



# VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

## FAKULTA STROJNÍHO INŽENÝRSTVÍ

FACULTY OF MECHANICAL ENGINEERING

## ÚSTAV AUTOMATIZACE A INFORMATIKY

INSTITUTE OF AUTOMATION AND COMPUTER SCIENCE

## NÁVRH KONSTRUKCE ROBOTICKÉHO HADA

DESIGN OF A ROBOTIC SNAKE CONSTRUCTION

### DIPLOMOVÁ PRÁCE

MASTER'S THESIS

### AUTOR PRÁCE

AUTHOR

Bc. Jakub Pollák

### VEDOUCÍ PRÁCE

SUPERVISOR

Ing. Tomáš Hůlka

BRNO 2023



# Zadání diplomové práce

Ústav:	Ústav automatizace a informatiky
Student:	<b>Bc. Jakub Pollák</b>
Studijní program:	Aplikovaná informatika a řízení
Studijní obor:	bez specializace
Vedoucí práce:	<b>Ing. Tomáš Hůlka</b>
Akademický rok:	2023/24

Ředitel ústavu Vám v souladu se zákonem č.111/1998 o vysokých školách a se Studijním a zkušebním řádem VUT v Brně určuje následující téma diplomové práce:

## Návrh konstrukce robotického hada

### Stručná charakteristika problematiky úkolu:

Úkolem studenta bude navrhnout konstrukci modelu robotického hada s využitím konceptu ROBOTIS, tu zrealizovat, otestovat a vyhodnotit výsledky.

### Cíle diplomové práce:

Stručná rešerše problematiky robotických hadů.

Návrh konstrukce robotického hada primárně pro pohyb ve 2D s ambicí pro pohyb ve 3D.

Praktická realizace navržené konstrukce.

Otestování funkčnosti konstrukce v reálném prostředí a následné srovnání rychlosti pohybu v závislosti na parametrizaci modelu, prostředí, řízení apod.

### Seznam doporučené literatury:

WORST, Rainer, and Ralf LINNEMANN. "Construction and operation of a snake-like robot." Proceedings IEEE International Joint Symposia on Intelligence and Systems. IEEE, 1996.

HOPKINS, James K., BRENT W. Spranklin, and SATYANDRA K. Gupta. "A survey of snake-inspired robot designs." Bioinspiration & biomimetics 4.2 (2009): 021001.



Termín odevzdání diplomové práce je stanoven časovým plánem akademického roku 2023/24

V Brně, dne

L. S.

---

prof. Ing. Radomil Matoušek, Ph.D.  
ředitel ústavu

---

doc. Ing. Jiří Hlinka, Ph.D.  
děkan fakulty



## ABSTRAKT

Táto diplomová práca nadväzuje na autorovu bakalársku prácu „Návrh modelu robotického hada“. Jej cieľom bola simulácia robota vytvoreného z dielcov sady Bioloid a ukážka možných pohybov v simuláciách. Cieľom tejto práce je fyzická konštrukcia daného robota a experimentálne overiť pohyblivosť testovaním na rôznych povrchoch. Prvá časť práce je zameraná na zhrnutie poznatkov biologických hadov a využitie robotických hadov v robotike. V druhej časti je opísaný spôsob riadenia a konštrukcia robotického hada. Posledná časť je zameraná na testovanie limitov a prekonávanie prekážok, z ktorých je vyvodенý záver.

## ABSTRACT

This master's thesis is a continuation of the author's bachelor's thesis "Design of a model of a robotic snake", a goal of which was the simulation of a robot created from parts of the Bioloid set and a demonstration of possible movements in the simulations. The aim of this thesis is the physical construction of the given robot and the experimental verification of its mobility by testing it on different surfaces. The first part of the thesis is focused on a summary of information about biological snakes and the use of robotic snakes in robotics. The second part describes the control method and the construction of the robotic snake. The last part deals with testing limits and overcoming obstacles. A conclusion is drawn from the results of the last part of the thesis.

## KLÚČOVÉ SLOVÁ

Robotický had, Arduino, Dynamixel, serpenoida

## KEYWORDS

Robotic snake, Arduino, Dynamixel, serpenoid curve







2023

## **BIBLIOGRAFICKÁ CITACE**

POLLÁK, Jakub. Návrh konstrukce robotického hada. Brno, 2024. Dostupné také z: <https://www.vut.cz/studenti/zav-prace/detail/154208>. Diplomová práce. Vysoké učení technické v Brně, Fakulta strojního inženýrství, Ústav automatizace a informatiky. Vedoucí práce Tomáš Hůlka.



## **POĎAKOVANIE**

Chcem poďakovať vedúcemu práce a vedúcemu ústavu za pomoc, ochotu a trpezlivosť, študentskému tímu strojLAB za nadobudnuté skúsenosti.



## ČESTNÉ PROHLÁŠENÍ

Prohlašuji, že, že tato práce je mým původním dílem, vypracoval jsem ji samostatně pod vedením vedoucího práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury.

Jako autor uvedené práce dále prohlašuji, že v souvislosti s vytvořením této práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a jsem si plně vědom následku porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestně právních důsledků.

V Brně dne 30. 9. 2023

.....  
Jakub Pollák



# OBSAH

<b>1</b>	<b>ÚVOD.....</b>	<b>17</b>
<b>2</b>	<b>PREHĽAD SÚČASNÉHO STAVU POZNANIA.....</b>	<b>19</b>
2.1	Pohyb biologických hadov .....	22
2.1.1	Laterálna undulácia (Lateral undulation) .....	22
2.1.2	Bočné vlnenie (Side winding) .....	23
2.1.3	Harmonikový pohyb (Concertina progression) .....	23
2.1.4	Priamočiary pohyb (Rectilinear/caterpillar movement) .....	23
2.2	Robotické hady .....	24
2.2.1	Vplyv parametrov na tvar krivky pohybu.....	24
2.2.2	Prehľad robotických hadov.....	28
2.2.3	Lietajúci had DRAGON .....	31
2.2.4	Obožžiteľný robot .....	32
2.2.5	Had s poháňanými plutvami .....	32
2.2.6	Had s poháňanými kolesami .....	33
<b>3</b>	<b>VLASTNÉ RIEŠENIE .....</b>	<b>34</b>
3.1	Použiteľný hardware a software .....	34
3.2	Stavebnica Bioloid.....	35
3.2.1	Aktuátory .....	36
3.3	Riadenie .....	37
3.4	Komunikácia.....	39
3.4.1	Štruktúra paketu.....	39
3.4.2	Komunikácia a riadenie .....	40
3.5	Konštrukcia.....	42
3.6	Napájanie .....	45
3.7	3D tlačené dielce .....	46
3.8	Matematický popis .....	50
3.9	Popis kódu .....	52
3.9.1	Nastavenie pre aktuátory Dynamixel .....	55
3.9.2	Funkcie na začiatku .....	55
3.9.3	Necyklické volanie kódu .....	56
3.9.4	Cyklické volanie kódu .....	58
3.9.5	Generátor hodnôt vlnenia .....	60
3.9.6	Hlavné pohyby:.....	62
3.9.7	Pomocné pohyby: .....	62
3.9.8	Funkcie na konci.....	64
<b>4</b>	<b>ZHODNOTENIE A DISKUSIA.....</b>	<b>66</b>
4.1.1	Testovanie.....	66
4.1.2	Výsledná konštrukcia .....	67
4.1.3	Výsledky experimentov .....	70
4.1.4	Ďalší vývoj.....	73
<b>5</b>	<b>ZÁVER .....</b>	<b>74</b>
<b>6</b>	<b>ZOZNAM POUŽITEJ LITERATÚRY.....</b>	<b>77</b>
<b>7</b>	<b>ZOZNAM OBRÁZKOV A TABULIEK.....</b>	<b>79</b>

<b>8</b>	<b>ZOZNAM PRÍLOH .....</b>	<b>81</b>
----------	----------------------------	-----------



# 1 ÚVOD

„Příroda, nekonečná inspirace vědy“ je názov knihy [1], ktorá výstižne opisuje túto prácu. Ľudstvo sa snaží napodobniť mechanizmy vyvinuté prírodou po milióny rokov evolúcie. Konvenčné pozemné roboty používajúce kolieska majú problém pri použití mimo upravených povrchov, ciest a chodníkov. Ďalším problémom sú úzke miesta a inak neprístupné miesta. Kontrola potrubí, ako zvonka tak aj zvnútra, šplhanie po stĺpoch osvetlenia za účelom kontroly, plazenie sa úzkymi rúrami odpadových vôd, ale aj podkopanie a pretlačenie sa pod prekážkou sú činnosti, v ktorých robotické hady excelujú.

Pohyb zvierat je často napodobňovaný v robotike za účelom zlepšenia priechodnosti neprístupného terénu. Medzi najčastejšie pohyby hadov patrí pohyb harmonikový, bočný, priamočiary a pohyb bočným vlnením. Väčšina pohybov má spoločný opis krivky, ktorú vytvárajú. Jej aproximácia sa nazýva serpenoida [2] a bola popísaná profesorom Shigeo Hirose, ktorý ako prvý vyvinul prvý model robotického hada inšpirovaného prírodou. V praxi sa používajú rôzne kombinácie koliesok, pásov, hnaných aj poháňaných.

Riadenie robota môže byť manuálne, čiastočne alebo plne autonómne. Navigácia neznámym a náročným terénom je zložitá a nepredpokladateľná, takže roboti sú väčšinou riadení operátorom s diaľkovým ovládaním, ktorý dozerá na pohyb hada a robí korekcie. Elektronika hada sa stará o vytváranie pohybov a operátor upravuje parametre pohybu podľa potrieb.

V prvej časti tejto práce je oboznámenie čitateľa s pohybmi biologických hadov, existujúcimi robotickými hadmi ako aj ich použitím. Súčasťou tejto práce je zhrnutie predchádzajúcich skúseností zo simuláciou robotického hada v prostredí CoppeliaSim, ktorá bola autorovou záverečnou prácou [3], využitie poznatkov nadobudnutých počas štúdia v danej problematike a skonštruovanie robotického hada.

Had je vyrobený zo stavebnice Bioid, obsahuje konštrukciu hada, ktorý sa pohybuje planárne, resp. všetky osi rotácie sú v jednej rovine. Toto znižuje priechodnosť robota členitým terénom a obmedzuje použitie na jeden typ pohybu. Otočenie každej druhej osi o  $90^\circ$  okolo pozdĺžnej osi umožňuje pohyb v 3D priestore, čo má za následok zlepšenie priechodnosti terénom. Robota je možné ovládať manuálne v reálnom čase priamo, s pomocou pred-definovaných pohybov. Po nastavení robota je možné odpojiť riadiaci kábel a had použije posledné nastavenia a senzory pre autonómne riadenie.

Testovanie po naklonenej rovine a prechádzaní a vyhýbaní sa prekážok prebehlo v dielni ÚAI FSI VUT.

Nadobudnuté poznatky z tejto práce môžu byť použité pre ďalšie vylepšenia modelu. Niektoré návrhy pre zlepšenia sú zhrnuté v závere práce.

V prílohe sú odkazy pre nástroje, ktoré boli použité pri práci a vizualizácií, STL súbory 3D tlačných článkov a krytov elektroniky.



## 2 PREHĽAD SÚČASNÉHO STAVU POZNANIA

Na prvý pohľad sa môže zdať, že hady sú v nevýhode kvôli chýbajúcim končatinám, ktoré by im pomáhali pri pohybe. Napriek tomuto predsudku sú hady schopné pohybovať sa rýchlo, efektívne po súši aj vo vode. Niektoré druhy sú dokonca schopné plachtiť vzduchom. Aj napriek podobnému vzhľadu tela sa hady líšia hlavne dĺžkou, hrúbkou a prierezom tela. Hady pohybujúce sa po súši majú spodok tela sploštený, pripomínajúci tvar písmena D, obráteného plochou stranou nadol. Pomer výšky a šírky tela sa odvíja od spôsobu lovu a polohy lovenej zveri. Väčšie, pomalé a silnejšie hady, ktoré obeť udusia, škrtiče, majú telo najširšie. Hady šplhajúce po korunách stromov sú obratnejšie. Svoju obratnosť potrebujú pre prispôbeniu tela vetvám stromov a pohybe v stiesnených priestoroch. (Obr. 1)



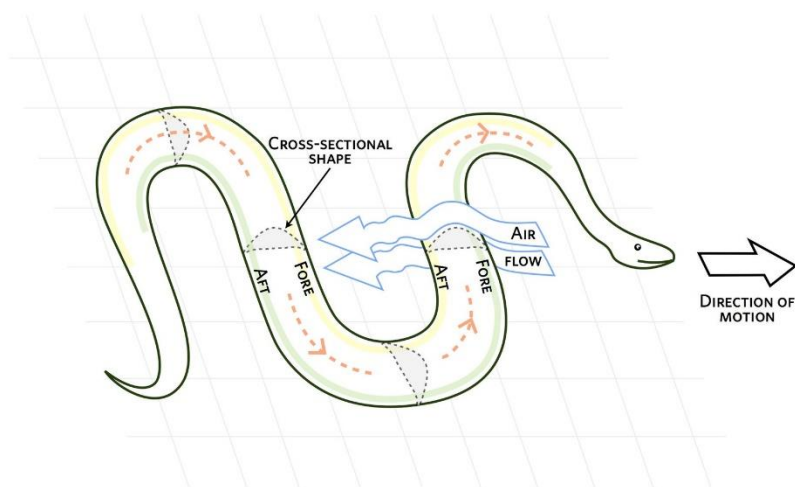
Obr. 1: Prierez tela (zľava: škrtič, nešpecializovaný tvar, šplhajúci had)

Koža hadov je tvorená šupinami, ktorých tvar závisí od prostredia, v ktorom žijú. Hady žijúce v drsných podmienkach, napríklad na púšti, majú hrubé a ostré kýlovité šupiny pre lepšie plazenie v sypkých materiáloch ako piesok. Hady žijúce v hustom poraste majú hladké šupiny s nízkym koeficientom trenia, pre ľahší pohyb. Existujú aj špecializované prvky na povrchu tela ako rohy na hlave, štít (česky kápě) na krku (typické pre kobry), alebo štrkotajúce šupiny, typické pre štrkáče (česky chřestíše). Na obrázku nižšie štrkáč, ktorý má zároveň rohy. Anglicky zvaný „sidewinder“, podľa typického pohybu zvaného „side-winding“. [18]



Obr. 2: Štrkáč rohatý (*Crotalus cerastes*) [4]

„Lietajúce hady“ sú schopné plachtiť vzduchom vďaka rozšírenému a ploškému telu. Spolu s pohybom a skrúcaním tela vytvárajú vztlak, ktorý spomaľuje ich pád. Prierez tela a poloha pri plachtení je na Obr.3.



Obr. 3: Lietajúci had [5]

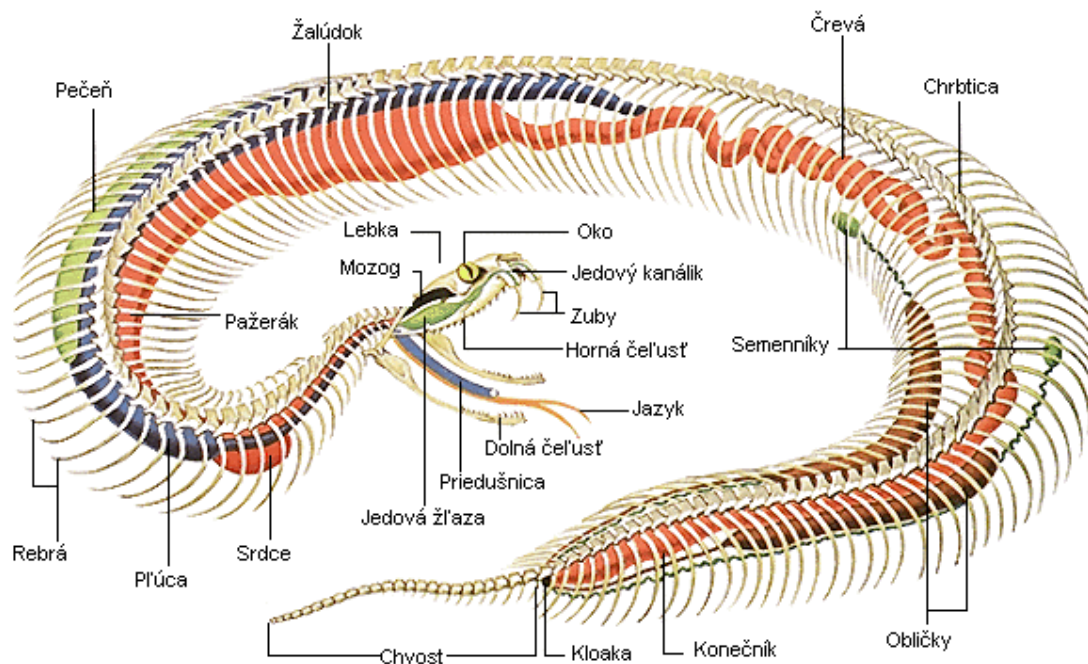
Hady žijúce pri vodných plochách používajú jednoduchý pre pohyb vo vode pohyb zvaný laterálna undulácia. Pohybom vo vode sú najbližšie úhorom. Na rozdiel od nich, nepotrebnú plutvy ani končatiny. [28]



## 2.1 Pohyb biologických hadov

[20, 21, 22, 23]

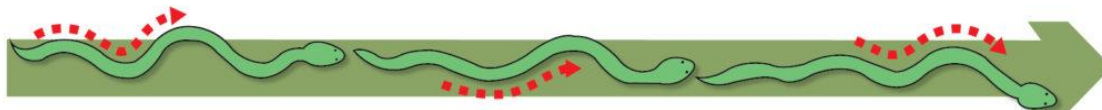
Z anatómie hada nás najviac zaujíma kostra, ktorá definuje tvar tela a povoľuje možné stupne voľnosti. Na obrázku nižšie (Obr.4) môžeme vidieť, že had interaguje s okolím stykom s rebrami kostry. Rebrá sú spojené množstvom zložitých svalov a pokryté šupinatou kožou. Nasledujúce podkapitoly priblížia najtypickejšie pohyby.



Obr. 4: Anatómia hada [6]

### 2.1.1 Laterálna undulácia (Lateral undulation)

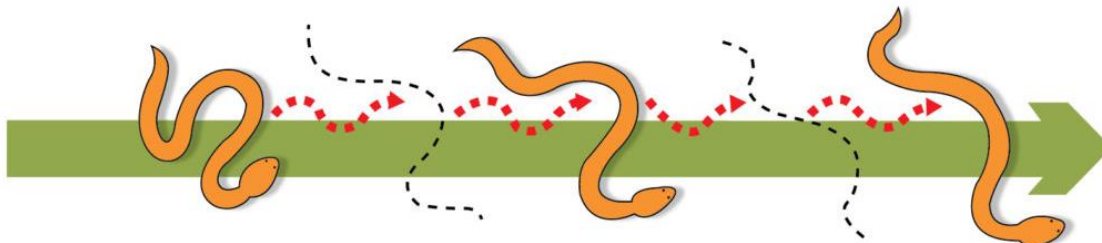
Vychýlením tela do strán opisuje had krivku pripomínajúcu písmeno „S“. Každý článok sa pohybuje po rovnakej trase ako hlava, had zanecháva len tenkú stopu v tvare sínusovej funkcie. Pohyb je vykonávaný v jednej rovine, čím je konštrukcia robota napodobňujúceho tento typ pohybu je jednoduchá. Pohyb je možné použiť pre prekonávanie sypkého terénu, a pri plávaní na hladine kvapalín.



Obr. 5: Laterálna undulácia [7]

### 2.1.2 Bočné vlnenie (Side winding)

Pohyb je tvorený použitím predošlého typu pohybu, ale aplikovaním vychýlenia článkov v dvoch na sebe kolmých rovinách. Výsledný pohyb smeruje diagonálne, presný uhol závisí na prostredí. Hady žijúce v horúcom prostredí používajú tento typ pohybu, pretože stičná plocha medzi hadom a povrchom je minimálna, vďaka čomu limitujú prenos tepla.



Obr. 6: Bočné vlnenie [7]

### 2.1.3 Harmonikový pohyb (Concertina progression)

Zložitejší typ pohybu, pri ktorom had udržuje skrútený tvar a natáhuje prednú a zadnú časť tela. Predná časť tela sa natiahne vpred, zatiaľ čo je zadná časť tela zapretá o povrch. Následne zaprie prednú časť a pritiahne zadnú ku sebe. Pohyb pripomína natáhovanie a skracovanie harmoniky. Tento pohyb používajú napríklad dážďovky.



Obr. 7: Harmonikový pohyb [7]

### 2.1.4 Priamočiary pohyb (Rectilinear/caterpillar movement)

Pri priamočiarom pohybe had používa medzi-rebrové svaly pre krátke pohyby kože na bruchu. Vlna takto vytvorená postupuje odzadu vpred a pripomína pohyb nôh mnohonožiek. Tento typ pohybu bol podrobne vysvetlený len pred pár rokmi. Pohyb je veľmi pomalý a preto ho hady používajú len pri love vo vysokom poraste alebo v prípade keď sa necítia ohrozené a nepotrebujú sa rýchlo presunúť. Napodobniť tento typ pohybu v robotike je zložité, pretože by bolo nutné vytvoriť dva protíahlé svaly pre každé rebro, ktoré je spojené zo šupinatou kožou na bruchu. Svaly nadvihnú kožu, presunú ju vpred, položia na zem a potiahnu dozadu, vytvárajúc efekt kráčania. Pre hardvérovú zložitosť, množstvo malých pohonov a nutnosť použitia materiálu pripomínajúceho vlastnosti elastickej šupinatej kože nie je v robotike rozšírený.



