



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

ÚSTAV AUTOMATIZACE A MĚŘICÍ TECHNIKY

DEPARTMENT OF CONTROL AND INSTRUMENTATION

SIMULÁTOR KOPU

KICK GESTURE SIMULATOR

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

Dušan Horák

VEDOUCÍ PRÁCE

SUPERVISOR

doc. Ing. Zdeněk Bradáč, Ph.D.

BRNO 2023

Bakalářská práce

bakalářský studijní program **Automatizační a měřicí technika**

Ústav automatizace a měřicí techniky

Student: Dušan Horák

ID: 230076

Ročník: 3

Akademický rok: 2022/23

NÁZEV TÉMATU:

Simulátor kopu

POKYNY PRO VYPRACOVÁNÍ:

Cílem práce je navrhnout a realizovat systém pro validaci a testování modulu pro bezkontaktní otevírání víka kufru automobilu vyvíjené ve společnosti ALPS Electric.

1. Seznamte se s požadavky zadavatele. Provedte literární rešerši/internetový průzkum.
2. Definujte požadavky na vyvíjený systém a navrhňte jeho koncepci.
3. Na základě navržené koncepce vytvořte mechanický a elektrický návrh řešení.
4. Realizujte řešení.
5. Navrhňte koncepci řídicího software včetně ovládacího rozhraní.
6. Implementujte SW.
7. Vytvořte dokumentaci a zhodnoťte dosažené výsledky.

DOPORUČENÁ LITERATURA:

Interní dokumentace zadavatele.

Termín zadání: 6.2.2023

Termín odevzdání: 22.5.2023

Vedoucí práce: doc. Ing. Zdeněk Bradáč, Ph.D.

doc. Ing. Václav Jirsík, CSc.
předseda rady studijního programu

UPOZORNĚNÍ:

Autor bakalářské práce nesmí při vytváření bakalářské práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č.40/2009 Sb.

ABSTRAKT

Cílem práce je vytvořit simulátor kopu, který bude sloužit pro testování snímačů pro bezdotykové otevírání zavazadlového prostoru. V práci jsou popsána gesta, která snímače kopu detekují. Pro simulátor je vybrán přímý kop. Je provedena analýza přímého kopu, na základě které je simulátor navržen a sestaven. Je popsána implementace řídicího softwaru umožňujícího nastavení parametrů kopu. Simulátor může být ovládán z počítače buď pomocí konzolové aplikace, nebo pomocí aplikačního rozhraní z cizí aplikace. Hotový simulátor je otestován a jsou zhodnoceny dosažené výsledky.

KLÍČOVÁ SLOVA

Simulátor, Přímý kop, Detektor kopu

ABSTRACT

Aim of bachelor's theme is to build kick gesture simulator, which will be used to test sensors for contactless opening of the car boot lid. Thesis describes the gestures, that the kick sensors detect. A direct (regular) kick is selected for the simulator. An analysis of the direct kick is performed. Based on the analysis, the simulator is designed and built. The implementation of the control software is described. The simulator can be controlled from a computer either through a console application or through an application interface from a third-party application. Finished simulator is tested and the results are evaluated.

KEYWORDS

Simulator, Regular kick, Kick sensor

HORÁK, Dušan. *Simulátor kopu*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav automatizace a měřicí techniky, 2023, 53 s. Bakalářská práce. Vedoucí práce: doc. Ing. Zdeňek Bradáč, Ph.D.

Prohlášení autora o původnosti díla

Jméno a příjmení autora: Dušan Horák
VUT ID autora: 230076
Typ práce: Bakalářská práce
Akademický rok: 2022/23
Téma závěrečné práce: Simulátor kopu

Prohlašuji, že svou závěrečnou práci jsem vypracoval samostatně pod vedením vedoucí/ho závěrečné práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené závěrečné práce dále prohlašuji, že v souvislosti s vytvořením této závěrečné práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a/nebo majetkových a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), ve znění pozdějších předpisů, včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

Brno

.....

podpis autora*

*Autor podepisuje pouze v tištěné verzi.

PODĚKOVÁNÍ

Rád bych poděkoval vedoucímu bakalářské práce panu doc.Ing. Zdeňku Bradáčovi, Ph.D. a kolegům ze společnosti Alps Alpine za odborné vedení, konzultace, trpělivost a podnětné návrhy k práci.

Obsah

Úvod	11
1 Úvod do problematiky	12
1.1 Snímače	12
1.1.1 Kapacitní snímače	12
1.1.2 Radarové snímače	13
1.1.3 Ultrazvukové snímače	13
1.2 Typy kopů	14
1.2.1 Definované Alps Alpine	14
1.2.2 Definované automobilkami	16
1.2.3 Shrnutí gest	16
2 Návrh koncepce	17
2.1 Analýza přímého kopu	17
2.1.1 Sběr dat	17
2.1.2 Zpracování dat	18
2.1.3 Rozdělení kopů	19
2.1.4 Trajektorie	20
2.1.5 Rychlost a zrychlení	21
2.2 Návrh konstrukce	21
2.2.1 Varianta A	22
2.2.2 Varianta B	22
2.2.3 Výběr konstrukce	22
2.3 Výběr motoru	23
2.3.1 Krokový motor s driverem	23
2.3.2 Modelářský servomotor	24
2.3.3 Vlastní řešení	24
2.3.4 Zvolený motor	24
2.4 Elektrický návrh	24
2.5 Topologie programu	25
3 Realizace	27
3.1 Konstrukce	27
3.2 Elektrické zapojení	30
3.3 Propojení driveru s počítačem	32
3.3.1 Fyzická vrstva	32
3.3.2 Zapojení převodníku	32

3.3.3	Komunikační protokol	33
3.3.4	Průběh kopu	36
3.4	Řídicí software	36
3.4.1	Využití v cizích aplikacích	36
3.4.2	Konzolová aplikace	39
4	Zhodnocení simulovaných kopů	40
4.1	Porovnání simulovaných kopů s lidskými	40
4.2	Opakovatelnost	41
	Literatura	45
	Seznam příloh	48
A	Analýza kopů	49
A.1	Lidské kopy	49
A.2	Kopy simulátoru	49
B	Soupis materiálu pro stavbu konstrukce	50
B.1	Rám simulátoru	50
B.2	Ostatní komponenty	50
C	Elektrické zapojení	51
D	Obsah elektronické přílohy	53

Seznam obrázků

1.1	Kapacitní snímač kopy[4]	13
1.2	Radarový snímač kopy [2]	13
1.3	Přímý kop	14
1.4	Kop v koleni	15
1.5	Vertikální kop	15
1.6	Pokyn pro otevření zavazadlového prostoru u Škody Superb [9]	16
2.1	Umístění stativu při sběru dat	17
2.2	Ukázka práce v programu Tracker	18
2.3	Průběhy rychlého kopy	19
2.4	Průběhy středního kopy	19
2.5	Průběhy pomalého kopy	20
2.6	Naměřená data proložená kružnicí	21
2.7	Vizualizace konstrukce A	22
2.8	Vizualizace konstrukce B	23
2.9	Blokové schéma elektrického zapojení	25
2.10	Topologie programu	26
3.1	První verze konstrukce	27
3.2	Vizualizace finální verze simulátoru	28
3.3	Řez dílem spojující rameno s odražečem	28
3.4	Detail čela motoru se snímači	29
3.5	Dokončená finální konstrukce	30
3.6	Schéma zapojení simulátoru	31
3.7	Zadní pohled na simulátor s přívodním konektorem	31
3.8	Schéma propojení převodníku s driverem	33
4.1	Porovnání trajektorie simulátoru s lidskými kopy	40
4.2	Porovnání střeních kopů lidí s kopy simulátoru	41
C.1	Celkové elektrické schéma simulátoru	52

Seznam tabulek

3.1	Ukázka příkazu pro nastavení rychlosti	33
3.2	Příkazy pro roztočení motoru	34
3.3	Příkazy pro nastavení parametrů driveru	35
3.4	Souhrn nastavitelných parametrů konzolové aplikace	39
4.1	Srovnání kvality opakovatelnosti jednotlivých kopů	42
A.1	Přehled testovacích osob	49
A.2	Parametry kopů použitých pro testy simulátoru	49
B.1	Soupis hliníkových profilů 30x30	50
B.2	Soupis ostatního materiálu potřebného pro stavbu rámu	50
B.3	Zbylé mechanické komponenty	50
C.1	Soupis elektronických komponentů	51

Úvod

Cílem práce je vytvořit simulátor kopů, sloužících pro bezdotykové otevírání zavazadlového prostoru u aut. Funkce bezdotykového otevírání využívá snímače umístěného pod zadním nárazníkem automobilu. Pokud snímač zaznamená některé z definovaných gest pošle do řídicí jednotky signál k otevření zavazadlového prostoru.

Simulátor je vytvářen pro společnost Alps Alpine, sídlící v Sebranicích. Alps Alpine se zabývá vývojem radarových snímačů kopy sloužících pro otevření zavazadlového prostoru. Pro potřeby vývoje se funkce senzoru pravidelně testuje sérií více kopů.

Zařízení bude sloužit k opakovatelnému testování těchto snímačů, které je s lidskými testery obtížně dosažitelné. Zařízení by mělo co nejvěrohodněji napodobovat lidský kop, sloužící pro otevření zavazadlového prostoru. Parametry kopy musí být nastavitelné individuálně pro každý kop, nebo pro danou větší sérii. Simulátor by také měl být snadno přenositelný, aby bylo možné provádět testy i mimo budovu.

Práce se zabývá gesty používanými pro otevření zavazadlového prostoru. Poté je proveden vlastní rozbor přímých kopů a na jeho základě definovány potřebné parametry simulátoru. Podle těchto parametrů jsem postupoval při návrhu koncepce. V rámci koncepce byl vytvořen návrh konstrukce simulátoru a vybrán vhodný motor. Dále je definována základní topologie ovládacího softwaru.

Koncepce konstrukce je rozpracována do konečného modelu, a na jeho základě je postavena z hliníkových profilů. Na základě výsledků testů jsou pak na konstrukci provedeny další úpravy pro vylepšení funkčnosti simulátoru. Motor je ovládán pomocí driveru, který komunikuje prostřednictvím RS232 s počítačem. Pro jeho ovládní je vytvořen program, ten je možno implementovat jako třídu do cizího softwaru, nebo lze volat z konzole.

Je provedeno validační měření kopů simulátoru a jsou zhodnoceny jeho výsledky.

1 Úvod do problematiky

Simulátor bude sloužit k testování snímačů kopu, kterých se využívá pro otevření zavazadlového prostoru. Cílem je vytvořit zařízení, které by provádělo skupinu referenčních kopů, aby mohl být během testování senzoru za různých podmínek omezen vliv rozdílných pohybů.

Funkce otevírání kufru slouží uživateli pokud má plné ruce a potřebuje otevřít zavazadlový prostor. Uživatel musí mít u sebe klíče od vozidla, dojít k zadnímu nárazníku a kopnout v daných místech pod nárazníkem, tím dojde k otevření zavazadlového prostoru. [1].

Simulátor by měl komunikovat s nadřazeným řídicím programem, který určí parametry kopu a dá pokyn k jejich vykonání. Prováděné kopy musí být mnohonásobně opakovatelné. Simulátor musí být přenosný, tak aby mohl být používán i na parkovišti u reálných vozidel a v případě potřeby jej bylo možné převézt.

1.1 Snímače

Pro detekci kopů se využívá samostatných modulů, které bývají umístěny pod zadním nárazníkem automobilu. Moduly využívají kapacitních, ultrazvukových, nebo radarových snímačů.

1.1.1 Kapacitní snímače

V současnosti se jedná o nejvíce rozšířený způsob detekování kopů. Pod zadním nárazníkem je upevněn kapacitní snímač polohy, který detekuje přítomnost nohy v jeho blízkosti. Pro větší přesnost jsou většinou použity 2 kapacitní snímače, v řídicí jednotce je pak na základě dat z obou snímačů vyhodnoceno, jestli se jedná o kop nebo ne. Nevýhodou těchto senzorů je zhoršení funkčnosti pokud je nárazník špinavý například od bláta, nebo sněhu.[5] Oproti ostatním snímačům mají také menší detekční vzdálenost. Celý snímač je poměrně velký, což komplikuje jeho montáž. Kapacitní snímač je na zobrazen na obrázku 1.1.

První byly instalovány v roce 2011. Tyto senzory vyrábí například Texas Instruments pod označením TIDA-01409, hotové moduly pro automotive pak vyrábí Huf Hülsbeck, nebo Brose Fahrzeugteile[3, 4].



