyhnutný. Svetelný lúč nedopadá na kolmý povrch, preto nemusí byť odrazený práve pod takým uhlom aby ho dokázal snímač adekvátne zachytiť a zanalyzovať. Na základe tohto faktu sme predpokladali, že by ideálnym riešením bolo teleso s povrchom, ktorý čo najviac rozptýli dopadajúci lúč aby šanca, že sa lúč od neho odrazí k snímaču bola čo najvyššia, čím by sa zlepšila kvalita merania. Ďalšou požiadavkou pri hľadaní ideálneho telesa bola dostatočná hmotnosť guľičky aby na zmenu stavu systému reagovala v reakčnom čase dostatočne dlhom aby náš riadiaci systém stíhal vykonávať potrebné výpočty, no zároveň nechceme aby bol zbytočne dlhý. Na túto požiadavku vplýva aj kvalita povrchu guľičky. Aby sa guľička mohla voľne pohybovať v trubičke, nič jej nebránilo a tak neplývalo na systém je potrebné aby bol jej tvar čo najviac podobný tvaru ideálnej gule. Poslednou požiadavkou bola jeho jednoduchá dostupnosť. Keďže sa jedná o open source projekt, je potrebné aby sa každý, kto by mal záujem o zostrojenie nášho zariadenia vedel ku danej guľičke bez väčších problémov dostať.

Z možností dostupných na trhu sme sa rozhodli otestovať náš snímač pre viacero typov materiálov: drevo, silikón, POM - polyoxymetylén, NBR - butadién-akrilonitrilový kaučuk (syntetická guma), NR - prírodná guma, oceľ a PP - polypropylén.

Pri meraní sme postupovali nasledovne. Guľičku sme nastavili do krajnej polohy v trubičke (vzdialenosť 90 mm), následne sme pomocouToF snímača

zmerali 100 hodnôt jej aktuálnej polohy. Hodnoty sme vložili do tabuľky a pomocou vzorca 2.2 sme vypočítali smerodajnú odchýlku pre dané meranie.

$$\sigma = \sqrt{\frac{1}{N} \sum_{i=1}^N (x_i - \bar{x})^2} \quad (2.2)$$

Tento postup sme zopakovali pre všetky materiály. V tabuľke 2.1 môžeme vidieť porovnanie smerodajných odchýlok pre jednotlivé materiály. Na základe merania, ktoré sme vykonali, môžeme tvrdiť, že náš snímač dosahuje najmenšie smerodajné odchýlky pri materiály PP – polypropylén a najväčšie odchýlky pri NBR – syntetická guma.

Materiál	silikón	POM	drevo	NBR	NR	oceľ	PP
smerodajná odchýlka	1,27	1,2891	1,3644	1,6271	1,3322	1,5	1,2595

Tabuľka 2.1: Smerodajné odchýlky pre dané materiály

Ako bolo vyššie spomenuté, dynamika systému má veľký vplyv na jeho riadenie. Z toho dôvodu je pri výbere guľičky dôležitá práve rýchlosť akou sa guľička v systéme pohybuje. Ak je guľička príliš rýchla, riadiaci systém má menej času reagovať na zmenu jej polohy, ktorá sa hlavne pri výraznej zmene referenčnej hodnoty výrazne mení. Keďže dĺžka trubičky je pomerne malá (100 mm), ak riadiaci systém nedokáže včas zareagovať na zmenu jej polohy bude dochádzať k nárazom guľičky do krajných polôh trubičky, čo je pri riadení nežiaduci jav. Z toho dôvodu sa snažíme voliť guľičku, ktorej rýchlosť je adekvátne pre našu aplikáciu. Rýchlosť guľičky je závislá jednak od jej hmotnosti, materiálu, kvality povrchu a rozmerov. Pri meraní sme zaznamenávali polohu guľičky a čas, v ktorom sa v danej polohe nachádzala. Pohyb guľičky bol z jednej krajnej polohy trubičky do druhej krajnej polohy pri naklonení trubičky o uhol 5°. Následne sme namerané hodnoty vykreslili do grafu (príloha) v programe MATLAB a pomocou jeho analýzy sme našli čas, za ktorý guľička prešla danú vzdialenosť trubičky. Postupovali sme nasledovne. V grafe sme si našli body kedy sa guľička nachádzala v vzdialenosti 80 mm a 20 mm od snímača. Z tabuľky sme si vypočítali za aký čas guľička túto vzdialenosť 60 mm prešla. Hodnoty sme zapísali do tabuľky (príloha) a vypočítali rýchlosť jednotlivých guľičiek.

Materiál	silikón	POM	drevo	NBR	NR	oceľ	PP
rýchlosť [m/s]	0,1348	0,1504	0,1412	0,0917	0,1279	0,1348	0,1412

Tabuľka 2.2: Rýchlosti guľičiek pre dané materiály

V tabuľke 2.2 môžeme vidieť porovnanie rýchlosti jednotlivých guľičiek pri pohybe mierne naklonenou trubičkou. Najpomalšie sa pohybujúcou je guľička z NBR, ktorá oproti guľičke z ocele, použitej v pôvodných verziách zariadenia dosahuje výrazne nižšie rýchlosti pohybu v trubičke. Na základe vykonaných meraní sme sa rozhodli zvoliť si za teleso v systéme guľičku vyrobenú z polypropylénu (PP). Snímač dosahuje pri danej guľičke najpresnejšie merania, takže

meraná poloha guľičky sa najviac približuje jej skutočnej polohe. Meranie s čo najmenšou smerodajnou odchylkou je pri riadení systému kľúčové, no treba uvažovať aj dynamiku guľičky. Z nameraných rýchlostí guľičiek by pre nás bola najvhodnejšou voľbou guľička z NBR, no jej presnosti merania sú výrazne horšie v porovnaní s guľičkou PP. Tá hoci na základe meraní nedosahuje najlepšie rýchlosti pre našu aplikáciu, no napriek tomu sme sa rozhodli zvoliť si ju, na základe dobrých výsledkov pri smerodajnej odchylke merania. Dobrou voľbou by mohli byť aj guľičky zo silikónu alebo materiálu NR. Tie však nespĺňajú požiadavky na tvar guľičky, keďže pri ich výrobe na ich povrchu ostáva stopa po forme, ktorá následne bráni plynulému pohybu guľičky v trubičke.

Ak porovnáme guľičku z ocele, ktorá je telesom použitým v pôvodnej verzii a guľičky z PP, ktorú sme na základe meraní vybrali, môžeme pozorovať výrazné zlepšenie merania snímača z hľadiska smerodajných odchýlok.

2.4 Prevod rotačného pohybu - servo, trubička

Aktuátorom v našom zariadení je servo motor, no aby mohol vytvárať akčný zásah (pootočenie trubičky) do systému, ktorým je naša trubička s guľičkou, musí prísť ku prenosu rotačnému pohybu.