Obr. 8: Priamočiary pohyb [7]

## 2.2 Robotické hady

Pohyby hada skúmal profesor Shigeo Hirose už v roku 1972, a ako prvý vytvoril aj matematický popis vlnenia biologických hadov[8]. Rovnice sa nazývajú serpenoida a popisujú niekoľko pohybov hadov len zmenou parametrov. Matematický popis:

$$\alpha(n, t) = \begin{cases} A_x * \sin(\omega_x t + n * \delta_x), & n = 2k + 1 \\ A_x * \sin(\omega_x t + n * \delta_x + \phi), & n = 2k, k \in \mathbb{N} \end{cases}$$

Parametre  $A_x$ ,  $\omega_x$ ,  $\delta_x$ , a  $\Phi$  definujú výsledný tvar vlny. Parameter „n“ je poradové číslo článku, vzostupne od hlavy robota.

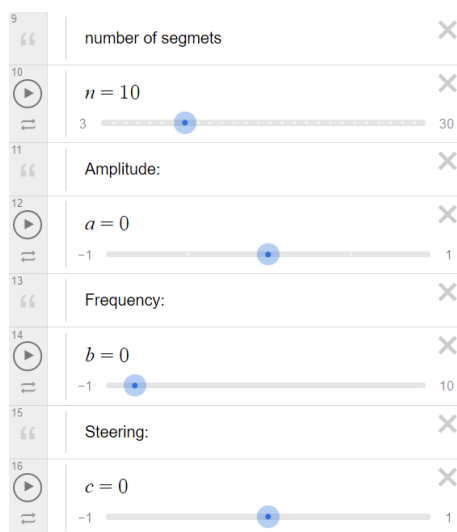
### 2.2.1 Vplyv parametrov na tvar krivky pohybu

Rovnica serpenoidy bola vložená do online nástroja Desmos pre vykreslenie krivky. Po úprave sme dostali rovnicu s parametrami, ktoré môžeme meniť v reálnom čase a sledovať ich vplyv na vytvorenú krivku. Nástroj pre vykresľovanie je voľne dostupný, a odkaz na projekt je v prílohe. Rovnice upravené pre program Desmos:

$$X = \int_0^s \cos(a * \cos(b * t + a_{nim}) + c * t) dt$$

$$Y = \int_0^s \sin(a * \cos(b * t + a_{nim}) + c * t) dt$$

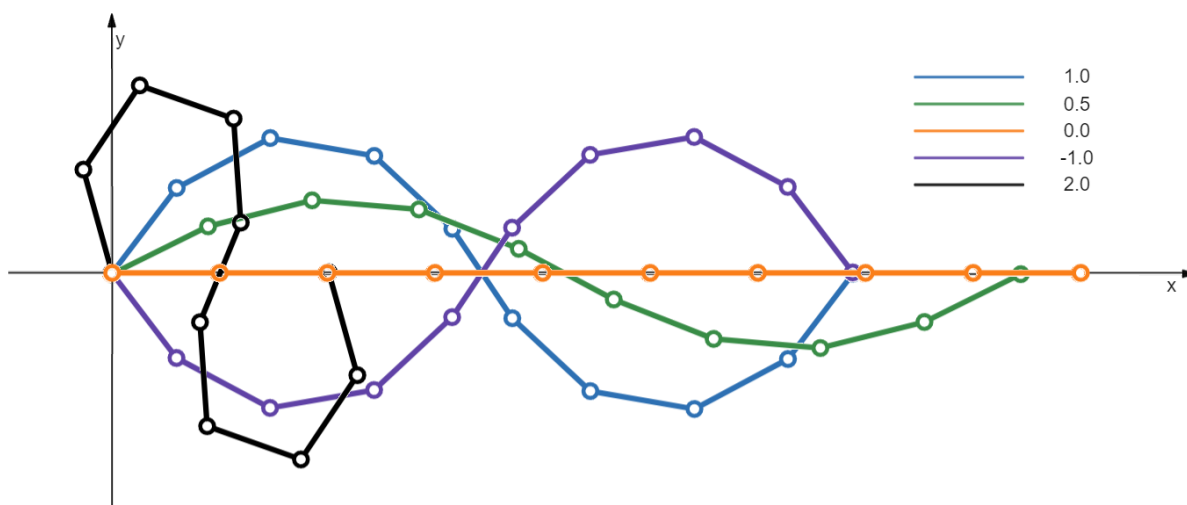
Parameter „s“ je tvorený zoznamom (array), obsahujúcim počet článkov „n“. Parameter „a“ nastavuje amplitúdu vychýlenia článkov. Parameter „b“ nastavuje frekvenciu vlnenia. Parameter „c“, nazývaný aj konštantná zložka, mení hodnotu okolo ktorej periodická funkcia osciluje, čím spôsobuje zatáčanie robota. Parameter „anim“ bol pridaný pre animáciu, vďaka čomu je možné vidieť priebeh vlnenia v reálnom čase, ako aj zistiť možné kolízie článkov. Parametre je možné priamo prepísať, alebo použiť posúvače (Obr.9).



Obr. 9: Posúvače vstupných parametrov v programe Desmos

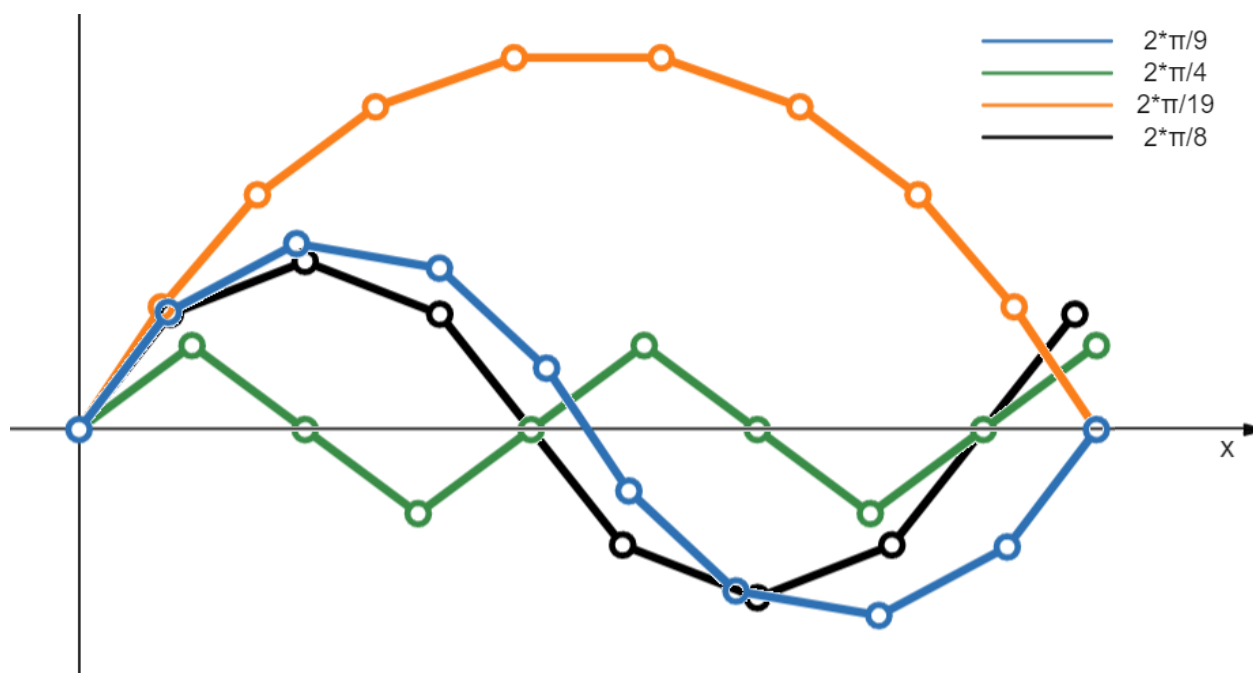


Amplitúdu vlnenia je možné nastaviť parametrom „a“. Ovplyvňuje veľkosť uhla, ktorým sú články vychýlené z nulovej polohy. Hodnota 0 spôsobí, že články ostanú vystreté, a nebudú sa vlniť. Je to efektívny spôsob ako vypnúť vlnenie, čo je využité pri pohybe vyžadujúcich pohyb len v rovine. Vyššie čísla amplitúdy spôsobia väčšie výkyvy tela, čo zvyšuje stabilitu pohybu na naklonenej rovine, a zvyšuje ťah vytvorený robotom. Ďalšou výhodou je zapieranie sa o členitý terén, a odrážanie sa od nerovností. Nevýhodou je pomalá dopredná rýchlosť, napriek veľkej relatívnej rýchlosti článkov a aktuátorov. Z toho plynie, že pohyb s veľkými vychýleniami je energetický náročnejší pre prejdennú vzdialenosť, ale je stabilnejší a schopný ťahať záťaž (napr. napájací kábel). Negatívne hodnoty sú povolené, výchylka je síce záporná, ale na funkcií sa to neprejaví, pretože smer postupu vlny je rovnaký, výsledný tvar je len fázovo posunutý o  $180^\circ$ . Podrobnejšie experimenty a vplyv na ťah robotického hada je zhrnutý v bakalárskej práci autora. [3]



Obr. 10: Vplyv parametra „a“ na tvar krivky

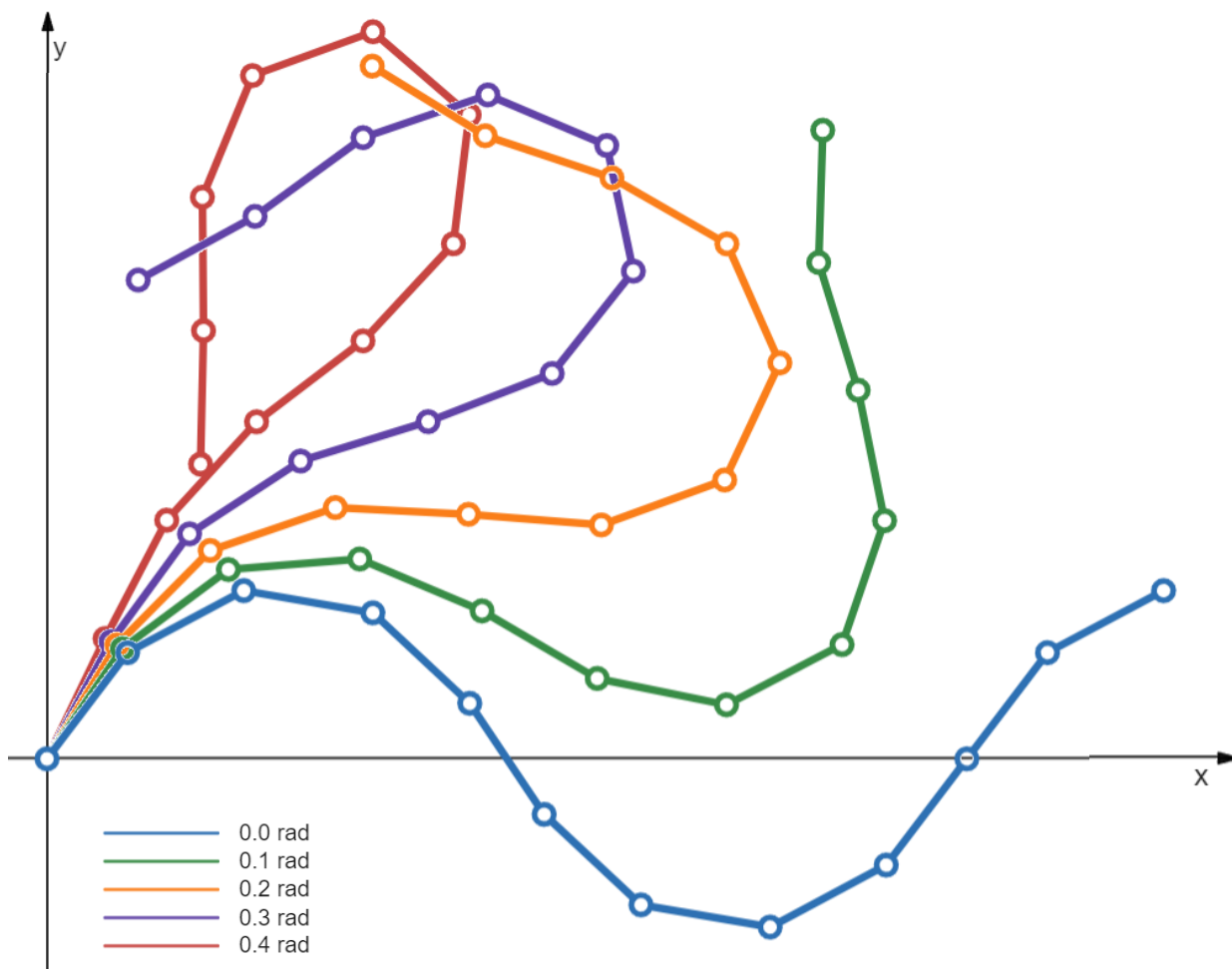
Napríklad pre parametre:  $n = 10$ ,  $a = 1$   $s = 0$ , meníme parameter „ $f$ “, potom výsledok môžeme vidieť na Obr.(11). Frekvencia ovplyvňuje počet vln, ktoré had vykoná na dĺžku. Ak je počet bodov rovnaký ako číslo, ktorým delíme periódu, dostaneme jednu vlnu na celkovú dĺžku hada. Na obrázku Obr.(11) je tento jav zvýraznený modrou farbou.



Obr. 11: Vplyv parametra „ $f$ “ na tvar krivky

Keďže rovnica vlnenia je spojitá, ale články hada sú početne konečné, dochádza tu ku problému spojeným s Shannonov-Nyquistov-Kotelnikovým teorémom. Ak je frekvencia príliš vysoká, robot nie je schopný dosiahnuť správny tvar, pretože dĺžka článku je väčšia ako jeden výkyv periodickej funkcie (polovica periódy). Preto je najvyššia možná frekvencia  $2 \cdot \pi / 4$ .

V ďalšej ukážke boli parametre nastavené na:  $n = 12$ ,  $a = 1$   $f = 2 \cdot \pi / 9$ , parameter „s“ bol menený, vplyv zmien na obrázku obr.(12). Parameter „s“ je konštantná zložka, ktorá sa pripočíta ku vychýleniu od sínusovej/kosínusovej funkcie. To spôsobí, že robotický had vychýľuje články na jednu stranu viac ako na druhú. Keď sa táto zložka aplikuje pre všetky články, had bude zatáčať. Záporné hodnoty zatáčajú robotického hada do opačnej strany. Parameter rovný nule vytvára symetrické výchylky a had sa pohybuje rovno.

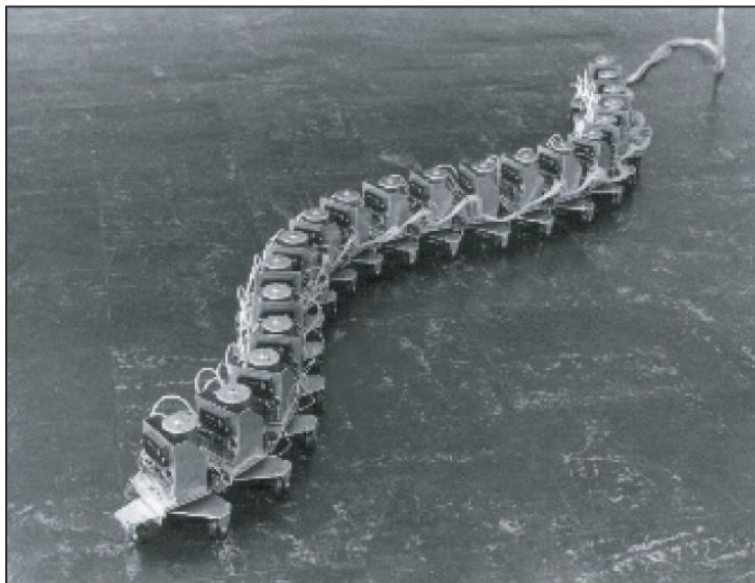


Obr. 12: Vplyv parametra „s“ na tvar krivky

Ak je hodnota parametra v absolútnej hodnote príliš veľká, môže dôjsť ku kolíziám článkov. Väčšinou hlava narazí do tela, čo môže spôsobiť problémy pri pohybe, navigácii, ale aj poškodení senzorov alebo poškodení aktuátorov z dôvodu preťaženia. Konkrétne hodnoty je nutné overiť experimentálne alebo simulačne, pretože nástroj vyrobený v aplikácii Desmos, nepočíta zo šírkou článkov, len s prekrytím článkov. Môžeme s určitou istotou povedať, že pre konfiguráciu nastavenú v Desmose, je hodnota 0.4 radiánu na hranici prípustná. Vyšší počet článkov alebo vyššia hodnota by už spôsobila kolíziu. Tento problém je možné potlačiť pasívne počtom článkov, obmedzením maximálnej výchylky článkov, alebo obmedzením povolených hodnôt parametra „s“, tak aby nedošlo ku kolíziám. Aktívnym riešením by bolo vypočítanie žiadanej polohy a kontrole kolízií. Tento postup by bol výpočtovo náročný.

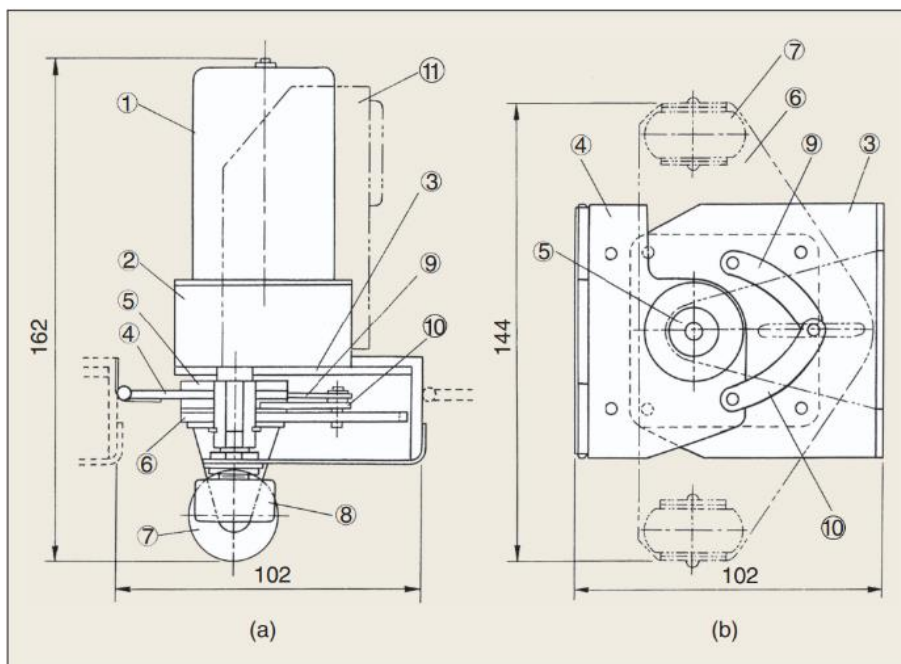
## 2.2.2 Prehľad robotických hadov

Vývoj robotických hadov začal robotmi vlnacimi sa v rovine rovnobežnej na plochu, po ktorej sa pohybovali a pre pohyb vpred využívali vlnenie popísané serpenoidou. Príkladom takéhoto robota je ACM-III. (Obr.13)



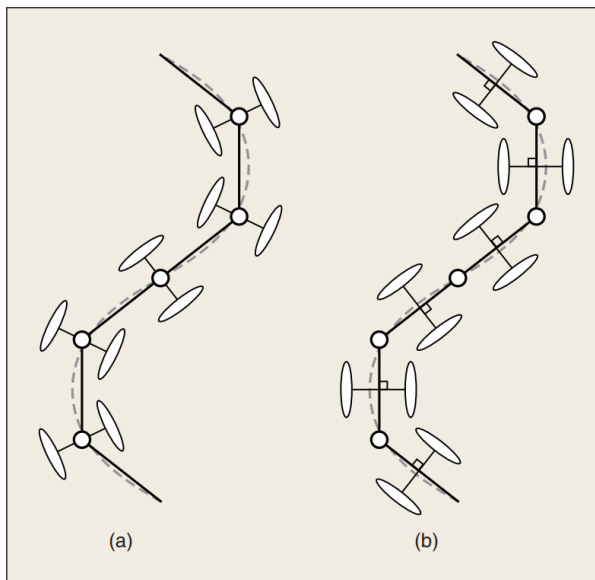
Obr. 13: Robot ACM-III [8]

Články sa otáčali vpravo/vľavo okolo vertikálnej motorizovanej osi. Hnané kolieska sa natáčali podľa uhlu dvoch spojených článkov. Na štruktúre článku robota ACM-III (Obr.14) je napravo spodný pohľad na kĺb spájajúci články. Napravo od osi je možné vidieť dve zahnuté ramená, spojené jedným koncom spolu a druhým ku článkom. Spoločný spoj týchto ramien sa pohyboval v drážke podvozku. Týmto mechanizmom zaručil natáčanie koliesok tak, aby zvierali polovičný uhol voči uhlu článkov, medzi ktorými sa nachádza.



