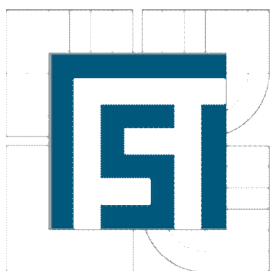




VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA STROJNÍHO INŽENÝRSTVÍ
ÚSTAV AUTOMATIZACE A INFORMATIKY

FACULTY OF MECHANICAL ENGINEERING
INSTITUTE OF AUTOMATION AND INFORMATICS

VYUŽITÍ OPAKOVANĚ POSILOVANÉHO UČENÍ PRO ŘÍZENÍ ČTYŘNOHÉHO ROBOTU

Using of Reinforcement Learning for Four Legged Robot Control

DOKTORSKÁ PRÁCE
DOCTORAL THESIS

AUTOR PRÁCE
AUTHOR

ING. VÍT ONDROUŠEK

VEDOUCÍ PRÁCE
SUPERVISOR

doc. RNDr. Ing, Tomáš Březina, CSc.

BRNO 2011

Abstrakt

Disertační práce je zaměřena na využití opakovaně posilovaného učení pro řízení chůze čtyřnohého robotu. Hlavním cílem je předložení adaptivního řídicího systému kráčivého robotu, který budu schopen plánovat jeho chůzi pomocí algoritmu Q-učení. Tohoto cíle je dosaženo komplexním návrhem třívrstvé architektury založené na paradigmatu DEDS. Předkládané řešení je založeno na návrhu množiny elementárních reaktivních chování. Prostřednictvím simultáních aktivací těchto elementů je vyvozena množina kompozitních řídicích členů. Obě množiny zákonů řízení jsou schopny operovat nejen na rovinném, ale i v členitém terénu. Díky vhodné diskretizaci spojitého stavového prostoru je sestaven model všech možných chování robotu pod vlivem aktivací uvedených základních i složených řídicích členů. Tento model chování je využit pro nalezení optimálních strategií řízení robotu prostřednictvím schématu Q-učení. Schopnost řídicí jednotky je ukázána na řešení tří komplexních úloh: rotace robotu, chůze robotu v přímém směru a chůze po nakloněné rovině. Tyto úlohy jsou řešeny prostřednictvím prostorových dynamických simulací čtyřnohého kráčivého robotu se třemi stupni volnosti na každou z noh. Výsledné styly chůze jsou vyhodnoceny pomocí kvantitativních standardizovaných ukazatelů. Součástí práce jsou videozáznamy verifikačních experimentů ukazující činnost elementárních a kompozitních řídicích členů a výsledné naučené styly chůze robotu.

Klíčová slova

kráčivý robot, kráčivá chůze, Q-učení, posilované učení

Abstract

The Ph.D. thesis is focused on using the reinforcement learning for four legged robot control. The main aim is to create an adaptive control system of the walking robot, which will be able to plan the walking gait through Q-learning algorithm. This aim is achieved using the design of the complex three layered architecture, which is based on the DEDS paradigm. The small set of elementary reactive behaviors forms the basis of proposed solution. The set of composite control laws is designed using simultaneous activations of these behaviors. Both types of controllers are able to operate on the plain terrain as well as on the rugged one. The model of all possible behaviors, that can be achieved using activations of mentioned controllers, is designed using an appropriate discretization of the continuous state space. This model is used by the Q-learning algorithm for finding the optimal strategies of robot control. The capabilities of the control unit are shown on solving three complex tasks: rotation of the robot, walking of the robot in the straight line and the walking on the inclined plane. These tasks are solved using the spatial dynamic simulations of the four legged robot with three degrees of freedom on each leg. Resulting walking gaits are evaluated using the quantitative standardized indicators. The video files, which show acting of elementary and composite controllers as well as the resulting walking gaits of the robot, are integral part of this thesis.

Keywords

walking robot, walking gait, Q-learning, reinforcement learning

Bibliografická citace

ONDROUŠEK, V. Využití opakovaně posilovaného učení pro řízení čtyřnohého robotu. Brno: Vysoké učení technické v Brně, Fakulta strojního inženýrství, 2011. 85 s. Vedoucí dizertační práce doc. RNDr. Ing. Tomáš Březina, CSc.

Čestné prohlášení

Prohlašuji, že jsem disertační práci zpracoval samostatně dle pokynů vedoucího disertační práce a s použitím uvedené literatury.

Poděkování

Tímto děkuji školiteli doc. Tomáši Březinovi za vedení této práce, cenné rady a připomínky. Dále bych chtěl poděkovat za pomoc a spolupráci svým kolegům Jiřímu Krejsovi, Stanislavu Věchetovi a Petru Schreibrovi. V neposlední řadě pak děkuji za velkou podporu nejen při vypracování této práce Pavle Oplatkové.

Obsah

1	Úvod	9
1.1	Motivace	9
1.2	Zadání a zaměření práce	10
1.3	Stanovené cíle	10
2	Teoretický rozbor problematiky	11
2.1	Kráčivý pohyb robotů	11
2.1.1	Popis robotu	11
2.1.2	Terminologie problematiky chůze kráčivých robotů	12
2.1.3	Popis a hodnocení chůze kráčivých robotů	13
2.2	Přístupy k plánování chůze	15
2.2.1	Přístup založený na modelu	15
2.2.2	Přístup založený na chování	16
2.2.3	Hybridní a ad-hoc přístupy	17
2.3	Architektura řízení využívající DEFS pro plánování chůze	18
2.3.1	Vrstva zpětnovazebných řídicích členů	18
2.3.1.1	Řídicí člen postoje	19
2.3.1.2	Řídicí člen kontaktu	20
2.3.1.3	Řídicí člen trajektorie	20
2.3.2	Vrstva zahrnující koordinační mechanismus DEFS	21
2.3.2.1	Složené řídicí členy	21
2.3.2.2	Reprezentace stavu řízené soustavy	22
2.3.2.3	Postup sestavení koordinačního mechanismu	23
2.3.3	Vrstva učení	25
2.3.4	Shrnutí	26
2.4	Strojové učení	27
2.4.1	Vybrané algoritmy strojového učení	27
2.4.2	Posilované učení	29
2.4.3	Q-učení	30
2.4.4	Q-učení v kontextu řízení kráčivého robotu	31
2.5	Zvolený přístup pro dosažení stanovených cílů	33
2.5.1	Kritické zhodnocení teoretických východisek	33
2.5.2	Zvolený přístup a přehled dílčích cílů	34
2.5.3	Postup řešení	35
3	Praktická část – souhrn dosažených výsledků	36
3.1	Volba prostředků pro verifikaci	36
3.2	Platforma použitá pro verifikaci návrhu řešení	37
3.2.1	Popis robotu	38
3.2.2	Volba souřadných systémů	39
3.2.3	Vytvoření modelu robotu	41
3.2.4	Vybrané konfigurace	42
3.3	Sestavení elementárních řídicích členů	44
3.3.1	Řídicí člen postoje	44
3.3.1.1	Popis vybraných instancí	44
3.3.1.2	Sestavení potenciálové funkce	45
3.3.1.3	Posouzení charakteru použité funkce	47

3.3.2	Řídicí člen kontaktu.....	51
3.3.2.1	Popis vybraných instancí.....	51
3.3.2.2	Sestavení potenciálové funkce	51
3.3.2.3	Posouzení charakteru použité funkce	53
3.3.3	Řídicí člen trajektorie	56
3.4	Projekt implementující navržené řešení	56
3.4.1	Popis činnosti programu.....	57
3.4.2	Struktura projektu.....	59
3.4.3	Způsob implementace vybraných problémů	60
3.4.3.1	Sestavení koordinačního mechanismu	60
3.4.3.2	Činnost základních řídicích členů	60
3.4.3.3	Činnost složených řídicích členů.....	61
3.4.3.4	Implementace učicího algoritmu	62
3.5	Výsledné elementy chování robotu	63
3.5.1	Chování základních řídicích členů při výskytu překážek.....	63
3.5.2	Chování kompozitních řídicích členů.....	67
3.6	Výsledná chůze robotu při komplexních úlohách	68
3.6.1	Úloha rotace robotu	69
3.6.1.1	Vstupní parametry úlohy	69
3.6.1.2	Odvození modelu chování a návrh parametrů učení	72
3.6.1.3	Výsledná naučená chůze.....	73
3.6.2	Úloha dopředné chůze	74
3.6.2.1	Vstupní parametry úlohy	74
3.6.2.2	Odvození modelu chování a návrh parametrů učení	76
3.6.2.3	Výsledná naučená chůze.....	77
3.6.3	Úloha dopředné chůze na nakloněné rovině.....	79
3.6.3.1	Vstupní parametry úlohy	79
3.6.3.2	Výsledná naučená chůze.....	79
4	Závěr.....	81
4.1	Shrnutí	81
4.2	Diskuse dosažených výsledků	82
4.3	Přínos práce	86
4.3.1	Přínos teoretický.....	86
4.3.2	Přínos praktický.....	86
4.3.3	Přínos pedagogický	86
	Literatura	87
	Seznam obrázků.....	91
	Seznam příloh	92

1 Úvod

1.1 Motivace

Problematika autonomních mobilních robotů v současné době stále spadá do popředí zájmu vědeckého výzkumu mobilní robotiky. Dle [9] jsou autonomní roboty třídou technických systémů, které nahrazují mobilní, lokomoční a intelektuální funkce člověka, jejichž základním rysem je cílová činnost bez přímé účasti operátora při jejich řízení, mimo činnost dohlížecí. Hledání možností realizace těchto složitých autonomních systémů má velký potenciál využitelnosti alternativních funkcí v mnoha oborech lidské činnosti. Právě míru využitelnosti jsem zohlednil i při specifikaci svého zájmu o kráčivé roboty namísto kolových a pásových mobilních robotů, k čemuž mě vedly tyto konkrétní důvody:

- přibližně polovina povrchu Země je nedostupná kolovým či pásovým vozidlům, jak uvádí [31] podle výzkumu americké armády [39], zatímco je běžně dostupná čtyřnohým živočichům,
- nabízejí větší možnosti při překonávání překážek a tím i dostupnost do míst kam se kolové roboty nedostanou, např. sestup robotu Danete II do kráteru sopky [42]
- oproti kolovému či pásovému pohybu lze předpokládat lepší schopnost vypořádat se s různými půdními podmínkami,
- jejich chůze může být reprezentována posloupností opěrných bodů (diskrétní stopou) [32], čehož lze s výhodou využít při pohybu v úzkých prostředích nebo v terénu s velmi členitým povrchem,
- nohy mohou mít v terénu méně ničivý dopad než kola či pásy.

Vývoj autonomního robotu představuje velmi komplexní a složitou úlohu, při níž je zapotřebí řešit velké množství problémů, často interdisciplinárního charakteru. Pro každou část robotického systému jsou typické jiné problémy, např.: návrh a výpočet konstrukce robotu, volba senzorické soustavy, návrh jednotky řízení, ad. Některé nevýhody kráčivých podvozků, které zvyšují náročnost jejich vývoje oproti kolovým dle [31]:

- kinematický a dynamický popis robotu je nelineární, výsledný model je pak velmi složitý, na druhou stranu zjednodušený model zase nemusí nabízet dostačující shodu s realitou. Navíc některé parametry robotu, jako např. rozměry a hmotnost přenášeného břemene, nejsou známy předem, což znemožňuje nalezení těžiště, apod.,
- pracovní prostředí, ve kterém se robot pohybuje bývá dynamické, případně zcela neznámé,
- kráčivé roboty mají vysoký počet stupňů volnosti vyžadujících velký počet akčních členů, což vede ke zvýšeným nárokům na jednotku řízení a rovněž k vyšší energetické náročnosti. (Např.: pro umožnění odděleného pohybu trupu

a noh kráčivého robotu jsou pro každou nohu zapotřebí 3 stupně volnosti, tj. pro šestinohý robot celkem 18 akčních členů),

- soustava mobilního robotu je často nestabilní, pro zajištění stability nestačí pouze ukončení činnosti všech akčních členů a uzamčení aktuální konfigurace robotu,
- cíl udržovat stabilitu je těžko rozložitelný na zákony řízení akčních členů.

Inteligentní robotický systém lze rozdělit na 4 stavební části: mechanická stavba a pohony (machineware), technické vybavení zahrnující elektronické a elektrické obvody a senzory, výpočetní, řídicí a komunikační zařízení (hardware), programové zabezpečení činnosti a diagnostiky (software) a automaticky tvořená báze znalostí (brainware), [9]. Disertační práce je zaměřena na problematiku související s programovým vybavením kráčivých robotů. Mezi hlavní problémy, které musí software kráčivého robotu dokázat vyřešit, patří:

- lokalizace robotu v mapě, případně tvorba mapy prostředí, např. [38]
- navigace robotu do cílové polohy v prostředí, např. [45]
- plánování pohybu (chůze) robotu, např. [42]
- zpracování(interpretace) dat od senzorů, např. [41]
- regulace a řízení akčních členů, např. [4]

V této disertační práci je teoreticky i prakticky řešena problematika plánování chůze autonomních robotů. Snahou autora je navrhnout a verifikovat jednotku sloužící pro plánování chůze čtyřnohého robotu ve členitém terénu. Jak dokládá rešeršní studie, která byla předložena v rámci státní doktorské zkoušky autora, jedná se o aktuální problém, který je stále v popředí zájmu vědeckého výzkumu.

1.2 Zadání a zaměření práce

Úkolem práce je předložit návrh a simulační ověření adaptivního řídicího systému čtyřnohého robotu, založeného na Q-učení. Prioritou je nalezení (prostřednictvím schématu Q-učení) optimálních simultánních aktivací řídicích členů v různých stavech robotu spolu s vhodnou diskretizací spojitého stavového prostoru. Navrhnutý řídicí systém by měl robotu umožnit řešení tří úloh: kráčivý pohyb v přímém směru, vyhýbání se překážkám a překonání vybraných nouzových stavů robotu.

1.3 Stanovené cíle

V rámci státní doktorské zkoušky byly s ohledem na zadání stanoveny tyto cíle disertační práce:

- návrh a implementace množiny elementárních způsobů chování vhodných pro plánování chůze čtyřnohého robotu,
- implementace vhodného koordinačního, resp. plánovacího schématu sloužícího pro propojování těchto chování, např. přístup založený na DEDS,
- dosažení různých stylů chůze čtyřnohého robotu pomocí navrženého řešení,
- verifikační experiment použitého přístupu.

2 Teoretický rozbor problematiky

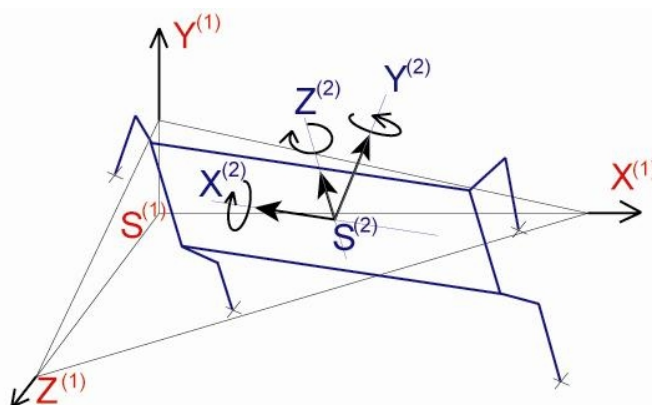
2.1 Kráčivý pohyb robotů

Kračivý pohyb jako fenomén je námětem řady vědeckých studií. Nejprve byl předmětem zájmu biologů, kteří sledovali tělesnou stavbu a pohyb zvířat. Naproti tomu technicky zaměřený výzkum se orientuje na hledání abstraktnějšího matematického popisu lokomoce zvířat, který umožňuje zkonstruování biologicky inspirovaných kráčivých strojů [3]. Biologická inspirace kráčivých robotů nejčastěji vychází ze savců, plazů, hmyzu aj. Přístupy k výběru a aplikaci poznatků z biologie pro potřeby konstrukce kráčivých robotů se značně liší: může se jednat o napodobení stavby těla a počtu nohou (tj. návrh konstrukce robotu) [29], existují práce zabývající se vývojem umělých svalů [2], řada podnětů je využívána i při navrhování stylu chůze robotu [13], rovněž existují snahy napodobit při návrhu řídicí jednotky činnost nervového systému živočichů (např. hmyzu) [26],[5] atd.

V této kapitole je uvedena terminologie související s problematikou chůze kráčivých robotů a nejčastěji používaná kritéria umožňující kvantitativní popis chůze, dále jsou zde popsány a ukázány pojmenované styly chůze, mající význam pro technickou praxi. Pozornost je věnována především čtyřnohým robotům, ovšem většina zde uvedených informací je použitelná obecně pro kráčivé stroje s libovolným počtem nohou.

2.1.1 Popis robotu

Kračivý robot tvořen trupem, nohama a (zejména u humanoidních) hlavou, [31]. Trup robotu (*trunk*) někdy označován jako tělo (*body*) představuje hlavní část robotu bez končetin a hlavy. Končetiny jsou s tělem spojeny prostřednictvím kyčelního kloubu (*hip*). Často lze nohu kinematicky reprezentovat jako soustavu prutů vzájemně spojených sférickými vazbami. Ovšem obecně lze říci, že je konstrukce nohy robotu různá, např. [29] zmiňuje několik speciálních konstrukcí noh robotů. Často je používána noha pantografická, např. [23]. Podrobný popis používaných konstrukcí nohou přesahuje rozsah této práce.



Obr. 1: Schématické naznačení os rotace

Dále jsou v souvislosti s popisem robotu, umístění (např. překážek) a směrů, používány tyto pojmy:

- laterální: vyskytující se na levé či pravé straně trupu robotu,
- ipsilaterální: vyskytující se na téže straně trupu robotu, [53]
- kontralaterální: vyskytující se na opačné straně těla,
- sagitální rovina: předozadní, rovnoběžná se střední (mediální) rovinou těla, [53]. Jedná se o rovinu kolmou na rovinu čelní, normála na $X^{(2)}$, viz. Obr. 1, rozděluje robot na dvě ekvivalentní poloviny (levou a pravou)
- longitudinální osa: podélná osa, viz. Obr. 1, osa $X^{(2)}$
- úhel klopení: (pitch angle) úhel popisující rotaci robotu kolem osy $X^{(2)}$ [19]
- úhel klonění: (roll angle) úhel popisující rotaci robotu kolem osy $Z^{(2)}$ [19]
- úhel bočení: (yaw angle) úhel popisující rotaci robotu kolem osy $Y^{(2)}$ [19]
- konfigurace robotu - (*posture*) je použita v literatuře několika způsoby, např. pozice nebo orientace těla robotu, charakteristická či předpokládaná pro speciální účel. V následujícím textu je tohoto pojmu použito pro označení držení těla, postoj, polohu noh robotu a výšku trupu nad zemí.

2.1.2 Terminologie problematiky chůze kráčivých robotů

V této kapitole jsou představeny a objasněny některé pojmy z oblasti chůze kráčivých robotů tak, jak jsou chápány v odborné literatuře a používány v této práci. Uvedené informace jsou čerpány především z [1], [22], [12] a [31].

- *Chůze*: (gait) pojem kráčení nebo též způsob či styl chůze je v odborné literatuře zaveden různými definicemi. Např.: Způsob pokládání nohou při kráčení nebo běhu [11], vzorec lokomoce charakteristický pro omezený rozsah rychlostí, popsany veličinami, které se (jedna či více z nich) mění nespojitě při přechodu na jiný způsob chůze [31] dle [1], koordinovaný způsob zvedání a pokládání noh, [12].
- *Událost chůze*: položení resp. zvednutí nohy během pohybu. Pro n -nohý stroj existuje $2n$ událostí během jednoho *dlouhého kroku*.
- *Cyklus (nohy)* : typický pohyb nohy při chůzi je cyklický. Za účelem snazší analýzy a popisu je cyklus nohy rozdělován na fáze; opornou fází a fází posuvu. Někdy se uvádí dělení na čtyři fáze; oporná fáze, fáze zdvihu, posuvná fáze a fáze položení.
- *Oporná fáze*: noha robotu je použita pro podpírání a posuv těla robotu. Někdy je tato fáze označována jako postoj (stance, příp. power stroke).
- *Fáze posuvu*: noha robotu je přesouvána z jednoho opěrného bodu do jiného. Někdy je pro tuto fázi použit pojem zhoupnutí (swing).
- *Krok*: v této publikaci je používán ve stejném smyslu jako *fáze posuvu*, viz. výše.

- *Délka kroku*: vzdálenost mezi dvěma po sobě následujícími body došlapu jedné nohy v $S^{(1)}$, předpokládá se že nedochází k podkluzům.
- *Zdvih* (λ): vzdálenost, kterou urazí noha při relativním posuvu vůči kyčelnímu kloubu nohy robotu během oporné fáze [12].
- *Dlouhý krok*: (stride) je složen z tolika kroků kolik má robot noh, tj. typicky každá noha dokončí jeden pohybový cyklus, délka dlouhého kroku je vzdálenost, kterou urazí tělo (resp. těžiště) robotu během jednoho dlouhého kroku. Rychlost dlouhého kroku je (*Délka dlouhého kroku*) / (*Doba trvání dlouhého kroku*).
- *Doba cyklu* (T): délka trvání jednoho dlouhého kroku, tj. doba potřebná k ukončení jednoho cyklu pohybů noh.
- *Činitel využití* (β_{cv}): (procentuální) vyjádření podílu fáze podpírání na celkovém trvání cyklu dlouhého kroku.
- *Relativní fáze nohy i* (ϕ_i): čas uběhlý od položení referenční nohy do položení i-té nohy, zadána jako (procentuální) část doby cyklu. Popisuje fázi nohy, i , vůči libovolně zvolené referenční noze.
- *Graf chůze*: představuje grafické znázornění fází všech nohou robotu v závislosti na čase, viz obr. 2, přičemž plné čáry představují opornou fázi.
- *Obrazec podpory*: množina bodů vytvořená konvexní obálkou průmětu podpírajících částí nohou na horizontální rovinu, jež je kolmá na působící tíhovou sílu.
- *Podpůrná oblast*: vnitřní část a ohraničení obrazce podpory. Je možné setkat se i s pojmem *podpůrný mnohoúhelník*. Toto použití vychází z myšlenky, že kontakt mezi nohou a povrchem je modelován jako bod.

2.1.3 Popis a hodnocení chůze kráčivých robotů

Chůzi lze chápat např. jako koordinovaný způsob zvedání a pokládání noh, viz. kap. 2.1.2. Každá chůze je tvořena posloupností tzv. událostí chůze (zvednutí nohy, položení nohy). Jak uvádí [12], [24] existuje u čtyřnohých teoreticky 5040 různých kombinací událostí chůze, ovšem prakticky je jen malé procento z tohoto množství používáno živočichy pro chůzi. A právě tyto způsoby lokomoce jsou významné pro technickou praxi a jsou předmětem následujícího rozboru.

Jednotlivé styly chůze lze členit do skupin a hodnotit podle několika kritérií. Pro kvantitativní vyhodnocení chůze robotu je možné použít např. *činitel využití*, β_{cv} ,

a *relativní fázi*, (ϕ_i), viz. kap. 2.1.2. Další možností je využití Froudeova čísla, $\frac{u^2}{gh}$, kde

u je rychlost pohybu, g je tíhové zrychlení, h je vzdálenost kyčelního kloubu od povrchu [1]. Pro kvalitativní vyhodnocení slouží tzv. *graf chůze*, který znázorňuje fáze všech nohou v závislosti na čase, viz. obr. 2, kde plná úsečka reprezentuje opornou fázi a prázdná představuje fázi posuvu nohy. Poprvé byl použit v [36]. Pro názorné zobrazení

chůze v rovině lze např. také použít zakreslení bodů došlapu noh v rovině podložky do $SS^{(1)}$.

Primárně lze rozdělit chůze na *cyklické* a *necyklické* (též *volné*). U cyklické chůze je stanoven pohyb každé z noh a ten je následně opakován v pravidelných časových intervalech. Tento typ chůze umožňuje snadnější analýzu a je používán především na rovném, případně jen málo členitém terénu. Naopak, jak poukázal [22], je nevhodný (mj. energeticky) pro zvlněný a nerovný terén, kde je využíváno volného (necyklického) způsobu chůze.

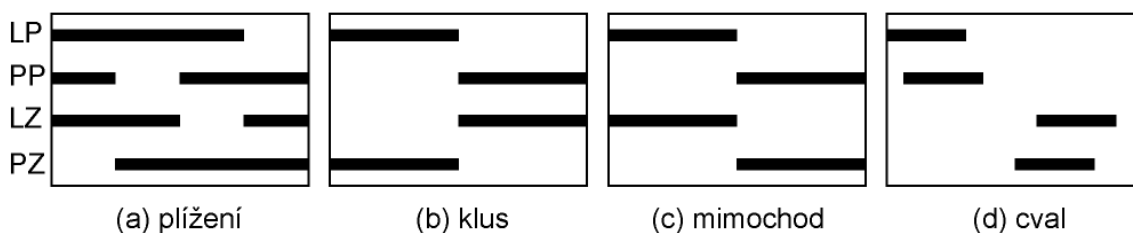
Mezi cyklické styly chůze patří např.:

- vlnitá chůze: (wave gait) zvedání (a pokládání) noh začíná zadními nohama a pokračuje v dopředném směru. Laterálně spárované nohy jsou fázově posunuty přesně o polovinu cyklu kroku. Jedná se o typickou chůzi hmyzu.
- plížení: (crawl gait) vlnitá chůze čtyřnohého robotu, ovšem s tím rozdílem, že počátek pohybu může být započat libovolnou nohou. Toto je typický styl chůze používaný čtyřnohými živočichy při pomalém pohybu, např. [31]. Podle [24] se jedná o optimální staticky stabilní chůzi čtyřnohých.
- krabí chůze: (crab-walking) představuje kráčivý pohyb jehož směr svírá s longitudinální osou robotu úhel φ , označovaný jako úhel bočení.
- plazivá chůze: (creeping gait) - velmi pomalá chůze, u níž je v přenosové fázi vždy nejvýš jedna noha.
- otáčivá chůze (turning gait) - ustálený kruhový pohyb, u kterého libovolně zvolený bod trupu robotu rotuje kolem neměnného středu otáčení.

Mezi volné styly chůze patří např.:

- chůze "následuj vůdce" - zadní nohy jsou pokládány na body (nebo velmi blízko bodům) došlapu předních noh.

Dále lze rozdělit chůzi na *spojitou* a *nespojitou*. U spojitě chůze se trup robotu neustále pohybuje nenulovou rychlostí v longitudinálním směru. Nespojitý styl chůze je charakteristický tím, že k pohybu trupu robotu dochází pouze v některých fázích kráčení. Chůzi lze označit za *symetrickou*, pokud je pohyb libovolného kontralaterálně umístěného páru nohou vzájemně posunut o půl cyklu. *Regulární* chůze je taková chůze, při níž má každá noha stejný činitel využití, β_{cv} , tj. během takovéto chůze je každá noha v kontaktu se zemí stejné množství času [22].



Obr. 2: Grafy vybraných pojmenovaných stylů chůze koně, [31].

V publikaci [12] je uplatněno rozdělení chůze čtyřnohých živočichů podle činitele využití, na základě čehož je definována chůze plazivá ($\beta_{cv} > 0,75$), rychlejší chůze ($0,5 < \beta_{cv} < 0,75$) a běh ($\beta_{cv} < 0,5$).

Nejznámější a zároveň nejčastěji se vyskytující styly chůze u čtyřnohých živočichů jsou pojmenovány. Např. kůň používá tyto styly chůze plížení ($\beta_{cv} = 0,75$), klus ($\beta_{cv} = 0,5$), mimochod ($\beta_{cv} = 0,5$) a rotační cval ($\beta_{cv} = \frac{5}{16}$), příslušné grafy chůze viz. Obr. 2.

Způsobu hodnocení a popisu chůze kráčivých robotů uvedených v této kapitole bude použito pro vyhodnocení a porovnání dosažených výsledků popsanych v této disertační práci.

2.2 Přístupy k plánování chůze

2.2.1 Přístup založený na modelu

Někdy je tento přístup, např. [22], též označován jako geometrický. Jedná se o analytické řešení, které pro plánování chůze využívá model kinematiky a případně i model dynamiky robotu a prostředí. Odvození kinematického a dynamického popisu kráčivého stroje lze nalézt např. v [32]. Je zřejmé, že tento přístup vyžaduje offline zpracování předcházející vlastní chůzi. Nezbytná je přesná znalost konstrukce robotu a mapy prostředí, ovšem ne vždy je možné takovýto model odvodit předem. Musí se jednat o statické prostředí, v němž nedochází k výrazným změnám. Navíc [22] poukázal, že bývá možnost výskytu neurčitostí a proměnlivosti prostředí v rámci uvedeného přístupu opomíjena. V případě výskytu nepřesností v modelu robotu či prostředí může docházet k selhávání řídicí jednotky, což může vést až k pádu a poškození robotu. Typické je využití cyklických stylů chůze, které umožňují snadnější analýzu a návrh výsledného pohybu. Využití cyklické chůze je vhodné pouze pro rovinný nebo jen velmi málo členitý terén, viz. kap. 2.1.3. Dle použitého kritéria stability může být dosaženo staticky i dynamicky stabilní chůze.

Tento přístup je využit např. v [18] pro návrh chůze čtyřnohého robotu. Autor uvádí, že alespoň částečné využití analytického přístupu k návrhu chůze je nezbytné. Snahou je v tomto případě vyvinout plazivý styl chůze přímo vpřed a otáčivý styl chůze čtyřnohého robotu s pantografickými nohami. Zvolená posloupnost pohybů noh pro chůzi v této studii odpovídá optimálnímu cyklickému stylu chůze čtyřnohých (viz. kap. 2.1.3), např. [24], a je stanoven předem, tj. offline. Problém návrhu chůze je tak v tomto příspěvku omezen pouze na implementaci pro konkrétní robot.

Výhody a nevýhody přístupu založeného na modelu se dají shrnout následovně:

Výhody

- při použití dostatečně přesného modelu robotu a pohybu ve známém statickém prostředí lze dosáhnout různých stabilních a udržitelných stylů chůze
- lze dosáhnout staticky i dynamicky stabilní chůze

- nevyžaduje čas pro učení se chůze za použití robotu (neriskuje se tak poškození robotu při upadnutí v počátečních fázích chůze)

Nevýhody

- dosažitelné výsledky jsou limitovány přesností použitého modelu
- model dynamiky robotu je nelineární
- vyžadována znalost konstrukce robotu (rozměry trupu a noh, hmotnost robotu, hmotnost a tvar přenášeného břemene apod.)
- vyžadováno offline zpracování
- nevhodné pro dynamická (proměnná) prostředí
- typicky používáno pro cyklické styly chůze

2.2.2 Přístup založený na chování

Dle [14] nebo [22] byl tento přístup poprvé použit v [7]. Paradigma řízení chůze robotu založeného na chování bylo vyvinuto ve snaze eliminovat nevýhody předchozího přístupu, zejména omezení závislosti návrhu řídicí jednotky na modelu (robotu a prostředí) a umožnění kráčení robotu v proměnlivém prostředí. Návrh řízení v tomto případě vychází z předpokladu, že inteligentní chování robotu může vzniknout na základě interakce malého počtu zpětnovazebných primitiv chování nejnižší úrovně [22]. Jednotlivá chování jsou většinou realizována pomocí konečných stavových automatů (dále jen KSA). Architektura řídicí jednotky je vícevrstvá, přičemž každá vrstva je tvořena několika KSA, např. jeden KSA řídí činnost jedné nohy robotu. Nejnižší vrstvy zajišťují jednoduché chování se zpětnou vazbou. Kterákoliv z nadřazených vrstev může omezit činnost podřízené vrstvy. Libovolná množina nižších vrstev vytváří samostatně řídicí subsystém. U [7] byla jednotka řízení vystavěna na zpětnovazebných primitivech chování pojmenovaných takto: *zvednutí se, jednoduchá chůze, vyvažování sil, zvednutí nohy, řízení úhlu klopení, plížení, otáčivé plížení*.

[22] zmiňuje, že tento přístup umožňuje snadné rozšíření pouhým přidáním další vrstvy KSA. S rostoucí složitostí řešené úlohy roste i náročnost práce konstruktéra se začleněním přidávaných vrstev do řídicí jednotky, což může v konečném důsledku vést k nutnosti opakovaného předefinování návrhu. [14] řadí mezi nevýhody toho přístupu nutnost navrhnout jednotlivá chování ručně pro každý robot, prostředí a úlohu, navíc jich je zapotřebí velké množství. Rovněž neexistuje jednotná architektura či předpis, podle kterého by bylo možné tato chování propojovat.

V současném výzkumu kráčivých robotů je možné nalézt mnoho prací využívajících tento přístup, z nichž např. [42] popisuje návrh řídicí jednotky založené na chování pro zajištění chůze osminohého robotu *Danete II* v extrémně členitém a nepřístupném terénu. Schopnost řídicí jednotky ukazuje na autonomním sestupu tohoto robotu do vulkánu hory Spurr na Aljašce. [30] úspěšně využívá řízení založené na chování pro zajištění dynamicky stabilního pohybu kráčivých, plazivých (člámkovitých) ale i hybridních (2 nohy, 2 kola) robotů. Využití tohoto přístupu pro chůzi čtyřnohého robotu *Sleipner III* v neznámém prostředí lze nalézt v např. v [28].