V pôvodnej verzii zariadenia bol tento prevod riešený priamym upevnením trubičky na Servo motor, čo znamená, že otočenie trubičky sa rovnalo otočeniu serva o príslušný uhol z rozsahu $\pm 30^\circ$, ktorý bol stanovený. Najmenší možný uhol pootočenia trubičky sa teda rovnal presnosti serva : 1° . Vzhľadom na malé rozmery trubičky ide o pomerne veľký uhol. Malé možné rozlíšenie uhlu, o ktorý vieme otočiť trubičku so sebou prináša obmedzenia pri riadení systému, kedy akčný zásah nemôže nadobúdať takú množinu hodnôt. To má za následok neschopnosť nastaviť tak presný akčný zásah aby sa čo najviac približoval hodnote vypočítanej riadiacim systémom.

Naším riešením tohto problému je použitie jednoduchého prevodu, cez ktorý budeme prenášať rotačný pohyb serva na trubičky. Prevod sa skladá z 2 koliesok rozdielnych priemerov, pričom jedno je upevnené na servo motor a k druhému je pripevnená trubička. Prenos pohybu z jedného kolieska na druhé je riešené pomocou ozubeného remeňa GT2 o šírke 6 mm, bežne používanom pri krokových motoroch alebo 3D tlačiarňach. Výhodou nášho riešenia je využitie takmer celého rozsahu pohybu nášho servo motora. Kým pri pôvodnom zariadení išlo o využitie rozsahu $\pm 30^\circ$ z možných $\pm 90^\circ$, v danom riešení využívame rozsah $\pm 75^\circ$.

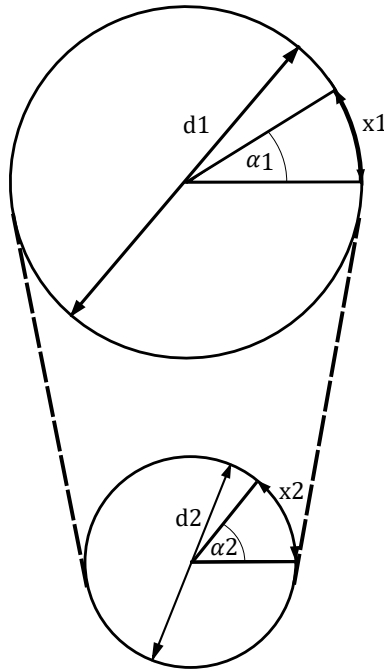
Prevodový pomer - p a následne aj rozmery koliesok (d_1 - vstupný priemer, d_2 - výstupný priemer) sme vypočítali na základe vzorcov (2.3) pre výpočet prevodov, kde najprv zistíme prevodový pomer na základe vstupného a výstupného rozsahu otočenia. Vzorce sa dajú jednoducho odvodiť na základe dĺžky posunutia remeňa - x , ktorá je na oboch kolieskach vždy rovnaká. Ide o dĺžku kružnicového oblúka, o priemere daného kolieska, ktorého uhol je rovný uhlu otočenia (obr.2.13). Tento uhol môžeme pri výpočtoch nahradiť rozsahom otáčania jednotlivých koliesok. Za výstupný rozsah sme si zvolili $\pm 15^\circ$, pre dĺžku

trubičky je tento uhol postačujúci keďže referenčná hodnota sa pohybuje od 0 po 100 mm a guľička sa pohybuje v trubičke dostatočne veľkou rýchlosťou bez väčšieho odporu, nemusí dochádzať k tak výrazným zásahom aktuátora ako je naklonenie o celých 30°, či už kladným alebo záporným smerom. Reakcia systému je aj pri nami zvolenom rozsahu dostatočne rýchla. Pre vstupný rozsah sme si zadali $\pm 75^\circ$ aby sme disponovali určitou rezervou voči maximálnemu rozsahu otočenia.

$$p = \frac{d_1}{d_2} \quad (2.3)$$

$$x_1 = \frac{\pi * d_1 * \alpha_1}{360} \quad (2.4)$$

$$x_2 = \frac{\pi * d_2 * \alpha_2}{360} \quad (2.5)$$



Obr. 2.13: Posunutie remeňa pri rotácii kolieska

$$x_1 = x_2 \quad (2.6)$$

$$p = \frac{d_1}{d_2} = \frac{\alpha_1}{\alpha_2} \quad (2.7)$$

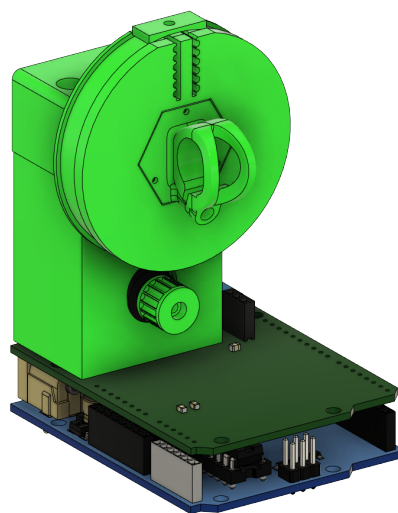
$$p = \frac{30^\circ}{150^\circ} = 0.2 \quad (2.8)$$

Na základe prevodového pomeru vieme povedať, že na 1°otočenia serva prislúcha 0,2°otočenia trubičky. Týmto prevodom sme teda 5 násobne zväčšili množinu hodnôt, ktoré môže aktuátor nadobudnúť a tým zvýšili presnosť akčného

zásahu do systému. Ak poznáme prevodový pomer rozmery koliesok získame voľbou priemeru jedného z nich a následným výpočtom priemeru druhého kolieska na základe vzorca 2.3. Za priemer kolieska pripojeného k servo motoru sme si zvolili $d1 = 10mm$. Je to dostatočná veľkosť pre uchytenie kolieska a prevod pohybu pomocou remeňa, no zároveň šetríme miesto aby sme zachovali minimálne rozmery pôvodnej verzie zariadenia. Z toho vyplýva, že priemer druhého kolieska bude $d2 = 50mm$.

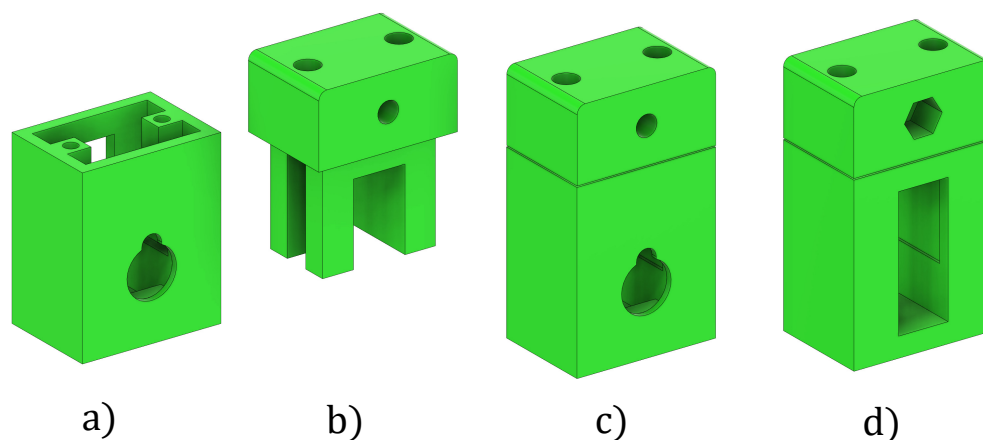
2.5 3D prvky

Z dôvodu zmeny hardvéru ako je napríklad vytvorenie vlastnej PCB dosky pre snímače, došlo k potrebe návrhu nového 3D modelu zariadenia a aktualizácii jeho starších prvkov. Najvýraznejšou zmenou je použitie prevodu (kap. 2.4) na prenos rotačného pohybu zo serva na trubičku. Pre implementáciu tohto prevodu bolo potrebné vytvoriť hneď niekoľko nových 3D prvkov a niektoré staršie prešli menšími zmenami. Tiež sme pri ich návrhu museli hľadiť na viacero požiadaviek kladených na tento prevodový systém, ako je potreba možnosti napínania remeňa alebo jeho upevnenie na prevodovom koliesku. Na obrázku 2.14 môžeme vidieť už výslednú formu 3D modelu zariadenia, ktorá sa skladá z niekoľkých samostatných prvkov, ktorých funkcie si vysvetlíme v tejto sekcii. Všetky prvky boli navrhnuté v CAD programe Fusion 360.