Obr. 14: Štruktúra článku robota ACM-III [8]

Model ACM-R3 vyvíjaný v roku 2001 je bezdrôtovo ovládaný had, schopný pohybu v 3D priestore. Má 20 článkov, dĺžku 1.8m a váži 12 kg. Batéria je schopná prevádzky na viac ako 2h. Had sa skladá z dvoch typov článkov, ktoré majú osi rotácie pootočené o  $90^\circ$  v pozdĺžnej ose hada, efektívne tvoriace kardanový kĺb. Poloha a upevnenie koliesok sa zmenili. Namiesto malých koliesok natáčajúcich sa voči článkom cez mechanizmy, boli veľké kolesa pevne uchytené ku článkom.



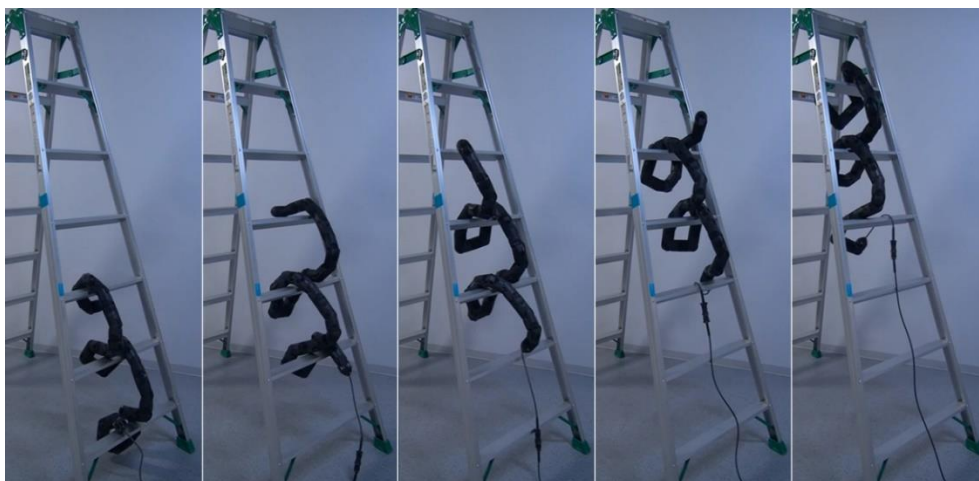
Obr. 15: Rozdiel polohy a uloženia kolies medzi robotmi ACM-III (a) a ACM-R3 (b) [8]

Toto riešenie je konštrukčne jednoduchšie, pevnejšie a kolesá, ktoré sú väčšie ako články pomáhajú pri prechode terénom. Vďaka týmto razantným zmenám vyzerá následný model veľmi odlišne od prototypu z roku 1972.



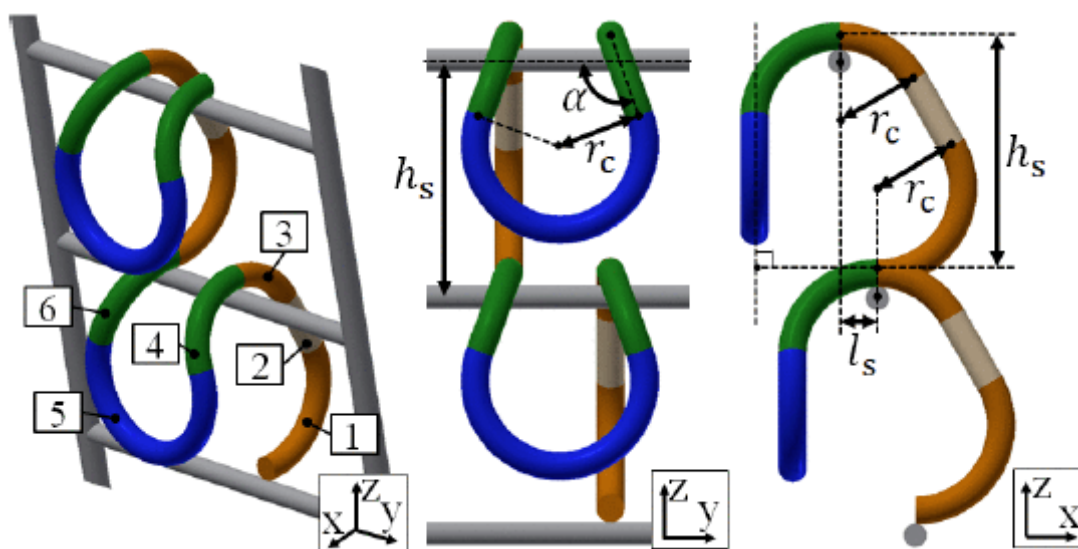
Obr. 16: Robot ACM-R3 [8, 25]

Robotické hady sa snažia napodobniť pohyb biologických hadov, prípadne ich kombinujú s inou formou pohonu, pre ešte širšie možnosti ako prekonať prekážky. Schopnosť zdolávania náročných prekážok ako aj ukážku obratnosti, pri šplhaní po rebríku, dokázal robot predstavený na výstave „International Conference on Intelligent Robots and Systems“ v Španielsku, ktorého autorom je Tatsuya Takemori.



Obr. 17: Ukážka lezenia po rebríku [9]

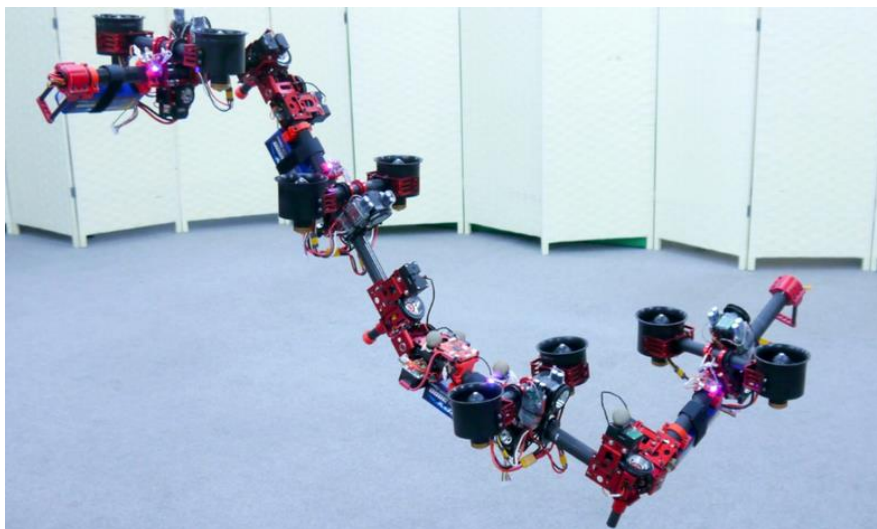
Had používa zložité metódy spájania tvarov, ktoré sa potom snaží napodobniť. Tieto tvary majú parametre pre prispôbenie rôznym vzdialenostiam priečok a sklonu rebríka.



Obr. 18: Vizualizácia vstupných dát [9]

### 2.2.3 Lietajúci had DRAGON

Robot vzhľadom pripomína hada, ale je kategorizovaný ako lietajúci dron s meniteľnou konfiguráciou. Na každom článku je pár motorov s vrtuľami, ktoré vytvárajú ťah.



Obr. 19: DRAGON [10, 24]

Ovládanie je výpočtovo náročné pretože vplyv motorov sa mení s konfiguráciou. Robot je schopný prejsť cez malé otvory a poskladať sa do tesných miest, následne rozložiť do situácie vhodnej konfigurácie. Konfigurácia v tvare písmena „L“ poskytuje robotovi dlhé rameno, vďaka čomu dokáže vyvinúť krútiaci moment 2 Nm potrebné pre uzavretie ventilu. Robot je zložitý a relatívne ťažký, čo spôsobuje že batéria vydrží len na tri minúty letu. Hlavnou úlohou robota je uzavretie kritických ventilov v prípade havárie elektrárne alebo chemických tovární.



Obr. 20: Konfigurácia pri uzatváraní ventilu [10]

## 2.2.4 Obojživelný robot

Ďalším vývojom profesora Hirose vznikol robot schopný podvodného prieskumu, má modulárnu konštrukciu, ktorá umožňuje pridať špecializované články a senzory. Každý článok má šesť radiálne symetrických krídelok, ktoré majú v špičke voľne otáčajúce sa koliesko. Robot je schopný pohybu vo vode aj po súši. V strede ohybnej gumovej časti sa nachádza kardanový kĺb poháňaný servo-motormi uloženými v pevných častiach hada.



Obr. 21: Vode-odolný robot ACM-R5 [8]

## 2.2.5 Had s poháňanými plutvami

Existujú aj roboty kombinujúce pohon článkov a prídavného pohonu. Vzorom pre tohto robota je salamandra. Konštrukcia je jednoduchá, dokáže plávať na vode a používa typ pohybu „lateral undulation“ v kombinácii s plutvami.

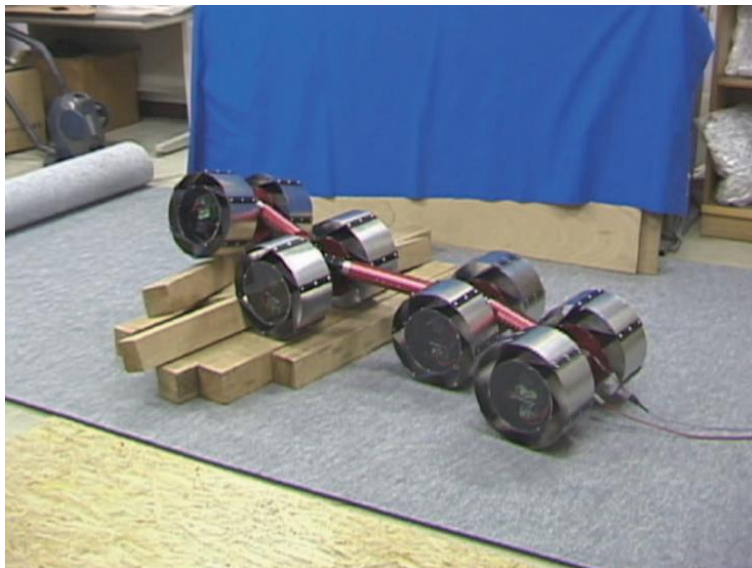


Obr. 22: Robot ACM-R5H schopný splývania s hladinou vody [11]



### 2.2.6 Had s poháňanými kolesami

Tento had používa hnané kolesá pre rýchlo a energeticky efektívny presun po náročnom teréne bez veľkých medzier na povrchu. Jeho článkovitá konštrukcia mu umožňuje prispôbiť sa profilu terénu.



Obr. 23: Robotický had Genbu s poháňanými kolesami [8]

Kolieska môžu byť nahradené pásmi pre lepšie rozloženie váhy pozdĺž tela hada. Séria robotov Soryu bola vytvorená pre záchranné zložky, ktoré prehľadávali trosky po zemetrasení. Celkovo bolo vytvorených 6 typov robotických hadov, s rôznymi počtami článkov, polohou pásov a mechanizmami natáčania kĺbov.



Obr. 24: Komerčne dostupná verzia robota Soryu-C [8]

## 3 VLASTNÉ RIEŠENIE

### 3.1 Použiteľný hardware a software

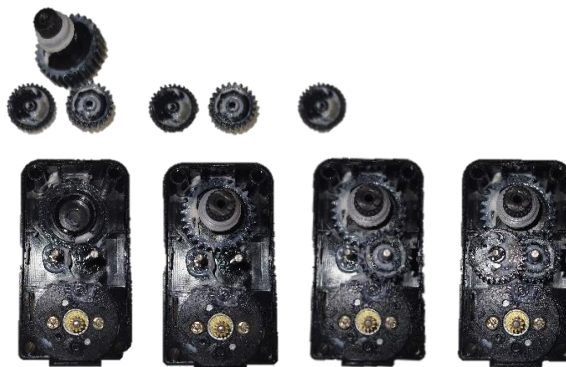
Zadanie práce vyžaduje návrh a konštrukciu hada z výukových stavebníc Dynamixel Bioloid, dostupných v dielňach Ústavu automatizácie. Robot využíva pohyby získané zo simulácií z predošlej autorovej bakalárskej práce. Výber komponentov bol volený s ohľadom na kompatibilitu, ako fyzickú tak aj softvérovú. V zozname sa nachádzajú kombinácie softvéru a hardvéru, ktoré boli zvažované pre riadenie a komunikáciu medzi jednotlivými komponentami. Všetky otestované možnosti sú zhrnuté nižšie. Posledná možnosť bola navyše vhodná pre naše potreby a preto bola nakoniec použitá.

Možnosti:

- Použitie RC vysielača/prijímača spolu s Arduino mikro-kontrolérom a Dynamixel shield-om pre komunikáciu s aktuátormi.  
Výhody: Jednoduché a kompaktné riešenie  
Nevýhody: Veľmi vysoká cena, zložitejšie debugovanie.
- Použitie Arduino mikro-kontroléru série MKR pre generovanie výstupných signálov za pomoci dodatočného shield-u pre komunikáciu s Dynamixel aktuátormi.  
Výhody: Možnosť bezdrôtovej komunikácie, možnosť pridať dodatočné senzory.  
Nevýhody: Nekompatibilné knižnice pre ovládanie aktuátorov
- Použitie mikro-kontroléru CM-530 zo sady Bioloid.  
Výhody: CM-530 používa kľúčové pozície (key-frames), ktoré robot zaujme na určitú dobu. Všetky pohyby sú definované v tabuľke.  
Nevýhody: CM-530 nedokážeme riadiť v reálnom čase.
- Vytvorenie vlastnej ovládačky s Arduino mikro-kontrolérom, umiestneným v ovládačke. Arduino vysiela do počítača vstupy užívateľa, kde zvolený program vstupy spracuje a generuje inštrukcie jednotlivým aktuátorom robotického hada cez adaptér USB2Dynamixel.  
Výhody: Riešenie umožňuje použitie výkonu počítača pre generovanie inštrukcií, had obsahuje len prevodník. Programy pre generovanie inštrukcií sú napríklad: Matlab, Simulink, C++, Python, LabView, Scratch, Scratch 2, ROS, ROS 2.  
Nevýhody: Komunikácia a napájanie káblom obmedzuje fyzický dosah.
- Arduino Leonardo je použité pre generovanie inštrukcií aj komunikáciu s počítačom. Stará sa aj o spracovanie vstupu od užívateľa, ktorý môže parametre pohybu meniť v reálnom čase. Možnosť pripojenia dodatočných senzorov.  
Nevýhody: Obmedzená pamäť a výpočetný výkon Arduina Leonardo sú dodatočné limity programu

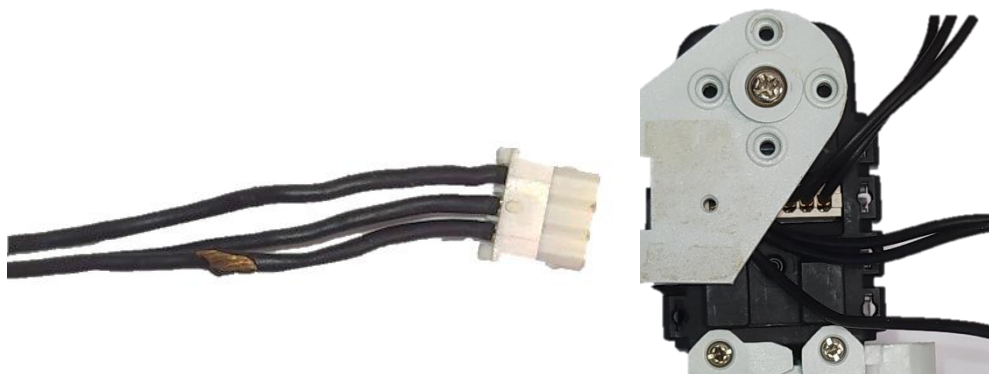
### 3.2 Stavebnica Bioloid

Pri príprave a oboznamovaní sa s dielcami sady Dynamixel Bioloid Pro bola vykonaná inventarizácia setov. Bolo nájdených niekoľko poškodených káblov a nefunkčný aktuátor AX-12. Nefunkčnosť aktuátora bola preskúmaná. Poškodená časť sa nachádzala za predným krytom. Jedno plastové koliesko prevodovky bolo prasknuté, čo spôsobovalo zlyhanie prenosu krútiaceho momentu na výstupnú osku. Náhradná prevodovka bola nájdená aj na stránkach Českého dodávateľa súčiastok. Na obrázku (Obr. 25) je zobrazený priebeh rozoberania prevodovky. Podľa týchto obrázkov je skladanie prevodovky jednoduchšie a po zložení dá sa vizuálne porovnať s obrázkom.



Obr. 25: Prevedovka aktuátora AX-12

Poškodené káble nie sú úplne nefunkčné a dajú sa jednoducho opraviť izolačnou páskou. Izolácia po čase krehne a pri pohybe aktuátora môže dostať k stretu rámu s káblami. Rám potom strhne izoláciu na jednej z krajných žíl kábla (Obr. 26). Vďaka konštrukcii kábla a nízkeho napájacieho napätia (12V) je odhalenie krajných žíl bezpečné. Žila s kladným napätím sa nachádza v strede, medzi žilou s uzemnením a žilou komunikácie ktorá pracuje s napätím 5V. Prípadný skrat medzi krajnými žilami s okolím spôsobí rušenie komunikácie, ale nezničí zdroj ani aktuátor. Poškodené káble boli použité na vytvorenie adaptéra pre napájanie a komunikáciu.



Obr. 26: Poškodený kábel a príčina vzniku

### 3.2.1 Aktuátory

Aktuátory použité pre zostavenie robotického hada sa nazývajú AX-12+. Pochádzajú zo série Dynamixel nahradenej sériou Dynamixel-X. Dokumentácia je dostupná online na stránkach výrobcu. Aktuátor sa skladá z motora, prevodovky a riadiaceho obvodu. Aktuátory AX-12+ majú plastové prevody, ktoré boli vylepšené na oceľové prevody v aktuátoroch AX-12A. Vonkajší kryt motora bol taktiež vylepšený a obsahuje otvory pre lepšie vedenie káblov. Pohon aktuátora zabezpečuje DC motor. Prevodový pomer prevodovky je 254:1, pre vyšší krútiaci moment. (Obr. 27)

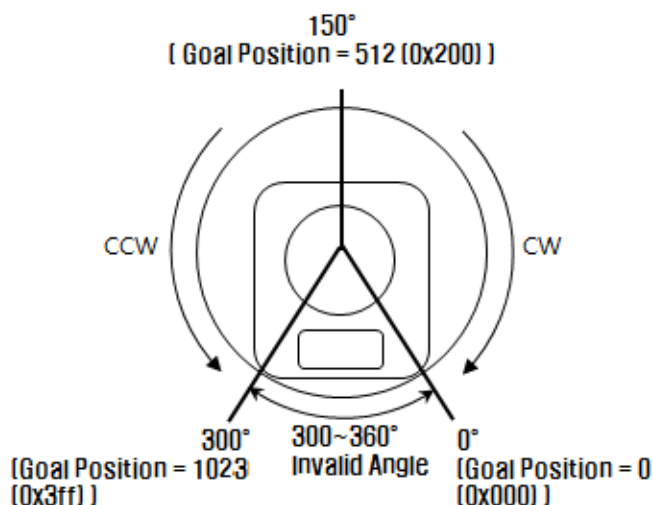


Obr. 27: Aktuátor Dynamixel AX-12 [12]

Výstupný hriadeľ je uložený v potenciometri snímajúci aktuálne natočenie, ktorý je použitý pri riadení so spätnou väzbou, vytvárajúc servo motor. Riadenie vykonáva vstavaný mikro-kontrolér ATmega 128. Aktuátor má dva módy pohybu so spätnou väzbou a bez spätnej väzby.

Bez použitia spätnej väzby sa motor otáča ako obyčajný DC motor. Potenciometer sníma natočenie, ale nie je možné použiť ho pre polohovanie v reálnom čase. Pre naše potreby je tento mód nepoužiteľný a preto sa oň nebudeme viac zaujímať.

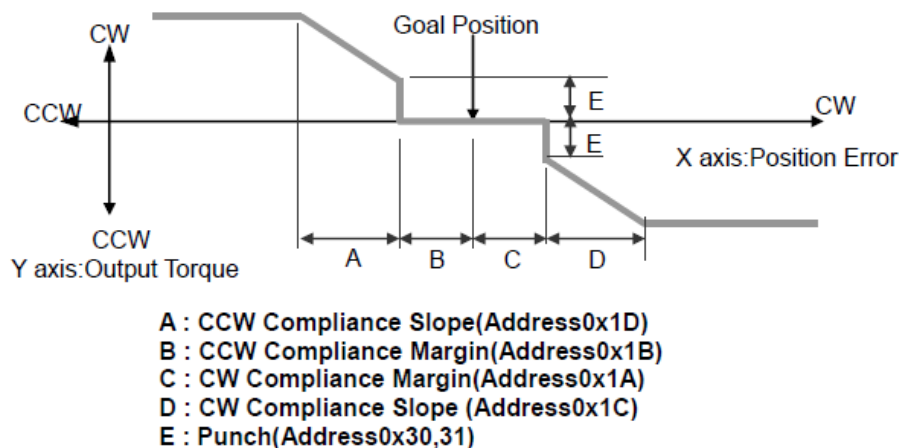
Pri použití spätnej väzby vzniká z aktuátora servo motor. Rozsah otáčania je mechanicky obmedzený 300° rozsahom potenciometra snímajúcim polohu výstupnej hriadele (Obr. 28). Natočenie výstupnej hriadele je kvantované AD prevodníkom na 10 bitov. Z týchto parametrov vyplýva rozlíšenie natočenia približne 0.29 stupňa.



