

Bakalářská práce



**České
vysoké
učení technické
v Praze**

F3

**Fakulta elektrotechnická
Katedra řídicí techniky**

Hexapod

Václav Šíp

**Vedoucí: doc. Ing. Radoslav Bortel, Ph.D.
Studijní program: Kybernetika a robotika
Srpen 2021**

I. OSOBNÍ A STUDIJNÍ ÚDAJE

Příjmení: **Šíp** Jméno: **Václav** Osobní číslo: **483639**
Fakulta/ústav: **Fakulta elektrotechnická**
Zadávací katedra/ústav: **Katedra řídicí techniky**
Studijní program: **Kybernetika a robotika**

II. ÚDAJE K BAKALÁŘSKÉ PRÁCI

Název bakalářské práce:

Hexapod

Název bakalářské práce anglicky:

Hexapod

Pokyny pro vypracování:

Navrhněte a zrealizujte hexapod.

- Navrhněte algoritmus pro výpočet inverzní kinematiky ramen hexapodu.
- Navrhněte algoritmus pro generování cest pro chůzi hexapodu.
- Navrhněte řídicí elektroniku hexapodu.
- Navrhněte mechanickou konstrukci hexapodu.
- Zrealizujte hexapod

Seznam doporučené literatury:

- [1] Dorf, C.R., Bishop, R.H.: Modern Control Systems, Prentice Hall, 2011.
- [2] Spong, M.W., Vidyasagar, M.: Robot Dynamics and Control, John Wiley & Sons, 1989.

Jméno a pracoviště vedoucí(ho) bakalářské práce:

doc. Ing. Radoslav Bortel, Ph.D., katedra teorie obvodů FEL

Jméno a pracoviště druhé(ho) vedoucí(ho) nebo konzultanta(ky) bakalářské práce:

Datum zadání bakalářské práce: **20.01.2021**

Termín odevzdání bakalářské práce: _____

Platnost zadání bakalářské práce: **30.09.2022**

doc. Ing. Radoslav Bortel, Ph.D.
podpis vedoucí(ho) práce

prof. Ing. Michael Šebek, DrSc.
podpis vedoucí(ho) ústavu/katedry

prof. Mgr. Petr Páta, Ph.D.
podpis děkana(ky)

III. PŘEVZETÍ ZADÁNÍ

Student bere na vědomí, že je povinen vypracovat bakalářskou práci samostatně, bez cizí pomoci, s výjimkou poskytnutých konzultací. Seznam použité literatury, jiných pramenů a jmen konzultantů je třeba uvést v bakalářské práci.

Datum převzetí zadání

Podpis studenta

Poděkování

Děkuji vedoucímu práce docentu Radoslavu Bortelovi za jeho vynikající vedení a skvělou pomoc při řešení projektu. Děkuji ČVUT, že mi je tak dobrou *alma mater*.

Prohlášení

Prohlašuji, že jsem předloženou práci vypracoval samostatně, a že jsem uvedl veškerou použitou literaturu.

V Praze, 8. srpna 2021

Abstrakt

Tato práce se zabývá navrhnutím a realizací hexapoda, což je pozemní robot s šesti nohama. Návrh hexapoda se skládá z mechanické konstrukce, úpravy serv, vybrání řídicí jednotky i návrhu algoritmu, který řídí jeho chůzi. Je zde spočítána inverzní kinematika ramen a přidána možnost natáčet tělo hexapoda.

Pro ovládání hexapoda na dálku je popsána funkčnost jednoduché aplikace na PC.

Klíčová slova: elektronika, kinematika, 3D tisk

Vedoucí: doc. Ing. Radoslav Bortel, Ph.D.

Abstract

This work deals with design and realization of a hexapod. A hexapod is a ground robot with six legs. The design consists of a mechanical construction, servo electronic design, selecting suitable control unit and algorithm design, which controls the walk of the hexapod. Inverse kinematics of arms is calculated and a limited possibility of rotating body is added.

Last, the work describes the functionality of a simple application on PC for remote control of hexapod.

Keywords: electronics, kinematics, 3D printing

Title translation: Hexapod

Obsah

1 Úvod	1		
2 Servo	2		
2.1 Volba serva	2		
2.2 Představení zakoupeného výrobku	2		
2.3 Návrh elektroniky	3		
2.4 Řízení polohy	7		
2.4.1 Regulátor	7		
2.4.2 Generování cest	7		
2.4.3 Implementace na servo	8		
2.4.4 Kalibrace	10		
2.5 Komunikace s řídicí jednotkou ..	12		
2.5.1 Protokol komunikace	12		
2.6 Mechanické úpravy	14		
3 Řídicí jednotka	15		
4 Mechanické díly	16		
4.1 Noha	16		
4.1.1 Návrh nových dílů	17		
4.1.2 Sestavení nohy v modelovacím programu	20		
4.2 Tělo	20		
5 Konstrukce	22		
5.1 Potřebný spojovací materiál a další elektronika	22		
5.2 Sestavení	22		
6 Ovládání hexapoda	25		
6.1 Inverzní kinematika	27		
6.2 Posunutí a rotace těla hexapoda	29		
		6.3 Chůze	30
		6.3.1 Chůze v osách x a y	31
		6.3.2 Trajektorie v ose z	33
		6.4 Ovládání přes aplikaci na PC pomocí WiFi	34
		7 Evoluce	35
		8 Závěr	36
		Literatura	37

Kapitola 1

Úvod

Roboti začínají být nedílnou součástí našich životů. Jsou tu jako naši pomocníci, kteří usnadní práci, nebo dokonce provedou úkony, které by pro člověka mohly být příliš nebezpečné, či dokonce nemožné.

Přestože navrhnout robota, který se pohybuje po kolečkách může být snazší, výhody pozemního robota s nohama tento fakt jistě převáží, neboť nohy dávají možnost pohybu v nerovném terénu a překonávání překážek.

Jedním z robotů, který se pohybuje po nohách je právě hexapod, jehož styl chůze je založen na chůzi mravenců a dokáže se tak pohybovat i ve špatně přístupném terénu.

Na trhu existuje mnoho předpřipravených vývojových kitů, které umožní si hexapoda postavit s minimálními znalostmi a nízkými náklady. Není to ovšem jediná varianta. Ve své práci navrhnu a zrealizuji hexapoda bez těchto kitů.

Navrhnu mechanickou konstrukci těla i nohou, vyberu serva a řídicí elektroniku, která bude schopná obstát i při nutnosti vyšší výpočetní síly pro libovolný účel hexapoda.

Vytvořím jednoduchý algoritmus chůze, který by sice v nehostinném prostředí neobstál, ale jako základní způsob pohybu bude dostatečný.

Před samotným závěrem se objeví několik nápadů na zlepšení vytvořeného hexapoda a případně i jeho následné rozšíření, které může být nutné pro autonomní provozování hexapoda.

Kapitola 2

Servo

2.1 Volba serva

Servo vhodné pro hexapod musí mít nízkou váhu a dostatečnou sílu.

Rozdělí-li se vhodná serva do 3 kategorií dle ceny, tak v **nejnižší** kategorii, s cenou do \$25, lze očekávat serva s kovovou převodovkou, nekvalitním enkodérem, obvykle v podobě potenciometru, a ovládání pomocí pulzně šířkové modulace (PWM). Patří sem například JX Servo RD-5622MG (\$21) a RDS3225 (\$9).

Ve **střední** třídě zhruba do \$100 již je možné serva ovládat přes UART, to přidává možnost plně kontrolovat pohyb pro různé aplikace. Některé již mají i magnetický enkóder a jsou schopné poměrně přesného pohybu s chybou pod $0,1^\circ$. Jejich síla je srovnatelná se servy, které najdeme v nižší kategorii. Zástupci jsou DYNAMIXEL XL430-W250-T (\$59) a OCSERVO ROBS-301 (\$72).

V **nejvyšší** kategorii již mizí omezení nižších tříd, jsou vytvořené z kvalitnějších materiálů, nabízejí vyšší točivý moment a jako enkodéry slouží i absolutní senzory polohy. Příkladem jsou DYNAMIXEL XH430-V350-R (\$330) a Feetech RS485 (\$202).

Pokud je potřeba, aby se hexapod pohyboval s vysokou přesností a finance nejsou problémem, pak je vhodné vybírat serva ze střední třídy. Když se ke zmíněnému přidá i požadavek na větší rozměry hexapoda, nebo vyšší nosnost, pak je již potřeba zvolit serva z nejvyšší kategorie. Jelikož ovšem žádný takový požadavek není stanoven, **postačí** v rámci úspory **serva z nejnižší** kategorie.

2.2 Představení zakoupeného výrobku

Bylo pořízeno servo RDS3225 z nejnižší kategorie (obr. 2.1), přičemž celkově jich je potřeba pro jednoho hexapoda 30, po pěti pro každou nohu.



Obrázek 2.1: Použité servo RDS3225 [Serpig]

Vybráno bylo kvůli jeho nízké ceně (\$9), použitelnému motoru, rozsahu otáčení (270°) a dobré převodovce. Navíc v ceně jsou i hliníkové držáky, které se využijí k uchycení serva.

Ovládat natočení serva je možné pomocí PWM. Toto ovládání není vhodné, neboť řídicí jednotka není schopná vysílat 30x PWM. Proto se ze serva odstraní původní elektronika a nahradí se novou.

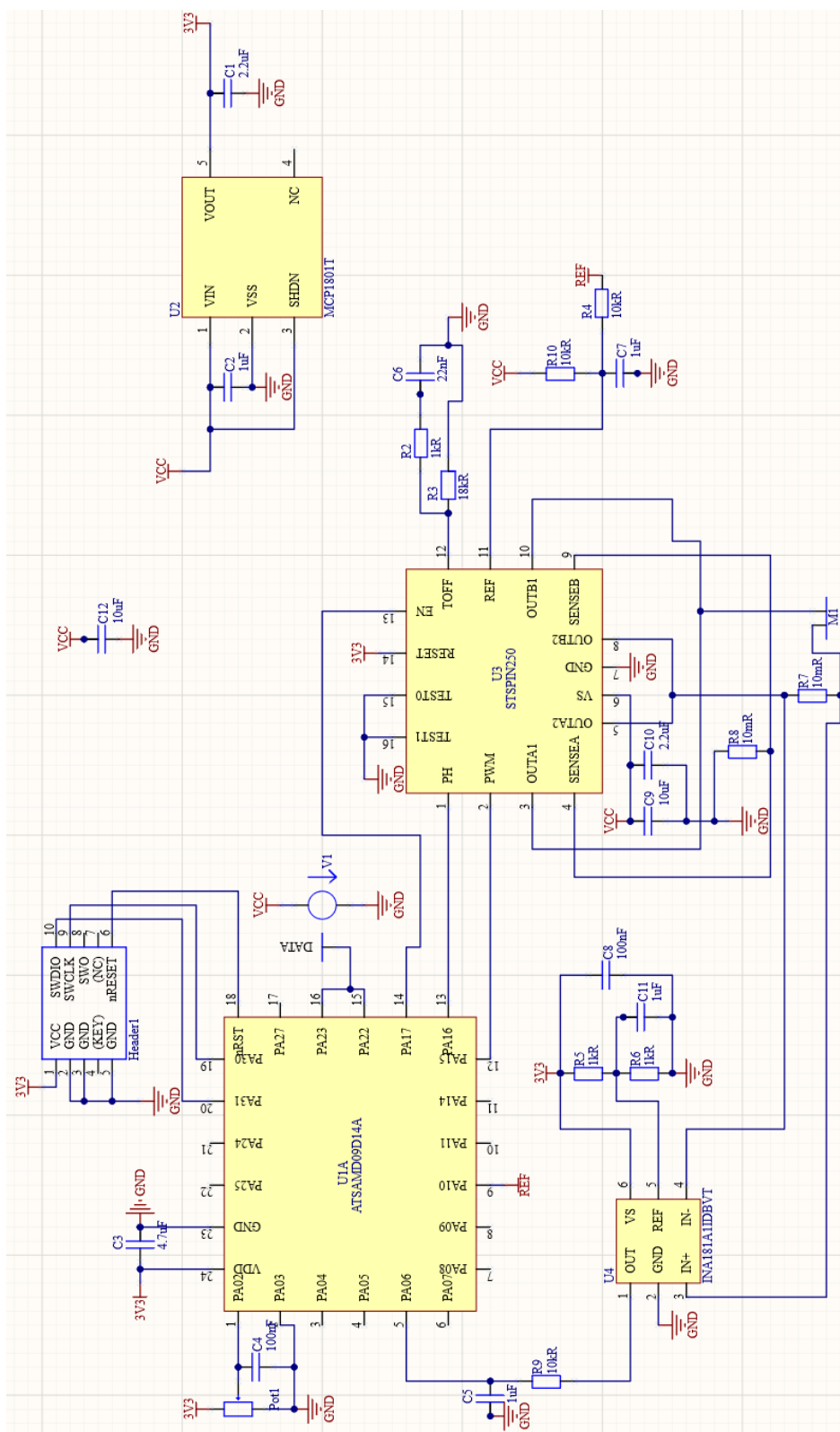
2.3 Návrh elektroniky

K návrhu byl použit program Altium Designer, který umožňuje jak schématický návrh, tak návrh plošného spoje.