Obr. 2.14: 3D model zariadenia

Prvým prvkom je diel uložený priamo na PCB doske, ktorý slúži na uloženie servo motora. Môžeme ho vidieť na obrázku 2.15 a) prvok. Pri umiestnení servo motora v danom diely sme sa snažili aby sa poloha hriadeľa nachádzala v čo možno najnižšej polohe aby sme tak šetrili miesto a minimalizovali rozmery zariadenia. Pripevnenie k prvku je prevedené pomocou závrtných skrutiek. Tiež

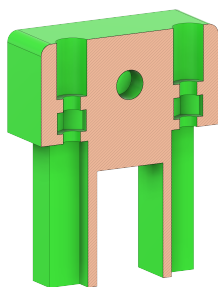


Obr. 2.15: Uloženie prvého a druhého dielu

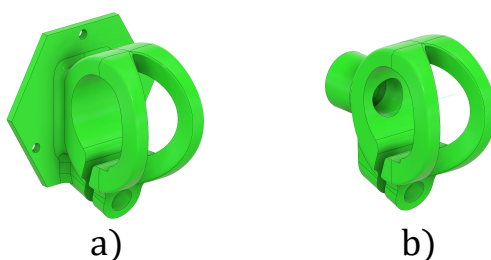
sme sa rozhodli o schovanie servo motora do vnútra prvku a jeho vkladanie cez zadnú stranu, čím jednak šetríme priestor no zároveň pri montáži nedochádza k problému s káblami servo motora, ktoré majú dostatočne veľa priestoru a nemalo by dôjsť k ich poškodeniu. Okrem toho sa v prvku nachádza drážka, v ktorú slúži na vedenie druhého dielu pri vykonávaní translačného pohybu smerom nahor a nadol, čím je splnená funkcia napínania remeňa. Taktiež slúži na pripojenie celého modelu k PCB doske pomocou 2 skrutiek.

Druhým prvkom (obr. 2.15, prvok b) je už spomenutý diel, ktorý sa zasúva do prvého dielu. Translačný pohyb je zabezpečený 2 skrutkami M3 o dĺžke 20 mm, ktoré sa zapierajú o spodný diel a tým zabezpečujú zdvih celého dielu. Výška zdvihu sa nastavuje zaskrutkovaním skrutiek do samoistiacich matíc, ktoré sú schované v samotnom telese (obr. 2.16). Ich uloženie sme vykonali zastaveným tlačom v danej vrstve, v ktorej sa v telese nachádzajú a po vložení matíc bola tlač znova obnovená. Aby nedochádzalo k postupnému uvoľňovaniu spoja pri prevádzke zariadenia, čo by malo za následok uvoľnenie napätia v remeni, rozhodli sme sa pre voľbu samotesniacich matíc. Sila, ktorou je teleso ťahané smerom dole je vyvolaná tlakom remeňa na prevodové koliesko, ktoré je pomocou skrutky M5 uchytené na tomto prvku. V zadnej časti môžeme vidieť výrez na umiestnenie matice pre daný skrutkový spoj. Teda okrem napínania remeňa je jeho úlohou taktiež uchytenie veľkého prevodového kolieska. Na obrázku 2.15 môžeme tiež vidieť uloženie druhého dielu v prvom diele z prednej a) a zadnej b) strany.

Ďalším prvkom je uchytenie trubičky na prevodové koliesko (obr. 2.17, prvok a). Ide o aktualizáciu a miernu úpravu pôvodnej verzie tohto prvku, ktorú môžeme vidieť na obrázku 2.17 ako prvok b). Keďže pôvodne sa zachytával priamo na hriadeľ servo motora, zmenou prešla práve časť uchytenia prvku. Pripevnenie držiaka ku koliesku sme sa rozhodli riešiť pomocou 3 samorezných skrutiek. Pre zabezpečenie jeho presného uloženia, tvar držiaka sme navrhli tak



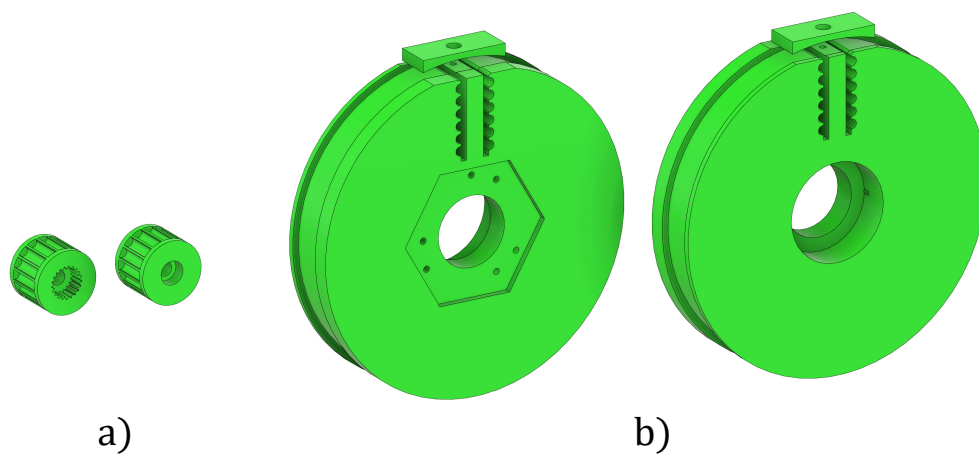
Obr. 2.16: Uloženie poistných matíc



Obr. 2.17: Uloženie poistných matíc

aby pri montáži zapadol do kolieska, v ktorom sa nachádza výrez rovnakého tvaru.