Obr. 28: Pracovný rozsah aktuátora Dynamixel AX-12 [12]

### 3.3 Riadenie

Rozdielom medzi aktuálnym uhlom a žiadaným uhlom vzniká odchýlka „e“, použitá pre vstup regulácie. Na obrázku (Obr. 29) je znázornený vzťah medzi odchýlkou a výstupným prúdom. Zmenou parametrov je možné meniť charakteristiku spätnej väzby.



Obr. 29: Spätaná väzba aktuátora Dynamixel AX-12 [12]

Zmenou parametrov (Obr.29 kóta A,B,C,D,E) je možné upravovať spätnú väzbu každého aktuátora zvlášť. Parametre sú uložené v pamäti RAM. Parameter „A“ a „D“ mení proporcionálne zložku, parameter „B“ a „C“ mení pásmo necitlivosti. Parameter E nastavuje minimálny prípustný prúd. Tento skok pomáha pri prekonaní statického trenia prevodovky a mechanizmu.

Vhodnou voľbou parametrov je možné zlepšenie priechodnosti terénom. Väčšie pásmo necitlivosti sa prejaví menšou presnosťou v polohovaní servo-motorov. Príliš úzke pásmo spôsobí kmitanie okolo žiadanej hodnoty. Zmenou proporcionálnej zložky je možné nastavovať tuhosť aktuátorov. Táto vlastnosť dá použiť pre lepšie kopírovanie povrchu terénu. Pri nižších hodnotách sa robot správa pružnejšie. Príliš nízka tuhosť má za následok nepresné pohyby a pomalé reakcie na zmeny žiadanej hodnoty. [26]



Obr. 30: Vysoký parameter proporcionálnej zložky

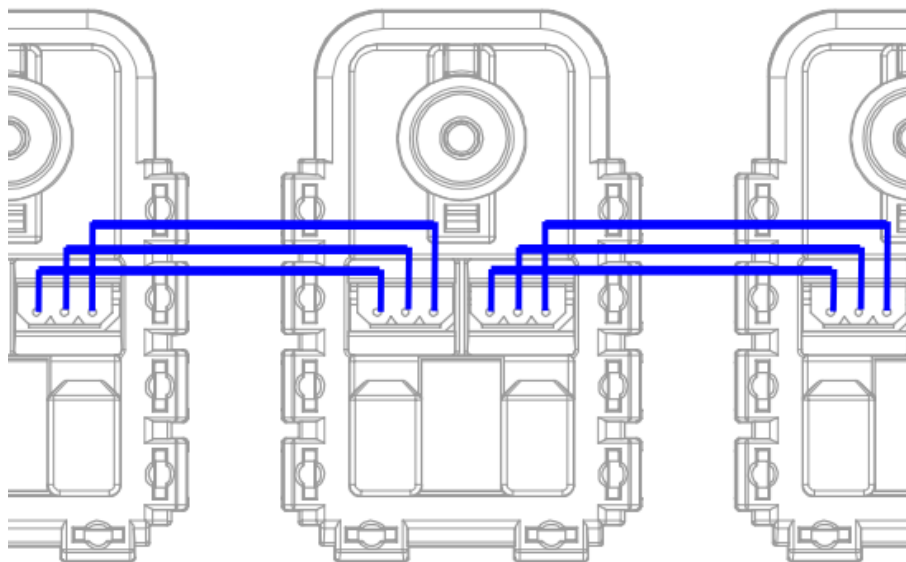


Obr. 31: Nízky parameter proporcionálnej zložky

Medzi obrázkom (Obr.30) a (Obr.31) je zmenený len parameter proporcionálnej zložky, ostatné parametre sú nezmenené. Na obrázku (Obr.30) je robot prehnutý len kvôli vôli v prevodovke a spojených súčasti. Na obrázku (Obr.31) sa robot dotýka podložky pod vlastnou váhou. Po zložení hada na rovný povrch sa pomaly vráti do pôvodného stavu ako pružina.

### 3.4 Komunikácia

Tranzistorovo-tranzistorová logika je použitá pre asynchrónnu seriálovú komunikáciu. Všetky zariadenia sú na spoločnej sieti pospájané stromovou štruktúrou. V konkrétnom prípade robotického hada stačí jedna vetva pozdĺž celej dĺžky hada (Obr.32). Dáta posielať zariadeniam musia odpovedať protokolu Protocol 1.0. Protokol určuje štruktúru posielaťch paketov. Každý paket obsahuje hlavičku, ID zariadenia a inštrukcie.



Obr. 32: Zapojenie typu „Daisy chaining“ [12]

#### 3.4.1 Štruktúra paketu

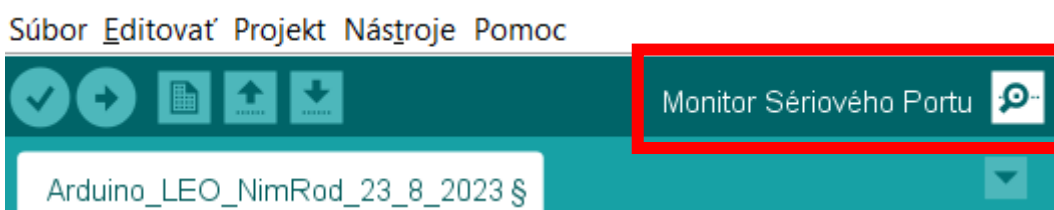
Hlavička značí začiatok prichádzajúceho paketu. Pokiaľ by došlo ku problému s komunikáciou, zariadenie prestane vykonávať inštrukcie a čaká na ďalší paket.

O nezávisle riadenie viacerých aktuátorov na spoločnej sieti sa stará časť paketu obsahujúca ID zariadenia. Ak sa ID zariadenia nezhoduje s ID číslom v prijatom pakete, inštrukcia sa ignoruje. Pokiaľ sa ID zhoduje, inštrukcia sa vykoná a zariadenie spätne odošle paket obsahujúci chybové hlásenia. Výnimku tvoria pakety s ID číslom 254, ktoré sa nazývajú „broadcast“ pakety (všeobecný obežník). Tento paket prijme každé zariadenie, inštrukciu vykoná, ale neposiela odpoveď, ktorá by spôsobila kolíziu pri viacerých zariadeniach v sieti. Kolíziu môžu spôsobiť aj dve rôzne zariadenia v spoločnej sieti s rovnakým ID. Pred pripojením zariadenia do siete je nutné overiť či ID už nie je obsadené. ID je možné zmeniť na hodnotu medzi 0–253.

Inštrukcie väčšinou prepisujú adresy v zariadeniach (žiadané natočenie), alebo žiadajú o preposielanie hodnoty niektorej adresy kontroléru (aktuálne natočenie). Existujú aj špeciálne inštrukcie pre identifikáciu (Ping), resetovanie (Factory Reset) a reštartovanie (Reboot).

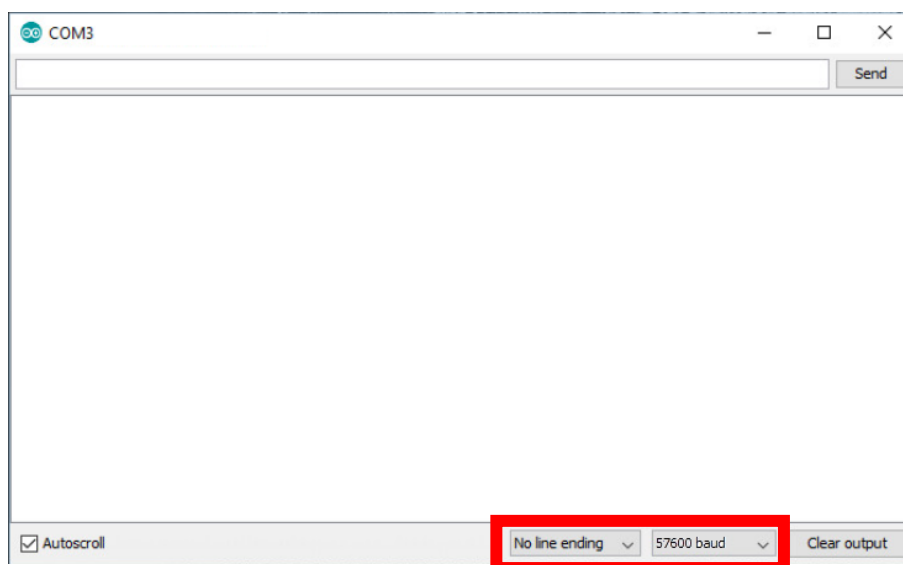
### 3.4.2 Komunikácia a riadenie

Arduino Leonardo je unikátne v tom, že doska bola vyrobená pre komunikáciu cez USB port s počítačom. Arduino obsahuje vstavané knižnice, vďaka ktorým je možné vyrobiť vlastnú klávesnicu, myš alebo joystick. Robota je možné riadiť v dvoch režimoch. Ak robot nie je pripojený s počítačom, použije nastavené hodnoty pohybu. Pokiaľ je robot pripojený káblom USB, je možné meniť parametre pohybov pomocou príkazov cez Monitor sériového portu v programe Arduino IDE. Tlačidlo umiestnené vpravo hore (Obr. 33) otvorí nové okno (Obr.34), do ktorého je možné písať príkazy (zoznam príkazov na ďalšej strane). Dôležitým detailom je písanie medzier (po Anglicky white space). Ak príkaz pozostáva z dvoch častí, je nutné použiť práve jednu medzeru. Arduino načíta segmenty oddelené medzerami a pri neplatných vstupoch vypíše chybovú hlášku. Chybové hlášky sa týkajú nerozpoznaného príkazu alebo neplatnej hodnoty parametra príkazu.



Obr. 33: Zvýraznenie tlačidla, ktoré otvorí okno „monitor sériového portu“

Príkazy sa píšu do vrchného riadku okna, výstup komunikácie sa zobrazuje v okne. Dôležité nastavenie pre správnu komunikáciu je vypnutie ukončovacieho znaku a nastavenie rýchlosti komunikácie na 57600 baud. Ukončovací znak je automaticky pridaný na koniec posielanej správy, avšak v tomto prípade by spôsoboval problémy, keďže Arduino očakáva presnú štruktúru príkazov bez ukončovacích znakov.



Obr. 34: Okno monitora sériového portu, dôležité nastavenia vpravo dole



Príkaz	Názov	Stručný popis funkcie
<b>Základné príkazy</b>		
On	Zapnúť generátor	Zapne cyklus generujúci základné pohyby
Off	Vypnúť generátor	Vypne cyklus generujúci základné pohyby
Inf	Informácie	Výpis tabuľky aktuálneho nastavenie generátora
<b>Nastavenie generátora</b>		
Ho n	Horizontálne odsadenie	Nastavuje konštantnú zložku generátora pre horizontálne aktuátory na hodnotu "n", povolený rozsah: 0-1023, pôvodne: 512
Hk n	Hor. krok	Nastavuje počet vzoriek generátora na hodnotu "n", platí len pre horizontálne aktuátory, povolené len kladné hodnoty odporúčané hodnoty: násobky dvoch
Ha n	Hor. amplitúda	Nastavuje maximálnu výchylku generátora pre horizontálne aktuátory na hodnotu "n", povolené aj záporné hodnoty
Hs n	Hor. rýchlosť	Nastavuje maximálnu rýchlosť horizontálnych aktuátorov na hodnotu "n", rozsah 0 - 1024, 0 = neobmedzená rýchlosť
Vo n	Vertikálne odsadenie	Rovnaké ako pre horizontálny príkaz, ale ovláda len vertikálne aktuátory
Vk n	Ver. krok	Rovnaké ako pre horizontálny príkaz, ale ovláda len vertikálne aktuátory
Va n	Ver. amplitúda	Rovnaké ako pre horizontálny príkaz, ale ovláda len vertikálne aktuátory
Vs n	Ver. rýchlosť	Rovnaké ako pre horizontálny príkaz, ale ovláda len vertikálne aktuátory
Delay n	Oneskorenie	Oneskorenie medzi jednotlivými cyklami generátora, povolené len kladné hodnoty
<b>Debugovanie</b>		
Debug	Výpis	Zapne/vypne výpis generátora v reálnom čase
S	Stop	Zastaví všetky pohyby, stačí "s" pre rýchle napísanie v prípade núdze. Možné alternatívy ako "Stop" alebo "STOP"
Ori	Orientácia	Manuálne spustí vykonanie kontroly orientácie, stačí napísať len prvé tri písmena slova "Orientácia"
Select n	Výber	Výber aktuátora, po výbere možno použiť príkazy: "move" a "pos"
Move n	Pohyb	Natočí aktuátor zvolený predchádzajúcim príkazom na zvolenú hodnotu n
Pos n	Pozícia	Výpis pozície zvoleného aktuátora
<b>Pomocné pohyby</b>		
Worm n	Červ	n je nutné nahradiť číslom udávajúcim počet vykonaní pohybu. Záporné hodnoty vykonajú pohyb v opačnom smere.
Roll n	Kotrmelec	n je nutné nahradiť číslom udávajúcim počet vykonaní pohybu. Záporné hodnoty vykonajú pohyb v opačnom smere.
Turn n	Otočka	n je nutné nahradiť číslom udávajúcim počet vykonaní pohybu. Záporné hodnoty vykonajú pohyb v opačnom smere.
Sit	Sadni	Zastaví všetky pohyby a zvinie robota do polohy pre ukážku
Yes	Áno	Kýva hlavou "Áno", možné použiť len v ukážkovej polohe (príkaz "Sit")
No	Nie	Krúti hlavou "Nie", možné použiť len v ukážkovej polohe (príkaz "Sit")
Slow	Pomaly	Nastaví všetky aktuátory na 10% maximálnej rýchlosti
Cen	Centrovať	Vyrovná všetky aktuátory, stačí napísať prvé tri písmena "Centrovať"

Tab. 1 Zoznam príkazov robotického hada

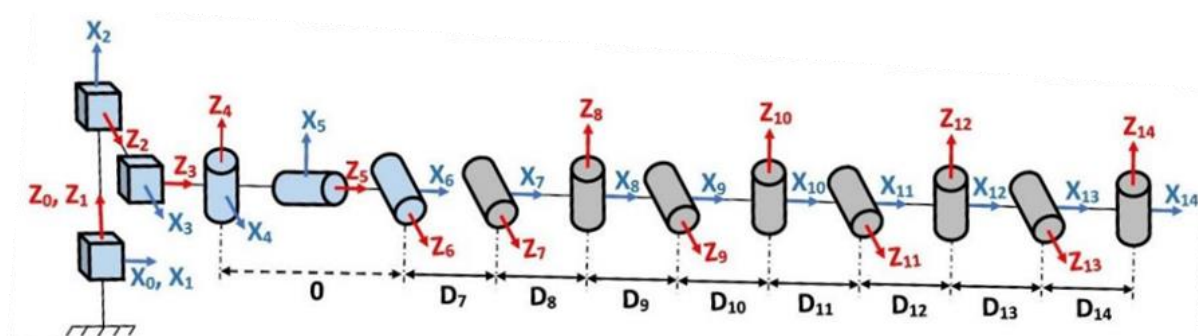
### 3.5 Konštrukcia

Stavebnica Dynamixel Bioloid je produktom kórejskej firmy Robotis. Rám sa skladá zo štyroch typov dielcov na obrázku nižšie.



Obr. 35: Dielce rámu [13]

Prvé tri dielce (Obr. 35 a, b, c) sú použité na vytvorenie pohyblivého spoja medzi aktuátormi hada. Ich vzájomné osi rotácie sú otočené o  $90^\circ$ , aby vytvorili kĺby schopné rotovať v dvoch rovinách ako kardanový kĺb (Obr. 36).

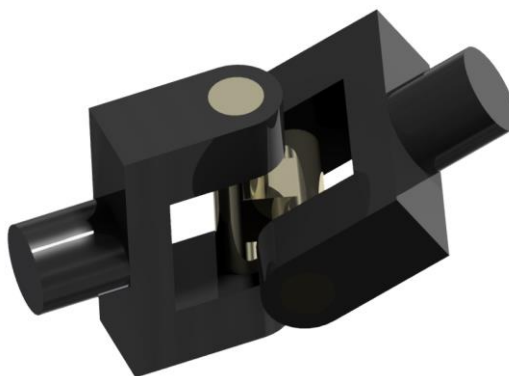


Obr. 36: Kinematická štruktúra [14]

Kardanový kĺb (Obr.37) spájajúci články robota umožňuje natočenie v osách kolmých na pozdĺžnu os, v ktorej prenáša krútiaci moment. Vďaka týmto vlastnostiam sa môže konštrukcia natáčať v 3D priestore, avšak bez rizika zamotania alebo pretrhnutia káblov. Stredový kríž kĺbu je tvorený rámom s pripevnenými aktuátormi umožňujúcimi kontrolované natáčanie. Kĺby sú spojené štvrtým dielcom rámu (Obr. 35, dielec c). Články sú identické a je ich možné pospájať pre vytvorenie dlhšieho robota. Limitom je odber motorov. Kabeláž je vedená pozdĺž robota, ktorá spája všetky aktuátory.



Obr. 37: Rozložený kardanový spoj (dve vidlice a kríž v strede)

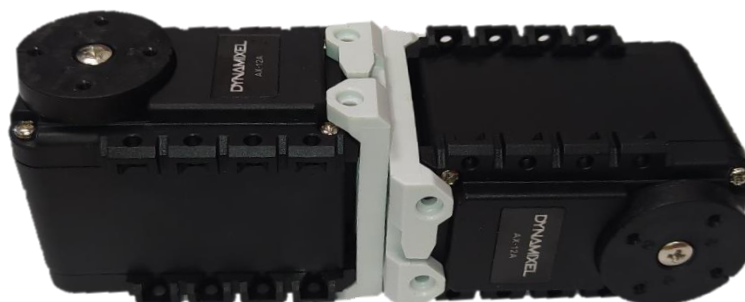


Obr. 38: Zložený a natočený kardanový spoj

Dva aktuátory sú spojené dokopy, uhol medzi výstupnými osami aktuátorov je pootočený o  $90^\circ$ . Táto zostava pripomína strednú časť kardanového kĺbu, kríž. Vidlice sú tvorené dvoma plastovými súčiastkami v tvare písmena „U“ obr.(35 c). Kardanový kĺb sa voľne otáča v oboch osách kríža. V tomto sa odlišuje s robotickým kĺbom, ktorý je v týchto osách poháňaný aktuátormi. Schopnosť natáčanie v oboch rovinách umožňuje vedenie káblov cez miesta, ktoré káble nenatáhujú ani nedrvia. Množstvo robotických hadov natáčajúcich sa v dvoch rovinách využíva práve tento princíp kardanového kĺbu. Každý článok hada je tvorený identickým kĺbovým mechanizmom a preto je ich dĺžky možné zväčšiť pridaním ďalších článkov.



Obr. 39: Rozložený článok robota



Obr. 40: Stredová časť článku, dva aktuátory spojené s 90° otočením, tvoriace stredový kríž kardanového spoja

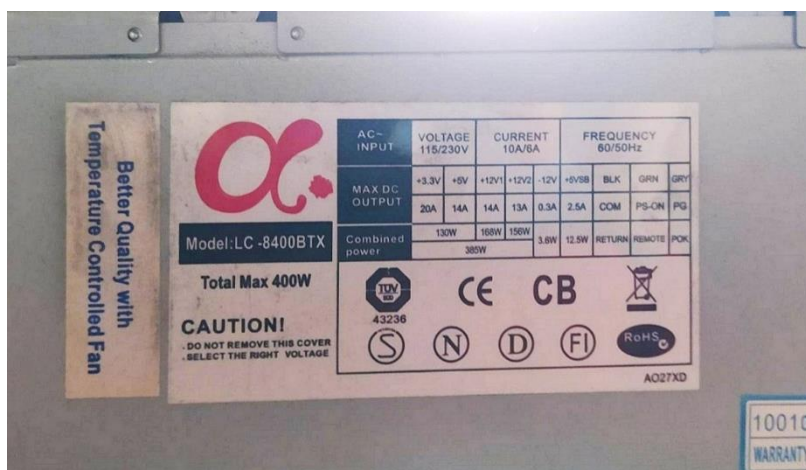
### 3.6 Napájanie

Robot za sebou ťahá dvoj-žilový kábel pre napájanie. Napájacie napätie musí byť v rozsahu 7-14V a každý motor vyžaduje prúd 1.2A.

Aktuátory v móde serva vyžadujú tvrdý zdroj napätia. Časté zmeny rotácie a rozbeh servo-motorov vytvára vysoké odbory prúdu, ktoré je treba vyhladiť. Následný pokles napätia slabého zdroja spôsobí resetovanie mikro-kontroléru a prepísaní RAM na pôvodné hodnoty. Najväčší problém je resetovanie maximálneho limitu rýchlosti, krútiaceho momentu a parametrov ovplyvňujúcich charakteristiku spätnej väzby. Po resetovaní je motor schopný komunikovať normálne, ale resetované parametre spôsobia prudké a trhané pohyby, spôsobujúce ďalšie poklesy napätia a resetovanie okolitých motorov. Je možné, že preťažený motor resetuje len svoj mikro-kontrolér a okolité motory ostatnú neovplyvnené. V najhoršom prípade jeden preťažený motor spôsobí reťazovú reakciu, ktorá resetuje všetky motory. Robot sa stane neovládateľný, pretože všetky limity sú resetované a akýkoľvek pohyb je trhaný, rýchly a môže spôsobiť ďalšie výpadky.