Provedená rešeršní studie ukazuje, že největší pozornost je věnována návrhu architektury řídicí jednotky a způsobu propojení jednotlivých elementárních chování. Výhody a nevýhody toho přístupu lze shrnout následovně :

Výhody

- není nutné vytvářet model robotu a prostředí
- vhodné i pro strukturovaná, neznámá, dynamická prostředí
- snadná rozšiřitelnost řídicí jednotky (jen při malé změně cílů úlohy)

Nevýhody

- nutnost sestavit velké množství primitiv chování
- absence jednotné architektury určující způsob vzájemného propojení těchto chování
- s rostoucím počtem primitiv chování roste náročnost vzájemného propojení
- změna úlohy může vést k nutnosti zcela nového návrhu řídicí jednotky

2.2.3 Hybridní a ad-hoc přístupy

Kromě těchto dvou výrazných skupin přístupů k plánování chůze robotů lze v odborném diskurzu nalézt celou řadu dalších řešení. Většina těchto přístupů se snaží využít výhody obou přístupů a potlačit jejich nevýhody.

Mnoho vědeckých studií hledá inspiraci u biologických systémů. Např. [17] se inspiruje imunitním systémem živočichů. Uvádí přístup založený na chování, který ovšem pro propojení jednotlivých primitiv využívá principy fungování imunitního systému. Obdobně [26] představuje řešení využívající elementy chování propojené pomocí mechanismů uplatňovaných u smyslového vnímání člověka.

Celá řada prací, např. [13], [5], [37], se zaměřuje na napodobení tzv. centrálního generátoru vzorů (central pattern generator, dále jen CPG). Dle [4] bylo dokázáno, že neuronová síť v míše produkuje rytmické vzory, které při propojení se svaly dokáží generovat pohyb (pohybový vzor) zvířete. Takovýto zabudovaný míšní okruh je označován jako CPG. Jeho přítomnost byla zjištěna u mnoha obratlovců. Časté jsou snahy napodobit fungování CPG pomocí neuronových sítí, např. typu *Hodgkin-Huxley* anebo *leaky integratoru*. Např. [4] pomocí neuronové sítě typu *leaky integrator* navrhuje řídicí jednotku zajišťující chůzi čtyřnohého robotu Aibo (komerčně vyráběný robot firmy Sony), která umožňuje plynulý přechod mezi chůzí, klusem a cvalem. Podrobnější rozbor lze nalézt v [5].

Příklad hybridního přístupu lze nalézt např. u [13]. V tomto příspěvku je uveden návrh řídicí jednotky, která zahrnuje CPG, jednotkou plánující chůzi a jednotku zajišťující optimalizaci. Činnost CPG je zde realizována pomocí řídicích členů s přímou regulací a zajišťuje vytváření základních stylů chůze. Jednotka plánující chůzi zabezpečuje laterální pohyb a otáčivou chůzi, jejíž návrh je založen na aproximačním řešení inverzní úlohy kinematiky. Činnost optimalizační jednotky je implementována pomocí neuronové sítě a hlavním cílem je oprava nedostatků zapříčiněných nepřesnostmi v aproximaci řešení inverzní kinematiky.

Zcela odlišný přístup k plánování chůze robotu představuje využití paradigmatu DEDS (Discrete Event Dynamic System), tj. modelování pomocí dynamického systému s diskretními událostmi. Toto řešení lze považovat za hybridní přístup, ovšem v následujícím textu bude předmětem detailnějšího rozboru, a proto je mu věnována samostatná kapitola 2.3.

2.3 Architektura řízení využívající DEDS pro plánování chůze

Tato architektura řízení byla poprvé představena v disertační práci [15]. Jedná se o třívrstvou architekturu řízení, která umožňuje autonomní pohyb robotu v neznámém prostředí. Celý systém je vystavěn na malé množině zpětnovazebných řídicích členů, které pracují spojitě v čase, a jsou spojeny se senzorickou soustavou robotu stejně tak jako s jeho akčními členy. Nadřazená vrstva zajišťuje diskretizaci spojitého prostoru na diskretní události, které je možné popsat formálně. Na základě tohoto formálního popisu pak tato vrstva vyřazuje nepřipustné aktivace řídicích členů, čímž se jednoznačně snižuje velikost stavového prostoru, se kterým pracuje nejvyšší vrstva řídicí jednotky. Ta implementuje vybraný algoritmus učení pro krátkodobé plánování akcí, které povedou ke splnění stanoveného cíle.

2.3.1 Vrstva zpětnovazebných řídicích členů

Nejnižší vrstvu popisované architektury řízení tvoří tzv. báze řízení, $\{\Phi\}$. Jedná se o množinu tvořenou řídicími členy, Φ_i . Každý člen této množiny je navržen pro převod obecného definičního prostoru, S , na výstupní prostor A . Pro každý z těchto zákonů řízení pak platí:

$$\Phi_{i a_i}^{s_i} : S_i \rightarrow A_i, a_i = f_i(s_i), \text{ kde } s_i \in S_i, a_i \in A_i \quad (1)$$

Instance řídicího členu je vytvořena pomocí propojení přípustných vstupních zdrojů, $\{\sigma\}$, a přípustných výstupních zdrojů, $\{\tau\}$, s řídicím členem, Φ_i . Zdroje jsou navázány na řídicí člen pomocí transformace charakteristické pro řízení. Vstupní transformace, σ , zobrazuje stav vstupních zdrojů do definičního prostoru zákona řízení. Výstupní transformace, τ , převádí výstup zákona řízení na aktivace výstupních zdrojů, pak platí:

$$\sigma : \sigma \rightarrow S_i, \quad \tau : A_i \rightarrow \tau \quad (2)$$

Instance řídicích členů, označované také jako zákony řízení, vytvořené tímto způsobem jsou tedy zobrazením vstupních zdrojů na zdroje výstupní, které umožňují informacím ze senzorů přímo ovlivňovat akční členy robotu, tj.:

$$\sigma_{i\tau}^{\sigma} : \sigma \rightarrow \tau \quad (3)$$

Pro daný robot pak lze tímto způsobem sestavit množinu řídicích členů, $\{C\}$, takto:

$$\{C\} = \{\Phi_{i\tau_j}^{\sigma_k}\} \quad (4)$$

kde Φ_i je i -tý člen báze řízení, σ_k a τ_j jsou podmnožiny přípustných vstupních a výstupních zdrojů robotu. Při návrhu řídicího členu je proto vždy nutné určit, které

vstupní a výstupní zdroje robotu mohou být používány, aby se předešlo vytvoření nekonzistentních řídicích členů. Jako vstupní a výstupní zdroje mohou být použity např. úhel klonění, klopení a bočení těla robotu nebo jeho pozice x, y, z v prostoru, případně kinematické řetězce noh, formálně označované 0,1,2,3.

Při hledání předpisu funkce, která zajistí vhodné zobrazení vstupních zdrojů na výstupní, je zapotřebí věnovat zvýšenou pozornost výslednému charakteru této funkce. Výsledná potenciálová plocha funkce by totiž měla mít právě jedno lokální a tím zároveň i globální minimum. Tento lokální extrém pak představuje cílovou konfiguraci robotu, které má být dosaženo při aktivaci daného zákona řízení, a jeho nalezení je možná i za použití méně náročnějších metod, jako například gradientní.

Následující text je zaměřen na tři vybrané řídicí členy, které mohou být použity pro plánování chůze čtyřnohého robotu.

2.3.1.1 Řídicí člen postoje

Návrh tohoto řídicího členu lze najít v [22]. Cílem tohoto zákona řízení, Φ_p , je dosažení optimálních poloh noh robotu vzhledem k jejich pracovnímu prostoru. Jako míry optimality je použito heuristické míry postoje, m :

$$m = \prod_{i < 4} \left(p_i \prod_{j < 3} \cos(\Theta_{\varepsilon ij}) \right) \quad (5)$$

kde i je počet noh, j počet vazeb na každé z noh, $\Theta_{\varepsilon ij}$ je hodnota úhlu mezi tělesy spojenými j -tou rotační vazbou na i -té noze. Přičemž tento úhel je dle uvedeného zdroje normalizován na interval $\langle -\pi, \pi \rangle$. Hodnota p_i pak označuje manipulovatelnost i -té nohy, což je míra polohy, pro kterou platí:

$$p_i = \det(\sqrt{J_i J_i^T}) \quad (6)$$

kde $\det$ značí determinant, index T značí transpozici, J značí čtvercovou matici, na jejíž hlavní diagonále jsou umístěny složky vektoru $\vec{tn} = N - T$, kde N je bod došlapu příslušné nohy a T je poloha těžiště robotu. Z uvedených rovnic (5) a (6) vyplývá, že heuristická míra postoje je tím větší, čím jsou vazby více vzdáleny svým mezním polohám, a také čím jsou nohy schopné vyvinout větší rychlost ve všech směrech [22].

Jako množin vstupních a výstupních zdrojů tohoto řídicího členu pro potřeby chůze robotu lze použít následující:

$$\sigma = \{0, 1, 2, 3\}, \quad \tau = \{0, 1, 2, 3, x, y, \varphi\} \quad (7)$$

kde čísla 0 až 3 formálně označují kinematické řetězce jednotlivých noh robotu, x, y je poloha středu těla robotu a φ je úhel bočení těla robotu. Např. instance $\Phi_{P_{x,y}}^{0,1,2}$ se tak snaží změnou polohy těla robotu v osách x, y docílit maximální hodnoty heuristické míry postoje vypočítané na základě konfigurace noh 0,1,2, zatímco konfigurace nohy 3 není uvažována.

2.3.1.2 Řídicí člen kontaktu

Tento řídicí člen je používán pro polohování došlapu noh na základě lokální geometrie prostředí tak, aby minimalizoval zbytkové síly a momenty působící na těžiště robotu. V disertační práci [15], která se podrobně věnuje popisované hybridní architektuře, je tento řídicí člen kontaktu popsán pouze povrchně a pro detailní popis odkazuje na práci [16]. Další práce, které se zabývají jednotlivými částmi popisované architektury [22] nebo [31] obsahují pouze zevrubný popis řídicího členu kontaktu a pro detailnější rozbor odkazují rovněž na práci [16]. Tato se však věnuje použití řídicího členu v úloze uchopení neznámého tělesa čtyřprstou robotickou rukou, nikoliv úloze chůze robotu. Detailní rozbor návrhu případně výčet nutných modifikací řídicího členu pro úlohu kráčivého robotu není v odborné literatuře zabývající se využitím architektury založené na DEDS popsán. Toto tvrzení dokládá i diplomová práce [6] popisující možný návrh řídicího členu kontaktu pro čtyřnohý robot operující na rovinném povrchu.

Na základě provedené rešerše bylo nutné vytvořit vlastní návrh řídicího členu kontaktu pro potřeby chůze robotu v neznámém členitém terénu. S uvážením informací uvedených ve výše citovaných zdrojích byly při návrhu respektovány následující zásady. Řídicí člen:

- se snaží docílit optimálního rozložení reakcí v došlapech
- při své činnosti mění konfiguraci těla robotu nebo konfiguraci zvolené nohy
- nemá k dispozici informaci o prostředí, noha může být použita k určení charakteru blízkého okolí došlapu, musí tedy být vybavena senzorem kontaktu
- potenciálová funkce musí mít pouze jedno lokální a zároveň globální minimum

Konkrétní návrh je uveden v kapitole 3.3.2.

2.3.1.3 Řídicí člen trajektorie

Tento řídicí člen byl poprvé detailně popsán v [8]. Zde se jedná o řídicí člen, který je schopen plánovat trajektorii manipulátoru s ohledem na jeho konfigurační prostor. Práce zahrnuje popis lokálního plánování, které zajišťuje nalezení cesty manipulátoru v bezprostřední blízkosti konkrétní překážky, a dále pak globální plánování trajektorie mezi různými překážkami vyskytujícími se na cestě. Jeho činnost je založena na využití harmonických funkcí, jejichž charakter zajišťuje, že gradientní spád určuje cestu k cíli. Řídicí člen trajektorie popsáný v uvedeném zdroji je použit v práci [15] a také [22], které se zabývají využitím popisované architektury řízení. Je nutné uvést, že obě práce ukazují výsledné chování robotu při obcházení překážky násobně větší než je použitý robot. Užití plánování lze označit spíše za globální, neboť v těchto publikacích není uveden příklad ukazující činnost řídicího členu v případě, že manipulovaná noha detekuje neznámou překážku. Řešení tohoto problému tak zůstává spíše v rovině teoretické.

V této práci je návrh řídicího členu trajektorie částečně zjednodušen s ohledem na stanovený rozsah práce, hlavní cíle a existující známá řešení v odborné literatuře. Do báze řízení je v této disertační práci začleněn řídicí člen, který je schopen zjednodušeného globálního plánování. Tento návrh je dále rozšiřitelný a připravený pro získávání informací o naplánované trajektorii ve známém či neznámém prostředí od jiného sofistikovaného algoritmu, například algoritmus RRT [21]. Naproti tomu důležité

a nezbytné reaktivní lokální vyhýbání se neznámým překážkám, které je v uvedených zdrojích zpracováno spíše teoreticky, je navrženo a zpracováno přímo do řídicího členu kontaktu i řídicího členu postoje. Popis vlastního návrhu řídicího členu trajektorie je uveden v kap. 3.3.3.

2.3.2 Vrstva zahrnující koordinační mechanismus DEDS

Jedná se o prostřední vrstvu popisované architektury řízení, která je vymezena dvěma hlavními cíli. První cílem je na základě formálního popisu jednotlivých členů báze řízení sestavit množinu složených řídicích členů, které jsou schopny generovat složitější chování robotu. Druhým cílem je pak sestavení koordinačního mechanismu, který zahrnuje všechny přípustné stavy robotu a pro každý z nich určuje množinu přípustných aktivací řídicích členů. Neposkytuje však informaci o tom, který z množiny přípustných zákonů řízení se má v daném stavu robotu aktivovat. Vedle těchto hlavních cílů poskytuje prostřední vrstva nezbytné oddělení spojitě činnosti senzorů a aktuátorů od diskrétních formálně popsanych událostí, se kterými pracuje nejvyšší vrstva. Dalším přínosem koordinačního mechanismu je pak snížení velikosti stavového prostoru řešené úlohy a snížení počtu možných aktivací v každém z těchto stavů, což výrazně zkracuje časové nároky algoritmu učení na nalezení vhodné posloupnosti aktivací řídicích členů. Tato tvrzení jsou podložena v následujících kapitolách detailnějším rozbořem, který je čerpán především z [15].

2.3.2.1 Složené řídicí členy.

Obdobně jako u systémů založených na chování, i zde řeší instance základních řídicích členů jednoduché, méně náročné úkoly, z nichž může být sestaveno řešení složitějších úloh. V mnoha případech založených na chování je ovšem v daný časový okamžik aktivní pouze jeden zpětnovazební člen řízení. Ten může omezovat účinnost jiných řídicích členů, což nutí konstruktéra přidávat nové členy, které se snaží docílit obou úkolů současně. Přičemž v případech založených na chování neexistuje univerzální předpis propojování elementů chování a složné řídicí členy se tak často stávají ad-hoc řešením pro danou úlohu [27].

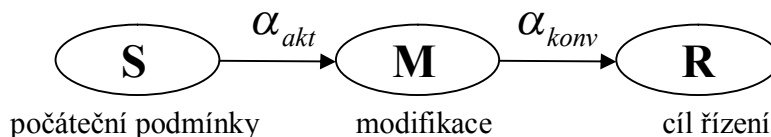
Naproti tomu, u zde popisované architektury řízení, je využíváno složených řídicích členů, které jsou tvořeny užitím hierarchické skladby pomocí operátoru podřízenosti, $<$. Tyto složené řídicí členy zajišťují složitější chování robotu skrze paralelní aktivace základních řídicích členů. Např. v zápisu $\Phi_f = \Phi_g < \Phi_h$ představuje Φ_g podřízený řídicí člen, který nesmí svojí činností ovlivnit činnost nadřazeného řídicího členu Φ_h . Množina řídicích členů, viz. rovnice (4), použitelných pro řešení dané úlohy může být tímto rozšířena o množinu složených řídicích členů:

$$C_m : \left(\dots < \left(\Phi_{i\tau_j}^{\sigma_j} < \Phi_{i\tau_l}^{\sigma_l} \right) \right) \quad (8)$$

kde C_m značí množinu složených řídicích členů, Φ_i značí i -tý člen množiny C a σ, τ jsou příslušné množiny vstupních a výstupních zdrojů.

Jednotlivé základní řídicí členy se obecně mohou snažit o dosažení vzájemně protichůdných cílů. Jejich propojení do složeného řídicího členu by pak způsobilo

nekonzistentní chování, při kterém by nebyl splněn ani jeden dílčí cíl řízení. Jelikož je každý základní řídicí člen charakterizován i svým formálním popisem, je možné zcela automaticky vyloučit nekonzistentní složené řídicí členy za použití predikátové logiky a u těch přípustných je možné vyvodit formální popis, opět zcela automaticky. Tímto popisem je myšlena trojice vektorů (S, M, R) , kde vektor S představuje počáteční podmínky, tj. stavy systému, ve kterých může být řídicí člen aktivován. Vektor M popisuje modifikaci stavu systému, tj. jakým způsobem se konfigurace může změnit. Vektor R reprezentuje cíl řízení, tj. o dosažení jakého stavu se řídicí člen snaží. Symbolický popis řídicího členu, základního či složeného, je zobrazen na Obr. 3, kde α_{akt} značí proces aktivace řídicího členu a α_{konv} značí proces konvergence k cíli daného zákona řízení. Pojem konvergence řídicího členu je použit pro označení procesu, během něhož se instance příslušného řídicího členu snaží dosáhnout svého cíle řízení.



Obr. 3: Symbolický popis řídicího členu, [15]

Složky jednotlivých vektorů (S, M, R) souvisí se způsobem, jakým je reprezentován stav systému, např. čtyřnohého robotu, při řešení dané úlohy, např. rotace robotu, [15]. Způsob reprezentace stavu systému při řešení dané úlohy popisuje následující kapitola.

2.3.2.2 Reprezentace stavu řízené soustavy

Pro využití formálních metod spojených s DEDS je nezbytně nutné odvodit model diskrétní části chování systému. Máme-li takovýto model k dispozici, pak může být teorie DEDS použita pro vymezení úloh a pro automatické vytvoření řídicího koordinačního mechanismu zadané oblasti úloh. Kvalita výsledného systému řízení a rozsah úloh, které je možné řešit, jsou tak závislé na struktuře a složitosti modelu systému [27].

V DEDS je diskrétní stavový prostor reprezentován funkčními cíli, kterých je dosahováno pomocí přípustné množiny akcí spojitého řízení. Tím vzniká ucelený model systému, který popisuje schopnosti robotu v daném prostředí pomocí výsledků chování. Vytvoření tohoto modelu je usnadněno formálními vlastnostmi spojitých řídicích členů a využitím báze řízení. Předložený přístup využívá řídicí členy po dobu procesu konvergence a množinu událostí v podobě aktivací a konvergence řídicích členů. Každý řídicí člen navíc na základě svého cíle rozděluje podřízený prostor fyzikálních stavů na množinu oblastí vlivu a množinu, kterou neovlivňuje. V této souvislosti může být na strategii řízení nahlíženo jako na posloupnost aktivací instancí řídicích členů, jež je charakteristická pro danou úlohu a jež vede systém robotu množinou vyhovujících rovnovážných stavů, přičemž přechody mezi nimi jsou určeny výsledky procesu konvergence. Toto umožňuje, aby bylo celkové chování systému popsáno pomocí DEDS, přičemž diskrétní stavový prostor je z velké části určen na základě atraktorů řídicích členů. Zmíněných vlastností je v této práci využito pro vytvoření modelu systému se

stavy reprezentovanými vektorem binárních hodnot (predikátů), p_i , které indikují, který z cílů řízení je či není splněn. Stavový vektor, $q = (p_0, p_1, \mathbf{K}, p_n)$, pak zcela charakterizuje stav, ve kterém se systém nachází.

Pro efektivnější práci s modelem všech možných chování robotu je možné při jeho sestavování zohlednit omezující podmínky. Tyto podmínky mají tvar logického výrazu, který obsahuje jednotlivé predikáty, p_i , použité ve stavovém vektoru, q . Je-li například stav systému popsán dvousložkovým vektorem, $q = (p_0, p_1)$, pak omezující podmínka může být např. stanovena ve tvaru $p_0 \vee p_1$, která určuje, že v každém stavu systému musí být pravdivý alespoň jeden z uvedených predikátů. Omezující podmínky mohou určovat buď bezpečnost, tj. určují stavy systému, které mohou vzniknout, ale řídicí strategie by se jich měly vyvarovat, a nebo stanovovat realizovatelnost, tj. určují stavy, které nemohou fyzicky vzniknout a proto se nemohou v uceleném modelu vyskytovat. Aplikace těchto omezujících podmínek omezuje jak počet stavů vyskytujících se v modelu, tak i počet přípustných aktivací zákonů řízení v těchto stavech. [15], [27].

2.3.2.3 Postup sestavení koordinačního mechanismu

V této kapitole bude uveden postup, který umožňuje automatické vyvození složených řídicích členů a poskytuje zjednodušení, která vedou k efektivnější činnosti nadřazené vrstvy popisované architektury. Jak bylo uvedeno v předchozí kapitole, stav systému je možné popsat pomocí stavového vektoru, q , jehož složky představují predikáty určující, který z parciálních cílů je či není splněn. Nyní je možné uvést i konkrétní hodnoty složek vektorů, (S, M, R) , viz. kap. 2.3.2.1, používaných pro popis řídicích členů.

Každý z těchto vektorů má tolik složek, kolik složek obsahuje stavový vektor, q , použitý v dané úloze. Hodnota i -té složky vektorů (S, M, R) se vztahuje k i -té složce vektoru, q . Složky nabývají hodnot z množiny $\{0, 1, *, -\}$. Znak 0 (resp. 1) značí nastavení příslušného predikátu na logickou nepravdu (resp. pravdu), znak * označuje libovolnou hodnotu predikátu a znak „-“ znamená, že příslušný predikát není dotčen. Mějme dvousložkový vektor, $q = \{p_0, p_1\}$, a instanci řídicího členu, c_i , jejíž specifikace jsou následující:

$$S(c_i) = (**), \quad M(c_i) = (*-), \quad R(c_i) = (1*). \quad (9)$$

Z těchto specifikací vyplývá, že daný zákon řízení může být použitý v libovolném stavu, svojí činností ovlivňuje hodnotu predikátu, p_0 a v žádném případě neovlivní hodnotu predikátu, p_1 . Zároveň se snaží o dosažení stavu systému v němž je predikát p_0 splněn.

Pro automatické sestavení modelu všech možných chování systému na základě právě uvedených specifikací základních řídicích členů je použito následujícího postupu popsaného nástroji formální logiky. Je-li stanovena počáteční množina, $C = \{c_0, c_1, \mathbf{K}, c_n\}$, základních řídicích členů, c_i , s odpovídajícími specifikacemi,

(S_i, M_i, R_i) , pak může být odvozena množina syntakticky vyhovujících složených řídicích členů, $C^\bullet = \{b_i\}$. Pro dodržení syntaktické správnosti je nutné, aby složené řídicí členy neobsahovaly jakékoliv nadbytečné instance základních řídicích členů. Pro množinu $C^\bullet$ tedy platí:

$$\begin{aligned} \varepsilon &\in C^\bullet \\ b_i &\in C^\bullet \Leftrightarrow b_i \in C^\bullet \\ b_i &\in C^\bullet \wedge b_j \in C^\bullet \wedge \neg \exists c_k [in(c_k, b_i) \wedge in(c_k, b_j)] \Leftrightarrow (b_i < b_j) \in C^\bullet \end{aligned} \quad (10)$$

kde ε značí prázdný řídicí člen, a $in(c_k, b_i)$ vyjadřuje, že řídicí člen c_k je součástí složeného řídicího členu b_i . Událost aktivace, α_{akt} , jejíž výskyt můžeme řídit a výsledek procesu konvergence, α_{konv} , jež nejsme schopni ovlivnit, jsou spojeny s každým prvkem této množiny, $C^\bullet$, a poskytují množiny událostí pro model DEDS. Pro prvotní snížení velikosti těchto množin je vhodné vyřadit ty složené řídicí členy, b_i , které byly vytvořeny ze základních řídicích členů majících vzájemně protichůdné cíle řízení. Takovéto nevyhovující složené řídicí členy lze popsat následovně :

$$\begin{aligned} Contra(b) \Leftrightarrow \neg \exists c_j, c_k [in(c_j, b) \wedge in(c_k, b) \wedge \exists l [(S_l(c_j) = 1) \wedge (S_l(c_k) = 0)) \vee \\ ((M_l(c_j) = 1) \wedge (M_l(c_k) = 0)) \vee \\ ((R_l(c_j) = 1) \wedge (R_l(c_k) = 0))] \end{aligned} \quad (11)$$

kde $R_l(c_j)$ označuje hodnotu l -té složky výstupního vektoru, R , řídicího členu, c_j . Obdobně platí i pro vektory M, S . Vyřazením takovýchto nevyhovujících řídicích členů dostaneme množinu všech přípustných řídicích členů, C^0 :

$$C^0 = \{b_i \mid b_i \in C^\bullet \wedge Contra(b_i)\} \quad (12)$$

Dále je nutné pro každý složený řídicí člen vyhodnotit vzájemný vliv všech základních řídicích členů, z nichž je vytvořen. Jelikož pro vytvoření složených řídicích členů bylo použito operátoru podřízenosti, $<$, může být řídicí člen, c , ovlivněn pouze některým z nadřízených elementárních řídicích členů. Vyhodnocení vlivu se tak omezuje na složku vyšší úrovně, $H(c, b)$. Tato nadřízená složka řízení řídicího členu, b , může být vzhledem k c určena takto:

$$H(c, c) = \varepsilon \quad (13)$$

$$H(c, b_i < b_j) = \begin{cases} H(c, b_i) < b_j & \text{když } in(c, b_i) \\ H(c, b_j) & \text{když } in(c, b_j) \end{cases} \quad (14)$$

kde ε značí prázdný řídicí člen jenž vyjadřuje, že neexistuje žádná nadřízená složka řízení. Důležitost této složky nadřízené řídicímu členu, c , spočívá v situaci, kdy se predikáty výstupních vektorů, R , těchto řídicích členů shodují. Jelikož pouze řídicí členy obsažené v této složce jsou pak schopné měnit hodnoty těchto predikátů. Možný vliv aktivace instance řídicího členu, b , na stav systému reprezentovaného stavovým vektorem, q , může být proto vyjádřen takto:

$$Conv(c) \Leftrightarrow \forall j[(R_j(c) = *) \vee (R_j(c) = q_j)] \quad (15)$$

$$Fix_i(b) \Leftrightarrow \exists c[in(c, b) \wedge Conv(c) \wedge (R_i(c) \neq *) \wedge \neg \exists j[(R_j(c) \neq *) \wedge Change_j(H(c, b))]] \quad (16)$$

$$\neg Change_i(\varepsilon) \quad (17)$$

$$Change_i(b) \Leftrightarrow \exists c[in(c, b) \wedge \neg Conv(c) \wedge (M_i(c) \neq -) \wedge \neg Fix_i(H(c, b))] \vee \exists c[in(c, b) \wedge (M_i(c) \neq -) \wedge \exists j[(R_j(c) \neq *) \wedge Change_j(H(c, b))]] \quad (18)$$

kde $Conv(b)$ je splněno pokud aktuální stav reprezentovaný stavovým vektorem, q , odpovídá dosažení cíle řídicího členu b . Výraz $Fix(b)$ je vektor pravdivostních ohodnocení, jenž určuje, které z predikátů odpovídají očekávaným výstupům řídicího členu po ukončení procesu jeho konvergence, a při tom již nemohou být změněny žádným z nadřazených řídicích členů. Výraz $Change(b)$ určuje, které z predikátů je možné změnit aktivací složeného řídicího členu. To odpovídá situaci, kdy se řídicí členy, u kterých ještě nebyl ukončen proces konvergence, pokoušejí o změnu predikátů a žádný z nadřazených řídicích členů se nepokouší tyto predikáty zachovat nezměněné. $Change(b)$ proto představuje všechny možné změny stavu, které může aktivace řídicího členu b vytvořit, jestliže se systém nachází ve stavu reprezentovaném stavovým vektorem, q . Množina možných následných stavů, $Succ(q, b)$, vzniklých ze stavového vektoru, q , aktivací řídicího členu, b , může být proto odvozena jako:

$$y \in Succ(q, b) \Leftrightarrow \forall j[y_j = q_j \vee Change_j(b)] \quad (19)$$

kde j označuje j -étou složku odpovídajícího stavového vektoru. Jestliže množinu, $Succ(q, b)$, sestavíme postupně pro kombinace všech přípustných stavů systému se všemi akcemi řízení, získáme po přidání přechodů mezi stavy celý abstraktní model systému. Popsaný postup byl čerpán z [14] kap. 4.2.2. a kap. 4.3, [15] kap. 4.1.2 a [27].

Uvedený postup sestavení koordinačního mechanismu byl implementován pomocí zvolených programovacích nástrojů a tvoří součást předloženého řešení. Konkrétní výsledky jsou uvedeny kapitole 3.4.

2.3.3 Vrstva učení

Nejvyšší vrstva popisované architektury řízení zahrnuje vybraný algoritmus učení. Na vstupu této vrstvy je seznam přípustných stavů systému. A pro každý z těchto stavů je k dispozici množina zákonů řízení, které mohou být v daném stavu aktivovány. Způsob, kterým byl tento ucelený model všech možných chování získán, je popsán v kapitole 2.3.2. Cílem této vrstvy je nalezení optimální aktivace v každém ze stavů systému tak, aby byl splněn stanovený cíl řízení, který je popsán stavovým vektorem q , a to pomocí co nejmenšího počtu aktivací dostupných zákonů řízení. Rovněž lze říci, že cílem zvoleného algoritmu učení je nalezení vhodné posloupnosti aktivací jednotlivých instancí řídicích členů, které povedou systém přes množinu přípustných stavů až po stav reprezentující splnění cíle řízení.

V [22] lze nalézt způsob plánování chůze robotu založený na bázi řízení. Autor ovšem sestavuje různé strategie řízením manuálně nikoliv automaticky pomocí učení. Určitá vylepšení popisované architektury řízení směřující k postihu dynamických vlivů na řízenou soustavu jsou uvedeny v [10]. Ani zde ovšem není zahrnuta problematika užití algoritmu učení. Naproti tomu v [15] je popsáno sestavení hybridní jednotky řízení robotu založené na DEDS včetně využití algoritmu Q-učení v rámci nejvyšší vrstvy. V publikaci jsou uvedeny výsledky využití algoritmu Q-učení pro úlohu rotace robotu a úlohu dopředné chůze na vodorovné rovině bez překážek. Teoretickému rozboru problematiky učení se komplexně věnuje kapitola 2.4, nyní proto následuje pouze výčet některých důležitých výsledků souvisejících s implementací nejvyšší vrstvy učení.

V uvedené publikaci autor popisuje možný způsob implementace algoritmu online Q-učení. Svůj úspěch dokládá výsledky pro úlohu rotace a pro úlohu chůze robotu. V první úloze tvoří ucelený model vyprodukovaný prostřední vrstvou celkem 16 různých stavů a 157 přípustných zákonů řízení. Po přibližně 550 různých aktivacích, jejichž provedení trvalo přibližně 11 minut, byl učící algoritmus schopen najít strategii řízení blízkou strategii optimální. Volba alternativních aktivací zákonů řízení v průběhu učení probíhala pomocí hladového algoritmu. Druhá úloha byla řešena pouze simulačně, přičemž podrobnosti způsobu simulace nejsou uvedeny. U tohoto experimentu obsahoval výchozí model celkem 72 různých stavů a 257 možných aktivací zákonů řízení. Jako strategie prozkoumávání alternativních aktivací byla použita metoda *softmax*. Autor nezmiňuje délku trvání procesu učení. V závěru pak autor uvádí, že byly nalezeny celkem dvě slibné strategie, které vedly jednak na cyklickou a jednak na acyklickou, tím tedy časově náročnější, chůzi.

S ohledem na řešeršní studii a rozbor vybraných publikací lze konstatovat, že využití opakovaně posilovaného učení v kombinaci s architekturou řízení založenou na paradigmatu DEDS nebylo pro úlohu chůze robotu v nerovném terénu s neznámými překážkami doposud detailně publikováno.

2.3.4 Shrnutí

Architektura řízení pospaná v této kapitole spadá do skupiny hybridních přístupů návrhu řídicí jednotky autonomních robotů. Některé části tohoto návrhu jsou typické pro přístup založený na chování, např. sestavení elementárních chování robotu, vzájemné propojování těchto chování, umožnění okamžitého reaktivního chování při změně prostředí. Jiné části jsou pak charakteristické spíše pro přístupy založené na modelu, např. základní řídicí členy mohou zahrnovat částečný model robotu, nepřípustná chování robotu lze předem vyloučit, je možné použít cyklické styly chůze, apod. Spojení obou přístupů pak přináší i tyto výhody:

- není nutné vytvářet komplexní model robotu a prostředí
- změna úlohy (cíle) nevyžaduje velký zásah do návrhu jednotky řízení
- je umožněno reaktivní chování robotu
- změnou parametrů učení lze dosáhnout různých stylů chůze

Protože architektura řízení založená na paradigmatu DEDS umožňuje použít opakovaně posilované učení pro řízení čtyřnohého robotu, bude tento přístup s ohledem

na dosavadní výsledky prezentované v rámci státní doktorské zkoušky, použit v této disertační práci. Hlavní problémy, které je nutné vyřešit při využití této architektury, tak jak je chápe autor této práce, jsou:

- korektní návrh řídicího členu kontaktu schopného činnosti ve strukturovaném prostředí
- zajištění reaktivního chování na neznámé překážky v bezprostřední blízkosti robotu
- nalezení parametrů Q-učení umožňujících robotu pohyb v členitém terénu

Následující kapitola je věnována problematice strojového učení, zejména opakovaně posilovanému učení a některým specifickým aspektům implementace do právě popsané architektury řízení.