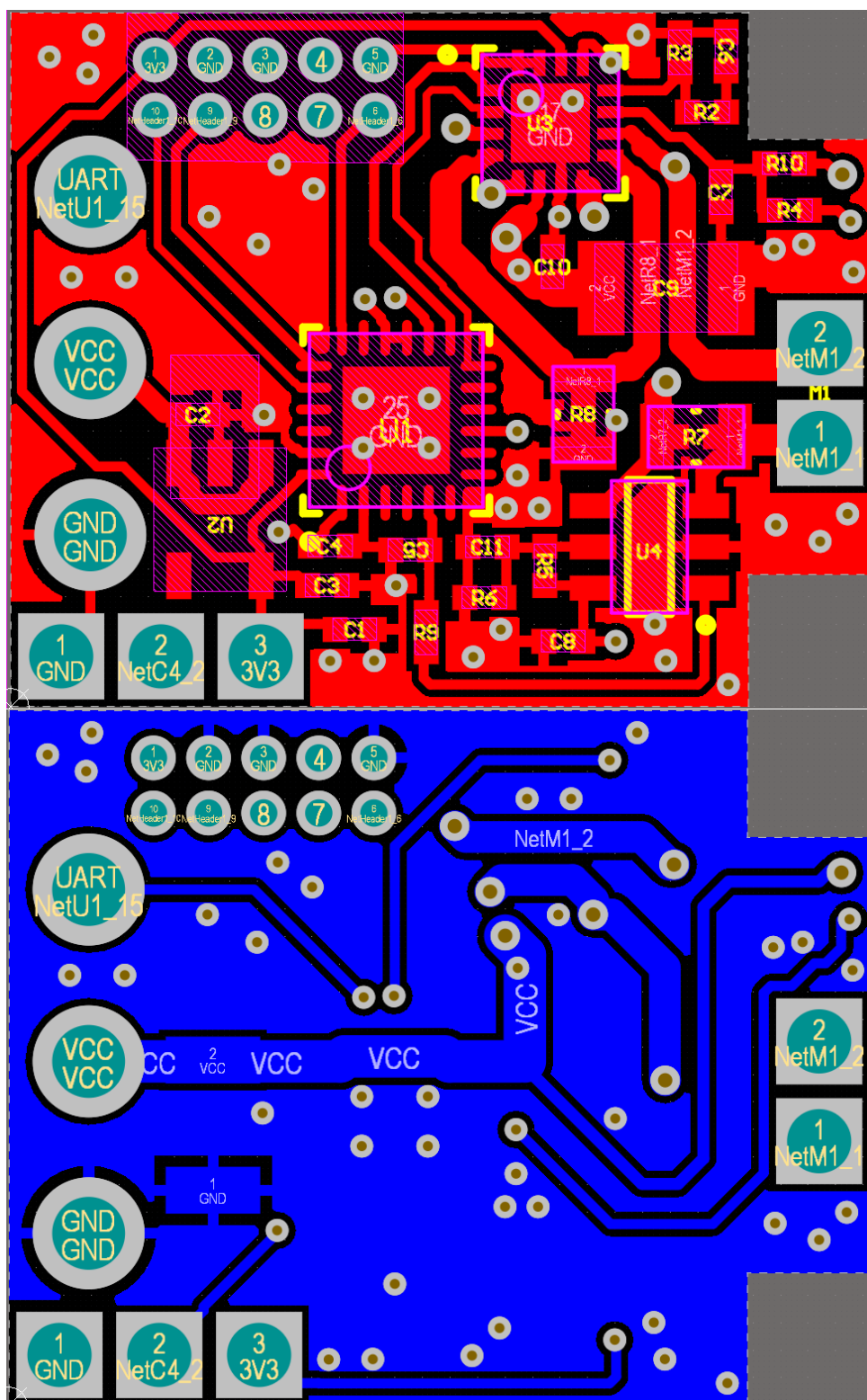
Od nové elektroniky je vyžadováno ovládání polohy, řízení serva pomocí příkazů přes UART a možnost poskytování základních údajů o stavu serva jako je současná poloha a proud protékající motorem. Navíc osazený plošný spoj musí velikostně pasovat do serva, proto je potřeba volit součástky v menších pouzdrech. Byly vybrány kondenzátory a rezistory v pouzdrech 0402 (1x0.5 mm).

Množství	Označení	Popis
2	C4,C8	100 nF
2	C9,C12	10 μ F Tantalový elektrolytický z původního serva
4	C2,C5,C7,C11	1 μ F
2	C1,C10	2,2 μ F
1	C6	22 nF
1	C3	4,7 μ F
3	R4,R9,R10	10 k Ω
2	R7,R8	10 m Ω 200 W
1	R3	Rezistor 18 k Ω
3	R2,R5,R6	1 k Ω
1	U1	ATSAMD09D14A-MUT nebo ATSAMD11D14A-MUT
1	Header1	Pinový header 2199SB-10G-301523
1	V1	Pinový header Molex 26-48-1036
1	U4	Operační zesilovač INA181A1IDBVT
1	U2	LDO regulátor napětí MCP1801T-3302I/OT
1	U3	Motor driver STSPIN250
1	-	Konektor TE Connectivity 3-644465-3

Tabulka 2.1: Použité součástky na jedno servo



Obrázek 2.2: Schéma zapojení obvodu



Obrázek 2.3: Návrh rozložení na plošném spoji

Jelikož servo bude mít zdroj s napětím v rozmezí 5 až 8 V, je potřeba použít LDO regulátor (Low-dropout regulator), jehož výstupem bude 3.3 V pro řídicí elektroniku. Byl vybrán model (MCP1801T-3302I/OT) s vysokým PSRR (Power Supply Rejection Ratio) kvůli potlačení šumu pro ADC.

Procesor ATSAM09D14A-MUT byl vybrán hlavně kvůli jeho vysoké hodinové frekvenci, která činí 48 MHz, a zároveň malým rozměrům. Další jeho výhodou je možnost programování ve studiu Atmel v jazyce C. Obsahuje i potřebný AD převodník s minimálním rozlišením na 12 bitů.

Je možné procesor nahradit i modelem ATSAM11D14A-MUT, který má totožné rozmístění pinů a případný program z prvního procesoru běží i na tomto.

Ke spínání motoru je potřeba plný H-můstek, který je integrován spolu s driverem (viz U3 v obr. 2.2). K jeho zapojení je potřeba poznamenat, že rezistor R10 nebyl osazen, a tedy spojení rezistoru R4 a kondenzátoru C7 je dolní propust pro PWM, kterou vysílá procesor, čímž se nastaví referenční napětí pro driver. Pokud driver naměří na rezistoru R8 vyšší napětí než je ono referenční, tak se kvůli nadproudové ochraně vypne.

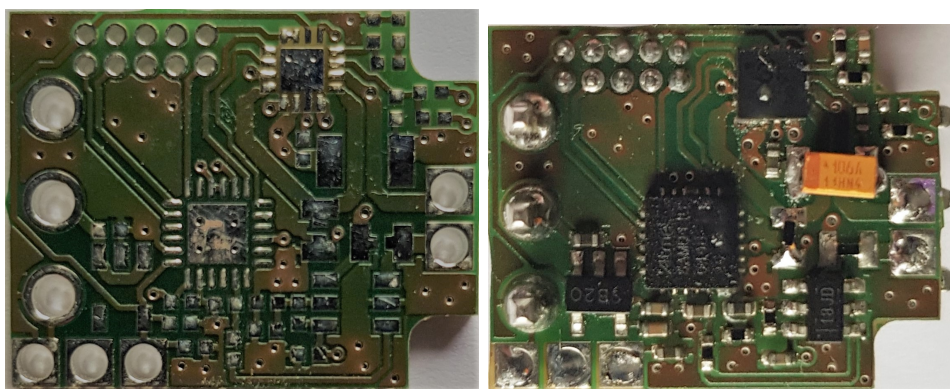
K měření proudu protékajícím motorem je využit rezistor R7 o velikosti 10 mΩ a rozdílový zesilovač, na jehož výstupu je dolní propust.

Způsob měření polohy je zachován z původní elektroniky v podobě potenciometru a napětí z něho je snímáno diferenciálně, tedy napětí mezi pinem procesoru 1 a 2.

Z původní elektroniky se zrecyklují dva tantalové elektrolytické 10 μF kondenzátory, přičemž jeden se umístí blízko Driveru, který tento náboj využije hlavně při otevírání tranzistorů, a druhý již libovolně.

Jak již bylo řečeno výše, součástky jsou v malých pouzdrech, proto je většina součástek na plošný spoj připájena pomocí pájecí pasty.

Zdůvodnění výběru komunikace přes UART bude uvedeno v kapitole 2.5.



Obrázek 2.4: Deska plošných spojů

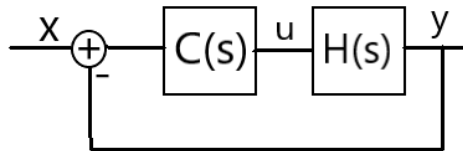
2.4 Řízení polohy

Motor se řídí PWM signálem, u kterého se mění jeho střída. Čím větší střída je, tím déle budou příslušné tranzistory v H-můstku otevřeny a tím více se může dodat energie motoru.

Vhodným regulátorem, který bude řídit střidu vstupního PWM signálu, lze dosáhnout požadované polohy.

2.4.1 Regulátor

Vyjdeme ze schématu na obr. 2.5, kde x představuje požadovanou polohu, C je hledaný regulátor a y je skutečná polohice, která v servu bude měřena. Celkový přenos z x na y je



Obrázek 2.5: Schéma systému s regulátorem

$$Y(s) = \frac{H(s)C(s)}{1 + H(s)C(s)} X(s). \quad (2.1)$$

Nejčastější a nejjednodušší volba regulátoru pro serva je PID regulátor. Jeho přenos v Laplasově transformaci je

$$C(s) = \frac{k_D s^2 + k_P s + k_I}{s}. \quad (2.2)$$

Konstanty se naleznou experimentálně v kapitole 2.4.3.

2.4.2 Generování cest

Jelikož je známá současná polohice natočení serva, cílová polohice a stanoví se doba, za kterou se do cílové polohice servo otočí, lze si vytvořit cestu, jejíž každý bod bude znázorňovat polohu natočení serva v určitém okamžiku. Což znamená, že dva známé body se proloží určitou křivkou. Tuto trasu lze udělat tak, aby se zajistilo splnění stanovených podmínek, které se mohou klást, jak na rychlost v některých bodech, tak i na zrychlení.