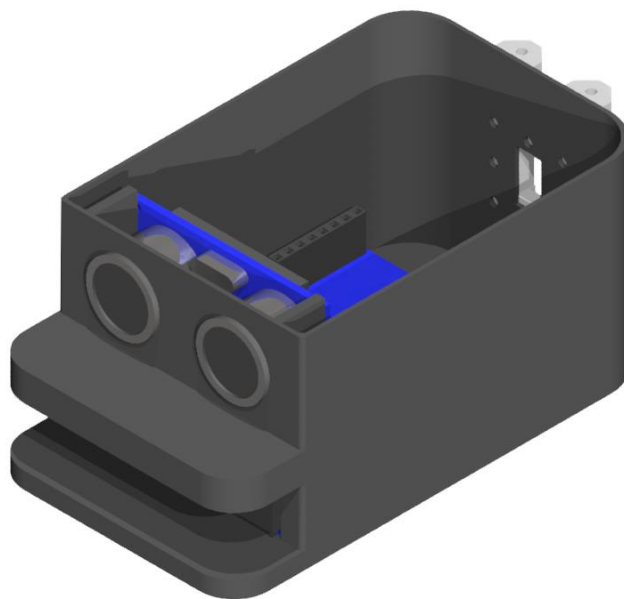
Použitím výkonného tvrdého zdroja prúdu (napr. počítačový zdroj, Obr.41), vyvarovaniu sa preťažovaniu aktuátorov a nepoužívaním náhlych pohybov je možné predísť výpadkom.

Rázové odbory prúdu možno znížiť úpravou spätnej odozvy. Nastavenie nižšieho maximálneho limitu odoberaného prúdu znižuje aj maximálny krútiaci moment. Menej agresívna charakteristika spätnej väzby taktiež dokáže znížiť prudké odbory. Ak aj napriek tomu dôjde k výpadku, ukončením sekvencie vykonávaných pohybov a opätovným nastavením limitov, je možné pokračovať bez väčších problémov. Ak by došlo ku problémom, kolízií alebo zaseknutí a je program je nutné ukončiť náhle, port pre komunikácie ostane obsadený.



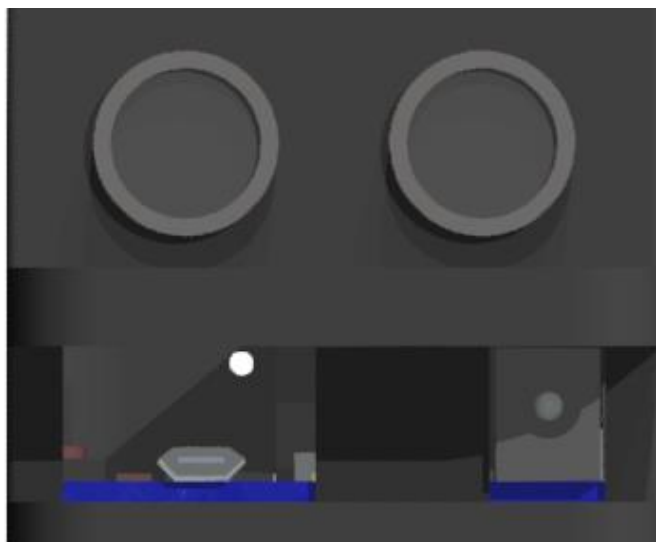
Obr. 41: Počítačový zdroj pre napájanie robota

### 3.7 3D tlačené dielce

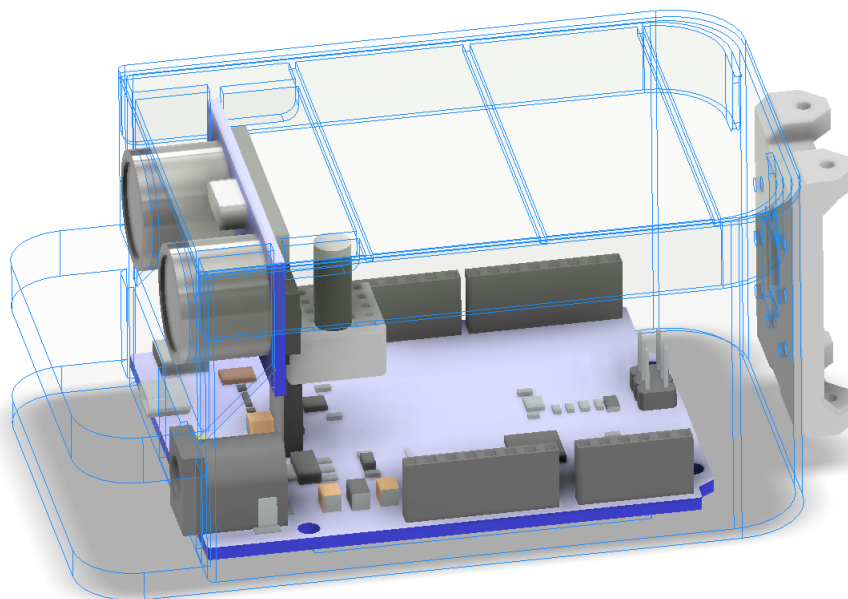


Obr. 42: Render hlavy robotického hada bez vrchného krytu

Konštrukcia hlavy bola vytvorená pre uloženie potrebných senzorov a riadiacej dosky Arduino Leonardo. Hlava chráni elektroniku pred poškodením. Oči pripomínajúci tvar je ultra-zvukový senzor vzdialenosti, ktorý sníma odraz zvuku od prekážky pred robotom a určuje vzdialenosť pomocou časového rozdielu medzi vyslaným zvukom a ozvenou. Na prednej strane na spodku sa nachádzajú ústa. Praktický význam tejto časti je v ochrane konektorov pred vyložením pripojeného dátového alebo napájacieho kábla.

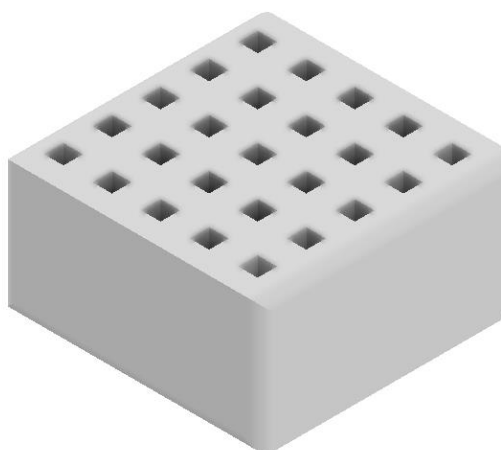


Obr. 43: Predný pohľad zobrazujúci konektory arduina Leonardo



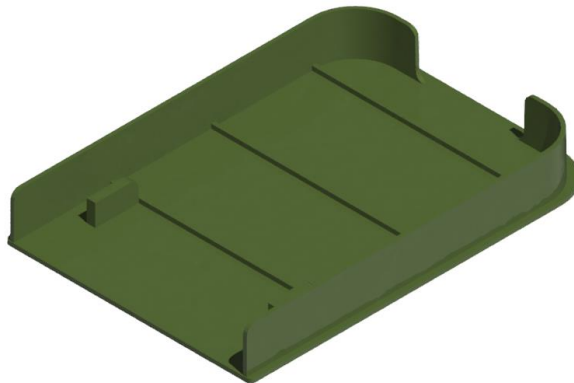
Obr. 44: Vnútorné rozloženie komponentov

Vnútri hlavy sa nachádza elektronika pre riadenie, snímanie vstupu senzorov, spracovanie údajov a komunikáciu. Na spodnej časti je uložené arduino Leonardo, ktoré je prilepené oboj-stranou páskou na dvoch výstupkoch tak, aby sa komponenty a konektory na spodnej strane arduina neprišli do kontaktu so krytom. Na vrchej časti sa nachádza ultrazvukový senzor pre snímanie prekážok. Oba senzory, ultrazvukový senzor a senzor náklonu sú vložené do malého nepájivého poľa o rozmere 5x5. Z tohto poľa vedú káblíky do dosky Leonrado na príslušné digitálne vstupy. Nepájivé pole bolo vymodelované podľa rozmerov klasického nepájivého poľa, ktoré by do hlavy nezmestilo. Z nepájivého poľa boli vybrané kontaktné pliešky, ktoré spájajú 5 susedných pinov dokopy. Bolo nutné mať 6 vstupov/výstupov. Dva výstupy požadovali pripojenie na zem (pin GND), čím ostávalo 5 nezávislých vstupov/výstupov a tak vzniklo pole o rozmeroch 5x5.



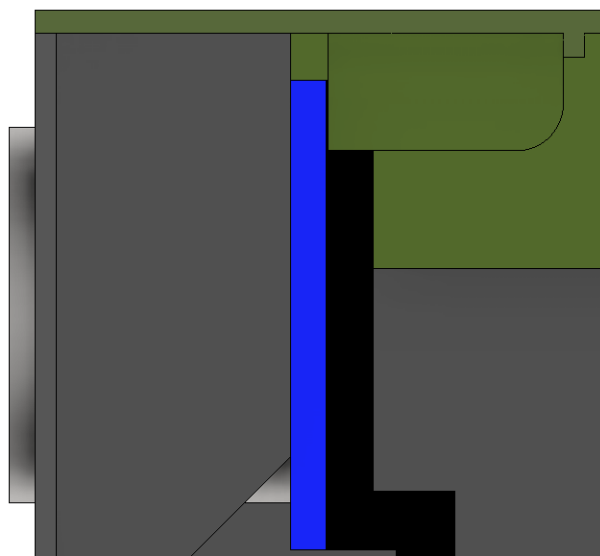
Obr. 45: Detail na model nepájivého poľa 5x5

Na spodnej časti vrchného krytu hlavy sa nachádzajú rebrá a dva zuby. Rebrá zvyšujú tuhosť vrchnej plochy a znižujú ohýbanie a skrúcanie celého dielca, pričom zachovávajú nízku hmotnosť a nezaberajú veľa vnútorného priestoru.



Obr. 46: Pohľad zospodu na vrchný kryt hlavy

Pevné uchytenie senzora je vďaka materiálovému styku. Modrá doska plošných spojov je z oboch strán stlačená medzi spodnou časťou hlavy (sivý materiál na obr. (47) ) a vrchným krytom (zelený materiál). Plasty sa opierajú o dosku na okrajoch, v miestach prichytenia a nezasahujú do funkčných častí plošného spoja.



Obr. 47: Bočný prierez hlavou zobrazujúci uchytenie dosky plošných spojov senzora

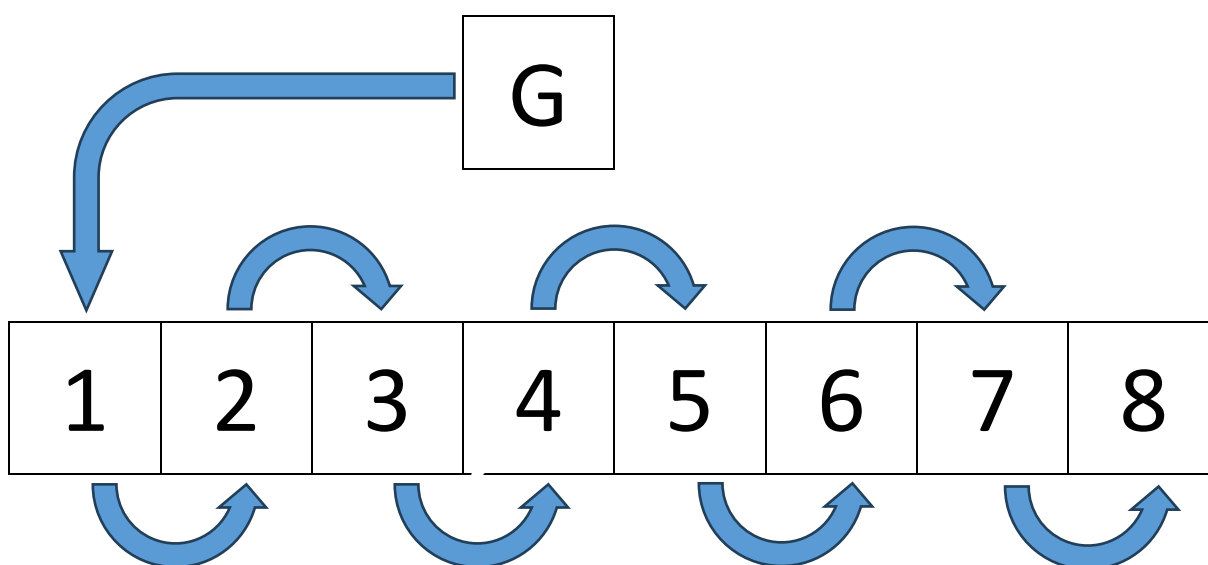




### 3.8 Matematický popis

Riadenie pomocou rovnice vlnenia je jednoduché a umožňuje množstvo rôznych pohybov len zmenou vstupných parametrov použitých pre generovanie a pridelovanie vygenerovaných hodnôt aktuátorom. Generovanie hodnôt pre všetky aktuátory pre každý cyklus programu nie je nutné a z experimentov v simuláciách z bakalárskej práce boli zrejme problémy spojené s prudkou zmenou parametrov. Riešením oboch zmienených problémov je použitie posuvného registra. Hodnoty sú uložené v zozname a je ich možné posúvať.

Hodnoty vlnenia z generátora sa vložia na prvé miesto do zoznamu (array) a pri opätovnom zavolaní sa posunú o jedno miesto doprava obr.(48). Každý článok má priradené jedno pole v zozname, začínajúc hlavou, ktorej je priradené prvé pole. V prípade konštrukcie planárneho robota so všetkými aktuátormi schopnými pohybu v jednej rovine stačí register s posunom o jedno miesto.



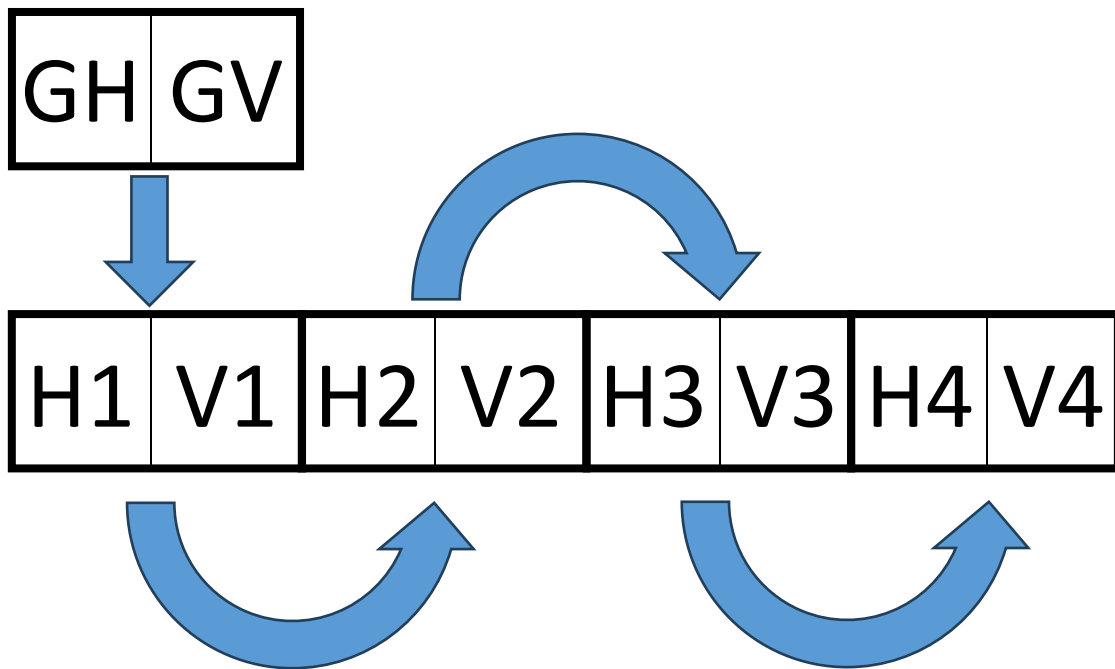
Obr. 48: Posuv registra o  $n=1$  prvkov, na prvé miesto sa dosadí hodnota z generátora

V prípade robotického hada schopného pohybu v dvoch rovinách je nutné odlišovať horizontálne a vertikálne otáčajúce sa aktuátory. Preto sú použité dva generátory vlnenia, každý pre odpovedajúce aktuátory a posuv registra je vždy o dve miesta.

Generátory sú označené ako **Generátor Horizontálneho** a **Generátor Vertikálneho** vlnenia Obr.(49). Rovnice použité generátormi sú identické, odlišujú sa menami premenných, aby každý mohol používať odlišné parametre. Správnym nastavením parametrov je možné napodobniť typické pohyby biologických hadov. Rovnice generujúce pohyby:

$$H = h_a * \cos(2 * t * (\pi / h_k)) + n_o$$

$$V = v_a * \cos(2 * t * (\pi / v_k)) + v_o$$



Obr. 49: Posuv registra o  $n=2$  prvkov, na prvé dve miesta sa dosadí hodnota z generátorov

### 3.9 Popis kódu

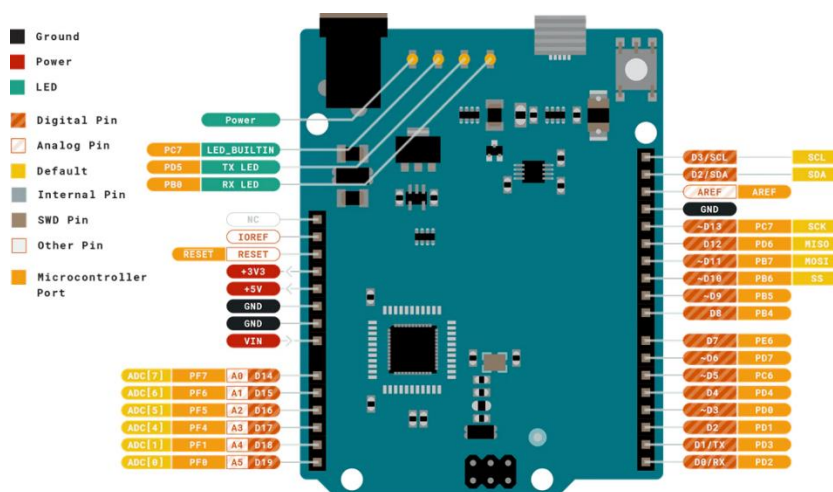
Prvá časť kódu obsahuje poznámky ku fyzickému zapojeniu DuPont káblov k perifériám a aktuátorom.

Komunikácia funguje prepínaním jedného pinu medzi režimom Tx a Rx. Režim Rx vysiela signál po zbernici a po odoslaní je pin nastavený na režim Tx a sníma signály zbernice. Doska Arduino musí mať pin, ktorý podporuje prerušenia (interrupt) a podporuje obe knižnice „SoftHalfDuplexSerial“ a „DynamixelAX“.

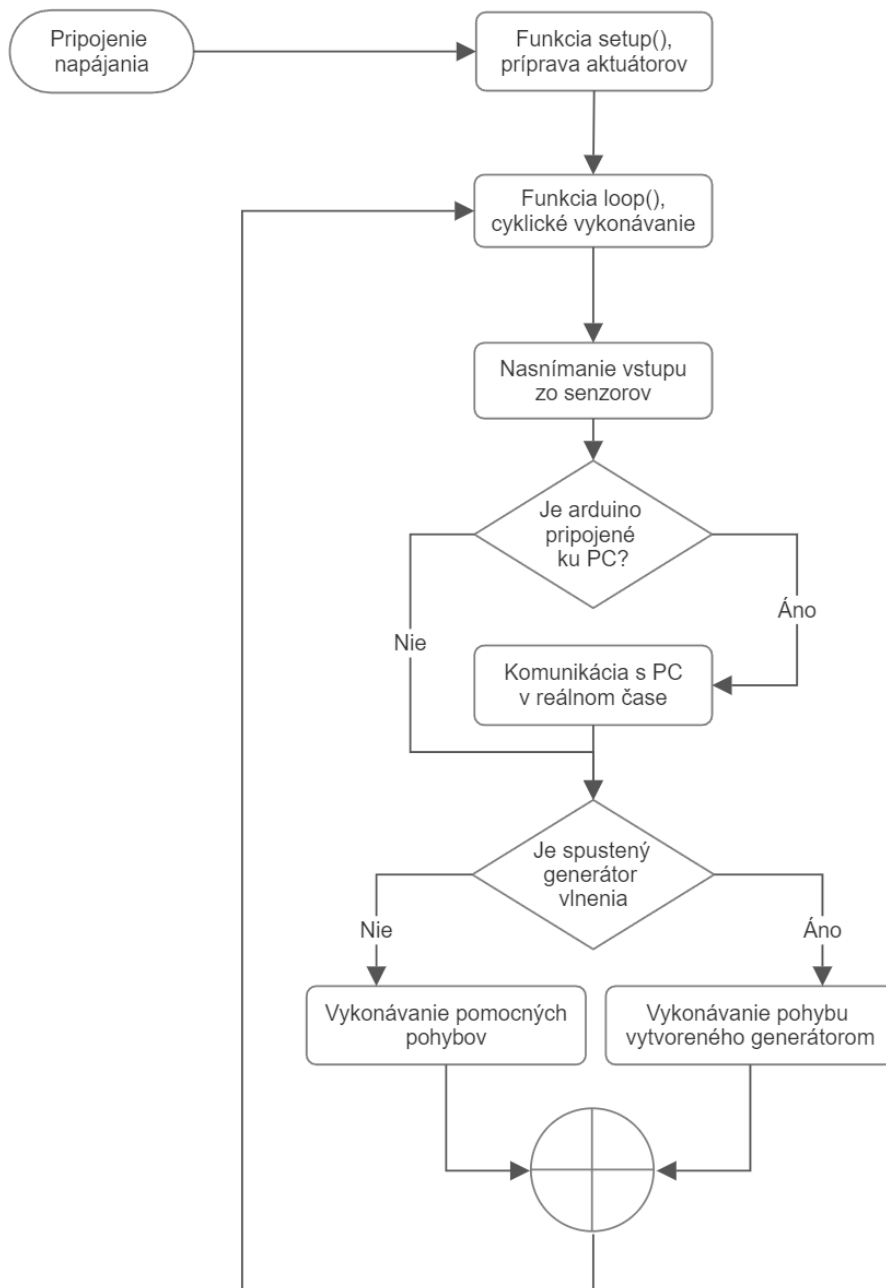
Senzor náklonu, klopný spínač, spája pin, ktorý je napájaný na 5V a nastavený na režim „pull-up“. Pri zopnutí sa uzatvorí obvod so zemou. „Pull-up“ režim využíva vstavaný odpor s hodnotou 20 k $\Omega$  a sníma zmeny napätia. Logika je v pohľadu spínača obrátená, pretože zopnutím obvodu (alebo napr. stlačením tlačidla) sa napätie stiahne a hodnota na pine je logická nula. Vhodnou polohou senzora je možné invertovať logiku tak, aby sa bol obvod zopnutý pri správnej orientácii a na pine bola logická 1 len v prípade, že je robot prevrätý.