2.4 Strojové učení

Pojem učení je velmi obsáhlý a zahrnuje širokou škálu procesů. Odborné publikace z různých vědeckých oborů uvádějí více či méně precizní definice tohoto pojmu. Často obsahují pojmy jako získávání znalostí, dovedností či zkušeností, porozumění apod. Biologové se zabývají učením u zvířat, psychologové u lidí, technické obory se zabývají učením strojů. Mezi těmito obory existuje velké množství souvislostí. Metody strojového učení jsou často odvozeny díky znalostem právě z oboru biologie či filozofie. Dle [47] lze velmi obecně říci, že stroj se učí kdykoliv, kdy provede změnu své struktury, programu či dat, se kterými pracuje, za účelem zlepšení svého výkonu.

V následující kapitole, 2.4.1, je uveden stručný přehled vybraných oblastí strojového učení. Dále je prezentován popis opakovaně posilovaného učení, kap. 2.4.2, neboť algoritmus Q-učení, viz. kap. 2.4.3, spadající do této oblasti je použit pro nejvyšší vrstvu popsané architektury řízení. Rozbor algoritmu Q-učení včetně vybraných aspektů implementace pro potřeby nalezení vhodné strategie řízení reprezentované diskrétní posloupností aktivací zákonů řízení je součástí kap.2.4.4.

2.4.1 Vybrané algoritmy strojového učení

Pojem strojové učení zahrnuje širokou škálu řešení různých problémů, jak bylo naznačeno výše. Používané metody řešení tak lze klasifikovat do kategorií dle různých kritérií. Například je možné použít kritérium vynaloženého úsilí na získání nových znalostí během procesu učení:

- učení zapamatováním
- učení se z instrukcí
- učení se z analogie
- učení na základě vysvětlení
- učení se z příkladů
- učení se pozorováním

Jiným kritériem může být způsob jakým stroj zjišťuje, že se učí správně:

- učení s učitelem
- učení bez učitele
- učení se napodobováním
- posilované učení

Koncept učení se s učitelem zahrnuje příklady rozčleněné do tříd, které se má systém naučit. V případě učení bez učitele nemá systém k dispozici žádné doplňující informace. U učení napodobováním jsou k dispozici pouze nepřímé náznaky a konečně u posilovaného učení je systém odměněn za správné chování a penalizován za nesprávné chování. Nyní je možné uvést vybrané metody, které spadají do oblasti strojového učení. Jedná se zejména o metody:

- prohledávání prostoru verzí
- rozhodovacích stromů
- rozhodovacích pravidel
- konceptuálního shlukování
- posilovaného učení
- učení založené na vysvětlování
- usuzování na základě analogií
- bayesovského učení
- fuzzy učení
- evolučních technik
- učení neuronových sítí

Metody prohledávání prostoru verzí mají k dispozici množinu všech popisů konceptů, které jsou konzistentní s trénovacími daty. Používají při své činnosti pozitivní i negativní příklady cílového konceptu, přičemž zobecnění je možné pouze z pozitivních příkladů. Do této oblasti spadají např. algoritmus generalizace, specializace, eliminace kandidátů apod.

Techniky rozhodovacích stromů vytváří stromy, ve kterých jsou listy ohodnoceny indexy tříd a ostatní uzly reprezentují atributy, hrany těchto uzlů jsou ohodnoceny hodnotami daného atributu. Mezi metody rozhodovacích stromů patří např. algoritmus ID3 nebo TDITD. Tyto stromy mají následně význam pro tvorbu pravidel, kdy každé cestě z kořene do listu odpovídá jedno pravidlo. Pro vyvozování těchto pravidel je následně používáno technik rozhodovacích pravidel, např. algoritmus AQ nebo CN4.

Konceptuální shlukování představuje iterativní proces výběru jader ze stávajících protokonceptů a vytváření nové množiny protokonceptů tak dlouho, dokud není splněno dané ukončovací kritérium. Příkladem může být procedura Cobweb [34].

U učení založeného na vysvětlování se nejprve vytvoří vysvětlení trénovacího příkladu, obvykle ve formě důkazu. Následně je zobecněním tohoto vysvětlení získána

definice cílového konceptu. Na základě zobecněného důkazního stromu se po té definuje pravidlo, jehož závěr je kořen stromu a předpoklad je konjunkcí listů.

Hlavní motivací metod založených na analogii je učení nových konceptů nebo odvozování nových řešení prostřednictvím použití podobných konceptů a jejich řešení. Do této oblasti spadají i tyto metody: případové usuzování, pravidlo nejbližšího souseda, líné učení aj. V těchto metodách jsou znalosti reprezentovány v podobě báze již vyřešených problémů. Při usuzování se k danému problému v této bázi hledá nejpodobnější případ a jeho řešení se adoptuje na novou situaci.

Uvedené informace byly čerpány především z [46], [47] a [25]. Podrobnější rozbor jednotlivých přístupů přesahuje stanovený rozsah této práce.

2.4.2 Posilované učení

Posilované učení je takový způsob učení, ve kterém je používána interakce soustavy přímo s prostředím. Algoritmus posilovaného učení není explicitně vyučován, učí se totiž z následků svých akcí řízení. Jednotlivé akce řízení volí jednak na základě svých dosavadních zkušeností, stejně tak ovšem provádí průzkum zcela nových voleb řízení metodou pokus omyl. Po provedení zvolené akce obdrží algoritmus číselnou odměnu, která vypovídá o úspěšnosti výsledku této akce. Algoritmus se snaží najít takovou posloupnost akcí, která vede v průběhu času k maximalizaci této odměny.

Nejlepších výsledků je dosahováno v případech, kdy lze řešený problém klasifikovat jako Markovský rozhodovací problém. Algoritmus může prohledat konečný počet stavů a v každém z nich obdrží odměnu, resp. pokutu v případě záporné hodnoty. Z každého stavu může být použitím akce (z dané množiny akcí) dosažena celá množina následujících stavů. Každý stav navíc nese vlastní hodnotu, která může být v čase měněna. Tato hodnota je dána průměrem všech možných budoucích odměn za aktivaci jednotlivých akcí v tomto stavu. Způsob výběru akcí v každém ze stavů je dán strategií řízení. Tato strategie může být rovněž měněna v průběhu času.

Metody posilovaného učení jsou obecně využívány pro řešení dvou hlavních problémů:

- problém predikce: popsané metody je možné použít pro nalezení hodnotící funkce. Tato funkce určuje, po dokončení činnosti metody, jak velkou odměnu získáme užitím dané akce v konkrétním stavu a to pro všechny možné kombinace dvojic stav-akce.
- problém řízení: cílem je nalezení takové strategie řízení, která vede k maximalizaci zisku při procházení stavového prostoru. Po ukončení činnosti metody lze získat optimální strategii, která umožňuje dosažení optimálního řízení skrze plánování akcí. Řešení tohoto problému vyžaduje řešení i problému predikce.

Mějme přechodovou funkci, $P(s, a, s')$, která určuje pravděpodobnost přechodu ze stavu s do stavu s' při užití akce a . Dále mějme odměňovací funkci, $O(s, a)$, která určuje odměnu získanou za aktivaci akce a ve stavu s . Metody, které umožňují oba

zmíněné problémy řešit bez znalosti přechodové a odměňovací funkce spadají do oblasti posilovaného učení. Mezi ty nejvýznamnější patří:

- *TD*: z angl. temporal difference, iterativní metoda umožňující nalezení optimální hodnotící funkce, [44],
- *Actor-critics*: rovněž iterativní metoda, užívaná převážně pro nalezení optimální strategie řízení,
- Q-učení - algoritmus schopný naučit se optimální hodnotící funkci i strategii řízení současně.

Algoritmus Q-učení bude použit v předkládaném řešení, proto je mu věnována samostatná kapitola 2.4.3. Informace uvedené v této kapitole byly čerpány z [51] a [20].

2.4.3 Q-učení

Algoritmus Q-učení se řadí do oblasti opakovaně posilovaného strojového učení a poprvé byl popsán ve [43]. Je tedy určen pro řešení úloh, které lze klasifikovat jako Markovský rozhodovací problém. Narozdíl od jiných metod je Q-učení schopné paralelního hledání optimální hodnotící funkce, která určuje ohodnocení pro každou dvojici stav-akce, a optimální strategie řízení, která určuje posloupnost akcí, které vedou k maximální odměně v průběhu času. Z charakteru opakovaně posilovaného učení vyplývá, že oba problémy je algoritmus schopen určit bez vnějšího zásahu učitele a navíc bez znalosti modelu řešené soustavy, v tomto případě čtyřnohého robotu. Učení proto probíhá pouze na základě přímého vlivu jednotlivých akcí na stav systému.

Jak bylo uvedeno, algoritmus hledá optimální hodnotící funkci, která každé dvojici stav-akce přiřadí číselnou hodnotu. Jedná se o tzv. Q-hodnotu, $Q(x_t, a_t)$, která vyjadřuje, jak moc perspektivní je volba akce řízení, a , ve stavu soustavy, s , v časovém kroku, t . Optimální hodnota může být dle Bellmanovy rovnice optimality spočtena následovně:

$$Q^*(x_t, a_t) = r_{t+1} + \gamma \max_{b \in A} Q^*(x_{t+1}, b) \quad (20)$$

kde Q^* je optimální Q-hodnota příslušící volbě akce řízení, a_t , ve stavu soustavy, x_t , v časovém kroku, t . Hodnota r_{t+1} určuje okamžitou odměnu soustavy za volbu akce řízení a_t ve stavu x_t . Výsledný stav soustavy po provedení této akce je x_{t+1} . Množina, A , představuje všechny přípustné akce řízení. Diskontní (srážkový) faktor, γ , pro který platí $0 < \gamma \leq 1$, je používán pro snížení vlivu vzdálených odměn na aktuální Q-hodnotu. Rovnice (20) určující výpočet optimální hodnotící funkce může být aproximována iterativním výpočtem takto:

$$Q_t(x_{t-1}, a_{t-1}) = (1 - \alpha) Q_{t-1}(x_{t-1}, a_{t-1}) + \alpha \left(r_t + \gamma \max_{b \in A} Q_{t-1}(x_t, b) \right) \quad (21)$$

kde α je parametr učení, určující vliv právě získané okamžité odměny za provedení akce na aktualizovanou Q-hodnotu příslušné dvojice stav-akce, $Q_t(x_{t-1}, a_{t-1})$. Tento iterativní způsob výpočtu již umožňuje využití algoritmu Q-učení pro nalezení odhadu optimální hodnotící funkce. Pro dosažení optima je nezbytné, aby byla Q-hodnota aktualizována

pro každou dvojici stav-akce nekonečně mnohokrát. Aktualizace je možné provádět pomocí offline nebo online učení. Při offline učení není uplatňována žádná strategie řízení, dochází pouze k opakovaným aktivacím zákonů řízení ve všech možných stavech za účelem nalezení výchozích Q hodnot. Offline fáze se nejčastěji provádí pomocí modelu soustavy z důvodu nižší časové náročnosti. Naproti tomu online učení již probíhá přímo na základě interakce řízené soustavy s prostředím. Q hodnoty jsou tak aktualizovány pouze ve stavech, ve kterých se řízená soustava právě nachází. Z výše popsaného přístupu plyne, že algoritmus musí vždy vhodným způsobem vyvážit dvě základní situace:

- akce jsou voleny pouze dle naučené strategie řízení, tj. nedochází k hledání lepší strategie
- akce jsou voleny náhodně, čímž dochází k prohledávání alternativních, potenciálně lepších strategií řízení

Z nalezené hodnotící funkce, která je reprezentována společnými Q hodnotami, je možné určit strategii řízení soustavy, která se bude blížit optimální strategii, a to tak, že v každém stavu je volena akce řízení $a_{\max}$:

$$a_{\max} = \max_{a \in A} Q(x, a) \quad (22)$$

Při snaze využít rovnici (22) pro nalezení optimální strategie řízení kráčivého robotu je potřeba uvážit a vyřešit několik kritických problémů. Tyto aspekty jsou popsány v následující kapitole.

2.4.4 Q-učení v kontextu řízení kráčivého robotu

Při využití algoritmu Q-učení pro plánování pohybu kráčivého robotu je nutné uvážit následující omezení:

- během učení nemůže být robot poškozen ani vyžadovat akci učitele
- využití pouze online učení povede na velkou časovou náročnost algoritmu
- rozšíření o offline předučení vyžaduje komplexní model robotu a prostředí

Přímé užití online Q-učení je časově velmi náročné pro tak komplexní roboty, jako jsou čtyřnohé kráčivé roboty, viz kap. 3.2 Platforma použitá pro verifikaci návrhu řešení. Online Q-učení by navíc mohlo vést k pádu robotu, popř. k jeho poškození, pokud nebudou stanoveny omezující bezpečnostní podmínky, tj. nebude apriori určeno, které akce řízení nesmí být v dané konfiguraci robotu použity. Tento požadavek je ovšem velmi těžké splnit, zvláště pokud má robot vykonávat svoji činnost v neznámém strukturovaném prostředí. Konstruktor by nemusel být schopen zapracovat všechna omezení předem. Problém by bylo možné vyřešit využitím offline Q-učení za použití matematického modelu robotu. Sestavení komplexního modelu zahrnujícího všechny nezbytné aspekty rovněž není triviální záležitost.

Uvedeným problémům lze předejít užitím architektury řízení popsané v kap. 2.3. Prostřední vrstva uvedené architektury reprezentovaná modelem DEDS, poskytuje potřebnou abstrakci spojitě činnosti akčních členů a senzorů robotu od diskrétní činnosti algoritmu učení, který pracuje s dvojicemi stav-akce. Vrstva rovněž zahrnuje omezující podmínky, které určují nepřípustné stavy robotu. Tím je výrazně omezena velikost

stavového prostoru. Pomocí formálního popisu vlivu aktivace na stav robotu, je navíc možné omezit i počet přípustných aktivací v každém ze stavů. Tímto je dále snížena složitost modelu reprezentujícího všechny možné stavy robotu a přechody mezi nimi, což vede i ke snížení času potřebného pro nalezení vhodné strategie řízení algoritmem Q-učení.

Vedle těchto zásadních výhod je nutné zmínit i dvě výchozí nevýhody, které jsou však odstranitelné. Rovnice určující způsob výpočtu nové Q hodnoty nejsou příliš vhodné pro použití v nejvyšší vrstvě popsané architektury řízení. Dle rovnice (21) jsou totiž aktualizace prováděny v každém časovém kroku, t . Naproti tomu v architektuře využívající model DEDS je klíčový výskyt událostí aktivace a konvergence používaných zákonů řízení. Přičemž časová náročnost konvergence různých zákonů řízení, resp. instancí řídicích členů, je obecně různá. S uvažováním výše uvedeného je vhodné modifikovat rovnici (21) následovně, [15]:

$$Q_k(\mathbf{p}_{k-1}, c_{k-1}) = (1-\alpha)Q_{k-1}(\mathbf{p}_{k-1}, c_{k-1}) + \alpha \left(r_k + \gamma \max_{c \in C^0} Q_{k-1}(\mathbf{p}_k, c) \right) \quad (23)$$

kde k je diskretní časový okamžik odpovídající události konvergence ($k-1$) aktivace řídicího členu, c_{k-1} . Dále r_k představuje odměnu získanou během činnosti tohoto řídicího členu a $\mathbf{p}_k$ je predikátový vektor reprezentující stav robotu po ukončení konvergence tohoto členu. Množina C^0 je tvořena všemi přípustnými řídicími členy, viz rovnice (12). Tato formulace procesu iterace výrazně snižuje časovou náročnost díky jednodušší struktuře výsledné strategie řízení a nižšímu počtu okamžiků, kdy je prováděn výpočet.

Druhou nevýhodou, která přetrvává i při použití modelu DEDS, je nemožnost užití tohoto modelu pro offline učení. Důvodem je nedeterminismus obsažený v tomto modelu všech možných přechodů, který je způsobený tím, že aktivace řídicího členu v daném stavu může potenciálně ovlivnit celé množství predikátů obsažených ve stavovém vektoru, q . Tato informace je dána vektorem (S, M, R) , viz. kap. 2.3.2.1. Dále je výsledek procesu konvergence závislý i na kinematických omezeních (např. dočasné omezení pracovního prostoru nohy), vzájemné interakci řídicích členů a prostředí. Tyto vlivy nemohou být zahrnuty přímo do koordinačního mechanismu DEDS, neboť ten je nezávislý na konkrétní platformě robotu či prostředí. V případě potřeby je ovšem možné dodatečně vyřešit i tento problém a to nalezením pravděpodobnosti přechodu, P :

$$P(y = x_{k+1} | c, x_k) \quad (24)$$

ze stavu x_k , do stavu x_{k+1} , při aktivaci řídicího členu c_k , v diskretním časovém okamžiku k . V nejjednodušším případě je možné tyto pravděpodobnosti spočítat po ukončení procesu učení dané úlohy z počtu výskytů přechodů:

$$P(x_{k+1} | c, x_k) = \frac{n(x_k, c, x_{k+1})}{n(x_k, c)} \quad (25)$$

kde čitatel značí počet výskytů přechodů ze stavu x_k do stavu x_{k+1} při aktivaci řídicího členu c_k a jmenovatel značí celkový počet aktivací řídicího členu c_k ve stavu x_k . Spočtené pravděpodobnosti mohou být následně začleněny do modelu všech možných

přechodů pro odstranění zmíněného nedeterminismu. Koordinační mechanismus DEDS pak může být použit nadřazenou vrstvou implementující algoritmus strojového učení pro potřeby offline učení a tím pro snížení časové náročnosti nalezení optimální strategie řízení pro další úlohy. Vypočtené pravděpodobnosti jsou totiž závislé na platformě robotu a na prostředí, ve kterém je provozován, nikoliv na charakteru úlohy.

2.5 Zvolený přístup pro dosažení stanovených cílů

V následující kapitole je uvedeno souhrnné zhodnocení teoretických východisek, která jsou využita v předkládaném řešení s důrazem na dosud nepublikovanou problematiku. V kapitole 2.5.2 je stručně shrnut zvolený přístup k řešení stanovených cílů a rovněž jsou zde vyjmenovány stěžejní problémy, které je nutné řešit. Navazující kapitola 2.5.3 se věnuje postupu, který bude uplatněn při řešení.

2.5.1 Kritické zhodnocení teoretických východisek

Rešeršní studie, která byla předložena v rámci státní doktorské zkoušky, ukázala různé přístupy k plánování chůze robotů. Výhody a nevýhody vybraných přístupů jsou uvedeny v kap. 2.2. Jako nejperspektivnější se jeví skupina hybridních přístupů, která přináší řadu výhod kombinací přístupů založených na modelu a na chování, zatímco nevýhody jsou potlačovány. Velice slibně se jeví vícevrstvá architektura řízení robotu založená na paradigmatu DEDS, teoretický rozbor je uveden v kap. 2.3, která je označována právě jako hybridní. Mezi výhody lze uvést např.:

- není nutné vytvářet komplexní model robotu a prostředí
- metodika propojování elementů chování je jednoznačně dána
- dodatečné přidávání elementů chování nevyžaduje změnu dosavadního návrhu
- změna úlohy (cíle) nevyžaduje zásah do návrhu jednotky řízení
- je umožněno reaktivní chování robotu
- změnou parametrů učení lze dosáhnout různých stylů chůze

Jak bylo uvedeno, viz. kap. 2.3.4, existuje několik prací, které se zabývají využitím této architektury pro potřeby plánování chůze. Přístup je založen na malé množině zpětnovazebných řídicích členů zajišťujících elementární chování robotu. A právě sestavení této množiny má výrazný vliv na výsledné schopnosti robotu. Příklady těchto řídicích členů jsou prezentovány v kap. 2.3.1. Ve zdrojích zabývajících se architekturou DEDS, které jsou citovány ve zmíněné kapitole, je zejména řídicí člen kontaktu rozebrán velmi vágně a pro jeho detailnější rozbor je odkazováno na publikaci řešící problém uchopení předmětu pomocí čtyřprsté mechanické ruky, detailně viz. kap. 2.3.1.2. Řešení řídicího členu kontaktu, který by mohl být použit ve zmíněné architektuře pro platformu kráčivého robotu operujícího v členitém neznámém prostředí, tak není publikováno. Pro ověření návrhu je především nutné publikovat výslednou potenciálovou plochu řídicího členu pro různé výchozí konfigurace robotu a provést experimenty pro různé úlohy.

Nadřazená prostřední vrstva zahrnující koordinační mechanismus DEDS je detailně popsána v kap. 2.3.2. Všechny aspekty návrhu a možností automatického sestavení uceleného modelu všech možných chování robotu jsou v odborné literatuře

rozebrány detailně. Chybí ovšem uvedení konkrétních specifikací všech řídicích členů použitých pro dopřednou chůzi.

S uvedeným nedostatkem nejnižší vrstvy dále souvisí i absence některých výsledků strojového učení implementovaného v nejvyšší vrstvě architektury, viz. kap. 2.3.3. Modifikace algoritmu Q-učení potřebné pro implementaci v nejvyšší vrstvě jsou popsány a detailně rozebrány, viz. kap. 2.4.4. Naučené styly kráčení pro úlohu chůze v longitudinálním směru a rotace těla robotu změnou úhlu bočení v rovinném prostředí jsou však popsány pouze slovně, resp. s použitím obrázků. Autor této práce se domnívá, že by bylo vhodné použít i popisu kvantitativního a to standardizovaným způsobem uvedeným v kap. 2.1.3. Schopnost algoritmu Q-učení není rovněž zhodnocena pro případ, kdy má robot provádět naučenou chůzi v jiném prostředí, tj. není ověřena adaptivnost řídicí jednotky. Naproti tomu potřebné modifikace algoritmu Q-učení potřebné pro implementaci v nejvyšší vrstvě jsou v rámci odborného diskurzu detailně rozebrány.

2.5.2 Zvolený přístup a přehled dílčích cílů

Pro potřebu plánování chůze robotu bude v této disertační práci využito algoritmu Q-učení náležitě do oblasti opakovaně posilovaného učení. Tento cíl bude realizován za použití komplexní architektury řízení založené na paradigmatu DEDS. Posouzení kvality navrženého řešení bude provedeno na třech základních úlohách:

- rotace robotu změnou úhlu bočení
- chůze robotu v longitudinálním směru
- chůze robotu po nakloněné rovině

Žádná část řídicí architektury by neměla mít k dispozici informaci o charakteru prostředí ve kterém se robot nachází. Jako testovací platforma poslouží čtyřnohý kráčivý robot se třemi stupni volnosti na každé noze, viz. kap. 3.2.1. Navržené řešení bude ovšem dodržovat maximální obecnost a přenositelnost na jiný typ robotu a pro plnění jiných než stanovených úloh. Experimentální ověření bude probíhat pomocí software simulace, viz. kap. 3.4, a modelu skutečného robotu, viz. kap. 3.2.3.

S uvážením výše uvedených informací lze vyjmenovat stěžejní problémy, jejichž řešení nebylo doposud dostatečně publikováno v souvislosti s architekturou DEDS a jejichž vyřešení lze pokládat za přínosné:

- korektní návrh řídicího členu kontaktu schopného činnosti ve strukturovaném prostředí
- uvedení průběhu funkcí použitých řídicích členů pro různé konfigurace robotu
- zajištění reaktivního chování na neznámé překážky v bezprostřední blízkosti robotu
- nalezení parametrů Q-učení, které zajistí naučení chůze robotu na jiném než vodorovném povrchu a posouzení schopnosti přizpůsobovat se
- vyhodnocení naučených stylů chůze pomocí standardizovaných ukazatelů

2.5.3 Postup řešení

Při řešení stanovených cílů bude postupováno následovně. Nejprve je nutné zvolit množinu základních řídicích členů a najít vhodný předpis jejich potenciálové funkce. Pro prvotní zhodnocení správnosti návrhu je zapotřebí posoudit a vykreslit úplný průběh těchto funkcí pro různé výchozí konfigurace robotu.

Dále je nutné implementovat návrh základních řídicích členů ve vhodném simulačním prostředí pro zvolený čtyřnohý robot. Z těchto řídicích členů lze pomocí vhodné algoritmizace sestavit množinu složených řídicích členů a vyvodit automaticky jejich specifikace. Chování robotu generovaná základními i složenými řídicími členy by měla být simulačně testována. Toto vyžaduje vytvoření modelu robotu ve zvoleném simulačním prostředí. V dalším kroku lze následně sestavit model všech možných chování robotu, reprezentovaný nedeterministickým stavovým automatem.

V následující fázi řešení budou hledány optimální strategie řízení pomocí algoritmu Q-učení. Pro tento účel je nutné vytvořit mechanismus, pomocí něhož budou ukládány a načítány jednotlivé Q hodnoty. Dále je zapotřebí vhodným způsobem reprezentovat nalezené optimální strategie řízení pro dané úlohy. V poslední části řešení je totiž nezbytné zrealizovat monitorovací jednotku, která bude schopná měřit a vyhodnocovat jednotlivé parametry chůze tak, aby bylo možné provést kvalitativní i kvantitativní porovnání těchto stylů.

Jednotlivé kroky řešení, stejně tak jako dosažené výsledky, jsou detailně diskutovány v praktické části této práce, viz. kap. 3.

3 Praktická část – souhrn dosažených výsledků

3.1 Volba prostředků pro verifikaci

Pro implementaci předkládaného řešení je nezbytně nutné zvolit vhodné programové vybavení, které by umožnilo realizovat jednotlivé kroky popsané v kap. 2.5.3. V první fázi řešení je přistoupeno k návrhu elementárních řídicích členů. Zde musí být nalezen vhodný předpis potenciálové funkce a spočteny její hodnoty pro různé výchozí konfigurace robotu. Kromě určení oboru hodnot funkce a její vykreslení je nutné sestavit základní výpočet konfigurace robotu založený na geometrickém popisu zvoleného robotu. Pro tyto účely lze použít výpočtový software Matlab, který je již tradičně považován za standard v technické praxi.

V dalších krocích řešení již musí být posuzováno chování robotu během procesu aktivace a konvergence navržených řídicích členů. Pro zajištění věrohodnosti je přitom vhodné postihnout nejen statické, ale i dynamické účinky působící na robot, jako např. setrvačné síly apod. Ze stanovených cílů plyne potřeba návrhu členitého prostředí, v němž má robot operovat. Z analýzy provedené v teoretické části vyplývá, že musí být umožněno rozšíření simulace o algoritmizaci výpočtů uvedených v kap. 2.3 a 2.4.4. S ohledem na stanovený způsob hodnocení chůze je vhodné, aby simulace poskytovala jednak textový a jednak grafický výstup. Simulační software tedy musí být schopen:

- modelovat soustavy tvořené pevnými tělesy, které jsou spojeny kinematickými vazbami
- modelovat členité prostředí s překážkami
- zahrnout statické i dynamické účinky sil působících na tělesa
- zahrnout komplexní algoritmy ovlivňující silová působení, polohy a rychlosti
- zajistit opakovatelnost experimentu s dodržáním požadované přesnosti
- uložit a načíst konkrétní stav simulace
- poskytnout grafický i textový výstup simulace
- poskytnout stabilní funkčnost i pro simulaci složitých robotických soustav

Nalezení vhodného nástroje, který by splňoval všechny stanovené požadavky, se nakonec ukázalo jako jeden z vážných vedlejších problémů, který musel být vyřešen. Při hledání byly uvažovány dvě hlavní možnosti, buďto využít již hotový komplexní simulační software nebo jako základ simulace použít některou z existujících knihoven sloužících pro tvorbu simulací pevných těles, tzv. *fyzikální engine*.

V prvním případě byl zvažován především program ADAMS, který je v technické praxi hojně využíván. Na základě rozboru úloh řešených v rámci FSI a konzultace se školitelem byl tento program zamítnut. Jeho přínos by totiž byl především při vytvoření modelu robotu, zkoumání vlivu jednotlivých řídicích členů na robot a sestavení nízkourovňového řízení jednotlivých pohonů. Hlavní cíl této práce je ovšem využití algoritmu Q-učení prostřednictvím komplexního návrhu vícevrstvé architektury využívající metody predikátové logiky, což vyžaduje rozsáhlejší algoritmizaci postupu. A

právě připojení vlastního kódu, který by zahrnoval celý návrh jednotky řízení k modelu robotu, by bylo velmi složité. Toto propojení bylo v době tvorby simulace možné jen prostřednictvím jazyka Fortron a nedostatečně zdokumentovaného rozhraní programu ADAMS. Autor práce se ve svém tvrzení odvolává na konzultaci v rámci odborné přednášky [52].

Druhou možnost představují knihovny zajišťující fyzikální simulace pevných a případně i pružných těles. Jako nejperspektivnější se v době výběru jevil Open Dynamics Engine (dále jen ODE), byl totiž průběžně aktualizován svými autory a používán v mnoha projektech řešící simulace mechatronických soustav, [48]. Aby bylo možné označit výstupy poskytované touto knihovnou za věrohodné, bylo nejprve nutné provést porovnávací experimenty. Vhodností použití a přesností výstupů pro potřeby simulace robotu se zabývalo několik bakalářských prací pod vedením autora této disertační práce. V práci [33] je ukázána schopnost ODE simulovat čtyřnohý robot Quasimodo se dvěma stupni volnosti na nohu a jeho dynamicky stabilní styl chůze. Předlohou tohoto modelu je robot vyvinutý na UAI FSI [40]. Výsledky zhodnocené v této bakalářské práci prokázaly dostatečnou shodu mezi simulovanou a reálnou chůzí. V práci [35] pak byl navržen automatický generátor prostředí pomocí ODE umožňující simulaci různých mobilních robotů a především testování různých algoritmů lokalizace, hledání cesty, plánování chůze atd. ve vnitřních prostředích. O absenci ucelených řešení, které by byly schopny zajistit komplexní dynamické simulace mobilních robotů a splňovaly by výše uvedené požadavky, svědčí i fakt, že jsou v současnosti stále uváděny na trh nová komerční řešení. Jako příklad lze uvést software V-REP švýcarské firmy K-Team sloužící právě pro tyto účely [50]. Tento simulační software je rovněž založen na zmíněné knihovně ODE.

Zkušenosti získané při tvorbě uvedených prací byly využity při návrhu komplexního simulačního software založeného na ODE, viz. kap. 3.3, použitého pro verifikační experimenty předkládaného řešení. K vytvoření tohoto simulačního programu byl použit programovací jazyk Python. Hlavní důvody pro použití tohoto jazyku jsou:

- snadná spolupráce s knihovnou ODE prostřednictvím modulu pyODE
- umožňuje objektové programování
- modul numPy lze použít pro maticové výpočty a numerická řešení
- modul matplotlib nabízí reprezentaci naměřených dat ve 2D i 3D grafech
- modul pyOpenGL zajišťuje spolupráci s grafickou knihovnou OpenGL

Pro vykreslení simulované scény byla použita grafická knihovna OpenGL. Požadavek na uložení stavu simulace a její opětovné spuštění byl vyřešen pomocí standardizovaného formátu XODE, což je textový soubor přesně definované struktury ukládaný ve formátu XML.

3.2 Platforma použitá pro verifikaci návrhu řešení

Předkládané řešení umožňující využití opakovaně posilovaného učení pro řízení robotu, jehož teoretický rozbor byl podán v kap. 2.3 a 2.4, klade maximální důraz na dodržení obecnosti a rozšiřitelnosti. Jedná se tedy o univerzální návrh použitelný pro různé kráčívé roboty plnící obecně různé úkoly, jejichž zadání může být měněno bez nutnosti zásadní změny jednotky řízení. Aby však bylo možné doložit vhodnost návrhu a poskytnout

porovnání s již publikovanými výsledky v odborné literatuře, je nutné vztáhnout tento obecný návrh ke konkrétní platformě. Jako testovací robot byl zvolen sériově vyráběný model EQ3-KT firmy Lynxmotion, viz. Obr. 4. V následující kapitole je uveden popis tohoto robotu. V kapitole 3.2.2 jsou zavedeny souřadné systémy, které jsou následně využívány při konstrukci základních řídicích členů a kapitola 3.2.3 popisuje sestavení modelu robotu pro potřeby verifikačního experimentu.



Obr. 4: Předloha simulačního experimentu, robot EQ3-KT [49]

3.2.1 Popis robotu

Platforma zvolená jako ukázkový příklad pro posouzení schopností navrhovaného řešení je čtyřnohý kráčivý robot EQ3-KT, který má tři stupně volnosti na nohu. Je tedy umožněn obecný pohyb těla robotu v prostoru aniž by docházelo ke změně bodů došlapu. Rozměr těla robotu je 240x200x55 mm. Každá z noh, formálně označených $(0,1,2,3)$, má tři části, dále označované C_k pro $k=1,2,3$, vzájemně propojené rovinnými rotačními vazbami (A^i, M^i, P^i) pro $i=1,2,3,4$. Příslušné body došlapu necht' jsou označeny N^i pro $i=1,2,3,4$. Necht' a značí vzdálenost vazeb $A^i M^i$, b značí vzdálenost $M^i P^i$ a c značí vzdálenost vazeb $P^i N^i$. Pak platí:

$$a = 30\text{mm}, b = 70\text{mm}, c = 110\text{mm} \quad (26)$$

Jednotlivé rotační vazby omezují jak počet stupňů volnosti, tak i vzájemné natočení těles. Úhlový rozsah vazeb pro nohu 0 je následující:

$$A^0 : \alpha \in \left\langle -\frac{\pi}{4}, \frac{\pi}{2} \right\rangle, M^0 : \beta \in \left\langle -\frac{\pi}{4}, \frac{\pi}{2} \right\rangle, P^0 : \gamma \in \left\langle -\frac{\pi}{4}, \frac{\pi}{4} \right\rangle \quad (27)$$

kde α, β, γ značí úhly vymezené příslušnými vazbami. Rozsah pro ostatní nohy je stejný, ovšem s jinými znaménky, jak ukazuje Obr. 6.

Celková hmotnost robotu je 1320 g, z toho 670 g připadá na tělo. Robot je osazen celkem 12ti servopohony Hitec HS-475HB, váha jednoho pohonu činí 40g.