Obr. 1.1: Kapacitní snímač kopy[4]

1.1.2 Radarové snímače

Modul vysílá elektromagnetické vlny a detekuje jejich odrazy od předmětů za nárazníkem, získaná data modul analyzuje a vyhodnocuje jestli se jedná o kop nebo ne. Snímač se podobně jako kapacitní nachází za nárazníkem aby nedošlo k jeho mechanickému poškození. Výhodou radarových snímačů je, že dokáží detekovat kopy i pokud je nárazník špinavý od bláta, nebo sněhu. Zároveň také dokáže rozpoznávat jednotlivá gesta od sebe a detekovat více gest než kapacitní snímač. Oproti kapacitnímu je mnohem menší, díky čemuž je jednodušší jeho montáž. Může být umístěn i za metalickým krytem nárazníku. [2, 6]

Tyto senzory vyrábí například Texas Instruments, hotové moduly pak vyrábí Innosent a Alps Alpine (obrázek 1.2), pro které simulátor vyrábím.[6, 7]



Obr. 1.2: Radarový snímač kopy [2]

1.1.3 Ultrazvukové snímače

Využívají stejných principů jako radarové, pouze místo radiových vln používají ultrazvukové. Stejně jako u kapacitních je zde problém se ztrátou funkčnosti pokud se ušpiní. Jejich nevýhodou je, že na rozdíl od kapacitního a radarového snímače

nemohou být umístěny až za nárazníkem, ale musí být kvůli němu vytvořen otvor do nárazníku, ve kterém je senzor umístěn. Je zde tedy větší riziko poškození senzoru.[8]

Senzor také vyrábí Texas Instruments, hotový modul pro auta se mi žádný nepodařilo dohledat.

1.2 Typy kopů

K otevření zavazadlového prostoru nedochází při každém kopu, ale pouze při provedení jednoho z definovaných gest. Ty mohou být různé pro různé snímače, nebo automobilky. Mimo to jsou definované případy, kdy k otevření zavazadlového prostoru dojít nesmí. V této kapitole se věnuji rozboru nejběžnějších gest.

1.2.1 Definované Alps Alpine

Ve společnosti Alps Alpine jsou interní dokumentací definována gesta pro otevření zavazadlového prostoru, která musí senzor rozpoznat. Kromě nich je popsána také skupina gest při kterých nesmí dojít k detekci kopu. [2] V kapitole popisují vybraná nejběžnější gesta.

Přímý kop

Jedná se o základní a nejvíce rozšířené gesto pro otevření kufru. Uživatel stojí kolmo k nárazníku a kopne s nataženou nohou pod nárazník, noha se hýbe pouze v kyčli, jak je znázorněno na obrázku 1.3. Celý kop trvá 0,3 až 3 s. Mezi pohybem nahoru a dolů může být drobná časová prodleva (do 0.3 s).[2]



Obr. 1.3: Přímý kop

Kop v koleni

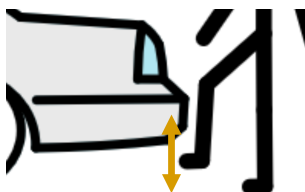
Stejně jako u předchozího gesta stojí uživatel kolmo k nárazníku, ale nohu ohýbá pouze v koleni. Ukázkou můžeme vidět na obrázku 1.4.[2]



Obr. 1.4: Kop v koleni

Vertikální kop

Uživatel stojí za autem a pohne s chodidlem kolmo nahoru až těsně pod nárazník a zpět. Jak ukazuje obrázek 1.5.[2]



Obr. 1.5: Vertikální kop

Boční kopy

Gesto je podobné jako přímý kop nebo kop v koleni, uživatel ale nestojí k nárazníku kolmo, ale pod úhlem. Kop je detekován pokud je úhel mezi nohou uživatele a nárazníkem menší nebo roven 45° . [2]

Nevalidní gesta

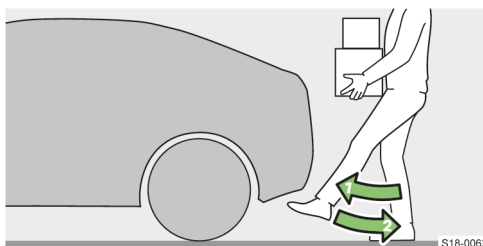
Jsou skupinou gest při kterých nesmí dojít k otevření zavazadlového prostoru. K otevření nesmí dojít, pokud osoba stojí za nárazníkem nebo v jeho blízkosti prochází. Dalším nevalidním gestem je pokud uživatel zvedá věci ležící na zemi za autem, nebo pokud dochází k nakládání, nebo vykládání věcí do zavazadlového prostoru. Za nevalidní gesto je považováno také, pokud uživatel zastaví pohyb při provádění z validních gest.[2]

Požadavkem simulátoru je ověřování detekce platných gest, proto se nevalidními gesty dále zabývat nebudu.

1.2.2 Definované automobilkami

Pro srovnání jsem se rozhodl zjistit jaká gesta detekují auta dnes dostupná na trhu. Při výběru aut jsem volil pokud možno nejnovější model dané automobilky.

Z koncernu Volkswagen jsem si zvolil vůz Škoda Superb verze 2022/06 pro český trh. V návodu k vozidlu je uvedeno, že vozidlo reaguje na přímé kopnutí podle obrázku 1.6 [9]. Téměř stejná gesta podle manuálu detekuje Volvo, Honda, Volkswagen a Chrysler [13, 12, 11, 10].



Obr. 1.6: Pokyn pro otevření zavazadlového prostoru u Škody Superb [9]

1.2.3 Shrnutí gest

Jelikož automobilky v současnosti používají hlavně přímé kopy, tak jsem se po dohodě se zástupci Alps Alpine dohodl, že simulátor bude simulovat pouze přímé kopy.

V případě provádění i dalších gest by musela být konstrukce výrazně složitější, ideální by bylo využití šestiosého robotického manipulátoru, simulátor by ale tím pádem nebyl jednoduše přenositelný a byl by výrazně dražší.

Jelikož je primárním účelem simulátoru ověření vlivu prostředí/software na snímač, bude simulátor provádět pouze jednu skupinu kopů. V práci se nebudu zabývat vedlejšími vlivy. Simulátor bude simulovat kopy na autě stojícím na rovném povrchu a kop bude prováděn kolmo proti senzoru.

Z hlediska doby trvání jsem se rozhodl rozdělit na rychlé, pomalé a normální (střední) kopy.

2 Návrh koncepce

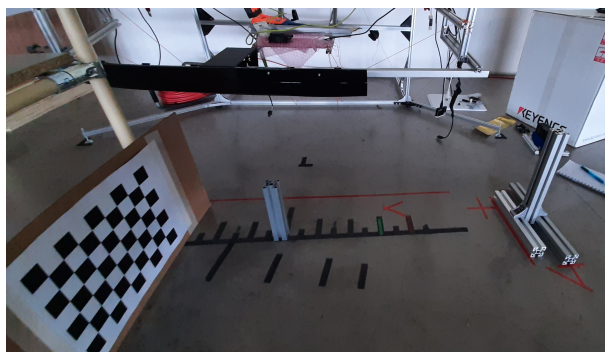
Kapitola shrnuje potřebné parametry simulátoru, na jejich základě pak jsou navrženy 2 možné varianty konstrukce. Dále jsou vybrány základní komponenty a je popsáno základní rozvržení programu.

2.1 Analýza přímého kopu

Pro postavení simulátoru je zapotřebí znát více parametrů, než je specifikováno automobilkami nebo společností Alps Alpine. Proto jsem udělal vlastní analýzu přímého kopu. Analýzu jsem vytvořil pomocí 5 testovacích osob. V příloze jsou popsány podrobnější informace o osobách, jako je výška, váha, věk. Protože jsem měl omezený počet osob, které mají přístup k pracovišti, na kterém jsem prováděl sběr dat, budou simulované kopy odpovídat primárně kopům mužů do 40 let. Díky zkušebním kopům testovacích osob jsem mohl zanalyzovat trajektorii kopů, jejich rychlost a zrychlení, určit optimální výšku a délku nohy simulátoru.

2.1.1 Sběr dat

Na botu každé z testovacích osob, jsem pro pohodlnější zpracování dat, přilepil trackovací bod a celý průběh jsem natáčel na kameru¹. U každé z osob jsem natočil 3 série po pěti kopech. V první sérii jsem osoby instruoval, aby přišly k nárazníku a kopy tak jak by chtěly otevírat zavazadlový prostor, tyto kopy považuji za střední, ve druhé sérii pak byly instruovány, aby kopy rychle a ve třetí sérii pomalu.



Obr. 2.1: Umístění stativu při sběru dat

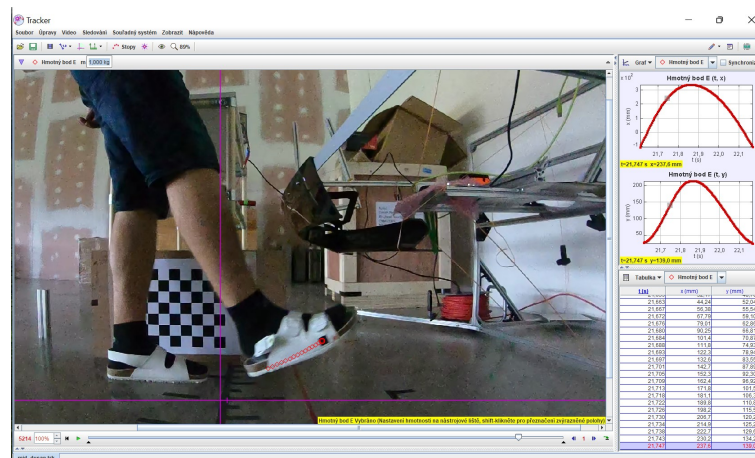
Aby bylo možné získaná data v budoucnu porovnat s kopy vytvořeného simulátoru, tak jsem pro kameru vyrobil jednoduchý stativ a stanovil přesné místo ze

¹natáčeno kamerou Gopro HERO 6 , snímkování 240 fps, res= 1080

kterého mají být kopy prováděny. Kopy byly prováděny 10 cm od nárazníku, na obrázku 2.1 můžeme v daném místě vidět hliníkový profil, který jsem používal místo kalibrační tyče pro nastavení měřítka ve videu. Stativ byl umístěn ve stejné vzdálenosti od nárazníku a 73 cm od místa kde byly kopy prováděny.

2.1.2 Zpracování dat

Nahraná videa jsem potom pomocí programu Tracker zpracoval. Program po označení trackovacího bodu prochází snímek po snímku a zaznamenává polohu bodu vzhledem ke stanovenému počátku souřadnic. Počátek jsem zvolil pro x osu na čáře nacházející se 10 cm od nárazníku a pro y osu na podlaze. Ukázka analýzy kopu je vidět na obrázku 2.2, fialový kříž určuje počátek souřadnic, červené body jsou předchozí pozice trackovacího bodu. V pravé části pak můžeme vidět časové průběhy a tabulku s naměřenými daty.



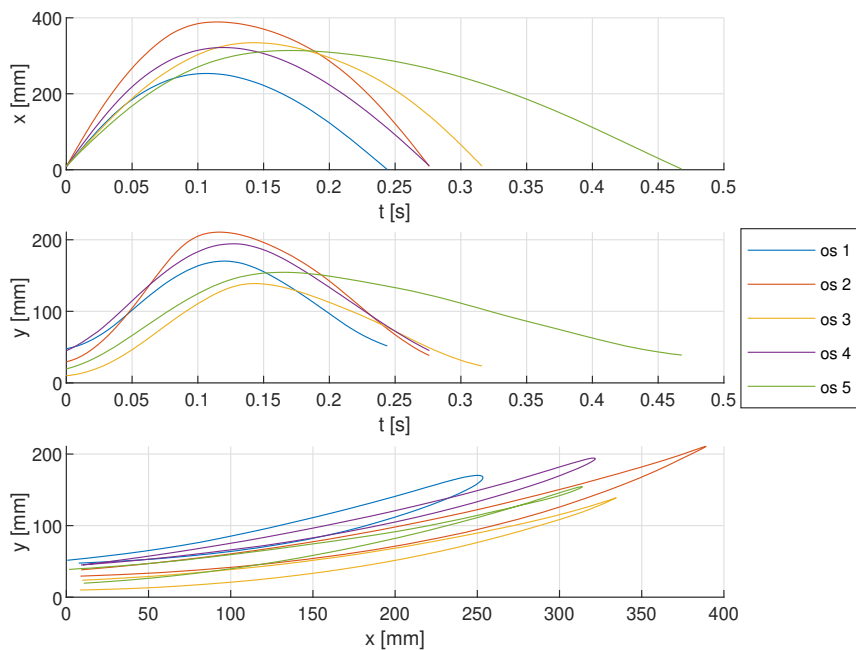
Obr. 2.2: Ukázka práce v programu Tracker

Získaná data jsem exportoval do txt souborů pro pozdější zpracování. Soubory obsahují x a y souřadnici od počátku a informaci ve kterém čase se bod na těchto souřadnicích nacházel.