V servu jsou implementovány 3 možnosti pojmenované jako profily zrychlení. Profil 0 je použití přímky, kde polohice v čase 0 odpovídá současné poloze x_0 a

v čase T cílové pozici x_1 .

$$p(t) = x_0 + c_1 t, \quad (2.3)$$

$$c_1 = \frac{x_1 - x_0}{T}. \quad (2.4)$$

Profil 1 je použití kvadratické funkce. Aby bylo možné funkci přesně určit, musí se stanovit ještě jedna podmínka kromě již daných pozic. V cílovém bodě má být rychlost otáčení serva nulová.

$$p(t) = x_0 + c_1 t + c_2 t^2, \quad (2.5)$$

$$p'(T) = c_1 + 2c_2 T = 0, \quad (2.6)$$

$$c_1 = -2c_2 T, \quad (2.7)$$

$$c_2 = \frac{x_0 - x_1}{T^2}. \quad (2.8)$$

Poslední je profil 2, který využívá polynom třetího stupně. Zde bude požadováno, aby i rychlost na začátku byla nulová.

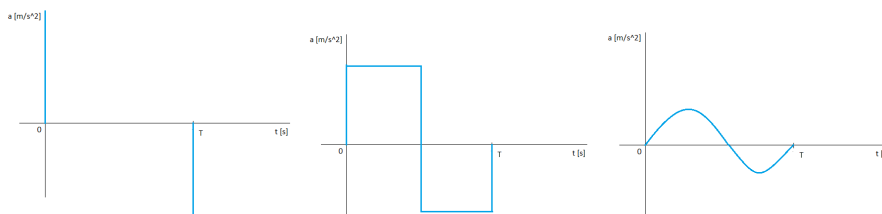
$$p(t) = x_0 + c_1 t + c_2 t^2 + c_3 t^3, \quad (2.9)$$

$$p'(0) = c_1 = 0, \quad (2.10)$$

$$c_3 = 2 \cdot \frac{x_0 - x_1}{T^3}, \quad (2.11)$$

$$c_2 = -\frac{3}{2} T c_3. \quad (2.12)$$

Vybráním jednoho z výše uvedených profilů se zároveň vybírá průběh zrychlení, kdy například pro profil 0 to budou Diracovy impulsy, viz níže.



Obrázek 2.6: Průběhy zrychlení pro jednotlivé profily
Zleva - profil 0, profil 1, profil 2

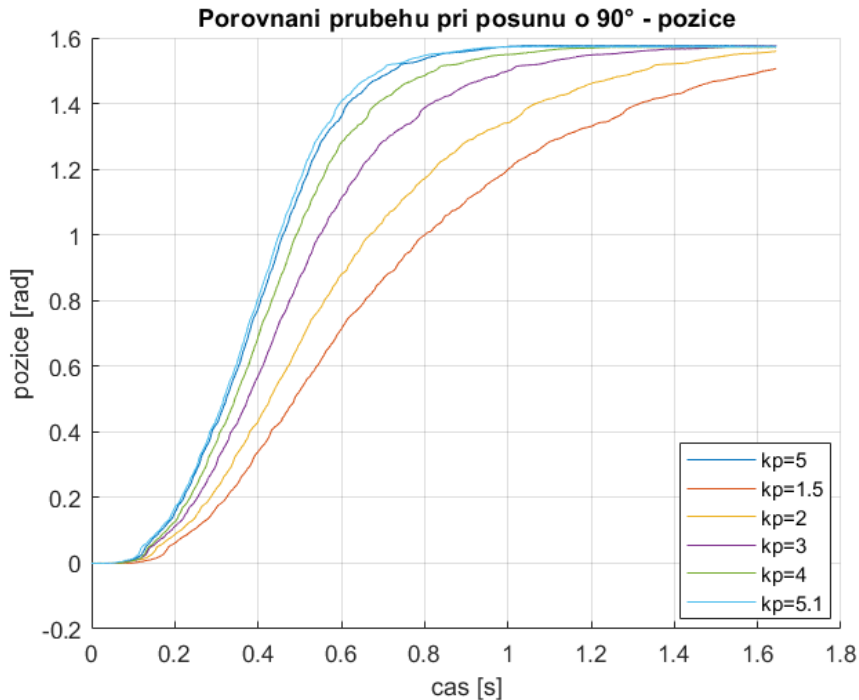
2.4.3 Implementace na servo

Výsledkem PID regulátoru je hodnota napětí. Četnost úpravy napětí poslané na motor se řídí dle rychlosti měření napětí na potenciometru pomocí AD převodníku v procesoru.

Měřením bylo zjištěno, že upravit hodnotu napětí lze každých $T_{\text{cycle}} = 2.14 \text{ ms}$.

Ze znalosti vzorkovací periody lze implementovat PID regulátor spolu s generováním cest.

V tuto chvíli je potřeba experimentálně zjistit konstanty pro PID regulátor. Jelikož je nyní servo bez zátěže, což na hexapodovi nenestane, ladí se konstanty jen za účelem ověření funkčnosti serva a vytvořených algoritmů. Zvolí-li se složky regulátoru $\mathbf{K}_I = 0.0001$ a $\mathbf{K}_D = 0.01$, pak výsledný průběh polohy při různých hodnotách proporcionalní složky ukazuje následující graf.



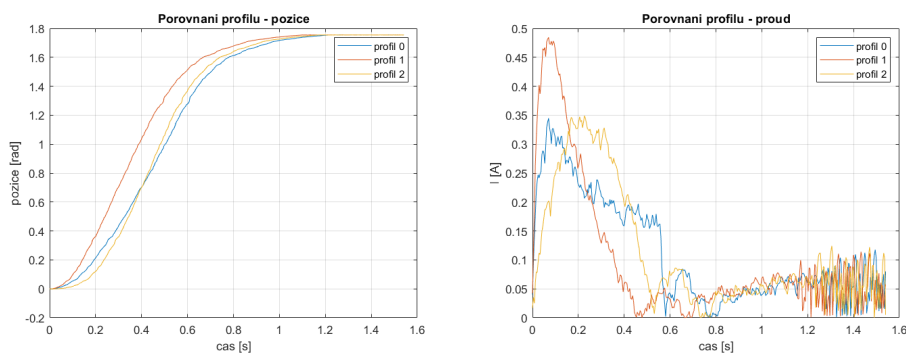
Obrázek 2.7: Porovnání průběhů pro různé proporcionalní složky (k_p)

$K_p = 5.1$ je nejvyšší možné, neboť při vyšší proporcionalní složce se již servo nárazově snaží odebrat více než 2.5 A proudu.

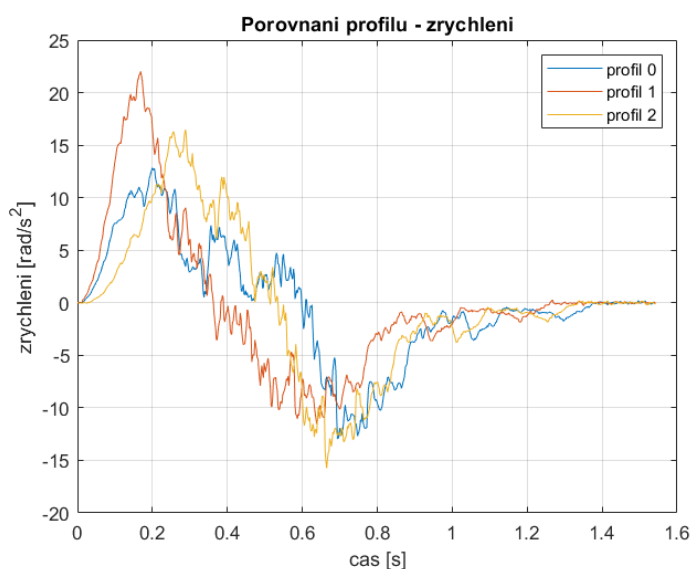
Z těchto proporcionalních složek je tedy dobré zvolit hodnotu v rozmezí 2.5 až 4.3. Zvolením $K_p = 4$ se zajistí dostatečná rychlost otáčení serva a přiměřená velikost odebíraného proudu.

Výsledný regulátor má složky $\mathbf{K}_p = 4$, $\mathbf{K}_I = 0.0001$ a $\mathbf{K}_D = 0.01$.

Pro toto nastavení regulátoru lze porovnáním jednotlivých profilů zjistit, jak moc se blíží realita k teorii. Na následujících grafech jsou průběhy naměřené pro jednotlivé profily při otočení o 90° .



Obrázek 2.8: Porovnání profilů pro průběhy polohy a proudu



Obrázek 2.9: Porovnání profilů pro zrychlení

Jednotlivé profily vygenerovaly velmi podobné průběhy pro pozici, kde plánovaná trajektorie z teorie je správně jen pro profil 2. Z toho důvodu průběhy zrychlení vypadají podobně.

Vygenerované průběhy pro jednotlivé profily se od sebe příliš neliší, přesto je vidět, že pro profil 2 zrychlení naroste pomaleji a tím méně namáhá servo.

2.4.4 Kalibrace

Jelikož skutečný úhel natočení se liší oproti požadované hodnotě, je potřeba servo zkalibrovat. Vybere se sedm různých poloh, u kterých se naměří skutečný úhel v radiánech. V takovém případě je k dispozici jak skutečný úhel, tak i hodnota polohy ze serva.

K interpolaci se využije **kubický spline**. Mějme x_1, \dots, x_n skutečných poloh a c_1, \dots, c_n jim příslušících hodnot získaných z měření napětí na potenciometru,

kde $n = 7$. Potom lze získat funkce

$$s_j(y) = k_{3,j}y^3 + k_{2,j}y^2 + k_{1,j}y + c_j, \quad (2.13)$$

kde $j = 1, \dots, n - 1$, $y = p - x_j$ a p je požadovaný úhel natočení v radiánech. Pro hodnotu j platí, že

$$x_j \leq p \wedge x_{j+1} > p, \quad (2.14)$$

s krajními výjimkami pro

$$p < x_1 \text{ platí } j = 1, \quad (2.15)$$

$$p > x_6 \text{ je } j = 6. \quad (2.16)$$

Funkce $s_j(y)$ vrátí hodnotu, která musí být na potencoimetru (z ADC), aby servo bylo na požadované poloze.

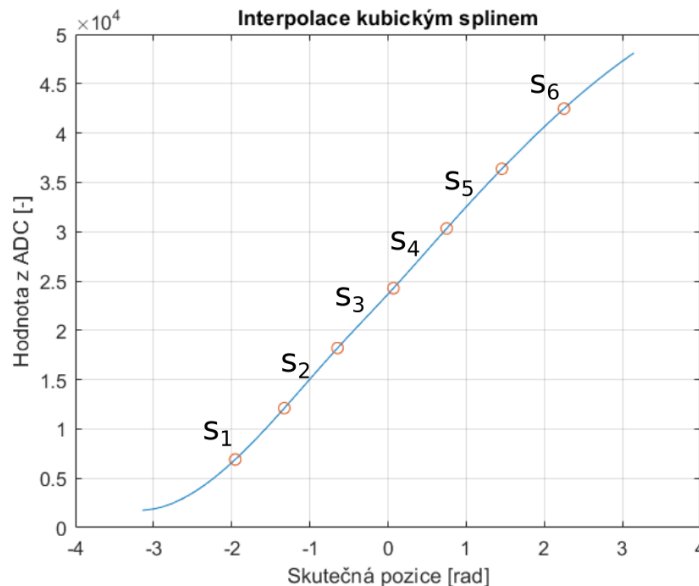
Pro $s_j(y)$ musí platit tyto podmínky:

$$s_j(0) = c_j, \quad (2.17)$$

$$s_j(x_{j+1} - x_j) = c_{j+1}, \quad (2.18)$$

a dále je potřeba, aby byly první a druhé derivace spojité. Navíc je potřeba splnit i tzv. not-a-knot koncovou podmínku, která požaduje, aby funkce s_1 i s_{n-1} měly spojitou i třetí derivaci.