Pre snímanie vzdialenosti bol použitý ultrazvukový senzor HC-SR04. Ide o bežný akustický senzor, ktorý vysiela ultrazvuk a sníma odraz od prekážky. Pre výpočet vzdialenosti je nutné zmerať čas medzi pulzom poslaným do senzora a pulzom ktorý senzor prijme. Pulz odoslaný na pin „TRIG“ musí mať presnú dĺžku 10 ms. Program následne zmeria čas pokým sa na pin vstupu, zvaný „ECHO“ dostane impulz od senzora a vynásobením konštantou dostaneme zmeranú vzdialenosť. Konštanta je daná od výrobcu. Napriek tomu bol senzor otestovaný, ale ukázalo sa, že pôvodné nastavenia sú presné a senzor je skalibrovaný.

Použitie knižnice: „SoftHalfDuplexSerial“ slúži pre prepínanie jedného pinu medzi módom prijímača (Rx) a vysielača (Tx). „SoftSerial“ je skrátene „Software Serial“, čiže softvérový seriálový port. Sériová komunikácia posieľa dáta za sebou po vodičoch, oproti paralelnej, ktorá posieľa paralelne po viacerých vodičoch naraz. Softvérový port znamená že nejde o fyzický konektor, ale iba pin na doske. Arduino má vstavaný fyzický USB konektor pre sériovú komunikáciu, ktorý bol použitý pre pripojenie s počítačom. HalfDuplex je typ obojsmernej (Duplex) komunikácie, ktorej smer dát prebieha striedavo. Vždy je teda použitá „polovica“ (Half) komunikačného kanálu. V poslednej časti kódu zapojenia je deklarácia premenných týkajúcich sa zapojenia a fungovania periférií a aktuátorov. Pokiaľ by bolo nutné použiť iné zapojenie, je potrebné zistiť, či alternatívne zapojenie pinov podporuje funkcie pinov pôvodných. Funkcie viazané ku pinom je vidieť na Obr.50.



Obr. 50: Arduino Leonardo Pinout [17]



Obr. 51: Vývojový diagram hlavného priebehu kódu

```

/*
Wiring: Communication
- COMM: Pin 8
- GND: GND

Wiring: Tilt sensor
- TILT: Pin 9 (must be set as pullup)
- TILT: GND

Wiring: Ultrasonic Sensor HC-SR04 -> Arduino:
- VCC -> 5VDC
- TRIG -> Pin 11
- ECHO -> Pin 10
- GND -> GND
*/

#include <SoftHalfDuplexSerial.h>
#include <DynamixelAx.h>

// Set pins and ports
softHalfDuplexSerial port(8); // data pin 8
dxlAx dxlCom(&port);
const int tiltPin = 9; // pullup for tilt switch
const int trigPin = 11; // TRIG pin
const int echoPin = 10; // ECHO pin

// tilt switch
int _time;
bool _state = true, _state_new;

// sonic sensor
String text;

// for communication
String _string;

```

Premenné pre generátor pohybu a počítadlá pre pred-generované pomocné pohyby.

```

// DEBUG
bool show_debug = false;
bool display_pos = false; // display current position of actuator
int display_pos_ID = 4; // show position of n-th actuator

// basic wave generator
bool activate = false;
// movement parameters
int t = 0, k = 2, o = 0; // time, step, offset (between H/V waves)
int Ha = 0, Hk = 2, Ho = 512, Hs = 100; // Horizontal wave: amplitude,
step, offset (steering), speed of actuators
int Va = 0, Vk = 2, Vo = 512, Vs = 100; // Vertical wave: amplitude,
step, offset (steering), speed of actuators
int _timer_steer = 0, delay_time = 30;

// roll movement: 0 = stop
int roll = 0; // roll around
int worm = 0; // worm type movement
int _turn = 0; // turn on spot

```

### 3.9.1 Nastavenie pre aktuátory Dynamixel

Každý aktuátor ma unikátne identifikačné číslo, zvané ID. Poradie zapojenia aktuátorov je dôležité. V aktuálnej konštrukcii sa podarilo nájsť 8 motorov začínajúcich od ID 1. Aby bolo možné použiť aj motory s iným identifikačnými číslami, zoznam `_IDs` musí byť aktualizovaný podľa aktuálneho zapojenia. Pokiaľ nesedia ID aktuátorov a ID v zozname, program bude posielat' dáta, ktoré budú ignorované všetkými aktuátormi. Ak by nesedelo poradie zoznamu s fyzickým poradím zapojenia aktuátorov, výsledné pohyby by neodpovedali zadaným. ID aktuátora je na štítku na zadnom paneli. Kód by mal fungovať pre všetky aktuátory série AX. Rôzne modely môžu mať iný rozsah otáčania. Rozsah otáčania je limitovaný aj fyzickými kolíziami jednotlivých článkov. Pre vyhnutie sa problémom s kolíziou susedných článkov boli vytvorené softvérové limity `u_limit` (upper limit = horný limit) a `l_limit` (lower limit = spodný limit). Konštanta „middle“ (stred) je hodnota pri ktorej je aktuátor v strede rozsahu natočenia.

```
// for dynamixel
const int u_limit = 762; // upper limit of rotation
const int middle = 512; // middle position of rotation
const int l_limit = 262; // upper limit of rotation

const int _IDs[8] = {1, 2, 3, 4, 5, 6, 7, 8}; // SET IDs of actuators
int _angles[8] = {512, 512, 512, 512, 512, 512, 512, 512}; // middle position
```

### 3.9.2 Inicializačné funkcie na začiatku chodu programu

Vlastné funkcie volané pred cyklickým chodom programu musia byť definované na začiatku dokumentu. Zvyšok funkcií, volaných cyklickým chodom, je definovaných na konci dokumentu. Príkaz `Dx_goal` nastaví natočenie aktuátora na zvolenú hodnotu, ktorá je v bezpečnom rozsahu. Prvým parametrom je poradové číslo aktuátora, vďaka čomu užívateľ nemusí vedieť ID aktuátora, stačí poradové číslo od hlavy začínajúce od 1.

```
////////// - custom function - //////////
void Dx_goal(int n, int input) {
    dxlCom.setGoalPosition(_IDs[constrain(n - 1, 0, 7)], constrain(input, l_limit, u_limit));
};
void Dx_move_all() {
    for (int i = 0; i < 8; i++) {
        dxlCom.setGoalPosition(_IDs[i], constrain(_angles[i], l_limit, u_limit));
        delay(delay_time);
        if (show_debug) {
            Serial.print(_angles[i]);
            Serial.print(" ");
        }
    }
}
void Dx_center() {
    for (int i = 0; i < 8; i++) {
        dxlCom.setGoalPosition(_IDs[i], middle);
        delay(50);
        // Serial.print(_angles[i]);
        // Serial.print(" ");
    }
    //Serial.println();
}
//////////
```

### 3.9.3 Necyklické volanie kódu

Necyklické volanie kódu prebehne len raz. Využíva sa pre nastavenie pinov dosky a nastavenie maximálnej rýchlosti aktuátorov. Funkcia `Serial.setTimeout()` slúži pre nastavenie času v milisekundách, ktorý program čaká pri prijatí dát. Pôvodná hodnota je 1000 ms (1 sekunda), čo je príliš rušivé pri ovládaní v reálnom čase, pretože dôjde ku úplnému zastaveniu programu a pohybu robota. Čas 50 ms je dostatočný pre komunikáciu a nepostrehnuteľnému zdržaniu pohybu pri odoslaní príkazu. Nižšie hodnoty by mohli spôsobiť kolízie na zbernici.

```
void setup() {
  Serial.begin(57600);
  while (!Serial) {} // wait for serial port to connect
  pinMode(trigPin, OUTPUT); // configure the trigger pin to output mode
  pinMode(echoPin, INPUT); // configure the echo pin to input mode
  pinMode(tiltPin, INPUT_PULLUP); //configure pin as an input
  Serial.setTimeout(50); // maximum milliseconds to wait for serial data
  dxlCom.begin(57600); // DO NOT CHANGE, adjust actuator to this baudrate
  delay(100);

  for (int i = 0; i < 8; i++) {
    dxlCom.setMovingSpeed(_IDs[i], 100);
    delay(100);
  };
};
```

Po zapnutí sa robotický had narovná a ohne posledný článok dostrany, aby sa nekotúľal po pozdĺžnej ose. Následne skontroluje senzor náklonu. Množstvo čakania v kóde je nutné pre odstránenie rušivých vplyvov ako je skákanie guľičky v senzore („bouncing“). Senzor náklonu obsahuje dva kontakty spojené vodivou guľičkou. Pri správnej orientácii spojí guľička oba kontakty spolu, pri vychýlení sa kontakt rozpojí. Guľička má zotrvačnosť, ktorú treba vziať do úvahy. Najjednoduchším riešením pri spúšťaní robota je počkať pár sekúnd pre ustálenie guľičky. Počas pohybu je nutné použiť filter na potlačenie vplyvu skákania guľičky („de-bouncing“) opakovaným meraním a detekciou zmeny trvajúcej niekoľko po sebe idúcich cyklov. Počet cyklov bol nastavený experimentálne, príliš nízke hodnoty reagujú veľmi citlivo na pohyb a senzor hlási prevrátenie aj pri otrasoch spôsobených pohybom. Veľmi vysoké hodnoty spôsobujú zdržanie, takže robot je zbytočne dlhý čas prevrátený.

```
// Make an L shape and check orientation

Dx_center(); // straighten all joints
delay(1000);
Dx_goal(7, l_limit); // make L shape
delay(1000);
Serial.print("\nChecking orientation");
delay(500);
Serial.print(".");
delay(500);
Serial.print(".");
delay(500);
Serial.print(". ");
_state = digitalRead(tiltPin); //check if upside down
```



Pri detekcii prevrátenia robot spustí poradie pohybov ktoré ho prevrátia naspäť. Na konci pohybu je skontrolované, či bolo prevrátenie úspešné, pokiaľ nie, pokus sa opakuje. Pri množstve experimentov sa had vo väčšine dokázal prevrátiť do správnej orientácie už pri prvom pokuse.

```
while (_state) {
  Serial.println("Upside down! Trying to right itself");
  Dx_goal(6, l_limit);
  Dx_goal(7, u_limit);
  delay(2000); // wait for actuators to reach position
  Dx_goal(6, middle);
  Serial.print("\nChecking orientation");
  delay(500);
  Serial.print(".");
  delay(500);
  Serial.print(".");
  delay(500);
  Serial.print(". ");
  delay(5000); // wait because of rocking
  _state = digitalRead(tiltPin); //check if upside down
}
Serial.println("Correct!");
}
```

### 3.9.4 Cyklické volanie kódu

Cyklická časť kódu sa delí na 4 časti: načítanie vstupov, komunikácia s PC, generovanie hodnôt pre pohyb, vysielanie dát. Cyklická časť je funkcia s názvom `loop()` (cyklus) vykonávaná po funkcii `setup()` (príprava). Kód sa vykonáva opakovane, tak rýchlo ako je možné. Limitom je výkon čipu a zložitosť inštrukcií kódu. Väčšinou kód obsahuje dodatočné oneskorenia, funkcia `delay()`, ktorá pozastaví vykonávanie kódu na zvolený počet milisekúnd.

```
void loop() {  
  
  // sensors  
  tilt_switch(10); // hysteresis in number of cycles  
  text = Sonic_distance();  
}
```

Pokiaľ je robot pripojený, skontroluje prichádzajúce dáta. Ak je robot odpojený od počítača, celá časť kódu sa preskočí. Na začiatku sa načíta vstup ako string dát. Všetky znaky sa prevedú na malé písmená, takže napríklad príkaz „ON“, „On“ alebo „on“ sú platné. V tabuľke (tab.1) je zoznam príkazov a ich popis. V prípade neplatného príkazu sa vypíše chybové hlásenie.

```
// communication  
if (Serial.available()) { // number of bytes (characters) from serial  
port  
  delay(100);  
  _string = Serial.readString(); // read incoming string  
  _string.toLowerCase(); // make it lowercase  
  
  if (_string.startsWith("off"))  
  ...  
  else if (_string.startsWith("on"))  
  else Serial.println("Unknown command");  
};
```

Viac o chybovom výpise v podkapitole 3.4.2 Komunikácia a riadenie.



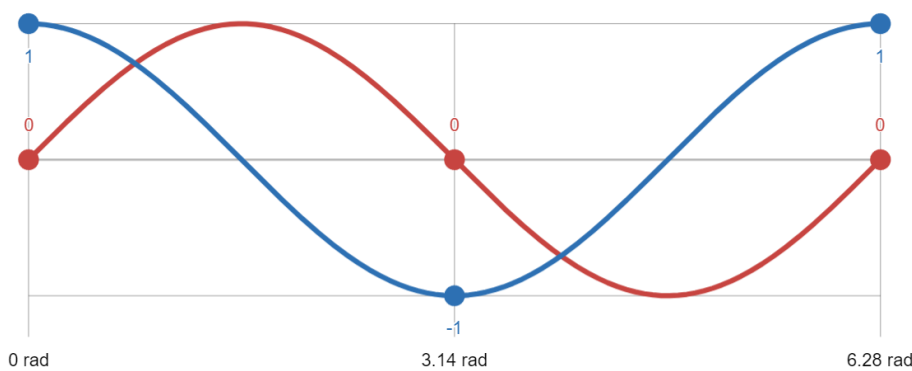
### 3.9.5 Generátor hodnôt vlnenia

Program týkajúci sa generátora hodnôt vlnenia:

Nastavenie `debug = true` ukazuje výstup hodnôt generátora v reálnom čase a dáta posielané aktuátorom. `display_pos` vypisuje aktuálnu pozíciu jedného aktuátora, ID aktuátora je zvolené na ďalšom riadku.

Premená `activate = true` povolí cyklické aktualizácie generátora a posielanie dát pre pohyb

Odstavec „movement parameters“ zahŕňa všetko potrebné premenné pre ovládanie a generovanie hodnôt generátora. Generátor používa goniometrické funkcie (vložiť funkcie) pre generovanie vlnenia. Pôvodný generátor používal funkciu sínus, ale pri testovaní sa ukázalo že je lepšie použiť funkciu kosínus. Na obrázku obr.(52) je ukážka výstupných hodnôt funkcií sínus a kosínus navzorkovaných na 8 vzoriek periódy obr.(53). Pokiaľ je počet vzoriek vysoký, funkcia sínus a kosínus vracia rovnaké hodnoty, posunuté o  $90^\circ$ .

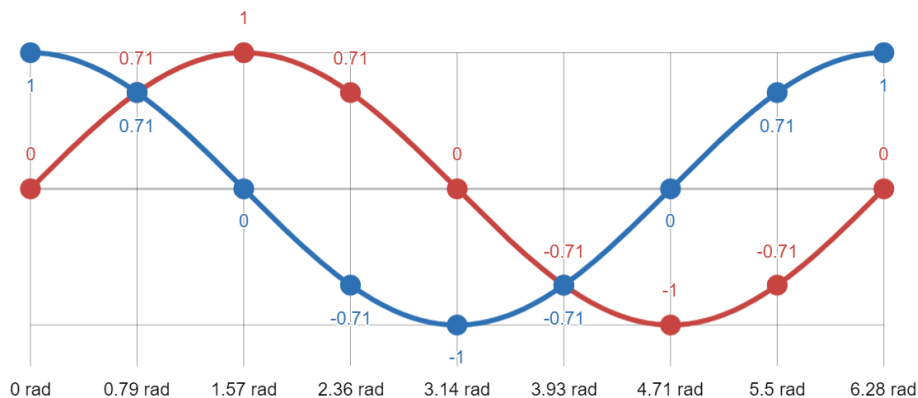


Obr. 52: Dve vzorky na periódu

k=2	n [rad]	sin(n)	cos(n)
1	0,00	+0,00	+1,00
2	3,14	-0,00	-1,00

Tab. 1 Výsledné hodnoty pri vzorkovaní k = 2

Fázový posun zvýraznený identickým výstupom, posunutým o dva vzorky z ôsmych, čo sa rovná posuvu  $90^\circ$  ( $\sim 1,57$  rad). Posun nespôsobuje žiadny problém pri vykonávaní pohybu. Problém nastane pri ak je  $k = 2$  obr.(52). Pri vzorkovaní funkcie sínus sú návratové hodnoty nuly, zatiaľ čo pri vzorkovaní kosínusu dostaneme hodnoty  $+1$  a  $-1$ . S týmito hodnotami je možné ďalej pracovať.



Obr. 53: Osem vzoriek na periódu

k=8	n [rad]	sin(n)	cos(n)
1	0,00	+0,00	+1,00
2	0,79	+0,71	+0,71
3	1,57	+1,00	+0,00
4	2,35	+0,71	-0,71
5	3,14	+0,00	-1,00
6	3,93	-0,71	-0,71
7	4,71	-1,00	+0,00
8	5,50	-0,71	+0,71

Tab. 2 Výsledné hodnoty pri vzorkovaní  $k = 8$ 

Odkaz pre interaktívny nástroj, s ktorého pomocou boli vytvorené obrázky (obr.52 a obr.53) a tabuľky (tab.1 a tab.2) je v prílohe.

### 3.9.6 Hlavné pohyby:

Hlavným pohybom je pohyb generovaný podľa rovnice vlnenia. Funkcia *side\_move()* posunie hodnoty v registri o dve miesta, vypočíta hodnoty podľa parametrov generátora a vloží ich na prvé dve miesta do registra. Funkcia *Dx\_move\_all()* pošle hodnoty registra príslušným aktuátorom. V prípade detekcie prekážky, sa parametra generátora upraví tak, aby sa robotický had dokázal prekážke vyhnúť. Po určitom čase sa parametre navrátia do predvolených hodnôt. Hlavný pohyb a celé generovanie môže byť vypnuté užívateľom pre použitie pomocných pohybov. Detail o funkcii a parametroch generátora pohybu v kapitole 3.8 Matematický popis.

```

if (activate) {
    //////////////// ACTUATIONS
    side_move(); // shift register and generate new position
    Dx_move_all(); // move all joints
} else
{}; // skipping whole generation above
}

```

### 3.9.7 Pomocné pohyby:

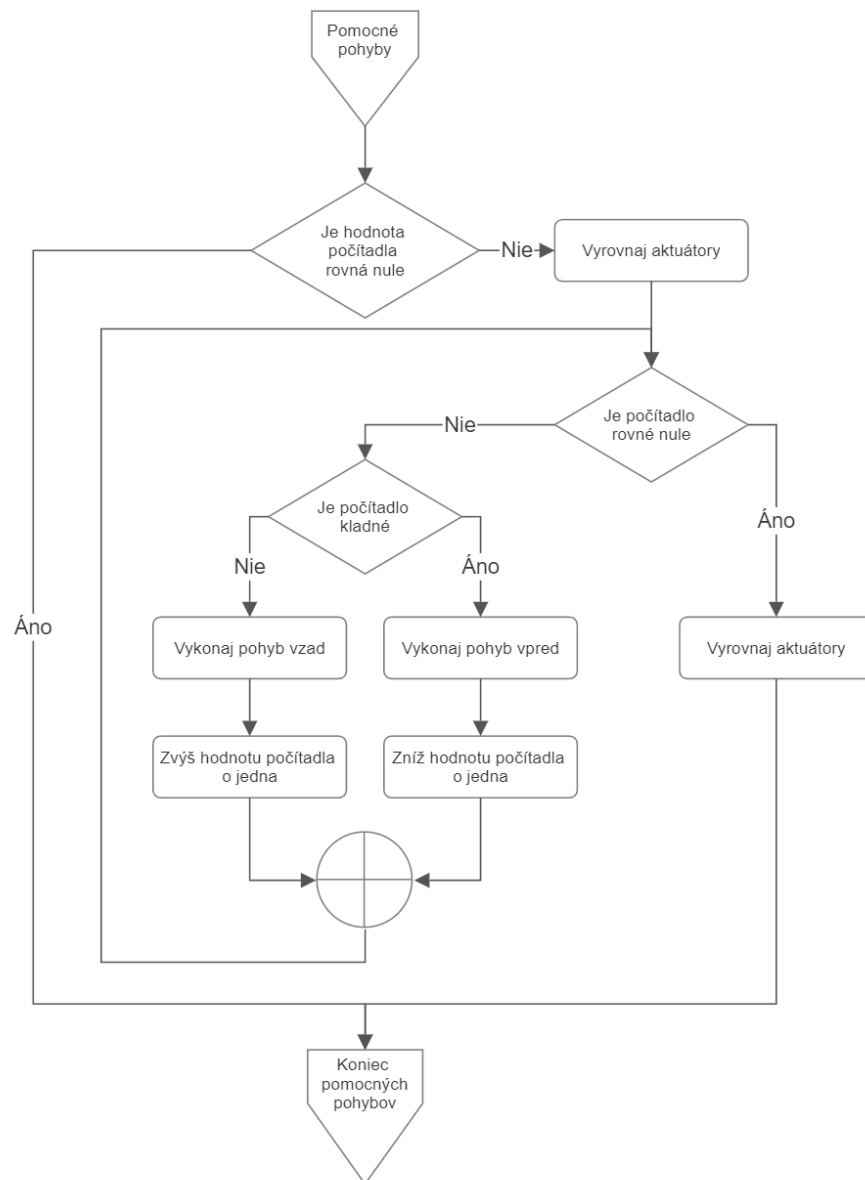
Každý pomocný pohyb je v kóde reprezentovaný ako sekvencia príkazov. Každý typ pohybu má pridelené počítadlo, s rovnakým názvom. Hodnota môže byť kladná aj záporná celé číslo. Pre spustenie pohybu roll/turn/worm je potreba príkaz roll/turn/worm s číslom počtu opakovaní, napr.: „worm 5“. Negatívne hodnoty vykonajú pohyb v opačnom smere. Pohyb sa vykonáva až kým počítadlo nedosiahne hodnoty 0. Pre predčasné zastavenie pohybu slúži príkaz „roll 0“ alebo „roll“. Pokiaľ nie je definovaný počet opakovaní, automaticky je dosadená hodnota 0, ktorá zastaví vykonávanie daného pohybu. Časť kódu obsahujúca sekvenciu natočení bola nahradená za „...“ pre čitateľnosť, pôvodný kód je v prílohe práce.

```

if (roll != 0) { // outside while, inside if: Execute once at the start
    Dx_center(); // straighten all joints
    while (roll != 0) //execute until all steps done
    {
        if (roll > 0) // if positive move forward/right
        {
            --roll; // subtract from counter each iteration
            ...
        }
        if (roll < 0) // if negative move backward/left
        {
            ++roll; // add to counter each iteration
            ...
        }
    }
    // outside while, inside if: Execute once at the end
    Dx_center(); // straighten all joints
}
}

```

Pohyb „turn“ a „worm“ majú podobnú štruktúru kódu ako pohyb „roll“, príkazy sú obmenou príkazu „roll“.