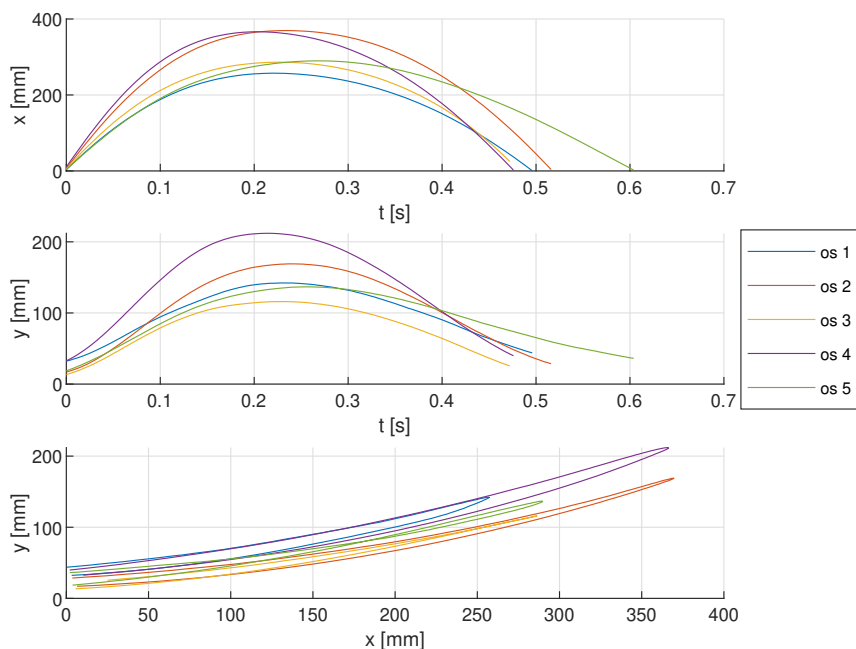
Tato data jsem pak v Matlabu dále zpracovával, kopy jsem časově posunul, aby každý kop začínal ve stejný okamžik a zobrazil je do grafů. Pro přehlednost dat jsem každou sérii stejných kopů od jedné osoby zprůměroval, získal jsem tedy sadu 3 kopů (rychlý, pomalý, střední) pro každou osobu a dál pracoval s těmito průměrnými kopy.

2.1.3 Rozdělení kopů

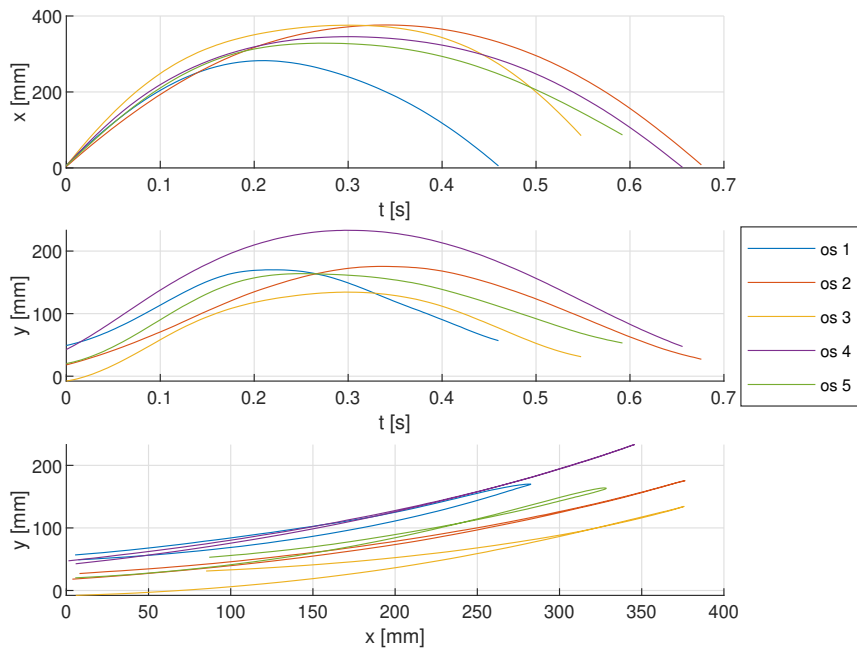
Získané průběhy jsem zobrazil do skupiny 3 grafů (obrázky 2.3 až 2.5) v prvním grafu je závislost souřadnice x na čase, ve druhém závislost y souřadnice na čase a ve třetím je zobrazena dráha kopu v x y souřadnicích.



Obr. 2.3: Průběhy rychlého kopu



Obr. 2.4: Průběhy středního kopu

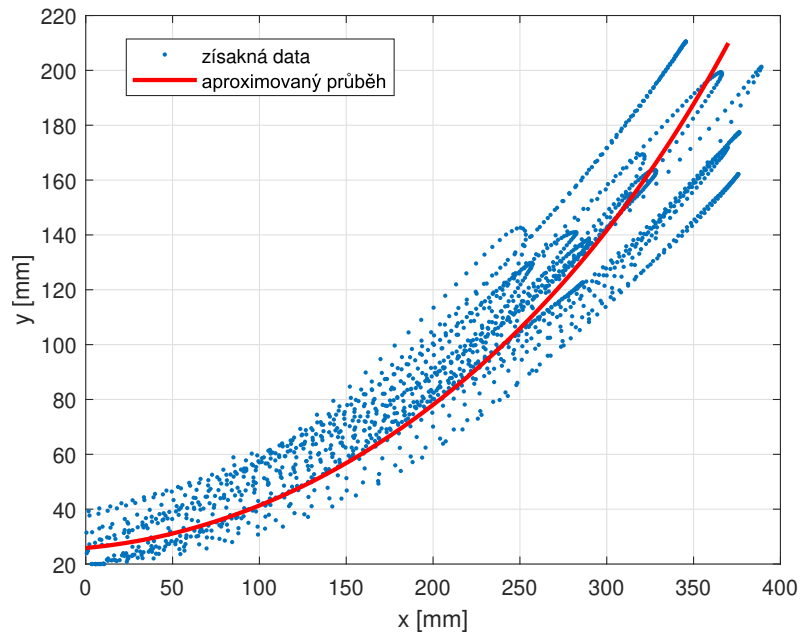


Obr. 2.5: Průběhy pomalého kopu

Jak je na grafech vidět, kopy jednotlivých osob si jsou velice podobné. Mezi kopy nelze najít žádná spojitost v závislosti na výšce či věku osoby. Větší odlišnost je vidět u osoby 1, kde její rychlé kopy odpovídají spíše středně rychlým kopům ostatních účastníků, rozhodl jsem se tedy jej přesunout do středních kroků pod označením os 1_P. Podobně jsem přesunul střední kop osoby 5 mezi pomalé (označen jako 5_S) a rychlý kop osoby 5 mezi střenění (označen jako 5_R).

2.1.4 Trajektorie

Pro konstrukci simulátoru bylo důležité zjistit, jak dlouhá musí být noha simulátoru. Data všech kopů jsem převedl do jednoho souboru a pomocí aplikace Curve fitter v Matlabu jsem je proložil rovnicí kruhu ve tvaru $y = h - \sqrt{r-x} \cdot \sqrt{r+x}$, kde h je vzdálenost osy otáčení od země a r je poloměr (obrázek 2.6). Vyšlo mi, že střed otáčení nohy by měl být přibližně 563 mm nad zemí. Pro pohodlnější další práci budu uvažovat umístění motoru 550 mm nad zemí. V případě potřeby pak bude možné motor už v hotové konstrukci přizvednut.



Obr. 2.6: Naměřená data proložená kružnicí

2.1.5 Rychlost a zrychlení

Na základě určené pozice středu otáčení jsem mohl data převést do polárních souřadnic. V polárních souřadnicích pak bylo možné pomocí diference určit maximální úhlovou rychlost kopolu $\omega = 10 \text{ rad/s} = 95 \text{ ot/min}$ a z ní následně maximální úhlové zrychlení $\omega' = 100 \text{ rad/s}^2$, vzhledem k míře zašumění dat lze tyto hodnoty brát pouze jako orientační.

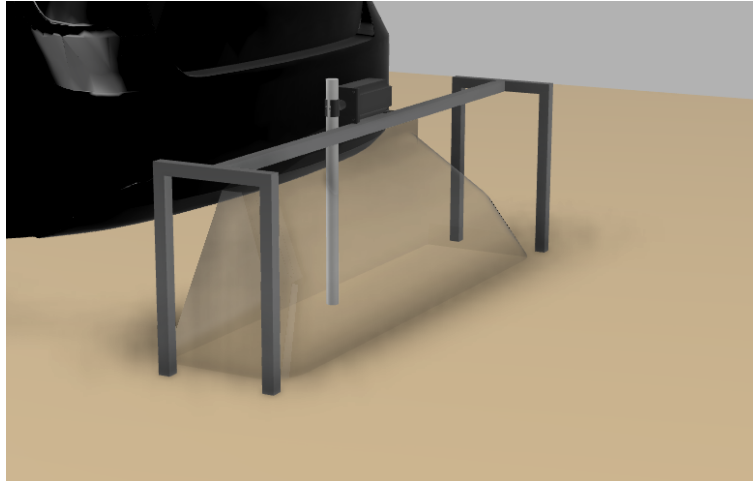
2.2 Návrh konstrukce

Při navrhování konstrukce bylo nutné zohlednit hlavně, aby větší části konstrukce nebyly v zorném poli radaru, tyto části by sice šlo překrýt absorbéry, to by ale celou konstrukci prodražilo. Z předchozí kapitoly už víme, že délka nohy musí být přibližně 50 cm. Aby byla konstrukce dostatečně stabilní a kvůli jednoduchosti montáže, jsem se rozhodl konstrukci poskládat z hliníkových profilů. Profily jsou zároveň lehké a v případě potřeby se dá celá konstrukce jednoduše upravit. Noha simulátoru by měla být cca 10 cm od nárazníku [2].

Jako náhradu nohy jsem pro první verzi, hlavně pro její nízkou hmotnost, zvolil PVC trubku na odpady o průměru 32 mm. Na jejím konci bude umístěn prvek sloužící k odrážení radarových vln.

2.2.1 Varianta A

První možností je, dát nohy konstrukce dostatečně daleko od sebe, aby nezasahovaly do prostoru radaru, jak je vidět na obrázku 2.7, kde průhledná část znázorňuje prostor snímáný radarem. Výhodou této varianty je, že se toto řešení dá upravit tak, aby byl motor s ramenem posuvný a mohl tak otestovat větší část zorného pole. Tato konstrukce je poměrně stabilní. Její nevýhodou jsou větší nároky kladené na propojovací tyč a větší rozměry.



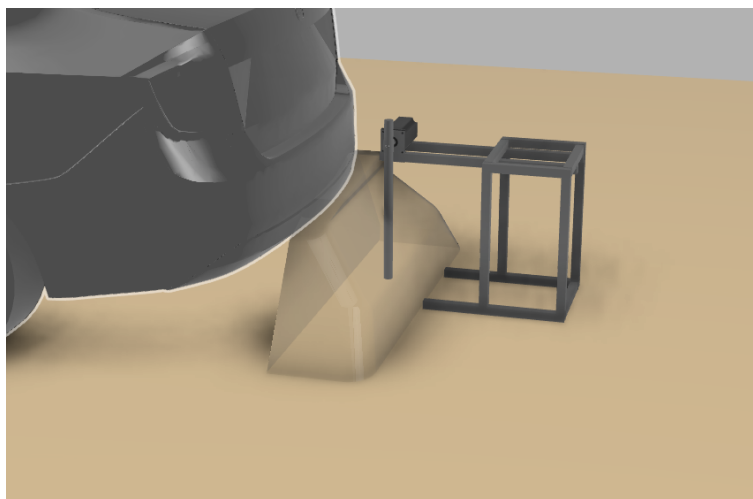
Obr. 2.7: Vizualizace konstrukce A

2.2.2 Varianta B

Druhou variantou je motor umístit na profily, které budou vysunuté mimo základnu tak, aby základna byla mimo zorné pole a zároveň noha byla v požadované vzdálenosti od nárazníku. Nevýhodou je, že může docházet k překlápění této konstrukce a možná bude nutné ji v zadní části zatížit, aby byla stabilní.

2.2.3 Výběr konstrukce

Oba návrhy potřebují přibližně stejné množství materiálu, dají se tedy předpokládat podobné náklady na realizaci. Vzhledem k tomu, že bude nutné simulátor kvůli provádění testů na více místech, přenášet a případně i převážet, tak jsem se rozhodl pro realizaci varianty B, která zabírá méně místa a půjde s ní tedy lépe manipulovat.



Obr. 2.8: Vizualizace konstrukce B

2.3 Výběr motoru

Jak je zmíněno v kapitole 2.1.5, motor by měl dosahovat maximální rychlosti alespoň 95 ot/min a zrychlení 90 rad/s^2 , a to při zátěži o hmotnosti přibližně 300 g na konci 55 cm dlouhé nohy (viz kapitola 2.1.4). Ze známých parametrů délky ramene, maximálního zrychlení a hmotnosti umístěné na konci ramene jsem pak vypočítal potřebný moment, který má motor dodat. Ten jsem vypočítal ze vzorce 2.2 získaného úpravou 2. Newtonova zákona.

$$F = m \cdot a = m \cdot \varphi \cdot x'' = m \cdot r \cdot \varphi'' \quad (2.1)$$

$$M = F \cdot r = m \cdot r^2 \cdot \varphi'' = 0,3 \cdot 0,55^2 \cdot 90 = 8,2 \text{ Nm} \quad (2.2)$$

Pro co nejlepší opakovatelnost kopů by mělo probíhat snímání aktuálního úhlu natočení. Zároveň by neměl být příliš velký, aby byla celá konstrukce pořád přenositelná. Vzhledem k tomu, že při kopu zůstává noha pod nárazníkem chvíli téměř nehybná, je tedy nutné, aby motor po zastavení držel požadovanou pozici. Dalším důležitým aspektem je také cena. Na základě těchto parametrů jsem vybral 3 možné varianty pohonu simulátoru.