Jednotlivé koeficienty byly vypočteny v matlabu pomocí funkce csapi.



Obrázek 2.10: Ukázka interpolace pro jedno z serv
Pozn.: Kolečka jsou naměřené plohy.

2.5 Komunikace s řídicí jednotkou

Využitý protokol ke komunikaci je UART na 3.3 V. Tento protokol byl vybrán z důvodu nutnosti řízení většího množství serv. Protokol zajišťuje výhodnou sběrníkovou topologii, čímž se zajistí minimální počet drátů a zároveň je možné komunikovat dostatečně vysokou rychlostí (viz dále). Z důvodu ušetření kabelů byl UART implementován jako 1-drátová verze místo obvyklých dvou.

Komunikace je nastavená tak, aby očekávala jako první nejméně významný bit (LSB), 1 stop bit a bez parity.

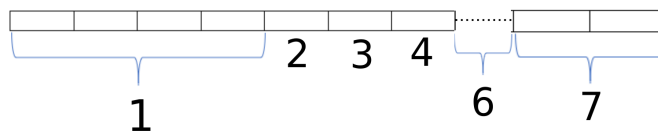
Rychlost komunikace je v základu nastavená na baud rate 9600, ovšem je možné to změnit až na 2 000 000. U prototypu se ukázalo jako nejlepší volba zvolit baud rate 1 000 000, neboť při rychlosti 1.5M+ servo častokrát nezvládne zprávu správně přijmout.

Při rychlosti 1M bylo potřeba řešit u prototypu i časování pro správné přijetí zpráv. Mezi jednotlivými zprávami musí být mezera alespoň 180 μ s a dále musí být mezera mezi zprávami pro stejné servo, která činí 2 - 5 ms, neboť to je doba, kterou servu obvykle trvá přijatou zprávu zpracovat. Nejvíce času je potřeba, pokud se mění cílová pozice viz Tabulka 2.5.

2.5.1 Protokol komunikace

Pro komunikaci mezi servem a řídicí jednotkou byl vytvořen komunikační protokol pro výměnu zpráv. Rámec jedné zprávy se liší v závislosti na tom, odkud zpráva pochází.

Pochází-li zpráva od řídicí jednotky, pak rámec jedné zprávy vypadá následovně:

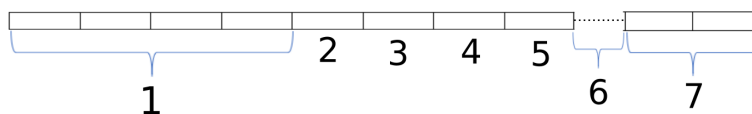


Obrázek 2.11: Rámec zprávy pro servo

Číslo	Název	Velikost v bytech	Popis
1	Hlavička	4	Vždy se jedná o 0xF8F00800
2	ID	1	Číslo serva, kde 0xFE je broadcast
3	Zasat	1	1 - má zapsat do registru, 0 - má přečíst
4	Délka	1	Počet bytů k zapsání nebo přečtení
5	Registr	1	Obsahuje číslo registru
6	Data	0 až 4	Obsahuje data k zapsání
7	CRC	2	Obsahuje CRC ke kontrole validity zprávy

Tabulka 2.2: Význam jednotlivých bytů ve zprávě pro servo

Pokud je vyžadována odpověď, nebo byl zaslán požadavek o čtení, servo odpoví následující zprávou:



Obrázek 2.12: Rámec zprávy pro řídicí jednotku

Číslo	Název	Velikost	Popis
1	Hlavička	4	Vždy se jedná o 0xF8F00800
2	ID	1	Číslo serva, kde 0xFE je broadcast
3	Chyba	1	viz tab. 2.4
4	Délka	1	Počet bytů v datech + velikost CRC
5	Data	0 až 4	Obsahuje přečtená data
6	CRC	2	Obsahuje CRC ke kontrole validity zprávy

Tabulka 2.3: Význam jednotlivých bytů ve zprávě pro řídicí jednotku

Číslo	Popis
0	Nenastala žádná chyba
1	Zpráva nebyla doručena správně, chyba v CRC
2	Byla spuštěna nadproudová ochrana nebo ochrana proti přehřátí

Tabulka 2.4: Význam chybového čísla ve zprávě pro řídicí jednotku

Pomocí těchto zpráv lze přistoupit k několika registrům (viz tab. 2.5), z kterých lze jak data číst, tak do některých i zapsat. Není možné číst data, pokud se použije ID broadcastu.

Číslo	Název	Velikost	Popis
0	ID	1	Obsahuje číslo serva
1	Baud rate	1	0-9600, 1-1M, 2-2M, 3-57600 a 4-112500
2	Odpověď	1	0 - Jen na čtení, 1 - vždy
3	Současná pozice	2	0 až 48496
4	Cílová pozice	2	Od 45° - 6062 do 315° - 42434
5	Zapnout motor	1	1 - zapnuto, 0 - vypnuto
6	Profil	1	Hodnoty - 0, 1 a 2. Viz kap. 2.4.2
7	K_p	4	Desetinné číslo typu float pro proporcionální složku
8	K_I	4	Desetinné číslo typu float pro integrační složku
9	K_D	4	Desetinné číslo typu float pro derivační složku
10	Proud	4	Desetinné číslo typu float pro proud [A]
11	Čekání	1	Doba mezi požadavkem na čtení a odesláním odpovědi
12	Měření	1	Poměr měření proudu ku pozici, výchozí je 50 - 1:50

Tabulka 2.5: Seznam registrů

Jediný registr, do kterého nelze zapsat hodnotu, je číslo 3- současná pozice.

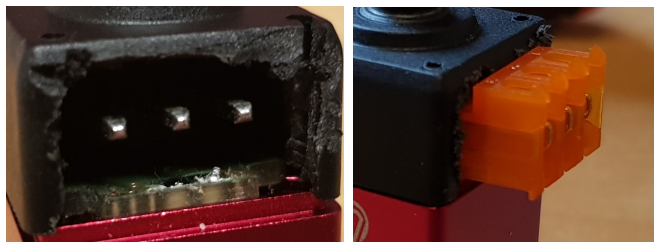
Všechny ostatní jsou přístupné pro obě operace.

První dva registry, tedy ID a Baud rate, jsou zapamatovány i po vypnutí napájení, neboť jsou uloženy v paměti EEPROM.

■ 2.6 Mechanické úpravy

Původně ze serva vychází 3 dráty, ke kterým by se měla připojit řídicí jednotka. Tento způsob je však nevyhovující v případě, že bychom potřebovali serva připojit na společnou sběrnici. Proto byl na plošném spoji umístěn pinový header (26-48-1036), ke kterému se pomocí konektoru (3-644465-3) dá snadno připojit.

Aby se dal konektor použít, bylo potřebné zvětšit otvor v plastovém krytu.



Obrázek 2.13: Mechanická úprava na servu pro konektor

Konektor není příliš mohutný, takže nebude bránit při pohybu serva a zároveň zjednoduší kabeláž, neboť kabely stačí silou přimáčknout do otvorů, kde kovové konce proříznou izolaci a tím se "připojí". Není tedy potřeba na kabel přidělovat žádný konektor.

Kapitola 3

Řídící jednotka

Řídící jednotka musí být dostatečně rychlá, aby byla schopna počítat postavění nohou několikrát za sekundu, přijímat data z aplikace přes WiFi nebo Bluetooth a zároveň být schopná posílat data přes UART rychlostí vyšší jak 1 000 000 baud rate.

Tyto požadavky není příliš těžké splnit, proto existuje mnoho kandidátů. Dobrou volbou může být některá vývojová deska Arduino, případně jiné podobné vývojové desky. Je tu samozřejmě i možnost vytvořit vlastní plošný spoj, kdy by se zvolil vhodný procesor a přidali potřebné součástky.

Jako vhodná řídicí jednotka byl nakonec vybrán rovnou mini počítač Raspberry Pi Model B 4 GB RAM.



Obrázek 3.1: Raspberry Pi 4 [Bb3]

Raspberry má jako mini počítač víc než dostatek výkonu na pokrytí potřeb pro řízení hexapoda. Má v sobě zabudované moduly pro Wifi a Bluetooth. Nabízí spoustu možností při budoucích rozšířeních, či přidání senzorů.

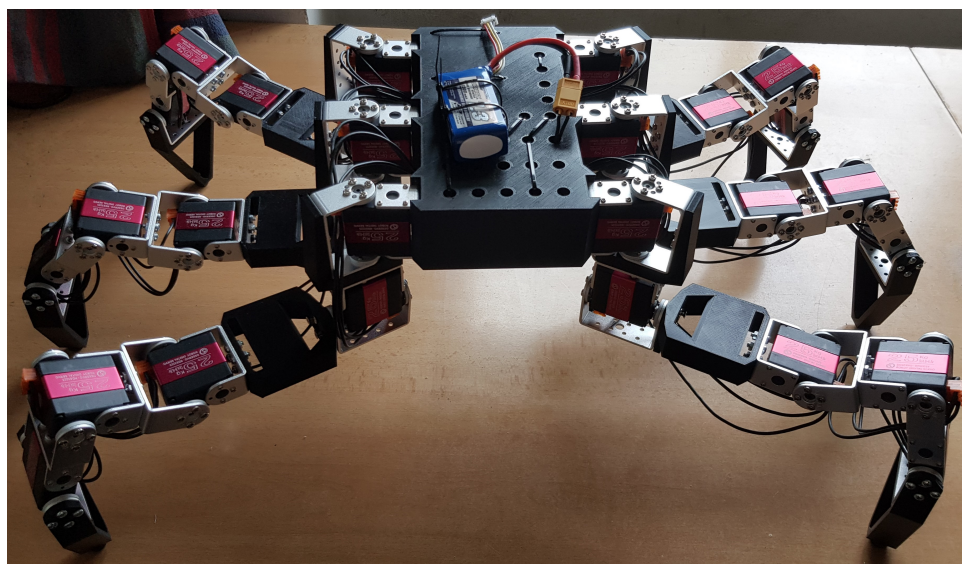
Běží na něm operační systém Linux, což může občas některé věci zkomplikovat. Jelikož byl UART implementován jako 1-drátová verze, je potřeba přepínat mezi vysíláním a přijímáním, kdy ale operační systém nedovolí přímé přistoupení do registrů, proto se musí využít driver. Než se vykoná nutný kód a změní se hodnota v příslušném registru uběhne 20 až 30 ms.

Na druhou stranu operační systém nabízí možnost paralelismu a je možné využít existující ovladače pro WiFi/Bluetooth.

Kapitola 4

Mechanické díly

Na následujícím obrázku je vyfocen výsledný hexapod, aby bylo zřetelnější, kde budou navržené díly upevněny.

