



TECHNICKÁ UNIVERZITA V LIBERCI
Fakulta mechatroniky, informatiky
a mezioborových studií ■

Návrh řídicího systému pro Hexapod s Raspberry Pi

Diplomová práce

Studijní program: N2612 – Elektrotechnika a informatika

Studijní obor: 1802T007 – Informační technologie

Autor práce: **Bc. Jakub Hirňal**

Vedoucí práce: Ing. Miroslav Holada, Ph.D.





TECHNICAL UNIVERSITY OF LIBEREC
Faculty of Mechatronics, Informatics
and Interdisciplinary Studies ■

Control system design for Hexapod with Raspberry Pi

Master thesis

Study programme: N2612 – Electrical Engineering and Informatics

Study branch: 1802T007 – Information Technology

Author: **Bc. Jakub Hirňal**

Supervisor: Ing. Miroslav Holada, Ph.D.



ZADÁNÍ DIPLOMOVÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Bc. Jakub Hirnšal**
Osobní číslo: **M15000166**
Studijní program: **N2612 Elektrotechnika a informatika**
Studijní obor: **Informační technologie**
Název tématu: **Návrh řídicího systému pro Hexapod s Raspberry Pi**
Zadávající katedra: **Ústav informačních technologií a elektroniky**

Z á s a d y p r o v y p r a c o v á n í :

1. Seznamte se s šestinohým mobilním robotem Hexapod. Prostudujte inverzní kinematiku robota a možnosti jeho věrohodného pohybu.
2. Navrhněte matematický model ovládání robota.
3. Realizujte ovládací software robota pomocí Raspberry Pi přes webové rozhraní s video streamem, případně realizujte další možnosti ovládání.
4. Prototyp prakticky otestujte a pomocí senzorů realizujte základní úlohu vyhýbání se překážkám.

Rozsah grafických prací: Dle potřeby dokumentace

Rozsah pracovní zprávy: cca 40-50 stran

Forma zpracování diplomové práce: tištěná/elektronická

Seznam odborné literatury:

- [1] ZÁDA, Václav. Robotika: matematické aspekty analýzy a řízení. Vyd. 1. Liberec: Technická univerzita v Liberci, 2012. ISBN 978-80-7372-882-3.
- [2] NOVÁK, Petr. Mobilní roboty: pohony, senzory, řízení. 1. vyd. Praha: BEN - technická literatura, 2005. ISBN 80-7300-141-1.

Vedoucí diplomové práce: **Ing. Miroslav Holada, Ph.D.**
Ústav informačních technologií a elektroniky

Datum zadání diplomové práce: **12. září 2016**

Termín odevzdání diplomové práce: **15. května 2017**

prof. Ing. Zdeněk Pliva, Ph.D.
děkan



prof. Ing. Ondřej Novák, CSc.
vedoucí ústavu

V Liberci dne 12. září 2016

Prohlášení

Byl jsem seznámen s tím, že na mou diplomovou práci se plně vztahuje zákon č. 121/2000 Sb., o právu autorském, zejména § 60 – školní dílo.

Beru na vědomí, že Technická univerzita v Liberci (TUL) nezasahuje do mých autorských práv užitím mé diplomové práce pro vnitřní potřebu TUL.

Užiji-li diplomovou práci nebo poskytnu-li licenci k jejímu využití, jsem si vědom povinnosti informovat o této skutečnosti TUL; v tomto případě má TUL právo ode mne požadovat úhradu nákladů, které vynaložila na vytvoření díla, až do jejich skutečné výše.

Diplomovou práci jsem vypracoval samostatně s použitím uvedené literatury a na základě konzultací s vedoucím mé diplomové práce a konzultantem.

Současně čestně prohlašuji, že tištěná verze práce se shoduje s elektronickou verzí, vloženou do IS STAG.

Datum: 15.5.2017

Podpis: 

Poděkování

Rád bych touto cestou poděkoval panu **Ing. Miroslavu Holadovi, Ph.D.** za možnost pracovat na vlastním projektu. Také bych mu chtěl poděkovat za vstřícný přístup při řešení problémů a dotazů. Dále bych chtěl poděkovat **panu Ing. Martinu Lachmanovi, Ph.D.** za odbornou konzultaci a výrobu dílů ve školní laboratoři. Také bych chtěl poděkovat **Pavlu Hirňalovi** za odborné konzultace a pomoc s elektroinstalací robota.

Abstrakt

Česky:

Cílem této diplomové práce je realizace mobilního robota typu Hexapod a realizace ovládacího software. V první části této práce je seznámení s typy podvozků. Dále se práce zaměřuje hlavně na podvozky se šesti končetinami. Je zde rozebrána jejich konstrukce končetin, přímá a inverzní kinematika, pohybové algoritmy a jejich výhody a nevýhody. V další části je popsána konstrukce vlastního robota a jeho hardwarové vybavení. Následuje popis matematického modelu ovládní robota a realizace ovládacího software na Raspberry Pi, který je v práci zdokumentován. Práce vychází ze zdrojových kódů a matematického modelu pro robota PhantomX, které jsou v rámci práce upraveny. Přínosem práce je kompletní ovládní robota z webového rozhraní obsahující video stream.

Závěrem této práce je robot otestován na různých typech terénů a je zrealizována základní úloha vyhýbání se překážkám. K tomu je využito ultrazvukových senzorů a mikrokontroleru Arduino MEGA.

Klíčová slova:

Hexapod, Raspberry PI, Arduino, HC-SR04, Dynamixel AX-12+, Ovládací software

Abstract

English:

The purpose of this diploma thesis is to realize Hexapod mobile robot and implement a control software. In the first part of this work is familiarization with types of chassis. Furthermore, the work focuses mainly on chassis with six limbs. Their limb construction, direct and inverse kinematics, motion algorithms and their advantages and disadvantages are analyzed and described. The next part describes the construction of the own robot and its hardware equipment. Follows a description of the robot mathematical model and the implementation of the control software on Raspberry Pi, which is also documented in this thesis. The work is based on the source codes and the mathematical model for the PhantomX robot. The codes are modified in this work. The main contribution of this thesis is the complete control of the robot from the web interface, which includes the video stream.

In conclusion, the robot is tested on different types of terrain and the basic task of avoiding obstacles is realized. The ultrasonic sensors and the Arduino MEGA microcontroller are used for this task.

Keywords:

Hexapod, Raspberry PI, Arduino, HC-SR04, Dynamixel AX-12+, Control software

Obsah

Prohlášení.....	4
Úvod.....	13
1 Typy podvozků a teorie	15
1.1 Kolové podvozky.....	15
1.2 Pásové podvozky	15
1.3 Kráčivé podvozky.....	16
1.3.1 Rozdělení kráčivých podvozků	18
1.4 Kráčivé podvozky typu hexapod	19
1.4.1 Inspirace konstrukce podvozku	19
1.4.2 Návrh konstrukce končetin.....	21
1.4.3 Transformace souřadnic článků robota.....	22
1.4.4 Přímá a inverzní úloha robotiky	24
1.5 Pohybové algoritmy robotů typu hexapod.....	26
1.5.1 Pohyb typu tripod	27
1.5.2 Pohyb typu wave (vlna).....	28
1.5.3 Pohyb typu ripple (vlnění).....	29
1.5.4 Reálné použití pohybových algoritmů.....	29
1.5.5 Rychlost	30
1.5.6 Volná chůze	30
2 Konstrukce vlastního robota	31
2.1 Servomotory Dynamixel AX-12+	32
2.2 Konstrukce končetin robota.....	33
2.3 Konstrukce těla robota.....	35
2.4 Výsledná konstrukce robota	36
2.5 Komponenty a vybavení robota.....	36
2.5.1 Mini počítač Raspberry PI3 model B	36
2.5.2 Li-Po baterie, jejich využití a údržba.....	37
2.5.3 Spínaný stabilizátor napětí	38
2.5.4 Převodník USB2Dynamixel	39
2.5.5 Napájecí deska	39
3 Matematický model ovládání robota	40
3.1 Inverzní kinematika končetin	40

3.2	Omezení končetin	42
3.2.1	Konstrukční omezení končetin	43
3.2.2	Pohybové omezení končetin	44
4	Realizace ovládacího software	46
4.1	Hlavní program	47
4.2	Příprava a konfigurace Raspberry	48
4.3	LAMP server	49
4.4	Software pro video stream	50
4.4.1	Použitá webová kamera	51
4.5	Konfigurace Raspberry do režimu WIFI AP	51
4.6	Webové rozhraní s video streamem	52
4.7	Nastavení videa a hlavního programu jako služba	53
5	Testování, ukázková úloha a výsledky	55
5.1	Testování pohybových algoritmů v terénu	55
5.2	Ukázková úloha vyhýbání se překážkám	57
5.3	Spotřeba energie robota	62
5.4	Test webového rozhraní	63
6	Závěr	64
	Seznam použité literatury	66
	 Přílohy	 70
A	Obsah přiloženého CD	70
B	Webové rozhraní	71
C	Váha konstrukce robota	72
D	Výsledná konstrukce robota	73

Seznam obrázků a tabulek

Obrázek 1- Kráčivé rypadlo [22]	16
Obrázek 2 - The ATHLETE Rover [23]	17
Obrázek 3- RiSE [20]	17
Obrázek 4- Stewardova platforma [24]	19
Obrázek 5- Stavba končetiny hmyzu [25]	20
Obrázek 6 - Konstrukce robotické končetiny [7].....	22
Obrázek 7 - Závislosti stehna a holeně končetiny [7].....	25
Obrázek 8 - Vztah končetiny vzhledem k tělu [7]	25
Obrázek 9 – Schéma pohybu typu tripod.....	27
Obrázek 10- Schéma pohybu typu vlna	28
Obrázek 11 - Lynxmotion robot A-Pod [28]	31
Obrázek 12- Lynxmotion robot se čtyřmi končetinami [28]	31
Obrázek 13 - PhantomX AX Metal Hexapod Mark III [29].....	32
Obrázek 14- Servomotory Dynamixel AX-12+.....	33
Obrázek 15 - Tibia side plate	33
Obrázek 16 - Chodidlo končetiny	34
Obrázek 17 - Kompletní končetina	34
Obrázek 18 - Hlavní komponenty těla robota.....	35
Obrázek 19 - Rozpěry	35
Obrázek 20 - Výsledná konstrukce robota.....	36
Obrázek 21 - Raspberry Pi3 model B [30]	37
Obrázek 22 - Spínaný stabilizátor FOXY UBEC 3A [27].....	39
Obrázek 23 - Úhel kyčelního kloubu [13]	41
Obrázek 24 - Úhel stehenního a holenního kloubu [13].....	42
Obrázek 25 - Pohybové omezení končetin [13].....	44
Obrázek 26 - Schéma stavby systému	46
Obrázek 27 - Webová kamera Trust SpotLight Webcam Pro [31].....	51
Obrázek 28 - Test robota ve vysoké trávě	56
Obrázek 29 - Test robota na šterku	56
Obrázek 30 - Test robota na písku	57
Obrázek 32 - Ultrazvukový senzor HC-SR04	58
Obrázek 31 - Vývojová deska Arduino MEGA.....	58

Obrázek 33- Měření vzdálenosti ultrazvukovým senzorem	58
Obrázek 34 - Rozmístění senzorů HC-SR04	61
Tabulka 1 - Pokles kapacity baterie [11]	38
Tabulka 2 - Konstanty použité při výpočtu IK končetin	42
Tabulka 3 - Konstrukční omezení servomotorů	43

Seznam zkratk

DOF	degree of freedom, stupně volnosti
RAM	random access memory, paměť s přímým přístupem
EEPROM	Electrically Erasable Programmable Read-Only Memory, elektronicky vymazatelná paměť pouze pro čtení
USB	Universal Serial Bus, univerzální sériová sběrnice
LAN	Local Area Network, místní síť
GPIO	general purpose input output, vstupně výstupní piny pro všeobecné využití
HDMI	High-Definition Multimedia Interface,
TTL	transistor-transistor-logic, tranzistorově-tranzistorová logika
PHP	Hypertext Preprocessor, hypertextový preprocesor
SQL	Structured Query Language, strukturovaný dotazovací jazyk
SDK	Software development kit, sada vývojových nástrojů
RDP	Remote Desktop Protocol, protokol pro vzdálenou plochu
VNC	Virtual Network Computing, vzdálené připojení ke grafickému uživatelskému rozhraní
VPN	virtual private network, virtuální privátní síť
IP	Internet Protocol
AES	Advanced Encryption Standard, standard pokročilého šifrování
AP	Access point, přístupový bod
UART	Universal Asynchronous Receiver and Transmitter, univerzální asynchronní sériové rozhraní

Úvod

Robotika je aktuální a stále se rozvíjející odvětví průmyslu. S novými technologiemi vznikají nové způsoby řízení a nové konstrukce robotů. Cílem výroby robotů je automatizace fyzických, časově náročných a opakujících se prací člověka [1]. V jistých případech pracuje robot lépe než člověk sám. Pracuje rychleji, bez chyb a vždy přesně. Dobrým příkladem je automatizovaná výrobní linka. Roboti jsou dále konstruovány jako pomocná ruka pro člověka a do nepřístupných či nebezpečných míst.

Při vývoji nových technologií a robotů, většinou člověk hledá inspiraci v přírodě [21]. Není potřeba vynalézat něco, co příroda již vytvořila. Inspirace pochází jak od živočichů, tak rostlin. Od způsobu jejich chování, tak od konstrukce těla. Například rychlovlak nazývané Šinkanzeny mají tvar odvozený od zobáku ledňáčka [19].

Kromě robotů pracujících na výrobních linkách jako mechanické paže, či „robotů“ v domácnostech, se lidé zaměřují také na výrobu robotů připomínajících zvířata či lidi samotné. Jednou firmou, zabývající se těmito roboty je Boston Dynamics [20]. Všichni roboti vyrobené touto firmou používají pro svůj pohyb kráčivé podvozky, které mohou mít na konci končetin kola. Tyto podvozky, na rozdíl od kolových a pásových, se výborně hodí do nerovného či velmi členitého terénu. Kráčivé podvozky mohou mít různý počet končetin.

Právě tyto roboti s kráčivými podvozky mě zaujali a rozhodl jsem se je důkladně prostudovat. Nejsou sice tak rychlé, ale mohou se dostat tam, kam kolové či pásové podvozky nemohou. Typickým příkladem robota s kráčivým podvozkem je Hexapod. Ten má 6 končetin a jeho konstrukce a pohyb je právě inspirován přírodou. Má několik výhod a velké využití. Tento typ robota přináší, ale i nevýhody. Jednou značnou nevýhodou je složitý ovládací systém.

Diplomová práce se zabývá právě výše zmíněným robotem typu hexapod. V první kapitole jsou popsány všechny základní typy podvozků. Jsou zde shrnuty výhody, nevýhody a uvedeny příklady. Dále se tato kapitola zaměřuje na kráčivé podvozky se šesti končetinami. Je zde zmíněna inspirace přírodou a popsána konstrukce končetin. Dále jsou zde popsány výpočty souřadnic článků robota a rozebrána přímá a inverzní úloha robotiky. Pro kráčivé podvozky se šesti končetinami jsou také popsány pohybové algoritmy, včetně jejich výhod, nevýhod a jejich reálné využití.

V další kapitole je popsána konstrukce vlastního robota a také seznámení s použitým materiálem potřebným k výrobě hexapodu. Důležitou část také tvoří popis matematického modelu robota, kde je především zmíněna inverzní kinematika končetin.

Dále se práce zabývá realizací kompletního ovládacího software, realizovaného na mini-počítači Raspberry PI 3. V závěru této diplomové práce byl robot otestován na různých terénech a byla vytvořena ukázková úloha, kde se robot vyhýbá překážkám pomocí vzdálenostních senzorů.

1 Typy podvozků a teorie

Podvozek je základ (spodní část) dopravních prostředků, strojů nebo právě robotů, které se pohybují po pevné, kapalné či sypké ploše. Součástí podvozku jsou většinou kola, která zajišťují minimální třecí odpor stroje při jeho pohybu po pevné ploše, dále obsahují pružiny k zachycení nárazů a tlumiče pro pohlcení energie těchto nárazů. Typů podvozků je několik a každý typ má své výhody a nevýhody. Díky tomu má každý typ podvozku jiné využití.

1.1 Kolové podvozky

Kolové podvozky, jak už je z názvu jasné, používají kolo pro přeměnu rotačního pohybu v pohyb posuvný za účelem snížení třecí síly. Tyto podvozky mohou disponovat jedním nebo více koly (zpravidla čtyřmi koly) a jsou relativně jednoduché. Vynález kola se odhaduje před více než 7 tisíci lety. Je tak jedním z nejvýznamnějších vynálezů v celých dějinách techniky. Kolo umožňuje dosahovat vysokých rychlostí pohybu, především v ideálních terénních podmínkách, kdy je velikost kola několikanásobná vzhledem k nerovnosti terénu. Nevýhodou je, že kolové podvozky mohou uvíznout v sypkém, měkkém nebo klzkém terénu, případně zůstat viset na rámu podvozku.