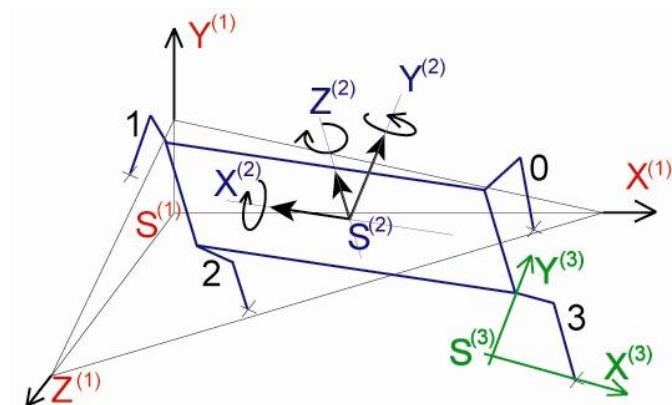
3.2.2 Volba souřadných systémů

Za účelem určení polohy a konfigurace právě popsaného robotu, stejně tak jako určení polohy překážek vůči robotu, byly v rámci této práce zavedeny následující souřadné systémy (dále jen SS), viz. Obr. 5:

souřadný systém spojený se zemí $SS^{(1)}$: tento SS je pevně spjat s prostředím, přičemž osa $Y^{(1)}$ tohoto kartézského SS má opačný směr než tíhová síla působící na robot,

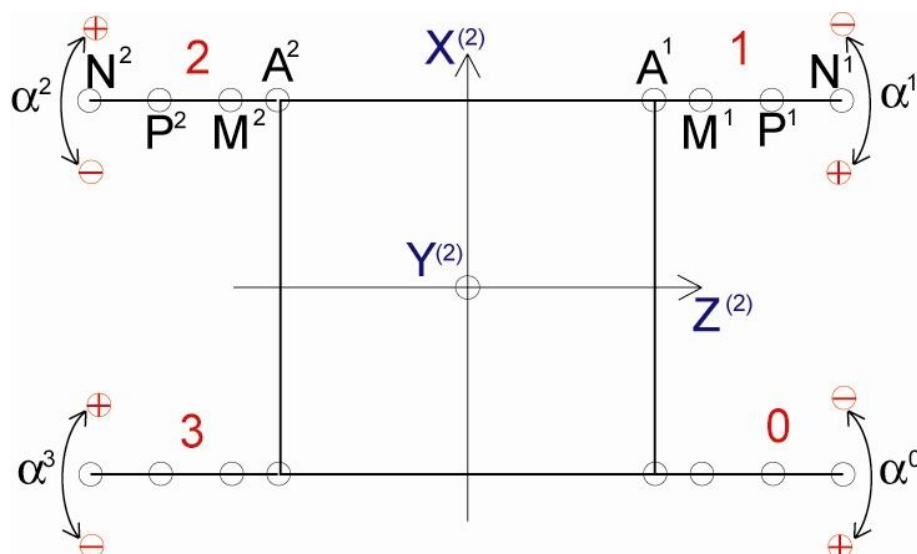
souřadný systém spojený s trupem robotu $SS^{(2)}$: tento SS je pevně spjat s trupem robotu, počátek je umístěn do hmotného středu těla,

souřadný systém nohy $SS^{(3)}$: tento SS leží v rovině nohy robotu, se kterou je pevně spjat, přičemž osa $Y^{(3)}$ prochází vazbou A^i a osa $X^{(3)}$ prochází bodem došlapu N^i .



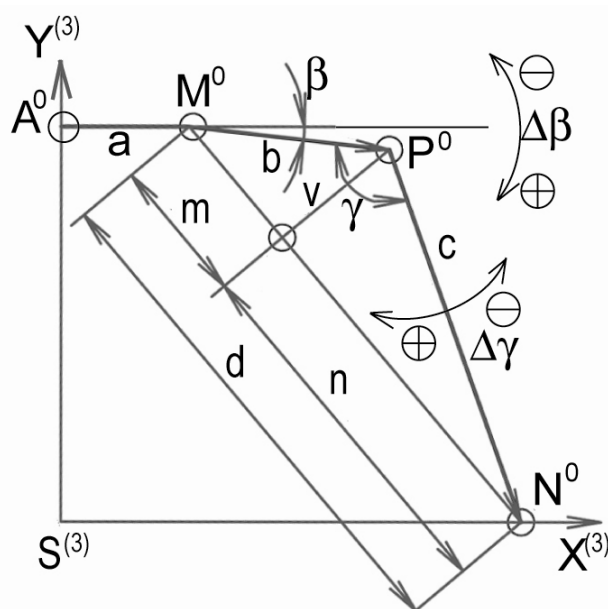
Obr. 5: Schéma zavedených souřadných systémů robotu

Na Obr. 6 je zobrazeno schéma robotu v rovině těla. Je zde patrné umístění jednotlivých bodů reprezentujících rotační vazby uvedené v kap. 3.2.1 a také kladný směr změny úhlu α^i pro jednotlivé nohy.



Obr. 6: Schéma robotu, zobrazení v rovině těla robotu

Pro úplnost je vhodné zobrazit souřadný systém $SS^{(3)}$ a to v rovině nohy, viz. Obr. 7, který zobrazuje nohu 0. Je patrné, že osa $X^{(3)}$ prochází bodem došlapu N^0 a osa $Y^{(3)}$ prochází vazbou A^0 . Body (A^0, M^0, P^0) opět značí příslušné kinematické vazby. Dále jsou zde zobrazeny kladné změny úhlů β^0 a γ^0 . Stejná orientace platí i pro nohu 1. Pro nohy umístěné na kontralaterální straně, tj. nohy 2 a 3, je kladný směr orientován opačně. Další proměnné uvedené na Obr. 7 budou použity při sestavení potenciálové funkce základních řídicích členů v kap. 3.3.



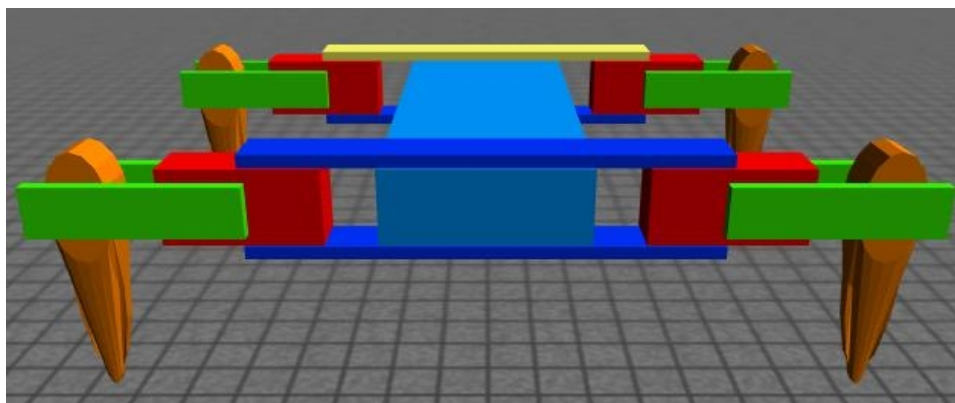
Obr. 7: Souřadný systém spjatý s nohou 0 robotu

3.2.3 Vytvoření modelu robotu

Pro potřeby simulačního experimentu byl sestaven model robotu, jehož parametry jsou popsány v kap. 3.2.1 a 3.3.2. K tomuto účelu posloužil již zmíněný *Open Dynamics Engine*, viz kap. 3.1. Vytvoření modelu robotu pomocí ODE neprobíhá interaktivním způsobem v grafickém režimu, tak jak je tomu zvykem u jiných aplikací, např. CAD systémy, ale výhradně prostřednictvím volání funkcí z ručně psaného kódu. ODE neposkytuje zobrazení vytvořeného modelu, grafický výstup musí být realizován ve vlastní režii, podrobně viz. kap. 3.4. Sestavení modelu probíhalo v tomto pořadí:

- volba základních těles, ze kterých bude model sestaven
- případný import členitých těles složitějších tvarů
- nastavení parametrů těles (rozměr, hmotnost, moment setrvačnosti, aj.)
- umístění těles do prostoru a vzájemné provázání požadovanými vazbami
- nastavení dodatečných parametrů vazeb (max. zatížení, rozsah, ad)
- umístění modelu do požadovaného prostředí

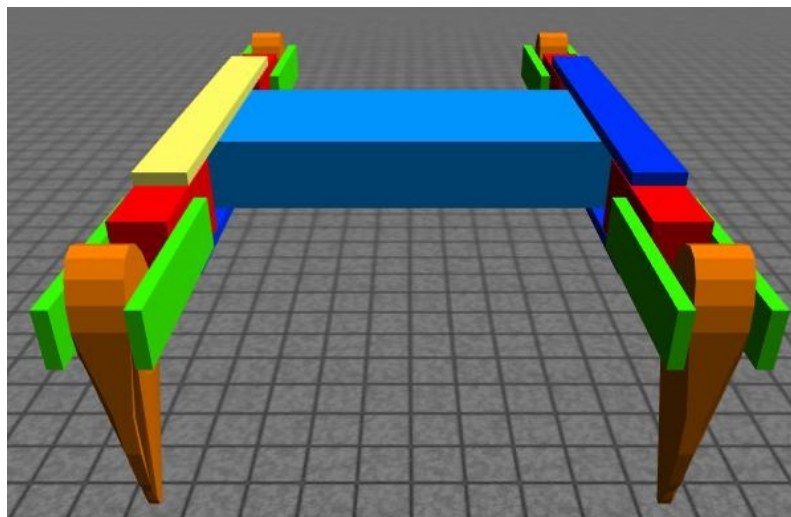
Robot EQ3-KT lze téměř celý modelovat s využitím základních těles ODE, tj. krychle a kvádr. Rozměry a hmotnosti těchto těles byly nastaveny dle skutečných parametrů uvedených v kap. 3.2.1. Výjimku představuje koncový článek noh, C_3 , který nelze modelovat užitím základních těles, neboť má specifické rozložení hmotnosti způsobené umístěním servopohonu do horní části, a rovněž spodní došlapová část článku má tvar polokoule. Modelování tohoto článku byla věnována zvýšená pozornost. V průběhu času bylo vytvořeno několik vývojových modifikací, konečná verze je patrná z Obr. 8 a Obr. 9. Model tohoto článku nohy byl vytvořen pomocí open source programu Wings 3D v 1.0.2, který je zcela postačující pro uvedené požadavky. Import modelu článku do ODE proběhl pomocí standardizovaného formátu OBJ a volání vestavěných funkcí. Tyto funkce zajišťují pouze načtení drátového modelu (mesh). Nastavení hmotnosti, momentu setrvačnosti, ad., probíhá již standardním způsobem jako u ostatních těles prostřednictvím vestavěných funkcí, viz. dokumentace ODE [48].



Obr. 8: Model robotu, pohled ve směru longitudinální osy $X^{(2)}$

Model robotu, viz. Obr. 8, je tvořen celkem 21 tuhými tělesy, přičemž tělo platformy tvoří pět těles vzájemně spojených vetknutím. Tato tělesa jsou naznačena

modře. Každá z noh je tvořena čtyřmi tělesy. Článek C_1 reprezentuje kvádr (červená barva) spojená s tělem rotační vazbou, která umožňuje článku rotační pohyb v rovině těla. Článek C_2 reprezentují dva kvádry (zelená barva), které jsou propojeny rotačními vazbami s článkem C_1 a koncovým článkem C_3 (zelená barva), čímž je vymezen rotační pohyb v rovině kolmé k mediální rovině. Z důvodu snadného vizuálního zjištění orientace robotu v prostoru, má jedno z těles, které vymezuje čelní rovinu robotu, žlutou barvu. Model robotu tvoří celkem 20 rotačních vazeb a 4 vetknutí.

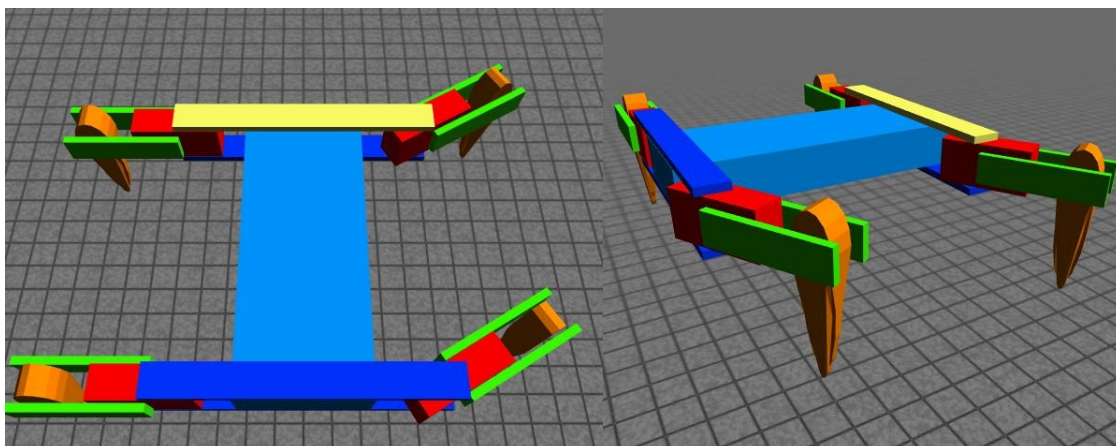


Obr. 9: Model robotu, pohled ve směru osy $Z^{(2)}$

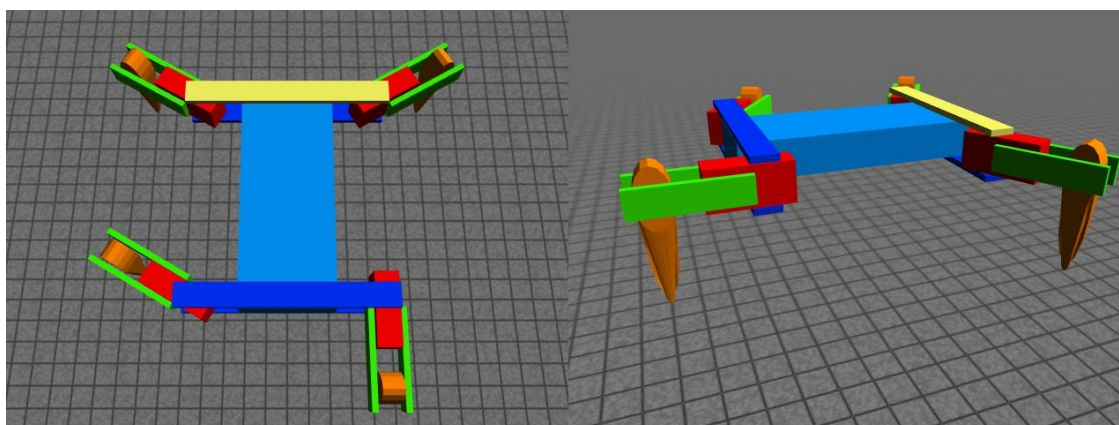
Zdrojový kód modelu je uveden v souboru *model_robotu_3dof.py* a parametry modelu jako jsou rozměry, hmotnost, úhlové rozsahy ad. jsou uvedeny v souboru *parametry.py*. Oba soubory jsou součástí předkládaného komplexního řešení simulace, podrobně popsané v kap. 3.4. Celý model je sestaven parametricky, může tak být například změněn rozměr všech těles robotu nebo pracovní prostor noh apod. s minimálním nutným zásahem do zdrojového kódu.

3.2.4 Vybrané konfigurace

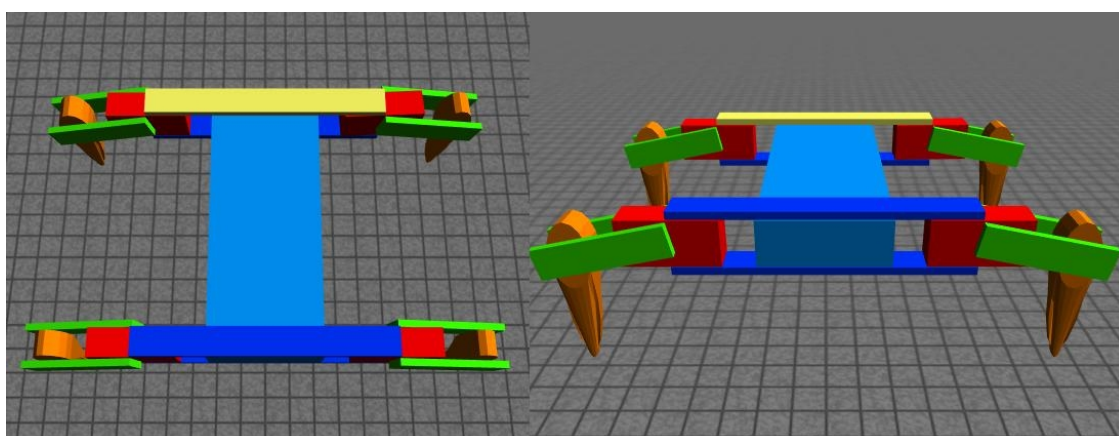
V této kapitole jsou z důvodu lepší orientace popsány vybrané konfigurace robotu, které jsou v další části práce využity pro posouzení průběhu potenciálových funkcí navržených řídicích členů a také pro ukázání elementárních chování robotu vlivem aktivace vybraných instancí těchto řídicích členů v rámci simulace. Numerický popis konfigurace je pro názornost vždy doplněn o grafické zobrazení modelu robotu nacházejícího se v dané konfiguraci. Grafické podklady jsou exportovány přímo z vytvořeného simulačního software:



Obr. 10: Konfigurace robotu č. 7



Obr. 11: Konfigurace robotu č. 8



Obr. 12: Konfigurace robotu č. 11

Tabulka 1: Hodnoty úhlů vybraných konfigurací robotů

č.	úhel α^i [rad]	úhel β^i [rad]	úhel γ^i [rad]	obr.
1	$(0,0,0,0)$	$(0,0,0,0)$	$\left(\frac{\pi}{2}, \frac{\pi}{2}, \frac{\pi}{2}, \frac{\pi}{2}\right)$	Obr. 8 a Obr. 9
7	$\left(-\frac{\pi}{6}, -\frac{\pi}{6}, 0, 0\right)$	$(0,0,0,0)$	$\left(\frac{\pi}{2}, \frac{\pi}{2}, \frac{\pi}{2}, \frac{\pi}{2}\right)$	Obr. 10
8	$\left(\frac{\pi}{2}, -\frac{\pi}{6}, \frac{\pi}{6}, \frac{\pi}{6}\right)$	$(0,0,0,0)$	$\left(\frac{\pi}{2}, \frac{\pi}{2}, \frac{\pi}{2}, \frac{\pi}{2}\right)$	Obr. 11
11	$(0,0,0,0)$	$\left(\frac{\pi}{12}, \frac{\pi}{12}, -\frac{\pi}{12}, -\frac{\pi}{12}\right)$	$\left(\frac{7\pi}{12}, \frac{7\pi}{12}, \frac{7\pi}{12}, \frac{7\pi}{12}\right)$	Obr. 12

3.3 Sestavení elementárních řídicích členů

Základem popsaného návrhu jednotky řízení robotu je báze řízení, viz. kap. 2.3.1, obsahující elementární řídicí členy. Pro potřeby plánování chůze robotu je vhodné do této množiny zahrnout řídicí člen postoje, Φ_P , kontaktu, Φ_K , a trajektorie, Φ_T . Každý z těchto členů je schopen díky propojení se vstupními a výstupními zdroji generovat elementární chování robotu. V první fázi návrhu byly navrženy potenciálové funkce těchto řídicích členů. Dále byl v prostředí Matlab sestaven geometrický model robotu, který posloužil pro vyhodnocení charakteru sestavených funkcí a pro posouzení správnosti návrhu jednotlivých instancí řídicích členů v různých konfiguracích robotu.

3.3.1 Řídicí člen postoje

Hlavním cílem tohoto řídicího členu, Φ_P , je dosažení optimálních poloh noh robotu vzhledem k jejich pracovnímu prostoru, jak je uvedeno v kap. 2.3.1.1. Jako kritérium optimality je použito heuristické míry postoje, viz. rovnice (5),(28).

$$\Phi_P = m \quad (28)$$

Tato rovnice udává charakter potenciálové funkce řídicího členu. Z rovnice (5) plyne, že pro výpočet hodnoty heuristické míry postoje, m , je v dané konfiguraci zapotřebí znát všechny úhly, které vymezují kinematické vazby.

3.3.1.1 Popis vybraných instancí

Optimálních poloh noh lze dosáhnout různými způsoby. Lze tak učinit rotačním nebo translačním pohybem těla robotu při zachování bodů došlapu, případně pomocí změny konfigurace jedné nohy při změně bodu došlapu. Tento postup umožňuje sestavit několik různých instancí řídicího členu postoje.

$$C_1 = \left\{ \Phi_{P\varphi}^{0,1,2,3}, \Phi_{P\chi}^{0,1,2,3}, \Phi_{P\psi}^{0,1,2,3} \right\}. \quad (29)$$

Rovnice (29) uvádí první množinu instancí, které se snaží o dosažení optimální konfigurace rotací těla robotu. Konkrétně změnou úhlu klopení, χ , tj. rotací kolem osy $X^{(2)}$, viz. Obr. 1. Nebo změnou úhlu bočení, φ , tj. rotací kolem osy $Y^{(2)}$. Případně

změnou úhlu klonění, ψ , tj. rotací kolem osy $Z^{(2)}$. Dále lze uvést instance, které se snaží o dosažení optimální konfigurace translačním pohybem těla, viz. rovnice (30).

$$C_2 = \left\{ \Phi_{P_{x,z}}^{0,1,2,3}, \Phi_{P_y}^{0,1,2,3} \right\}. \quad (30)$$

První instance provádí translační pohyb v rovině těla robotu, zatímco druhá mění výšku těla robotu nad zemí. Instance jsou uvedeny odděleně, neboť např. pro řešení úlohy rotace robotu je nežádoucí, aby robot měnil svoji polohu v osách $X^{(2)}, Z^{(2)}$. Zároveň je ale možné, v případě potřeby docílit chování, kdy tělo robotu provádí translační pohyb ve všech osách a to pomocí kompozitního řídicího členu zahrnujícího obě uvedené instance.

Třetí styl chování, který je možné docílit instancí řídicího členu postoje, je změna konfigurace zvolené nohy, jak uvádí následující rovnice. Tyto instance mění vždy konfiguraci pouze jedné nohy, nemění proto ani polohu či natočení těla ani konfiguraci ostatních noh.

$$C_3 = \left\{ \Phi_{P_i}^i \right\} \quad \text{pro } i = 0, 1, 2, 3 \quad (31)$$

3.3.1.2 Sestavení potenciálové funkce

Nyní je zapotřebí nalézt průběh potenciálové funkce, viz. rovnice (5), uvedených instancí řídicího členu postoje pro vybrané konfigurace robotu. Obecně je nutné řešit následující problém. Robot se nachází ve známé výchozí konfiguraci, $(x, y, z, \varphi, \chi, \psi, N^i, \alpha^i, \beta^i, \gamma^i)$ pro $i = 0, 1, 2, 3$. V této konfiguraci je provedena jedna z těchto akcí:

- posuv těla robotu,
- rotace těla robotu,
- změna došlapu vybrané nohy.

Po provedení akce hledáme nové hodnoty úhlů všech noh $\alpha^i, \beta^i, \gamma^i$ pro $i = 0, 1, 2, 3$, pomocí nichž je možné spočítat hodnotu heuristické míry postoje. Ověření průběhu potenciálové funkce stačí provést pouze pro změnu úhlu bočení a posuv v rovině těla robotu. V případě vyhovujícího výsledného charakteru funkce lze usuzovat přijatelné výsledky i pro změnu ostatních parametrů, což bude následně prokazováno již přímo pomocí simulace chování robotu za pomoci 3D modelu popsaného v kap. 3.2.3.

Pro změnu úhlu bočení, tj. rotační pohyb, platí:

$$\begin{aligned} N_{Z'}^{(2)} &= N_Z^{(2)} \cos(\varphi) + N_X^{(2)} \sin(\varphi), \\ N_{X'}^{(2)} &= -N_X^{(2)} \sin(\varphi) + N_Z^{(2)} \cos(\varphi). \end{aligned} \quad (32)$$

kde $[N_X^{(2)}, N_Z^{(2)}]$ je počáteční bod došlapu vybrané nohy, $[N_{X'}^{(2)}, N_{Z'}^{(2)}]$ je bod došlapu po změně úhlu bočení těla robotu, φ je úhel, o který došlo k rotaci těla kolem osy $Y^{(2)}$. Obdobně pro změnu polohy, tj. translační pohyb, platí:

$$\begin{aligned} N_{Z'}^{(2)} &= N_Z^{(2)} - \delta_Z, \\ N_{X'}^{(2)} &= N_X^{(2)} - \delta_X. \end{aligned} \quad (33)$$

kde δ_x a δ_z je posuv v ose $X^{(2)}$ resp. $Z^{(2)}$. Pro vektory $\vec{u}, \vec{v}$ platí:

$$\begin{aligned}\vec{u} &= N^{(2)} - A^{(2)} = (u_1, u_2), \\ \vec{v} &= Z^{(2)} - A^{(2)} = (v_1, v_2)\end{aligned}\quad (34)$$

kde $A^{(2)}$ je bod A po provedení rotačního, resp. translačního pohybu. Bod Z je libovolný bod ležící na přímce určené body $\overline{A^3 A^0}$ resp. $\overline{A^2 A^1}$. Pro úhel α platí:

$$\alpha = \arccos \left(\frac{u_1 v_1 + u_2 v_2}{\sqrt{u_1^2 + u_2^2} \cdot \sqrt{v_1^2 + v_2^2}} \right). \quad (35)$$

Následující rovnice lze vztáhnout k libovolné noze $i = 0, 1, 2, 3$. Nyní je možné určit úhly β^i a γ^i . Velikost těchto úhlů je nutné určit v rovině nohy, tj. v $SS^{(3)}$, viz. Obr. 7. Pro velikost vektoru $\vec{u}$ platí:

$$\left| \vec{u} \right| = \sqrt{u_1^2 + u_2^2} = N_x^{i(3)} \quad (36)$$

kde $N_x^{i(3)}$ je x souřadnice bodu došlapu nohy i v souřadném systému $SS^{(3)}$. V rovině nohy dále platí:

$$m + n = d, \quad v^2 + m^2 = b^2, \quad v^2 + n^2 = c^2. \quad (37)$$

Hodnotu m lze určit pomocí rovnice (37) takto:

$$m = \frac{b^2 - c^2}{2d} + \frac{d}{2}. \quad (38)$$

a pro velikost výšky v pak platí:

$$v = \sqrt{b^2 - m^2}. \quad (39)$$

Ze znalosti hodnot v a d je možné určit průsečík $X^{i(3)}$ takto:

$$\begin{aligned}X_X^{i(3)} &= M_X^{i(3)} + \left(\frac{m}{d} \right) \cdot (N_X^{i(3)} - M_X^{i(3)}), \\ X_Y^{i(3)} &= M_Y^{i(3)} + \left(\frac{m}{d} \right) \cdot (N_Y^{i(3)} - M_Y^{i(3)}).\end{aligned}\quad (40)$$

Odtud pro bod $P^{0(2)}$ plyne:

$$\begin{aligned}P_X^{i(3)} &= X_X^{i(3)} \pm \left(\frac{v}{d} \right) \cdot (M_Y^{i(3)} - N_Y^{i(3)}), \\ P_Y^{i(3)} &= X_Y^{i(3)} \pm \left(\frac{v}{d} \right) \cdot (M_X^{i(3)} - N_X^{i(3)}).\end{aligned}\quad (41)$$

Pomocí bodu $P^{0(2)}$ lze určit hledaný úhel β^0 :

$$\beta = -\arctan \left(\frac{M_Y^{i(3)} - P_Y^{i(3)}}{P_X^{i(3)} - M_X^{i(3)}} \right). \quad (42)$$

Pro výpočet úhlu γ je vhodné zavést vektory $\vec{s}, \vec{t}$ takto:

$$\begin{aligned}\mathbf{r} &= (P_X^{i(3)} - M_X^{i(3)}, P_Y^{i(3)} - M_Y^{i(3)}) = (s_1, s_2), \\ \mathbf{t} &= (N_X^{i(3)} - P_X^{i(3)}, N_Y^{i(3)} - P_Y^{i(3)}) = (t_1, t_2).\end{aligned}\quad (43)$$

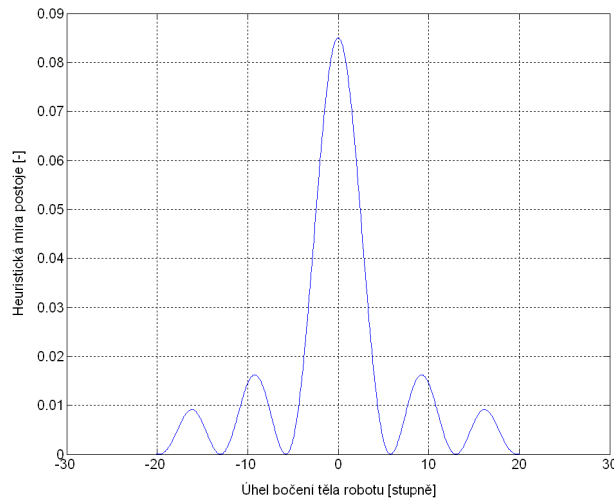
Pro úhel gama pak platí:

$$\gamma = \arccos \left(\frac{s_1 \cdot t_1 + s_2 \cdot t_2}{\sqrt{s_1^2 + s_2^2} \cdot \sqrt{t_1^2 + t_2^2}} \right). \quad (44)$$

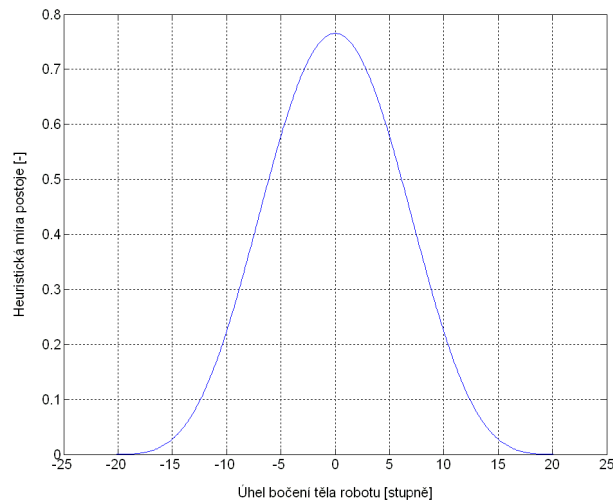
Rovnice (35), (42) a (44) vyjadřují velikosti hledaných úhlů $\alpha^i, \beta^i, \gamma^i$ pro jednotlivé nohy $i = 0, 1, 2, 3$. Tyto hodnoty lze použít pro výpočet heuristické míry postoje, dle rovnice (5). Uvedený postup byl algoritmován a následně implementován v prostředí Matlab pro potřeby nalezení průběhu potenciálové funkce jednotlivých řídicích členů v různých konfiguracích robotu.

3.3.1.3 Posouzení charakteru použité funkce

Nejprve je posuzován řídicí člen postoje, který využívá jako výstupního zdroje úhel bočení, φ . Jedná se o instanci $\Phi_{P\varphi}^{0,1,2,3}$. Průběh jeho funkce je znázorněn na Obr. 13 při použití normalizace úhlů na interval $\langle -\pi, \pi \rangle$, která je uvedena v odborné literatuře, viz. kap. 2.3.1.1. Je zjevné, že charakter je zcela nevhodný pro hledání globálního extrému pomocí gradientní metody, tak jak je požadováno, viz. kap. 2.3.1. Výsledný průběh při využití normalizace úhlů na interval $\langle -\pi/2, \pi/2 \rangle$ je ukázán na Obr. 14. Prvotní návrh tedy ukázal na nevhodný způsob normalizace úhlů. Oba výše zmíněné grafy představují průběh funkce pro výchozí konfiguraci robotu č. 1, popis konfigurace je uveden v kap. 3.2.4.

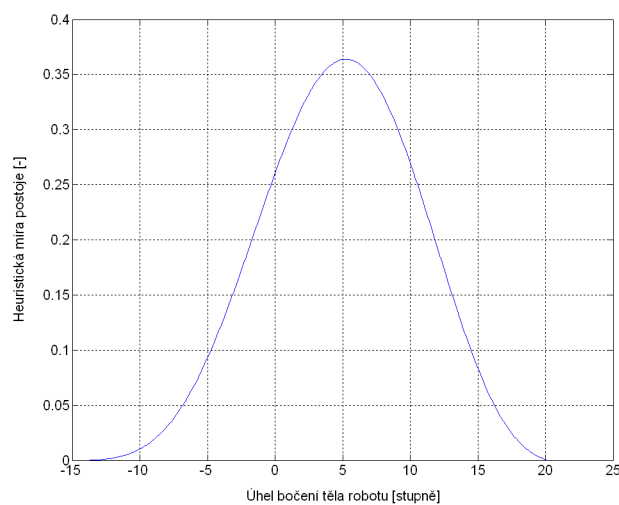


Obr. 13: Potenciálová funkce řídicího členu postoje, nevhodná normalizace

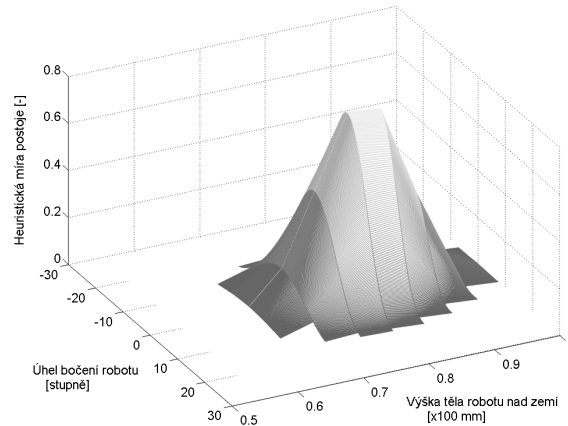


Obr. 14: Potenciálová funkce řídicího členu postoje, konfigurace č.1

Pro jinou výchozí konfiguraci robotu (č. 7, Obr. 10) získáme obdobný průběh, ovšem globální maximum bude v grafu posunuto v ose X . Výsledný graf je uveden na Obr. 15. Pokud budeme zkoumat průběh funkce pro výchozí konfigurace, které vzniknou z konfigurace č. 1 posuvem těla robotu v ose $Z^{(2)}$, tj. změnou úhlů β, γ na všech nohách a tedy změnou výšky těla robotu nad zemí, obdržíme totožné průběhy funkcí. Maximální hodnota heuristické míry postoje bude pro jednotlivé konfigurace různá, tj. globální extrém bude na grafu posunut v ose Y . Průběh heuristické míry postoje pro konfigurace robotu s rozdílnou výškou těla nad zemí je uveden na Obr. 16. Je patrné, že existuje pro dané neměnné body došlapu ideální výška těla robotu nad zemí, při které lze docílit maximální hodnoty heuristické míry postoje. Z grafu rovněž plyne, že instance řídicího členu postoje, $\Phi_{P_y}^{0,1,2,3}$, která optimalizuje výšku těla robotu nad zemí prostřednictvím translačního pohybu, bude vyhovovat požadavkům kladeným na tvar plochy funkce.