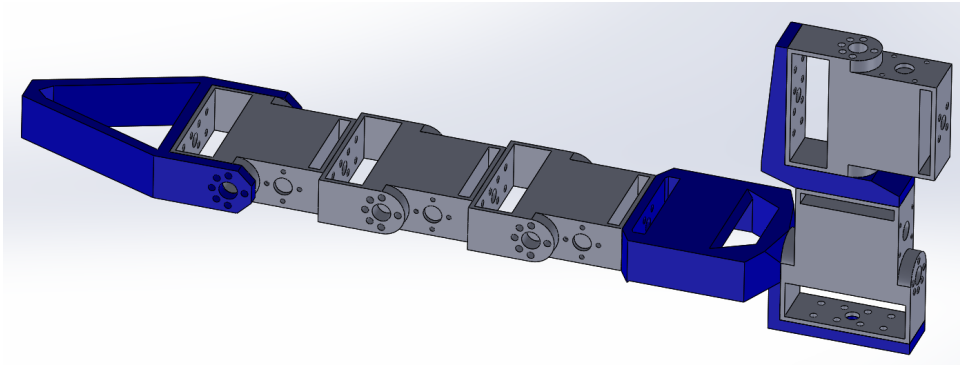


Obrázek 4.1: Hotový hexapod

Mechanická část konstrukce hexapoda se skládá ze dvou částí. V první části se zaměřuje na díly v nohách a v druhé se navrhne a vytvoří tělo.

4.1 Noha

Je několik možností, jak by výsledná noha mohla vypadat. Vhodná stavba nohy je například na následujícím obrázku.



Obrázek 4.2: Sestavený model nohy
Pozn.: Modře jsou vyznačené navrhnuté plastové díly

K servům se dodávají hliníkové držáky, které se pro každé servo použijí a navíc se navrhnou a vytvoří ještě tři další díly.



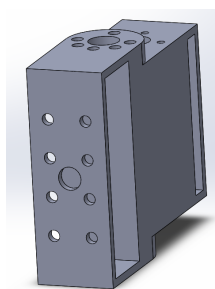
Obrázek 4.3: Dodávané hliníkové držáky [Alupg]

■ 4.1.1 Návrh nových dílů

Nové díly jsou vyrobeny z plastu na 3D tiskárně. Je to jednoduchý způsob, jak si vytvořit požadované díly. Přičemž jejich pevnost je dostatečná pro danou velikost a váhu hexapoda.

K návrhu byl použit program Solidworks, který poskytuje dostatečně mocný nástroj pro tyto účely. Samozřejmě by šlo použít i jiné programy, jako jsou Fusion 360, SketchUp, OnShape a jiné.

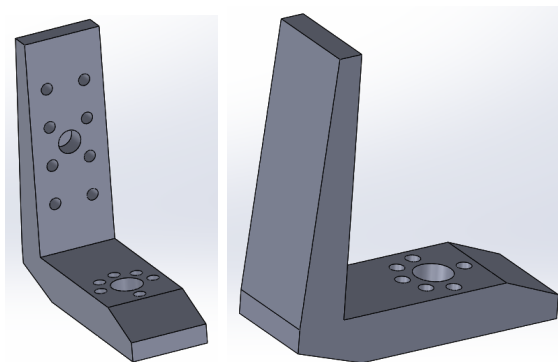
Pro jednodušší modelování součástek byl nejdříve vytvořen přesný model serva i s dodávanými držáky. To umožní jednotlivé součástky snadněji modelovat a dokonce si i v programu samotnou nohu sestavit.



Obrázek 4.4: Vytvořený model serva

■ První díl

První díl byl přidán kvůli lepšímu přichycení serv, neboť jejich natočení způsobí, že k sobě mohou být přišroubované pouze přes 2 šrouby.



Obrázek 4.5: První vymodelovaný díl

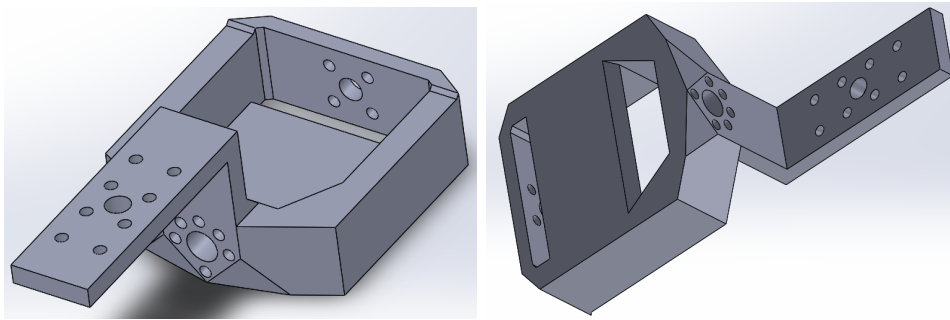
Díl posouvá vzdálenost mezi dvěma servy o 1 cm, přidává možnost úchytu z předu a navíc tyto šroubky schová, takže při pohledu nebudou nevzhledně vykukovat. Různé zkosení je spíše z estetických důvodů.

Účel lepšího přichycení by se lépe naplnil, pokud by tento díl obejmul menší hliníkový držák na druhém servu. Toto řešení se po experimentech ukázalo jako nevhodné, neboť na soukolí serva vyvíjí příliš vysoký tlak a tím zamezí jeho otáčení.

■ Druhý díl

Druhý díl byl přidán ze stejného důvodu jako ten první, neboť by také serva byla přichycená jen přes 2 šrouby. Navíc tímto dílem se dá noha trochu prodloužit.

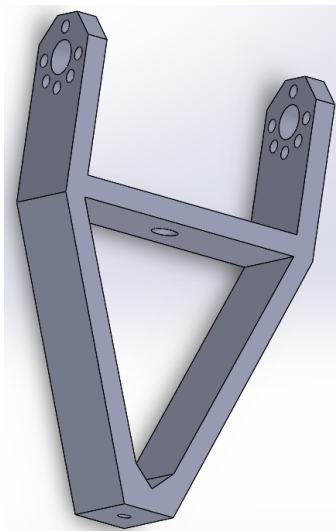
Díl kromě zmíněných účelů umožní i částečně schovat kabely, případně možnost je k tomuto dílu přichytit, bude-li to potřeba.



Obrázek 4.6: Druhý vymodelovaný díl

■ Zakončení nohy

Poslední díl musí nést jistou část váhy hexapoda, takže musí jeho design zajišťovat dostatečnou odolnost a zároveň by svým vzhledem neměl narušovat celkový dojem.



Obrázek 4.7: Vymodelované zakončení nohy

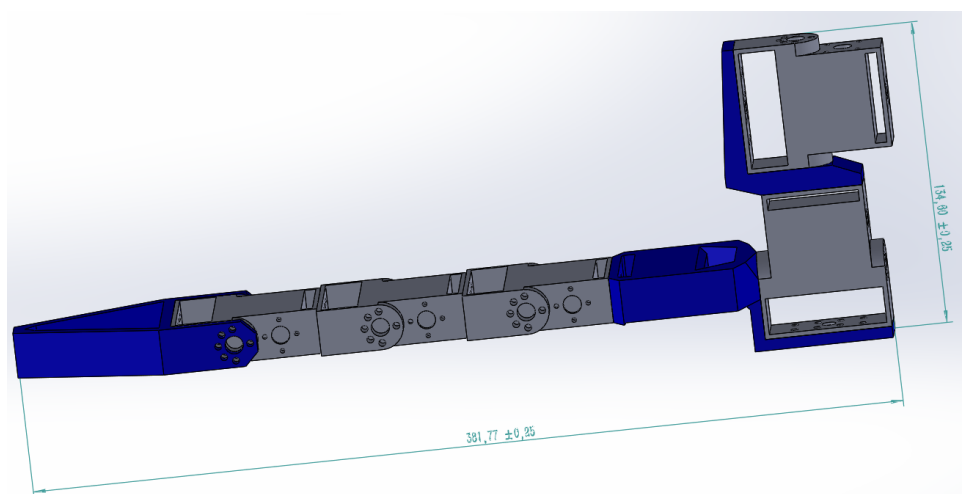
Díl dobře dolehne na hliníkový držák, čímž zvýší svou pevnost a rozprostření sil. Otvor uprostřed zajistí v případě přidání nějakého senzoru možnost vedení drátů. Otvor dole lze využít i na přimontování gumového zakončení nohy, jak je ukázáno na následujícím obrázku.



Obrázek 4.8: Díl s přimontovaným gumovým zakončením

4.1.2 Sestavení nohy v modelovacím programu

Jak bylo zmíněno dříve, díky vlastnímu modelu serva si lze celou nohu v modelovacím programu sestavit a snadněji tak modelovat nové součástky.

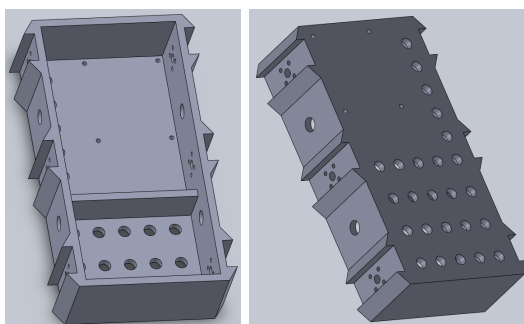


Obrázek 4.9: Sestavený model nohy

Pozn.: Modře jsou vyznačené navrhnuté plastové díly.
Hodnoty u kót jsou v mm.

4.2 Tělo

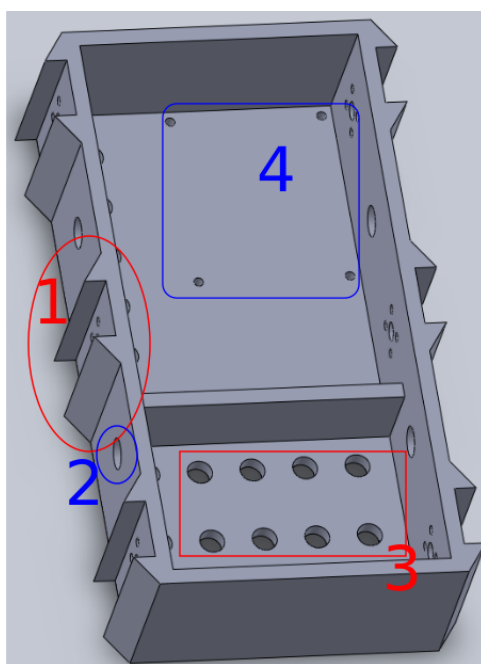
Tělo hexapoda slouží pro přichycení nohou, řídicí jednotky, zdroje s elektronikou a kabeláže. Zároveň by mělo poskytnout i místo navíc pro případnou další elektroniku nebo náklad.



Obrázek 4.10: Návrh těla

Výčnělky u míst, kde se připevní serva (viz 1 na obrázku 4.11), zajistí pevné uchycení. Otvory mezi nimi jsou pro kabeláž (viz 2). Větší otvory (viz 3) mohou být využity stejně nebo k připevnění elektroniky. Je zde i místo pro řídicí jednotku (viz 4), kde jsou udělány 4 menší otvory, které kopírují otvory na řídicí desce.

Uvnitř je malá přepážka, která slouží pouze pro větší zpevnění těla. Rozměry těla jsou 20x10 cm.



Obrázek 4.11: Tělo s vyznačenými částmi

Kapitola 5

Konstrukce

5.1 Potřebný spojovací materiál a další elektronika

Před samotným sestavením je potřeba navrhnout ještě zapojení napájení se servy a řídicí deskou. Jako zdroj byla využita **Li-Po baterie 3S** s kapacitou 1300 mAh, jejíž napětí je 11,1 V, proto bude potřeba mezi baterii a elektroniku zařadit step-down měnič.

Využijí dva měniče. Pro řídicí jednotku stačí mini step-down měnič P2H02890A s výstupním proudem 2 A (max 3 A), který sníží napětí na 5 V, a pro serva se využije měnič XL4016 s žebrováním schopný dodávat větší proudy, jelikož zvládne výkon až 280 W. Sníží napětí na 6 V.

Dále bude potřeba spojovací materiál, jako jsou různé šrouby a dráty.

5.2 Sestavení

Noha se sestaví přesně dle návrhu z obrázku 4.2 a 4.1.