1.2 Pásové podvozky

Pásové podvozky svým použitím navazují na podvozky kolové. Mají na upravených kolech nasazené pásy. Tyto pásy se liší podle využití v terénu, ve kterém se mají pohybovat. Rychlost pásových podvozků je nižší než u podvozků kolových, hlavně díky konstrukčnímu omezení a vlivem třecí síly. Dosahují však lepších výsledků v členitějším prostředí, kde kolové podvozky selhávají. Hlavním důvodem je velikost styčné plochy s terénem, která je oproti kolovým podvozkům mnohem větší. Jako dobrý příklad lze uvést tank v členitém terénu a sněžný skútr v měkkém a klzkém terénu.

1.3 Kráčivé podvozky

Kračivé podvozky jsou zcela odlišné od obou předchozích variant. Podvozek není poháněn koly, ale pohybuje se pomocí přesouvání svých končetin, prakticky chodí. Tento způsob pohybu má hned několik výhod. Kráčivé podvozky dokáží překonat velké terénní nerovnosti, a to až do výše zdvihu svých končetin, resp. do limitu své konstrukce. Mohou překonat také díry v terénu, do kterých by kolové či pásové podvozky zapadly. Dokáží dokonce pokračovat v pohybu i při ztrátě nebo poškození některé z končetin. Díky těmto vlastnostem jsou jedním z nejuniverzálnějších typů podvozků. Tělo podvozku ve většině případů zůstává stále v jedné rovině, a proto je zde vhodné umístit různé senzory nebo řídicí kabinu. Kromě výhod přináší kráčivé podvozky i nevýhody. Jejich konstrukce je složitá a spotřeba energie je větší než u předchozích typů, protože kráčivé podvozky spotřebovávají energii, i když se nepohybují. Další nevýhodou jsou složité systémy pro řízení.

Existují také podvozky, které kombinují více typů podvozků dohromady. Rozšířené jsou kombinace kráčivých a kolových podvozků, kde ovšem jeden typ podvozku je hlavní, a druhý pouze vylepšuje nedostatky či zvyšuje jeho schopnosti. Dobrým příkladem je kráčivé rypadlo, které je vybaveno zpravidla čtyřmi hydraulicky ovládanými končetinami, dvě nebo čtyři končetiny jsou vybaveny koly. Díky tomuto uspořádání je kráčivé rypadlo (bagr) schopen pohybu a práce i v naprosto extrémních svazích, ale i např. v korytech vodních toků. Rovněž je schopen překonávat kolmé stupně, např. nalézt a slézt z nákladního auta či slézt do výkopu či vodního toku.



Obrázek 1- Kráčivé rypadlo [22]

Jako dalším zajímavým příkladem je „The Athlete Rover,“ který byl vyroben v JPL (Jet Propulsion Laboratory). Tento robot byl navržen pro použití na Měsíci. Každá z šesti končetin má 6 stupňů volnosti a na konci končetin jsou kola. Ke končetinám lze připojit i pracovní nástroje (vrták, klepeto) pro práci v terénu. Robot je schopen unést až 450 kg.



Obrázek 2 - The ATHLETE Rover [23]

Posledním příkladem, který mě velmi zaujal je robot RiSE od Boston Dynamics. Tento robot má šest končetin a jeho specializací je šplhání po svislých stěnách jako jsou stěny domů, stromy a ploty. RiSE k tomu používá nohy s mikro drápy. Dokáže adaptovat držení těla pro lepší stabilitu a schopnost šplhat. Pohyblivý ocas pak drží stabilitu při svislém stoupání. Robot je 0,25 m dlouhý, váží 2 kg a pohybuje se rychlostí 0,3 m/s. Každá noha tohoto robota je poháněna dvojicí elektromotorů. Řídící jednotka robota ovládá pohyb nohou, řídí komunikaci a čte data z řady senzorů, mezi kterými jsou například senzor polohy, deformační senzory a kontaktní čidla.



Obrázek 3- RiSE [20]

1.3.1 Rozdělení kráčivých podvozků

Kráčivé podvozky můžeme rozdělit do různých skupin podle počtu končetin, podle počtu kloubů, podle typu pohonu a v neposlední řadě také podle pohybové techniky, která souvisí s počtem končetin.

První variantou je rozdělení podle počtu končetin. Počet končetin ovlivňuje stabilitu robota i způsob pohybu. V případě dvou, tří nebo někdy i čtyř končetin se podvozky nedokáží pohybovat tak, aby zůstaly stále ve stabilní poloze, jedná se tak o dynamickou stabilitu. Dynamicky stabilní podvozek se v určité fázi dostane do stavu nestability, tedy balancuje na jedné či více nohou, nebo padá. Typ pohybu v tomto případě může být podobný lidem (především pro 2 končetiny), kde pohyb probíhá kolem horizontální osy. Pokud má podvozek čtyři a více nohou, jedná se o staticky stabilní podvozek. Ten je v každém kroku pohybu vždy ve stabilní poloze. Pohyb v těchto případech je podobný hmyzu, kde se končetiny pohybují po vertikální ose.

Dále lze rozdělit kráčivé podvozky podle počtu kloubů. Aby kráčivý podvozek dokázal chodit, jsou potřeba alespoň 2 klouby, označovány jako 2 DOF. Jeden kloub je potřeba pro pohyb končetiny nahoru a dolů a druhý kloub pro pohyb dopředu a dozadu. Ovšem s těmito parametry nedokáže robot zcela efektivně chodit. Konec končetiny, který se dotýká země, v tomto případě opisuje kružnici a pokud má robot jít rovně, končetina by prokluzovala. Přidáním třetího kloubu a správnou synchronizací pohybů v kloubech se problém vyřeší. Těmto 3 kloubům se také říká majoritní osy. Existují i končetiny s více než třemi klouby (minoritní osy). Například 4 DOF končetiny, kde část končetiny za čtvrtým kloubem většinou slouží jako nášlapná plocha. Kromě nášlapné plochy pak u končetin s více než 3 DOF slouží další klouby (minoritní osy) k zajištění orientace nástroje či chapadla v pracovním prostoru, případně zvýšit manipulační pružnost., viz. The Athlete Rover. Samotnou chůzi čtvrtý kloub nijak nevylepší.

Použitý pohon také ovlivňuje a rozděluje kráčivé podvozky. Jedním typem pohonu je elektrický, který je jednodušší, rychlejší a tím i dynamičtější. Klouby jsou ohýbány pomocí servomotorů, či elektromotorů. Dále se používá hydraulický pohon. Ten je na rozdíl od elektrického výkonnější ale pomalejší. Proto se hodí spíše pro manipulaci s těžkými předměty. Typickým příkladem pro tento typ podvozků je Stewardova platforma. Skládá se z šesti končetin připojených po dvou k jedné plošině.

Díky této konstrukci lze rovinu nastavit do libovolného náklonu. Využívá se především pro letecké simulátory nebo pro zábavné atrakce.



Obrázek 4- Stewardova platforma [24]

Práce se dále zaměřuje na kráčivé podvozky se šesti končetinami. O této problematice je podrobněji napsána celá následující kapitola. Výsledný robot je pak touto kapitolou inspirován.

1.4 Kráčivé podvozky typu hexapod

Tato podkapitola se podrobněji zabývá roboty s kráčejícím podvozkem se šesti končetinami. Tento typ robota se nazývá Hexapod. Stavba robota i jeho pohyb je inspirován přírodou. V následujícím textu je inspirace podrobněji rozebrána a dále je popsána kinematika robota a pohybové algoritmy.

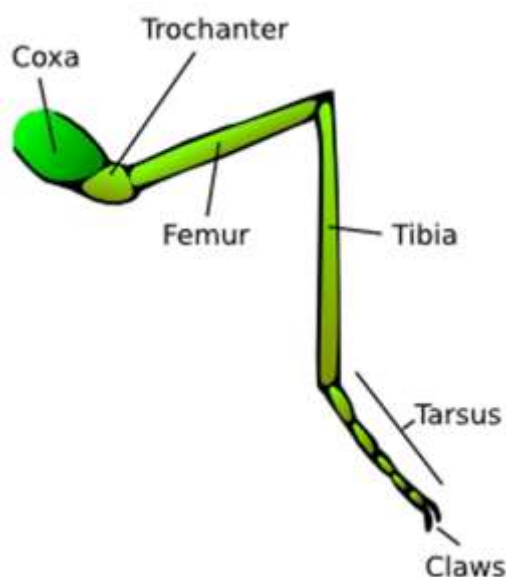
1.4.1 Inspirace konstrukce podvozku

Již dávno se konstruktéři robotů inspirovali přírodou. Především napodobovali některá zvířata či lidi. Jedním z velikánů byl Leonardo da Vinci, který sestrojil kráčející figurínu lva. Dále například Pierre Jasquet Droz vytvořil písáře pohybujícího rukama a prsty, automatickou kreslící figurku a pianistku sledující očima noty, která se po představení sama ukláněla. Důvody, proč spočívá zájem člověka o roboty a různé

mechanické hračky, lze zdůvodnit následovně. Snaha změřit síly s přírodou, ověřit si kam až sahají schopnosti člověka a pokusem o napodobení přírody lépe pochopit její zákonitosti. Dále je to snaha vytvořit dokonalého pomocníka, který by dovedl totéž, nebo i více než člověk a na něhož by se člověk mohl spolehnout více nežli na sebe. [1]

Proto i konstrukce kráčivých podvozků je inspirována přírodou. V případě čtyřnohých robotů to mohou být šelmy a koně. U šestinohých robotů to jsou brouci (členovci). A v případě osminohých robotů je inspirace v pavoucích. Stále zatím konstrukce robotů není tak dokonalá, jako stavba těla živočichů. Protože technologie není zatím tak vyspělá jako miliony let dlouhá evoluce. Stále dochází k tomu, že nově vzniklá technologie je založena na inspiraci z přírody.

Konstrukce končetiny robota typu hexapod je také inspirována přírodou. V odborné literatuře se pro označení částí končetin používají původní názvy. Na následujícím obrázku je končetina hmyzu včetně názvů částí končetiny. První částí je tzv. coxa, která je hned za kyčelním kloubem připojeným k tělu. Následně je stehenní kost femur, na kterou je napojena kost holení, tibia. Následuje tarsus připojený na holení kost. Jelikož tarsus představuje čtvrtý stupeň volnosti, konstrukce robotické končetiny ho pro pohyb nevyžaduje. Pokud je ale potřeba větší než bodová styčná plocha, je konstrukce čtvrtého stupně volnosti s nastavitelnou styčnou plochou vhodnou volbou.



Obrázek 5- Stavba končetiny hmyzu [25]

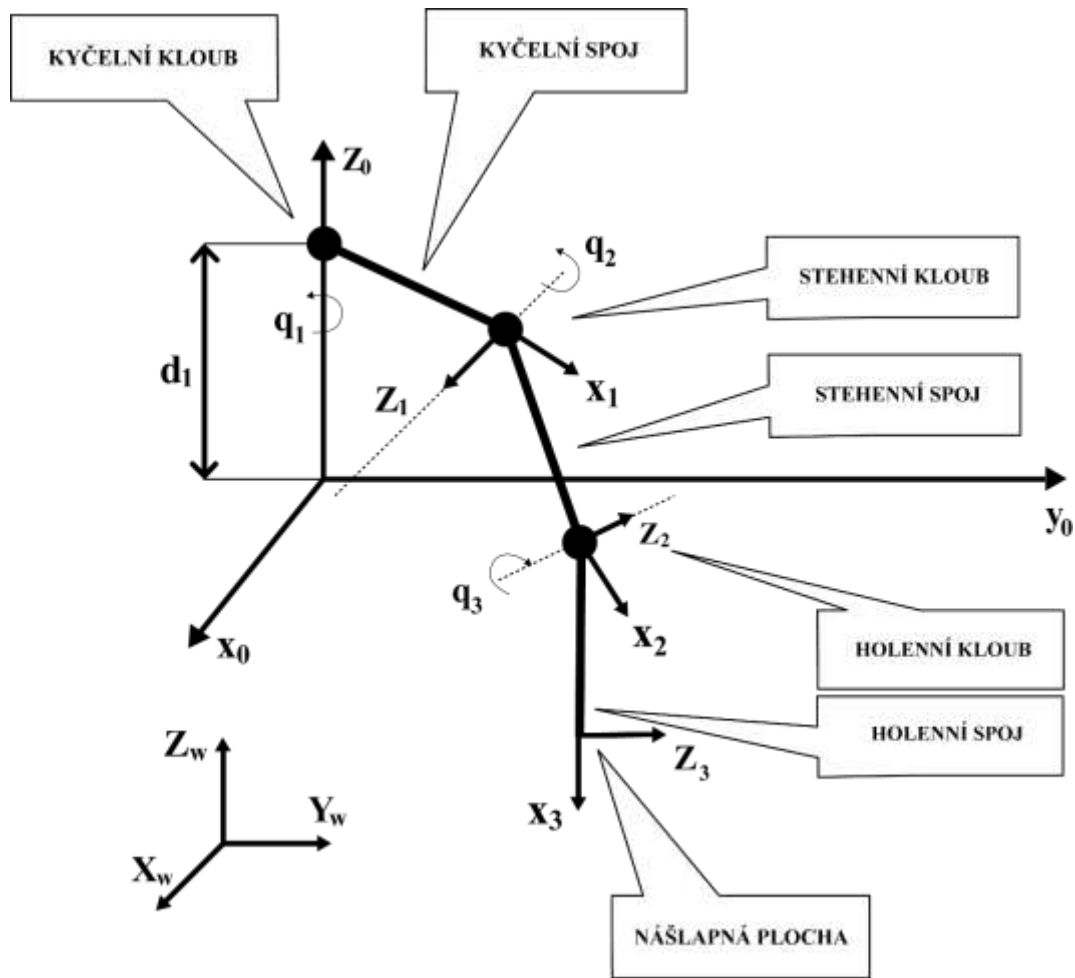
Kromě konstrukce končetin se člověk inspiruje i u konstrukce těla. Zde se využívá symetrických tvarů a v úvahu přichází tak dva zjednodušené tvary. Prvním je

obdélník, kde počet končetin i umístění na jedné straně je shodné s druhou stranou. Tento tvar je osově souměrný. V některých případech mohou být prostřední končetiny více vzdáleny od těla, než končetiny přední a zadní. Ty mají vzdálenost od těla shodnou. Osově souměrný tvar umožňuje větší dosah předních i zadních končetin při pohybu. Což může být výhodou při pohybu v členitém terénu. Druhým typem je pravidelný šestiúhelník. Tady je každá končetina umístěna na jednom vrcholu. Jedná se tak o středově souměrný tvar. Jeho výhodou je, že nemá žádnou část, kterou lze označit jako přední či zadní. Proto se pro změnu směru nemusí hexapod otáčet, ale stačí nastavit chůzi daným směrem.

1.4.2 Návrh konstrukce končetin

Při návrhu končetin je třeba myslet na tvar, počet stupňů volnosti či rozměry jednotlivých spojů. Klouby by měly mít dostatečný prostor pro manévrování a chůzi. Některé servomotory umožňují otáčení jen do 180 stupňů či méně, proto je dobré s tímto omezením při návrhu počítat.

Končetina robota typu hexapod se skládá ze tří částí. První část je k tělu robota připojena kyčelním kloubem a nazývá se coxa. Ta umožňuje pohybovat končetinou dopředu a dozadu ve směru pohybu celého robota. Následuje část zvaná femur (stehenní kloub), která slouží ke zvedání končetiny ve vertikálním směru. Na femur je napojen holenní kloub tibia. Slouží především k vyrovnávání chůze viz. předchozí kapitoly. Pohybuje se směrem k tělu a od něj, pokud je coxa nastavena rovně od těla. Ke konci končetiny pak může být připojena nášlapná plocha. Celou konstrukci končetiny lze pozorovat na následujícím obrázku.