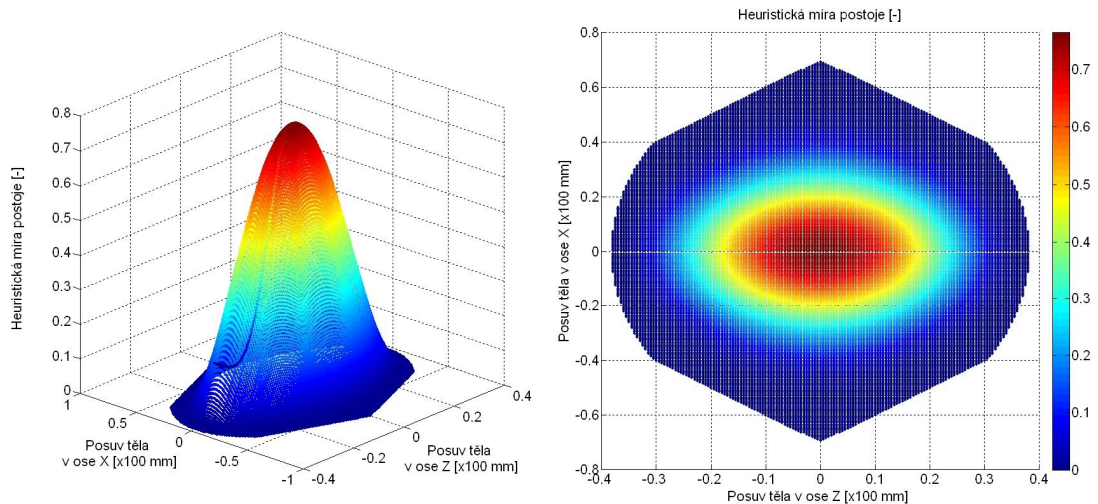


Obr. 15 : Potenciálová funkce řídicího členu postoje, konfigurace č. 7

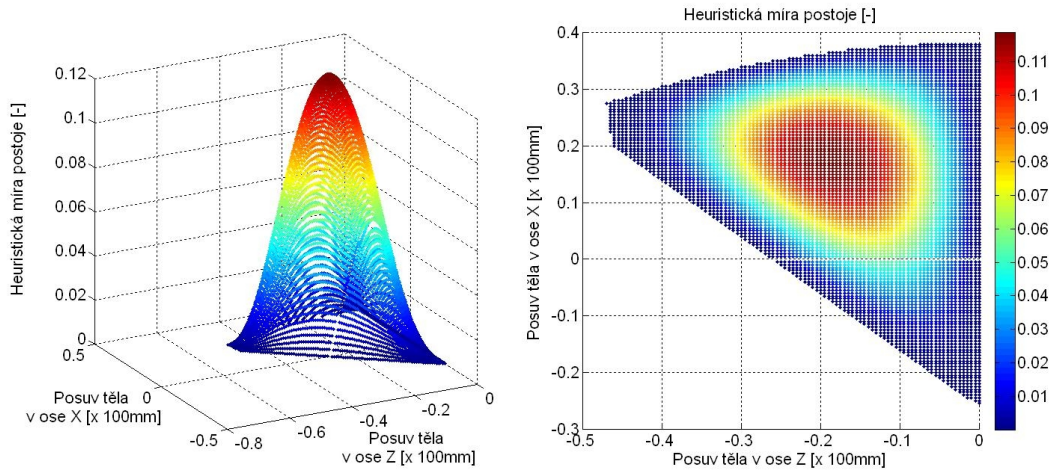


Obr. 16: Potenciálové funkce řídicího členu postoje pro různé výšky těla nad zemí

Řídicí člen postoje je možné použít také s výstupními zdroji x, z . Příslušná instance pak má tvar $\Phi_{P_{x,z}}^{0,1,2,3}$. Na Obr. 17 je uveden průběh potenciálové funkce tohoto řídicího členu pro výchozí konfiguraci č.1. Graf ukazuje závislost velikosti heuristické míry postoje na poloze těla robotu, která se mění v rovině těla robotu, tj. ve směru longitudinální a laterální osy těla. V případě použití odlišné výchozí konfigurace robotu, č. 7, je výsledný tvar potenciálové funkce opět přijatelný, neboť má právě jeden globální extrém, viz. Obr. 18. Jiný tvar ohraničující křivky grafu v osách X, Z je dán jinou výchozí konfigurací a omezeným rozsahem vazeb noh, které neumožňují tělu robotu dosáhnout stejných posunutí jako v předchozím případě.



Obr. 17: Potenciálové funkce řídicího členu $\Phi_{P_{x,z}}^{0,1,2,3}$ pro konfiguraci č.1



Obr. 18: Potenciálové funkce řídicího členu $\Phi_{P_{X,Z}}^{0,1,2,3}$ pro konfiguraci č. 7

Poslední skupinu instancí řídicího členu postoje představují zákony řízení, které jako výstupní zdroj používají pouze jednu nohu robotu. Jak bylo uvedeno v kap. 3.3.1.1, cílem je dosažení optimální konfigurace zvolené nohy změnou jejího došlapu. Blíže není v odborné literatuře tato instance komentována. V předkládaném řešení byl použit modifikovaný výpočet heuristické míry postoje, který neuvažuje úhly všech noh, ale pouze nohy manipulované, viz. rov. (45):

$$\Phi_{P_i}^i = m^i = p_i \prod_{j < 3} \cos(\Theta_{\varepsilon ij}), \quad i = 0, 1, 2, 3 \quad (45)$$

Důvodem tohoto postupu je snaha o dosažení středních poloh úhlů dané nohy, nikoliv dosažení co nejlepší manipulovatelnosti robotu jako celku. Pokud by se robot nacházel na vodorovné rovině bez překážek, je možné předpokládat stejný charakter potenciálové funkce jako u instancí $\Phi_{P_{X,Z}}^{0,1,2,3}$. Snahou je ovšem navrzení takového řídicího členu, který by byl schopen své činnosti i v neznámém členitém prostředí s překážkami. V tomto případě je nutné uvažovat potenciálně velkou neúspěšnost při snaze umístit nohu do teoreticky spočtené konfigurace, např. díky výskytu nerovnosti nebo obecně překážky. Pro činnost tohoto řídicího členu byl navržen iterativní postup, kdy je koncový bod nohy polohován opakovaně o experimentálně stanovenou hodnotu Δx_z směrem k předpokládanému maximu heuristické míry postoje stanovené dle rovnice (45). Změna výšky bodu došlapu polohované nohy nad povrchem je řízena reaktivně dle charakteru prostředí. Jestliže narazí noha během přesunu do nové polohy na překážku resp. členitost terénu, což lze detekovat zvýšeným odběrem proudu na servopohonech při nulové změně úhlů nohy, pokouší se řídicí člen překážce vyhnout zvýšením resp. snížením výšky bodu došlapu nad povrchem. V případě, že toto není možné učinit s ohledem na omezení daná rozsahem vazeb nohy, je noha navracena do poslední známé konfigurace a řídicí člen vrací nadřazené vrstvě architektury řízení informaci o svém neúspěchu. Konkrétní chování řídicího členu je ukázáno pomocí prostorové simulace robotu v prostředí s překážkami, viz. kap. 3.5.1.

3.3.2 Řídicí člen kontaktu

Hlavním cílem řídicího členu kontaktu, Φ_K , je docílení optimálního rozložení sil působících v došlapech. Tohoto stavu se snaží docílit translačním pohybem těla nebo změnou konfigurace vybrané nohy. S ohledem na informace uvedené v teoretickém rozboru tohoto řídicího členu, viz. kap. 2.3.1.2, bylo přistoupeno k vlastnímu návrhu tohoto řídicího členu. Výsledné potenciálové funkce tohoto návrhu jsou uvedeny v kap. 3.3.2. Nejprve je ovšem vhodné uvést možné instance tohoto řídicího členu a poskytnout popis očekávaného chování robotu, které tyto instance mají zajišťovat.

3.3.2.1 Popis vybraných instancí

Jestliže má řídicí člen kontaktu dosáhnout optimálního rozložení sil působících v došlapech robotu, může tak učinit vykonáním translačního pohybu těla nebo změnou konfigurací jednotlivých noh, přičemž je nutné rozhodnout, jestli budou v průběhu konvergence uvažovány reakce ve všech nebo jen ve vybraných bodech došlapu. Necht' první skupina instancí, které jsou použity pro plánování chůze robotu, vypadá takto:

$$C_4 = \left\{ \Phi_{K_{x,z}}^{0,1,2}, \Phi_{K_{x,z}}^{0,2,3}, \Phi_{K_{x,z}}^{0,1,3}, \Phi_{K_{x,z}}^{1,2,3}, \Phi_{K_{x,z}}^{0,1,2,3} \right\} \quad (46)$$

Tyto zákony řízení optimalizují rozložení reakcí v došlapech prostřednictvím translačního pohybu těla v osách $X^{(2)}$ a $Z^{(2)}$. Při své činnosti berou v úvahu jen tři vybrané nohy nebo v případě posledně uvedené instance všechny čtyři. Instance, které berou v potaz pouze tři nohy, mohou být s výhodou využity pro sestavení kompozitních řídicích členů, které by zajistili složitější chování robotu. Např. noha, jejíž konfigurace není uvažována řídicím členem kontaktu, může být využita řídicím členem postoje, který bude v rámci paralelní aktivace optimalizovat manipulovatelnost této nohy. Instance, $\Phi_{K_{x,z}}^{0,1,2,3}$, která využívá všechny čtyři nohy, nebude v další části práce využívána s ohledem na její přínos pro řešené úlohy.

Druhou množinu představují instance, které se snaží docílit optimálního rozložení reakcí změnou bodu došlapu vybrané nohy:

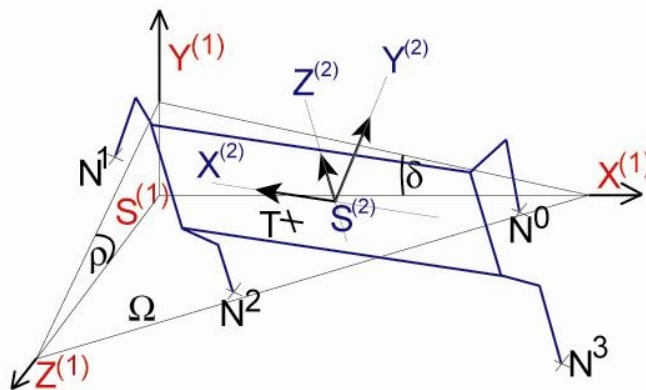
$$C_5 = \left\{ \Phi_{K_*}^{0,1,2}, \Phi_{K_*}^{0,2,3}, \Phi_{K_*}^{0,1,3}, \Phi_{K_*}^{1,2,3} \right\} \quad (47)$$

Znak * je substituční, na jeho místě může být použit kterýkoliv ze vstupních zdrojů. Zápis $\Phi_{K_*}^{0,1,2}$ tak označuje tři různé instance, které mají jako výstupního zdroje použítu nohu č. 0,1 nebo 2. Celkem tedy množina C_5 obsahuje 12 různých instancí řídicího členu kontaktu. Každá z nich mění konfiguraci pouze jedné nohy. Pro jejich činnost byl uplatněn stejný algoritmus, který byl navržen pro činnost řídicího členu postoje manipulující jednou nohou, viz. kap. 3.3.1.3, zajišťující reaktivní chování v případě výskytu překážky.

3.3.2.2 Sestavení potenciálové funkce

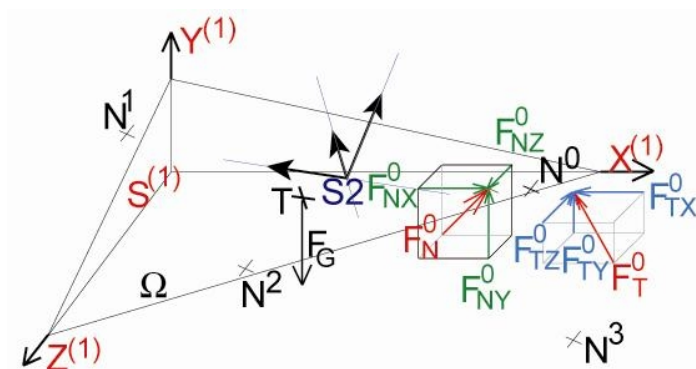
Řídicí člen kontaktu musí být dle stanovených cílů schopen plánovat pohyb robotu, který se nachází nejen na vodorovné rovině, ale i v členitém terénu. Po konzultaci se školitelem bylo přistoupeno k řešení modelové situace, kdy se robot nachází na nakloněné rovině, viz. Obr. 19. Body N^i opět označují místa kontaktu robotu se zemí. Bod T označuje

těžiště robotu. Rovina, Ω , na níž se robot nachází, vznikla rotací o úhel δ kolem osy $Z^{(1)}$ a o úhel ρ kolem osy $X^{(1)}$. V této situaci je nutné zjistit velikost sil působících v došlapěch tří zvolených noh, jelikož výše uvedené instance se snaží optimalizovat rozložení reakcí vždy právě ve třech nohách robotu. V případě čtvrté nohy není při výpočtu uvažován kontakt s podložkou.



Obr. 19: Robot umístěný na nakloněné rovině

Pokud se robot nachází ve statické rovnováze, lze po uvolnění vazeb uvažovat působení sil naznačených na Obr. 20, kde je pro přehlednost krom tíhové síly, F_G , naznačeno uvolnění pouze v bodě N^0 . Jedná se o normálovou složku reakce, F_N^0 , která působí v bodě došlapu ve směru kolmém k rovině. Tečná složka reakce, F_T^0 , působí rovněž v bodu došlapu, ale v rovině Ω . Směr této složky je shodný se směrem průmětu tíhové síly do roviny Ω . Na Obr. 20 je dále naznačen rozklad tečné a normálové složky reakce na složky působící v osách souřadného systému $SS^{(1)}$. Obdobným způsobem lze postupovat i pro zbylé body došlapu N^1, N^2 .



Obr. 20: Síly působící na robot v bodě došlapu N^0

Pro statickou rovnováhu robotu nacházejícího se na nakloněné rovině musí platit rovnice silové rovnováhy, (48)(49)(50), a rovnice momentové rovnováhy, (51)(52)(53):

$$F_{NX}^0 - F_{TX}^0 + F_{NX}^1 - F_{TX}^1 + F_{NX}^2 - F_{TX}^2 = 0 \quad (48)$$

$$F_{NY}^0 + F_{TY}^0 + F_{NY}^1 + F_{TY}^1 + F_{NY}^2 + F_{TY}^2 - F_G = 0 \quad (49)$$

$$F_{NZ}^0 - F_{TZ}^0 + F_{NZ}^1 - F_{TZ}^1 + F_{NZ}^2 - F_{TZ}^2 = 0 \quad (50)$$

$$-F_{NY}^0 A_Z - F_{TY}^0 A_Z + F_{NZ}^0 A_Y - F_{TZ}^0 A_Y - F_{NY}^1 B_Z - F_{TY}^1 B_Z + F_{NZ}^1 B_Y - F_{TZ}^1 B_Y - F_{NY}^2 C_Z - F_{TY}^2 C_Z + F_{NZ}^2 C_Y - F_{TZ}^2 C_Y + F_G T_Z = 0 \quad (51)$$

$$+F_{NX}^0 A_Z - F_{TX}^0 A_Z - F_{NZ}^0 A_X + F_{TZ}^0 A_X + F_{NX}^1 B_Z - F_{TX}^1 B_Z - F_{NZ}^1 B_X + F_{TX}^1 B_X + F_{NX}^2 C_Z - F_{TX}^2 C_Z - F_{NZ}^2 C_X + F_{TX}^2 C_X = 0 \quad (52)$$

$$-F_{NX}^0 A_Y + F_{TX}^0 A_Y + F_{NY}^0 A_X + F_{TY}^0 A_X - F_{NX}^1 B_Y + F_{TX}^1 B_Y + F_{NY}^1 B_X - F_{TX}^1 B_X - F_{NY}^2 C_Y + F_{TY}^2 C_Y + F_{NZ}^2 C_X + F_{TX}^2 C_X - F_G T_X = 0 \quad (53)$$

Jednotlivé složky normálových a tečných sil je možné spočíst pomocí následujících doplňujících rovnic.

$$F_{NX}^A = F_N^A \sin \delta, \quad F_{NY}^A = F_N^A \cos \delta, \quad F_{NZ}^A = F_N^A \sin \rho \quad (54)$$

$$F_{NX}^B = F_N^B \sin \delta, \quad F_{NY}^B = F_N^B \cos \delta, \quad F_{NZ}^B = F_N^B \sin \rho \quad (55)$$

$$F_{NX}^C = F_N^C \sin \delta, \quad F_{NY}^C = F_N^C \cos \delta, \quad F_{NZ}^C = F_N^C \sin \rho \quad (56)$$

$$F_{TX}^A = F_T^A \cos \delta, \quad F_{TY}^A = F_T^A \sin \delta, \quad F_{TZ}^A = F_T^A \cos \rho \quad (57)$$

$$F_{TX}^B = F_T^B \cos \delta, \quad F_{TY}^B = F_T^B \sin \delta, \quad F_{TZ}^B = F_T^B \cos \rho \quad (58)$$

$$F_{TX}^C = F_T^C \cos \delta, \quad F_{TY}^C = F_T^C \sin \delta, \quad F_{TZ}^C = F_T^C \cos \rho \quad (59)$$

Pomocí rovnic (48) až (59) je možné nalézt velikost reakcí v došlapech tří zvolených noh robotu, který se nachází na nakloněné rovině ve statické rovnováze. Cílem řídicího členu kontaktu by mělo být dosažení takové konfigurace, kdy budou jednotlivé síly působící v došlapech rozloženy rovnoměrně. Proto je možné formulovat výpočet potenciálové funkce řídicího členu kontaktu následovně:

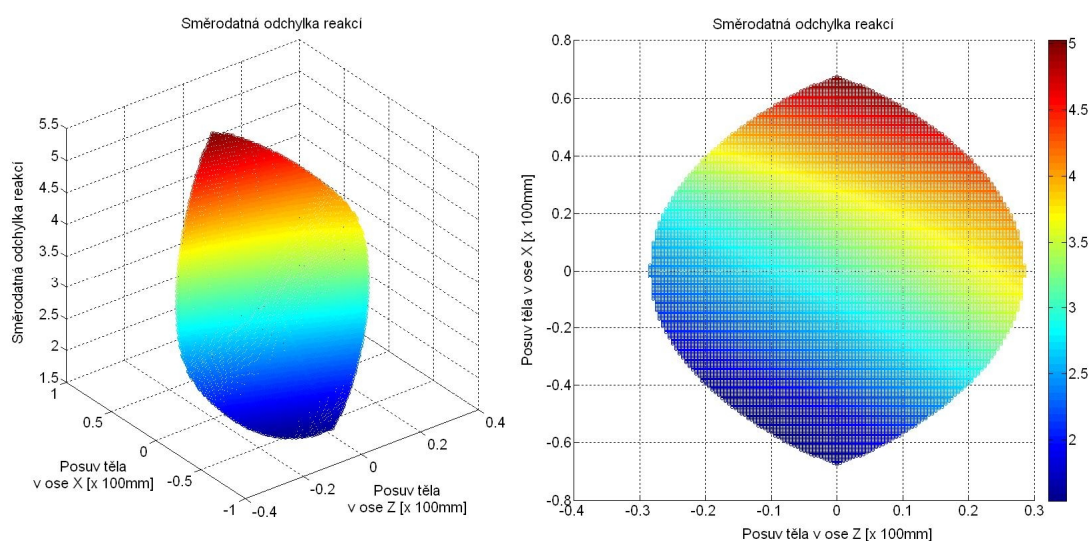
$$\Phi_K = \sigma(F_{1..n}) = \sqrt{\frac{1}{n} \sum_{i=1}^n (F_i - \bar{F})^2}, \quad n=3 \quad (60)$$

kde $\sigma(F)$ značí směrodatnou odchylku tří spočtených sil působících v došlapech robotu, F_i jsou jednotlivé síly, $\bar{F}$ je aritmetický průměr těchto sil. Průběh funkce takto navrženého řídicího členu byl opět zkoumán pro různé výchozí konfigurace robotu. Vybrané průběhy jsou zobrazeny v následující kapitole.

3.3.2.3 Posouzení charakteru použité funkce

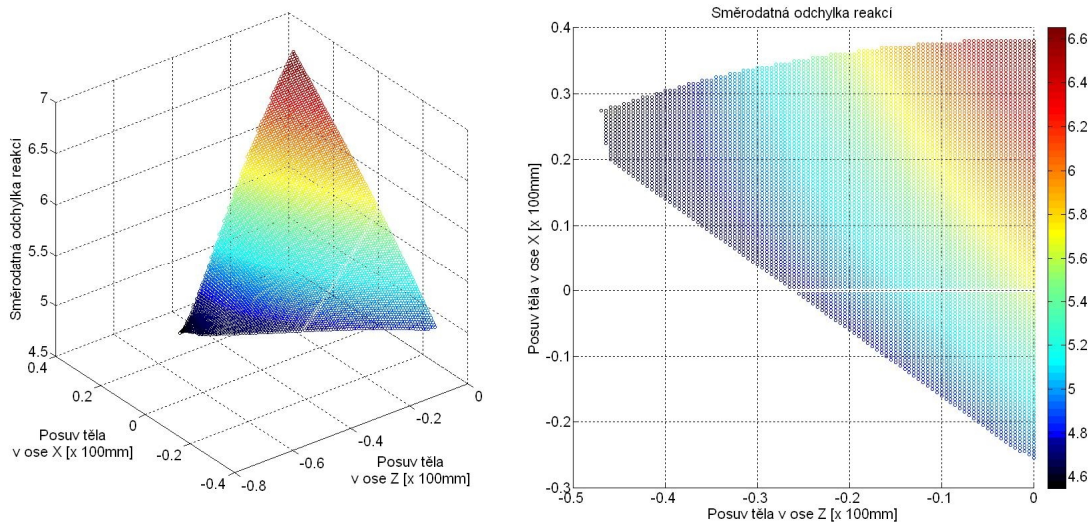
Nejprve je posuzován průběh funkce řídicího členu kontaktu, $\Phi_{K_{x,z}}^{0,2,3}$, který se snaží o minimalizaci směrodatné odchylky sil působících v došlapech noh 0,2,3 translačním pohybem těla robotu. Na Obr. 21. je tato funkce ukázána pro výchozí konfiguraci robotu č. 11. Průběh lze považovat za přijatelný, neboť funkce má právě jedno globální maximum a je klesající na celém definičním oboru. Především z pravé části Obr. 21 vyplývá, že tento řídicí člen bude při své činnosti posouvat střed těla robotu (resp. průmět středu těla do roviny kolmé na směr tíhové síly) ke středu trojúhelníku, jehož vrcholy tvoří body došlapu noh 0,2,3 (resp. průměty těchto bodů do roviny kolmé na směr tíhové síly). Řídicí člen se tak zároveň snaží o dosažení trojúhelníkové statické stability, přičemž podpůrný trojúhelník je tvořen nohami robotu, které daná instance

používá jako vstupní zdroje. Porovnáním maximálních možných translačních posuvů těla v této konfiguraci s maximálními posuvy v konfiguraci č. 1, viz. Obr. 17, lze konstatovat, že v případě konfigurace č. 11 jsou posuvy menší. Toto je dáno větší výškou těla robotu nad zemí, a tím i omezením pracovního prostoru noh. Zatímco zvětšování vzdálenosti těla robotu od podložky vedlo u řídicího členu postoje k potenciálně větší heuristické míře postoje, u řídicího členu kontaktu by tento pohyb vedl k omezení maximálních možných posuvů těla a tím i ke snížení šance na dosažení optimálního rozložení sil. Tento fakt je nutné zohlednit především při pozdějším návrhu formálních specifikací uvedených řídicích členů.



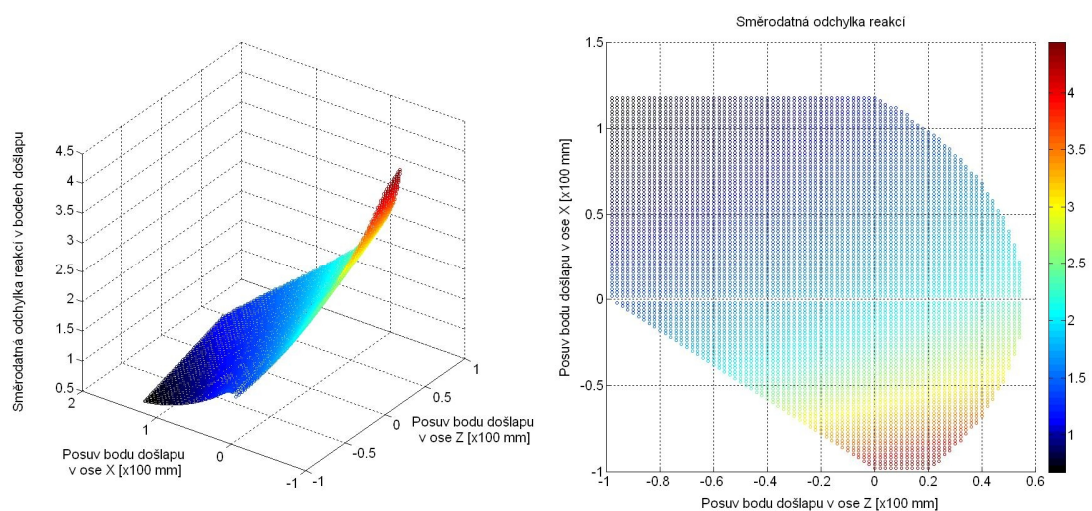
Obr. 21: Potenciálová funkce řídicího členu $\Phi_{K\underline{x,z}}^{0,2,3}$ pro konfiguraci č. 11

Na Obr. 22 je ukázán průběh funkce řídicího členu kontaktu, $\Phi_{K\underline{x,z}}^{0,2,3}$, pro méně příznivou výchozí konfiguraci č. 8. Z průběhu je patrné, že i v tomto případě je funkce klesající a má právě jedno lokální a zároveň globální minimum. I v této konfiguraci bude řídicí člen posouvat průmět těžiště robotu (v rovině kolmé na tíhovou sílu) směrem ke středu podpůrného trojúhelníku.



Obr. 22: Potenciálová funkce řídicího členu $\Phi_{K_{x,z}}^{0,2,3}$ pro konfiguraci č.8

Druhou množinu instancí řídicího členu kontaktu představují ty zákony řízení, které mění konfiguraci pouze jedné nohy za účelem dosažení optimálního rozložení sil v bodech došlapu, viz. rov. (47). Na Obr. 23 je zobrazen průběh funkce řídicího členu $\Phi_{K_I}^{0,1,2}$, který manipuluje nohou ve výchozí konfiguraci robotu č.1. V pravé části tohoto obr. je dobře vidět pracovní prostor manipulované nohy. Počátek souřadného systému je v tomto případě z důvodu přehlednosti umístěn do výchozího bodu došlapu nohy 1. V tomto konkrétním případě se bude řídicí člen snažit posunout bod došlapu o $\Delta Z = -9,8$ cm a $\Delta X = 11,8$ cm.



Obr. 23: Potenciálová funkce řídicího členu $\Phi_{K_I}^{0,1,3}$

Během procesu aktivace a konvergence instancí řídicího členu kontaktu, které mění konfiguraci jedné nohy, musí být dodrženo bezpečnostní omezení, které zamezí

pádu robotu. Toto omezení lze realizovat přímým měřením velikosti reakce v bodě došlapu nohy protilehlé vůči noze manipulované. V případě instance $\Phi_{K1}^{0,1,3}$ to je v bodě došlapu nohy č. 3.

3.3.3 Řídicí člen trajektorie

Řídicí člen trajektorie může zajišťovat globální i lokální plánování pohybu robotu, jak bylo popsáno v teoretické části, viz. kap. 2.3.1.3. Vzhledem k tomu, že lokální plánování pohybu již bylo začleněno do obou předcházejících řídicích členů, je v této práci řídicí člen trajektorie navržen pro dosahování poloh těla robotu v absolutním souřadném systému $SS^{(1)}$. Tyto polohy v globálním souřadném systému by měl poskytnout globální plánovací algoritmus. S ohledem na stanovené cíle a rozsah práce, nebyl globální plánovací algoritmus řešen. Schopnost implementované jednotky řízení má být prokázána na dvou základních typech úloh, viz. kap. 2.5.2. V případě rotace robotu na místě není využití řídicího členu trajektorie zapotřebí, neboť změnu úhlu bočení lze dosáhnout řídicím členem postoje. V případě dopředné chůze je možné v rámci procesu učení motivovat pohyb robotu směrem k cílovému bodu udělováním okamžité odměny v případě snížení vzdálenosti robotu od tohoto bodu. Navíc translační pohyb těla robotu je možné zajistit i řídicím členem kontaktu, jak bude ukázáno v kap. 3.6.2.

V počátečních fázích návrhu báze řízení byly ovšem navrženy dvě instance řídicího členu trajektorie:

$$C_6 = \left\{ \Phi_{TX,Z}^{0,1,2,3}, \Phi_{T\varphi}^{0,1,2,3} \right\} \quad (61)$$

Obě instance potřebují ke své činnosti bod v souřadném systému $SS^{(1)}$, který je poskytnut globálním plánovačem. První z uvedených instancí mění polohu robotu v osách $X^{(2)}$ a $Z^{(2)}$. Druhá instance mění úhel bočení robotu tak, aby longitudinální osa robotu směřovala do cílového bodu. Prvotní experimenty však prokázaly, že výsledné konfigurace robotu po ukončení procesu konvergence obou uvedených instancí jsou natolik nepříznivé pro činnost dalších zákonů řízení, že od jejich použití bylo ustoupeno. Návrh tohoto řídicího členu musí být, dle názoru autora, optimalizován v součinnosti s globálním plánovačem, který ovšem není v této práci řešen s ohledem na stanovené cíle a rozsah práce.

3.4 Projekt implementující navržené řešení

Předchozí kapitola 3.3 se zabývala návrhem báze řízení tvořící základ řídicí jednotky robotu. Po dokončení této první fáze, bylo nutné implementovat pomocí zvolených prostředků, viz. kap. 3.1, všechny tři vrstvy architektury založené na paradigmatu DEDS, jejíž teoretický rozbor je uveden v kap. 2.3 a to včetně algoritmu Q-učení, viz. kap. 2.4.4. Pro tento účel byl sestaven komplexní software projekt, jehož hlavním účelem je:

- ověření funkčnosti koordinačního, resp. plánovacího schématu
- ověření funkčnosti vytvořeného modelu robotu, viz. kap. 3.2.3
- zjištění výsledných elementů chování robotu pomocí navržených řídicích členů
- nalezení vhodného schématu paralelních aktivací kompozitních řídicích členů

- nalezení parametrů algoritmu Q-učení pro řešení zadaných úloh robotu
- posouzení výsledných chování robotu v rámci řešených úloh, viz. kap. 2.5.2

Pro dosažení těchto cílů bylo nezbytné:

- algoritmizovat chování všech navržených instancí základních řídicích členů
- propojit činnost těchto zákonů řízení s modelem robotu
- zajistit automatické sestavení kompozitních řídicích členů
- implementovat koordinační mechanismus DEDS pomocí stavového automatu
- implementovat algoritmus Q-učení a proces učení propojit s podřízenou vrstvou
- zajistit grafické zobrazení chování robotu
- zajistit ukládání a načítání simulace a jejich parametrů

V následující kapitole 3.4.1 je stručně popsána funkčnost programu a způsob jeho využití. V kap. 3.4.2 je pak uvedena struktura programu a jsou zde rozebrány jednotlivé moduly, které tvoří projekt. Způsob implementace vybraných problémů je uveden v kap. 3.4.3. Výsledky, kterých bylo dosaženo tímto programem jsou diskutovány až v následujících kapitolách, především kap. 3.5 a 3.6.

3.4.1 Popis činnosti programu

Všechny tyto dílčí cíle a požadavky uvedené v předchozí kapitole byly splněny a jsou do projektu zapracovány. Projekt umožňuje na vstupu načíst model prostředí a model simulovaného robotu. Tento vstup může být realizován buďto programově, použitím modulu napsaného v jazyce python, nebo pomocí standardizovaného formátu XODE. Jedná se o textový soubor ve formátu XML obsahující přesně definované údaje, příklad vstupu je uveden na Obr. 24.