2.3.1 Krokový motor s driverem

Krokové motory mají přenosné řízení podle počtu kroků a lze u nich dobře měnit rychlost. V kombinaci s driverem, který snímá polohu motoru a zajišťuje jeho chod se označují jako hybridní motory, jejich výhodou je jednoduché ovládání.[14]

Vhodnou variantou pro simulátor by byl motor Nema 34, který se prodává včetně driveru v přepočtu za 3 640 Kč. Jeho maximální moment je 12 Nm, s jedním krokem

po 1,8°, možností microsteppingu, součástí motoru už je i enkodér pro zpětnovazební řízení polohy. Driver navíc může být ovládán pomocí počítače přes RS232. Nevýhodou je neúplná dokumentace k motoru a jeho hmotnost (5 kg), která zvyšuje požadavky na konstrukci.[15]

2.3.2 Modelářský servomotor

Tyto motory většinou obsahují stejnosměrný motor na který je připojena převodovka, která zajišťuje jeho velký moment. Motor obsahuje svoji řídicí elektroniku, která ovládá motor tak, aby dosáhl požadovaného úhlu.[16]

Motor s odpovídajícími parametry stojí v přepočtu 3 700 Kč[17]. Jeho nevýhodou je, že se u něj obtížně reguluje rychlost. Má také omezený úhel natočení do 180°, což je ale pro simulátor kopu dostačující.

2.3.3 Vlastní řešení

Další možností je, použít běžný krokový motor a snímač natočení, a poté si postavit vlastní zpětnovazební řízení. Výhodou tohoto řešení je plná kontrola nad motorem. Nevýhodou je značná časová náročnost realizace.

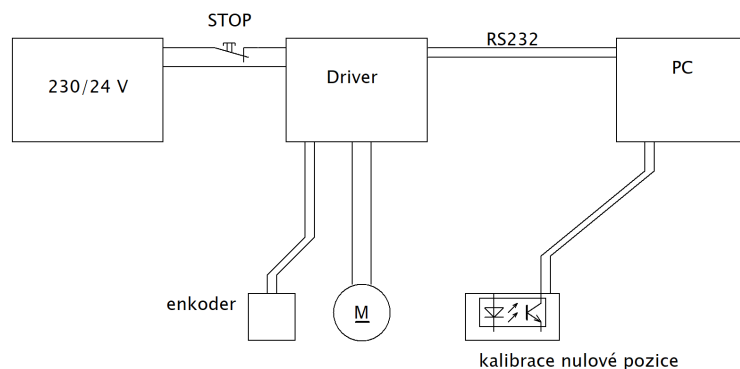
Cena vhodného motoru je přibližně 1 600 Kč, jedná se o stejný motor, který jsem zmiňoval v kapitole 2.3.1, ale bez driveru [18]. Je ale nutné počítat s dalšími náklady na snímač otáček a hardware pro jeho ovládání.

2.3.4 Zvolený motor

Vzhledem k časovým možnostem jsem se rozhodl koupit už hotové řešení, kde se bude motor sám starat o řízení na požadovaný úhel. Jelikož bude nutné během kopu měnit aktuální rychlost, rozhodl jsem se pro koupi krokového motoru s driverem. Podobných variant k vybranému motoru jsem našel více, zvolený je však levnější a má rozumnou dodací lhůtu. Výhodou také je, že v případě potřeby je možné motor zaměnit za jiný, ale ponechat stávající driver a s ním i software.

2.4 Elektrický návrh

Návrh elektrického zapojení je blokově zobrazen na obrázku 2.9, driver krokového motoru bude napájen ze stejnosměrného zdroje 24 V, do driveru je připojen krokový motor a enkodér motoru. Protože je enkodér pouze relativní, bude nutné po zapnutí simulátoru nakalibrovat nulovou pozici, proto jsem do schématu přidal ještě jeden snímač polohy. Driver bude řízen přes sériovou linku pomocí počítače.

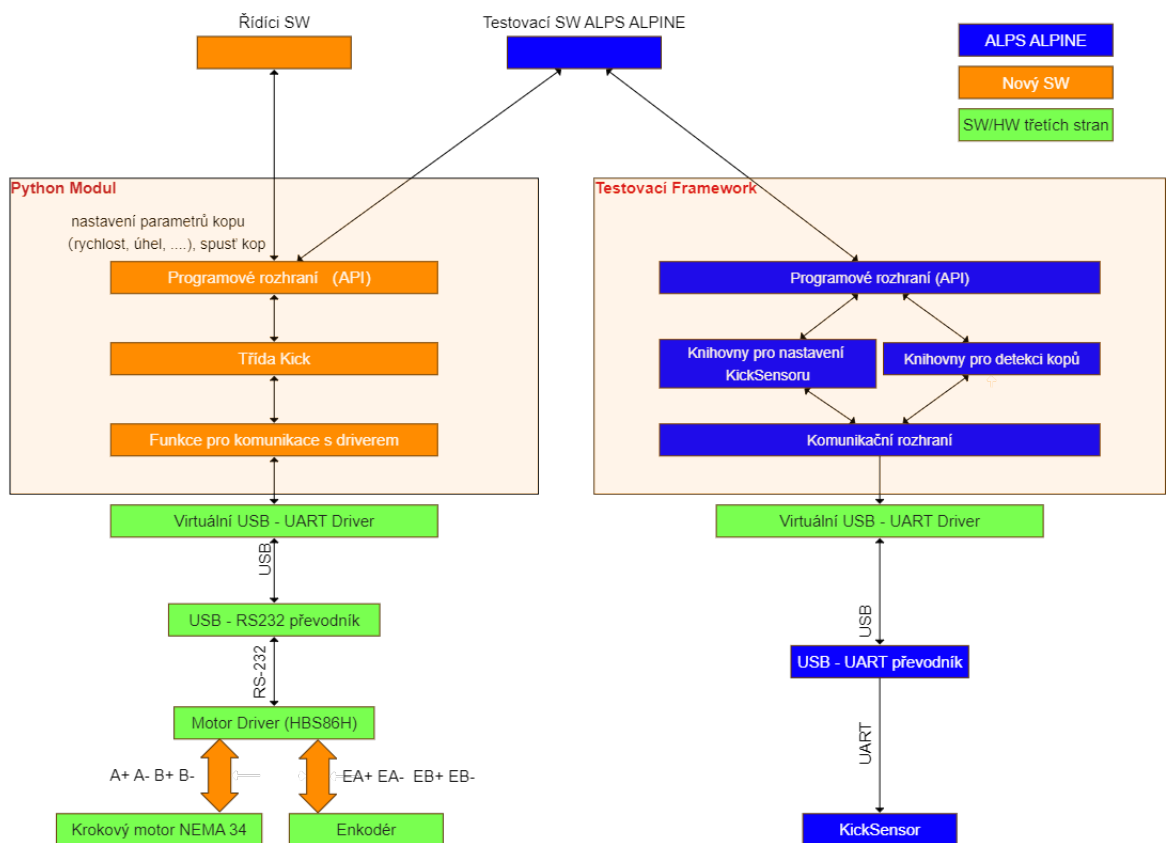


Obr. 2.9: Blokové schéma elektrického zapojení

2.5 Topologie programu

Jak je vidět na obrázku 2.10, hlavním programem je systém pro testování senzoru, ten stejně jako ostatní modře podbarvené části je již vytvořen Alps Alpine a používán pro testování senzorů pomocí lidských testerů. Program pro testování senzoru komunikuje s radarovým senzorem a zároveň bude komunikovat se mnou vytvořeným programem, ten bude do něj vložen jako třída Kick. Program na testování bude třídě předávat požadované parametry kopů, a poté zašle požadavek na vykonání kopu. Třída kick na základě těchto požadavků pak zavolá další vlastní funkce, pomocí kterých bude komunikovat s driverem krokového motoru.

Dále vytvořím můj řídicí software, aby bylo možné simulátor používat i bez testovacího softwaru Alps Alpine.



Obr. 2.10: Topologie programu

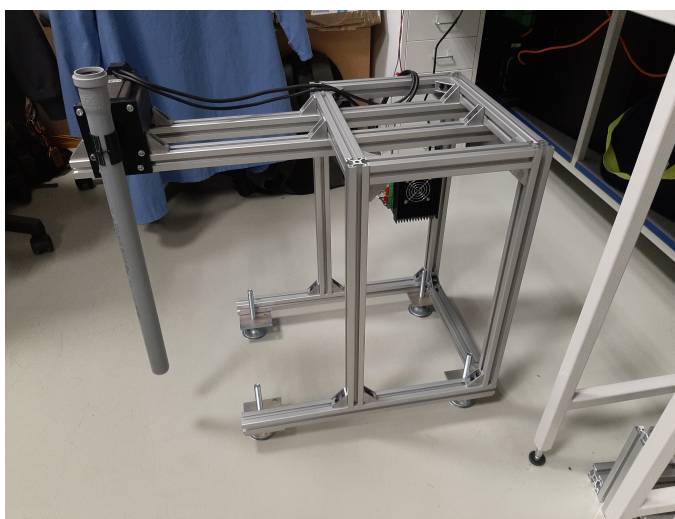
3 Realizace

Na základě navržené koncepce jsem simulátor sestavil. Kapitola se zabývá stavbou a úpravami konstrukce, propojení a komunikací driveru s počítačem. Dále je popsána funkčnost programu a možné možnosti ovládání simulátoru.

3.1 Konstrukce

Pro realizaci konstrukce jsem vyšel z navržené koncepce v kapitole 2.2. Zvolenou koncepcí jsem podrobněji zpracoval, rozdělil ji na jednotlivé profily, přidal k ní nastavitelné nožičky, aby bylo možné podle potřeby upravit výšku celé konstrukce. Kompletní soupis materiálu potřebného ke stavbě konstrukce je vložen v příloze B.1.

Dále jsem vymodeloval úchyt motoru, díl pro upevnění trubky simulující nohu na hřídel motoru. Namodelovaný úchyt motoru a trubky jsem vytisknu na 3D tiskárně. První verze je vidět na obrázku 3.1. Během testování se objevily nedostatky této první verze, proto jsem konstrukci postupně upravoval až do finální podoby (obrázek 3.2).



Obr. 3.1: První verze konstrukce

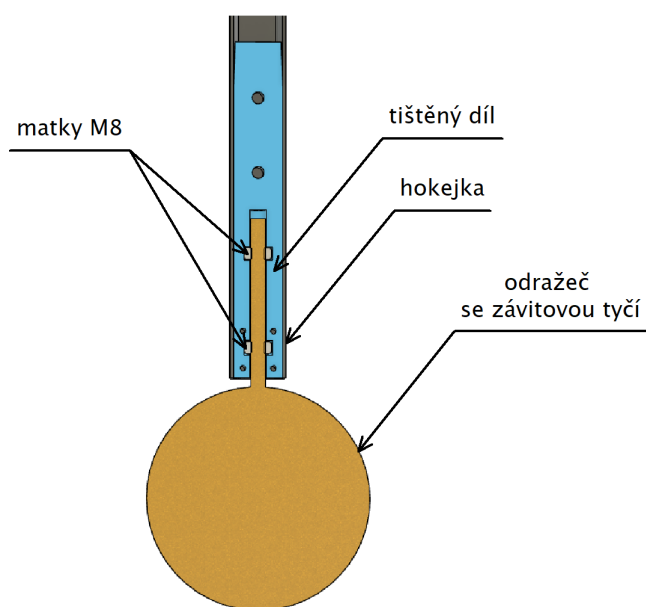
Rameno

Kanalizační trubka se během testování ukázala jako nevhodná pro simulátor kopu, protože se u rychlejších kopů příliš prohýbala. Nahradil jsem ji proto částí hokejky, která už dosahuje požadované tuhosti a zároveň stále nemá velký radarový odraz a odolává vodě.



Obr. 3.2: Vizualizace finální verze simulátoru

Do hokejky je vložen tištěný díl pro uchycení spodního nástavce, díl je pomocí šroubu M6 přichycen skrz hokejku. Uvnitř dílu jsou zatištěny matky M8 sloužící pro našroubování nástavce, náčres v řezu je vidět na obrázku 3.3.



Obr. 3.3: Řez dílem spojující rameno s odražečem

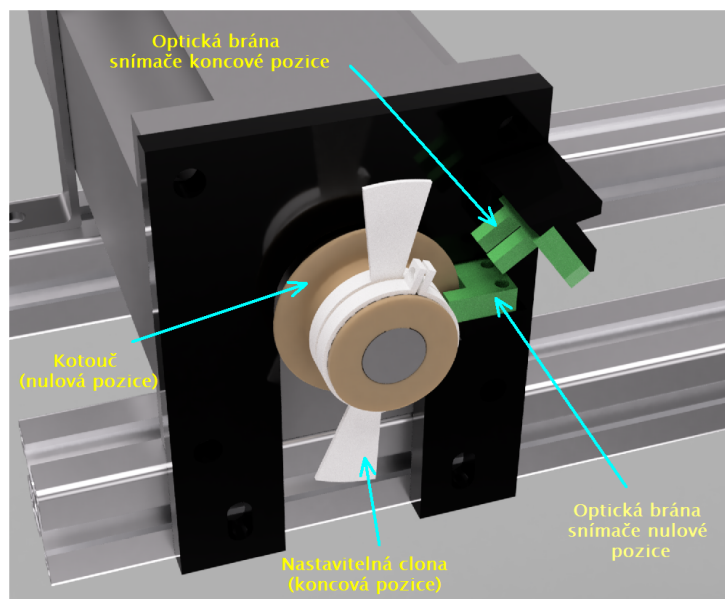
Nástavec

Jako nástavec jsem se rozhodl použít dutou ocelovou kouli o průměru 10 cm, takže v případě potřeby vyměnit za větší s průměrem 15 cm. Tyto koule jsem vybral jelikož jejich RCS (radar cross section, neboli efektivní obrazová plocha) nejlépe podobá lidské noze. Noha má RCS mezi -16 až -18 dBsm, 10 cm koule, která má RCS -16 dBsm, odpovídá tedy nejhorší variantě nohy. [2]

Díky tomu, že jsou nástavce přichyceny pomocí závitové tyče, mohou být jednoduše vyměnitelné a v budoucnu není problém přidat další, jako například model nohy s nasazenou botou.