Obrázek 6 - Konstrukce robotické končetiny [7]

1.4.3 Transformace souřadnic článků robota

K transformaci souřadnic článků robota se široce využívá Denavit-Hartenbergův (D-H) algoritmus. Jedná se o systematický přístup volby souřadných systémů pro jednotlivé články otevřených kinematických řetězců [1]. Pro popis kinematických vlastností je použita končetina robota, která disponuje třemi stupni volnosti, viz obrázek Obrázek 6. Geometrický model pro každou končetinu odpovídá vztahu mezi pohyblivým rámcem koncového bodu končetiny $O_i(x_i, y_i, z_i)$, kde $i = 1, 2, 3$ a fixním rámcem $O_w(x_w, y_w, z_w)$, viz. Obrázek 6. Klouby se číslovají od 1 do 3 počínaje kyčelním kloubem a konče holením kloubem. Končetiny jsou rozmístěny symetricky kolem osy ve směru pohybu. Obecná forma transformační matice z kloubu i do kloubu $i - 1$ se skládá ze čtyř operací [4]:

1. Rotace okolo osy z_{i-1} o úhel θ_i
2. Posun podél osy z_{i-1} o hodnotu d_i

3. Posun podél osy x_{i-1} o hodnotu a_i

4. Rotace okolo osy x_{i-1} o úhel α_i

Těmto čtyřem operacím odpovídají ve stejném pořadí následující homogenní matice [1]:

$$Rot_{z_{i-1}}(\theta_i) = \begin{bmatrix} \cos \theta_i & -\sin \theta_i & 0 & 0 \\ \sin \theta_i & \cos \theta_i & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (1.1)$$

$$Trans_{z_{i-1}}(d_i) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (1.2)$$

$$Trans_{x_{i-1}}(a_i) = \begin{bmatrix} 1 & 0 & 0 & a_i \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (1.3)$$

$$Rot_{x_{i-1}}(\alpha_i) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos \alpha_i & -\sin \alpha_i & 0 \\ 0 & \sin \alpha_i & \cos \alpha_i & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (1.4)$$

Po vynásobení matic v uvedeném pořadí, získáme výslednou matici ve tvaru:

$$T_i^{i-1} = \begin{bmatrix} \cos \theta_i & -\cos \alpha_i \sin \theta_i & \sin \alpha_i \sin \theta_i & a_i \cos \theta_i \\ \sin \theta_i & \cos \alpha_i \cos \theta_i & -\sin \alpha_i \cos \theta_i & a_i \sin \theta_i \\ 0 & \sin \alpha_i & \cos \alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (1.5)$$

Matice 1.5 představuje transformační matici, která transformuje souřadnice i-tého spoje (kloubu) do kloubu i-1 [1]. Celková transformace je získána vzájemným vynásobením tří transformačních matic [7]:

$$T_{base}^{coxa} = T_{femur}^{coxa} T_{tibia}^{femur} T_{base}^{tibia} \quad (1.6)$$

kde jednotlivé transformační matice jsou v následujících tvarech:

$$T_{femur}^{coxa} = \begin{bmatrix} \cos \theta_1 & 0 & \sin \theta_1 & L_1 \cos \theta_1 \\ \sin \theta_1 & 0 & -\cos \theta_1 & L_1 \sin \theta_1 \\ 0 & 1 & 0 & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (1.7)$$

$$T_{tibia}^{femur} = \begin{bmatrix} \cos \theta_2 & \sin \theta_2 & 0 & L_2 \cos \theta_2 \\ \sin \theta_2 & -\cos \theta_2 & 0 & L_2 \sin \theta_2 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (1.8)$$

$$T_{base}^{tibia} = \begin{bmatrix} \cos \theta_3 & -\sin \theta_2 & 0 & L_3 \cos \theta_2 \\ \sin \theta_3 & \cos \theta_3 & 0 & L_3 \sin \theta_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (1.9)$$

Podle článku [7] lze užitím rovnice a uvážením 1.7, 1.8 a 1.9 vyjádřit souřadnice nášlapné plochy končetiny následovně:

$$\begin{aligned} x &= \cos \theta_1 (L_1 + L_2 \cos \theta_2 + L_3 \cos(\theta_2 - \theta_3)) \\ y &= \sin \theta_1 (L_1 + L_2 \cos \theta_2 + L_3 \cos(\theta_2 - \theta_3)) \\ z &= d_1 + L_2 \sin \theta_2 + L_3 \sin(\theta_2 - \theta_3) \end{aligned} \quad (1.10)$$

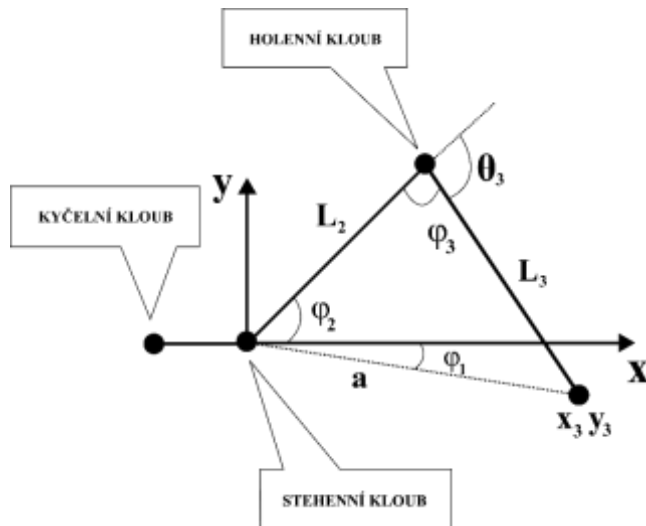
kde:

d_1 je vzdálenost od země ke kyčelnímu kloubu,
 L_i jsou jednotlivé délky spojů končetiny.

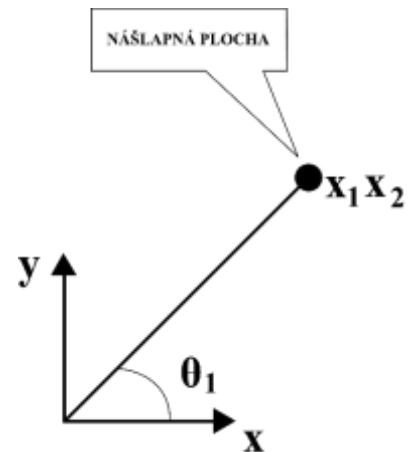
1.4.4 Přímá a inverzní úloha robotiky

Přímá úloha kinematiky robotů se zakládá na určení rovnice ramene či končetiny robota, nebo jeho rychlostí a zrychlení. K tomu jsou potřeba známé hodnoty proměnných $\theta_1, \theta_2, \theta_3$, označovány jako kloubové proměnné či strojové souřadnice. Pokud jsou zadány tyto souřadnice jako funkce času, lze na základě rovnic v předchozí podkapitole určit časový průběh veškerých poloh, rychlostí a zrychlení.

Inverzní úloha robotiky řeší problém nalezení cílových hodnot pro správné natočení kloubů. Cílem je tedy nalézt tři proměnné $\theta_1, \theta_2, \theta_3$, podle kterých se mají klouby nastavit tak, aby byla dosažena koncová poloha končetiny.



Obrázek 7 - Závislosti stehna a holeně končetiny [7]



Obrázek 8 - Vztah končetiny vzhledem k tělu [7]

Obrázek 8 znázorňuje vztah končetiny k tělu robota a obrázek 7 zobrazuje závislosti stehna a holeně končetiny.

Podle zdroje [1] se v robotice definuje tzv. čtyř kvadrantová verze funkce $arctg$:

$$atan2(y, x), \quad (1.11)$$

která dává správný úhel odpovídající vektoru (x, y) . V základním principu jde o funkci $arctg\left(\frac{y}{x}\right)$, kde výsledný úhel je opraven na správnou hodnotu podle toho, ve kterém kvadrantu leží bod o souřadnicích (x, y) . Pokud není tato funkce součástí softwaru, lze ji snadno naprogramovat. Uvážením rovnice (1.10) a následujících omezení: všechny klouby rotují pouze v rámci jedné osy, holenní (tibia) a stehenní (femur) klouby rotují v rámci dvou rovnoběžných os a samozřejmě všechny fyzická omezení všech kloubů, pak lze určit úhel jednotlivých kloubů. Úhel pro kyčelní kloub (coxa) můžeme získat s využitím rovnice 1.11 a obrázku 8 následovně:

$$\theta_1 = atan2(y_1, x_1) \quad (1.12)$$

Dále k zjištění zbývajících dvou úhlů lze využít geometrického přístupu. Podle zdroje [7] se pro zjednodušení aplikuje následující transformace na požadované souřadnice nášlapného bodu končetiny:

$$T_{coxa}^{femur} = \begin{bmatrix} (R_{femur}^{coxa})^T & -(R_{femur}^{coxa})^T * d_{femur}^{coxa} \\ 0 & 1 \end{bmatrix} \quad (1.13)$$

Úhel φ_2 reprezentující natočení stehenního kloubu, odvodit přímo z trojúhelníku:

$$\theta_2 = \varphi_2$$

Úhel φ_1 na obrázku 7 svírá osa x s přímkou a . Úhel je možné spočítat následovně:

$$\varphi_1 = \operatorname{atan2}(y_3, x_3) \quad (1.14)$$

kde x_3 a y_3 představují požadované souřadnice koncového bodu končetiny. Úhel stehenního kloubu lze získat následovně [7]:

$$\theta_2 = \operatorname{acos}\left(\frac{L_2^2 + a^2 - L_3^2}{2L_2a}\right) + \operatorname{atan2}(y_3, x_3) \quad (1.15)$$

Použitím Kosinovy věty lze vyjádřit úhel θ_3 . Výsledný tvar vypadá následovně:

$$\varphi_3 = \operatorname{acos}\left(\frac{L_2^2 + L_3^2 - a^2}{2L_2L_3}\right) \quad (1.16)$$

Následně z obrázku 7 lze vyčíst úhel θ_3 jako doplněk úhlu φ_3 do π :

$$\theta_3 = \pi - \varphi_3 \quad (1.17)$$

1.5 Pohybové algoritmy robotů typu hexapod

Kráčivé podvozky používají pro svůj pohyb chůzi, viz. kapitola 0. Chůze se popisuje jako sekvence kladení končetin požadovaným směrem a přenášení váhy podvozku. Při chůzi se předpokládá, že alespoň jedna končetina se musí dotýkat země (nejedná se o skoky).

U člověka je chůze jednoduchá a snadno popsatelná. Člověk disponuje dvěma končetinami, a proto se nenabízí více možností, jak sekvenčně klást končetiny. Pokud je jedna končetina nakročena dopředu, musí následovat druhá. Po ní opět následuje první a sekvence se tak neustále opakuje.

Jelikož hexapod disponuje šesti končetinami, existuje mnoho sekvencí, jak klást končetiny za sebou. Jelikož počet nohou ovlivňuje množství možných sekvencí pohybu. Je tento počet možný vyjádřit následovně [4]:

$$N = (2K - 1)! \quad (1.18)$$

Celkem pro šestinohého robota je to $11! = 39\,916\,800$ možných sekvencí pohybu. Ovšem jen některé sekvence nabízí dobrou stabilitu, rychlost či vůbec umožňují pohyb. V práci jsou popsány tři základní typy pohybu (sekvencí), tripod, vlna a vlnění.

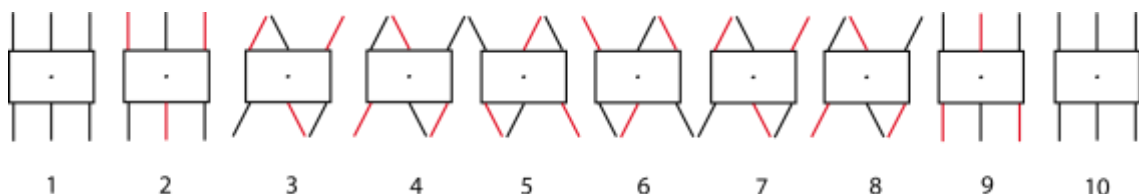
Ty se používají především na rovném terénu. Na členitém terénu se končetiny umisťují dynamicky. Například na základě senzorů.

Při reprezentaci pohybových algoritmů kráčivých robotů se musí uvažovat tyto informace. Základem je identifikace končetiny a identifikace konkrétního kloubu. K němu se definuje úhel, do kterého se kloub musí nastavit. Poslední důležitou informací je čas. Čas se ovšem neuvažuje absolutně, ale stačí informace, která akce (natočení kloubu) navazuje na předchozí. Pro zjednodušení se v následujícím textu uvažuje pohyb pouze rovně vpřed či vzad.

1.5.1 Pohyb typu tripod

V tomto typu pohybu jsou končetiny rozděleny do dvou skupin. První skupina obsahuje levou přední, levou zadní a pravou prostřední končetinu. Druhá skupina zbylé tři. Skupiny končetin jsou tak rozděleny zrcadlově. Pohyb probíhá tak, že se zvedne jedna skupina končetin, posune se dopředu a pak stoupne (položí) na zem. Následně se zvedne druhá skupina a začne se posouvat dopředu. Současně se první skupina končetin posouvá dozadu. V poslední fázi se druhá skupina položí a cyklus se opakuje. Zároveň je nutné, aby zvednuté končetiny se nejdříve položily a až pak se může zvednout druhá skupina. Tělo robota tak zůstane vyrovnané [4][5][12][13].

Pohyb tripod je jeden z nejrychlejších, ale také nejméně stabilní. Robot totiž během chůze vždy stojí na třech končetinách. Dvě jsou na jedné straně a jedna končetina je na druhé. V koncových bodech pohybu jsou navíc končetiny nejdále od těžiště robota a největší váha tak zůstává na dvou končetinách. Jedna končetina navíc vždy nese polovinu váhy celého robota. Tento pohyb se hodí pro lehkého či nezátíženého robota a robota který se pohybuje na rovnějším terénu [4][5][12][13].



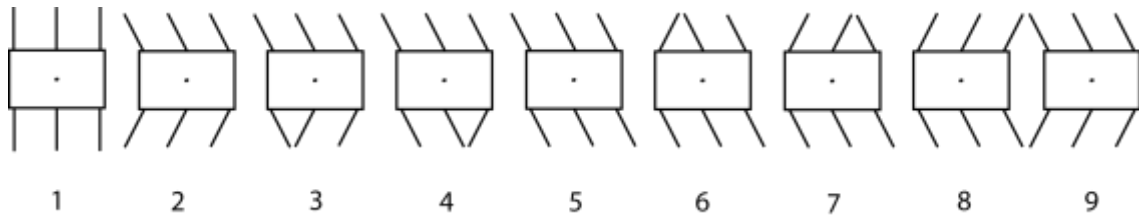
Obrázek 9 – Schéma pohybu typu tripod

Na obrázku Obrázek 9 je zjednodušeně znázorněn schématem pohyb typu tripod. Červené čáry představují končetiny ve vzduchu, černé pak končetiny na zemi. Pro

jednoduchost zde nejsou zobrazeny kroky, kdy robot nejdříve položí zvednutou skupinu končetin až pak zvedne druhou skupinu, ale tyto kroky jsou sjednoceny v jeden (krok 4, 6, 8). Z obrázku je vidět, že krok 7 je stejný jako krok 3, tedy zde se již sekvence opakuje. Celkem je možné pozorovat 6 kroků v rámci jedné sekvence.

1.5.2 Pohyb typu wave (vlna)

Tento algoritmus se zakládá na postupném posouvání končetin. Vpřed či vzad se hýbe pouze jedna končetina. Resp. jedna končetina se posune ve směru pohybu. Ostatních pět končetin se pomalu posouvají vzad a přemísťují tak tělo do další pozice. Toto je plynulejší verze pohybu. Existuje také varianta, kde se nejprve všechny končetiny postupně posunou vpřed a pak se všechny končetiny současně posunou vzad.



Obrázek 10- Schéma pohybu typu vlna

Tento typ pohybu je nejpomalejší, ale nabízí největší stabilitu a spolehlivost. Robot ve svém pohybu vždy stojí nejméně na pěti končetinách. Využívá se především v nepříznivých terénních podmínkách. Dobrým důkazem je robot RiSE (viz kapitola Kráčivé podvozky), který tímto typem pohybu dokáže vylézt stěnu panelového domu. Na končetinách má mikro drápy, kterými se vždy pět končetin drží stěny domu [4][5][12][13].

Obrázek 10 znázorňuje pohyb typu vlna. Jedná se o tu jednodušší verzi, kdy robot nejdříve nastaví postupně všechny končetiny dopředu (ve směru pohybu) a poté všechny současně posune dozadu. V jednotlivých krocích (2-8) na obrázku 10 se provádí více operací, než je znázorněno. Je to nejdříve zvednutí končetiny, pak posunutí do předu a nakonec položení na zem. Celkem se pohyb typu vlna skládá z 19 kroků. Krokem se myslí jedna operace algoritmu pohybu. V rámci těchto 19 kroků se robot posune o jednu skutečnou krokovou délku dopředu. Ta je vidět na obrázku Obrázek 10, v rámci kroků 8 a 9.

1.5.3 Pohyb typu ripple (vlnění)

Tento algoritmus pohybu je inspirován přírodou a je podobný vlně. Robot nastavuje končetiny směrem vpřed, kde každá končetina vykovaná stejný sled pohybů: zvedne se, posune se dopředu, položí na zem a pomalu se posouvá dozadu (tím posouvá robota vpřed). [4] Ovšem v okamžiku, kdy se jedna končetina pokládá na zem, se současně další končetina zvedá a začíná nakračovat vpřed. Vždy se ale zvedá končetina na druhé straně robota, tím se zvyšuje stabilita. Končetiny na jedné straně robota jsou fázově posunuty o 180° oproti končetinám na druhé straně a cyklus končetin na levé straně začíná v půlce průběhu cyklu končetin na straně pravé.