Poslednými prvkami sú samotné prevodové kolieska, ktoré spĺňajú nami vypočítané parametre (prevodový pomer). Môžeme ich vidieť na obrázku 2.18. Na koliesku o priemere 10 mm sa nachádza po celom vonkajšom obvode ozubenie dimenzované pre náš remeň a pre upevnenie kolieska ku hriadeľu serva sme namodelovali vnútorné ozubenie, ktoré zodpovedá zúbkovaniu hriadeľa. Uchytenie k servo motoru po nasunutí kolieska na hriadeľ je prevedené pomocou skrutky M2. Koliesko o priemere 50 mm má po obvode drážku pre vedenie remeňa, ktorá bráni pred jeho vysunutím. Upevnenie remeňa na koliesko je zabezpečené pomocou výrezu v tvare profilu remeňa, a nachádza sa v hornej časti kolieska. Po nasunutí remeňa do výrezov je následne ešte poistený priskrutkovaním malej plošky ku samotnému koliesku, ktorá na remeň pôsobí pritlačnou silou. Upevnenie kolieska na konštrukciu modelu je prevedené cez ložisko, aby sa mohlo koliesko voľne pohybovať pri otáčaní servo motora. Ložisko je uložené vo výreze v strede kolieska, kde sa zadnou stenou vonkajšieho krúžka opiera o koliesko a ku modelu je pripevnené skrutkou M5, ktorá sa zapiera o zadnú stranu jeho vnútorného krúžka.



Obr. 2.18: Uloženie poistných matíc

3 BoBshield API

3.1 API pre Arduino IDE

3.1.1 Filtrácia

Pri meraniach presnosti ToF snímača pre rôzne materiály, povrchy a farby guľičiek sme mohli pozorovať šum, ktorý nám pri nameraných hodnotách vznikal. Hoci tento nežiaduci jav nie je možné celkom odstrániť, môžeme jeho dopad na namerané hodnoty výrazne znížiť pomocou implementácie jednoduchšej filtrácie nameraných hodnôt vhodnou metódou. Pre našu aplikáciu sme testovali 3 typy filtrácia a to pomocou aritmetického priemeru, aritmetického plávajúceho a váženého plávajúceho priemeru. Pri testovaní sme postupovali nasledovne. Základom bolo získanie súboru nameraných hodnôt, na ktorý budeme môcť aplikovať filtre a následne navzájom porovnať ich výstupy. Vstup je pre všetky typy filtrov identický a preto môžeme spoľahlivejšie rozhodnúť o tom, ktorý z filtrov si zvolíme. Tieto hodnoty vstupu sme získali meraním polohy guľičky pri jej posune z jednej koncovej polohy trubičky k druhej, pri minimálnom naklonení trubičky. Hodnoty boli získané a merané ešte na pôvodnej verzii zariadenia – BOBSshield R2.

Pri aritmetickom priemere sme z určitého počtu nameraných hodnôt urobili aritmetický priemer a výstupom bola 1 hodnota, z čoho vyplýva, že frekvencia, s ktorou sme merali je násobne väčšia ako frekvencia, získavania filtrovaných hodnôt. To môže mať za následok spomalenie systému z hľadiska rýchlosti reakcie na zmenu polohy guľičky. Vlastnosti daného typu filtráciu vieme ovplyvniť počtom nameraných hodnôt, z ktorých robíme aritmetický priemer. Zvyšovanie počtu hodnôt má za následok väčšie vyhladenie výslednej krivky, no zároveň spôsobuje určité oneskorenie, čo môže byť pozorované hlavne keď guľička rýchlo mení svoju polohu.

Rozdiel pri plávajúcom aritmetickom priemere je ten, že je založený na princípe zberu a ukladaní určitého počtu hodnôt, pričom pri každom ďalšom meraní sa nám najstaršia hodnota vymaže a pozíciu prvej hodnoty zaujme najnovšie nameraná hodnota. Z tohto súboru hodnôt sa nám pri každom meraní vypočíta aritmetický priemer, teda frekvencia merania a frekvencia získavania nameraných hodnôt je identická.

Vážený plávajúci priemer je založený na princípe je založený na rovnakom princípe práce s nameranými hodnotami ako plávajúci aritmetický priemer. Rozdielom je, že každá hodnota má inú váhu (ako vyplýva z názvu), teda má

rozdielny vplyv na výslednú hodnotu. Výsledná hodnota sa následne vypočíta podľa nasledujúceho vzorca:

$$priemer = \frac{\sum_{i=1}^N x_i * p(x_i)}{\sum_{i=1}^N p(x_i)} = \frac{x_1 * p(x_1) + x_2 p(x_2) + \dots + x_N * p(x_N)}{p(x_1) + p(x_2) + \dots + p(x_N)} \quad (3.1)$$

Vlastnosti filtrácie teda vieme ovplyvňovať jednak počtom hodnôt, z ktorých vážený priemer robíme, ale aj zmenou parametrov ovplyvňujúcich váhu jednotlivých hodnôt, teda vplyv na výsledok.

Vzorku hodnôt, na ktorej sme porovnávali vplyv jednotlivých typov filtrov sme si vykreslili do grafu v prostredí MATLAB spolu s grafmi hodnôt získaných po aplikácii filtrov. Následne sme menili parametre filtrov, či už išlo o počet hodnôt, z ktorých sme tvorili priemer alebo pri váženom priemere, o nastavenie parametrov váženia.

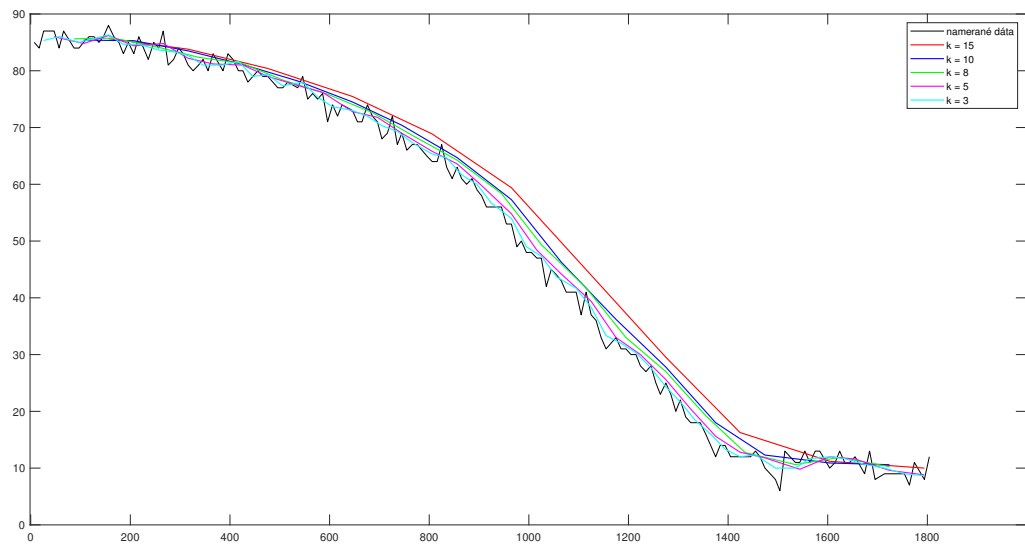
Pri procese hľadania parametrov pre vážený priemer sme uvažovali faktor zabúdania. Ide o bezrozmerné číslo z intervalu 0 až 1 a od jeho výberu závisí aké budú parametre jednotlivých váh pre dané hodnoty. Ak si zvolíme faktor zabúdania 0.9, väčšiu váhu budú mať práve staršie dáta. Naopak ak si zvolíme číslo blížiac sa k 0, dôraz bude kladený na najnovšie hodnoty. Jednotlivé koeficienty váh sa na základe faktora zabúdania získajú zo vzorca 3.2, kde λ predstavuje faktor zabúdania. Pre rýchle systémy je lepšou voľbou výber menších hodnôt faktora.