<pre> - <code> - <world name="world"> - <space name="space"> - <geom name="simFloor,255,32,128"> <plane a="0" b="1" c="0" d="0" /> </geom> - <geom name="gBox1,192,192,192"> - <transform> <position x="-16.9" y="1.25" z="-16.9" /> - <rotation> <euler x="0" y="0" z="0" aformat="degrees" /> </rotation> </transform> <box size="9.8" sizey="2.5" sizez="0.2" /> </geom> - <geom name="gBox2,192,192,192"> </pre>	<pre> body,geom = create_box(space,world, rstoj_x,rstoj_y,rstoj_z, mStojka,55,255,0) body.setPosition((zslz_x,zslz_y,zslz_z)) body.name = "IzStojkaZadani" listOfBodies.append(body) listOfGeoms.append(geom) t1 = getTelesoByName(listOfBodies, "IzKycel") t2 = getTelesoByName(listOfBodies, "IzStojkaPredni") vazba = ode.HingeJoint(world) vazba.attach(t1,t2) vazba.setAnchor((j_kzl_pslz_x,j_kzl_pslz_y,j_kzl_pslz_z)) vazba.setAxis((1,0,0)) vazba.setParam(ode.ParamLoStop, stojka_dolu_leva) vazba.setParam(ode.ParamHiStop, stojka_nahoru_leva) vazba.setParam(ode.ParamVel,0) vazba.setParam(ode.ParamFMax,fMmax) </pre>
--	---

Obr. 24: Příklad vstupu programu, a) Xode formát prostředí b) přímá tvorba modelu

Po načtení zadaného modelu prostředí a modelu simulovaného robotu, je nutné, aby uživatel zadal typ řešené úlohy. Předdefinovány jsou úloha rotace robotu kolem osy $Y^{(2)}$ a úloha dopředné chůze. Další činnost programu již probíhá bez zásahu uživatele. Na základě zadané úlohy vybere program předdefinované predikáty stavového vektoru, q , a množinu instancí základních řídicích členů včetně jejich specifikací. Z této množiny je automaticky sestavena množina kompozitních řídicích členů včetně jejich specifikací pomocí rovnic (10) až (18). Následně je sestaven model systému zahrnující všechny možné stavy a přechody systému. Postup vytvoření modelu vychází

z teoretického rozboru podaného v kap. 2.3.2. Model poskytuje pro každý stav množinu přípustných aktivací základních i složených řídicích členů. Tento model je po vytvoření uložen v podobě textového souboru na pevný disk, viz. Obr. 25.

```
ID01 [True False True False 0 0 ]
0 řídicí člen postoje s rotací těla
1 řídicí člen kontaktu s pohybem nohy, troj: [0, 2, 3], noha: 2
2 řídicí člen kontaktu s pohybem nohy, troj: [1, 2, 3], noha: 3
3 řídicí člen postoje s translací v ose Y
4 řídicí člen postoje se změnou klonení/klopení
5 kompozitní řídicí člen:
  pod-->
  řídicí člen postoje s rotací těla
  nad-->
  řídicí člen kontaktu s pohybem nohy, troj: [0, 2, 3], noha: 2
6 kompozitní řídicí člen:
  pod-->
  řídicí člen postoje s rotací těla
  nad-->
  řídicí člen kontaktu s pohybem nohy, troj: [1, 2, 3], noha: 3
7 kompozitní řídicí člen:
  pod-->
  řídicí člen postoje s translací v ose Y
  nad-->
  řídicí člen postoje se změnou klonení/klopení
```

Obr. 25: Ukázka uložených informací o modelu chování systému

Následuje zahájení procesu online Q-učení, během kterého se řídicí jednotka snaží nalézt optimální posloupnost aktivací pro dosažení cíle řešené úlohy. Během procesu je zobrazován jak grafický tak textový výstup. Grafický výstup ukazuje model robotu v daném prostředí a jeho chování v reálném čase pod vlivem aktivací jednotlivých zákonů řízení. Textový výstup během procesu učení poskytuje i tyto informace (viz. též obr. Obr. 26):

- číslo iteračního kroku
- aktuální stavový vektor popisující stav robotu
- informaci o aktivaci vybraného zákona řízení
- průběh a výsledek procesu konvergence zvoleného zákona řízení
- informace použité pro výpočet okamžité odměny
- velikost okamžité odměny za provedení zvolené akce řízení

```
Prave byla spustena 4 . aktualizace Qhodnoty v procesu Quceni.
Volba akce ve stavu [True, False, True, False, 1, 0] ID: 2
pokus 0
Aktivace zakladni:
ridici clen postoje s translaci v ose Y
nejnizsi vrstva --> aktivace ridciho clenu postoje se zmenou vysky
konv: iteracni krok ridciho clenu POSTOJE s translaci v Y
Jeden krok aktivace controlleru byl ukoncen, celkem provedeno 20 simulacnich kroku ODE.
konv: iteracni krok ridciho clenu POSTOJE s translaci v Y
dalsi iteracni krok neprinese zlepšení
index vybrane akce 2
deltaUhlv -0.0255287585205 [st.]
odmena -1
prepnuto do stavu: [True, False, True, False, True, True] ID: 3
```

Obr. 26: Příklad textového výstupu během procesu učení

Předkládaný projekt není opatřen grafickým uživatelským rozhraním. Jeho použití vyžaduje alespoň základní znalosti programování. Veškerá základní funkcionalita projektu je ovšem poskytována co nejjednodušším způsobem prostřednictvím vysokoúrovňových funkcí. U všech funkcí je uveden komentář. Autor předpokládá další využití vytvořeného programu jak pro vědecké tak i pedagogické účely.

3.4.2 Struktura projektu

Celý projekt je naprogramován objektově pomocí jazyka Python s využitím dalších nástrojů a knihoven uvedených v kap. 3.1. Při návrhu projektu byl kladen maximální důraz na obecnost a modifikovatelnost v případě změny požadavků. Toto tvrzení dokazuje i fakt, že předkládaný projekt byl ve starší verzi použit např. i pro simulaci dynamické chůze čtyřnohého robotu Qasimodo v práci [33]. Výsledné řešení projektu je rozděleno do několika vzájemně propojených modulů:

- modul zajišťující práci s knihovnou ODE
- modul poskytující parametrický model robotu
- modul zajišťující grafický výstup
- modul elementárních řídicích členů
- modul koordinačního mechanismu a algoritmu Q-učení

Modul zajišťující práci s ODE obstarává inicializaci a nastavení parametrů simulace realizované pomocí knihovny *Open Dynamics Engine*. Dále poskytuje důležitou mezivrstvu pro jednodušší vytváření těles, přidávání vazeb do simulace a nastavování jejich parametrů. Výrazně tak zjednodušuje následnou práci s návrhem modelu zvoleného robotu a jeho import do simulovaného prostředí a také při spouštění či ukončování simulace. Do tohoto modulu patří třídy *Sim* a *TaskToSim*.

Modul poskytující parametrický model robotu zajišťuje vytvoření modelu robotu z poskytnutého vstupního souboru, který může být zadán ve formátu XODE nebo ve formátu přímo psaného kódu, jak bylo uvedeno v předchozí kapitole. Platforma použitá pro verifikaci i její model je podrobně diskutována v kap. 3.2. Modul je tvořen dvěma zdrojovými soubory *model.py* a *params.py*.

Modul zajišťující grafický výstup je vystavěn na použití knihovny OpenGL prostřednictvím modulu *pyOpenGL*. Veškerá funkcionality je umístěna do třídy *PaintSim*. Jak již bylo uvedeno při diskusi použitých prostředků, fyzikální engine ODE neobsahuje žádné nástroje pro zobrazení simulované scény, veškeré vykreslování simulovaného robotu a prostředí tak musely být obhospodařeny ve vlastní režii, což přineslo mnoho dílčích problémů, které musely být řešeny.

Modul elementárních řídicích členů je tvořen dvěma třídami. Třída *Configuration* zjišťuje informace o aktuální konfiguraci robotu a předává tyto informace nejnížší vrstvě architektury řízení. Tato vrstva je implementována v třídě *Controller*, která zajišťuje výpočty nutné pro činnost jednotlivých zákonů řízení, popsanych v kap. 3.3.1.1 a 3.3.2.1. Tato třída rovněž zajišťuje provádění akčních zásahů na použitém modelu.

Implementace dvou nadřazených vrstev je obsažena v posledně jmenovaném modulu, který je tvořen především třídou *CoordMech*. V této třídě je algoritmizován postup sestavení modelu všech možných přechodů systému na základě přípustných aktivací a také algoritmus Q-učení. Dvě hlavní vlastnosti třídy jsou seznam základních řídicích členů a seznam složených řídicích členů. Třída *BaseController* reprezentuje základní řídicí člen, jeho vstupní a výstupní zdroje a především specifikace S, M, R . Třída *CompositController* je použita pro reprezentaci složeného řídicího členu. Obsahuje informace o podřízeném a nadřízeném řídicím členu a o specifikacích, které jsou v tomto

případě vyvozovány zcela automaticky. Rovněž je zde algoritmován proces aktivace a konvergence kompozitního řídicího členu.

3.4.3 Způsob implementace vybraných problémů

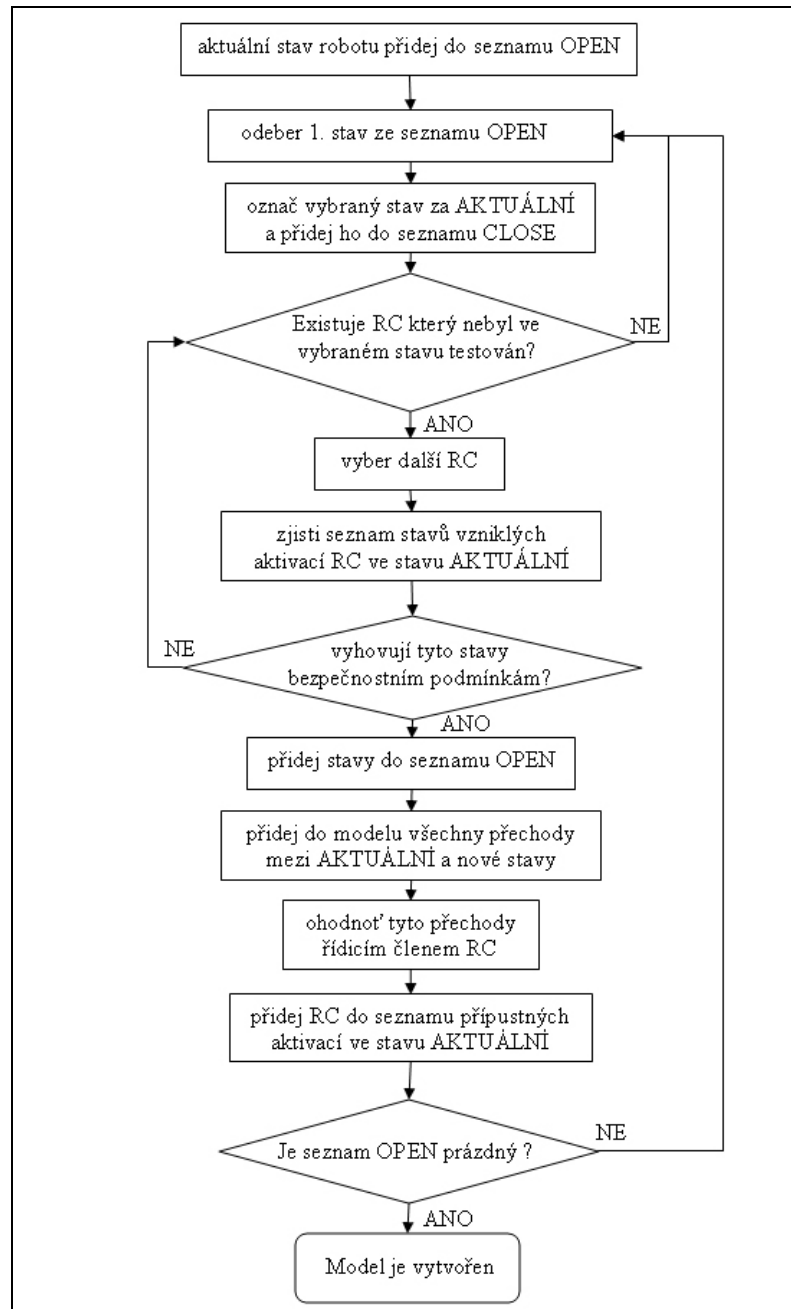
3.4.3.1 Sestavení koordinačního mechanismu

Pojmem koordinační mechanismus je myšlen model všech možných přechodů systému pod vlivem přípustných aktivací základních i složených řídicích členů. Teoretický rozbor problematiky je podán v kap. 2.3.2 zabývající se návrhem prostřední vrstvy architektury řízení. Sestavení modelu je v předkládaném projektu algoritmováno v metodě *CreateModel* třídy *CoordMech*.

Činnost algoritmu je schématicky naznačena na Obr. 27., kde RC označuje řídicí člen. Algoritmus používá seznam stavů systému OPEN pro uchování stavových vektorů, které mají být zkoumány. Seznam CLOSED obsahuje již zpracované stavy. Uvedené bezpečnostní podmínky jsou logické výrazy obsahující jednotlivé predikáty stavového vektoru, viz. teoretický rozbor v kap. 2.3.2.2. Ze zápisu algoritmu vyplývá, že v žádném případě není připuštěna aktivace řídicího členu, která by mohlo potenciálně vést systém do stavu, ve kterém nejsou zmíněné bezpečnostní podmínky dodrženy. Seznam stavů, do kterých může aktivace řídicího členu vést, lze vyvodit ze specifikace M řídicího členu. Tento vektor totiž udává, které predikáty tvořící popis systému jsou měněny.

3.4.3.2 Činnost základních řídicích členů

V předkládaném projektu je algoritmována činnost všech instancí řídicích členů popsaných v kapitole 3.3. Jedná se o množiny C_1 až C_6 , viz. rov. (29), (30), (31), (46), (47) a (61), obsahující celkem 28 různých instancí základních řídicích členů. Algoritmy, zjišťující provádění nezbytných výpočtů jsou umístěny ve třídě *Controller*. Konkrétně se jedná o 6 různých funkcí, které přebírají parametry v podobě konkrétních vstupních a výstupních zdrojů σ, τ . Pro každou z uvedených množin instancí je tedy určena právě jedna funkce třídy *Controller*. Jednotlivé funkce se liší způsobem sestavení potenciálové funkce, nicméně hledání extrému této funkce probíhá vždy stejným způsobem prostřednictvím gradientní metody. Nalezení globálního extrému je zajištěno díky vyhovujícímu průběhu všech navržených funkcí, viz. kap. 3.3. Gradientní metoda postupuje směrem ke globálnímu extrému iterativně ve směru největšího spádu. Každému bodu plochy funkce vždy odpovídá jedinečná konfigurace robotu, která má být v daném iteračním kroku dosažena. Při aktivaci základního řídicího členu je spuštěn proces hledání globálního extrému a činnost je ukončena až v okamžiku jeho nalezení. Naproti tomu v případě aktivace kompozitního řídicího členu je typicky proveden vždy pouze jeden krok iterativního procesu hledání globálního extrému. Způsobu algoritmování činnosti složených řídicích členů je věnována následující kapitola.

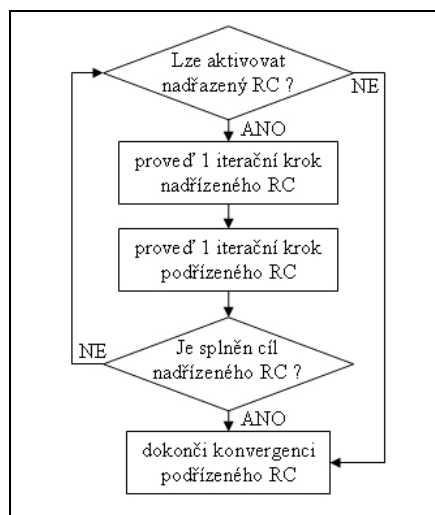


Obr. 27: Algoritmus sestavení modelu všech možných chování systému

3.4.3.3 Činnost složených řídicích členů

V práci jsou použity složené řídicí členy tvořené vždy právě dvěma instancemi základních řídicích členů. Jejich vzájemný vztah je vymezen operátorem podřízenosti, viz kap. 2.3.2.1. Na základě uvedeného teoretického rozboru fungování složených řídicích členů, byl navržen následující algoritmus činnosti složeného řídicího členu, viz. vývojový diagram Obr. 28. Nejprve je opakovaně aktivován nadřízený a po té podřízený řídicí člen, přičemž je provedena vždy jen jedna iterace. Tímto je zajištěna paralelní činnost, která

vede ke složitějšímu chování robotu. Iterace podřízeného řídicího členu je realizována jen v případě, že nedojde ke snížení hodnoty potenciálové funkce nadřazeného řídicího členu. Tato první fáze je ukončena v okamžiku, kdy je splněn cíl nadřazené instance resp. již není možná jeho aktivace z důvodu nesplněných podmínek aktivace daných vektorem S . V následující druhé fázi činnosti kompozitního členu je opakovaně aktivován již pouze podřízený řídicí člen, jehož aktivace je ukončena v okamžiku, kdy by došlo k porušení již splněného cíle nadřazeného řídicího členu nebo není možné aktivaci provést s ohledem na podmínky aktivace.



Obr. 28: Algoritmus činnosti kompozitního řídicího členu

3.4.3.4 Implementace učicího algoritmu

Pro realizaci algoritmu Q-učení byl použit iterativní výpočet Q hodnot dle rov. (23). Každá z hodnot přísluší dvojici stav, akce. Stav je reprezentován stavovým vektorem, q , a akce je reprezentována jednou z přípustných instancí řídicích členů v tomto stavu. Seznam přípustných aktivací byl vyvozen postupem uvedeným v kap. 3.4.3.1 a je uchováván ve třídě *SysState*, stejně tak jako seznam aktuálních Q hodnot. Dále je rovněž ukládán počet provedených aktualizací jednotlivých Q hodnot a to z důvodu dodatečné analýzy procesu učení. Z důvodů předejití neočekávané ztráty dat je seznam Q hodnot a počtu aktualizací ukládán průběžně na disk. V případě přerušení simulace jsou naučené hodnoty automaticky použity jako výchozí při znovuspuštění procesu učení. Toto bezpečnostní opatření slouží především pro zvýšení robustnosti navrženého řešení. Během provádění testovacích úloh však nedocházelo k pádu programu a simulace byla stabilní.

Proces učení probíhá online, tj. přímo během činnosti robotu, a to tak dlouho, dokud nedojde ke splnění cíle nebo není proces ukončen uživatelem. Vlastní algoritmus je uveden v metodě *RunQLearning* třídy *CoordMech*. V této metodě jsou ve smyčce opakovány tyto kroky:

- zjištění množiny přípustných řídicích členů pro aktuální stav z modelu chování
- aktivace řídicího členu s největší uloženou Q -hodnotou

- po ukončení konvergence určení stavového vektoru nového stavu robotu
- aktualizace příslušné Q hodnoty dle rov. (23)

Metoda přebírá na vstupu dva parametry. Učící faktor *alfa* a srážkový faktor *gama*. Konkrétní nastavení těchto parametrů, stejně jako další výsledky procesu učení jsou uvedeny v následujících kapitolách, především 3.6.1.2 a 3.6.2.2.

3.5 Výsledné elementy chování robotu

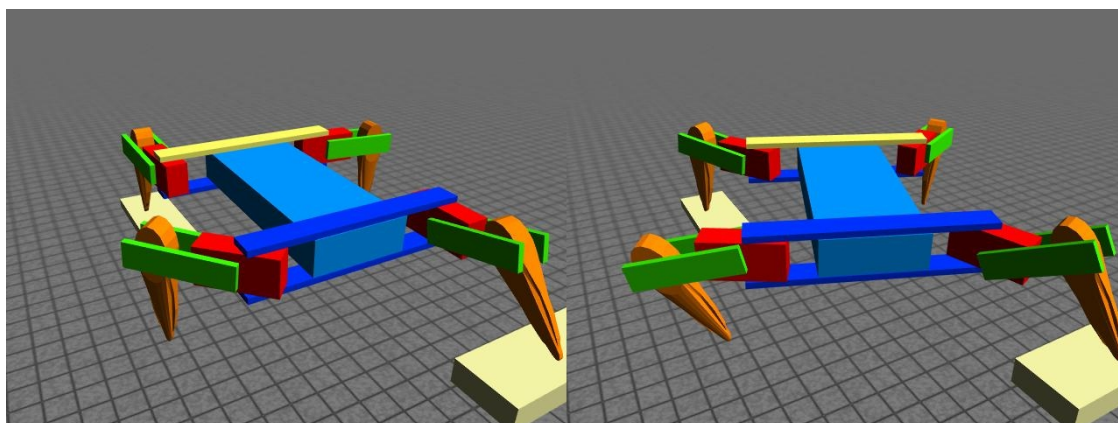
V předchozí kapitole 3.4 je popsána činnost, struktura a implementace vytvořeného software, který je určen pro plánování pohybu robotu pomocí komplexní třívrstvé architektury řízení založené na paradigmatu DEDS. V této kapitole jsou vyhodnoceny výstupy popsaného software související především sází řízení, kap. 3.5.1, a zajišťováním složitějších chování robotu prostřednictvím kompozitních řídicích členů, kap. 3.5.2, a také s funkčností modelu a simulace jako celku. Schopnost algoritmu učení nalézt optimální strategie řízení pro řešení komplexních úloh je diskutována v následující kapitole 3.6.

U předloženého software projektu byla nejprve testována stabilita simulace a parametry jejího nastavení. Při této činnosti byla využita zkušenost autora s řešením dřívějších projektů, zejména simulace dynamické chůze robotu Qasimodo, viz. kap. 3.1. Lze konstatovat, že simulace nevykazuje známky nestability. Během manipulace s robotem nedochází např. k explozi vazeb, což je častý problém u mnoha nesprávně navržených řešení. Vzájemná kolize těles rovněž funguje bez výhrad, nedochází k vzájemnému prolínání tuhých těles. Vytvořené kinematické vazby mezi tělesy jsou schopny přenést vznikající silová působení. Tato tvrzení podporují i videozáznamy simulace, na které je odkazováno v následujících kapitolách. Po provedení výchozích dlouhodobých testů, během kterých byl robot pouze umístěn na vodorovnou podložku, nakloněnou rovinu a později do nerovného prostředí, bylo přistoupeno k testování činnosti jednotlivých členů báze řízení.

3.5.1 Chování základních řídicích členů při výskytu překážek

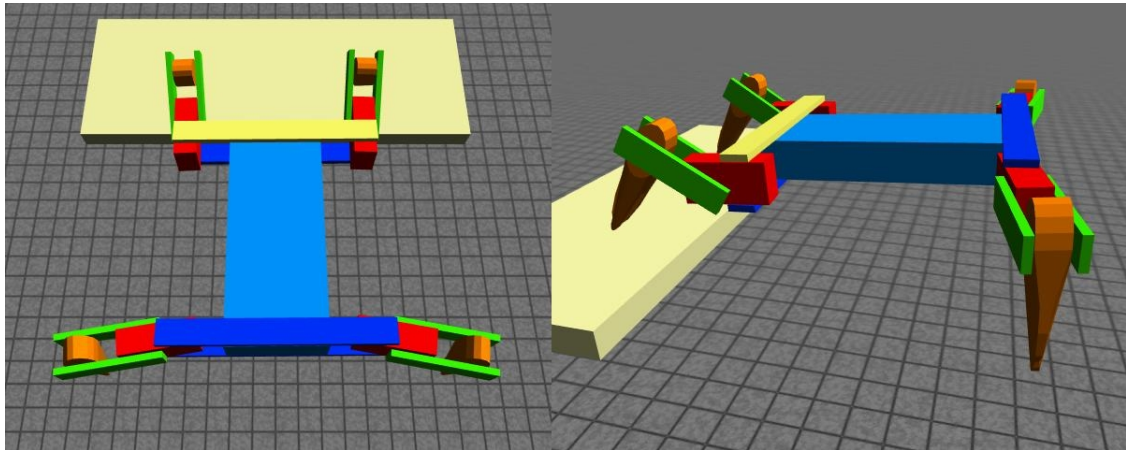
V této práci jsou použity celkem tři řídicí členy (postoje, kontaktu a trajektorie). Řídicí člen je dán obecným předpisem své potenciálové funkce, konkrétní chování robotu však nezajišťuje. Jednotlivé konkrétní elementy chování zajišťují až instance těchto řídicích členů. Jedná se o zákony řízení, které jsou určeny propojením vybraných vstupních a výstupních zdrojů robotu s obecným předpisem řídicího členu. Pro potřeby plánování pohybu robotu bylo v této práci navrženo celkem 28 různých instancí základních řídicích členů, které jsou rozděleny dle návrhu potenciálové funkce do 6 množin C_1 až C_6 , představených v kap. 3.3. Níže jsou popsána výsledná simulovaná chování robotu pod vlivem aktivace a konvergence vybraných zástupců z těchto množin. Slovní popis je doplněn o video soubory, umístěné v příloze A, zachycující pohyb robotu zpomaleně o 50% vůči reálnému průběhu především z důvodu snadnější analýzy pozorovatelem. Uvedené výstupy byly získány za použití vytvořeného software řešení, popsaného v kap. 3.4.

Množina C_1 , viz. rov. (29), obsahuje instance řídicího členu postoje, které jako výstupní zdroj používají jeden ze tří úhlů natočení těla robotu. Např. zákon řízení $\Phi_{P\varphi}^{0,1,2,3}$ mění úhel bočení těla. Pohyb robotu v členitém prostředí pod vlivem aktivace a konvergence tohoto zákona řízení je zachycen na videozáznamu, viz. příloha A soubor *postoje_boceni_1.avi*. Výchozí a cílová konfigurace je ukázána také na Obr. 29, kde žlutou barvou jsou naznačeny překážky simulující nerovnost terénu. Ze záznamu chování plyne, že tělo robotu rotuje kolem svého středu, tj. osy $Y^{(2)}$. Během pohybu se mění velikost všech úhlů na všech nohách robotu. Nedochází však ke změně bodů došlapu během celé doby rotace.



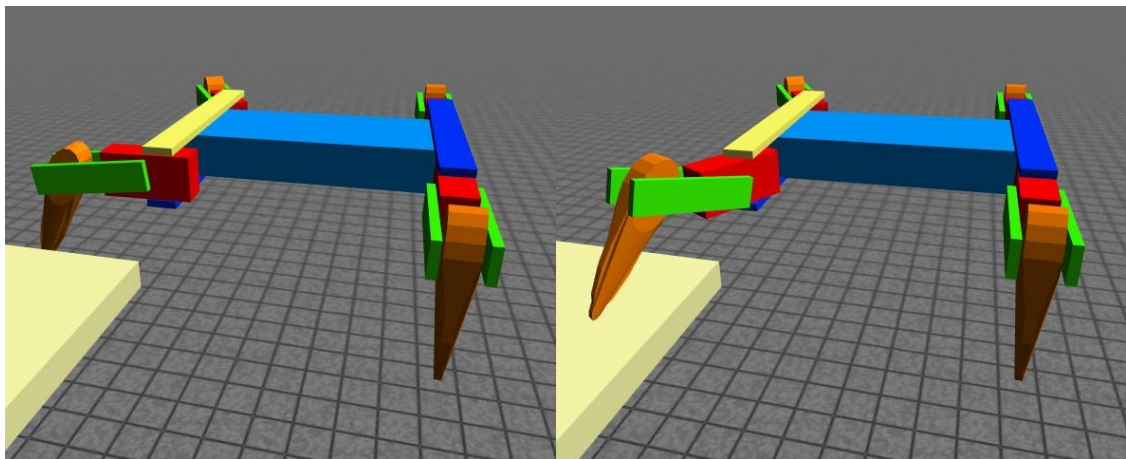
Obr. 29: Vliv aktivace $\Phi_{P\varphi}^{0,1,2,3}$ v členitém terénu, výchozí a cílová konfigurace

Instance řídicího členu postoje, které maximalizují heuristickou míru postoje, m , pomocí translačního pohybu těla robotu jsou obsaženy v množině C_2 , viz. rov. (30). Např. instance $\Phi_{P_{x,z}}^{0,1,2,3}$ realizuje pohyb těla v osách $X^{(2)}, Z^{(2)}$. Pro ukázání průběhu konvergence tohoto řídicího členu je použita výchozí konfigurace, ve které jsou přední nohy robotu, tj. nohy 1,2, umístěny na překážce, viz. Obr. 32 vlevo. Výsledná konfigurace robotu je ukázána ve dvou různých pohledech na Obr. 30. Pohyb robotu po aktivaci toho řídicího členu je zachycen na videozáznamu, viz. příloha A soubor *postoje_translace_1.avi*. V koncové konfiguraci se polohy jednotlivých článků nohy blíží svým středním polohám. Robot se tak nachází ve výhodnějším postavení pro další pohyb, během něhož bude např. překonávat uvedenou překážku.



Obr. 30: Výsledek konvergence instance $\Phi_{p_{x,z}}^{0,1,2,3}$, různé pohledy

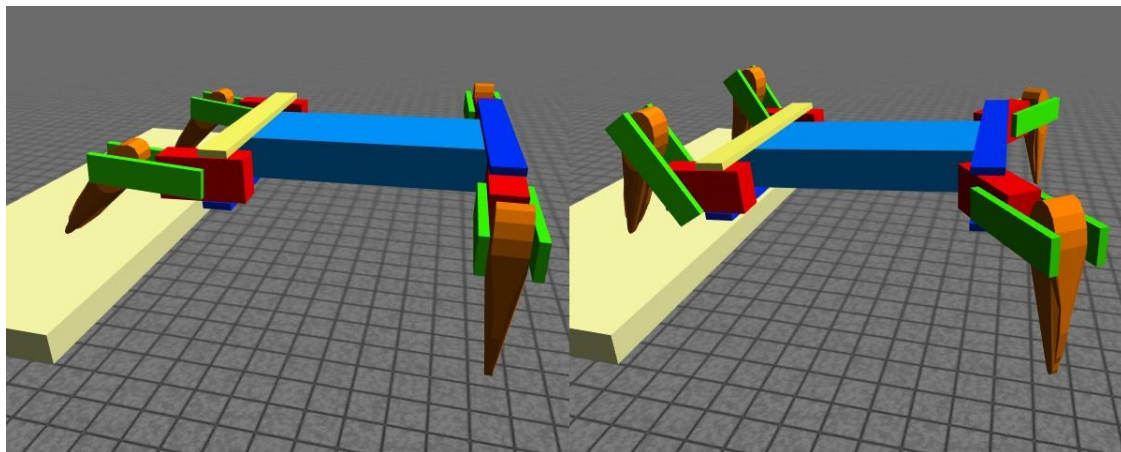
Poslední množina instancí řídicího členu postoje, C_3 , obsahuje zákony řízení, které mění konfiguraci pouze jedné nohy, viz. rov.(31). Např. zákon řízení $\Phi_{p_2^2}$ se snaží o dosažení optimální konfigurace nohy 2. Průběh činnosti je patrný z vidozáznamu, viz. příloha A soubor *postoje_noha_1.avi*. Výchozí a cílová konfigurace je ukázána na Obr. 31. Během své činnosti manipuluje řídicí člen nohu přes neznámou překážku (těleso žluté barvy). Koncová hodnota úhlů nohy 2 je $\alpha^2 = 30.2^\circ, \beta^2 = 14.1^\circ, \gamma^2 = 22.2^\circ$. Tyto hodnoty se výrazně přibližují středním hodnotám, viz. rov. (27). Právě z důvodu překážky není možné docílit ideálního postavení. Např. změna úhlu γ^2 by výrazně zhoršila nastavení úhlu β^2 .



Obr. 31: Vliv aktivace $\Phi_{p_2^2}$ v členitém terénu, výchozí a cílová konfigurace

Instance řídicího členu kontaktu, které se snaží o dosažení optimálního rozložení sil v došlapech prostřednictvím translačního pohybu těla robotu, jsou obsaženy v množině C_4 , viz. rov. (46). Např. zákon řízení $\Phi_{K_{x,z}}^{1,2,3}$ se snaží dosáhnout optimálního rozložení sil v bodech došlapu noh 1,2,3. Jeho činnost je zachycena na vidozáznamu,

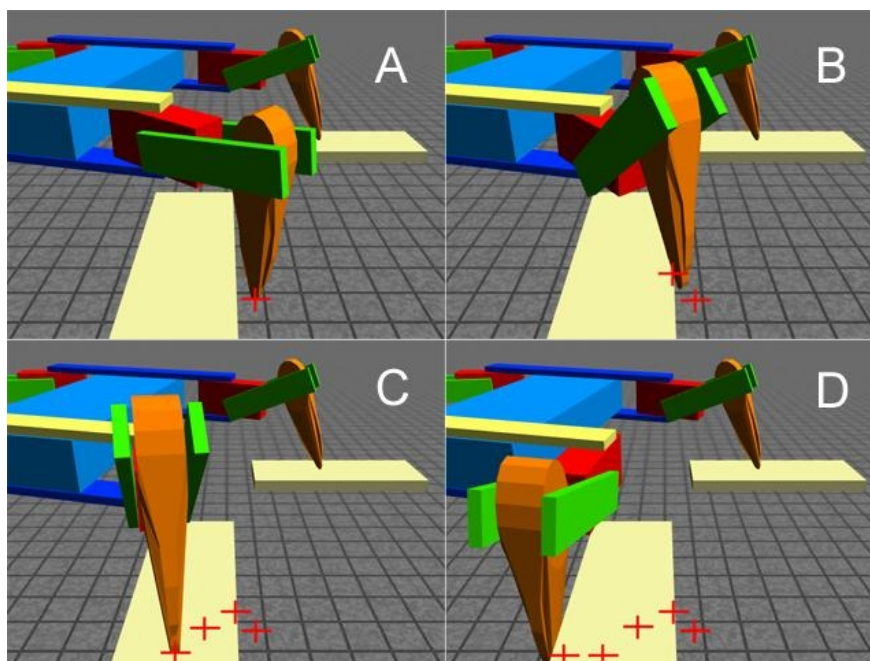
viz. příloha A soubor *kontakt_translace_1.avi* a také na Obr. 32. Pro tento element chování je použito stejné výchozí konfigurace jako u příkladu činnosti instance řídicího členu postoje $\Phi_{P_{x,z}}^{0,1,2,3}$. Cílová konfigurace zákona řízení $\Phi_{K_{x,z}}^{1,2,3}$ je méně výhodná pro další pohyb vzhledem k nevýhodnému postavení noh, ovšem průmět těžiště robotu do roviny podložky se nachází uvnitř podpůrného trojúhelníku noh 1,2,3.