V tomto typu pohybu hraje roli rychlost končetiny v jednotlivých částí kroků. Celková rychlost chůze robota je ovlivněna rychlostí posunu jednotlivých končetin vzad. Ostatní pohyby končetiny probíhají v maximální možné rychlosti, díky tomu je doba, po kterou je končetina zvednutá, minimální. Pohyb je tak velice stabilní a plynulý [4][5][12][13].

1.5.4 Reálné použití pohybových algoritmů

V praxi se robot pohybuje i jinými směry, než jen dopředu či dozadu. V jednotlivých typech pohybu dokáže robot chodit do strany, šikmo, či se otáček kolem osy procházející středem svého těla. Dále by robot měl být schopen i zatačet. Tedy při svém pohybu opisovat kružnici o různých poloměrech. Při tomto pohybu se vnitřní končetiny (končetiny uvnitř kružnice) pohybují pomaleji a nenakročují tak daleko, či vnější končetiny protahují a zrychlují svůj krok.

Výběr vhodného algoritmu závisí na členitosti terénu. Do členitého terénu je lepší zvolit algoritmus vlna či vlnění a na rovný terén pak pohyb tripod. Důležité je také zvážit maximální nosnost robota s ohledem na výkon servomotorů. S těžkým nákladem je třeba zvolit algoritmus, který při svém pohybu má co nejvíce končetin na zemi.

Jednotlivé algoritmy pohybu je možné měnit i za provozu robota na základě členitosti terénu, stavu robota, či zatížení. Změnu algoritmu je dobré provést ve stavu, kde nehrozí destabilizace robota.

1.5.5 Rychlost

Jak už bylo řečeno výše, nejrychlejší algoritmus je tripod, protože se skládá z nejméně kroků a posouvá nejvíc končetin najednou. Nejpomalejší je pohyb typu vlna. Kompromis mezi rychlostí a stabilitou (mezi tripod a vlnou) nabízí pohyb vlnění.

U všech výše popsaných pohybů se jedná o chůzi. Pro rozlišení, zda se jedná o stále o chůzi či už běh se používá následující rovnice [5]:

$$b = \frac{a}{t} \quad (1.19)$$

kde a je doba pobytu robota na zemi, tj. alespoň jedna končetina se dotýká země. A t je doba trvání jednoho cyklu algoritmu pohybu. Pokud je $b < 0,5$, jedná se o běh. Robot při svém pohybu stojí méně než polovinu času na zemi.

Kromě zvolení typu pohybu, lze rychlost změnit i jinými způsoby. Prvním způsobem, jak zvýšit maximální rychlost, je prodloužit délku jednoho kroku končetiny. Zde ovšem musíme brát v úvahu pracovní prostor ostatních končetin, aby nedocházelo v některých situacích ke kontaktu končetin. Dále prodloužení délky kroku může mít za následek zmenšení stability robota, především u pohybu tripod. U ostatních pohybů se stabilita téměř neovlivní, protože stojí na čtyřech a více končetinách. Pokud se prodlouží krok, může se pro zachování rychlosti zmenšit frekvence kroků končetin a na tělo robota tak bude působit menší setrvačnost z pohybu končetin [5]. Druhou možností zvýšení rychlosti pohybu je zvýšení frekvence pohybu končetin. To lze udělat i programově a lze tak měnit rychlost dynamicky až na maximální rychlost servomotorů.

1.5.6 Volná chůze

Jedná se o pohyb, kde končetiny nemají pevně stanovenou souřadnici, kam stoupnou. O této poloze rozhoduje tvar terénu, který nemusí být detekován orientačními senzory. Jedná se např. o malé díry či překážky. U Robota, který používá algoritmus s předem stanovenými souřadnicemi došlapu končetin, se může stát, že některé končetiny mohou zůstat viset ve vzduchu. Robot pak předpokládá při své chůzi, že končetina je pevně na zemi. Pokud je ve vzduchu, bude v takových případech robot padat. Problém lze vyřešit použitím nášlapných sensorů na konci končetiny. Robot pak bude pokládat končetinu tak dlouho, dokud nedošlápne na terén, či se nedostane na konec svého pracovního prostoru. [5]

2 Konstrukce vlastního robota

Existuje několik firem, které se zabývají výrobou konstrukcí hexapodů. Například firma Lynxmotion vyrábí mnoho robotů. Používají konstrukci pro standardní modelářské serva, jako jsou HS-645MG od firmy Hitec. Lynxmotion nabízí, kromě mnoho dalšího, konstrukce pro robotické ruky, biped roboty (humanoidní roboty), quadrapods a hexapod roboty.



Obrázek 12- Lynxmotion robot se čtyřmi končetinami [28]



Obrázek 11 - Lynxmotion robot A-Pod [28]

Na obrázku 12 je čtyřnohý robot, jehož konstrukce vyžaduje 12 servomotorů a na obrázku 11 je robot označován jako A-Pod a obsahuje 25x servo HS-645MG.

Další firmou, zabývající se výrobou robotů, je Trossen Robotics. Nabízí také velkou škálu robotů, ale jejich konstrukce je odlišná od konstrukce Lynxmotion. Je navržena pro serva Dynamixel od firmy Robotis. Tyto serva jsou navržena speciálně pro roboty. Trossen Robotics nabízí konstrukci hliníkového hexapoda, kterou je možné vidět na obrázku 13. Robot se skládá z 18 servomotorů Dynamixel AX-12A nebo silnějších a rychlejších AX-18A.

V neposlední řadě je to výše zmíněná firma Robotis, která kromě vyráběných servomotorů nabízí také vlastní stavebnice robotů pro vzdělávání.

Na pracovišti vedoucího diplomové práce je k dispozici stavebnice Robotis Premium včetně servomotorů Dynamixel AX-12+. Tato stavebnice a servomotory byly využity ke konstrukci robota. Obdobné díly a konstrukci nabízí právě výše zmíněná firma

Trossen Robotics. Stavebnice Robotis ovšem nenabízí tak propracované díly těla a končetin nohou, a proto bylo potřeba některé díly vyrobit.



Obrázek 13 - PhantomX AX Metal Hexapod Mark III [29]

2.1 Servomotory Dynamixel AX-12+

V současné době se již tato verze servomotorů nevyrobí a je nahrazena novější verzí AX-12A, která má stejné parametry, ale liší se pouze v externím designu. V obou případech se jedná o „chytré“ digitální serva pracující v rozmezí 9-12 V [8]. Výhodou těchto servomotorů je, že se zapojují za sebou jako řetěz a ovládání probíhá přes sériovou linku s rychlostí až 1 MBPS. Každé servo obsahuje ovládací tabulku složenou z EEPROM a RAM paměti. RAM paměť se resetuje do výchozích hodnot pokaždé, kdy se servo zapne a obsahuje provozní data. EEPROM paměť udrží stav i po vypnutí a obsahuje konfigurační data. V této paměti je právě uloženo ID servomotoru, které musí být na jedné sběrnici unikátní. Všechny serva poslouchají příchozí datové packety, ale reagují na ně jen serva, pro které je packet určen. Kromě mnoho dalších konfigurací, lze povolit například „Alarm Shutdown.“ Servo se pak samo vypne a zabrání tak poškození při detekované chybě či alarmu (přehřátí, špatné napětí, ...). V RAM paměti se nachází mnoho informací. Mezi ně patří například: pozice, rychlost, zátěž, napětí a teplota.

Na obrázku 14 je zobrazené servo AX-126 z přední i zadní strany. Na přední straně serva je vidět ryska, která znázorňuje středovou polohu. Na zadní straně se servo připojuje do řetězu. Jeden konektor se použije jako vstup a druhý může být použit jako

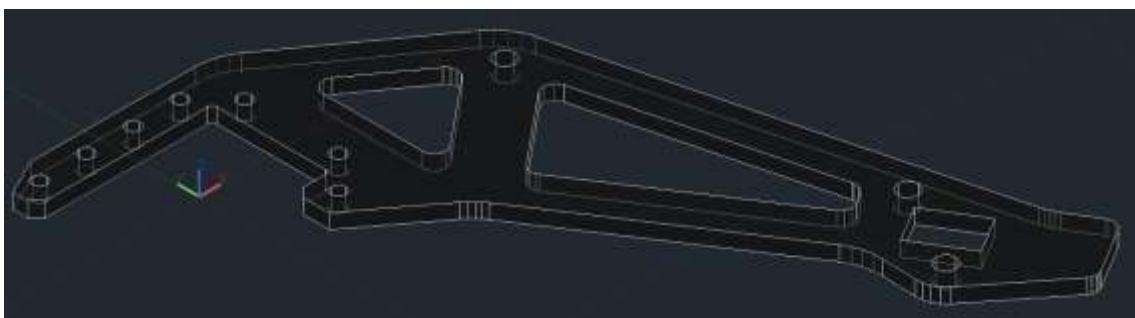
výstup pro další servo. Díky otvorům na stranách, které jsou určeny pro šroubky, lze servo montovat mnoha způsoby.



Obrázek 14- Servomotory Dynamixel AX-12+

2.2 Konstrukce končetin robota

K sestavení končetin byla použita stavebnice Bioloid Premium a servomotory AX-12+. Konkrétně k částem coxa a femur. Při řešení byly nalezeny vhodné a volně dostupné 3D modely pro končetiny. Konkrétně se jedná o model plochého dílu pro tibia a model pro „chodidlo,“ které se na konci nohy připevní. Na konec chodidla byly umístěny gumové výstupky, které zamezí smekání na hladkém povrchu.



Obrázek 15 - Tibia side plate

Na obrázku 15 je skica ploché části končetiny v programu AutoCAD. Tento model je k dispozici na adrese <https://grabcad.com/library/phantomx-hexapod-mark-ii-1> ve formátu pro 3D program firmy SolidWorks. Model byl převeden do formátu pro

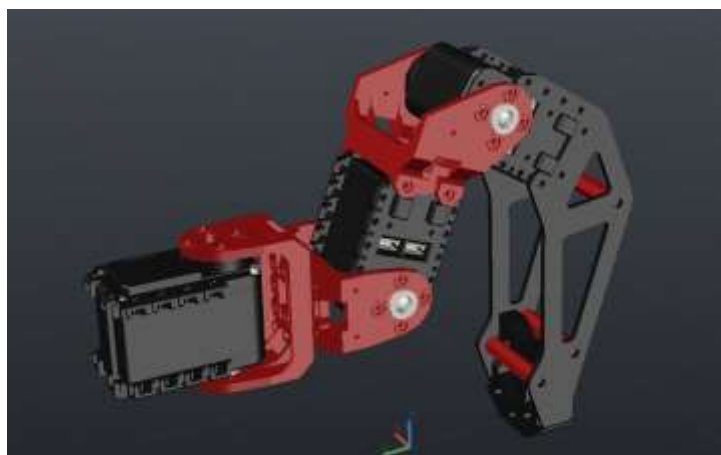
AutoCAD a připraven na výrobu. Bylo mi doporučeno vyrobit tento díl z železných plátů, a ne z plastu, protože by výroba na 3D tiskárně byla příliš drahá. Tento díl byl proto vyřezán z 1,5 mm železného plátu ve školních laboratořích. Menší tloušťka plátu by mohla způsobit prohýbání, a naopak silnější plát by byl příliš těžký. Dohromady bylo vyrobeno 12 stejných dílů.

Na tento díl navazuje „chodidlo,“ které se vkládá mezi dva ploché pláty v místě obdélníkového výřezu. Model tohoto dílu by stažen z výše zmíněného zdroje a formát upraven pro vtištění na 3D tiskárně. Jelikož bylo potřeba 6 těchto malých dílů, které mají na různých místech různou tloušťku, byla 3D tiskárna vhodná volba. Díly byly vyrobeny na soukromé tiskárně. Na následujícím obrázku (16) je zobrazen tento díl v 3D programu.



Obrázek 16 - Chodidlo končetiny

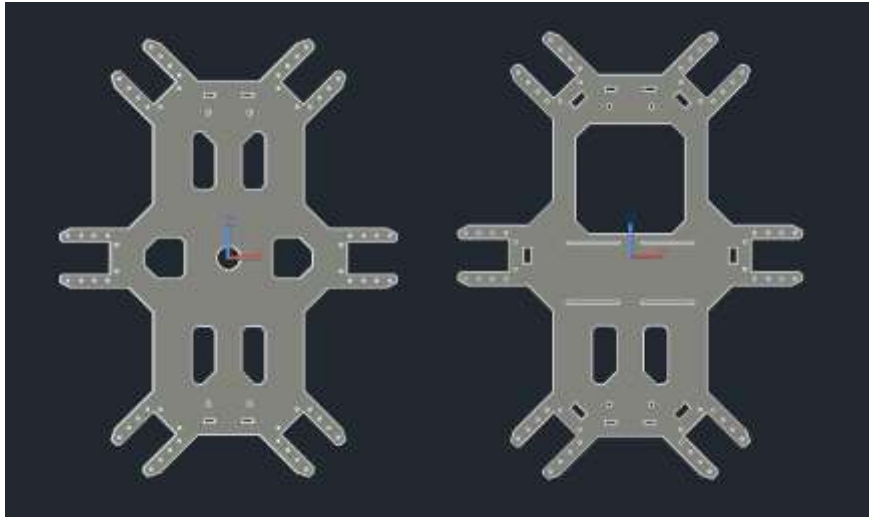
Po smontování těchto dílů vznikne kompletní končetina, kterou je možné vidět na obrázku 17. Kde zleva servomotor a 2 plastové spojovníky představují část coxa, prostřední část femur je složena z jednoho servomotoru a jednoho šikmého spojovníku. Tibia pak tvoří zbytek, tj. servo a vyrobené díly popsané výše.



Obrázek 17 - Kompletní končetina

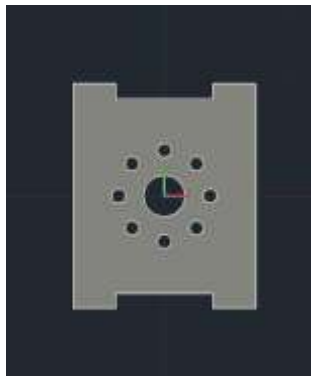
2.3 Konstrukce těla robota

Konstrukce těla se skládá ze dvou hlavních částí, servomotorů a dvou rozpěrek. Hlavní části byly opět kvůli ceně a pevnosti vyřezány laserem z 1,5mm tlustého železného plátu.



Obrázek 18 - Hlavní komponenty těla robota

Na obrázku Obrázek 18 je na levé straně zobrazen horní díl těla robota a na pravé straně spodní díl. Dohromady je spojuje 6 servomotorů na připravených místech. V přední i zadní části obou plátů jsou připravené obdélníkové výřezy pro nasazení rozpěr. 3D model rozpěry je možné vidět na obrázku 19. Na spodním dílu těla robota jsou připravené 4 drážky, kterými lze vést např. pásky na suchý zip a připevnit tak baterii v těžišti robota. Velké výřezy v plátech slouží pro dobrou manipulaci s dalšími komponenty robota a také pro odlehčení konstrukce. Na rozpěry lze umístit senzory.



Obrázek 19 - Rozpěry

2.4 Výsledná konstrukce robota

Výsledný 3D model robota je zobrazen na obrázku 20. V modelu není vymodelována žádná elektronika, baterie, šroubky ani dráty. Finální sestavený model je možné vidět v příloze.



Obrázek 20 - Výsledná konstrukce robota

2.5 Komponenty a vybavení robota

V rámci zadání práce bylo nutné splnit využití mini-počítače Raspberry PI pro ovládání celého robota přes webové rozhraní s video streamem. Zvolil jsem poslední verzi Raspberry Pi3 model B, protože je rychlejší než předchozí verze, obsahuje vestavěné WiFi a v ČR stojí téměř stejně jako předchozí verze [9]. Pro video stream jsem zvolil cenově dostupnou webovou kameru připojitelnou přes USB rozhraní.

2.5.1 Mini počítač Raspberry PI3 model B

Tento cenově dostupný mini-počítač velikosti kreditní karty nabízí obrovské využití a největší komunitu uživatelů. To potvrzují magazíny o Raspberry, vycházející každý měsíc [10]. Raspberry Pi nabízí výrazně vyšší výkon, než například Arduino a disponuje real-time operačním systémem. Pro komunikaci s uživatelem na webovém rozhraní se hodí vestavěná Wi-Fi. Počítač má 4 USB porty, díky kterým lze připojit

webovou kameru i USB převodník pro komunikaci se servomotory. Celá konfigurace Raspberry Pi3B[9]:

- 4 jádrový 64bitový processor ARMv8 s frekvencí 1,2GHz
- 802.11n bezdrátová LAN
- Bluetooth 4.1
- 4 USB porty
- 40 programovatelných pinů (GPIO)
- HDMI port
- Ethernetový port
- Kombinovaný 3,5mm jack
- Rozhraní pro kameru
- Rozhraní pro display
- Slot pro Micro SD kartu



Obrázek 21 - Raspberry Pi3 model B [30]

Pro provoz Raspberry Pi je potřeba použít nějaký operační systém. Byl použit oficiální operační systém Raspbian Jessie, který lze stáhnout zdarma z oficiálních stránek www.raspberrypi.org. Jedná se o OS na bázi linuxového jádra.