$$p_{n+1} = p_n * \lambda + 1 \quad (3.2)$$

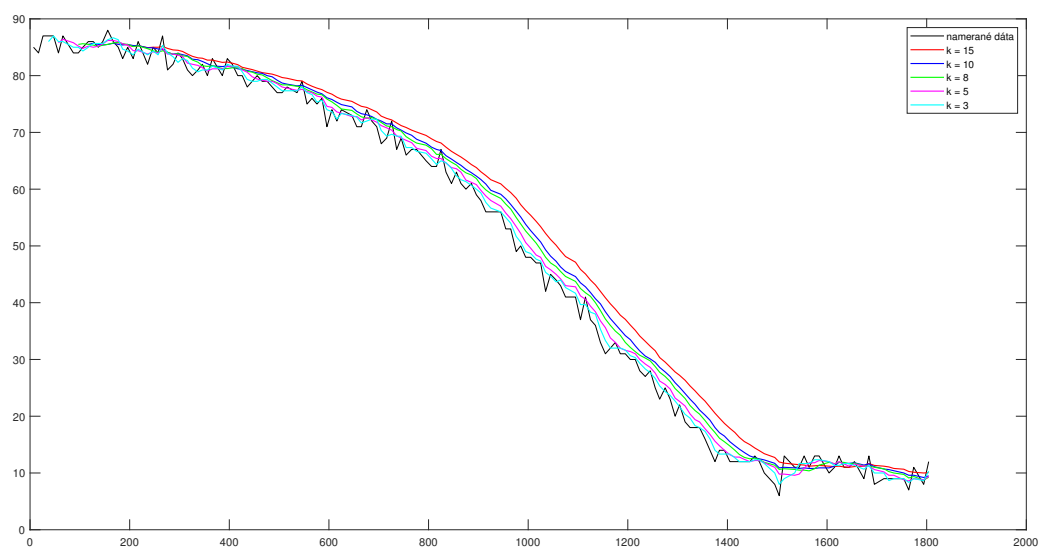
Na obrázku 3.1 môžeme vidieť graf filtrácie pomocou aritmetického priemeru pre rozdielny počet hodnôt, z ktorých sme priemer získali. Hoci vyhľadanie krivky vyzerá veľmi dobre, oproti plávajúcemu priemeru, hodnoty získavame s oveľa nižšou frekvenciou, čo vytvára oneskorenie reakcie na rýchlu zmenu polohy guľičky. Preto je určite vhodnejšou variantom jeden z typov plávajúcich priemerov.

Na ďalšom grafe (obr. 3.2) môžeme vidieť filtráciu prostredníctvom plávajúceho aritmetického priemeru. Rovnako sme testovali rôzne počty hodnôt z ktorých sme daný priemer robili. Z grafu môžeme vidieť, že pri vyšších počtoch hodnôt sa krivka výrazne lepšie vyhladí, no zároveň je na grafe očividné oneskorenie oproti nefiltrovaným dátam. Z daných možností, ktoré sme testovali sme sa rozhodli pre filtráciu pomocou 5 posledných hodnôt. Krivka dobre kopíruje nefiltrované dáta z hľadiska oneskorenia a zároveň dosahuje prijateľné vyhladenie šumu.

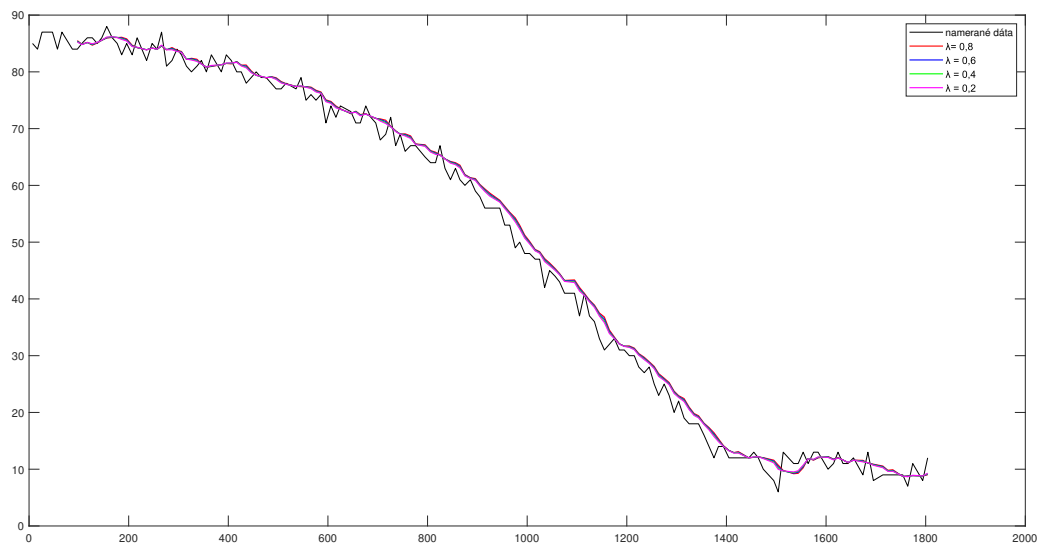
Na grafe (obr. 3.3) vidíme porovnanie filtrácie pomocou plávajúceho váženého priemeru z 5 hodnôt pre rozdielne faktory zabúdania. Môžeme vidieť, že rozdiel medzi jednotlivými krivkami je minimálny. Rozhodli sme sa zvoliť faktor zabúdania $\lambda = 0.4$, keďže potrebujeme rýchlu reakciu na zmenu aktuálnej vzdialenosti, čiže dôraz na posledné hodnoty.



Obr. 3.1: Aritmetický priemer



Obr. 3.2: Plávajúci aritmetický priemer



Obr. 3.3: Plávajúci vážený priemer

Na obrázku 3.4 môžeme vidieť porovnanie filtrácie plávajúceho aritmetického a váženého priemeru spolu s pôvodnými, nefiltrovanými nameranými hodnotami. Môžeme vidieť, že rozdiel medzi krivkami je minimálny. Jednoduchšou voľbou z hľadiska výpočtov je určite aritmetický priemer, čím budeme šetriť pamäť mikropočítača a výsledok bude takmer identický. Z tohto dôvodu sme sa rozhodli pre filtráciu pomocou plávajúceho aritmetického priemeru, počítaného z 5 posledných nameraných hodnôt.