Při kabeláži je vhodné nohu rozdělit na dvě části, které se zapojí zvlášť. Kvůli své topologii lze spojit spolu první dvě serva (obr. 5.1) a pak poslední 3 k sobě (obr. 5.2). Rozdělení na 2 části sníží proud, který musí projít jedním kabelem, což zlepší vlastnosti hexapoda zejména při vyšších zátěžích.

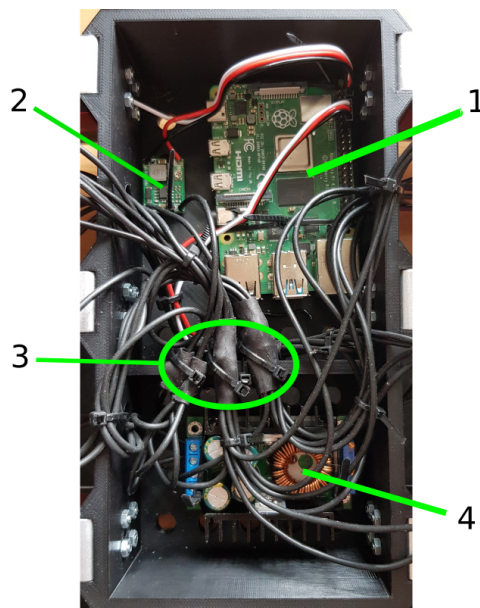


Obrázek 5.1: Smontovaná noha se zapojenými kabely - první část



Obrázek 5.2: Smontovaná noha se zapojenými kabely - druhá část

Možné uložení elektroniky a způsob kabeláže ukazuje následující obrázek.

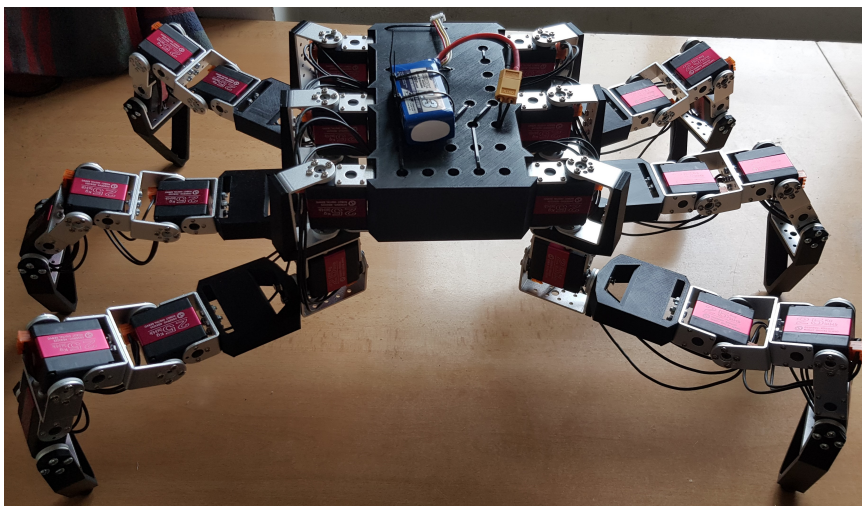


Obrázek 5.3: Hotový hexapod - ukázka uložení elektroniky a kabelů

Vpravo nahoře (1) je řídicí deska přichycená stahovacími pásky. Je spojena pomocí kabelů ze serv s konektorem k mini step-down měniči vlevo (2) a k UARTu od serv.

Dole (4) je větší step-down měnič s žebrováním, který dodává napájení servům. Uprostřed (3) jsou přichyceny spojení kabelů, které byly překryty smršťovacími bužírkami kvůli ochraně.

Samotná baterie je umístěna na vrh hexapoda, což usnadní případné zapojení/vypojení. Lze samozřejmě umístit i zespu.

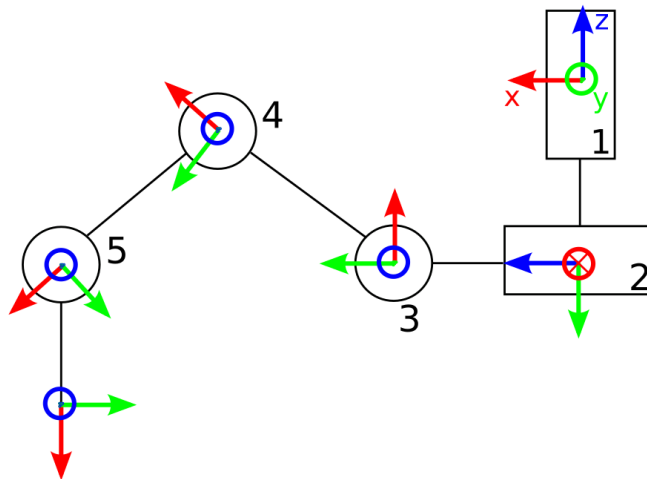


Obrázek 5.4: Hotový hexapod

Kapitola 6

Ovládání hexapoda

Pro ovládání hexapoda bude potřeba hned několik algoritmů, aby byl zajištěn správný pohyb. Jelikož všechny algoritmy pracují se souřadnicemi, zavedu několik souřadných soustav. Nejdříve pro rameno hexapoda, kde první servo je označeno číslovkou jedna a poslední (páté) servo číslovkou pět.

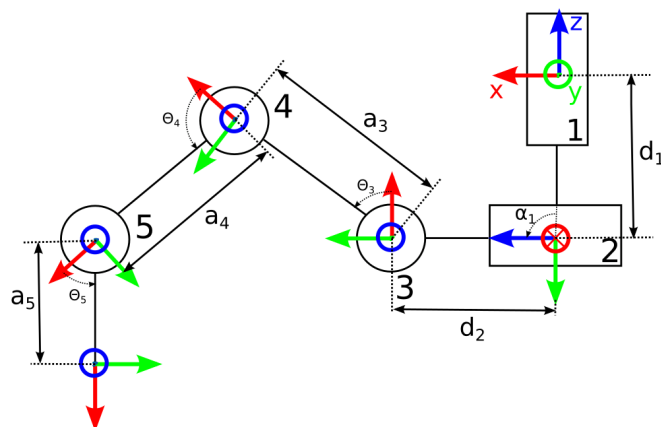


Obrázek 6.1: Zavedení souřadných soustav

Souřadnicové soustavy byly zvoleny tak, aby dle nich šlo kinematický řetězec popsat pomocí Denavit–Hartenbergovy notace. V DH konvenci je každá homogenní transformace A_i reprezentovaná součinem dvou základních rotací a dvou translací

$$A_i = \text{Rot}_z(\Theta_i)\text{Trans}_z(d_i)\text{Trans}_x(a_i)\text{Rot}_x(\alpha_i), \quad (6.1)$$

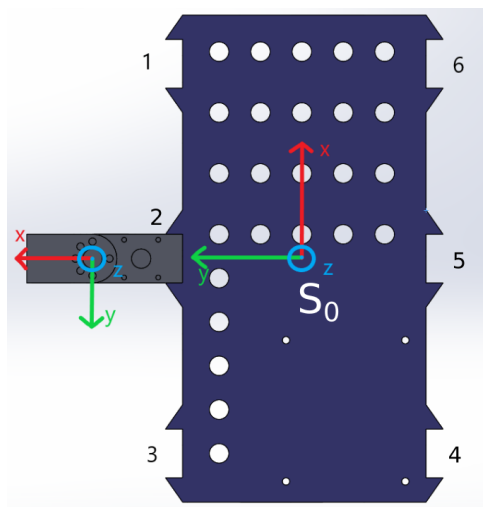
kde Θ_i , d_i , a_i a α_i jsou čtyři parametry článku i a kloubu i [aMV3X]. Pro rotační kloub je Θ_i proměnnou, ostatní parametry jsou konstanty. Zobrazení některých parametrů, je na následujícím obrázku.



Obrázek 6.2: Zobrazení některých DH parametrů

Počátky souřadných systému se vždy nachází uvnitř serv a to tak, že **osa z je identická s osou otáčení** daného serva a počátek je přesně ve středu serva.

Jelikož nohou bude 6, zvolím ještě jednu souřadnicovou soustavu (S_0), která bude sloužit jako hlavní. Tedy například v ní se budou počítat koncové body nohou. V některých částech textu na ni bude referováno jako soustava číslo 0.



Obrázek 6.3: Zavedení souřadné soustavy pro tělo

Na obrázku je pohled z vrchu na tělo hexapoda. Počátek souřadnicové soustavy těla je umístěn ve středu těla, výškově (tedy podle osy z) je ve stejné výšce jako souřadnicové systémy prvních serv všech nohou (Na obrázku bylo naznačeno pro nohu 2).

Číslice označují číslování jednotlivých nohou.

Dopředná kinematika, tedy zjištění umístění koncového bodu nohy, je jen násobení homogenních transformací pro jednotlivé klouby a články, proto ji zde nebudu nijak vypisovat a popíšu rovnou inverzní kinematiku.

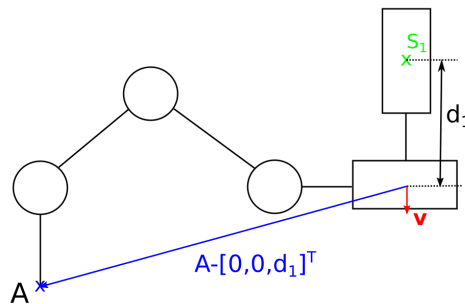
6.1 Inverzní kinematika

Inverzní kinematika slouží k výpočtu úhlů (Θ_i) jednotlivých serv v případě, kdy je známo umístění konce nohy.

Mějme dán koncový bod nohy \mathbf{A}_1 a vektor \mathbf{v}_1 , který reprezentuje osu x na konci nohy na obrázku 6.1. Vektor \mathbf{v}_1 tedy říká, jakým směrem má směřovat zakončení nohy. Index 1 znamená, že jsou zadány v souřadnicové soustavě kloubu/serva číslo 1. Navíc, nebude-li řečeno jinak, všechny **výpočty** budou probíhat **v soustavě 1. serva**, proto to již dále nebude explicitně značeno.

V rovnicích se využije DH notace (d_i , Θ_i , a_i a α_i), kde index bude značit, ke kterému kloubu patří, takže například 1 je první kloub (viz obr. 6.2).

Natočení prvních dvou kloubů nastavuje rovinu, kterou tvoří vektor \mathbf{v} a $\mathbf{A}-[0\ 0\ d_1]^T$.



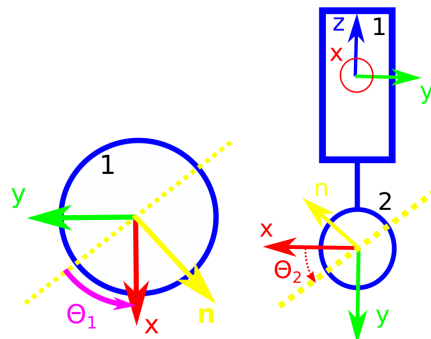
Obrázek 6.4: Vyznačení vektorů potřebných k výpočtu

Spočte se normálový vektor této roviny

$$\mathbf{n} = \mathbf{v} \times (\mathbf{A} + [0\ 0\ -d_1]^T). \quad (6.2)$$

Normálový vektor se ještě znormalizuje

$$\mathbf{n}^0 = \frac{\mathbf{n}}{|\mathbf{n}|}. \quad (6.3)$$



Obrázek 6.5: Znázornění normálového vektoru a hledaných úhlů

Úhly pro první dvě serva se získají pomocí goniometrických funkcí z normovaného normálového vektoru

$$\Theta_1 = -\text{atan2}(n_x^0, -n_y^0), \quad (6.4)$$

$$\Theta_2 = \arcsin(n_z^0). \quad (6.5)$$

Kde například n_x^0 znamená x-souřadnice normovaného vektoru n .