2.5.2 Li-Po baterie, jejich využití a údržba

V rámci práce byla využita Li-Po baterie FOXY G2 s 2600 mAh / 11.1 V [26]. Existují i baterie s větší kapacitou, ale ty mají bohužel také větší velikost a nevejdou se do těžiště robota.

Tato baterie nabízí vybíjecí rychlost 40 C. To znamená, že ji lze vybijet stálým proudem 20 x kapacita baterie v ampérech. V tomto případě 104 A. Pro použití v mobilním robotu je to více než dostatečné. Krátkodobě lze baterii vybijet až 80 C. Baterie nabízí energii 28,9 Wh a je těžká 208 g. Nabíjet tuto baterii je možné 1-2 C, ideálně s nabíječkou obsahující balancer [26].

S Li-Po bateriemi je potřeba zacházet rozumně. V případě nabíjení více článkových baterií je doporučeno použít balancer. Baterie se připojuje k nabíječce vybíjecím konektorem a balancerem. Nabíječka pak hlídá každý článek zvlášť a nestane

se tak, že by jeden článek byl přebítý a druhý málo nabitý. Nabíjení by mělo probíhat doporučeným proudem, který je většinou 1 C. Před nabíjením by měla být baterie studená, a neměla by se nabíjet hned po provozu. Plně nabité články baterie mají napětí 4,2 V. [11]

Při vybíjení je potřeba baterii hlídat. Například pomocí jednoduchého „pípání“, nebo průběžným měřením baterie. Při 2,7 V na jednotlivých člancích dochází již k zničení baterie. Je dobré baterii odpojit například při 3,4-3,6 V na jednom článku. V následující tabulce je příklad chování kapacity baterie při různém napětí. Uvažuje se baterie s 5 000 mAh. Nejprudší pokles kapacity baterie nastává mezi 4,1V a 3,8V. [11]

Napětí článku	4,2 V	4,1 V	3,8 V	3,7 V	3,6 V	3,5 V	3,4 V	
Zbylá kapacita	5 040	4 540	1 740	640	140	65	0	
Spotřebovaná kapacita	0	500	3 300	4 450	4 900	4 975	5 040	
Stav nabití v %	100 %	90,1 %	34,5 %	12,7 %	2,8 %	1,3 %	0,0 %	

Tabulka 1 - Pokles kapacity baterie [11]

Baterie by se nikdy neměly ohřát na více než 50 °C. Pro skladování baterie na dobu delší, než měsíc je doporučeno nabít články na 3,8 V. Uložení baterií při úplném nabití snižuje jejich životnost. Li-Po baterie se mohou při provozu nafukovat, v tomto případě by se měly ihned přestat používat, hrozí požár či exploze. Je doporučeno používat při nabíjení a skladování LiPo-SAFE sáčky, pro větší bezpečnost. [11].

2.5.3 Spínaný stabilizátor napětí

Z důvodu úspory prostoru a hmotnosti byla použit jedna baterie pro napájení servomotorů i pro napájení Raspberry Pi. Pro převod z 11 V na stálých 5 V pro Raspberry byl použit spínaný stabilizátor napětí FOXY UBEC 3 A [27]. Na vstupu tohoto regulátoru se může objevit 6-25 V a na výstupu je stálých 5 V nebo 6 V. Výstup lze přepínat pomocí zkratovací přepojky. Stabilizátor nabízí stálý výstupní proud 3 A, což je dostatečné i pro plně zatížené Raspberry. Napájení Raspberry je realizováno standardním micro USB konektorem.



Obrázek 22 - Spínaný stabilizátor FOXY UBEC 3A [27]

2.5.4 Převodník USB2Dynamixel

Pro ovládání servomotorů Dynamixel AX-12+ byl použit USB převodník USB2Dynamixel. Převodník má jako vstup USB port a má tři výstupy, sériový konektor, 3pinový konektor a 4pinový konektor. Na převodníku je přepínač, který nastavuje režimy komunikace na TTL, RS-485, RS-232. Při realizaci robota byl zvolen režim TTL a 3pinovým konektorem byl připojen výstup na serva.

2.5.5 Napájecí deska

Jak už bylo řečeno, servomotory Dynamixel AX-12+ se zapojují za sebe jako řetěz. Protože je technicky nemožné zapojit všechny servomotory za sebou, musela se použít napájecí deska. Byla vyrobena vlastní deska, na kterou je napojena baterie. Deska má 7 výstupů, které umožňují připojit 6 trojic (trojice jsou připojeny za sebou) servomotorů a jeden vstup na data z USB2Dynamixel.

3 Matematický model ovládání robota

Při rešerši bylo nalezeno několik volně dostupných řešení ovládání robota Hexapod. Firma Lynxmotion používá pro ovládání svých robotů „Phoenix code“ a firma TrossenRobotics tento zdrojový kód upravila pro svého robota PhantomX. Tato diplomová práce vychází ze zdrojových kódů a matematického modelu pro robota PhantomX. Vše je volně dostupné na GitHub.com. Zdrojové kódy jsou napsané v jazyce C++.

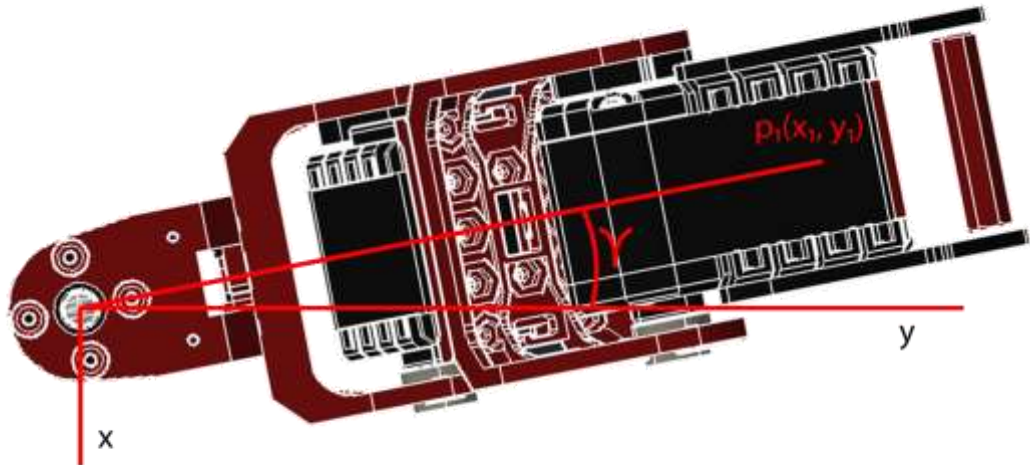
3.1 Inverzní kinematika končetin

Cílem je spočítat kloubové proměnné (úhly) všech tří servomotorů tak, aby se nášlapná plocha dostala na souřadnice $p_1(x_1, y_1, z_1)$. Úhel pro coxa se počítá podle 3.2, kde $atan2$ je definován podle 3.3. Pokud je $\gamma = 0$ přepokládá se, že servo je v úhlu 150° . Tedy ve středové poloze. Pro kladné γ se úhel serva zvyšuje, otáčí se po směru hodinových ručiček (Obrázek 23 - Úhel kyčelního kloubu) [13].

$$\gamma = -atan2(x_1, y_1) \quad (3.1)$$

$$\gamma_{coxa} = 150^\circ + \gamma \quad (3.2)$$

$$atan2(y, x) = \begin{cases} \arctan \frac{y}{x} & x > 0 \\ \arctan \frac{y}{x} + 180^\circ & y \geq 0, x < 0 \\ \arctan \frac{y}{x} - 180^\circ & y < 0, x < 0 \\ 90^\circ & y > 0, x = 0 \\ -90^\circ & y < 0, x = 0 \\ \text{nedefinováno} & y = 0, x = 0 \end{cases} \quad (3.3)$$



Obrázek 23 - Úhel kyčelního kloubu [13]

Pro výpočet úhlů femur a tibia se uvažuje situace zobrazená na obrázku 24. Nejprve se počítá L_1 a L_2 podle 3.4 a 3.5. Užitím kosinové věty lze vypočítat kloubové proměnné pro femur a tibia. Předpokládá se, že servomotory byly ve výchozí středové poloze (150°) viz. obrázek 24. Při zvětšení úhlů γ_{femur} a γ_{tibia} dochází k posunu bodu p_1 v kladném směru z . Pro výpočet $offset_{femur}$ a $offset_{tibia}$ se uvažují kloubové proměnné v pozici 150° . Konstanty použité ve výpočtech IK končetin jsou zobrazeny v tabulce 2. [13]

$$L_1 = \sqrt{x_1^2 + y_1^2} - L_{coxa} \quad (3.4)$$

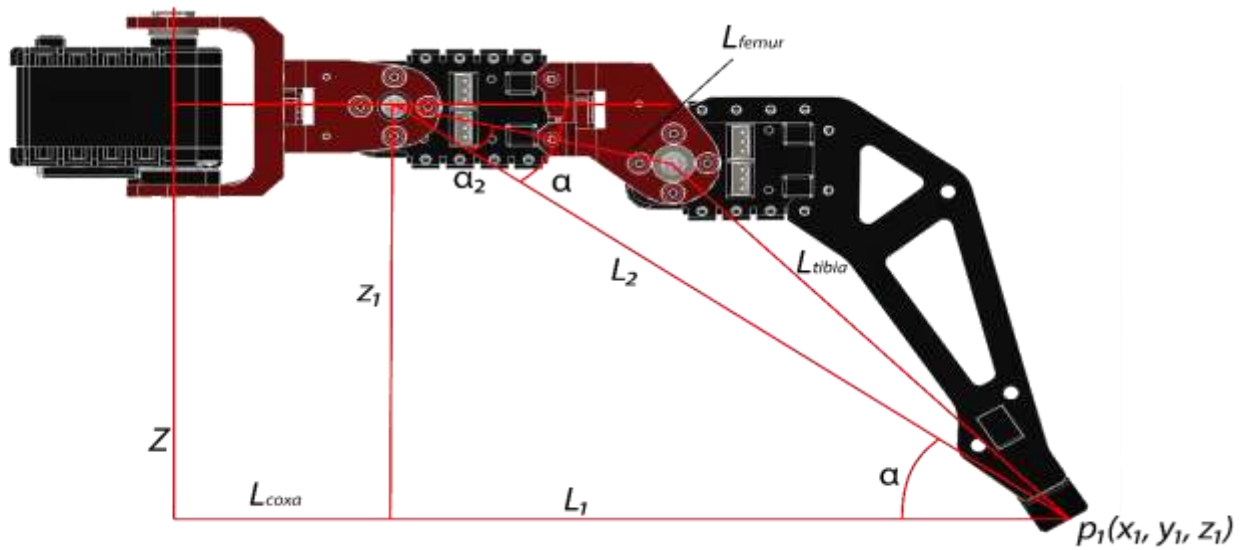
$$L_2 = \sqrt{L_1^2 + z_1^2} \quad (3.5)$$

$$\alpha = \text{atan2}(z_1, L_1) \quad (3.6)$$

$$\alpha_2 = \arccos\left(\frac{L_{femur}^2 + L_2^2 - L_{tibia}^2}{2 * L_{femur} * L_2}\right) \quad (3.7)$$

$$\gamma_{femur} = 150 - (\alpha_2 - \alpha + offset_{femur}) \quad (3.8)$$

$$\gamma_{tibia} = 150 - \arccos\left(\frac{L_{tibia}^2 + L_{femur}^2 - L_2^2}{2 * L_{femur} * L_{tibia}}\right) + offset_{tibia} \quad (3.9)$$



Obrázek 24 - Úhel stehenního a holenního kloubu [13]

Konstanta	Hodnota
L_{coxa}	54 mm
L_{coxa}	66 mm
L_{coxa}	137 mm
$offset_{femur}$	13.7°
$offset_{tibia}$	147.6°

Tabulka 2 - Konstanty použité při výpočtu IK končetin

V tabulce Tabulka 2 jsou délky kyčelního, stehenního a holenního spoje používaného při výpočtech inverzní kinematiky končetin. K výpočtu jsou potřeba také úhly, které vznikly konstrukcí končetiny.

3.2 Omezení končetin

Při chodu robota se berou v úvahu omezení pohybu končetin. Tyto omezení řeší kontrolér. První omezení se aplikuje na základě konstrukce končetiny. Úhel je omezen tak, aby servomotor nenarazil na svoji konstrukci.

Další omezení se týká celé končetiny. Ta je omezena na svůj pracovní prostor, ve kterém je schopná inverzní kinematika počítat souřadnice. Tyto omezení jsou kombinací

fyzických a matematických limitů. Další omezení jsou také aktuální pozice ostatních končetin.

3.2.1 Konstrukční omezení končetin

Servomotory Dynamixel AX-12+ používané v této práci mají pole působnosti 300°. Mechanická konstrukce robota neumožňuje servomotorům využít celý rozsah. Pokud by z nějakého důvodu kontrolér poslal úhel tam, kam se fyzicky servo nemůže dostat z důvodu mechanického limitu, servo by narazilo do konstrukce. Servomotory však mají svůj vlastní kontrolér, který má naprogramované bezpečnostní prvky. Pokud by servo narazilo při plné síle, došlo by ke stavu chyby a kontrolér by odpojil motor do restartu celého zařízení. I přes to, že servo má tyto opatření, je lepší se těmto chybám zcela vyhnout [13].

Nejlepším způsobem, jak tento problém vyřešit je implementace omezení na výstupy do servomotoru. Aby tyto omezení neovlivňovaly další výpočty, měly by být implementovány až na závěr. Poslední výpočty kontroléru jsou inverzní kinematika všech končetin, které přepočítávají pozici servomotorů v rozsahu 0 až 1023. Po těchto výpočtech se aplikuje toto omezení.

Omezující hodnoty lze získat pokusným způsobem. Jelikož servomotory Dynamixel umožňují uživateli číst aktuální hodnotu, není těžké určit omezující hodnoty. Výstupní hodnoty na servomotory, kde se aplikuje omezení, již nejsou uloženy v úhlech. Jsou uloženy jako hodnota pro serva, a proto se omezující hodnoty nemusejí přepočítávat. V následující tabulce (tabulka 3) je příklad omezujících hodnot pro konkrétní servomotory. [13]

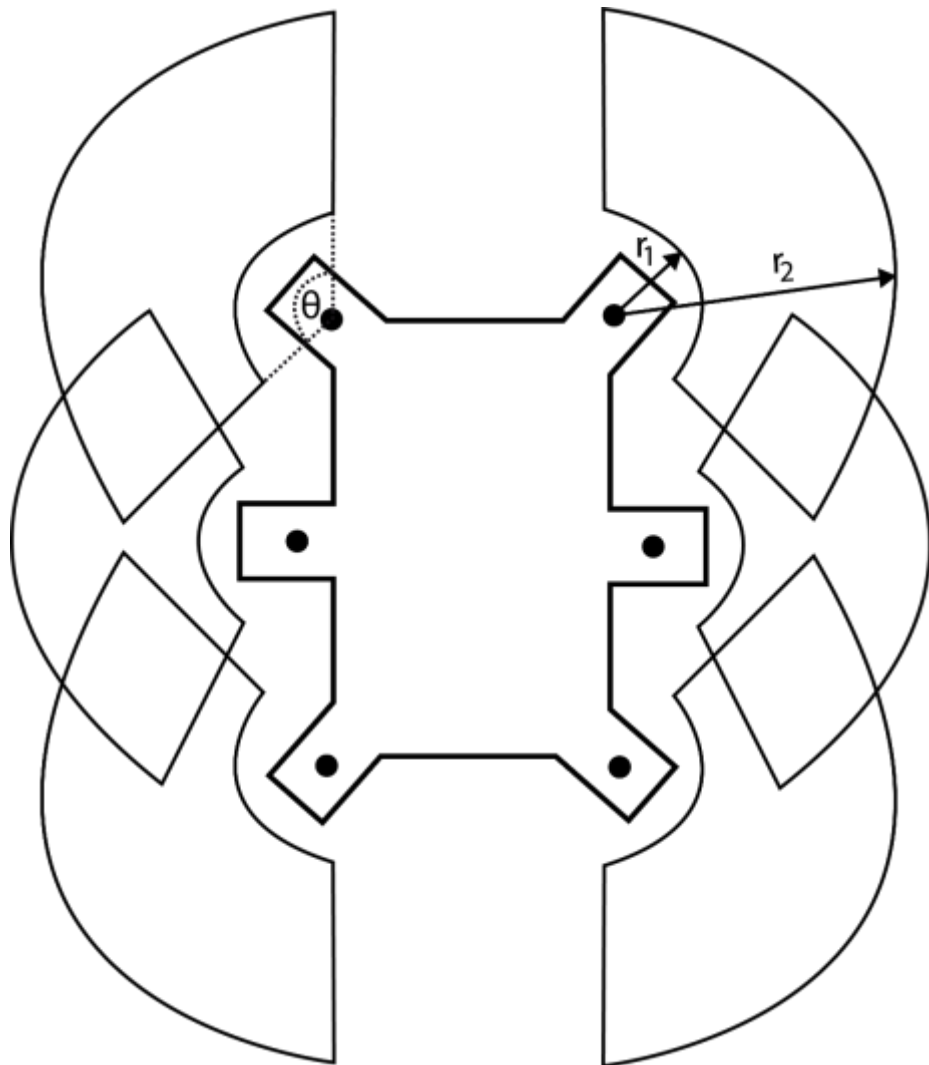
Servo	Max	Min
RRCoxa	850	180
RRFemur	850	170
RRTibia	880	280
LRCoxa	850	180
LRFemur	850	170
LRTibia	700	130

Tabulka 3 - Konstrukční omezení servomotorů

Jelikož jsou všechny končetiny na jedné straně identické, jsou hodnoty pro druhou stranu stejné. Hodnoty jsou v rozsahu 10 bitů.