Koncový snímač

Aby byl zabezpečený bezpečný provoz zařízení, v případě selhání ovládacího softwaru, nebo jiných částí řízení, rozhodl jsem se přidat koncové dorazy. Ty by měly zabránit nejen poškození simulátoru, ale hlavně poškození testovaných senzorů a nárazníků u kterých budou testy prováděny.



Obr. 3.4: Detail čela motoru se snímači

Pro detekci překročení povoleného rozsahu jsem se rozhodl umístit na hřídel motoru 2 clony. Na čelo motoru jsem umístit optickou bránu, která v případě že zaznamená přítomnost clony předá signál driveru, ten zajistí zastavení motoru. Zapojení snímače a jeho funkčnost bude diskutována níže. Část modelu na kterém je vidět umístění koncových snímačů je na obrázku 3.4.

Jelikož může být simulátor používán u různých modelů aut s různě vysokým nárazníkem, může se lišit maximální úhel kopu, je pozice obou clon nastavitelná.

Snímač nulové pozice

Jelikož driver má pouze relativní snímání natočení, tak po zapnutí nemá k dispozici aktuální pozici. Je proto nutné po zapnutí zajistit automatickou kalibraci. Proto jsem na hřídel motoru přidal kotouč a na pevnou část motoru optickou bránu. Kotouč je rozdělen do 2 částí. Pokud je kotouč plný, tak se rameno nachází nalevo od nulové pozice, prázdná část udává, že se rameno nachází napravo. Jeho umístění lze vidět na obrázku 3.4. Robot po inicializaci sám provede kalibraci do svislé polohy.

Po dokončení montáže se ukázalo, že konstrukce je dostatečně stabilní a není nutné ji zatěžkávat, jak jsem se původně domníval. Její vizualizace je na obrázku 3.2. Hotová konstrukce je vidět na obrázku 3.5.

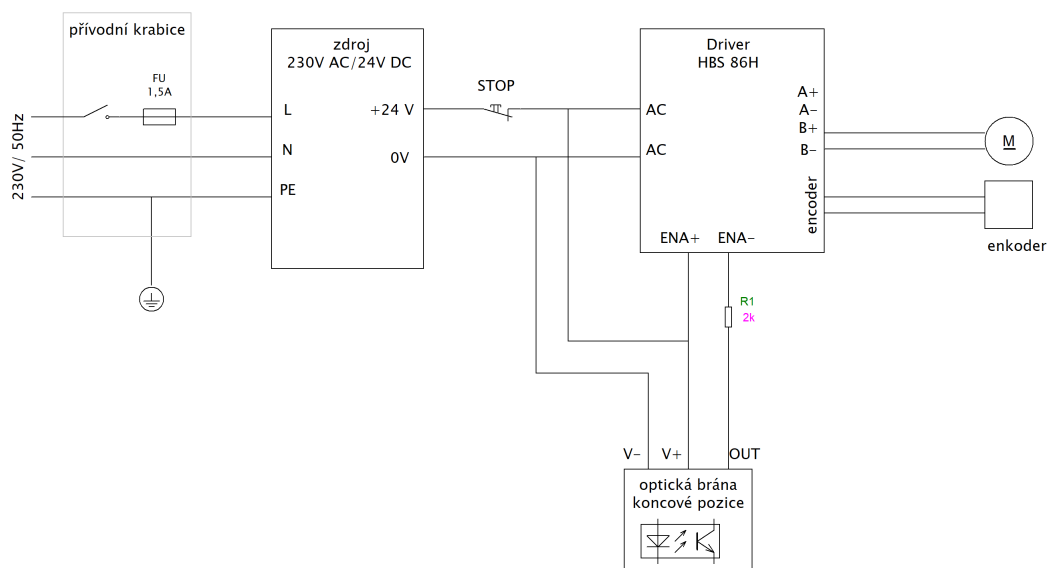


Obr. 3.5: Dokončená finální konstrukce

3.2 Elektrické zapojení

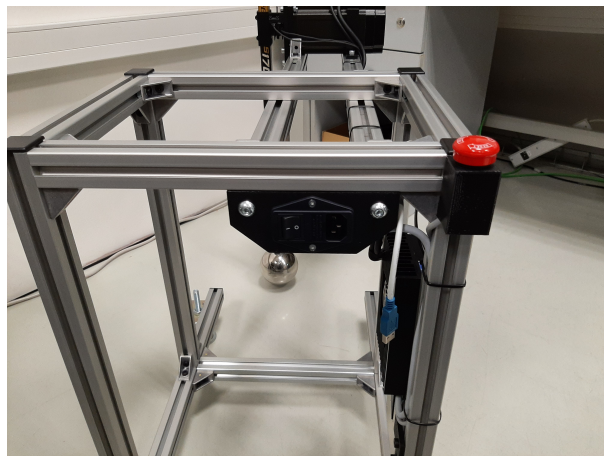
Elektrické zapojení simulátoru je zobrazeno na obrázku 3.6. Přesné typy použitých komponent jsou shrnuty v příloze v tabulce C.1.

Driver může být napájen stejnosměrným napětím mezi 24 V a 100 V, nebo střídavým mezi 18 V a 70 V[19]. Protože optická brána, kterou jsem se rozhodl použít pro detekování koncové pozice má maximální napájecí napětí 24 V, rozhodl jsem se použít pro napájení celého simulátoru 24 V zdroj[20].



Obr. 3.6: Schéma zapojení simulátoru

Simulátor je napájen pomocí univerzálního konektoru 230 V, použil jsem konektor, který má v sobě zabudovaný i vypínač a pojistku. Umístil jsem jej do vlastní krabičky přístupné zezadu simulátoru, jak je ukázáno na obrázku 3.7. Jelikož napájecí zdroj vyžaduje uzemnění, provedl jsem i uzemnění celé konstrukce.



Obr. 3.7: Zadní pohled na simulátor s přívodním konektorem

Pro potřeby rychlého vypnutí simulátoru jsem na něj umístil také nouzové stop tlačítko. Protože přívodní napájení už má vlastní vypínač integrovaný v konektoru, umístil jsem stop tlačítko až za zdroj 24 V, nebude díky tomu docházet ke zbytečnému vypínání a zapínání zdroje.

Výstup optické brány pro vymezení koncových pozic jsem připojil na enable pin driveru, jelikož je na vstupu enable pinů optočlen uzpůsobený na 5 V, tak jsem podle pokynů výrobce driveru připojil 24 V signál z optické brány přes 2 kΩ rezistor.

Připojení motoru a enkodéru pak odpovídá výchozímu zapojení stanovenému výrobcem. Přepínače v horní části driveru jsou všechny nastaveny do polohy OFF. První 4 spínače nastavují microstepping na maximální hodnotu. Spínačem 5 je pak nastaven směr kladných otáček, 6. spínač pak určuje typ ovládaného motoru.[19]

3.3 Propojení driveru s počítačem

Kapitola se věnuje způsobu předávání dat ze simulátoru do počítače. Popisuje propojení driveru a převodníku, včetně připojení signálu ze snímače nulové polohy a signálu pend. V kapitole jsou také popsány příkazy pro ovládání driveru.

3.3.1 Fyzická vrstva

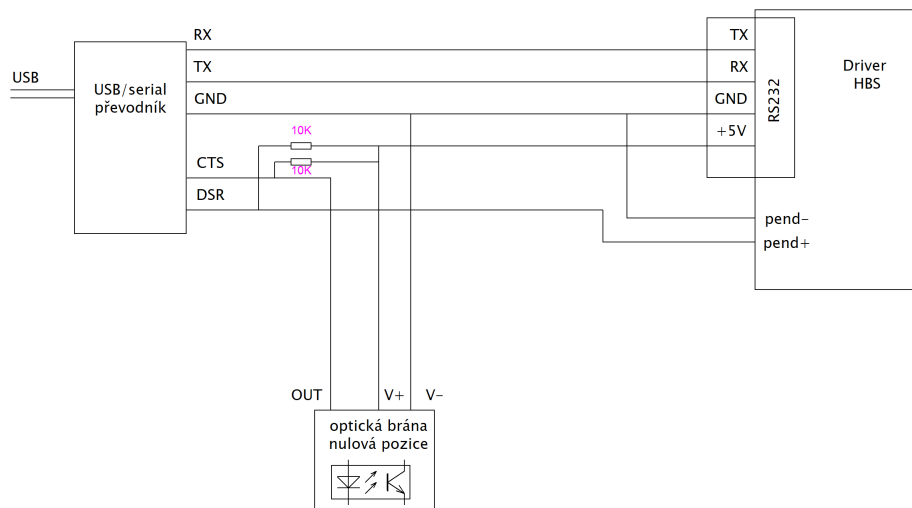
Driver s počítačem komunikuje pomocí RS232. Jak jsem změřil tak driver komunikuje na logické úrovni $\pm 7,8 V$. Problém je, že většina převodníků RS232 na UART využívá logické úrovně $\pm 5 V$, $\pm 10 V$, nebo $\pm 15 V$, když je driver připojen pomocí převodníku s logikou $\pm 5 V$, tak lze driveru posílat instrukce, ale nelze číst jeho odpovědi. Naopak pokud je použit převodník s vyšší napětovou úrovní, tak zafungují vstupní ochrany driveru a nelze mu poslat žádný povel.

Proto jsem se rozhodl použít převodník jehož součástí je levelshifter, konkrétně jsem použil převodník CHIP-10, který obsahuje levelshifter FT3243S, se vstupním rozsahem až $\pm 25 V$, ten převede signál na TTL logiku pro standardní FTDI USB/serial převodník [21, 22]. Vysílaný signál převodníku využívá logiky $\pm 5 V$ kterou driver zvládá přijímat.

3.3.2 Zapojení převodníku

Pro komunikaci driveru s počítačem jsou využity pouze signály Rx a Tx převodníku. Další volné signály (CTS a DSR) jsem použil pro signál z optické brány signalizující nulovou pozici a signál pend signalizující ukončení pohybu driverem. Využití signálu pend bude zmíněno později.

Oba zdroje signálu mají na výstupy NPN tranzistor, nelze je tedy na piny připojit přímo. Vstupní levelshifter bere jako logickou 0 pokud je napětí menší než 0,8 V a logickou 1 pokud je vyšší než 2,4 V. Mohl jsem si tedy dovolit použít zapojení na obrázku 3.8 [22]. Kde pro napájení snímače využívám 5 V výstupu driveru.



Obr. 3.8: Schéma propojení převodníku s driverem

Na CTS (DSR) pin převodníku je pak vyveden výstup snímače s pull up rezistorem připojeným na napájecích 5 V. Pokud je tedy tranzistor zavřen, je to převodníkem bráno jako logická 1. Měřením jsem ověřil, že při sepnutí tranzistoru pro pull up 10 k Ω je úbytek napětí na tranzistoru pod 0,8 V, bude tedy vyhodnocen jako logická 0.

3.3.3 Komunikační protokol

Výrobce k motoru dodává i program HBS protuner, přes který lze motor ovládat a nastavovat jeho parametry. Jde však pouze o manuální ovládání, což pro potřeby simulátoru není vhodné. Jelikož komunikační protokol není od výrobce zadokumentován, bylo tedy nutné zjistit jednotlivé příkazy pro ovládání motoru, které by poté bylo možné posílat po sériové lince z vlastního programu.