Pro výpočet Θ_3 je potřeba spočítat pozice 3. (\mathbf{p}_3) a 5. serva (\mathbf{p}_5). K tomu se vytvoří ortonormální báze - \mathbf{n}^0 , \mathbf{u}_1 a \mathbf{u}_2 , kde \mathbf{u}_1 je osa x prvního serva po otočení o úhel Θ_1 (viz obr. 6.5) a \mathbf{u}_2 je zbylý kolmý vektor, který z předchozích dvou lze dopočítat

$$\mathbf{u}_1 = [\cos(\Theta_1) \quad \sin(\Theta_1) \quad 0]^T, \quad (6.6)$$

$$\mathbf{u}_2 = \mathbf{n}^0 \times \mathbf{u}_1. \quad (6.7)$$

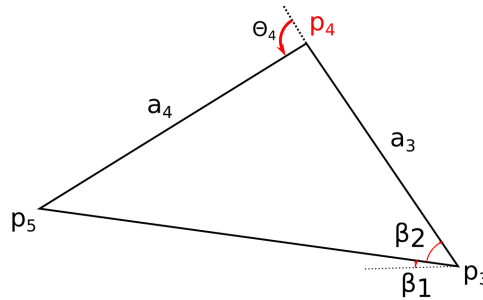
Z obrázku 6.2 lze vidět případ, kdy $\Theta_1 = 0$, pak vektor \mathbf{u}_1 je identický s osou x prvního serva na obrázku. Je potřeba onen vektor zvětšit o d_2 a posunout o d_1

$$\mathbf{p}_3 = d_2 \mathbf{u}_1 - [0 \quad 0 \quad -d_1]^T. \quad (6.8)$$

Pozice serva 5 je bod \mathbf{A} posunutý o zadaný vektor \mathbf{v}

$$\mathbf{p}_5 = \mathbf{A} - a_5 \mathbf{v}^0. \quad (6.9)$$

Z vypočtených pozic lze dopočítat pomocné úhly, z kterých jejich sečtením vznikne Θ_3 . K výpočtu se využije kosínová věta a goniometrická funkce atan2 .



Obrázek 6.6: Znázornění pomocných úhlů (p_4 není známo)

$$\beta_2 = \arccos\left(\frac{a_4^2 - a_3^2 - \|\mathbf{p}_3 \mathbf{p}_5\|^2}{-2a_3 \|\mathbf{p}_3 \mathbf{p}_5\|}\right), \quad (6.10)$$

$$\beta_1 = \text{atan2}((\mathbf{p}_5 - \mathbf{p}_3)^T \mathbf{u}_2, (\mathbf{p}_5 - \mathbf{p}_3)^T \mathbf{u}_1), \quad (6.11)$$

$$\Theta_3 = \beta_1 + \beta_2. \quad (6.12)$$

Z pozic 3. a 5. serva, respektive z jejich vzdálenosti, lze pomocí kosínové věty zjistit úhel natočení 4. serva. Jsou možné 2 varianty. Zvolí se ta varianta,

kde ve výsledku bude z-souřadnice pozice 4. serva větší rovna z-souřadnici 3. serva.

$$\Theta_4 = \arccos\left(\frac{\|\mathbf{P}_3\mathbf{P}_5\|^2 - a_3^2 - a_4^2}{-2a_3a_4}\right) - \pi. \quad (6.13)$$

K dopočtení posledního úhlu se spočte pomocný úhel β_3 mezi vektory \mathbf{v} a \mathbf{u}_1

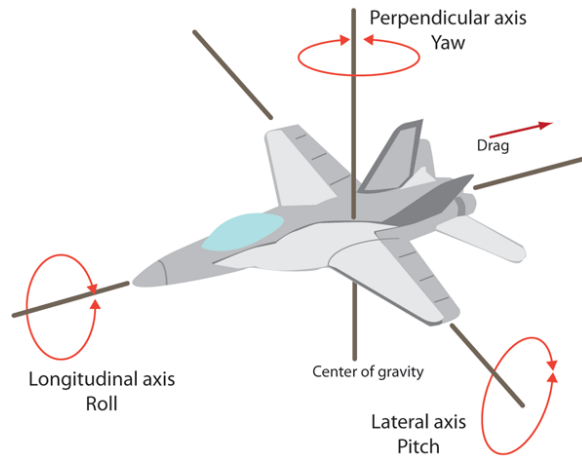
$$\beta_3 = \text{atan2}(\mathbf{u}_2^T \mathbf{v}, \mathbf{u}_1^T \mathbf{v}), \quad (6.14)$$

$$\Theta_5 = \beta_3 - \Theta_3 - \Theta_4. \quad (6.15)$$

Inverzní kinematika je hotová a lze ji využít během práce dalších algoritmů.

6.2 Posunutí a rotace těla hexapoda

K popisu rotace těla se využije eulerových úhlů pojmenovaných jako yaw(Ψ), pitch(Θ) a roll(Φ).



Obrázek 6.7: Znáznornění úhlů rotace [Mraaw]

Zavede se transformační matice využívající popsané úhly.

$$\mathbf{Y} = \begin{bmatrix} \cos(\Psi) & \sin(\Psi) & 0 \\ -\sin(\Psi) & \cos(\Psi) & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad \mathbf{P} = \begin{bmatrix} \cos(\Theta) & 0 & -\sin(\Theta) \\ 0 & 1 & 0 \\ \sin(\Theta) & 0 & \cos(\Theta) \end{bmatrix} \quad (6.16)$$

$$\mathbf{R} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\Phi) & \sin(\Phi) \\ 0 & -\sin(\Phi) & \cos(\Phi) \end{bmatrix} \quad (6.17)$$

Mejme matici \mathbf{L} velikosti 3×6 , kde první sloupec odpovídá koncovému bodu první nohy, druhý sloupec odpovídá koncovému bodu druhé nohy atd. Koncové body jsou uvedeny v souřadnicové soustavě S_0 .

Využitím transformačních matic na matici \mathbf{L} se získá nové rozmístění nohou, které způsobí požadované natočení těla

$$\mathbf{T} = \mathbf{Y} \cdot \mathbf{P} \cdot \mathbf{R}, \quad (6.18)$$

$$\mathbf{L}_n = \mathbf{T} \cdot \mathbf{L}. \quad (6.19)$$

Posunutí lze dosáhnout přímým přičtením hodnoty k pozici nohou.

Z nově vypočtených pozic nohou lze pomocí inverzní kinematiky dopočítat požadované natočení serv.

6.3 Chůze

Existují tři hlavní způsoby chůze pro hexapoda: Tripod Gait, Ripple Gait a Wave gait. Každý způsob chůze by měl zajistit, že alespoň 3 nohy vždy zůstanou na zemi kvůli stabilitě [DarhD].

V této práci se budu zabývat pouze variantou Tripod Gait.

Algoritmus lze implementovat pomocí neuronových sítí a založit tak jeho fungování například na chůzi mravenců [aFM36]. Ovšem zde bude využit analytický přístup.

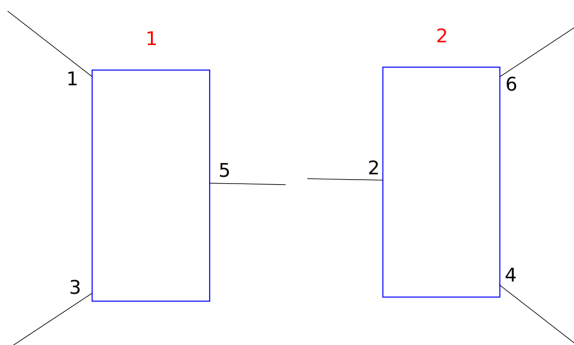
Trajektorie nohy v rovině XZ soustavy S_0 by vypadala jako na obrázku níže.

Během černé části je noha v kontaktu se zemí. V modré části je zvednutá.



Obrázek 6.8: Cesta vykonaná jednou nohou v rovině XZ

Jelikož se jedná o variantu chůze Tripod, nohy se rozdělí do dvou sad, které se vůči sobě budou pohybovat s posunutou fází. Tím se zajistí, že 3 nohy ponesou váhu hexapoda, zatímco zbylé 3 nohy budou ve vzduchu.



Obrázek 6.9: Rozdělení nohou na dvě sady

Pro každou nohu je definována její výchozí pozice v matici \mathbf{L} (viz sekce 6.2) tak, aby hexapod stál ve stejné pozici jako na obrázku 4.1.

V algoritmu se dále využijí tyto proměnné

Název	Vysvětlení
T	Doba jednoho kroku
T_e	Doba zvednutí/položení nohy
T_s	Krok výpočtu
t_l	Zbývající čas do přepnutí na další část chůze
v_c	Současná rychlost dané sady nohou
a_c	Současné zrychlení dané sady nohou
z_d	O kolik se zvedne noha při kroku (v m)
\mathbf{L}_s	matice 3x6, kde každý sloupec je koncový bod 1 nohy v danou chvíli
v_x	cílová rychlost v ose x - definuje uživatel při řízení

Tabulka 6.1: Využívané proměnné v algoritmu

Pohyb je možné rozložit do os x, y a z.

6.3.1 Chůze v osách x a y

Trajektorii pohybu v ose x a y je možné generovat na základě stejného algoritmu.

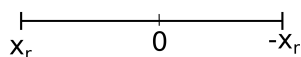
Každá sada nohou si ponese informaci v jaké fázi se nachází. Fáze je z intervalu $\langle 0; T \rangle$. Začátek je $t_1 = 0$ a $t_2 = \frac{T}{2}$.

Na zemi

Tato část algoritmu je určena pro sadu nohou, která má fázi v rozmezí 0 až $\frac{T}{2}$. Mějme proměnnou i , kdy $i = 1$ $t_1 < \frac{T}{2}$ jinak $i = 2$.

Spočte se relativní poloha p_r v ose x. Dále pak pozice x_r , do které by se noha měla dostat v dané etapě. Pro výpočet x_r se využije vzorec na výpočet dráhy při rovnoměrném pohybu ($s = vt$).

V případě, že je relativní poloha $p_r = 0$ a noha je na zemi, tak to znamená, že noha je ve výchozí pozici.



Obrázek 6.10: Relativní poloha jedné nohy

$$p_r = \mathbf{L}_s[1, i] - \mathbf{L}[1, i], \quad (6.20)$$

$$x_r = \frac{v_x \frac{T}{2}}{2}, \quad (6.21)$$

kde $\mathbf{L}[1, i]$ znamená prvek matice \mathbf{L} na řádce 1 ve sloupci i .

Vypočte se potřebné zrychlení, kterým by se noha za zbývající čas dostala

do požadované polohy. Všimněte si, že v této etapě se používá vzorec na rovnoměrně zrychlený pohyb a jeho různé úpravy.

$$t_l = \frac{T}{2} - t_1, \quad (6.22)$$

$$a(t_l) = 2\left(\frac{x_r - p_r}{t_l^2} - \frac{v_c}{t_l}\right). \quad (6.23)$$

Pokud je zrychlení vyšší než maximální a_m , které se stanoví dle možností serv ($1.5 \text{ m} \cdot \text{s}^{-2}$), tak místo a se použije a_m . Dále se vypočte, o kolik se nohy posunou během dalšího kroku

$$x = \frac{1}{2}aT_s^2 + v_cT_s. \quad (6.24)$$

Aktualizuje se současná rychlost a pozice

$$v_c = v_c + aT_s, \quad (6.25)$$

$$\mathbf{L}_s = \mathbf{L}_s + \begin{bmatrix} x & 0 & x & 0 & x & 0 \\ & & \mathbf{0} & & & \end{bmatrix} \quad i == 1, \quad (6.26)$$

$$\mathbf{L}_s = \mathbf{L}_s + \begin{bmatrix} 0 & x & 0 & x & 0 & x \\ & & \mathbf{0} & & & \end{bmatrix} \quad i == 2. \quad (6.27)$$

■ Ve vzduchu

Trajektorie, kterou má sada nohou vykonat, se aproximuje polynomem 5. stupně. 5. stupeň zajistí, že nebude během pohybu docházet ke skokovým změnám zrychlení, čímž se zajistí plynulý pohyb[Woeft].