3.2.2 Pohybové omezení končetin

Rozměry částí coxa, femur a tibia jsou pevně dány, a proto končetina nedosáhne větší vzdálenosti než součet těchto délek. Pohyb končetin vytváří kruhovou výseč kolem kyčelního kloubu (coxa). Výseč je omezena konstrukčním omezením kyčelního kloubu. Ve vnější části není schopná inverzní kinematika spočítat úhly servomotorů, protože taková místa jsou nedosažitelná. Také existují místa v blízkosti kyčelního kloubu, které jsou končetinou nedosažitelné z důvodu rozměrů jednotlivých částí končetiny. Toto vytváří opět kruhovou výseč. [13]



Obrázek 25 - Pohybové omezení končetin [13]

Na předchozím obrázku je tato situace znázorněna. Uvažují se pouze dva rozměry. Pozice končetiny je ve stejné výšce jako je tělo robota.

Pro každou výšku robota budou tyto vyseče obdobné. Výškou robota je myšleno to, že souřadnice y končetiny je rozdílná od souřadnice y těla. Tělo je několik jednotek nad zemí. Celkem všechny omezení vytvářejí pracovní prostor končetiny robota.

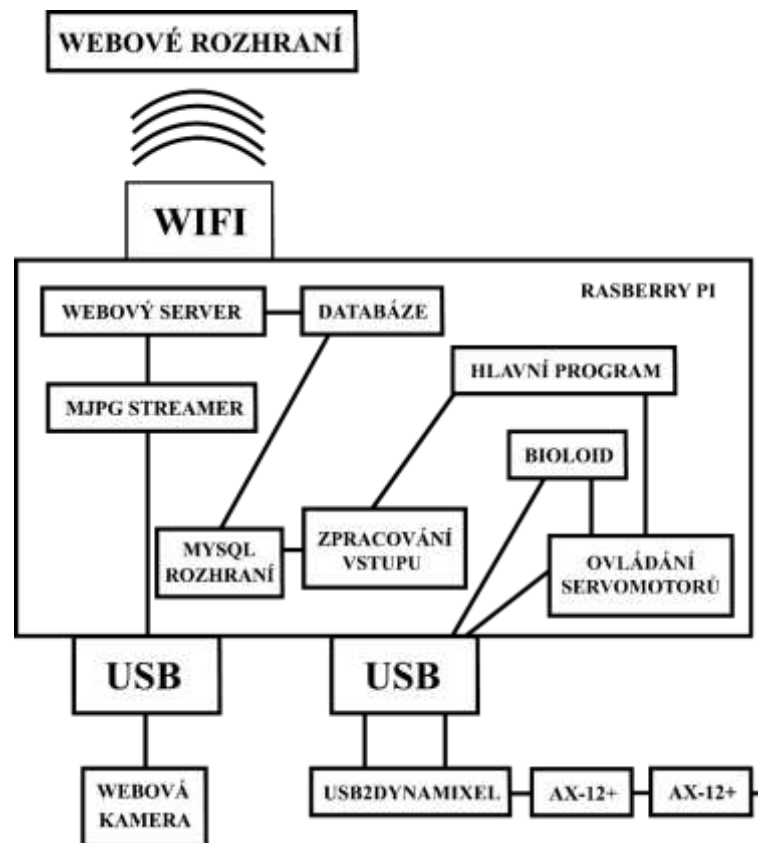
V jistém případě se může dostat konec končetiny pod kyčelní kloub i dále. Implementace algoritmu inverzní kinematiky končetin tyto polohy nespočítá a končí chybou. Jedná se o známý problém, ale zatím není třeba tento problém řešit. Tyto polohy nemají pro pohyb žádné využití, a proto v algoritmu jsou nastaveny limity, viz Obrázek 25. [13]

Z obrázku 25 je dále patrné, že se jisté prostory překrývají. To představuje další problém, kde se dvě končetiny mohou srazit. Vzhledem k tomu, že končetiny mají nějakou šířku (objem), jsou s ohledem na to vytvořeny další omezující podmínky, který ořezávají jednu souřadnici.

4 Realizace ovládacího software

K realizaci ovládacího software bylo potřeba dostatek výkonu pro výpočty, dostatek USB portů a programovatelných pinů pro komunikaci s periferiemi a možnost WIFI připojení pro ovládání přes webové rozhraní. Všechny tyto možnosti nabízí Raspberry Pi. Byla zvolena poslední verze Raspberry Pi 3 model B, protože nabízí nejlepší poměr cena/výkon a integrovaný WIFI a bluetooth modul, viz. kapitola 2.5.1. Na Raspberry byl nainstalován oficiální operační systém Raspbian Jessie.

Celý software je rozdělen do samostatných celků. Hlavní program je napsán v C++ a je dále rozdělen do několika tříd a souborů. Kód, který řeší výpočty souřadnic a kloubových proměnných je uložen v Phoenix_Code.cpp. Na obrázku Obrázek 26 je vyznačen jako hlavní program. Dále na Raspberry běží LAMP server. Apache s PHP řeší ovládání z webového rozhraní a MySQL server slouží k ukládání a předávání dat mezi samostatnými celky. Dále zde běží služba MJPG Streamer pro přenos videa. Raspberry bylo také nakonfigurováno jako přístupový bod k bezdrátové síti WIFI.



Obrázek 26 - Schéma stavby systému

4.1 Hlavní program

Hlavní program pro ovládání pohybu robota vychází ze zdrojových kódů pro roboty PhantomX od firmy Trossen Robotics, dostupný na GitHub.com. Program je napsaný v jazyce C++. Zdrojový kód byl celý upraven a zjednodušen. Je logicky uspořádán do jednotlivých souborů, které řeší specifické funkce.

Hlavní program používá Dynamixel SDK verze 1, který pracuje s USB převodníkem USB2Dynamixel. Dokumentace, ze které jsem vycházel, je dostupná online [14]. Dále program používá mysql knihovnu, která se sestavuje společně se zdrojovými kódy přímo na Raspberry pomocí programu Make.

Nejdůležitější část programu řeší pohyb robota na základě příkazů. Jsou zde naprogramovány pohybové algoritmy, výpočet všech souřadnic končetin i těla, inverzní kinematika, optimalizační algoritmy a také kontrola mechanických limitů. Řídí ostatní části programu.

Další část programu řeší zpracování vstupů. Zpracovává předaná data a připravuje je pro použití v hlavním programu. Samotná data jsou z webového rozhraní uložena v MySQL databázi. Odtud je načítá třída s MySQL rozhraním a předává je právě výše zmíněné třídě (na obrázku 26 je vyznačena jako „zpracování vstupů“). Toto řešení má výhodu v tom, že lze snadno rozšířit. Lze napsat třídu, která bude získávat data například z bezdrátového ovladače, jako je gamepad.

Velmi důležitou částí je také třída, která se stará o ovládání servomotorů Dynamixel AX12+. Při dodržení stejného rozhraní lze vytvořit třídu i pro jiné servomotory. Třída se dále stará o kontrolu napětí na jednotlivých servomotech a případě nízkého napětí vypne napájení pro ochranu baterie. Nutno podotknout, že baterie je stále využívána počítačem Raspberry Pi a v případě vybité baterie se musí včas odpojit. Třída dále počítá rychlost otáčení pro jednotlivé servomotory. Tento výpočet řeší jeden zásadní problém. Servomotory se pohybují do zadaného úhlu s konstantní rychlostí. Pokud by měly všechny servomotory stejně nastavenou rychlost, došlo by k desynchronizování. Jestliže bude jedno servo natočené blíže k cílovému úhlu a druhé dál, první servo se stejnou rychlostí dostane do cílového úhlu dříve než druhé. Ve výsledku by se pak končetina nepohybovala správně a samotná chůze robota by nebyla možná.

Samotná třída při pohybu neposílá data přímo do servomotorů, ale předává data třídě Bioloid.

Třída Bioloid slouží pro komunikaci a ovládání servomotorů AX12+. Při inicializaci vytvoří pole pro id, aktuální pozici a následující pozici všech servomotorů a vytvoří nové vlákno. To se stará o plynulou aktualizaci servomotorů včetně správné rychlosti. Komunikace se servomotory probíhá pomocí složeného packetu pro synchronizovaný zápis.

4.2 Příprava a konfigurace Raspberry

Pro snadné ovládání a konfiguraci Raspberry z prostředí Windows byl na Raspberry nainstalován balíček xrdp. Umožňuje uživateli připojit se z Windows protokolu RDP (Remote Desktop Protocol) do prostředí Linux. Xrdp používá Xvnc. Pro instalaci xrdp bylo nejprve potřeba odstranit RealVNC server, který je již součástí OS. Poté byl nainstalován VNC server, který pracuje s xrdp:

```
sudo apt-get install tightvncserver  
sudo apt-get install xrdp
```

Dále pro snadnou práci se soubory a modifikaci souborů z prostředí Windows byl na Raspberry nainstalován Samba server, který se instaluje příkazem [16]:

```
sudo apt-get install samba samba-common-bin
```

Dále byl otevřen konfigurační soubor (/etc/samba/smb.conf), kde byla povolena podpora windows, nastavena pracovní skupina a pomocí následující konfigurace byla nasdílena domovská složka administrátorského profilu [16].

```
[hexorhome]  
comment= Hexor Home folder  
path=/home/hexor  
browseable=Yes  
writeable=Yes  
only guest=no  
create mask=0777  
directory mask=0777  
public=no
```

Konfigurace umožňuje číst a editovat soubory jen přihlášeným uživatelům. Pomocí následujícího příkazu byl přidán účet, potřebný k přihlášení do sdílené složky. Účet byl pro jednoduchost nastaven stejný jako pro přihlášení do Raspberry [16].

```
Smbpasswd -a hexor
```

4.3 LAMP server

Na Raspberry byl nainstalován Apache server, PHP a MySQL databáze. Toto řešení bylo zvoleno pro svou jednoduchost. Apache a PHP umožňují se na Raspberry připojit přes webové rozhraní, odkud lze robota kompletně ovládat. MySQL databáze umožňuje ukládání dat a příkazů od uživatele a hlavní program pak může tyto data jednoduše číst.

Pro instalaci Apache a PHP5 byly použity následující příkazy (s root):

```
apt-get install apache2  
apt-get install php5  
apt-get install libapache2-mod-php5
```

Kořenový www adresář se naistaloval od `/var/www/`. Dále byl nainstalován MySQL server pomocí těchto příkazů:

```
apt-get install mysql-server  
apt-get install php5-mysql
```

Ke snadné správě databáze byl nainstalován PHPMyAdmin:

```
apt-get install phpmyadmin
```

Serveru apache byl přidán konfigurační soubor tak, že se příkazem

```
nano /etc/apache2/apache2.conf
```

otevře konfigurační soubor apache2 a na konec souboru se vloží

```
Include /etc/phpmyadmin/apache.conf
```

Dále bylo zjištěno, že MySQL ukládá defaultně data do InnoDB. To zpomalovalo značně komunikaci, a proto byly do konfiguračního souboru (/etc/mysql/my.cnf) přidány tyto nastavení (na konec sekce [mysqld]):

```
ignore_builtin_innodb  
default_storage_engine=MyISAM
```

4.4 Software pro video stream

Pro zachytávání videa a zobrazování ve webovém rozhraní byl nejprve použit program motion. Ten se později ukázal jako nedostatečný. Motion je určený pro detekci pohybu a výstupy z kamer předzpracovává. Nedokáže zobrazovat dostatečné množství snímků za sekundu a obraz není plynulý.

Proto byl použit balíček MJPG Streamer, který slouží výhradně k přenosu videa přes webové rozhraní. Jedná se o open source software, který není dostupný z oficiálních repozitářů. Nejprve byl nainstalován balíček ffmpeg, k získání videa či nahrání videa z kamery. V novější verzi Raspbianu už tento balíček není potřeba.

```
apt-get install ffmpeg
```

Následujícími příkazy bylo potřeba doinstalovat nezbytné programy do Raspberry:

```
apt-get install libv4l-dev  
apt-get install libjpeg8-dev  
apt-get install subversion  
apt-get install imagemagick
```

kde libv4l-dev a libjpeg8-dev jsou potřebné závislosti, subversion slouží k stahování a instalování open source software a imagemagick k instalaci. Dále byly použity příkazy k samotnému stažení a instalaci balíčku MJPG Streamer:

```
svn co https://svn.code.sf.net/p/mjpg-streamer/code/  
cd /home/pi/code/mjpg-streamer/  
make USE_LIBV4L2=true clean all  
make DESTDIR=/usr install
```

Spustit program lze pomocí následujícího příkazu. Příkaz má několik parametrů, které nastavují např. rozlišení a počet snímků za sekundu.

```
mjpg_streamer -i "/usr/lib/input_uvc.so -d /dev/video0 -y -r 1024x768 -f 32 -q 60" -o "/usr/lib/output_http.so -p 8090 -w /var/www/mjpg_streamer"
```

Program byl dále nastaven jako služka, která se spouští po startu zařízení.

4.4.1 Použitá webová kamera

Pro záznam obrazu byla použita standardní a dostupná webová kamera Trust SpotLight Webcam Pro. Kamera dokáže snímat obraz v maximálním rozlišení 1280×1024 . Připojuje se k Raspberry přes USB port. Při zhoršeném osvětlení lze manuálně regulovat intenzitu přisvitu či přisvit vypnout.



Obrázek 27 - Webová kamera Trust SpotLight Webcam Pro [31]

4.5 Konfigurace Raspberry do režimu WIFI AP

Pro ovládání robota je potřeba, aby Raspberry i klientské zařízení, kde poběží webové rozhraní, byly připojeny na jedné síti WIFI. Případně lze nastavit veřejnou IP adresu, či VPN pro ovládání Raspberry přes internet.

K dosažení vždy dostupné sítě bylo Raspberry nakonfigurováno jako přístupový bod k bezdrátové síti WIFI. Přístupový bod byl zabezpečen autentizačním a šifrovacím algoritmem IEEE 802.11i, známý jako WPA2, který používá blokovou šifru AES. Raspberry PI3 sice obsahuje integrovanou WIFI anténu, ale tato anténa byla použita pouze k připojování k okolním sítím. Pro AP byla použita externí anténa připojená přes

USB. Z testovacích důvodů byl použit standardní miniaturní WIFI USB adaptér Edimax EW-7811Un. Výhodou toho řešení je, že lze použít silnější USB adaptér či jinou anténu, pro zajištění většího dosahu. Z experimentálních důvodů byly obě sítě (integrovaná a USB wifi) propojeny a Raspberry tak překládá adresy z jedné sítě do druhé. Instalace proběhla podle podrobného návodu [15].

4.6 Webové rozhraní s video streamem

Rozhraní bylo napsáno v PHP 5.6. Webová aplikace obsahuje několik souborů. Pro rychlost a jednoduchost úlohy nebylo potřeba žádného PHP frameworku. Pro javascript funkce a komunikaci se serverem byla použita knihovna jQuery a Ajax. Dále pro snadnější kódování webové aplikace byl použit modul Pure.css včetně Responsive Grids. Aplikace je primitivně ošetřena tak, aby se na ní připojilo pouze jedno klientské zařízení a nedocházelo tak ke kolizi ovládání.

Ve webové aplikaci je v iframe zobrazen real-time stream z webové kamery. Ten je dostupný pro neomezený počet klientů díky programu mjpg streamer také na samostatné adrese <http://hexorv3:8090/?action=stream>. Více viz kapitola 4.4 . Dále je ve webovém rozhraní 9 tlačítek, kterými lze nastavovat funkce a vlastnosti robota. Jednotlivé stavy robota jsou zobrazovány v tabulce pod tlačítky.

První tlačítko přepíná mezi pěti pohybovými algoritmy. Přepnout lze pouze pokud robot není v pohybu. Dalším tlačítkem lze přepínat mezi dvěma styly pohybu. Jeden styl umožňuje uživateli otáčet robota kolem své osy a pohybovat se přímočaře nebo do stran, například po diagonále. Druhý styl umožňuje robotovi jít rovně, zatáčet, či se otáčet kolem své osy. Tento pohyb lze přirovnat k pohybu vozidlům s pásovým podvozkem. Dále lze z webového rozhraní softwarově vypnout všechny servomotory a šetřit tak baterii. Servomotory budou sice pod napětím, ale nebudou držet svou pozici. Také lze regulovat rychlost pohybu a výšku robota nad zemí.