Nevhodnější variantou pro zjištění ovládacích příkazů pro mě bylo použít logický analyzátor. Ten jsem připojil přímo na signály Rx Tx jdoucí do driveru. V aplikaci pro ovládání driveru jsem postupně různě měnil parametry pohybu a pomocí aplikace k logickému analyzátoru si ukládal komunikaci počítače s driverem. Ze získaných dat jsem pak vydedukoval následující zákonitosti.

start	typ	nula	nastavovaný parametr	požadovaná hodnota		kontrola celistvosti	
0x01	0x06	0x00	0x16	0x00	0x64	0x69	0xE5

Tab. 3.1: Ukázka příkazu pro nastavení rychlosti

Komunikace probíhá pomocí daných rámců, o délce buď 8 (posílá instrukci včetně nastavované hodnoty) nebo 6 (pouze příkazy bez parametrů) bajtů. Ukázkou jednoho z rámce můžeme vidět v tabulce č 3.1, jedná se o příkaz pro nastavení rychlosti otáčení. Posílané hodnoty jsou zapsány v šestnáctkové soustavě. Příkaz se skládá z počátečního bajtu 0x01. Poté následuje informace o délce zprávy, pokud zpráva předává i požadovanou hodnotu, následuje za tímto bajtem ještě dalších 6, hodnota bajtu je tedy 0x06. Pro ostatní příkazy, které hodnotu nepředávají je 0x03. Další bajt je 0x00. Pak následuje kód nastavovaného parametru (pro rychlost 16), následovaný 2 bajty s požadovanou hodnotou parametru. V uvedeném příkladu 0x00 0x64, po převedení do desítkové soustavy zjistíme, že odpovídá 100 ot/min. Poslední 2 bajty slouží pro kontrolu celistvosti zprávy. Pro výpočet kontrolního součtu se využívá CRC (konkrétně CRC-16/MODBUS s prohozeným pořadím bajtů). Pokud je zaslána zpráva se špatným crc, driver pošle zpět chybovou hlášku, a provede pohyb s náhodnými parametry, proto je v programu potřebné kontrolovat odpovědi driveru.

funkce	kód příkazu	hodnoty	popis
směr otáčení	0x1A	0x00'01 0x00'00	proti směru hod. ručiček po směru hod. ručiček
aktivace opakování ¹	0x1C	0x00'01 0x00'00	aktivace / deaktivace opakovaného pohybu
zrychlení a zpomalení	0x15	0x00'C8	[$ot \cdot s^{-2}$]
požadovaný úhel	0x18	0x00'19	počet otočení ·100, př.: pro otoční o 90°, 90°= 0,25 ot, nastavená hodnota je $(25)_{10} = (19)_{16}$
zpoždění mezi cykly ¹	0x1B	0x00'64	[s]
počet cyklů ¹	0x19	0x00'01	
maximální rychlost	0x16	0x00'3C	[ot/min]
neznámá funkce	0xD0 0x41 0x42 0x09 0x14	0x00, 0x64 0x00, 0x01 0x00, 0x00 0x00, 0x01 0x00, 0x01

Tab. 3.2: Příkazy pro roztočení motoru

Při pokynu k pohybu vyšle aplikace do driveru sekvenci 12 příkazů s informacemi, jak se má motor pohybovat, po této sekvenci se začne motor otáčet. Příklad

¹Pro ovládání simulátoru je nevyužíván.

sekvence pro roztočení je uveden v tabulce 3.2. Řídící program pak jen udržuje komunikaci opakujícími se dotazy, jednou za sekundu pak vyšle dotaz na historii odchylky od požadované hodnoty za poslední sekundu a poté na jejich základě překreslí graf v aplikaci. V případě, že během pohybu přestane aplikace s driverem komunikovat, nebo dojde k rozpojení komunikace, driver na toto nijak nereaguje a dokončí nastavený pohyb.

Aplikace umožňuje nastavit i sekvenci opakujících se pohybů, nelze ale nastavit zpoždění mezi pohybem dopředu a zpět, navíc musí mít oba pohyby stejnou rychlost a zrychlení/zpomalení, což je pro simulátor nedostačující, proto budu využívat pouze příkazy pro nastavení pohybu a cyklování bude řešeno pomocí vlastní aplikace.

funkce	kód příkazu	hodnota	poznámka
kp i	0x00	0x00'C8	proporcionální složka proudové smyčky
ki i	0x01	0x00'64	integrační složka proudové smyčky
poč.pulzů na otáčku	0x0E	0x9C'40	
rozlišení enkodéru	0x0F	0x0F'A0	
max. chyba pozice	0x12	0x03'E8	po překročení se driver zastaví
kp p	0x06	0x09'C4	proporcionální složka poziční smyčky
ki p	0x07	0x01'F4	integrační složka poziční smyčky
kd p	0x08	0x00'64	derivační složka poziční smyčky
kvff	0x0D	0x00'1E	dopředná vazba rychlosti
I h	0x50	0x00'28	maximální proud přídržný [%]
I d	0x51	0x00'3C	dynamický proudový limit [%]
neznámá funkce	0x52	0x00'55	
	0x53	0x03'E8	
	0x96	0x00'01	
	0x97	0x00'01	
	0x54	0x00'01	
	0x55	0x64'00	
	0x4F	0x00'00	
	0xFF	0x00'04	
	0xFF	0x00'04	
0xFF	0x00'04		

Tab. 3.3: Příkazy pro nastavení parametrů driveru

Další skupinou příkazů jsou pokyny pro nastavení parametrů jako maximální proud, hodnoty PID regulátoru implantovaného v driveru a podobně, celá sekvence je uvedena v tabulce 3.3. Tuto sekvenci jsem se rozhodl poslat do driveru pokaždé při připojení simulátoru, aby byla jistota správného nastavení driveru.

3.3.4 Průběh kopu

Samotný kop probíhá v několika částech, nejdříve je poslán pokyn pro pohyb nahoru, jakmile je detekováno ukončení pohybu, program počká definované zpoždění, pokud je požadováno. Poté je vyslán pokyn k pohybu zpět dolů.

Detekci ukončení pohybu jsem nejdříve zjišťoval pomocí komunikace s driverem přes RS232, z programu jsem vyslal dotaz na aktuální position error (odchylku od požadované pozice) driveru, pokud byla odchylka rovna nule bylo jasné, že pohyb byl dokončen a noha se nachází na požadované pozici. Nevýhodou tohoto způsobu bylo, že občas docházelo k velkým zpožděním, protože i když byla noha na požadované pozici pořád existovala malá odchylka, kterou driver měřil a snažil se ji vyregulovat. Tento problém by šlo vyřešit zavedením minimální možné odchylky, kterou bych stanovil za přijatelnou, ale jak jsem zjistil, odchylku driver určuje jako rozdíl aktuální pozice a pozice ve které by měl motor v následujícím okamžiku být. Tudíž pokud bych povolenou odchylku zvýšil, tak by byl konec pohybu detekován výrazně dříve, než by se noha přiblížila požadované pozici.

Proto jsem hledal jinou možnost detekování dokončení pohybu. Na detekci jsem se rozhodl využít signálu pend driveru, ten spojí piny pend+ a pend - , pokud je pohyb dokončen tento signál potom pomocí pinu DSR zasílám do počítače. Tento způsob se osvědčil jako dostatečně stabilní.

3.4 Řídící software

Hlavní část řídicího softwaru je napsána ve formě třídy Kopac tak, aby mohla být volána buď z testovacího softwaru, který je již v Alps Alpine používán pro testování senzorů, nebo ze mnou vytvořené konzolové aplikace simulator_controler. Pro psaní programu jsem si vybral programovací jazyk python. Pro správnou funkci programu je nutné mít doinstalované knihovny pyserial(v3.5), pro konzolovou aplikaci pak i keyboard(v0.13.5). Jejich seznam pro instalování je také v souboru requirements.txt.

3.4.1 Využití v cizích aplikacích

Pro implementaci do stávajícího testovacího softwaru jsem po konzultaci s kolegy z Alps Alpine nadefinoval následující vlastní metody třídy, které budou sloužit jako rozhraní aplikace pro její implementaci.

get_device_list

Metoda prozkoumá všechny porty připojené k počítači a poté na základě sériového čísla vybere všechna kompatibilní zařízení, vrátí jejich seznam.

connect

Metodě je předán port, přes který má komunikovat s driverem, metoda se pokusí otevřít port a pošle testovací zprávu na otestování, jestli driver komunikuje správně. Pokud je komunikace navázána v pořádku, nahraje od driveru požadované nastavení (sekvence podobná jako tab.3.3).

kick_define

Metoda slouží k nadefinování parametrů kopu, jako je úhel kopu, maximální rychlost, zrychlení/zpomalení pro každou polovinu kopu a prodlevy mezi pohybem nahoru a dolů. Dalším volitelným parametrem metody je počáteční úhel kopu, pokud není zadána je jeho hodnota poloha 0° . Dalším parametrem lze vypnout kontroly maximálních povolených hodnot, tyto hodnoty jsem stanovil sám pro potřeby simulátoru. Posledním parametrem lze zakázat automatická kalibrace, která se jinak automaticky provádí v rámci metody.

kick_execute

Po zavolání této metody dojde k provedení jednoho kopu. Metoda má jeden volitelný parametr pomocí kterého lze provést jen vybranou polovinu kopu (pouze pohyb nahoru, nebo pohyb dolů).

emergency_stop

Pošle do driveru příkaz pro jeho okamžité zastavení. Konkrétně vyšle požadavek na posun o 0° vůči aktuální pozici, simulátor přeruší aktuálně vykonávaný pohyb a zastaví se.

home

Slouží pro přesun simulátoru do výchozí pozice. Běžně je kalibrace provedena během volání metody `kick_define`, nicméně pokud je z nějakého důvodu potřebné ji provést znovu je metoda uživateli zpřístupněna.

move

Metoda umožňuje nadřazenému programu pohnout s nohou o libovolný úhel bez použití `kick_execute`.

set_without_flag

Slouží pro přepnutí provádění do módu ve kterém neprobíhá žádná komunikace mezi driverem a počítačem, tento mód slouží pouze pro testování programu bez nutnosti mít připojený hardware k počítači.

Ukázka volání

Na výpisu 3.1 můžeme vidět ukázkou práce se simulátorem, nejdříve je volána funkce connect (řádek č.5) s číslem portu na kterém je simulátor připojen, ten je zadán uživatelem. Pro jeho výběr lze také využít metody get_device_list. Po úspěšném připojení je volána metoda kick_define, které jsou předány parametry kopu, v ukázce je definován kop o 20° s maximální rychlostí 10 ot/min při pohybu nahoru a 8 ot/min při pohybu dolů, se zrychleními $2 \text{ ot} \cdot \text{s}^{-2}$ při pohybu nahoru, $1 \text{ ot} \cdot \text{s}^{-2}$ při pohybu dolů. Zpoždění mezi pohybem nahoru a dolů je nastaveno na 0 s. Poté je v této ukázce ve for cyklu 6 krát za sebou volána metoda kick_execute, která kop vykoná, mezi jednotlivými kopy je 1 s rozestup.

Výpis 3.1: Příklad možné implementace simulátoru

```
1 import KopacControler
2 import time
3
4 kop=KopacControler.KopacControler()
5 kop.connect('COM5')
6 kop.kick_define(20,10,8,0,2,1)
7
8 for i in range(6):
9     kop.kick_execute()
10    time.sleep(1)
```

3.4.2 Konzolová aplikace

Konzolová aplikace pro ovládání simulátoru je vytvořena tak, aby provedla jednu sérii jednoho typu kopu. Parametry jsou předány při volání aplikace z příkazového řádku, možné parametry k nastavení jsou shrnuty v tabulce 3.4.

název parametru	popis	jednotka	výchozí hodnota
-port	port na kterém je simulátor připojen	-	-
-angle	požadovaný úhel kopu	°	10
-UpSpeed	maximální rychlost při pohybu nahoru	ot/min	10
-DwSpeed	maximální rychlost při pohybu dolů	ot/min	10
-UpDelay	zpoždění mezi pohybem nahoru a zpět	s	0,1
-UpAccel	zrychlení a zpomalení při pohybu nahoru	$ot \cdot s^{-2}$	2
-DwAccel	zrychlení a zpomalení při pohybu dolů	$ot \cdot s^{-2}$	2
-StartAngle	výchozí úhel pro kopy	°	0
-repeats	počet opakování kopu	-	1
-delay	časový rozestup mezi kopy	s	1

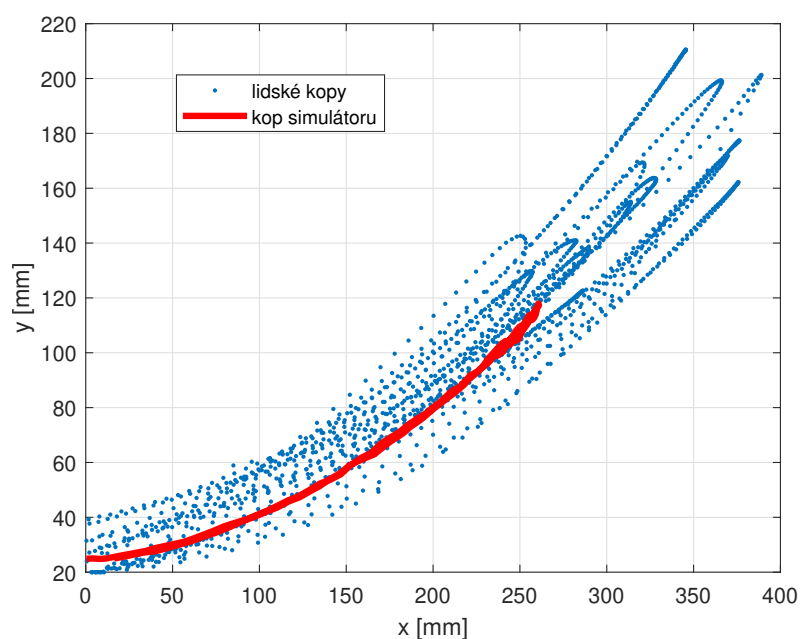
Tab. 3.4: Souhrn nastavitelných parametrů konzolové aplikace

4 Zhodnocení simulovaných kopů

S hotovým simulátorem jsem provedl několik srovnávacích testů. Pro porovnání jsem použil data natočena kamerou a data z radarového snímače kopu.