Obr. 32: Vliv aktivace $\Phi_{K_{x,z}}^{1,2,3}$ v členitém terénu, výchozí a cílová konfigurace

Další množinu, C_5 , zákonů řízení představují instance řídicího členu kontaktu, které jako výstupní zdroj používají pouze jednu nohu, viz. rov. (47). Pro změnu konfigurace nohy v neznámém prostředí je použito stejného algoritmu jako v případě manipulace nohou u instancí řídicího členu postoje v množině C_3 . I když je v tomto případě využíváno jiné potenciálové funkce. Jedná se o pohyb nohy, během kterého je měněn bod došlapu o malou vzdálenost směrem k předpokládanému globálnímu extrému funkce. Tento směr je po každém došlapu průběžně upravován dle zjištěného charakteru terénu. Jedná se tedy o reaktivní chování. Podrobně byl tento postup popsán v kap. 3.3.1.3. Výsledná trajektorie koncového bodu došlapu vlivem aktivace instance $\Phi_{K_2}^{0,2,3}$ je ukázána na Obr. 33 a rovněž na videozáznamu, viz. příloha A soubor *kontakt_noha_1.avi*.

Poslední množina C_6 obsahuje dvě instance řídicího členu trajektorie, viz. rov. (61). Jedná se o zákony řízení které pohybují tělem robotu směrem k dílčímu cíli, který je reprezentován bodem v prostoru. Např. instance $\Phi_{T_\phi}^{0,1,2,3}$ mění úhel bočení těla robotu tak, aby longitudinální osa robotu směřovala do cílového bodu. Průběh činnosti tohoto řídicího členu je zachycen na videozáznamu, viz. příloha A soubor *trajektorie_boceni_1.avi*. Na záznamu je vidět, že koncová konfigurace robotu, může být velmi nepříznivá, jelikož nohy robotu jsou v tomto konkrétním případě umístěny téměř do svých mezních poloh. Výběr zákonů řízení v dalším časovém okamžiku je tak velmi omezen. Toto je hlavní důvod, kvůli kterému nebyly řídicí členy trajektorie zařazeny do výchozí množiny řídicích členů při řešení komplexních úloh.



Obr. 33: Reaktivní chování řídicího členu $\Phi_{K2}^{0,2,3}$ při výskytu neznámé překážky

Informace uvedené v této kapitole ukazují splnění několika dílčích cílů. Především zjištění výsledných elementů chování robotu pomocí navržených základních řídicích členů, dále zajištění reaktivního chování robotu v případě výskytu překážek v bezprostřední blízkosti robotu, a rovněž funkční návrh řídicího členu kontaktu schopného činnosti ve strukturovaném prostředí. Průběh chování příslušných zákonů řízení je ukázán jednak na vložených obrázcích a jednak na videozáznamech uvedených v příloze A této práce. Ze záznamů je rovněž patrné, že model robotu je navržen vhodným způsobem a pro potřeby simulace je dostačující. Uvedené prvotní testy poukázaly na nutnost změny návrhu modelu koncového článku nohy. V práci je uvedena již pouze jeho finální podoba.

Uvedené instance řídicích členů byly dále použity pro automatické vyvození složených řídicích členů, jejichž výsledná chování jsou ukázána v následující kapitole.

3.5.2 Chování kompozitních řídicích členů

V práci je sestaveno a popsáno celkem 28 různých instancí zajišťujících elementární chování robotu. Na základě jejich specifikací může být sestaveno až 784 kompozitních řídicích členů, které jsou tvořeny vždy dvěma základními členy. Velké množství z uvedeného počtu kompozitních členů je nevyhovující. Sestavení množiny složených řídicích členů a vyřazení těch nevyhovujících je prováděno automaticky v rámci prostřední vrstvy architektury řízení. Ve výsledných modelech všech možných chování robotu je zahrnuto jen asi 15% kompozitních řídicích členů z maximálního možného počtu, jak bude ukázáno v následující kapitole 3.6. Činnost těch nejčastěji používaných je popsána v této kapitole.

Řídicí člen $\Phi_{p_y}^{0,1,2,3} < \Phi_{p_\varphi}^{0,1,2,3}$ zajišťuje změnu úhlu bočení těla robotu a zároveň mění výšku těla robotu nad zemí. Obě složky maximalizují heuristickou míru postoje. Výsledné konfigurace robotu je díky paralelní aktivaci dosaženo během kratší doby. Je však také pravděpodobné, že proces konvergence bude ukončen ve stavu s vyšší hodnotou heuristické míry postoje než v případě oddělených aktivací. Změna výšky těla robotu nad zemí totiž může umožnit větší změnu úhlu bočení. Výsledné chování robotu ukazuje videozáznam, viz. příloha A soubor *postoj_rotace_vyska.avi*.

Dobrych výsledků je dosahováno také pomocí členu, kde je změna úhlu bočení doplněna o změnu konfigurace jedné z noh. Např. instance $\Phi_{K0}^{0,2,3} < \Phi_{p_\varphi}^{0,1,2,3}$ paralelně dosahuje optimálního rozložení reakcí v nohách 1,2,3 pohybem nohy 3. V tomto případě bude řídicí člen vykazovat dobrých výsledků, pokud bude probíhat změna úhlu bočení v kladném směru. V opačném směru by činnost obou řídicích členů byla protichůdná. O tom, zda bude tento kompozitní řídicí člen použit ve výsledném chování, pak rozhodují formální specifikace obou složek a velikost okamžité odměny udělené algoritmem Q-učení. Výsledné chování je zachyceno na videozáznamu, viz. příloha A soubor *kontakt_noha_postoj_rot1.avi*.

Další často používané kompozitní členy, $\Phi_{p_y}^{0,1,2,3} < \Phi_{p_z}^*$, se snaží o dosažení optimální konfigurace jedné nohy při současně změně výšky těla robotu nad zemí. Rovněž podřízený řídicí člen maximalizuje heuristickou míru postoje a proto svojí činností přímo podporuje činnost nadřízeného členu. Průběh konvergence členu $\Phi_{p_y}^{0,1,2,3} < \Phi_{p_0}^0$ je zachycena na videozáznamu, viz. příloha A soubor *postoj_vyska_noha.avi*.

Řídicí člen $\Phi_{p_y}^{0,1,2,3} < \Phi_{Kx,z}^{1,2,3}$ se snaží o optimální rozložení sil v došlapech a zároveň mění výšku těla robotu nad zemí. Nadřízený člen provádí posuv těla robotu v rovině těla. Požadovaný posuv těla může být znemožněn nepříznivou konfigurací noh. Paralelní aktivace řídicího členu postoje se snaží o dosažení takových konfigurací noh, ve kterých se blíží svým středním polohám. Tímto způsobem výrazně podporuje nadřazený řídicí člen, který je schopen dosáhnout lepší rozložení sil v požadovaných došlapech. Průběh konvergence tohoto členu je rovněž ukázán ve výsledné chůzi robotu po rovině, viz. soubor *chuze_rovina.avi* příloha A.

Samostatné aktivace těchto kompozitních členů prokázaly funkčnost navrženého koordinačního schématu popsaného v kap. 3.4.3.3. Ukázaly také konkrétní chování robotu, která mohou být využívána. Výsledná chování kompozitních řídicích členů tak představují výrazný přínos pro hledání optimálních strategií chůze robotu, kterým je věnována následující kapitola 3.6.

3.6 Výsledná chůze robotu při komplexních úlohách

V předcházející kapitole 3.5 byla ukázána elementární chování robotu generovaná elementárními řídicími členy a také některá složitější chování robotu, realizovaná kompozitními řídicími členy. Po dosažení těchto dílčích výsledků, bylo přistoupeno

k návrhu řešení vybraných komplexních úloh robotu, tak jak jsou uvedeny v kap.2.5.2. Jedná se především o tyto úlohy:

- rotace robotu změnou úhlu bočení
- chůze robotu v longitudinálním směru po vodorovné rovině
- chůze robotu v longitudinálním směru po nakloněné rovině

Pro každou z těchto úloh byla navržena vlastní reprezentace stavu robotu, zvolena množina základních řídicích členů a zadány omezující podmínky úlohy. Tyto údaje byly použity jako vstup vytvořené aplikace, viz. kap. 3.4, pomocí níž byla vytvořena množina kompozitních řídicích členů, model chování robotu pro danou úlohu a nalezena vhodná posloupnost aktivací jednotlivých zákonů řízení prostřednictvím algoritmu Q-učení. V žádné z následujících úloh nemá algoritmus učení k dispozici znalost o tom, zda se robot nachází na vodorovné nebo nakloněné rovině, ani zda jsou v okolí robotu umístěny překážky.

3.6.1 Úloha rotace robotu

Cílem této úlohy bylo nalezení posloupnosti aktivací dostupných zákonů řízení, které by vedly k rotaci robotu kolem osy $Y^{(2)}$ na vodorovné rovině. Nejlepších výsledků bylo dosaženo při použití následujícího návrhu.

3.6.1.1 Vstupní parametry úlohy

Pro reprezentaci konfigurace robotu v této úloze bylo zvoleno celkem 6 predikátů. Stavový vektor ukazuje následující rovnice:

$$q = (p_0, p_1, p_2, p_3, p_4, p_5) \quad (62)$$

kde jednotlivé složky jsou definovány takto:

$$\begin{aligned} p_0 &\leftarrow \Phi_{K^*}^{\underline{0,1,2}}, p_1 \leftarrow \Phi_{K^*}^{\underline{0,2,3}}, p_2 \leftarrow \Phi_{K^*}^{\underline{0,1,3}}, p_3 \leftarrow \Phi_{K^*}^{\underline{1,2,3}}, \\ p_4 &\leftarrow \Phi_{P_\varphi}^{\underline{0,1,2,3}}, p_5 \leftarrow \Phi_{P_y}^{\underline{0,1,2,3}}. \end{aligned} \quad (63)$$

Predikáty p_0 až p_3 jsou splněny, resp. jsou pravdivé, pokud je dosaženo cíle řídicího členu kontaktu, který jako vstupní zdroj používá vybranou trojici noh robotu. Zároveň není stanoveno, pomocí kterých výstupních zdrojů je těchto cílů dosaženo, zda translačním pohybem těla, nebo změnou konfigurace vybrané nohy. Např. predikát p_0 je pravdivý, pokud jsou reakce v došlapech noh 0,1,2 rozloženy rovnoměrně, tj. průmět těžiště robotu do roviny podložky se nachází uvnitř trojúhelníku, jehož vrcholy tvoří došlapy uvedených noh. Predikát p_4 je splněn pokud je dosažen cíl řídicího členu postoje, který jako výstupní zdroj používá úhel bočení těla robotu, φ . Predikát p_5 je pravdivý, pokud je výška těla robotu nad zemí optimální vzhledem k heuristické míře postoje.

Pro potřeby sestavení modelu všech možných chování bylo nutné stanovit omezující podmínky. Podmínka realizovatelnosti udává, které stavy nemohou být v žádném případě dosaženy. Tento typ podmínky je důležitý především pro omezení

velikosti modelu chování, což má za následek i snížení doby učení. V tomto případě byla použita následující podmínka:

$$(p_0 \wedge p_1) \vee (p_2 \wedge p_3) \quad (64)$$

Stavy splňující tuto podmínku nemohou být v žádném případě dosaženy, neboť v případě že jsou reakce rovnoměrně rozloženy v bodech došlapu trojúhelníku 012, nemohou být zároveň rozloženy rovnoměrně i v protilehlém trojúhelníku 123.

Druhý typ omezujících podmínek, jsou podmínky bezpečnostní, které určují jaké stavy jsou bezpečné. Tento typ podmínek musí být splněn v každém okamžiku. Pro řešení diskutované úlohy byla použita tato podmínka:

$$p_0 \vee p_1 \vee p_2 \vee p_3 \quad (65)$$

Zavedení této podmínky opět vedlo na výrazné zjednodušení modelu chování robotu. Z modelu jsou totiž vyřazeny aktivace takových zákonů řízení, které mohou potenciálně vést robot do nepřipustného stavu, a to bez ohledu na výsledek procesu konvergence.

Pro řešení úlohy rotace robotu byla zvolena báze řízení obsahující pouze dva členy, $\{\Phi_K, \Phi_P\}$, tj. řídicí člen kontaktu a postoje. V popisovaném řešení, pomocí kterého bylo dosaženo nejlepších výsledků, nebyl použit řídicí člen trajektorie. I když pomocí tohoto řídicího členu bylo dosahováno větších změn úhlu bočení během jedné konvergence, byla koncová konfigurace natolik nevýhodná pro další pohyb, že výsledné chování robotu bylo tvořeno velmi složitou posloupností aktivací. Toto chování pak vyžadovalo delší dobu potřebnou pro vykonání jednoho otočení o 2π .

Výchozí množina řídicích členů obsahuje celkem 14 různých instancí:

$$C = \left\{ \Phi_{P\varnothing}^{0,1,2,3}, \Phi_{Py}^{0,1,2,3}, \Phi_{P0}^0, \Phi_{P1}^1, \Phi_{P2}^2, \Phi_{P3}^3, \Phi_{K2}^{0,1,2}, \Phi_{K0}^{0,2,3}, \right. \\ \left. \Phi_{K1}^{0,1,3}, \Phi_{K3}^{1,2,3}, \Phi_{Kx,z}^{0,1,2}, \Phi_{Kx,z}^{0,2,3}, \Phi_{Kx,z}^{0,1,3}, \Phi_{Kx,z}^{1,2,3} \right\} \quad (66)$$

Do množiny jsou zahrnuty dvě instance řídicího členu postoje, který optimalizuje heuristickou míru postoje celého robotu prostřednictvím změny úhlu bočení, resp. změnou výšky těla robotu nad zemí. Další čtyři instance optimalizují heuristickou míru postoje jedné nohy změnou její konfigurace. Zbýlých 6 zákonů řízení, které se snaží o optimální rozložení sil v došlapech, představují instance řídicího členu kontaktu. Svého cíle dosahují buďto translačním pohybem těla robotu, nebo změnou konfigurace jedné z noh. Všechny tyto instance byly podrobně popsány v kap. 3.3 a jejich chování bylo ukázáno v kap. 3.5.1.

Pro všechny tyto základní řídicí členy musely být stanoveny specifikace ve formě vektorů S, M, R , tj. podmínka aktivace, vliv konvergence a cílový stav, který má být dosažen, teoretický rozbor viz. kap. 2.3.2.1 a 2.3.2.2. Protože instance zajišťující stejná chování robotu, např. Φ_{P0}^0 a Φ_{P1}^1 mají téměř totožné specifikace, budou dále uvedeny pouze vybrané specifikace.

$$\Phi_{P\varnothing}^{0,1,2,3}: S = [*, *, *, *, 0, *], \quad M = [-, -, -, -, *, -], \quad R = [*, *, *, *, 1, *] \quad (67)$$

$$\Phi_{p_y}^{0,1,2,3} : S = [*, *, *, *, *, 0], \quad M = [-, -, -, -, -, *], \quad R = [*, *, *, *, *, 1] \quad (68)$$

Specifikace řídicího členu $\Phi_{p_\varphi}^{0,1,2,3}$, viz. rov. (67), určují, že může být aktivován pouze v případě, kdy je predikát p_4 nepravda, modifikuje predikát p_4 , a snaží se o dosažení libovolného stavu ve kterém je predikát p_4 pravdivý. Podobně je tomu se specifikací instance $\Phi_{p_y}^{0,1,2,3}$, která ovlivňuje pouze predikát p_5 . Takto zavedené specifikace např. umožňují pozdější automatické sestavení složeného řídicího členu $\Phi_{p_\varphi}^{0,1,2,3} < \Phi_{p_y}^{0,1,2,3}$, který paralelně mění úhel bočení těla a rovněž mění výšku těla robotu nad zemí.

Pro řídicí člen postojе měnící konfiguraci nohy 0 byly zavedeny specifikace dle rovnice (69). Podmínkou aktivace je splnění predikátu p_3 , který zároveň není modifikován. Pokud by tento predikát nebyl splněn, pak by při snaze změnit konfiguraci nohy, tj. při přechodu nohy z oporné fáze do fáze posuvu, robot ztratil stabilní postavení. Ostatní instance řídicího členu postojе měnící konfiguraci nohy mají totožný vektor R . Vektory S, M se pak liší pouze číslem predikátu jehož splnění je vyžadováno a během konvergence zachováno.

$$\Phi_{p_0}^0 : S = [*, *, *, 1, *, *], \quad M = [*, *, *, -, *, *], \quad R = [*, *, *, *, *, 1] \quad (69)$$

Do množiny C jsou dále zahrnuty instance řídicího členu kontaktu. Ty, které mění konfiguraci vybrané nohy vyžadují, aby byly síly rovnoměrně rozloženy v došlapech tvořící trojúhelník protilehlý této noze. Specifikace instance $\Phi_{K_2}^{0,1,2}$ udává rovnice (70). Protože tato instance vyžaduje pro svoji činnost splnění predikátu p_2 nemůže být tento predikát modifikován. Z toho plyne, že nemůže být modifikován ani predikát p_3 , neboť platí výše uvedená podmínka realizovatelnosti, viz. rov. (64). Cílem zmíněné instance je splnění predikátu p_0 a tím je zároveň aktivně měněn predikát p_1 na nepravdu (logickou nulu).

$$\Phi_{K_2}^{0,1,2} : S = [0, *, 1, *, *, *], \quad M = [*, *, -, -, *, *], \quad R = [1, 0, *, *, *, *] \quad (70)$$

$$\Phi_{K_{x,z}}^{0,1,2} : S = [0, *, *, *, *, *], \quad M = [*, *, -, -, *, *], \quad R = [1, 0, *, *, *, *] \quad (71)$$

Poslední skupinu představují instance řídicího členu kontaktu, které provádí translační pohyb těla robotu. Příklad specifikace členu $\Phi_{K_{x,z}}^{0,1,2}$ je uveden v rovnici (71). Rozdíl oproti specifikaci v rov. (70) je pouze v podmínce aktivace. V okamžiku aktivace uvedeného řídicího členu totiž robot stojí na všech nohách. Není proto nutné stanovovat, který z predikátů p_1 až p_3 musí být splněn. Zákony řízení, které provádí translační pohyb těla robotu, jsou do řešení rotace robotu začleněny především pro potřeby fáze učení se. U dřívějších návrhů, které tyto členy neobsahovaly, totiž v průběhu učení vznikaly stavy, ze kterých se robot nemohl dostat jiným pohybem než translačním. Tato situace byla dána především zkoušením různých posloupností aktivací učícím algoritmem. Výskyt aktivace těchto zákonů řízení ve výsledné chůzi rotačního pohybu je však nežádoucí.

3.6.1.2 Odvození modelu chování a návrh parametrů učení

Vstupní data uvedená v kapitole 3.6.1.1, tj. reprezentace stavu robotu, báze řízení, omezující podmínky a specifikace jednotlivých instancí řídicích členů, byly vytvořeny ručně a slouží jako vstup pro software simulaci, viz. kap. 3.4. Po té, co aplikace tyto informace načte, je schopná pomocí uvedených specifikací odvodit množinu složených řídicích členů a jejich specifikace a sestavit model všech možných chování robotu při řešení zadané úlohy.

V řešené úloze rotace robotu bylo ze 14ti základních řídicích členů vyvozeno celkem 29 různých kompozitních řídicích členů použitelných pro sestavení modelu všech možných chování robotu. Ve výsledné chůzi pak z těchto kompozitních členů byly použity především řídicí členy uvedené v rovnici (72). První kompozitní člen z této množiny zajišťuje rotaci robotu paralelně se změnou konfigurace vybrané nohy. Druhý pak při rotaci těla průběžně optimalizuje postavení vybrané nohy.

$$\left\{ \Phi_{P\varphi}^{0,1,2,3} < \Phi_{Kz}^{a,b,c}, \Phi_{P\varphi}^{0,1,2,3} < \Phi_{Pa}^a \right\} \quad a, b, c \in (0, 1, 2, 3), a \neq b \neq c \quad (72)$$

Uvedená množina celkem 43 základních a kompozitních členů byla následně použita pro vyvození modelu všech možných chování robotu. Tento model představuje prostřední vrstvu popsané architektury řízení, viz. kap. 2.3.2. Konkrétní algoritmus, který byl použit pro vyvození tohoto modelu je uveden v kap. 3.4.3.1. Výsledný model obsahuje celkem 32 různých stavů, ostatní stavy, které nebyly do modelu zahrnuty nesplňují uvedené omezující podmínky, viz. rov. (64) a (65). V každém stavu zahrnutém do modelu je umožněno v průměru 8 různých aktivací zákonů řízení.

Uvedený model je následně používán algoritmem Q-učení pro nalezení optimální strategie řízení robotu. Tento algoritmus vyžaduje nastavení parametrů α , γ a r . Parametr učení α může nabývat hodnot v rozmezí 0 až 1. Nejlepších výsledků bylo dosaženo při počátečním nastavení na hodnotu $\alpha = 0,6$ a postupným snižováním této hodnoty s krokem $\Delta\alpha = 0,2$ až na hodnotu 0, při dosažení optimální chůze. Snižování bylo realizováno simulačním programem automaticky po provedení 500 aktualizací Q-hodnot dle rov. (23). Srážkový faktor γ byl nastaven experimentálně na hodnotu 0.2. Pro stanovení okamžité odměny r byl použit výpočet uvedený v rov. (73):

$$\begin{aligned} r &= -1 && ; \Delta\varphi < 2^\circ \\ r &= -1 / \Delta\varphi && ; 2^\circ \leq \Delta\varphi \leq 10^\circ \\ r &= 0 && ; \Delta\varphi > 10^\circ \end{aligned} \quad (73)$$

Kde označuje změnu úhlu bočení. Pro rotaci robotu v opačném směru postačuje změnit kladné hodnoty úhlů na záporné a zaměnit operátory. Kromě nastavení uvedených parametrů je důležité zvolit výchozí Q hodnoty pro všechny dvojice stav, akce. Nejlepších výsledků bylo dosahováno při nastavení výchozích hodnot na 0. Tímto je na začátku procesu učení dosažena situace, ve které jsou všechny akce stejně vhodné k výběru pro optimální strategii. Pokud některá akce opakovaně nevede k úspěchu, pak je příslušná Q hodnota postupně snižována a to až na hodnotu -1 v nejméně příznivém případě. Naopak pokud určitá akce v daném stavu vždy vede k vysoké odměně, zůstává příslušná Q hodnota stále nastavena na 0.

3.6.1.3 Výsledná naučená chůze

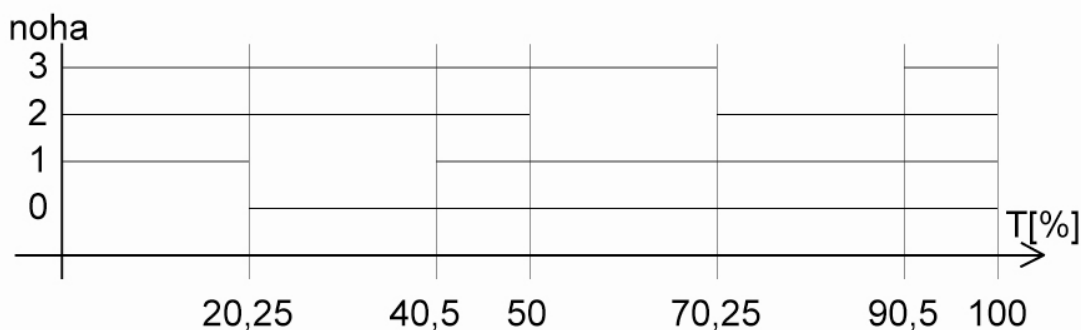
Pomocí zvolených parametrů učení byla nalezena výsledná chůze robotu, která zajišťuje rotaci robotu. Chování robotu je zachyceno na videozáznamu, viz. příloha A soubor *rotace_rovina.avi*. Ze záznamu je patrné, že se jedná o chůzi:

- cyklickou: pohyb noh je realizován v pravidelných časových intervalech
- otáčivou: libovolný bod těla rotuje kolem osy $Y^{(2)}$
- regulární: každá noha má stejný činitel využití β_c
- plazivou: v každém okamžiku je v přenosové fázi maximálně jedna noha a $\beta_{cv} > 0,75$
- nespojitou: k pohybu těla dochází pouze v některých fázích chůze

Činitel využití β_{cv} , viz. kap. 2.1.3, je u jednotlivých noh shodný. Průměrná hodnota tohoto ukazatele z pěti cyklů je 79,75%. Relativní fáze, Φ_{ri} , noh 1,2,3 vůči referenční noze 0 nabývá hodnot:

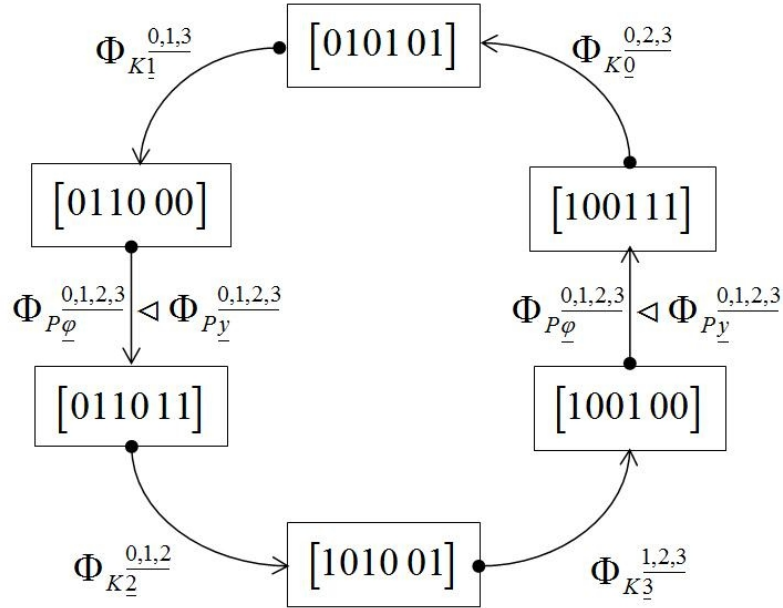
$$\Phi_{ri} = \{0, 29.75, 70.25\} \quad (74)$$

Graf výsledné chůze, ukazující střídání oporné a posuvné fáze noh v průběhu jednoho cyklu, je ukázán na Obr. 34. Na ose X jsou vyneseny procentuální hodnoty z doby trvání jednoho cyklu, T . K rotaci těla dochází dvakrát během tohoto cyklu a to vždy, když se všechny nohy nachází v oporné fázi. Celkový čas, během kterého je manipulováno tělem robotu, činí 19% doby trvání cyklu.



Obr. 34: Graf chůze v úloze rotace

Ve výsledné chůzi jsou aktivovány především 4 instance základního řídicího členu kontaktu a dvě kompozitní instance zajišťující rotaci robotu a změnu výšky těla nad povrchem. Konkrétní posloupnost aktivací a přechody mezi stavy robotu jsou ukázány na Obr. 35.



Obr. 35: Přejchody mezi stavy robotu v úloze rotace

Uvedené přechody mezi stavy nastávají v případě rotace robotu na vodorovné rovině nejčastěji. Pokud by ovšem některý z aktivovaných zákonů řízení nebyl úspěšný, např. vlivem nepříznivé konfigurace robotu, výskytem překážky apod., došlo by k přechodu soustavy do stavu, který není ve schématu uveden. V tomto případě by ovšem činnost robotu nebyla zastavena, ale došlo by k aktivaci nejlépe ohodnoceného zákona řízení pro daný nový stav. K takovýmto událostem docházelo především v počátečních stádiích procesu učení, kdy byly testovány i velmi nepříznivé posloupnosti aktivací vedoucí robot do konfigurace, kdy se např. více noh nacházelo ve svých mezních polohách. Pro dosažení výsledné chůze bylo provedeno přibližně 1500 aktualizací Q hodnoty.

3.6.2 Úloha dopředné chůze

Cílem této úlohy bylo nalezení optimální posloupnosti aktivací zákonů řízení, která zajistí chůzi robotu ve směru jeho longitudinální osy, tj. osy $X^{(2)}$, po vodorovné rovině. Nejlepších výsledků bylo dosaženo při použití následujícího návrhu.

3.6.2.1 Vstupní parametry úlohy

Pro reprezentaci stavu robotu bylo opět použito stavového vektoru, q , tvořeného šesti složkami. Jednotlivé predikáty jsou definovány takto:

$$\begin{aligned} p_0 &\leftarrow \Phi_{K_-^*}^{0,1,2}, p_1 \leftarrow \Phi_{K_-^*}^{0,2,3}, p_2 \leftarrow \Phi_{K_-^*}^{0,1,3}, p_3 \leftarrow \Phi_{K_-^*}^{1,2,3}, \\ p_4 &\leftarrow \Phi_{P_0}^0 \vee \Phi_{P_1}^1, p_5 \leftarrow \Phi_{P_3}^2 \vee \Phi_{P_3}^3. \end{aligned} \quad (75)$$

Složky p_0 až p_3 mají stejný význam jako při řešení předchozí úlohy. Predikát p_4 je pravdivý, pokud je splněn cíl řídicího členu postoje měnící konfiguraci nohy 0 nebo

nohy 1. Obdobně predikát p_5 je splněn pokud je řídicím členem postoje nastavena noha 2 nebo 3. Pro potřeby zjednodušení výsledného modelu všech možných chování byly použity dvě omezující podmínky, stejné jako v předchozí úloze, viz. rov. (64) a (65).

Pro řešení dopředné chůze robotu byla použita dvoučlenná báze řízení, $\{\Phi_K, \Phi_P\}$, tj. řídicí člen kontaktu a postoje. Instance těchto členů posloužily pro sestavení výchozí množiny zákonů řízení, obsahující celkem 14 prvků:

$$C = \left\{ \Phi_{P_{x,z}}^{0,1,2,3}, \Phi_{P_y}^{0,1,2,3}, \Phi_{P_0}^0, \Phi_{P_1}^1, \Phi_{P_2}^2, \Phi_{P_3}^3, \Phi_{K_2}^{0,1,2}, \Phi_{K_0}^{0,2,3}, \right. \\ \left. \Phi_{K_3}^{0,1,3}, \Phi_{K_1}^{1,2,3}, \Phi_{K_{x,z}}^{0,1,2}, \Phi_{K_{x,z}}^{0,2,3}, \Phi_{K_{x,z}}^{0,1,3}, \Phi_{K_{x,z}}^{1,2,3} \right\} \quad (76)$$

V této množině je oproti předchozí úloze použita instance řídicího členu postoje, který zajišťuje translační pohyb těla robotu v osách $X^{(2)}, Z^{(2)}$. A dále jsou nově použity dvě instance řídicího členu kontaktu, který mění konfiguraci vybrané nohy, $\Phi_{K_3}^{0,1,3}, \Phi_{K_1}^{1,2,3}$. Zbylé instance tohoto řídicího členu, viz. rov. (47), nejsou začleněny. Např. instance $\Phi_{K_0}^{0,1,3}$ by totiž svými specifikacemi nesplnila bezpečnostní podmínku danou rovnicí (65). Instance $\Phi_{K_1}^{0,1,3}$, jejíž činnost byla ukázána v předcházející úloze, zase vede na takovou konfiguraci nohy 1, která následně výrazně omezuje možnost posuvu těla robotu ve směru osy $X^{(2)}$. Důvodem je především mezní hodnota úhlu α^i . Všechny uvedené instance byly podrobně popsány v kap. 3.3 a jejich chování bylo ukázáno v kap. 3.5.1.

Po sestavení množiny elementárních řídicích členů, byly navrženy jejich specifikace. Protože stavový vektor, q , se od předchozí úlohy liší, budou odlišné i specifikace jednotlivých instancí, a to především složky související s predikáty p_4 a p_5 . Opět platí, že jednotlivé skupiny řídicích členů mají podobné specifikace, proto budou uvedeny podmínky aktivace, vliv konvergence a cílový stav jen pro vybrané zástupce těchto množin.

Řídicí člen, $\Phi_{P_{x,z}}^{0,1,2,3}$, provádějící translační pohyb nevyžaduje pro svoji aktivaci splnění žádného z predikátů, viz. rov. (77). Při svoji činnosti se snaží o zlepšení heuristické míry postoje, proto potenciálně i o nastavení predikátů p_4 a p_5 , které může modifikovat. Přestože potenciálně může tento zákon řízení změnit nastavení prvních čtyř složek stavového vektoru, je ve specifikaci, M , uvedeno, že tyto složky nemodifikuje. Výsledkem konvergence tohoto zákona řízení je totiž vždy stav, který má alespoň jeden z predikátů p_0 až p_3 splněn. Totožné specifikace byly stanoveny i pro zákon řízení $\Phi_{P_y}^{0,1,2,3}$.

$$\Phi_{P_{x,z}}^{0,1,2,3}: S = [*, *, *, *, *, *], \quad M = [-, -, -, -, *, *], \quad R = [0, 0, 0, 0, 1, 1] \quad (77)$$

$$\Phi_{P_y}^{0,1,2,3}: S = [*, *, *, *, *, *], \quad M = [-, -, -, -, *, *], \quad R = [0, 0, 0, 0, 1, 1] \quad (78)$$

Řídicí člen, $\Phi_{P_0}^0$, který mění konfiguraci nohy 0, vyžaduje, aby byla výchozí hodnota predikátu $p_4 = 0$ neboť jeho cílem je splnění tohoto predikátu, viz. rov. (79).