Nejdříve se zjistí, zda se s maximálním možným zrychlením mohou nohy dostat do pozice $x_d = -x_r$. Pokud ano, pak $v = v_x$, jinak $v = \frac{2(x_r - x_m)}{T}$ a $x_d = x_m$, kde

$$x_m(t_l) = \pm \frac{1}{2}a_m t_l^2 + v_c t_l + p_r. \quad (6.28)$$

Pro výpočet p_r viz (6.20). (Znaménko $x_m(t_l)$ je dle strany, na kterou se má noha vydat - viz obrázek 6.10)

Vypočtou se konstanty polynomu 5. stupně

$$c_0 = p_r, \quad (6.29)$$

$$c_1 = v_c, \quad (6.30)$$

$$c_2 = \frac{a_c}{2}, \quad (6.31)$$

$$c_5 = \frac{1}{t_l^5} \left(6(x_d - c_0 - \frac{t_l(c_1 + v)}{2}) - c_2 t_l^2 \right), \quad (6.32)$$

$$c_4 = \frac{1}{2t_l^3} (c_1 - v - 5c_5 t_l^4 + c_2 t_l), \quad (6.33)$$

$$c_3 = -2c_4 t_l - \frac{10}{3} c_5 t_l^2 - \frac{c_2}{3t_l}. \quad (6.34)$$

Pomocí vypočtených konstant se dopočte požadovaná poloha nohou pro daný čas a aktualizují se současná rychlost a zrychlení

$$x = c_0 + c_1 T_s + c_2 T_s^2 + c_3 T_s^3 + c_4 T_s^4 + c_5 T_s^5, \quad (6.35)$$

$$v_c = c_1 + 2c_2 T_s + 3c_3 T_s^2 + 4c_4 T_s^3 + 5c_5 T_s^4, \quad (6.36)$$

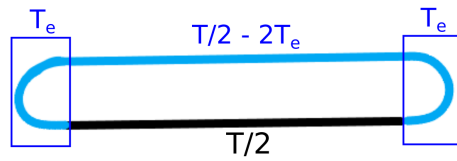
$$a_c = 2c_2 + 6c_3 T_s + 12c_4 T_s^2 + 20c_5 T_s^3, \quad (6.37)$$

$$\mathbf{L}_s = \mathbf{L}_s + \begin{bmatrix} x & 0 & x & 0 & x & 0 \\ & & \mathbf{0} & & & \end{bmatrix} \quad i == 1, \quad (6.38)$$

$$\mathbf{L}_s = \mathbf{L}_s + \begin{bmatrix} 0 & x & 0 & x & 0 & x \\ & & \mathbf{0} & & & \end{bmatrix} \quad i == 2. \quad (6.39)$$

Je důležité, aby počítání konstant polynomu probíhalo v každém kroku, neboť tak lze docílit správné rychlosti s minimálním zpožděním při změnách během řízení.

6.3.2 Trajektorie v ose z



Obrázek 6.11: Doba trvání jednotlivých částí trajektorie

Trajektorie v ose z při zvedání a pokládání nohy (na obrázku v modrých rámečkách) se opět proloží polynomem pátého stupně. Tentokrát ho ovšem není potřeba počítat pokaždé, neboť zde se požadovaná výška ani rychlost nebude během řízení měnit.

Pokud tedy nohy nejsou ve fázích zvedání a pokládání (trvá po dobu T_e), pak je $z = 0$ nebo $z = z_d$ dle toho, jestli je noha ve vzduchu nebo na zemi.

Konstanty polynomu se vypočtou následovně

$$c_5 = \frac{6z_d}{T_e^5}, \quad (6.40)$$

$$c_4 = -\frac{5}{2}c_2 T_e, \quad (6.41)$$

$$c_3 = -2c_1 T_e - \frac{10}{3}c_2 T_e^2. \quad (6.42)$$

Ostatní konstanty jsou nulové, neboť se předpokládá na začátku i na konci nulové zrychlení i rychlost.

Cílová pozice pro z je

$$z = c_3 t^3 + c_4 t^4 + c_5 t^5. \quad (6.43)$$

$$\mathbf{L}_s = \mathbf{L}_s + \begin{bmatrix} \mathbf{0} \\ z & 0 & z & 0 & z & 0 \end{bmatrix} \quad i == 1, \quad (6.44)$$

$$\mathbf{L}_s = \mathbf{L}_s + \begin{bmatrix} \mathbf{0} \\ 0 & z & 0 & z & 0 & z \end{bmatrix} \quad i == 2, \quad (6.45)$$

kde t je, pro zvedání nohy, doba, jak dlouho nohu již zvedá, a pro pokládání nohy zbývající doba pokládání nohy.

6.4 Ovládání přes aplikaci na PC pomocí WiFi

Vytvoří se jednoduchá konzolová aplikace, která pomocí TCP protokolu vysílá data do řídicí jednotky hexapoda.

Pomocí kláves lze ovládat natočení hexapoda, neboť se nastavují přímo Eulerovy úhly a výška, ve které bude mít hexapod tělo.

Nejdůležitější je ovládání směru a rychlosti pohybu hexapoda. To je uděláno tak, že se po navázání spojení s hexapodem nastaví myš do středu obrazovky. Směr a velikost rychlosti udává vektor s počátkem ve středu obrazovky a koncem v ukazateli myši.

V aplikaci není nijak kritické, aby byly informace posílány v přesných časových intervalech a ani příliš nezáleží na délce těchto intervalů. Je jen potřeba, aby data byla odeslána alespoň dvakrát mezi jednotlivými výpočty nových pozic nohou pomocí algoritmu na chůzi, neboť to sníží latenci při ovládání a uživatel tak nebude mít pocit, že zaznamenává zpoždění mezi svým řízením a reakcí hexapoda. Ve svém provedení jsem zvolil vysílání přibližně jednou za 20 ms.

Zatímco celý hexapod je programován v jazyce C, pro tuto aplikaci z důvodu pohodlí byl zvolen jazyk C#.

```
Connection was successfully established.
Height: 70 Yaw: 000 Pitch: 000 Roll: 000 SpeedX: 0,0356 SpeedY: 0,0060_
```

Obrázek 6.12: Okno aplikace

Kapitola 7

Evoluce

Stejně jako živá bytost se musí vyvíjet, aby přežila, je potřeba dále zlepšovat a přetvářet již vyrobené. Což dává i další možnosti na navázání na mou práci.

První vylepšení je zjevné. Některé aplikace můžou vyžadovat poměrně vysokou **preciznost pohybu**, kterou sestavený hexapod není schopen dodat. Pro její zlepšení se nabízí tři cesty.

Zpřesnit data, kterými se kalibrují serva například pomocí vylepšení metody, kterou se měří skutečný úhel, ve kterém je servo natočeno.

Zvolit vhodnější interpolaci dat než je kubický spline, čímž se servo více přiblíží k poloze, která skutečně odpovídá cílenému natočení.

Zmíněné cesty však nakonec dříve či později narazí na možnosti senzoru polohy, kterým je potenciometr. Proto poslední cestou je **vyměnit potenciometr** za vhodný enkodér (např. magnetický).

Jedno zlepšení bylo zmíněno v kapitole 6.3 o **chůzi**. Je možné natrénovat neuronovou síť, která by spravovala samotnou chůzi, řídila, kolik nohou se bude naráz zvedat v závislosti ať už na rychlosti či terénu.

Další vylepšení již závisí na senzorech, kterými se hexapod osadí a na jeho vlastní velikosti.

První **senzory** by šly pravděpodobně na nohy, čímž by se zajistilo, že by měl hexapod informaci o tom, zda již **došlápl na zem**.

Další senzory, kterými by se pravděpodobně osadil jsou lidar, termokamera, případně i FPV kamera. Čímž hexapod získá potřebné informace o svém okolí a lze ho již programovat pro autonomní pohyb, nebo v případě FPV kamery může operátor například hledat lidi v troskách při záchranných misích.



Kapitola 8

Závěr

Navrhl jsem mechanickou konstrukci, která se skládala z dílů dodávaných přímo s vybranými servy a z výtisků z 3D tiskárny.

Vybraná serva neměla vhodné ovládání a nedala se zapojovat do řetězců, což se vyřešilo nahrazením původní elektroniky v servech. Tento krok v sobě zahrnoval návrh a realizaci plošného spoje i s jeho osazením. Dále pak návrh regulátoru pro ovládaní serv, jejich kalibrace a v neposlední řadě i vytvoření protokolu pro komunikaci s řídicí jednotkou.

Vytvořil jsem algoritmus, který obstarává chůzi, jehož principem je proložení požadované cesty polynomem. Byla spočítána inverzní kinematika ramen a přidal jsem i možnost natáčet tělo hexapoda, které je implementováno pomocí Eulerových úhlů.

Aplikováním zmíněného se podařilo navrhnout a zrealizovat hexapoda, jehož ramena mají 5 stupňů volnosti.

Jelikož byl hexapod udělán s ovládaním přes WiFi, vytvořil jsem jednoduchou konzolovou aplikaci na PC, která řídí hexapoda pomocí tlačítek a myši.



Literatura

- [aFM36] Q. Chang a F. Mei, *A bioinspired gait transition model for a hexapod robot*, *Journal of Robotics* [online]. 2018, 2018, 1-11 [cit. 2021-6-20]. ISSN 1687-9600. Dostupné z: doi:10.1155/2018/2913636.
- [Alupg] Holder Aluminum, *Aliexpress*, [online]. Čína: ShenZhen Easy to Robot Store, 2010 [cit. 2021-7-11]. Dostupné z: https://ae01.alicdn.com/kf/HTB14W_oNmrqK1RjSZK9q6xyypXaL.jpg.
- [aMV3X] M. W. Spong a M. Vidyasagar, *Robot dynamics and control*, USA: John Wiley & Sons, 1989, ISBN 10-47161243X.
- [Bb3] Raspberry Pi 4 Model B, *Alza*, [online], [cit. 2021-7-11], Dostupné z: <https://cdn.alza.cz/ImgW.ashx?fd=f16&cd=RK100b3>.
- [DarhD] N. H. Darbha, *An optimization strategy for hexapod gait transition*, [online]. Indie, 2017 [cit. 2021-8-8]. Dostupné z: https://corescholar.libraries.wright.edu/etd_all/1865/?utm_source=corescholar.libraries.wright.edu%2Fetd_all%2F1865&utm_medium=PDF&utm_campaign=PDFCoverPages. Diplomová práce. Wright State University. Vedoucí práce Luther R. Palmer III, Ph.D.
- [Mraaw] S. Mraz, *What's the difference between pitch, roll, and yaw?*, *MachineDesign* [online]. 12. 6. 2014 [cit. 2021-6-20]. Dostupné z: <https://www.machinedesign.com/learning-resources/engineering-essentials/article/21834526/whats-the-difference-between-pitch-roll-and-yaw>.
- [Serpj] RDS3225 Servo, *Aliexpress*, [online]. Čína: ShenZhen Easy to Robot Store, 2010 [cit. 2021-4-29]. Dostupné z: <https://ae01.alicdn.com/kf/HTB1Hpj0NhnaK1RjSZFtq6zC2VXaW.jpg>.
- [Woeft] R. Woering, *Simulating the "first steps" of a walking hexapod robot*, [online]. Eindhoven, 2011 [cit. 2021-8-8]. Dostupné z:

<http://www.mate.tue.nl/mate/pdfs/12670.pdf>. Diplomová práce.
Technische Universiteit Eindhoven. Vedoucí práce Ir. M.J.C. Ronde
a Dr. ir. M.J.G. van de Molengraft.