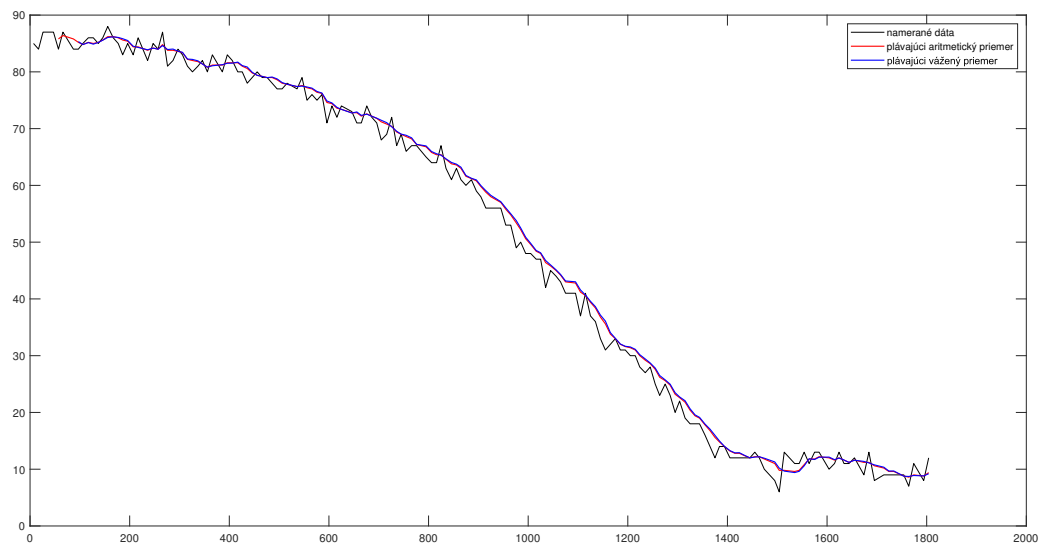
Po výbere vhodného filtra pre odstránenie šumu sme v knižnici BOBSshield.h vytvorili novú funkciu – `filterSensorRead()`. Jej úlohou je teda odstránenie šumu pri čítaní hodnôt zo snímača a tým zlepšiť kvalitu riadenia systému.

3.1.2 Knižnica

Zmeny prevedené na hardvéry sa samozrejme odrážajú aj v softvérovej časti zariadenia. Z toho dôvodu je potrebné upraviť už existujúcu knižnicu `BOBSshield.h`, vytvorenú pre jednoduchú prácu so zariadením, aby bola kompatibilná aj s našou verziou zariadenia.

Prvou vecou, ktorú sme spravili bola tvorba premennej `BOB_SHIELD_VERSION` definovanej na začiatku knižnice. Slúži pre nastavenie verzie zariadenia, ktorú práve používame. Momentálne v nej môžu byť nastavené 2 hodnoty a to: 2 pre verziu R2 a 3 pre verziu R3. Túto premennú využívame pri funkciách, ktorých parametre sa pre jednotlivé verzie zariadenia líšia ako je napríklad `calibration` alebo `actuatorWrite`. Jedná sa teda hlavne o funkcie, v ktorých sa nachádza nastavenie rozsahu pohybu servomotora.

Pri funkcii `actuatorWrite`, bolo potrebné zmeniť rozsah pohybu servomotora a tiež prevod hodnôt zadávaných do funkcie na hodnoty, ktoré budú vstupovať do servomotora pri jeho riadení. Pri pôvodnej verzii išlo a jednodu-



Obr. 3.4: Porovnanie plávajúceho aritmetického a váženeého priemeru

chý prevod za použitia funkcie `map`, kedy sa vstupné hodnoty z rozsahu -30 až 30 menili na hodnoty 65 až 125, pričom išlo o prevod celých čísel na celé čísla.

Keďže naším cieľom bolo zlepšiť presnosť otáčania trubičky a teda zvýšiť počet hodnôt, ktoré môže nadobúdať, rozhodli sme sa pre implementáciu prevodu, v ktorom 1° na servomotore sa rovná $0,2^\circ$ na trubičke. Z tohto dôvodu nie je možné využiť funkciu `map`, keďže už ide o prevod desatinných čísel na celé. Pre prevod sme sa rozhodli použiť funkciu `mapFloat` – naprogramovanú v knižnici `AutomationShield`, ktorá prevádza hodnoty desatinných čísel opäť na desatinné čísla. Teda z premennej `degree`, ktorá slúži na posielanie hodnôt na servomotor, sa nám stane desatinné číslo. Servomotor však dokáže pracovať len s celými číslami a preto je potrebné použiť funkciu `round()`, ktorá nám premennú `degree` zaokrúhli na celé číslo, ktoré už následne môžeme použiť pri riadení servomotora. Za hodnoty vstupu sme si zvolili rozsah -17 až 17 a za výstupný rozsah uhlov na servomotore 10 až 170.

Funkcia `calibration` funguje na princípe hľadania krajných hodnôt vzdialeností guľičky v trubičke pri jej pohybe. Najprv hľadá minimálnu hodnotu. Nastaví trubičku do krajnej hodnoty rozsahu jej otáčania a zmeria 100 hodnôt, z ktorých následne urobí aritmetický priemer a ten vloží do premennej `minCalibrated`. To isté sa deje aj pri hľadaní maximálnej hodnoty, ktorú vloží do premennej `maxCalibrated`. Jediné čím sa funkcia líši vo verzii R3 od pôvodnej verzie sú krajné hodnoty otočenia trubičky, tie bolo teda potrebné upraviť pre náš hardvér.

3.2 API pre Simulink

Simulink je grafické programovacie prostredie založené na MATLABe a určené pre modelovanie, simuláciu a analýzu dynamických systémov. Jeho hlavným rozhraním je nástrojom pre vytváranie grafických blokových diagramov a sada blokových knižníc, určených pre rôzne typy aplikácií. Ide o nástroj často využívaný pri automatickom riadení a využíva sa aj pri digitálnom spracovaní signálov. Táto časť práce sa venuje API vytvorenému v tomto prostredí a určenému pre riadenie nášho zariadenia.

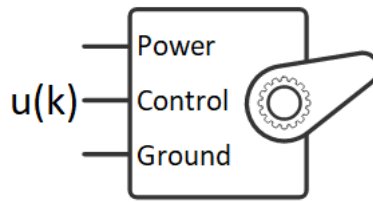
3.2.1 Knižnica

Pre tvorbu rozhrania pre programovanie nášho zariadenia v prostredí Simulink je potrebné vytvoriť vlastnú knižnicu, ktorá obsahuje potrebné funkcie pre komunikáciu so zariadením. Táto sa ale svojou formou líši od knižnice vytvorenej pre prostredie Arduino IDE. V porovnaní s ňou neobsahuje kód ako taký ale skladá sa z jednotlivých blokov, ktorých úlohou je zastrešiť určitú funkciu a následným spájaním blokov do diagramu vytvárame výsledný program, ktorý je následne pri kompilácii nahraný do mikropočítača. Knižnica je teda klasický súbor pre Simulink model (`.slx`) obsahujúci dané bloky a do zoznamu knižníc v programovacom prostredí je pridaný prostredníctvom MATLAB kódu. Aby sme knižnicu vedeli dostať je potrebné spustiť súbor s názvom `InstallMatlabAndSimulink` nachádzajúci sa v knižnici `AutomationShield`. Jeho úlohou je nastaviť všetky cesty tak aby program vedel nájsť potrebné súbory pre implementáciu knižnice. Knižnica `BOBShield` pre Simulink už bola rozpracovaná. V našej práci sme ju sfunkčnili a doplnili o potrebné funkcie. Následne sme ju vložili medzi ostatné knižnice nachádzajúce sa v `AutomationShield` API pre Simulink.