4.1 Porovnání simulovaných kopů s lidskými

Pro porovnání chování simulátoru s daty získanými v kapitole 2.1 jsem použil stejné kamery ve stejné vzdálenosti od nohy, natočené video jsem opět pomocí programu Tracker převedl na xy souřadnice v čase. Jak můžeme vidět na obrázku 4.1, trajektorie simulátoru odpovídá trajektoriím dříve změřených lidských kopů .

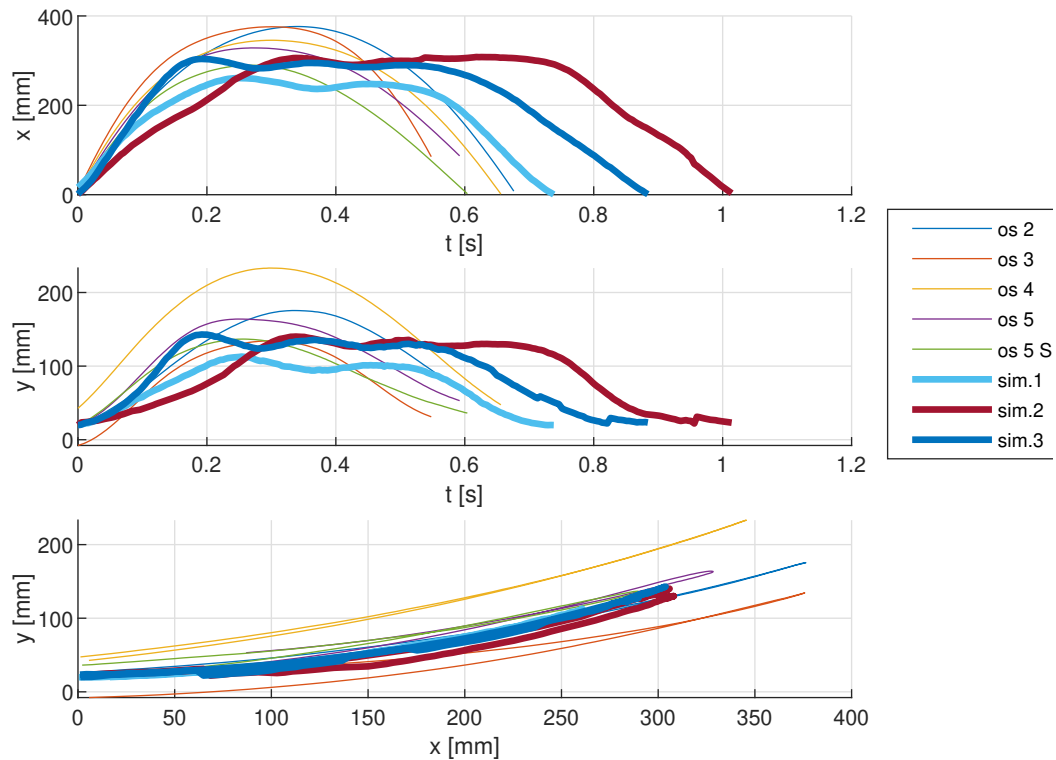


Obr. 4.1: Porovnání trajektorie simulátoru s lidskými kopy

Na obrázku 4.2 pak je srovnání kopů simulátoru se pomalými kopy testovacích osob. První graf zobrazuje závislost souřadnice x na čase, druhý závislost souřadnice y a třetí závislost x a y. Průběhy pojmenované jako os.xx, jsou zkušebních osob. Průběhy označené sim.xx, pak odpovídají kopům simulátoru. Kopy simulátoru jsem nastavil na úhel 40° , s maximální rychlostí 10 ot/min pro jednotlivé kopy jsem pak změnil zrychlení u kopu sim 1 je rovno $2 \text{ ot} \cdot \text{s}^{-2}$, sim 2 je $3 \text{ ot} \cdot \text{s}^{-2}$ a sim 3 je $3 \text{ ot} \cdot \text{s}^{-2}$.

Jak lze na grafech vidět kopy jsou srovnatelné s lidskými. Odlišností simulátoru je dlouhý čas mezi pohybem nahoru a dolu. Toto zpoždění je způsobeno prodlevami při komunikaci a celkově limitací operačního systému. Dalším zdrojem zpoždění je i driver. Ten se totiž ke konci pohybu snaží pomocí PID regulátoru nastavit polohu motoru na požadovanou pozici. Z dokumentace však není patrné jak je regulátor

implementován, takže jej nelze efektivně naladit. Jak je popsáno v kapitole 1.2.1, i tyto kopy jsou považovány za validní a jsou tak snímačem vyhodnoceny.



Obr. 4.2: Porovnání střehních kopů lidí s kopy simulátoru

Toto zpoždění by mohlo jít, při zachování stávajícího driveru, snížit tak, že by ovládací software nebyl spouštěn z počítače, ale například z jednodeskového počítače, jako například Raspberry Pi. Tím by se mohla snížit prodleva v komunikaci, navíc by šlo využít digitálních vstupů pro signál pend.

Protože je analýza kopy z videa poměrně časově náročná, tak jsem další měření s nastavením, které by přesněji odpovídalo ostatním kopům, již neprováděl.

4.2 Opakovatelnost

Funkčnost simulátoru jsme také ověřovali přímo na radarovém snímači kopů. Pomocí něj jsem zjistil konfigurace kopů které detekovány. Z těchto kopů jsem sestavil testovací sekvenci (příloha A.2), kopy jsou seřazeny od nejpomalejších po nejrychlejších. Každý z kopů jsem provedl 10krát za sebou. Poté jsem provedl analýzu dat radaru získaných z této zkušební sekvence. Kopy jsem v Matlabu poskládal přes sebe tak, aby byly časově synchronizované. Spočítal jsem rozptyl kopů v jednotlivých časových okamžicích podle vzorce 4.1. Poté jsem spočítal aritmetický průměr těchto rozptylů (označím jej A), podle tohoto parametru jsem poté schopen porovnávat jednotlivé

série mezi sebou, i přesto že každý trvá jinak dlouho. Protože se maximální hodnota jednotlivých kopů výrazně liší, rozhodl jsem se zavést pomocný ukazatel K, jako poměr průměru rozptylů A a maximální hodnoty daného kopu. Tuto hodnotu K budu označovat jako kvalitu kopu. Výsledky jsou shrnuté v tabulce 4.1.

$$\sigma_j^2 = \frac{1}{N-1} \sum_{i=1}^N |x_i - \mu| \quad (4.1)$$

kde μ určím jako

$$\mu = \frac{1}{N} \sum_{i=1}^N x_i \quad (4.2)$$

N....počet hodnot v daný časový okamžik [-]

x_ii-tá hodnota výstupu radaru [-]

σ_j^2rozptyl pro daný časový okamžik [-]

$$A = \frac{1}{N} \sum_j^M = \sigma_j^2 \quad (4.3)$$

$$K = \frac{A}{X_{max}} \quad (4.4)$$

A....střední hodnota odchylek dané série [-]

x_{max}maximální hodnota výstupu radaru [-]

K.... kvalita kopu [-]

Pro zhodnocení výsledků jsem použil data naměřená na podobném simulátoru, který využívá japonská pobočka Alps Alpine. K dispozici jsem měl pouze data z jedné série kopů. Kvalita kopů tohoto simulátoru je brána jako referenční.

kop	A[-]·10 ⁻³	K [-]·10 ⁻³
kop č.1	5,3	2,5
kop č.2	12	3,6
kop č.3	41	8,4
kop č.4	69	10,8
kop č.5	14	4,2
kop č.6	99	14,5
referenční	69	4,7

Tab. 4.1: Srovnání kvality opakovatelnosti jednotlivých kopů

Jak je vidět z dat tabulky 4.1, má simulátor u některých kopů lepší opakovatelnost než je reference (kopy 1 a 2). Hůře dopadly rychlejší kopy. Díky pomalému návratu dosahuje dobrých výsledků i kop č.5.

Horší opakovatelnost je způsobena mimo jiné nedostatečnou tuhostí hokejky, která se při kopech nepatrně prohýbá. Jako náhradu by šlo použít buď kovové rameno, které by bylo obloženo absorbéry, nebo nějaký pevnější plast.

Závěr

Bakalářská práce se zabývá návrhem a výrobou simulátoru kopů, pro společnost Alps Alpine. V práci jsou rozebrány možné kopy, které mohou sloužit k otevírání zavazadlového prostoru. Jelikož většina automobilek detekuje pouze přímé kopy, simulátor simuluje pouze je. Jedná se o rovné kopy s nataženou nohou pohybující se pouze v kyčli.

Pro určení parametrů simulovaných kopů jsem provedl vlastní analýzu přímého kopu. Pomocí dat získaný od 5 osob. Z analýzy, jsem získal požadované parametry motoru a ideální výšku konstrukce, která je 55 cm.

Pro pohon robota jsem vybral vhodný krokový motor s enkodérem, řízený driverem. Pro simulátor jsem navrhl vhodné řešení konstrukce. Konstrukci jsem poté poskládal z hliníkových profilů a doplnil ji 3D tištěnými díly.

Aby bylo možné motor ovládat z vlastní aplikace, odposlechl jsem komunikaci driveru s počítačovou aplikací dodanou k driveru. Na základě analýzy této komunikace, jsem zjistil zákonitosti použitého protokolu.

Zjištěný protokol jsem poté implementoval do vlastního programu. Pro ovládání simulátoru jsem napsal Pythonu vlastní třídu, kterou je možné použít v aplikacích 3. stran, nebo je možné ji použít přes mnou napsanou konzolovou aplikaci, pomocí které lze simulátor také ovládat.

Se simulátorem jsem provedl ověřovací testy. Pomocí analýzy videa jsem kopy simulátoru porovnal z lidskými. Ověřil jsem, že trajektorie simulátoru odpovídá trajektoriím lidských kopů. Analyzované kopy simulátoru jsou oproti lidským o něco pomalejší a mají delší prodlevu mezi pohybem nahoru a dolů, toto zpoždění je ale stále únosné. Při testování simulátoru pomocí radarového snímače kopu jsem odladil nastavení profilu kopu tak, aby odpovídaly lidským. Dále jsem na základě radarových dat zhodnotil jejich opakovatelnost, tou se podařilo přiblížit simulátoru, který je používán v japonské pobočce Alps Alpine.

Do budoucna pro zlepšení parametrů simulátoru je možné provést následující změny. Výměnu jeho ramene, za nové z pevnějšího materiálu, jelikož se stávající rameno při rychlejších kopech nepatrně prohýbá. Nahráním stávajícího softwaru do jednodeskového počítače s digitálními vstupy, by mohlo dojít ke snížení zpoždění mezi pohybem nahoru a dolů. Pro zjednodušení práce se simulátorem by také mohla být upravena konzolová aplikace. Do ní by mohlo být přidáno více možností, jak zasáhnout do běžícího testu.

Literatura

- [1] HONDA MOTOR. How to Activate the Hands-Free Power Tailgate. Honda [online]. [cit. 2022-12-18]. Dostupné z: <https://www.hondainfocenter.com/2020/Odyssey/How-To-Videos/Elite/Comfort-and-Convenience/How-to-Activate-the-Hands-Free-Power-Tailgate/>
- [2] ALPS ELECTRIC. Interní dokumentace společnosti
- [3] Brose [online]. [cit. 2022-12-18]. Dostupné z: <https://www.brose.com/de-en/products/technology-for-vehicle-doors-and-liftgates/>
- [4] Huf group / kick sensors [online]. [cit. 2022-12-18]. Dostupné z: <https://www.huf-group.com/en/products/kick-sensors-0>
- [5] TEXAS INSTRUMENTS. Automotive Capacitive Kick-to-Open Reference Design [online]. 2017 [cit. 2022-12-18]. Dostupné z: <https://www.ti.com/lit/ug/tidud21/tidud21.pdf?>
- [6] TEXAS INSTRUMENTS. AWR1642BOOST-ODS obstacle detection sensor [online]. [cit. 2022-12-18]. Dostupné z: <https://www.ti.com/tool/AWR1642BOOST-ODS>
- [7] INNOSENT. InnoSenT RADAR-BASED KICK SENSOR FOR CARS [online]. [cit. 2022-12-18]. Dostupné z: <https://www.innosent.de/en/automotive/kick-sensor>
- [8] TEXAS INSTRUMENTS. PGA460-Q1 in Automotive Ultrasonic Kick-to-Open Liftgate Systems [online]. 2018 [cit. 2022-12-18]. Dostupné z: <https://www.ti.com/lit/an/snva783b/snva783b.pdf?>
- [9] ŠKODA AUTO. Návod k obsluze Škoda Superb [online]. 2022 [cit. 2022-12-18]. Dostupné z: https://digital-manual.skoda-auto.com/w/cs_CZ/show/fb7512f314a769d47b46418a81134bfb_6_cs_CZ?ct=fb7512f314a769d47b46418a81134bfb_6_cs_CZ#titled19519644e17012
- [10] CHRYSLER. CHRYSLER PACIFICA User guide [online]. [cit. 2023-01-01]. Dostupné z: <https://www.fcacanada.ca/owners/en/manuals/2018/2018-Pacifica-UG-2nd.pdf>
- [11] VOLKSWAGEN AG. VW Tiguan Owner's manual [online]. [cit. 2023-01-01]. Dostupné z: <https://manual-directory.com/view-manual-pdf/?mid=29313>