Další zajímavou funkcí je přepnutí z chůze do dvou funkcí nastavování pozice těla. První umožňuje nastavovat tělo do všech možných pozic vzhledem k postavení nohou. Tělo zůstává vždy ve vodorovné pozici. Tělo lze posunout dopředu, dozadu, či do stran. Druhá funkce umožňuje tělo naklápět, a to jak na oba boky, tak dopředu i dozadu. V obou funkcích lze tělo natočit proti a po směru hodinových ručiček

a regulovat výšku nad zemí. Dále lze ve webovém rozhraní jedním tlačítkem prodloužit robotu krok a dalším tlačítkem měnit výšku zdvihu končetin při chůzi. Posledním tlačítkem lze robota přepnout do „balančního“ režimu. Tento režim je spíše experimentální, robot se při chůzi houpe a simuluje tak další pohyby živočichů.

Ve webovém rozhraní jsou dále naprogramovány dva joysticky, díky kterým lze ovládat všechny výše popsané funkce a samotný pohyb robota. Byla na to použita knihovna niplejs. Joysticky je možné ovládat pomocí myši, nebo prsty na dotykovém zařízení.

4.7 Nastavení videa a hlavního programu jako služba

Pro snadné, a hlavně praktické ovládání bylo potřeba nastavit spouštění 2 služeb po startu systému. První služba zapíná mjpg streamer a druhá zapíná hlavní C++ program. Bylo zkoušeno několik variant, jak spouštět hlavní program při startu systému, žádný ale nefungoval správně. Problém byl s načasováním spouštění služby. Vhodným řešením bylo použití „systemd.“ Při konfiguraci lze nastavit typ služby, která určuje jak a kdy se má služba spustit. Systemd je dostupný u operačního systému Raspbian, pouze ve verzi Jessie.

Při nastavování hlavního programu jako služba bylo postupováno následovně. Nejprve byl vytvořen soubor nové služby přímo ve složce systemd:

```
sudo nano /lib/systemd/system/hexor.service
```

do tohoto souboru uložena následující konfigurace,

```
[Unit]
Description=Hexar robot control service
After=multi-user.target

[Service]
Type=idle
ExecStart=/home/hexor/hirnsal/hexorv3/MainProgram.hexor

[Install]
WantedBy=multi-user.target
```

kde v ExecStart je nastavena absolutní cesta k hlavnímu programu. Type=idle zajistí, že služba bude spuštěna až po kompletním naběhnutí operačního systému. Dále bylo potřeba vytvořenému souboru nastavit práva.

```
sudo chmod 644 /lib/systemd/system/hexor.service
```

Nakonec bylo potřeba zaregistrovat službu do systemd.

```
sudo systemctl daemon-reload
```

```
sudo systemctl enable hexor.service
```

Při vytváření služby pro stream videa se postupovalo stejně. Byl vytvořen soubor stream.service a do ExecStart uložen následující příkaz.

```
ExecStart=/usr/bin/mjpg_streamer -i "/usr/lib/input_uvc.so  
-d /dev/video0 -r 640x480 -f 60 -q 60" -o "/usr/lib/output_http.so  
-p 8090 -w /var/www/mjpg_streamer"
```

Dále je možné služby spravovat dalšími příkazy. Vypnout automatické spouštění lze provést příkazem

```
sudo systemctl disable hexor.service
```

či zjistit stav služby.

```
sudo systemctl status hexor.service
```

Více příkazů, konfigurací a informací lze nalézt v [17].

5 Testování, ukázková úloha a výsledky

Robot byl otestován v různých terénech. Nejprve byly testovány pohybové algoritmy. Dále byla vytvořena ukázková úloha, kde robot díky čtyřem vzdálenostním senzorům detekuje překážky. Na základě dat ze senzorů pak může reagovat. Také bylo testováno samotné webové rozhraní a byla analyzována spotřeba energie celého robota.

5.1 Testování pohybových algoritmů v terénu

Při testování pohybových algoritmů byl robot nejprve testován na rovném a hladkém povrchu v podobě plovoucí podlahy. Zde fungovaly bez problému všechny pohybové algoritmy. Díky protiskluzovým výstupkům na koncích končetin se robot nesmekal. Rychlosti a sekvence pohybu odpovídaly teoretickým poznatkům v předchozích kapitolách. Pohybový algoritmus tripod se zapnutým prodlouženým krokem umožňuje robotu chodit velmi rychle.

Dále byl robot testován na travnatém terénu. Nejprve v trávě vysoké do 8 cm. Zde byl potřeba nastavit velký zdvih nohou při pohybu, aby nedocházelo k zaseknutí končetin a také byla zvednuta celková výška těla robota. I přes to občas docházelo k zaseknutí končetiny ve velmi husté trávě. Při nízké výšce těla tráva robota brzdila a končetiny částečně prokluzovaly.

Dále byl robot testován ve vysoké trávě vyšší než 8 cm. Byla nastavena maximální možná výška robota, která ještě umožňovala pohybovat končetinami, viz. obrázek 28. Test spočíval v překonání travnatého porostu vyššího než robot samotný. Všechny pohybové algoritmy trávu překonaly, tripod však měl největší problém, protože tři končetiny na zemi neměly takovou stabilitu a sílu ke snadnému překonání husté a vysoké trávy.



Obrázek 28 - Test robota ve vysoké trávě

Dalším testovacím terénem byly drobné kamínky a také štěrky. Při chůzi se končetiny lehce zarývaly, ale stabilitu robota to skoro neovlivnilo. Všechny pohybové algoritmy fungovaly dobře. Nebylo nutné příliš upravovat výšku těla robota.



Obrázek 29 - Test robota na štěrku

Posledním testovacím terénem byl jemný písek. Končetiny se při chůzi bořily, jak může být viděno na následujícím obrázku (29). I přes zaboření končetin a nerovnosti pískového terénu byl robot schopen se plynule pohybovat.



Obrázek 30 - Test robota na písku

Všechny testy a pohybové algoritmy byly nahrávány na video. Dále byl dokumentován test rychlosti a úpravy pozice těla. Všechny videa jsou dostupná zde [32].

5.2 Ukázková úloha vyhýbání se překážkám

Pro detekci vzdálenosti byly použity 4 senzory HC-SR04. Jedná se o ultrazvukové senzory, které pracují na frekvenci 40 kHz, což je 2krát více než hranice slyšitelnosti pro lidské ucho. Senzory pracují při napětí 5 V. Zde byl první problém s připojením k Raspberry. GPIO piny na Raspberry totiž pracují na 3,3 V. Změnu napětí lze provést vhodným regulátorem, či pouze pomocí odporů. Ovšem nevýhodou tohoto řešení je velké množství pinů, které by bylo potřeba redukovat z 5 V na 3,3 V. Každý sensor má 4 piny (VCC, Trig, Echo, GND) a sensorů byly použity také čtyři. Senzory HC-SR04 skvěle

pracují s mikrokontrolery Arduino a mají stejné operační napětí 5 V, a proto bylo zvaženo použití mikrokontroleru Arduino.

K dispozici byl mikrokontroler Arduino MEGA, který byl použit. Tento kontrolér je sice pro použití pouze na 4 senzory zbytečně veliký, byl však jediný k dispozici. Arduino MEGA má 54 programovatelných digitálních pinů a obsahuje 16 MHz krystal. Piny jsou připraveny pro standardní konektory a není potřeba piny pájet.



Obrázek 31 - Ultrazvukový senzor HC-SR04



Obrázek 32 - Vývojová deska Arduino MEGA

Pro programování Arduina bylo staženo oficiální vývojové prostředí Arduino IDE. Při instalaci se automaticky naistalují i potřebné ovladače. Stačí pak Arduino MEGA zapojit do počítače přes USB konektor a začít programovat v podporovaném jazyce C++. Pro měření ze senzorů byla napsána metoda, která pro aktuální senzor získá vzdálenost od překážky a uloží hodnotu do pole všech vzdáleností.

```
void readSensor()  
{  
    digitalWrite(trigPins[sensorIndex], LOW);  
    delayMicroseconds(2);  
    digitalWrite(trigPins[sensorIndex], HIGH);  
    delayMicroseconds(10);  
    digitalWrite(trigPins[sensorIndex], LOW);  
    duration = pulseIn(echoPins[sensorIndex], HIGH, timeout);  
    distances[sensorIndex] = duration*0.034/2;  
}
```

Obrázek 33- Měření vzdálenosti ultrazvukovým senzorem

V dokumentaci je uvedeno, že pro aktivování senzoru je potřeba nastavit trigger pin do logické 1 po dobu 10 mikrosekund. Následně senzor odešle 8 ultrazvukových vln, které se šíří rychlostí zvuku. Echo pin pak bude v logické 1 po dobu, kterou trvalo vlnám

dorazit k překážce, odrazit se a vrátit se zpět do senzoru. Pro získání vzdálenosti v centimetrech se použil následující vzorec:

$$s = \frac{t \cdot v}{2} \quad (5.1)$$

Kde t je doba v mikrosekundách, po kterou byl echo pin v logické 1. v je rychlost šíření zvuku. Součtin je vydělen dvěma, protože zvuk po dobu t urazil měřenou vzdálenost 2x. Rychlost byla převedena na jednotku centimetr za mikrosekundu.

$$v = 0,034 \text{ cm}/\mu\text{s} \quad (5.2)$$

Do funkce `pulseIn()` byl dále zaveden `timeout`. Pokud bude měření trvat déle než 5,8 milisekund, měření se ukončí. Tato doba odpovídá vzdálenosti přibližně 200/2 cm. Důvod zavedení časového limitu je prostý. Není potřeba čekat na odraz od předmětu, který je příliš daleko. Odraz by také nemusel vůbec přijít. Čekání by zbytečně zdržovalo program v Arduinu.

Dále bylo potřeba propojit Arduino s Raspberry. Byla zvolena sériová komunikace UART. Obě zařízení mají několik pinů pro UART, ale také podporují sériovou komunikaci přes USB. Rozhodl jsem se zvolit variantu s USB, řeší problém s rozdílným operačním napětím na pinech a také připojení přes USB zajistí napájení Arduina. Výhodou je také rychlé připojení a odpojení, než rozpojovat 4 piny. Piny by se mohly zapojit nesprávně, či se jeden odpojit a mohlo by tak dojít k poškození zařízení. Byla zvolena přenosová rychlost 115 200 bit/s.

Celý program pro Arduino nevykonává žádné náročné funkce. V hlavní smyčce se čte příkaz ze sériové komunikace a pokud je nějaký rozpoznán, tak na něj Arduino reaguje. Dále se ve smyčce čte vzdálenost z jednoho senzoru. V další smyčce se čte z následujícího senzoru atd. Čtení ze všech čtyřech senzorů v jedné smyčce by trvalo dlouho a příkazy ze sériové komunikace by nebyly tak rychle obslouženy. Komunikace probíhá v textové podobě. Každý příkaz i odpověď je zakončen znakem pro nový řádek `'\n'`. Arduino reaguje na příkazy „start, stop, get, ping“. Odpověď na „get“ je ve tvaru čtyř čísel oddělených znakem „:“. Odeslány jsou aktuálně uložené vzdálenosti, které se aktualizují jednou za 4 cykly.

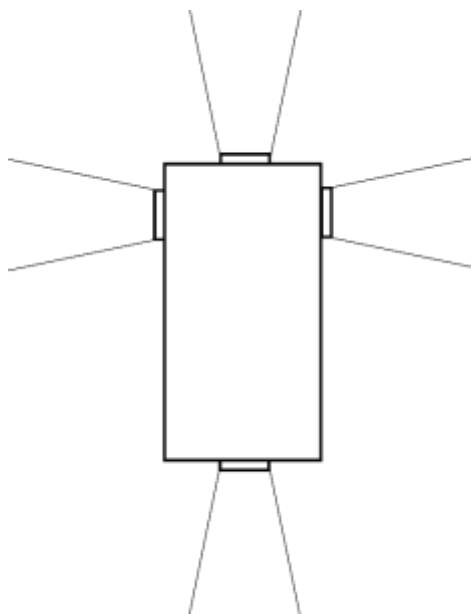
Na Arduinu byla testována průměrná doba jednoho cyklu, protože senzory vyžadují pro měření poměrně dlouhou dobu. Byla napsána jednoduchá funkce, která průměruje dobu 100 cyklů a vypisuje hodnotu v mikrosekundách. Pokud je před všemi

senzory umístěna překážka ve vzdálenost 5 cm, pak je průměrná doba cyklu okolo 700 mikrosekund. Při vypnutém limitu se doba cyklu může protáhnout až na 50 000 mikrosekund, v případě, kdy není v dosahu žádná překážka. Z těchto informací vyplývá, že měření vzdálenosti pomocí ultrazvuku není vhodné pro real-time aplikace. Tyto senzory navíc pro správné měření potřebují odrazovou plochu alespoň $0,5m^2$. Dále by plocha měla být hladká a snímaná pod maximálním úhlem 30° . Překážky jako jsou nohy židlí a stolů jsou pro senzory HC-SR04 prakticky nezjistitelné. Při zapnutém výše zmíněném limit, doba cyklu nepřekročí 5 900 mikrosekund.

Dále byla na Raspberry nainstalována knihovna WiringPi [18]. Nabízí jednoduché ovládání sériové komunikace přes UART či USB. Byl napsán program v odděleném souboru UARTDriver.cpp, který po zavolání funkce zkontroluje příchozí data, které případně načte a poté odešle příkaz „get“, Příchozí data přečte až v dalším kroku. Tato varianta je výhodná, protože hlavní program nemusí čekat až arduino dokončí právě probíhající měření a pošle data. Je lepší mít o pár milisekund staré data, než tuto dobu nečinně čekat. Přijatá data se rozdělí podle znaku „:“ a převedené řetězce na čísla se uloží do globální struktury. S touto strukturou se pak dále pracuje.

Hlavní program byl ošetřen tak, aby se robot bezpečně zastavil před překážkou a nedovolil pokračovat dále. Limit by nastaven na 26 cm. Senzory byly umístěny do papírové krabičky a tu lze na suchý zip připnout na tělo robota. V krabičce je také uložena webová kamera. Díky tomu, že jsou senzory v krabičce umístěny dostatečně vysoko, nebude pohyb končetin ovlivňovat měření.

Vzhledem k tomu, že jsou použity pouze 4 senzory, které navíc měří vzdálenost pouze před sebou, vzniká mnoho slepých míst. Celá situace je znázorněna na následujícím obrázku (Obrázek 34).



Obrázek 34 - Rozmístění senzorů HC-SR04

Na obrázku Obrázek 34 je znázorněna krabička se čtyřmi senzory uloženými uvnitř. Pro čidla byly vyřezány otvory. V přední části krabičky je přední senzor a po stranách levý a pravý senzor. Na zadní straně krabičky je zadní senzor, který snímá prostředí za robotem. Pro kompenzaci slepých míst mezi levým a předním senzorem a také mezi pravým a předním senzorem byla napsána funkce, která dopočítává vzdálenost od překážky pod úhlem 45° . Předpokládá se, že překážka bude rovná dlouhá stěna. Pokud budou známy obě vzdálenosti (z předního a bočního senzoru), lze pomocí Pythagorovy a Sinovy věty dopočítat vzdálenost od překážky pod úhlem 45° . Tedy středem úhlu, který svírají oba senzory. Tato vzdálenost by byla užitečná, pokud by byl robot natočen šikmo k překážce (stěně) a mohl by tak lépe vyhodnotit situaci.

V praxi se ukázalo, že senzory nedokáží měřit vzdálenost pod velkým úhlem. Ani v případě, kdy oba senzory měří překážku pod úhlem 45° , nedokáží senzory určit vzdálenost. Situaci lze vyřešit pořízením dalších senzorů, či volbou vhodnějších. V této konfiguraci robot detekuje překážku, pouze pokud se k ní přibližuje čelem, bokem, či na ní couvá. Program byl ošetřen tak, aby neumožňoval uživateli pokračovat v pohybu směrem k překážce, pokud je dostatečně blízko. Tím pádem zamezí kontaktu s překážkou.

5.3 Spotřeba energie robota

Podle dokumentace je doporučeno Raspberry Pi poskytnout stálý proud 3 A v případě plného zatížení. Tento proud dokáže použitý stabilizátor bez problému poskytnout. Reálná spotřeba Raspberry byla naměřena od 650 do 750 mAh na 5 V zdroji. Spotřeba byla včetně zapojené kamery, Arduina, USB WiFi i převodníku USB2Dynamixel. Z toho webová kamera spotřebuje přibližně 120 mAh, Arduino přibližně 80 mAh a převodník 20 mAh. Spotřeba kompletně zapojeného Raspberry je okolo 3,25 Wh, baterie má kapacitu 28,9 Wh. V případě účinnosti stabilizátoru 90 % by samotné Raspberry vydrželo na baterii přibližně 8 hodin. Následně byla tato úvaha ověřena. Kompletně zapojené Raspberry bylo zapojeno do stabilizátoru a ten připojen na baterii. Z baterie byl s tímto zapojením odebírán proud 310 mAh a s kapacitou baterie 2600 mAh by toto zapojení teoreticky vydrželo přes 8 hodin.