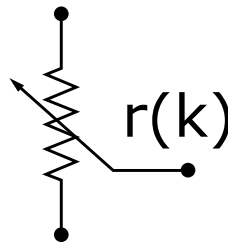
Jednotlivé bloky nachádzajúce sa v `BOBShield` knižnici teda slúžia rovnako ako funkcie naprogramované v knižnici pre Arduino IDE a zastrešujú rovnaké úlohy.

Prvým blokom je `Actuator Write`, ktorého úlohou je riadenie servomotora. Spracúva vstupnú hodnotu vchádzajúcu do bloku a následne nastavuje uhol otočenia servomotora v rozsahu, ktorý sme preň určili. Pri nastavovaní parametrov bloku si môžeme zvoliť typ vstupu, vchádzajúceho do bloku a to z 2 možností – v percentách a vo vzdialenosti reprezentovanej v mm. Dôležitým parametrom je taktiež rozsah uhlov, ktorý pre blok nastavíme. Môžeme buď nechať defaultne prednastavené hodnoty v rozsahu -30° až $+30^{\circ}$ alebo zvoliť možnosť manuálneho nastavenia vlastného rozsahu kde si zvolíme minimálny a maximálny uhol otočenia.

Blok `Reference Read` nám v diagrame reprezentuje potenciometer umiestnený na `BOBShielde`. Slúži na získavanie hodnôt z potenciometra, ktorý môže byť využitý ako referenčná hodnota pri riadení systému. Po kliknutí pravým tlačidlom myši na daný blok v schéme si môže užívateľ nastaviť spôsob akým budú hodnoty z potenciometra prezentované v Simulinku. Je možné zvoliť si z 3 možností, ktoré môžeme vidieť v sekcii `Readout type`. Prvou je získanie



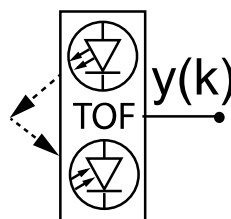
Obr. 3.5: blok Actuator Write



Obr. 3.6: blok Reference Read

hodnoty v percentách, kde hodnoty, ktoré získavame, budú z intervalu od 0 po 100 v závislosti od polohy bežca potenciometra. Ďalšou možnosťou je získavanie hodnôt v podobe napätia v rozsahu od 0 po 5 V. Poslednou dostupnou možnosťou nastavenia čítania hodnôt z potenciometra je analógová hodnota z intervalu od 0 po 1023. V závislosti od potreby užívateľa blok poskytuje teda určitú flexibilitu v závislosti od aplikácie v ktorej ho chceme využiť.

Ďalším blokom je Sensor Read a slúži nám na získavanie aktuálnej polohy guľičky v trubičke. Ide o blok slúžiaci na komunikáciu mikropočítača s naším ToF snímačom VL6180X prostredníctvom I2C protokolu. Používateľ si môže vybrať z typov výstupných hodnôt zo snímača. Prvou možnosťou je vzdialenosť v mm, ktorá nám dáva vzdialenosť guľičky od snímača v trubičke. Ďalšou z možností je pozícia guľičky v mm. Ide o úpravu vzdialenosti, kde sa za počiatočnú pozíciu považuje koncový bod trubičky v ktorom sa nenachádza ToF snímač, teda výstupné hodnoty sú inverzné voči vzdialenosti. Pri voľbe výstupu v percentách sú potrebné kalibračné hodnoty aby vedel blok učiť minimálnu a maximálnu meranú hodnotu pre daný typ telesa v trubičke. Tie je možné zadať nižšie v okne parametrov bloku kedy si môžeme zvoliť buď defaultne nastavené hodnoty – 0 až 100 mm alebo nastaviť nami namerané krajné hodnoty pomocou manuálneho zadávania.

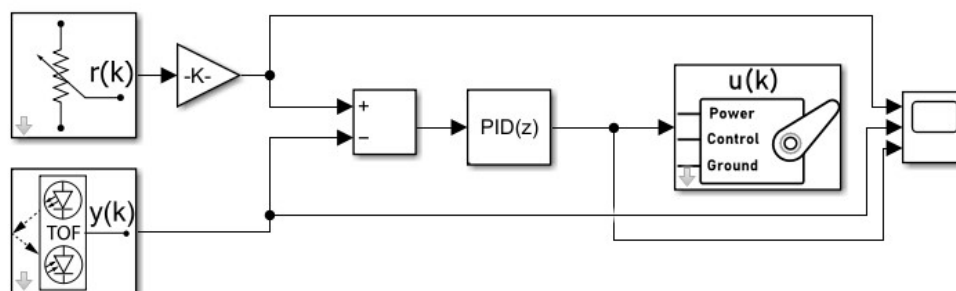


Obr. 3.7: blok Sensor Read

Posledným blokom je blok Filtrated Sensor Read (obr.), ktorého úlohou je vyhladenie šumu snímača pri meraní vzdialenosti guľičky. Ide o implementácia filtrácie signálu, ktorú sme si zvolili na základe testovaní viacerých typov filtrácií na nami nameranej zložke dát. Na vyhladenie šumu používame plávajúci aritmetický priemer počítaní z posledných piatich nameraných hodnôt. Daný blok sme vytvorili ako subsystém, do ktorého sme vložili blok Sensor Read a výstup z neho sme vložili do bloku Moving Avere, v ktorom sme si za veľkosť okna (parameter window length) zvolili 5 hodnôt a jeho výstup sme zaviedli na výstup zo subsystému.

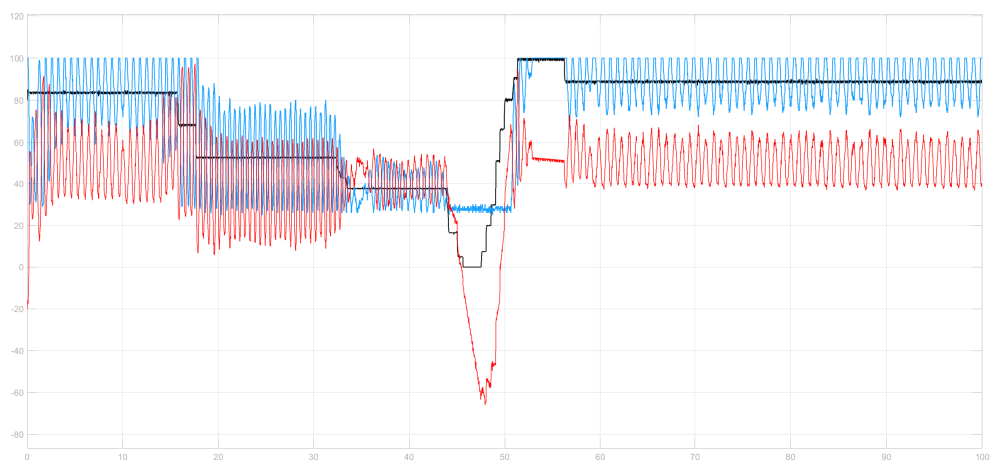
3.2.2 PID príklad

Riadenie systému pomocou PID regulátora je jeden z najznámejších typov riadenia systému. Naše zariadenie sme sa teda rozhodli riadiť pomocou PID regulátora v uzavretej riadiacej slučke (obr.3.8). V princípe sa v schéme nachádza slučka so spätnou väzbou zo zariadenia, ktorá porovnáva aktuálny stav systému s referenčnou hodnotu a na základe toho vyhodnocuje akčný zásah do systému. V našom zapojení referenčnú hodnotu dostávame z bloku Reference Read, ktorý nám ponúka aktívne ju meniť pomocou otáčania bežcom na potenciometri. Spätnú väzbu o stave zariadenia sme získavame prostredníctvom bloku Filtered Sensor Read, ktorý nám hovorí o aktuálnej pozícii guľičky v zariadení. Blok PID nám následne na základe vypočítanej chyby určí akčný zásah do systému a pošle ho do bloku Actuator Write, ktorý následne nastaví pozíciu servomotora do požadovanej hodnoty.