Obr. 54: Vývojový diagram vykonávania pomocného pohybu

### 3.9.8 Funkcie deklarované na konci dokumentu

Vlastné funkcie na konci dokumentu: nie je ich potreba volať pred cyklickým chodom programu, preto ich je možné vložiť na koniec dokumentu.

Funkcia `isValidNumber()` testuje či je vstup platná číselná hodnota. Pokiaľ by vstup obsahoval znak iný ako sú cifry, vráti hodnotu `false` a vypíše chybovú hlášku pre užívateľa.

```
// functions
boolean isValidNumber(String str) { //return true if only one number
    String _str = str;
    if (_str.startsWith("-"))
    {
        _str.remove(0, 1); // remove from start
    }
    for (byte i = 0; i < _str.length(); i++)
    {
        if ((not(_str.indexOf(" ") == -1)) or
        (not(isDigit(_str.charAt(i)))))) {
            Serial.print(_str);
            Serial.println(" <--- is not a valid input!");
            return false;
        }
    }
    return true;
}
```

Funkcie spracovávajúce vstupy zo senzorov. Ultra-zvukový senzor vysiela a prijíma odraz, arduino merá čas medzi pulzami a vypočíta vzdialenosť, ktorá je návratovou hodnotou funkcie. Senzor náklonu potrebuje filter nízkych frekvencií. Pokiaľ sa výstup zo senzora nezmenil na viac cyklov ako parameter `_delay` udáva, počítadlo cyklov sa resetuje a nevykoná sa žiadna akcia. Touto funkciou sa potlačuje vplyv vibrácií.

```
int Sonic_distance() {
    // generate 10-microsecond pulse to TRIG pin
    digitalWrite(trigPin, HIGH);
    delayMicroseconds(10);
    digitalWrite(trigPin, LOW);
    int dist = 0.017 * pulseIn(echoPin, HIGH);
    // measure duration, calculate distance
    return dist;
}

void tilt_switch(int _delay) {
    _state_new = digitalRead(tiltPin); //update new
    if (_state xor _state_new) // if change
        _time++;
    else
        _time = 0;
    if (_time > _delay)
    {
        _state = !_state; //toggle
        _time = 0;
    }
    return _state;
}
```





## 4 ZHODNOTENIE A DISKUSIA

### 4.1.1 Testovanie

Otestované boli 4 pohyby: bočný pohyb, otočka na mieste, kotúľanie sa do strany [27] a húsenicový pohyb. Posledný typ pohybu bol otestovaný po naklonenej rovine [19]. Uhol roviny bolo možné meniť zdvižným vozíkom. Na pevnú dosku bol pripevnený plát kartónu, na ktorom boli prilepené značky s rozstupom 10 cm. Pohyb bol nahrávaný na mobilný telefón, ktorý slúžil aj na odmeranie sklonu podložky. Vizualným porovnaním polohy robota s časom záznamu bolo možné vypočítať rýchlosť robota. Namerané hodnoty boli spriemerované a zobrazené do grafu. Uhol bol postupne zvyšovaný. Robot spomaľoval postup zo zvyšujúcim sa sklonom roviny a pri  $\sim 23^\circ$  boli straty vzniknuté prešmykovaním približne podobné pohybu vpred.



Obr. 55: Testovacia zostava s nastaviteľným sklonom



Obr. 56: Značky na testovacej rovine, rozteč 10 cm

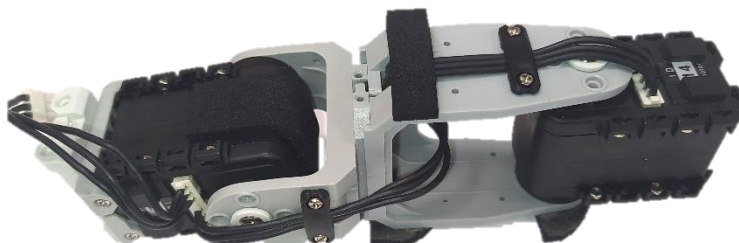
#### 4.1.2 Výsledná konstrukcia

Prvotne robot používal kolieska pre uľahčenie pohybu po rovnom povrchu. Pri experimentoch sa ukázalo, že kolieska majú malý priemer. Väčší priemer koliesok by zmenšil rozsah pohybu, pretože články by pri vychýlení tlačili na kolieska. Zmena ich polohy nie je bez zložitejšej zmeny do konštrukcie možná. Robot je konštruovaný pre zložitejšie terény, a kolieska pomáhajú len pri základnom pohybe použiteľným na rovnom povrchu.



Obr. 57: Články s kolieskami

Pri ostatných typoch pohybu robota boli prekážkou a preto boli nahradené za penové podložky. Penové podložky sú nalepené obojstrannou páskou na dielcoch, ktoré najčastejšie prichádzajú do kontaktu s povrchom. Pena pohlcuje nárazy, čím chráni rám, zakrýva káble pred poškodením a zvyšuje trenie čo pomáha pri pohybe.



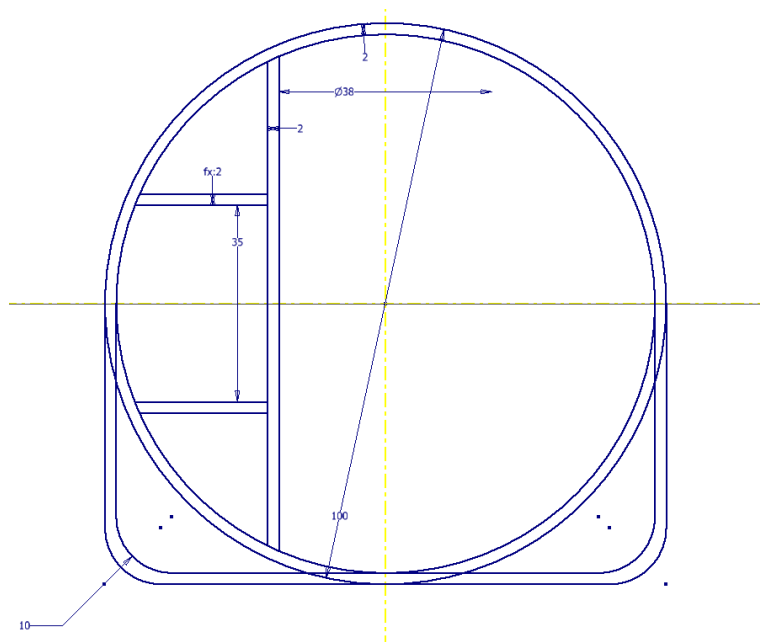
Obr. 58: Články bez koliesok

Poslednou zmenou bolo pridanie prstencov na rám robota. Prstence zlepšili pohyblivosť terénom a poskytujú lepšiu ochranu citlivým častiam robota. Boli vyrobené na 3D tlačiarňi a pri ich poškodení je možné vyrobiť náhradu za pár desiatok minút.

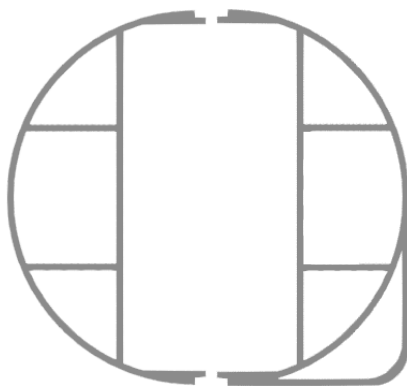


Obr. 59: Články s prstencami

Jeden náčrt modelu bol použitý pre výrobu prstencov pre články hada (obr. 60). Prstence majú dve variácie: Kruhový prstenec „O“ a prstenec v tvare písmena „D“. Rozdiel v 3D modeli je vo vysunutí rohových častí z náčrtu. Prstence sú vertikálne symetrické, takže je možné v prípade poškodenia nahradiť prstenec bez nutnosti riešenia polohy vpravo/vľavo. Na koncoch prstencov sa nachádzajú zuby, ktoré zapadajú do protíahlého prstenca a udržia hladký prechod. Je možné spojiť dva rôzne prstence, jeden kruhový, druhý prstenec s tvarom „D“ (obr. 61). Výsledný tvar nemal využitie a preto nebol použitý.



Obr. 60: Rozmery modelu prstencov z náčrtu programu Inventor



Obr. 61: Rozdiel tvaru prstencov „O“ a „D“ v spodnej časti

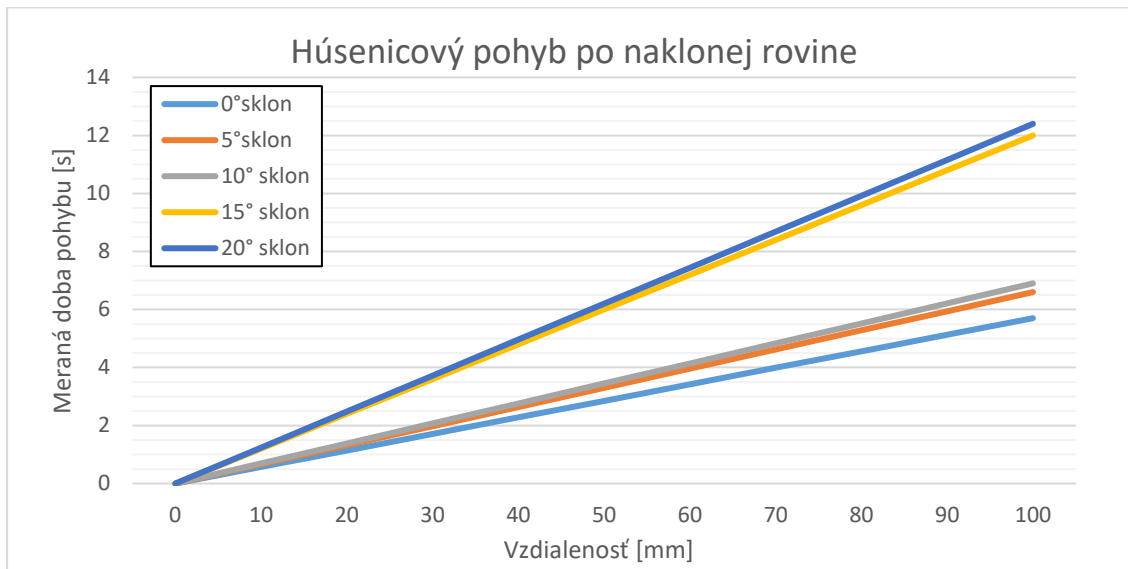
Prvý prstenec hada, na článku s hlavou, je kruhového tvaru, druhý prstenec ma tvar písmena „D“. Prstence sa takto striedajú pozdĺž celej dĺžky robotického hada. Samotné prstence kruhovitého prierezu sa ukázali ako príliš nestabilné pri úplnom vyrovnaní hada. Už jeden článok s tvarom „D“ pomohol zo stabilitou, ale pri pohybe je vždy jeden článok oddelený od podkladu. Preto boli nahradené dva prstence, čím sa zaručí že aspoň jeden článok je v kontakte s podkladom. Pena bola prilepená aj na spodnú časť na miestach kde dochádza ku kontaktu s podkladom pre zvýšenie trenia s povrchom. Plochá časť článkov má penovú podložku po celej dĺžke, zatiaľ čo okrúhle články majú penovú podložku len v strede.



Obr. 62: Penové podložky nalepené na spodnej strane prstencov

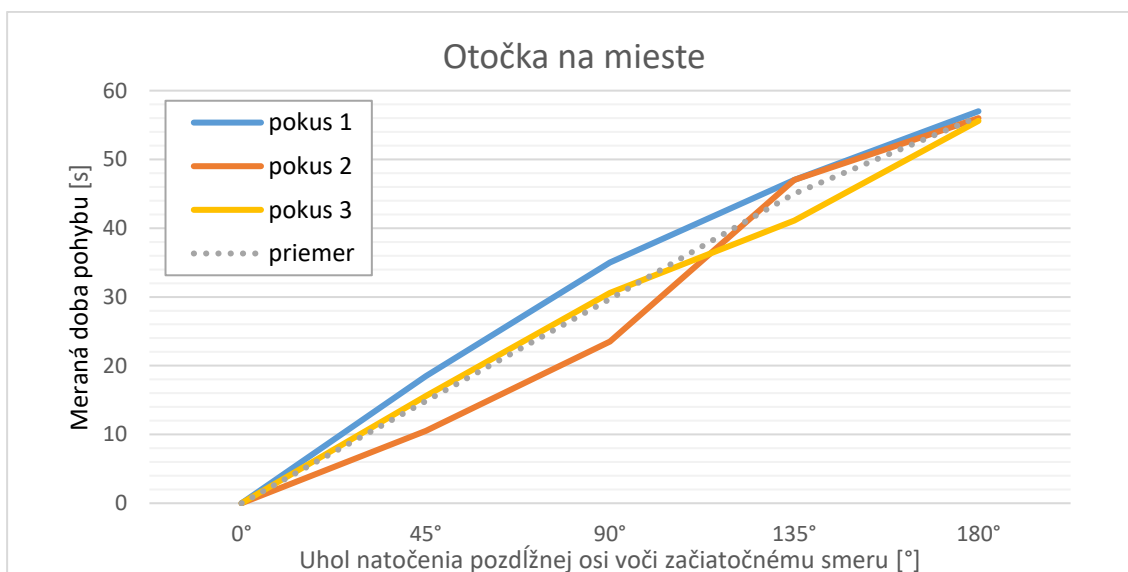
### 4.1.3 Výsledky experimentov

Efektívnosť pohybov bola porovnaná pri šplhaní po naklonenej rovine (Obr.63). Húsenicový pohyb je najstabilnejší, ale priechodnosť terénom je nízka. Záver sa dal predpokladať, pretože vertikálny pohyb je obmedzený na minimum, čím je ťažisko držané nízko nad povrchom. Zlá priechodnosť sa ukázala, keď robot nedokázal prejsť cez hranu medzi podlahou a naklonenou rovinou vytvorenou skokovou zmenou náklonu. Pri testovaní sklonu preto začínal na naklonenej rovine. Čas prejdenej vzdialenosti bol odčítaný z video záznamu pohybu. Had bol položený na prvú značku a zo snímku, na ktorom hlava pretla značku bol odčítaný čas záznamu (Obr. 56).



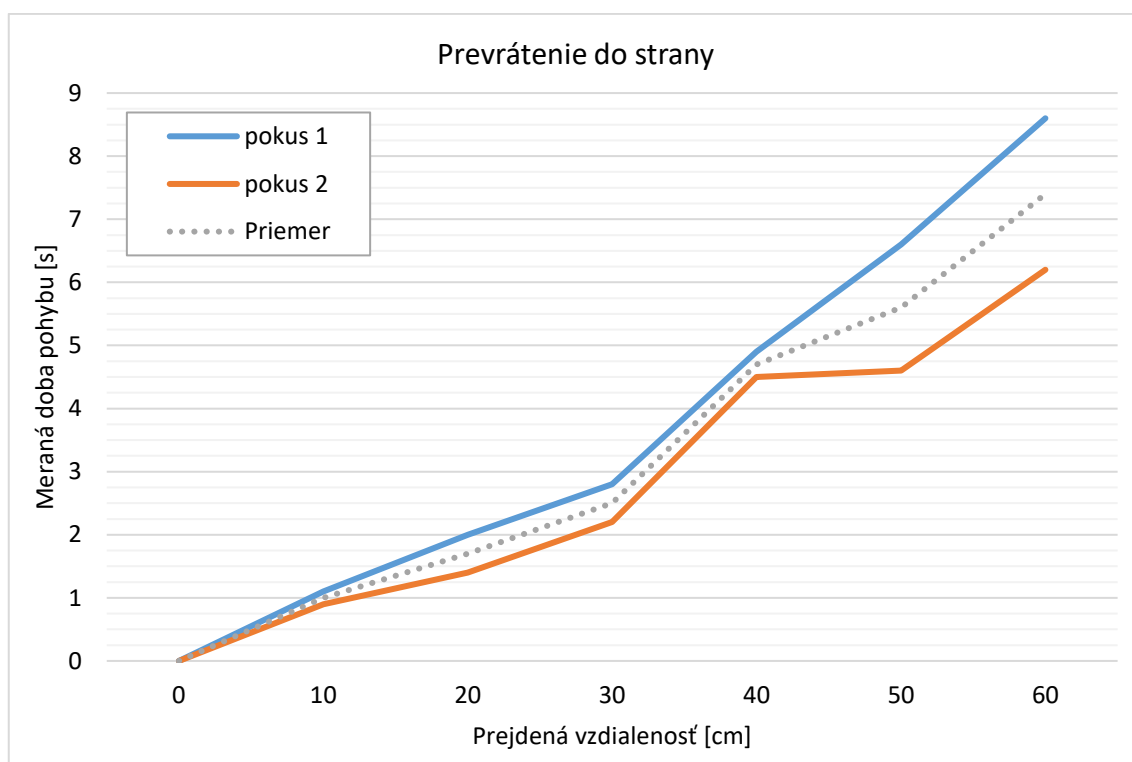
Obr. 63: Graf závislosti rýchlosti na sklone roviny

Správnou sekvenciou je robotický had schopný pohybom koncovým článkom otáčať sa na mieste. Pribeh bol vykonaný na nenaklonenej rovine.



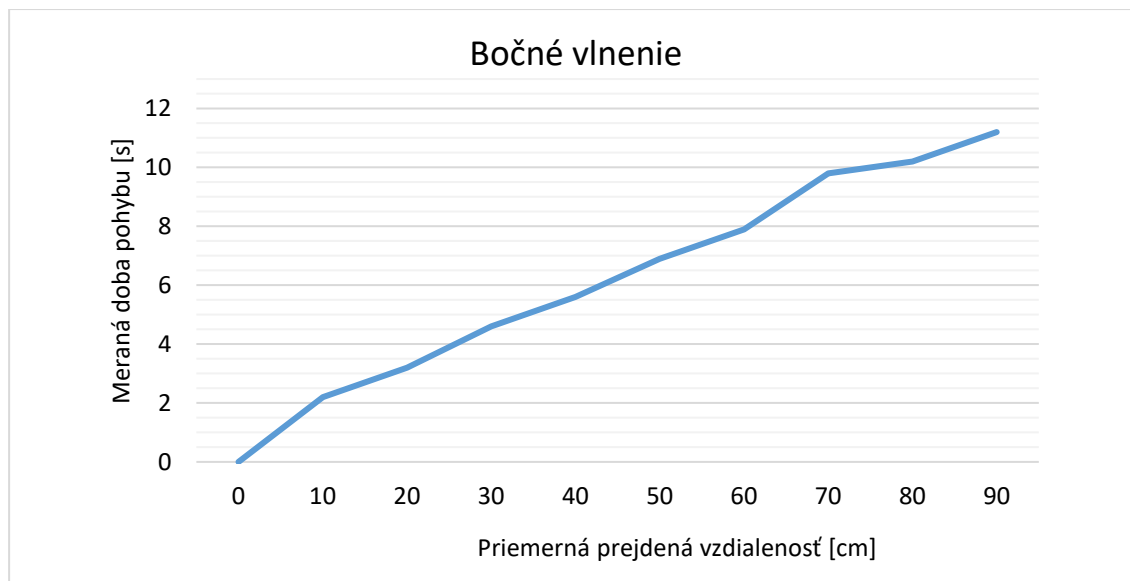
Obr. 64: Graf uhlovej rýchlosti počas pribehu otáčania sa o 180°

Prevrátenie robotického hada je väčšinou nežiadúce. Napriek tomu, že pohyb nie je inšpirovaný biologickými hadmi, vďaka konštrukcií kĺbov je možné použiť tento pohyb pre odvaľovanie hada do strán. Pohyb do strán zachováva pozdĺžnu os paralelnú voči počiatočnému natočeniu, čo je možné použiť pre vyhnutie prekážok. Pohyb je vykonaný postupom príkazov a je zahrnutý do pomocných pohybov ktoré je had schopný vykonať. Tento pohyb je vhodný napríklad pre odmotanie skrúteného napájacieho kábla. Priebeh s okrúhlymi „O“ prstencami ma za výsledok plynulé odvaľovanie, ale absencia prstencov „D“ spôsobuje nestabilitu pre ostatných pohyboch. Pri použití oboch typov prstencov dochádza ku menej plynulému pohybu, pretože pri každom otočení je nutné prejsť cez dva trčiace rohy prstencov „D“. Tento problém sa odzrkadľuje aj na rýchlosti pohybu (Obr. 65). Pokusy 1 a 2 boli vykonané v opačných smeroch odvaľovania.