Dále bylo měřeno zapojení samotných servomotorů na baterii. Podle dokumentace, servomotory AX-12+ odebírají při nečinnosti 50 mAh a celkem by teoreticky mělo 18 servomotorů odebírat 900 mAh. Při měření se ukázalo, že všechny servomotory odebíraly dohromady 650 mAh.

Servomotory Dynamixel AX-12+ mají maximální odebíraný proud 900 mAh. Pro všechny servomotory to vychází na 16,2 Ah a takto by na baterii vydrželi 10 minut. Servomotory, ale neodebírají maximální proud, ani všechny nepracují najednou. Při pohybu typu vlna se v jednu chvíli pohybuje 8 servomotorů, při vlnění se pohybuje 10 servomotorů a při pohybu tripod je 12 servomotorů v pohybu.

Pokud by servomotory odebíraly průměrně 500 mAh, pak by robot v pohybu vlna vydržel přibližně 39 minut, v pohybu vlnění 31 minut, a v pohybu tripod 25 minut.

Při testování, kdy robot střídal pohybové algoritmy a více než třetinu času stál, vydržel přibližně 45 minut. Při 11,2 V na baterii byl robot bezpečně vypnut, ale ještě by mohl pár minut pracovat.

Spotřeba celého robota byla následně měřena měřícím přístrojem. Kompletně zapojený robot spotřebovával 990–1050 mAh a to například, kdy servomotory nedržely svou pozici. Po zapnutí/aktivování servomotorů robot odebíral 1,4 Ah ve „vzduchu“ a 1,5 – 1,8 Ah po položení na zem, nebo když se robot zastavil. Při pohybu byla naměřena spotřeba 2,5 -3 Ah. Nejvíce úsporný byl pohyb vlna, pak vlnění a jen s malým rozdílem

byl nejméně úsporný pohyb tripod. Při plném zatížení byl naměřen maximální proud 9,10 Ah a robot by s touto spotřebou vydržel v provozu 16 minut.

Dále bylo provedeno testování založeno na výdrži robota v nečinnosti. Robot byl po celou dobu zapnutý, ale byl v nečinnosti (servomotory nedrželi pozici). Takto robot vydržel 140 minut a po naměření 3,4 V na jednotlivých člancích baterie byl robot vypnut. Z předchozího odstavce lze usoudit, že by robot v nečinnosti měl vydržet 148 minut. To se potvrdilo a měření lze považovat za odpovídající.

5.4 Test webového rozhraní

Cílem toho testu bylo ověřit, zda lze robota navádět bez vizuálního kontaktu. Test byl proveden ve tmě a byl použit přísvit webové kamery. Reakce robota na příkazy z webového rozhraní byla do 100 ms. Stream z kamery nebyl zcela plynulý, ale dostatečný pro orientaci.

Kamera poskytuje malé zorné pole pohledu a pro lepší orientaci v prostoru bylo potřeba robotem otáčet. Tím, že byla kamera umístěna v přední části robota, nebylo možné kontrolovat prostor okolo končetin. Na blížící se překážku končetin neupozornil ani žádný senzor. Naměřené vzdálenosti ze senzorů však poskytovaly velký přehled nad okolními objekty. Řízení robota na velkém prostoru bylo bez problémů, ale nasměrování robota mezi dva blízké objekty nebylo jednoduché.

6 Závěr

Cílem této práce byla realizace kráčivého podvozku se šesti končetinami tzv. Hexapodu. Využit inverzní kinematiku a dosáhnout věrohodného pohybu robota. Dále navrhnout matematický model ovládní robota a realizovat ovládací software primárně přes webové rozhraní s video streamem. Nakonec prototyp robota otestovat na různých terénech a pomocí senzorů zrealizovat základní úlohu vyhýbání se překážkám.

V rámci této práce jsou nejprve popsány podvozky, jejich výhody, nevýhody a dále se práce zaměřuje na podvozky kráčivé se šesti končetinami – Hexapody. Dále je v práci rozebrána přímá a inverzní kinematika, výpočty souřadnic nášlapné plochy, pohybové algoritmy inspirované přírodou, jejich výhody, nevýhody a použití v reálném prostředí

V neposlední řadě se práce zaměřuje na konstrukci robota. K tomu byla využita stavebnice Bioloid Premium od společnosti Robotis. Tato stavebnice včetně servomotorů Dynamixel AX-12+ byla k dispozici na pracovišti vedoucího práce. Konstrukce těla byla vyrobena na univerzitě v podobě dvou železných 1,5 mm silných plátů a dvou rozpěrek. Pro konstrukci každé končetiny byly také využity 2 železné pláty. Vše bylo vyřezáno laserem. Konec končetiny je vyroben z plastu na 3D tiskárně.

Při realizaci robota bylo dále využito: Raspberry Pi3 model B, Li-Po baterie o kapacitě 2600mAh, spínaný stabilizátor napětí, převodník USB2Dynamixel a vyrobená napájecí deska. V této práci se vycházelo z volně dostupného matematického modelu vytvořeného pro robota PhantomX Hexapod Mark III a také se vycházelo z C++ programu, který byl použit při realizaci ovládacího software. Tento program se zjednodušil a dále upravoval. Hlavním přínosem této práce je webové rozhraní, které umožňuje robota kompletně ovládat. Rozhraní bylo napsáno v PHP a na Raspberry byl nainstalován LAMP server. MySQL databáze byla použita jako prostředník pro komunikaci mezi webovým rozhraním a C++ programem. Webové rozhraní obsahuje video stream, o který se stará MJPG Streamer.

Byla testována chůze na několika typech povrchů. Všechny pohybové algoritmy fungují velmi dobře na rovném povrchu. Pro ukázkovou úlohu byly použity 4 ultrazvukové senzory připojené na Arduino MEGA. Arduino komunikuje s Raspberry přes UART. Ukázalo se, že tyto senzory jsou celkem pomalé, nedokáží měřit pod velkým úhlem a nedokáží měřit předměty s malou odrazovou plochou. Při reálném použití se na ně nelze zcela spolehnout. Vzdálenosti ze senzorů jsou zobrazovány ve webovém rozhraní a při detekované překážce program neumožňuje pokračovat daným směrem. V rámci práce byla také měřena spotřeba energie robota.

Seznam použité literatury

- [1] ZÁDA, Václav. Robotika: matematické aspekty analýzy a řízení. Vyd. 1. Liberec: Technická universita v Liberci, 2012. ISBN 978-80-7372-882-3.
- [2] NOVÁK, Petr. Mobilní roboty: pohony, senzory, řízení. 1. vyd. Praha: BEN – technická literatura, 2005. ISBN 80-7300-141-1.
- [3] ŽÁK, Marek. Návrh a konstrukce šestinožného mobilního robotu. Bakalářská práce. Vysoké učení technické v Brně. Fakulta informačních technologií. 2013
- [4] ŽÁK, Marek. Řízení pohybu robota typu hexapod. Diplomová práce. Vedoucí práce Jaroslav Rozman. Vysoké učení technické v Brně. Fakulta informačních technologií, 2015. Dostupné z: <http://hdl.handle.net/11012/52213>.
- [5] PELZ, Zdeněk. Řízení robota typu hexapod. Bakalářská práce. Vedoucí práce Ing. Marek Žák. Vysoké učení technické v Brně. Fakulta informačních technologií, 2016. Dostupné z: <http://hdl.handle.net/11012/62191>.
- [6] ZHONG, Guoliang, Long CHEN a Hua DENG. A Performance Oriented Novel Design of Hexapod Robots. IEEE/ASME Transactions on Mechatronics [online]. 1-1 [cit. 2017-03-20]. DOI: 10.1109/TMECH.2017.2681722. ISSN 1083-4435. Dostupné z: <http://ieeexplore.ieee.org/document/7876812/>
- [7] MĂNOIU, Olaru Sorin, Mircea NITULESCUR and Viorel STOIAN, "Hexapod robot. Mathematical support for modeling and control," 15th International Conference on System Theory, Control and Computing, Sinaia, 2011, pp. 1-6. Dostupné z: <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6085694&isnumber=6085648>
- [8] Robotis. Dynamixel AX-12 – User's manual. Dostupné z: URL: [http://www.trossenrobotics.com/images/productdownloads/AX-12\(English\).pdf](http://www.trossenrobotics.com/images/productdownloads/AX-12(English).pdf)

- [9] Raspberry Pi Learning Resources [online]. Cambridge (Massachusetts): Raspberry Pi Foundation, [cit. 2017-03-31]. Dostupné z: <https://www.raspberrypi.org/learning/hardware-guide/components/raspberry-pi/>
- [10] The MagPi: The official Raspberry Pi magazine [online]. Raspberry Pi Foundation [cit. 2017-03-31]. Dostupné z: <https://www.raspberrypi.org/magpi/>
- [11] Jak nabíjet lipol baterie. LipolBaterie.cz [online]. Praha [cit. 2017-04-14]. Dostupné z: <http://www.lipolbaterie.cz/navody/jak-nabijet-lipol-baterie/>
- [12] ASIF, Umar, Javaid IQBAL a M. Ajmal KHAN. *Kinematic Analysis of Periodic Continuous Gaits for a Bio-Mimetic Walking Robot*. IEEE International Symposium on Safety, Security, and Rescue Robotics. Kyoto. 2011. 80-85. DOI: 10.1109/SSRR.2011.6106784.
- [13] THILDERKVIST, Dan a Sebastian SVENSSON. *Motion Control of Hexapod Robot Using Model-Based Design*. Department of Automatic Control, 2015. Lund University. ISSN: 0280-5316. Dostupné z: <http://lup.lub.lu.se/student-papers/record/7365764>.
- [14] Dynamixel SDK: ROBOTIS e-Manual v1.30.00 [online]. Robotis, 2010 [cit. 2017-04-16]. Dostupné z: <http://support.robotis.com/en/software/dynamixelsdk.htm>
- [15] LADY ADA. Setting up a Raspberry Pi as a WiFi access point [online]. Adafruit Industries, 2016 [cit. 2017-04-19]. Dostupné z: <https://learn.adafruit.com/setting-up-a-raspberry-pi-as-a-wifi-access-point/overview>
- [16] Share your Raspberry Pi's files and folders across a network [online]. Raspberry Web Server, 2014 [cit. 2017-04-23]. Dostupné z: <http://raspberryyeserver.com/serveradmin/share-your-raspberry-pis-files-and-folders-across-a-network.html>

- [17] Systemd. *Archlinux* [online]. 2017 [cit. 2017-05-07]. Dostupné z: <https://wiki.archlinux.org/index.php/systemd>
- [18] HENDERSON, Gordon. Serial Library. *Gordons Projects* [online]. [cit. 2017-04-30]. Dostupné z: <https://projects.drogon.net/raspberry-pi/wiringpi/serial-library/>
- [19] SOVY, ŠINKANZEN A LEDŇÁČCI. Výzkum vývoj automotive [online]. Boomerang Publishing, 2015 [cit. 2017-05-08]. Dostupné z: <http://www.vvautomotive.cz/sovy-sinkanzen-a-lednacci/>
- [20] Boston Dynamics [online]. Massachusetts [cit. 2017-05-08]. Dostupné z: <http://www.bostondynamics.com>
- [21] KONEČNÁ, Kateřina. INSPIRACI HLEDÁME V PŘÍRODĚ, ŘÍKÁ UZNÁVANÝ VĚDEC JOSEF JANČÁŘ. *Lidé z VUT.cz* [online]. 2015 [cit. 2017-05-12]. Dostupné z: <https://zvut.cz/lide/lide-f38102/inspiraci-hledame-v-prirode-rika-uznavany-vedec-josef-jancar-d103365>
- [22] Stavební stroje a doprava [online]. Hrobce [cit. 2017-05-12]. Dostupné z: <http://www.ssad.cz/>
- [23] Jet Propulsion Laboratory: California Institute of Technology [online]. California [cit. 2017-05-09]. Dos *Jet Propulsion Laboratory: California Institute of Technology* [online]. California [cit. 2017-05-09]. Dostupné z: <https://www-robotics.jpl.nasa.gov/systems/> Dostupné z: <https://www-robotics.jpl.nasa.gov/systems/>
- [24] Stewartova platforma. *PressReleaseFinder* [online]. [cit. 2017-05-12]. Dostupné z: http://www.pressreleasefinder.com/primages/MGPR1006a_57471114.jpg
- [25] Femur. *Amateur Entomologists' Society* [online]. London [cit. 2017-05-12]. Dostupné z: <https://www.amentsoc.org/insects/glossary/terms/femur>

- [26] Foxy G2 - LC Li-Pol 2600mAh/11,1V 40/80V 28,9Wh. *PelikanDaniel.com* [online]. [cit. 2017-05-12]. Dostupné z: <http://www.pelikandaniel.com/?sec=product&id=73923>
- [27] Foxy UBEC 3A. *PelikanDaniel.com* [online]. [cit. 2017-05-12]. Dostupné z: <http://www.pelikandaniel.com/?sec=product&id=53494>
- [28] *Lynxmotion* [online]. RobotShop [cit. 2017-05-12]. Dostupné z: <http://www.lynxmotion.com/>
- [29] *Trossen Robotics* [online]. [cit. 2017-05-12]. Dostupné z: <http://www.trossenrobotics.com/>
- [30] *Raspberry Pi* [online]. The Raspberry Pi Foundation [cit. 2017-05-13]. Dostupné z: <https://www.raspberrypi.org/>
- [31] Spotlight webcam pro. *Trust* [online]. [cit. 2017-05-13]. Dostupné z: <http://www.trust.com/en/product/16428-spotlight-webcam-pro>
- [32] HIRNŠAL, Jakub. HexorV3. *YouTube* [online]. [cit. 2017-05-13]. Dostupné z: <https://www.youtube.com/playlist?list=PL1wbrPDP0K2RwE5JZhZFYXk6K2uz1Xfb4>

A Obsah příloženého CD

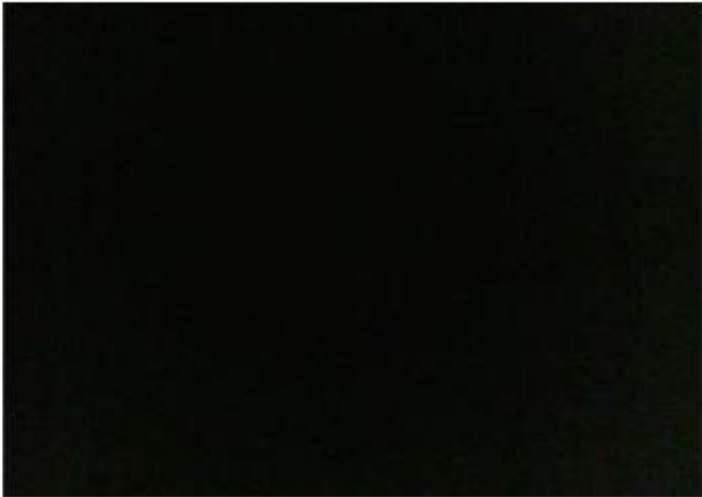
- 3D modely – obsahuje model celého robota a modely částí robota
- Fotografie – fotografie pořízené při testování v terénu
- Text diplomové práce
 - DP_Hirnšal_Jakub.pdf
- Zdrojové kódy
 - Arduino – zdrojové kódy v C++
 - HexorV3 – zdrojové kódy v C++
 - Test rychlosti DB
 - Test UART
 - Webová aplikace
 - create_database.sql

B Webové rozhraní

TECHNICKÁ UNIVERZITA V LIBERCI
Fakulta mechatroniky, informatiky
a mezioborových studií

HEXOR V3


Webové rozhraní pro ovládání robota



Galil Walk mode Napájení BalanceMode Výška Rychlost a výška Zdvih Ovládací mód Další krok

Baterie	teplota	Napájení	Rychlost	Ovládací mód	BalanceMode
11.8V	0	0	0	0	0

Výška	Zdvih	Delší krok	Pohybový alg	walkMethod
35	50	1	Tripod 6 steps	1



Vlevo	Vpravo	Vpředu	Vzadu
35	61	76	0

Vytvořil Bc. Jakub Hrnšál
Vedoucí práce: Ing. Miroslav Holeda, Ph.D.

Diplomová práce

Školní rok 2016/2017
Technická univerzita v Liberci

C Váha konstrukce robota

Při konstrukci robota byly váženy následující díly robota:

- Železný plát pro část Tibia
 - 27 g
- Plastové chodidlo
 - 8 g
- Tibia – část končetiny
 - 215 g
- Horní díl těla
 - 243 g

Celková váha robota je 2 805 g

D Výsledná konstrukce robota