Obr. 3.8: Schéma zapojenia - príklad PID riadenia

Na obrázku 3.9 môžeme vidieť výstup z priebehu riadenia, kedy krivka čiernej farby reprezentuje referenčnú hodnotu, modrá aktuálnu pozíciu guľičky a červená akčný zásah do systému. Môžeme vidieť, že zariadenie na zmenu referenčnej hodnoty aktívne reaguje a snaží sa pomocou zásahu do systému regulovať polohu guľičky v danej polohe. Guľička zatiaľ len osciluje okolo referenčnej polohy ale po dôkladnom naladení by mal byť systém schopný udržať jej pozíciu priamo na požadovanej hladine. Momentálne z hľadiska problémov pri komunikácii hardvéru s rozhraním Simulink sme neboli schopný aplikovať riadny a detailný proces ladenia PID regulátora. No môžeme vidieť, že daný



Obr. 3.9: Výstup z priebehu riadenia

proces PID riadenia pomocou programu Simulink je možné uskutočniť a knižnica je plne funkčná.

4 Záver

Cieľom tejto bakalárskej práce bolo zrevidovať existujúci hardvér, navrhnuť a realizovať jeho zmeny, vytvoriť API v programovacom prostredí Simulink a zrealizovať príklady riadenia systému na novom hardvéry a v novom programovacom prostredí.

Podarilo sa nám výrazne vylepšiť hardvérovú časť zariadenie, kde sme pomocou aplikácie prevodu medzi servomotorom a trubičkou zvýšili presnosť jej otáčania z 1° na $0,2^\circ$. Tiež sme dôkladnou analýzou výberu vhodnej guličky pre náš systém znížili smerodajnú odchýlku pri meraní snímača. Z hľadiska softvéru sme pridali do pôvodnej knižnice v Arduino IDE, funkcie na filtráciu šumu a upravili sme pôvodné funkcie aby boli kompatibilné s obidvoma verziami zariadenia R2 a R3. Vytvorili sme API pre Simulink a overili jeho funkčnosť na príklade PID riadenia systému.

Taktiež sme sa snažili o tvorbu API v programovacom prostredí MATLAB, ku ktorému sme vytvorili knižnicu, no narazili sme na problém v komunikácii medzi programovacím prostredím MATLAB a ToF snímačom, ktorý sme v našom zariadení použili. Tento problém mohol vzniknúť

Do budúcnosti zariadenie stále ponúka priestor na zlepšenie. Môže dôjsť ku tvorbe nových príkladov LQ riadenia pre obe API, taktiež ku procesu identifikácie systému na novom hardvéry. Taktiež ku tvorbe funkcie na nastavenie vodorovnej polohy trubičky pri kalibrácii systému, ktorá nemohla byť vytvorená z dôvodu nedodanie komponentov na jej realizáciu.

Literatúra

- [1] InvenSense Inc. Mpu-6000 and mpu-6050 product specification revision 3.4. Online., 2013. 21.4.2022,<https://invensense.tdk.com/wp-content/uploads/2015/02/MPU-6000-Datasheet1.pdf>.
- [2] STMicroelectronics. VL6180x - proximity and ambient light sensing module. Online., 2016. 21.4.2022,<https://invensense.tdk.com/wp-content/uploads/2015/02/MPU-6000-Datasheet1.pdf>.
- [3] STMicroelectronics. ST732 datasheet. Online., 2021. 28.4.2022,<https://www.mouser.sk/ProductDetail/STMicroelectronics/ST732M28R?qs=MLItCLRbWsxtXiiWQqtIMw%3D%3D>.

Literatúra

- [1] **STMicroelectronics**, *VL6180X - Proximity and ambient light sensing module* [online], 2016, [cit. 2022-4-21] Dostupné z: <https://www.st.com/resource/en/datasheet/vl6180x.pdf>
- [2] **InvenSense Inc.**, *MPU-6000 and MPU-6050 Product Specification Revision 3.4* [online], 2013 [cit. 2022-4-21] Dostupné z: <https://invensense.tdk.com/wp-content/uploads/2015/02/MPU-6000-Datasheet1.pdf>
- [3] **STMicroelectronics**, *ST732 Datasheet* [online], 2021 [cit. 2022-4-28] Dostupné z: <https://www.mouser.sk/ProductDetail/STMicroelectronics/ST732M28R?qs=MLItCLRbWsxtXiiWQqtIMw%3D%3D>
- [4] **A. Vargová**, *BoBShield: a miniature "ball on beam" experiment*, Slovenská technická univerzita v Bratislave, Strojnícka fakulta, 2021, Bratislava, Slovenská republika
- [5] **CTMS - Control tutorials for MATLAB and SIMULINK**, *Ball and Beam: System Modeling* [online], [cit. 2022-5-12] Dostupné z: <https://github.com/gergelytakacs/AutomationShield/wiki/BoBShield>
- [6] **G. Takacs**, *BOBShield* [online], [cit. 2022-4-22] Dostupné z: https://www.hennlich.sk/fileadmin/user_upload/sk_gufer%C3%A1_-_preh%C4%BEdad_profilov.pdf
- [7] **ISL Products International Ltd.**, *Servo Motor Fundamentals* [online]. [cit. 2022-5-20] Dostupné z: <https://islproducts.com/design-note/servo-motor-fundamentals/>
- [8] **DFROBOT**, *9g Metal Gear Micro Servo (1,8Kg)* [online]. [cit. 2022-4-22] Dostupné z: <https://www.dfrobot.com/product-1338.html>
- [9] **SAVOX-SERVO**, *Savox - Servo - SH-0253 - Digital - DC - Motor* [online]. [cit. 2022-5-22] Dostupné z: <https://www.savox-servo.com/Servos-c-1338/Brushed-Motor-c-1340/Savox-Servo-SH-0253-Digital-DC-Motor/>
- [10] **MOUSER Electronics**, *MPU-6050* [online]. [cit. 2022-5-12] Dostupné z: <https://www.mouser.sk/ProductDetail/TDK-InvenSense/MPU-6050?qs=u4fy%2FsgLU9014B5JgyQFvg%3D%3D>

- [11] **MathWorks**, *Sliding Window Method and Exponential Weighting Method* [online]. [cit. 2022-5-22] Dostupné z: <https://www.mathworks.com/help/dsp/ug/sliding-window-method-and-exponential-weighting-method.html>