Obr. 65: Rýchlosť pri prevracaní do strany

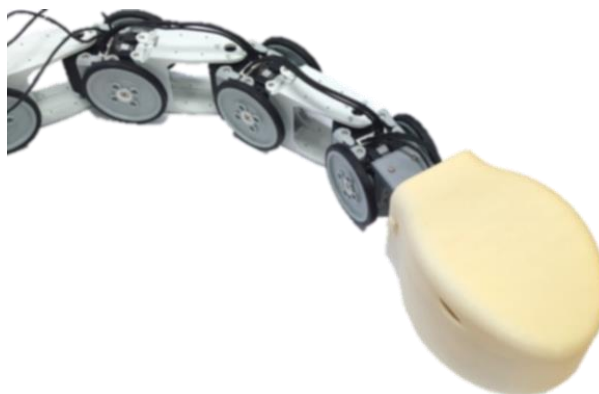
Bočný pohyb je menej stabilný z dôvodu vyššieho dvíhania článkov. Priechodnosť terénom je zlepšená, robot dokáže bez problémov prejsť po ~5 cm schode a prekonávať malé nerovnosti. Robot dokázal prejsť rozhranie medzi podlahou a naklonenou rovinou napriek skokovej zmene náklonu. V niekoľkých prípadoch došlo k prevráteniu robota z dôvodu nestability. Spomalením pohybov a zvýšením amplitúdy horizontálneho pohybu sa podarilo zlepšiť šplhanie. Väčšia základňa a pomalé pohyby znížili rýchlosť pohybu, napriek tomu je pohyb rýchlejší ako húsenicový. Maximálny sklon, po ktorom sa had dokázal plaziť zostal nezmenený, čo naznačuje, že limitom je koeficient trenia medzi robotom a povrchom. Hlavnou výhodou pohybu bola najvyššia rýchlosť v porovnaní s ostatnými pohybmi robotického hada.



Obr. 66: Graf priemernej rýchlosti pri pohybe do strany



Robotický had schopný planárneho pohybu (Obr.67) bol použitý pre otestovanie pohybu laterálnej undulácie. Jeho konštrukcia povoľuje len použitie jedného typu pohybu a má kolieska, ktoré pomáhajú pri pohybe na rovnom povrchu. Nemožnosť ovládania v tretej ose a nízka svetlá výška obmedzujú použitie v teréne. Z týchto dôvodov bol použitý len pre ukážku schopnosti kódu fungovať aj na obecných robotoch používajúcich aktuátory série AX. Robot používa aktuátory AX-18, ktoré sa odlišujú od AX-12 vyšším výkonom.



Obr. 67: Detail planárneho robotického hada

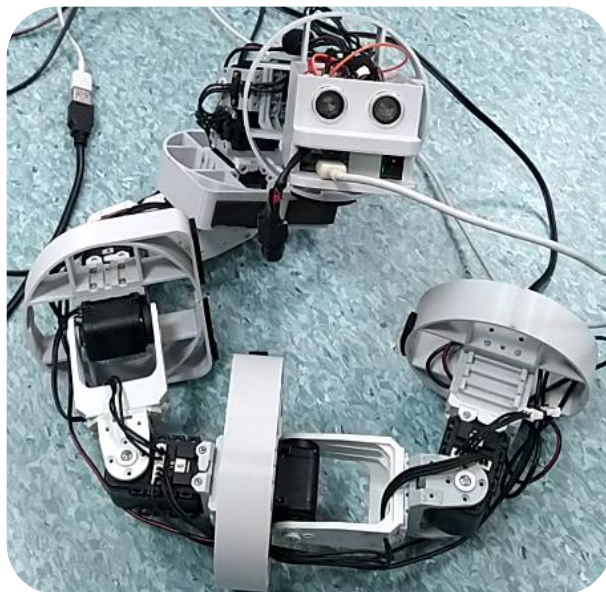
#### 4.1.4 Ďalší vývoj

Pri použití vode-odolných aktuátorov je možné, aby robotický had plával na hladine, alebo sa potápal do hĺbok. Pohyby pod vodou sú podobné pohybom po súši a bude možné použiť existujúce funkcie. Bezdrôtová komunikácia pod vodou je zložitá, preto je odporúčané použitie aktuálneho riešenia, keď had za sebou ťahá kábel pre napájanie aj komunikáciu. V prípade problému bude možné robota vytiahnuť z vody za kábel. Pre navigáciu pod vodou bude nutné použiť kameru alebo sonar. Použitie kamery ju možné len vo veľmi čistej vode. Sonar vytvára odrazy ktoré je nutné filtrovať aby nedošlo k strate orientácie.

## 5 ZÁVER

Funkčnosť kódu bola overená na dvoch rôznych modeloch robotického hada.

Robotický had (Obr.66) je schopný pohybu v 3D priestore a z experimentov bolo dokázané, že je schopný skokového prechodu z roviny podlahy na rovinu pod sklonom 20°. Robot je ovládaný pomocou programu Arduino IDE, príkazy možno písať do okna monitoru sériového portu v reálnom čase. Robot je schopný aj pohybu podľa nastavení aj bez pripojenia káblom. Robot je schopný vykonať úkony na základe vstupu zo senzorov. Pre detekovanie prekážok a kontrolu orientácie používa senzory umiestnené v prvok článku, ukryté v hlave vyrobenej na 3D tlačiarňami.



Obr. 68: Robotický had v polohe predvzajajúca pohyblivosť

Táto práca sa zaoberala riadením hada a pohybu po hladkých povrchoch, prechodu medzi povrchmi a prekonávaní malých prekážok, šplhaniu po naklonenej rovine, autonómnemu vyhýbaniu prekážok a automatickej detekcie vyvrátenia a pohybu pre prevrátenie naspäť. Bol zostavený jeden model, ktorý prešiel množstvom zmien týkajúcich sa konštrukcie, elektroniky, použitého programu a riadenia. Konštrukcia prešla niekoľkými zmenami za účelom nájdenia kompromisu medzi priechodnosťou a tvarom, ktorý by neobmedzoval mobilitu robota. Robot obsahuje dva typy senzorov. Preklápací spínač pre detekciu vyvrátenia a ultra-zvukový senzor pre meranie vzdialenosti od prekážky.

Bolo využitých niekoľko programov spoločnosti Dynamixel, program MATLAB a programovací jazyk Wiring v prostredí ArduinoIDE.

Detailne preskúmanie a optimalizácia množstva parametrov a kombinácii pohybu nie je časovo možná. Pri experimentovaní s parametrami boli nájdené limity robota a nastavenia, ktoré vykonávali určitý druh pohybu. Každý pohyb má špecifické hodnoty a limity v ktorých je možné nastavené parametre meniť tak, aby nedošlo ku problémom akou je kolízia samotných článkov.

Celkový počet článkov je 8, ale kmitavý pohyb je v dvoch rovinách. Na každú rovinu teda ostávajú 4 články. Niektoré pohyby vyžadujú väčší počet článkoch a nemohli byť z tohto dôvodu realizované. Príliš veľký počet článkov by ale negatívne ovplyvnil mobilitu a značne zvýšil požiadavky na napájanie. Väčší počet článkov by zvýšil aj hmotnosť a dĺžku hada. V aktuálnom prevedení je možné celého robota vrátane napájania a potrebnej kabeláže zbalit' to školského batohu. Vyššia hmotnosť by spôsobovala vyššie sily a tým by sa zvýšil aj odber. Pre stabilný chod je nutný tvrdý zdroj napájania. Zahrievanie a teplota je ďalším problémom pri predĺžení. Chladenie je možné, ale zvyšuje hmotnosť a zložitosť dizajnu článkov, čím znižuje priechodnosť a vytvára miesto náchylne na poškodenie. Predĺženie robota pridaním článkov nemení výpočtovú náročnosť a pamäťové nároky by sa zvýšili minimálne.

Najväčším problémom ostáva napájanie. Jedna z možností je použiť dodatočných kondenzátorov. Ich kapacita pomáha vyhladzovať prudké nárazy prúdu, spôsobených rýchlymi pohybmi. Toto riešenie môže byť nedostatočné ak je vyžadovaný trvalý odber prúdu, napríklad pri šplhaní. V tomto prípade pomôže výkonnejší zdroj alebo silnejšia batéria.

Limitom pri zdolávaní prekážok je hlavne koeficient trenia medzi povrchom a robotom. Jeho nedostatok na klzkých povrchoch môže mať za následok uviaznutie robota aj pri najmenších prekážkach. Veľmi vysoký koeficient trenia zbytočne namáha aktuátory a plytvá krútiacim momentom. To môže mať za následok prehrievanie aktuátorov a vo výsledku zníženie priechodnosti terénom.



## 6 ZOZNAM POUŽITEJ LITERATURY

- [1] FOURNIER, Mat. *Příroda: nekonečná inspirace vědy : historie technických vynálezů, k nimž nás přivedlo zkoumání živých organismů*. Čestlice: Rebo, 2013. ISBN 978-80-255-0736-0.B
- [2] HIROSE, Shigeo, Peter CAVE a Charles GOULDEN. *Biologically Inspired Robots: Serpentine Locomotors and Manipulators*. USA: Oxford University Press, 1993. ISBN 978-0-19-856261-0.
- [3] POLLÁK, Jakub. *Návrh modelu robotického hada*. Brno, 2021. Dostupné také z: <https://www.vut.cz/studenti/zav-prace/detail/132853>. Bakalářská práce. Vysoké učení technické v Brně, Fakulta strojního inženýrství, Ústav automatizace a informatiky. Vedoucí práce Tomáš Hůlka.
- [4] Snake worx avoidance *Mojave desert Sidewinder* [online]. Dostupné z: <https://snakeworxavoidance.com/new-gallery-2>
- [5] HPC wire. *mystery flying snakes* [online]. Dostupné z: <https://www.hpcwire.com/2013/10/31/gpus-mystery-flying-snakes/>
- [6] Moje hady. *Moje hady* [online]. Dostupné z: <https://mojehady.webnode.sk/news/anatomia-hada/>
- [7] How it works. *How it works: All about snakes* [online]. 25.10.2018 [cit. 2023-05-25]. Dostupné z: <https://www.howitworksdaily.com/all-about-snakes/>
- [8] Hirose, S.; Yamada, H.: *Machine Design of Biologically Inspired Robots*. 2009.
- [9] TATSUYA, Takemori, Tanak MOTOYASU a Matsuno FUMITOSHI. *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS): Ladder Climbing with a Snake Robot*. 1-9. Dostupné z: doi:10.1109/IROS.2018.8594411
- [10] JSK: *Multi-Link Aerial Robot* [online]. [cit. 2023-05-25]. Dostupné z: <http://www.jsk.t.u-tokyo.ac.jp/research/multilink.html>
- [11] EPFL: *Salamandra robotica* [online]. 2013 [cit. 2023-05-25]. Dostupné: [https://www.epfl.ch/labs/biorob/research/amphibious/salamandra/?utm\\_source=robots.ieee.org](https://www.epfl.ch/labs/biorob/research/amphibious/salamandra/?utm_source=robots.ieee.org)
- [12] *Robotis e-manual: DYNAMIXEL Shield for Arduino MKR Series* [online]. 2023 [cit. 2023-05-25]. Dostupné z: [https://emanual.robotis.com/docs/en/parts/interface/mkr\\_shield/](https://emanual.robotis.com/docs/en/parts/interface/mkr_shield/)
- [13] *ROBOTIS: bioloid frames* [online]. 2023 [cit. 2023-05-25]. Dostupné z: <https://www.robotis.us/dynamixel-ax-bioloid-frames/>
- [14] J. J. Marín-Arciniegas, O. A. Vivas-Albán, "Design and Construction of a Snake-Like Robot Implementing Rectilinear and Sidewinding Gait Motions," *Tecnológicas*, vol. 26, nro. 56, e2412, 2023. [cit. 2023-09-20]. Dostupné z: <https://doi.org/10.22430/22565337.2412>
- [15] WORST, RAINER, and RALF LINNEMANN. "Construction and operation of a snake-like robot." *Proceedings IEEE International Joint Symposia on Intelligence and Systems*. IEEE, 1996.
- [16] HOPKINS, JAMES K., BRENT W. SPRANKLIN, and SATYANDRA K. GUPTA. "A survey of snake-inspired robot designs." *Bioinspiration & biomimetics* 4.2 (2009): 021001

- [17] Mouser Arduino Leonardo [online]. [cit. 2023-09-20]. Dostupné z: <https://cz.mouser.com/new/arduino/arduino-leonardo>
- [18] The Amazing Sidewinder. ZACKANDSCOTTKARMACHAMELEONS [online]. 4. Marec 2016 [cit. 2023-09-25]. Dostupné z: <https://zackandscottkarmachameleons.wordpress.com/2016/03/04/the-amazing-sidewinder/>.
- [19] Lim, J.; Yang, W.; Shen, Y.: *Analysis and Validation of Serpentine Locomotion Dynamics of a Wheeled Snake Robot Moving on Varied Sloped Environments*. In *IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM 2020)*, May 2020
- [20] Maity, A.; Majumder, S.: *Serpentine robots: A study of design philosophy*. *IEEE 15th International Conference on Advanced Robotics: New Boundaries for Robotics, ICAR 2011*, 06 2011, doi:10.1109/ICAR.2011.6088572
- [21] PERRY, Lacy. *How Snakes Work*. HowStuffWorks [online]. [cit. 2023-09-25]. Dostupné z: <https://animals.howstuffworks.com/snakes/snake.htm>
- [22] CARTER, Lou. *How Do Snakes Move? (4 Snake Movement Types Explained)*. *Snakes for Pets* [online]. 15. december 2020 [cit. 2023-09-25]. Dostupné z: <https://www.snakesforpets.com/how-do-snakes-move/>
- [23] REZAPOUR, Ehsan. (2015). *Model-based Locomotion Control of Underactuated Snake Robots*.
- [24] ZHAO M. et al., "Flight Motion of Passing Through Small Opening by DRAGON: Transformable Multilinked Aerial Robot," 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Madrid, Spain, 2018, pp. 4735-4742, doi: 10.1109/IROS.2018.8593368.
- [25] MORI, M.; Yamada, H.; Hirose, S.: *Design and Development of Active Cord Mechanism "ACM-R3" and its 3-dimensional Locomotion Control*. *Journal of the Robotics Society of Japan*, ročník 23, 2005: s. 886–897.
- [26] FU Qiyuan and Li Chen 2020 *Robotic modelling of snake traversing large, smooth obstacles reveals stability benefits of body compliance*. *Soc. open sci.* 7191192191192 <http://doi.org/10.1098/rsos.191192>
- [27] MORI, M.; Hirose, S.: *Three-dimensional serpentine motion and lateral rolling by active cord mechanism ACM-R3*. *IEEE/RSJ International Conference on Intelligent Robots and Systems*, ročník 1, 2002: s. 829–834.
- [28] CRESPI, Alessandro & Badertscher, André & Guignard, Andre & Ijspeert, A.J.. (2005). *Swimming and Crawling with an Amphibious Snake Robot*. *Proceedings - IEEE International Conference on Robotics and Automation*. 2005. 3024 - 3028. 10.1109/ROBOT.2005.1570574.

## 7 ZOZNAM OBRÁZKOV A TABULIEK

- Obr. 1: HiBot ACM-R5H
- Obr. 2: Štrkáč rohatý
- Obr. 3: Lietajúci had
- Obr. 4: Anatómia hada
- Obr. 5: Laterálna undulácia
- Obr. 6: Bočné vlnenie
- Obr. 7: Harmonikový pohyb
- Obr. 8: Priamočiary pohyb
- Obr. 9: Posúvače vstupných parametrov v programe Desmos
- Obr. 10: Vplyv parametra „a“ na tvar krivky
- Obr. 11: Vplyv parametra „f“ na tvar krivky
- Obr. 12: Robot ACM-III [8]
- Obr. 13: Štruktúra článku robota ACM-III [8]
- Obr. 14: Rozdiel polohy a uloženia kolies medzi robotmi ACM-III (a) a ACM-R3 (b) [8]
- Obr. 15: Robot ACM-R3 [8]
- Obr. 16: Ukážka lezenia po rebríku [9]
- Obr. 17: Vizualizácia vstupných dát [9]
- Obr. 18: DRAGON [10]
- Obr. 19: Konfigurácia pri uzatváraní ventilu [10]
- Obr. 20: Vode-odolný robot ACM-R5 [8]
- Obr. 21: Robot ACM-R5H schopný splývania s hladinou vody [11]
- Obr. 22: Robotický had Genbu s poháňanými kolesami [8]
- Obr. 23: Komerčne dostupná verzia robota Soryu-C [8]
- Obr. 24: Prevodovka aktuátora AX-12
- Obr. 25: Poškodený kábel a príčina vzniku
- Obr. 26: Aktuátor Dynamixel AX-12 [12]
- Obr. 27: Pracovný rozsah aktuátora Dynamixel AX-12 [12]
- Obr. 28: Spätná väzba aktuátora Dynamixel AX-12 [12]
- Obr. 29: Vysoký parameter proporcionálnej zložky
- Obr. 30: Nízky parameter proporcionálnej zložky
- Obr. 31: Zapojenie typu „Daisy chaining“ [12]
- Obr. 32: Zvýraznenie tlačidla, ktoré otvorí okno „monitor sériového portu“
- Obr. 33: Okno monitora sériového portu, dôležité nastavenia vpravo dole
- Obr. 34: Dielce rámu [13]
- Obr. 35: Kinematická štruktúra [14]

- Obr. 36: Rozložený kardanový spoj (dve vidlice a kríž v strede)
- Obr. 37: Zložený a natočený kardanový spoj
- Obr. 38: Rozložený článok robota
- Obr. 39: Stredová časť článku, dva aktuátory spojené s 90° otočením,
- Obr. 40: tvoriace stredový kríž kardanového spoja
- Obr. 41: Počítačový zdroj pre napájanie robota
- Obr. 42: Render hlavy robotického hada bez vrchného krytu
- Obr. 43: Predný pohľad zobrazujúci konektory arduina Leonardo
- Obr. 44: Vnútorne rozloženie komponentov
- Obr. 45: Nepájivé pole 5x5
- Obr. 46: Pohľad zospodu na vrchný kryt hlavy
- Obr. 47: Bočný prierez hlavou zobrazujúci uchytenie dosky plošných spojov senzora
- Obr. 48: Posuv registra o  $n=1$  prvkov, na prvé miesto sa dosadí hodnota z generátora
- Obr. 49: Posuv registra o  $n=2$  prvkov, na prvé dve miesta sa dosadí hodnota z generátorov
- Obr. 50: Vývojový diagram hlavného priebehu kódu
- Obr. 51: Arduino Leonardo Pinout [17]
- Obr. 52: Dve vzorky na periódu
- Obr. 53: Osem vzoriek na periódu
- Obr. 54: Vývojový diagram vykonávania pomocného pohybu
- Obr. 55: Testovacia zostava s nastaviteľným sklonom
- Obr. 56: Značky na testovacej rovine, rozteč 10 cm
- Obr. 57: Články s kolieskami
- Obr. 58: Články bez koliesok
- Obr. 59: Články s prstencami
- Obr. 60: Rozmery modelu prstencov z náčrtu programu Inventor
- Obr. 61: Rozdiel prstencov kruhového tvaru a tvaru „D“
- Obr. 62: Penové podložky nalepené na spodnej strane prstencov
- Obr. 63: Graf závislosti rýchlosti na sklone roviny
- Obr. 64: Graf uhlovej rýchlosti počas priebehu otáčania sa o 180°
- Obr. 65: Rýchlosť pri prevracaní do strany
- Obr. 66: Graf priemernej rýchlosti pri pohybe do strany
- Obr. 67: Detail planárneho robotického hada
- Obr. 68: Robotický had v polohe predvádzajúca pohyblivosť

Tab. 1: Zoznam príkazov robotického hada

Tab. 2: Výsledné hodnoty pri vzorkovaní  $k = 2$

Tab. 3: Výsledné hodnoty pri vzorkovaní  $k = 8$



## 8 ZOZNAM PRÍLOH

Súbor obsahujúci kód Arduina

Fyzický model robotického hada

Odkaz pre nástroj Desmos: <https://www.desmos.com/calculator/39ge0listl>