- [12] HONDA. Uživatelská příručka Honda CR-V [online]. [cit. 2023-01-01]. Dostupné z:[https://www.honda.cz/cars/owners/owners_manuals/_jcr_content/par1/textcolumnwithimage/textColumn/richtextdownload_90f/file.res/19%20CR-V%20PET%20JSS%20\(KE%20KG\)-3KTLA601_CS_web_compressed.pdf](https://www.honda.cz/cars/owners/owners_manuals/_jcr_content/par1/textcolumnwithimage/textColumn/richtextdownload_90f/file.res/19%20CR-V%20PET%20JSS%20(KE%20KG)-3KTLA601_CS_web_compressed.pdf)
- [13] VOLVO. Operating the tailgate with foot movement [online]. [cit. 2023-01-01]. Dostupné z:<https://www.volvocars.com/uk/support/topics/use-your-car/car-functions/operating-the-tailgate-with-foot-movement>
- [14] TME. KROKOVÝ MOTOR [online]. [cit. 2023-01-02]. Dostupné z:<https://www.tme.eu/cz/news/library-articles/page/41861/krokovy-motor-druhy-a-priklady-aplikaci-krokovych-motoru/>
- [15] VEVOR. Nema34 12nm Closed Loop Hybrid Servo Motor Stepper Motor 205mm 6a 2phase [online]. [cit. 2023-01-02]. Dostupné z:https://eur.vevor.com/dc-stepper-motor-c_11221/nema34-12nm-closed-loop-hybrid-servo-motor-stepper-motor-205mm-6a-2phase-p_010367219430?v_tag=7515bc90-8a7c-11ed-8529-378706dc9728.1#js-HASH_FAQ
- [16] Servo motor [online]. [cit. 2023-01-02]. Dostupné z:<https://navody.drateg.cz/arduino-projekty/servo-motor.html>
- [17] AMAZON. Metal-Gear Steering-Servo [online]. [cit. 2023-01-02]. Dostupné z:https://www.amazon.com/AGFRC-Metal-Gear-Steering-Servo-Motor-High-Torque-Programmable/dp/B09KRPFQNR/ref=sr_1_155?keywords=servo%2Bmotors&qid=1665080539&qu=eyJxc2MiOiI1LjQxIiwicXNhIjoicjoiNC43OSIsInFzcCI6IjQuN%3D%3D&prefix=servo%2B%2Caps%2C262&sr=8-155&th=1
- [18] Amazon Nema 34 Stepper Motor 6A 12Nm [online]. [cit. 2023-05-12]. Dostupné z: https://www.amazon.com/Stepper-Motor-156mm-Length-Router/dp/B077XHBTZ9/ref=sr_1_1?keywords=stepper+motor+nema+34&qid=1683882163&sr=8-1
- [19] Hybrid Stepper Servo Drive Manual: HBS86H [online]. [cit. 2023-01-02]. Dostupné z: https://hardware-cnc.nl/images/PDF/ACT/HBS86H_English_Manuel.pdf
- [20] OMRON. Photomicrosensor with Slim Cable (Non-modulated) EE-SX77/87 [online]. [cit. 2023-05-15]. Dostupné z:<https://docs.rs-online.com/1ff8/0900766b81398ec9.pdf>

- [21] FTDI CHIP. Chipi-X Cable Datasheet [online]. 2012 [cit. 2023-04-23]. Dostupné z: <https://www.farnell.com/datasheets/1730387.pdf>
- [22] FT3243S datasheet [online]. [cit. 2023-04-23]. FTDI Chip. Dostupné z: https://ftdichip.com/wp-content/uploads/2020/08/DS_FT3243S.pdf

Seznam příloh

A	Analýza kopů	49
A.1	Lidské kopy	49
A.2	Kopy simulátoru	49
B	Soupis materiálu pro stavbu konstrukce	50
B.1	Rám simulátoru	50
B.2	Ostatní komponenty	50
C	Elektrické zapojení	51
D	Obsah elektronické přílohy	53

A Analýza kopů

A.1 Lidské kopy

osoba	výška [cm]	věk [let]	velikost boty
os1	187	33	44
os2	174	28	42
os3	178	39	41
os4	172	21	43
os5	183	34	44

Tab. A.1: Přehled testovacích osob

A.2 Kopy simulátoru

kop	úhel [°]	ω_u [ot/min]	ω_d [ot/min]	t_d [s]	ϵ_u [ot · s ⁻²]	ϵ_d [ot · s ⁻²]
kop č.1	40	3	3	0	1	1
kop č.2	40	20	20	0	1	1
kop č.3	40	20	20	0	2	2
kop č.4	40	20	20	0	3	3
kop č.5	40	90	20	0	4	2
kop č.6	40	90	20	0	5	2

Tab. A.2: Parametry kopů použitých pro testy simulátoru

ω_u [ot/min]	maximální úhlová rychlost při pohybu nahoru
ω_d [ot/min]	maximální úhlová rychlost při pohybu dolů
t_d [s]	zpoždění mezi pohybem nahoru a dolů
ϵ_u [ot · s ⁻²]	úhlové zrychlení a zpomalení při pohybu nahoru
ϵ_d [ot · s ⁻²]	úhlové zrychlení a zpomalení při pohybu dolů

B Soupis materiálu pro stavbu konstrukce

B.1 Rám simulátoru

Pro stavbu rámu jsem využil dílů z e-shopu : <https://uk.misumi-ec.com/>, uvedená katalogová čísla jsou pro tento e-shop.

Tab. B.1: Soupis hliníkových profilů 30x30

délka[mm]	katalogové číslo	počet kusů
500	HFS6-3030-500	6
310	HFS6-3030-310	3
240	HFS6-3030-240	2
700	HFS6-3030-700	2

Tab. B.2: Soupis ostatního materiálu potřebného pro stavbu rámu

materiál	katalogové číslo	počet kusů
nohy	HAJPS6	4
rohové spojky	HBLFSN6	28
šrouby	79848009060010	70
zátky	HFC6-3030-B	12
matice	HNTT6-6	76

B.2 Ostatní komponenty

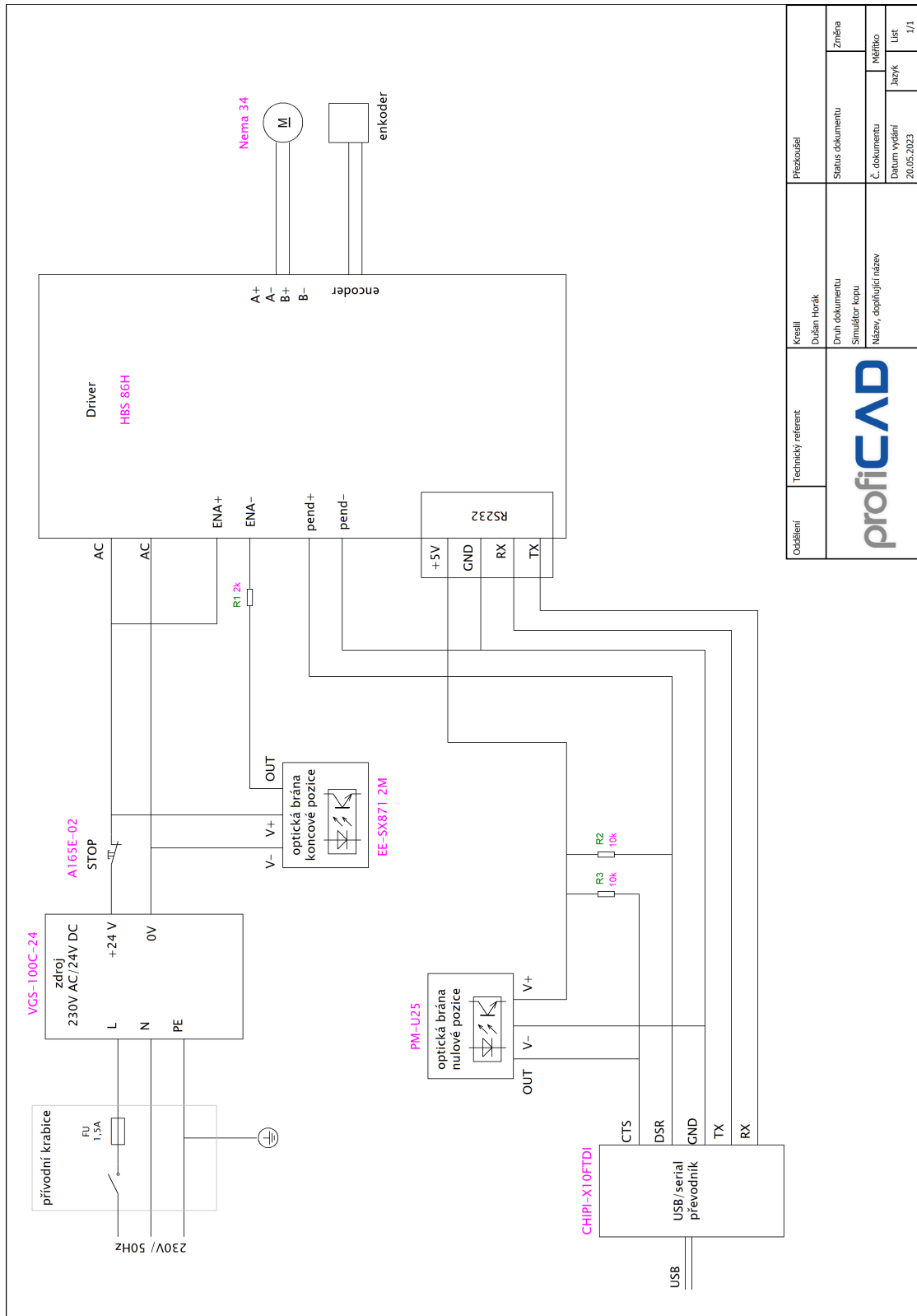
Tab. B.3: Zbylé mechanické komponenty


komponent	označení	e-shop
nerezová koule $\phi 10$ cm	Dekoratивní koule nerezová 10 cm	obi
nerezová koule $\phi 15$ cm	Dekoratивní koule nerezová 15 cm	obi

C Elektrické zapojení

Tab. C.1: Soupis elektronických komponentů

komponent	označení	e-shop
driver	HBS 86H	Vevor
motor	Nema 34	-
USB uart převodník	CHUPI-X10FTDI	tme
optická brána nulové pozice	PM-U25 PANASONIC	tme
optická brána koncové pozice	omron EE-SX871 2M	omron
napájecí zdroj	VGS-100C-24	mouser
stop tlačítko	omron A165E-02	mouser
přívodní konektor	SCHURTER 4304.6058	tme



Oddělení	Technický referent	Kreslí	Přezkoušel
		Dušan Horák	
		Druh dokumentu Simulátor kopu	Status dokumentu Změna
		Název, doplňující název	Č. dokumentu
			Měřítka
			Jazyk
			1/1

Obr. C.1: Celkové elektrické schéma simulátoru

D Obsah elektronické přílohy

/.....	kořenový adresář přiloženého archivu
├ Simulator_kopu.pdf	elektronická verze semestrální práce
├ software	
│ └ Kopac.py	třída ovládající simulátoru
│ └ simulator_controler.py	konzolová aplikace pro ovládání simulátoru
│ └ requirements.txt	baličky které je nutné doinstalovat
├ konstrukce	
│ └ konstrukce_komplet.f3d	kompletní model konstrukce
│ └ clona_koncove_poz.stl	
│ └ krabice_privodA.stl	
│ └ krabice_privodB.stl	
│ └ kryt_kabelaze.stl	
│ └ stativ.stl	
│ └ stop_tlacitko_uchyceni.stl	
│ └ stop_tlacito_kryt.stl	
│ └ uchyt_motor-rameno.stl	
│ └ uchyt_motoru_predni.stl	
│ └ uchyt_motoru_zadni.stl	
│ └ uchyt_nastvec-rameno.stl	
│ └ uchyt_zdrojeA.stl	
│ └ uchyt_zdrojeB.stl	
├ analyza kopu	
│ └ data	složka s txt soubory s nasbíranými daty pro analýzy
│ └ rozbor_kou.mlx	skript pro vyhodnocení dat natočených kamerou
│ └ kresli.mlx	pomocná funkce volaná skriptem rozbor_kopu.mlx
│ └ kresli2.mlx	pomocná funkce volaná skriptem rozbor_kopu.mlx
│ └ prolozeni.m	pomocná funkce volaná skriptem rozbor_kopu.mlx
│ └ prumer.mlx	pomocná funkce volaná skriptem rozbor_kopu.mlx
│ └ DataLoad.m	pomocná funkce volaná skriptem rozbor_kopu.mlx
│ └ opakovatelnost.m	skript pro vyhodnocení opakovatelnosti
│ └ seradit.m	pomocná funkce volaná skriptem opakovatelnost.m