Pro aktivaci je nutné splnění predikátu p_3 , v opačném případě by nemohla noha 0 přejít z oporné fáze do posuvné. Obdobně je sestavena i specifikace instance $\Phi_{p_3}^{\frac{3}{2}}$, jejíž činnost je ovšem spojena s predikátem p_5 , viz. rov. (80).

$$\Phi_{p_0}^0 : S = [*, *, *, 1, 0, *], \quad M = [*, *, *, -, *, -], \quad R = [*, *, *, *, 1, *] \quad (79)$$

$$\Phi_{p_3}^{\frac{3}{2}} : S = [1, *, *, *, *, 0], \quad M = [-, *, *, *, -, *], \quad R = [*, *, *, *, 1, *] \quad (80)$$

Specifikace řídicího členu, $\Phi_{K_2}^{\frac{0,1,2}{x,z}}$, se liší oproti předchozí úloze pouze v nastavení posledních dvou složek, viz. rov. (81). Predikát p_4 není měněn, protože nedochází ke změně konfigurace noh 0 a 1, ani k posuvu těla robotu. Posuv těla totiž v případě činnosti řídicího členu kontaktu mění hodnotu heuristické míry postoje jednotlivých noh. Proto je u řídicího členu $\Phi_{K_2}^{\frac{0,1,2}{x,z}}$ uvedeno, že mění predikáty p_4 a p_5 , viz. rov. (82). Ostatní hodnoty zůstávají oproti předchozí úloze nezměněny.

$$\Phi_{K_2}^{\frac{0,1,2}{x,z}} : S = [0, *, 1, *, *, *], \quad M = [*, *, -, -, -, *], \quad R = [1, 0, *, *, *, *] \quad (81)$$

$$\Phi_{K_{x,z}}^{\frac{0,1,2}{x,z}} : S = [0, *, *, *, *, *], \quad M = [*, *, -, -, *, *], \quad R = [1, 0, *, *, *, *] \quad (82)$$

3.6.2.2 Odvození modelu chování a návrh parametrů učení

Návrh predikátů, rov. (75), určující stav robotu při řešení úlohy dopředné chůze, množina základních řídicích členů, rov. (76), a jejich specifikace, rov. (77) až (82), byly použity jako vstup pro software simulaci, která zajistila vyvození kompozitních řídicích členů a modelu všech možných chování robotu při řešení zadané úlohy.

Z výchozí množiny obsahující 14 elementárních řídicích členů bylo vyvozeno celkem 24 různých kompozitních řídicích členů. Ve výsledné chůzi bylo nejčastěji aktivováno 6 složených členů uvedených v rov. (83). Např. zákon řízení $\Phi_{p_{x,z}}^{\frac{0,1,2,3}{x,z}} < \Phi_{K_{x,z}}^{\frac{1,2,3}{x,z}}$ se snaží o dosažení optimální polohy robotu translací těla robotu v rovině těla a zároveň mění výšku těla robotu nad zemí, pokud pohyb vede k vyšší hodnotě heuristické míry postoje.

$$\left\{ \begin{array}{l} \Phi_{p_{x,z}}^{\frac{0,1,2,3}{x,z}} < \Phi_{K_3}^{\frac{0,1,3}{x,z}}, \Phi_{p_{x,z}}^{\frac{0,1,2,3}{x,z}} < \Phi_{K_0}^{\frac{0,2,3}{x,z}}, \\ \Phi_{p_y}^{\frac{0,1,2,3}{x,z}} < \Phi_{K_{x,z}}^{\frac{1,2,3}{x,z}}, \Phi_{p_y}^{\frac{0,1,2,3}{x,z}} < \Phi_{K_{x,z}}^{\frac{0,1,2}{x,z}}, \\ \Phi_{p_y}^{\frac{0,1,2,3}{x,z}} < \Phi_{p_2}^{\frac{2}{x,z}}, \Phi_{p_{x,z}}^{\frac{0,1,2,3}{x,z}} < \Phi_{p_y}^{\frac{0,1,2,3}{x,z}} \end{array} \right\} \quad (83)$$

Uvedená množina 38 základních a složených řídicích členů byla použita pro vyvození modelu všech možných chování robotu při řešení úlohy chůze v longitudinálním směru. Výsledný model obsahuje opět 32 různých stavů, neboť jsou uplatňovány stejné omezující podmínky jako při řešení předcházející úlohy. V každém z těchto stavů umožňuje koordinační mechanismus prostřední vrstvy architektury řízení aktivovat v průměru 10 různých zákonů řízení. Výsledný model tedy obsahuje více přechodů než v případě řešení úlohy rotace.

Zmíněný model je následně využíván algoritmem Q-učení pro nalezení optimální strategie řízení robotu. Parametr učení byl nastaven na výchozí hodnotu $\alpha = 0,6$ a postupně snižován s krokem $\Delta\alpha = 0,2$ až na hodnotu 0, při dosažení výsledné chůze. Ke snižování bylo přistoupeno nejprve po 500 iteracích Q hodnoty a následně po 750 iteracích. Srážkový faktor γ byl nastaven experimentálně na hodnotu 0.15, protože bylo očekáváno nalezení cyklické strategie s větším počtem aktivací než při řešení předchozí úlohy. Pro stanovení okamžité odměny r byl použit výpočet uvedený v rov.(84):

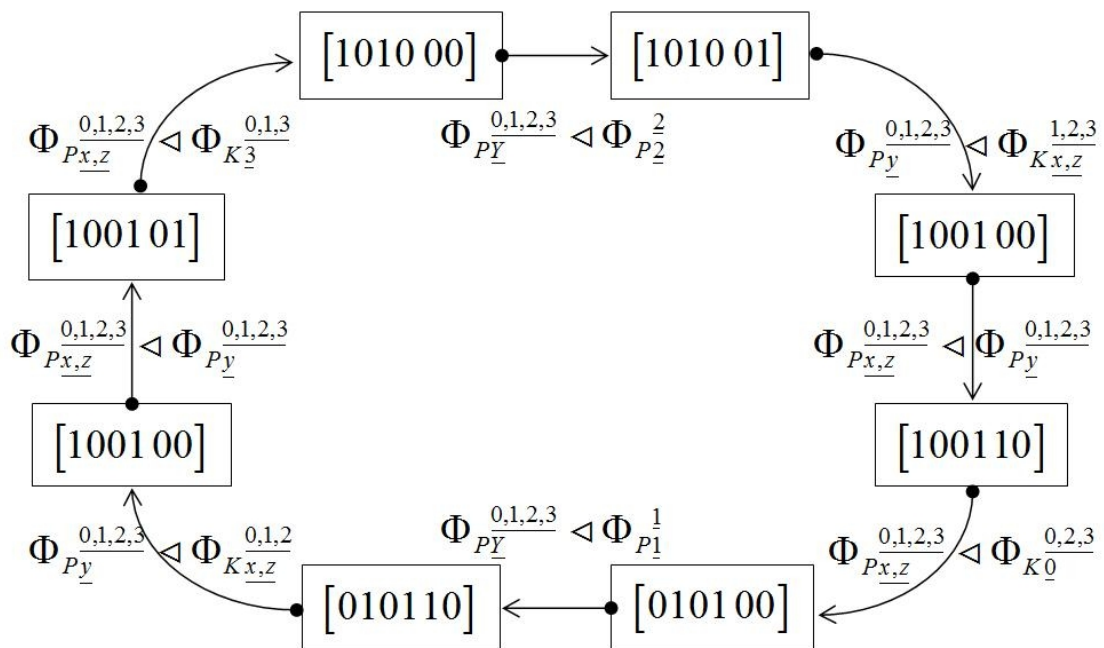
$$\begin{aligned} r &= -1,5 && ; \Delta x < -\varepsilon \\ r &= -1 && ; \Delta x < \varepsilon \\ r &= 0 && ; \Delta x > \varepsilon \end{aligned} \quad (84)$$

Výraz Δx označuje velikost posuvu těla robotu ve směru osy $X^{(2)}$. Hodnota ε je minimální posuv, který lze považovat za přínosný. Experimentálně byl stanoven na 10% délky těla robotu. Provedené experimenty ukázaly nutnost výrazně znevýhodnit aktivace, které vedou na významný posuv robotu v opačném směru než je požadováno. Všechny výchozí Q hodnoty byly opět nastaveny na 0.

3.6.2.3 Výsledná naučená chůze

Uvedené parametry učení vedly k nalezení strategie, která zajišťuje chůzi robotu ve směru longitudinální osy. Chování robotu je zachyceno na videozáznamu, viz. příloha A soubor *chuze_rovina.avi*. Výslednou chůzi robotu je možné považovat za cyklickou. Hlavní cyklus, který zajišťuje chůzi je ovšem několikrát narušen a to v okamžiku, kdy některý z aktivovaných řídicích členů nedosáhne svého cílového stavu. V následujícím textu bude popsán tento hlavní cyklus. Možná porušení tohoto hlavního cyklu a příčiny neúspěchu konvergence řídicích členů budou diskutovány v následující kapitole 3.6.3.

Přechody mezi jednotlivými stavy robotu v průběhu zmíněného cyklu jsou ukázány na Obr. 36. K posuvu robotu v požadovaném směru dochází dvakrát během jednoho cyklu, a to pod vlivem aktivací $\Phi_{P_y}^{0,1,2,3} < \Phi_{K_{x,z}}^{1,2,3}$ a $\Phi_{P_y}^{0,1,2,3} < \Phi_{K_{x,z}}^{0,1,2}$. V těchto případech je rovněž optimalizován postoj robotu změnou výšky těla nad podložkou. Mezi těmito hlavními posuvy těla nastává fáze chůze, kterou je možné označit jako přípravou. V těchto částech cyklu jsou totiž měněny konfigurace noh levé, resp. pravé, strany robotu. Zatímco přední nohy jsou nastavovány pomocí řídicího členu postoje, zadní nohy jsou nastavovány pomocí řídicího členu kontaktu. Přední nohy robotu, tj. 1 a 2, nemohou být nastaveny instancí řídicího členu kontaktu, neboť nebyly zařazeny do výchozí množiny C , viz rov. (76), jak bylo diskutováno v předchozí kapitole. Naopak zadní nohy robotu, tj. 0 a 3, nejsou nastavovány řídicím členem postoje, protože výsledná konfigurace je nevýhodná pro dosažení dostatečně velkého posuvu, který je motivován udělením odměny, r , dle rov. (84).



Obr. 36: Přejechy mezi stavy robota v úloze dopředné chůze po rovině

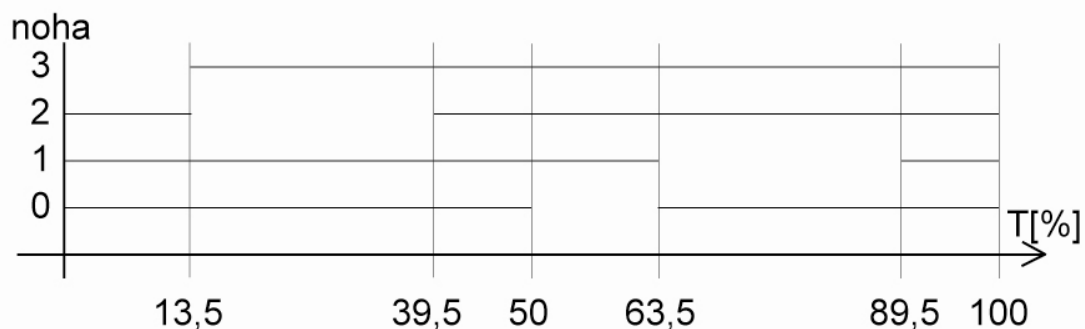
Pokud dochází k úspěšným konvergencím řídicích členů uvedených na Obr. 36, pak robot vykonává chůzi:

- cyklickou: pohyb noh je realizován v pravidelných časových intervalech
- vlnitou: pohyb noh začíná zadní nohou a pokračuje v dopředném směru, laterálně spárované nohy jsou fázově posunuty přesně o polovinu cyklu kroku
- plazivou: v každém okamžiku je v přenosové fázi maximálně jedna noha
- nespojitou: k pohybu těla dochází pouze v některých fázích chůze

Činitel využití, viz. kap. 2.1.3, pro nohy 0,3 nabývá hodnoty $\beta_{cv} = 86,5\%$ a pro nohy 1,2 je $\beta_{cv} = 74\%$. Hodnoty tohoto ukazatele se liší, protože pohyb předních noh je realizován řídicím členem postoje, který manipuluje nohou jiným způsobem než řídicí člen kontaktu, který je používán pro změnu konfigurace noh zadních. Relativní fáze, Φ_{ri} , noh 1,2,3 vůči referenční noze 0 nabývá hodnot:

$$\Phi_{ri} = \{0, 40, 26.5\} \quad (85)$$

Graf výsledné chůze, ukazující střídání oporné a posuvné fáze noh v průběhu jednoho cyklu, je ukázán na Obr. 37. Na ose X jsou vyneseny procentuální hodnoty z doby trvání jednoho cyklu, T . K posuvu těla dochází dvakrát během tohoto cyklu a to vždy, když se všechny nohy nachází v oporné fázi. Celkový čas, během kterého je manipulováno tělem robota, činí 21% doby trvání cyklu.



Obr. 37: Graf chůze v úloze dopředné chůze po rovině

3.6.3 Úloha dopředné chůze na nakloněné rovině

Cílem této úlohy je ověřit schopnost robotu pohybovat se po nakloněné rovině v longitudinálním směru. Pro potřeby experimentálního ověření byl robot umístěn na nakloněnou rovinu, která je pootočená o úhel stoupání 5° kolem osy $Z^{(1)}$.

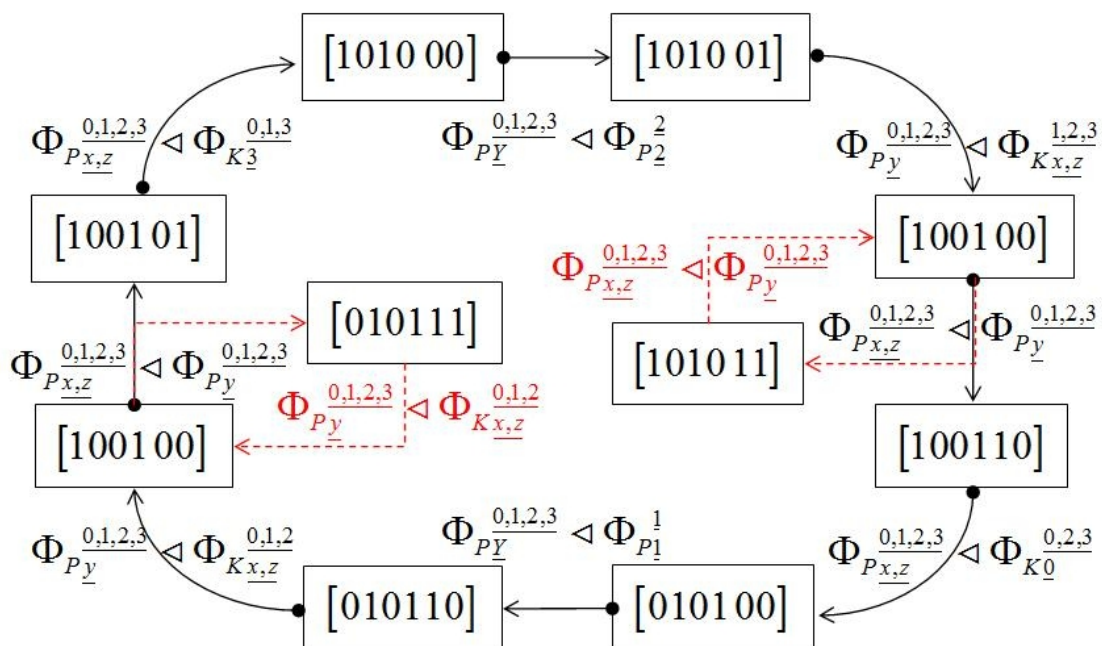
3.6.3.1 Vstupní parametry úlohy

Pro ověření schopností řídicích členů a učicího algoritmu byla jako výchozí strategie použita chůze naučená při řešení úlohy chůze na vodorovné rovině, viz. podrobný popis v předchozí kapitole 3.6.2. Tento postup je možný, protože řídicí členy v sobě mají zapracováno reaktivní chování, viz. kap. 3.3.1.3, a musí být schopny pracovat jak na vodorovné tak nakloněné rovině, s překážkami i bez nich.

Pro potřebu řešení chůze po nakloněné rovině byl parametr učení α nastaven na hodnotu 0.1. Tímto je zajištěna možnost změny již naučené strategie. Vyšší hodnoty parametru vedly k porušení naučené strategie avšak nedošlo k naučení strategie výhodnější. Veškeré ostatní vstupní parametry nutné pro řešení úlohy se shodují s řešením navrženým v předchozí kapitole. Přestože je robot umístěn do jiného prostředí, může být zachována báze řízení, množina elementárních řídicích členů, model všech chování i parametry učení, γ a r , tj. srážkový faktor resp. velikost okamžité odměny. Minimum vynucených změn při změně úlohy lze považovat za jednu z výhod zvoleného přístupu.

3.6.3.2 Výsledná naučená chůze

Při snaze uplatňovat již naučenou strategii aktivací zákonů řízení se ukázalo, že některé instance sice plní svůj cíl řízení, ale výsledná konfigurace robotu je odlišná než konfigurace při řešení předchozí úlohy. Jedná se především o instance $\Phi_{Kx,z}^{0,1,2}$ a $\Phi_{Kx,z}^{1,2,3}$, které se snaží rovnoměrně rozložit síly v došlapech předních noh a jedné z noh zadních. Cíl aktivace je splněn, protože je dosaženo stabilního postavení na nohách, které jsou použity jako vstupní zdroje instance, ale koncová hodnota směrodatné odchylky reakcí, dle rov. (60), je větší než v předchozí úloze. To znamená, že i konfigurace robotu po dokončení procesu konvergence je odlišná. Následná aktivace řídicího členu vede na jiný než očekávaný stav, čímž dochází k opuštění hlavního cyklu. Nové přechody mezi stavy jsou znázorněny na Obr. 38 červenou barvou.



Obr. 38: Přechody mezi stavy robotu v úloze dopředné chůze po nakloněné rovině

Popsaná situace, kdy není splněn cíl aktivovaného zákona řízení, může nastat i v případě chůze na vodorovné rovině ovšem s nižší pravděpodobností. Čím bude mít rovina větší úhel stoupání, anebo čím více překážek, resp. nerovností terénu, se bude v prostředí vyskytovat, tím větší bude pravděpodobnost výskytu přechodu do jiných než naznačených stavů.

K obdobné situaci může dojít i v případě jiných než dvou výše zmíněných instancí. Výslednou chůzi proto nelze považovat za cyklickou a nelze ji hodnotit pomocí kvantitativních kritérií popsaných v kap. 2.1.3. Vizualně lze posoudit výslednou chůzi pomocí přiloženého videozáznamu, viz. příloha A soubory *chuze_sklon_1.avi* a *chuze_sklon_2.avi*. Popsané dodatečné aktivace řídicích členů odpovídají opakované snaze robotu o docílení stabilního postavení na nohách 0,1,2 (resp. 1,2,3), což lze ve videozáznamu chůze pozorovat. Výslednou chůzi je možné označit jako nespojitou a plazivou.

4 Závěr

4.1 Shrnutí

Tato disertační práce se zabývá způsobem využití opakovaně posilovaného učení pro plánování chůze čtyřnohého robotu. Je rozdělena do 4 hlavních částí Úvod, Teoretický rozbor, Souhrn dosažených výsledků a Závěrečná diskuse.

V první kapitole je vedle motivační části uvedeno zadání práce a jsou zde stanoveny hlavní cíle, kterých má být dosaženo.

Druhá kapitola se věnuje teoretickému rozboru problematiky. Jejím hlavním cílem je poskytnut podrobný popis a kritické zhodnocení teoretických východisek. Kapitoly 2.1 a 2.2 zavádí terminologii a způsob hodnocení chůze kráčivých robotů. Těchto kritérií je později použito pro vyhodnocení dosažených stylů chůze. Kapitola 2.3 popisuje zvolenou třívrstvou architekturu, založenou na paradigmatu DEDS, která umožní plánování chůze robotu pomocí algoritmu Q-učení. Jednotlivé vrstvy architektury jsou diskutovány v kap. 2.3.1, 2.3.2 a 2.3.3. Problematice strojového učení je věnována samostatná kapitola 2.4. Zde je vedle teoretického rozboru a popisu vybraných algoritmů této oblasti obecně popsán algoritmus Q-učení, kap. 2.4.3, a především konkrétní způsob jeho propojení se zmíněnou architekturou řízení robotu, kap. 2.4.4. Závěrečná kapitola této části poskytuje kritické zhodnocení teoretických východisek, kap. 2.5.1, na základě kterých jsou stanoveny konkrétní dílčí cíle práce, kap. 2.5.2. Kapitola 2.5.3 uvádí postup práce, kterým se autor řídil při dosahování stanovených cílů.

Třetí kapitola předkládá dosažené výsledky a představuje vlastní práci autora. Kapitola 3.1 popisuje programové prostředky použité pro dosažení cílů a uvádí důvody jejich výběru. Následující kapitola 3.2 popisuje robotickou platformu čtyřnohého kráčivého robotu, která byla použita jako předloha modelu pro ověření vhodnosti navrženého řešení. Dále je zde popsán model robotu, který byl sestaven pro potřeby provedení simulačních experimentů. Kapitola 3.3 popisuje návrh jednotlivých elementárních řídicích členů, a ukazuje průběh jejich potenciálových funkcí pro vybrané konfigurace robotu. V kapitole 3.4 je popsán vytvořený simulační program, který slouží pro provádění simulačních verifikačních experimentů. Program implementuje všechny tři vrstvy architektury řízení popsané v teoretickém rozboru, dále zahrnuje vytvořený model robotu a prostředí a zajišťuje činnost algoritmu Q-učení. Výstupy tohoto programu jsou diskutovány v následujících kapitolách.

Kapitola 3.5 ukazuje výsledné elementy chování robotu pod vlivem aktivací elementárních řídicích členů. Dále jsou zde diskutována vybraná chování robotu pod vlivem paralelních aktivací kompozitních řídicích členů. V obou případech je robot umístěn na vodorovnou rovinu do blízkosti neznámých překážek. V kapitole 3.6 jsou demonstrovány schopnosti algoritmu Q-učení nalézt optimální strategii chůze robotu pro tři odlišné úlohy: úloha rotace robotu, kap. 3.6.1, úloha dopředné chůze po vodorovné rovině, kap. 3.6.2, a úloha chůze po nakloněné rovině, kap. 3.6.3. Pro každou z těchto úloh je stanovena báze řízení, výchozí množina elementárních řídicích členů, odvozena množina kompozitních řídicích členů, sestaven model všech možných chování robotu a navrženy parametry učení. Výsledné chůze robotů jsou následně hodnoceny jak

kvantitativně, pomocí kritérií uvedených v kap. 2.1.3, tak kvalitativně prostřednictvím videozáznamů zveřejněných v příloze A.

V kapitole 4 je kromě celkového shrnutí práce uvedena diskuse dosažených výsledků a přínos práce.

4.2 Diskuse dosažených výsledků

Hlavním úkolem této práce bylo předložit návrh adaptivního řídicího systému čtyřnohé robotu založeného na Q-učení. Prioritu představovalo nalezení optimálních simultánních aktivací řídicích členů v různých stavech robotu spolu s vhodnou diskretizací spojitého stavového prostoru, která by umožnila využití algoritmu Q-učení pro plánování chůze robotu. Konkrétní požadavky na řídicí systém uvádí, že by měl robotu umožnit řešení tří úloh: kráčivý pohyb v přímém směru, vyhýbání se překážkám a překonání vybraných nouzových stavů robotu. V rámci státní doktorské zkoušky byly s ohledem na zadání dále stanoveny tyto cíle:

- návrh a implementace množiny elementárních chování, vhodných pro plánování chůze čtyřnohé robotu,
- implementace vhodného koordinačního, resp. plánovacího schématu sloužícího pro propojování těchto chování, např. přístup založený na DEDS
- dosažení různých stylů chůze čtyřnohé robotu pomocí navrženého řešení,
- verifikační experiment použitého přístupu

Návrh adaptivního řídicího systému robotu byl v práci vyřešen zpracováním a implementací hybridního řídicího systému založeného na paradigmatu DEDS. Jedná se o komplexní třívrstvou architekturu řízení, která kombinuje výhody dvou základních přístupů, využívajících buďto ucelený model soustavy nebo množinu elementárních chování. Jednotka řízení byla sestavována po jednotlivých vrstvách. Návrh každé z nich si vyžádal řešení specifických problémů.

V rámci nejnižší vrstvy bylo nutné sestavit množinu řídicích členů, které by zajistili provádění elementů chování robotu. Pro dosažení chůze robotu se jako stěžejní ukázal návrh řídicího členu postoje, kontaktu a trajektorie. Každý z těchto řídicích členů je charakterizován svým předpisem potenciálové funkce, který je parametrizován. Jako parametry lze použít vstupní a výstupní zdroje robotu. Tímto způsobem je v práci sestaveno celkem 28 různých instancí řídicích členů, které zajišťují různé elementární chování robotu. Pomocí návrhu tří různých funkcí tak lze dosáhnout velkého množství různých chování, což lze jednoznačně považovat za výhodu tohoto způsobu řešení.

Řídicí člen postoje je v odborném diskurzu popsán dostatečně a v rámci práce je implementován způsobem uvedeným v kap. 3.3.1. Během návrhu se ukázalo, že je v odborné literatuře nevhodně uvedena normalizace úhlů použitých pro výpočet. Uvedený způsob by totiž vedl na průběh funkce, který má více lokálních extrémů, znemožňujících pozdější využití gradientní metody pro hledání globálního extrému funkce.

Rešeršní studie dále ukázala nutnost vytvořit vlastní návrh řídicího členu kontaktu, podrobně viz. kap. 2.3.1.2. Jeho cílem je dosažení optimálního rozložení sil reakcí působících v bodech došlapu noh. Navržený člen je založen na výpočtu velikosti směrodatné odchylky sil působících v došlapech vybraných noh robotu. Navržený

předpis zajistil vhodný charakter výsledné funkce. Jak je ukázáno pro jednotlivé instance v kap. 3.3.2, funkce má vždy jen jeden lokální a zároveň globální extrém. Řídicí člen je schopný pracovat nejen na vodorovné, ale i nakloněné rovině. Nad rámec zadání byl totiž návrh obou zmíněných řídicích členů dále upraven tak, aby byly schopny plánovat pohyb noh robotu i v případě výskytu nerovností. Z tohoto důvodu je noha robotu použita pro prozkoumávání a mapování nejbližšího okolí bodu došlapu, čehož je úspěšně využíváno k predikci optimální polohy došlapu. Pokud např. během posuvné fáze nohy dojde ke kontaktu nohy s neočekávanou překážkou resp. nerovností, pokusí se řídicí člen překážce vyhnout změnou trajektorie, viz. kap. 3.5.1. Pokud dojde opakovaně k neúspěšnému vyhnutí se překážce, je noha vrácena do poslední známé konfigurace opěrné fáze a nadřazené vrstvě řídicí jednotky je předána informace o neúspěchu. Takovéto chování lze označit jako reaktivní, což je jeden ze stanovených požadavků.

Další z výše uvedených cílů vyžaduje implementaci vhodného koordinačního schématu sloužícího pro propojování navržených elementů chování. Propojování chování je ve zvolené architektuře řídicí jednotky prováděno v rámci prostřední vrstvy. Na této úrovni dochází pomocí formálního popisu elementárních zákonů řízení a operátoru podřízenosti, $<$, k vytváření kompozitních řídicích členů. Tyto složené členy provádí paralelní aktivace instancí základních řídicích členů, čímž je dosahováno složitějších chování robotu. Může tak například docházet v jeden okamžik k rotačnímu pohybu těla robotu a optimalizaci postavení vybrané nohy. Z uvedených 28 základních členů je možné sestavit až 784 dvoučlenných kompozitních řídicích členů. Množinu kompozitních řídicích členů je možné vyvodit automaticky a celý proces lze efektivně algoritmizovat, což je jedna z výhod tohoto řešení. Hlavní problém, který musel být řešen v rámci prostřední vrstvy architektury DEDS, byla právě algoritmizace činnosti paralelních aktivací teoreticky popsaných v kap. 2.3.2, a také sestavení modelu všech možných chování robotu. Pro oba tyto problémy byl stanoven a následně implementován algoritmus uvedený v kap. 3.4.3. Výsledná chování kompozitních členů splnila očekávání, jak ukazuje i kap. 3.5.2, popisující vybrané instance. O vhodnosti návrhu svědčí i fakt, že jsou složené řídicí členy zařazeny do výsledných naučených strategií řízení robotu.

V rámci nejvyšší vrstvy řídicí jednotky byl implementován algoritmus Q-učení, jakožto zástupce strojového učení. Podřízená vrstva poskytuje učicímu algoritmu informaci o tom, ve kterém stavu se robot nachází a které z dostupných zákonů řízení (elementárních či kompozitních) je možné aktivovat. Pro výpočet jednotlivých Q-hodnot byla použita rovnice (21) uplatňující iterativní výpočet. Aby bylo možné ověřit vhodnost použití algoritmu Q-učení pro potřeby plánování chůze robotu, bylo přistoupeno k simulačnímu řešení několika komplexních úloh a následnému vyhodnocení dosažených stylů chůze.

Dle stanovených cílů je nutné provést verifikační experimenty použitého přístupu. Pro tyto potřeby byl vytvořen rozsáhlý komplexní software projekt, který je určen pro řešení dynamických simulací robotických soustav v rozličných prostředích. Je schopen načíst prostředí, zvolenou robotickou soustavu a při dodržení stanovených požadavků také algoritmus, který má být použit pro řízení soustavy. Dále zajišťuje provádění simulace, během které poskytuje jak grafický tak textový výpis aktuálního stavu. Tento simulační program lze považovat za jeden z přínosů práce, neboť je navržen obecně a byl

již v předchozích verzích použit i pro simulační řešení jiných problémů z oblasti robotiky, např. simulaci dynamické chůze robotu se dvěma stupni volnosti na nohu. Během vytváření projektu musela být vyřešena celá řada dílčích problémů souvisejících především se zvolenými prostředky, jako je fyzikální *engine* ODE, vykreslování simulace ve vlastní režii pomocí OpenGL apod. Řešení je podrobně diskutováno v kap. 3.1 a 3.4.

Verifikační experimenty jsou prováděny s využitím dynamického modelu čtyřnohého robotu se třemi stupni volnosti na každou z noh. Předlohou tohoto modelu je robot EQ3-KT, viz. kap. 3.2. Přestože je řídicí jednotka použita pro plánování chůze konkrétního robotu, je celý přístup zpracován s maximálním důrazem na obecnost tak, aby jej bylo možné použít i pro jiné robotické platformy. Navíc celý přístup lze použít i pro řízení jiných mechatronických soustav než pro kráčivé roboty.

Pomocí vytvořeného simulačního řešení byla testována schopnost algoritmu Q-učení nalézt optimální strategii simultánních aktivací pro řešení celkem tří hlavních úloh: rotace robotu, chůze v přímém směru po vodorovné rovině a chůze po nakloněné rovině. Během úloh nemá řídicí jednotka k dispozici informaci o okolním prostředí. Jestliže má být robot schopen kráčet po libovolné trajektorii, naplánované globálním plánovačem, pak musí být schopen nejen přímé chůze ve směru longitudinální osy, ale i rotace kolem svislé osy. Z tohoto důvodu bylo přistoupeno k řešení úlohy rotace robotu, i když není explicitně uvedena ve stanovených cílech.

Prostřednictvím učícího algoritmu byla při řešení úlohy rotace nalezena cyklická chůze, která je u čtyřnohých považována za energeticky optimální. Stav robotu je reprezentován stavovým vektorem se šesti složkami. Výchozí množina základních řídicích členů obsahuje 14 různých instancí řídicího členu postoje a kontaktu. V rámci prostřední vrstvy řídicí jednotky bylo vyvozeno celkem 29 kompozitních řídicích členů a výsledný model chování obsahuje celkem 32 různých stavů. Nejlepších výsledků dosáhl učící algoritmus při počátečním nastavení parametru učení $\alpha = 0,6$. Tento parametr byl postupně snižován s krokem $\Delta\alpha = 0,2$ až na hodnotu 0, při dosažení výsledné chůze. Srážkový faktor γ byl nastaven experimentálně na hodnotu 0.2. Okamžitá odměna, r , byla volena z intervalu $\langle -1, 0 \rangle$ dle velikosti změny úhlu bočení těla robotu. Ve výsledné strategii řízení je uplatňována aktivace čtyř instancí základních řídicích členů a dvou instancí složených řídicích členů. Výslednou chůzi lze považovat za cyklickou, otáčivou, regulární, plazivou a nespojitou. Činitel využití je u všech noh totožný 79,75%. Vyhodnocení výsledné chůze robotu pomocí těchto standardizovaných ukazatelů představuje splnění dalšího ze stanovených cílů. Chůzi lze posoudit také pomocí diagramu chůze, viz. Obr. 34, a přiloženého videozáznamu, viz. příloha A.

Při řešení úlohy chůze ve směru longitudinální osy robotu, byla algoritmem Q-učení rovněž nalezena cyklická chůze. Pro reprezentaci stavu byl použit opět vektor se šesti složkami, ovšem jednotlivé predikáty byly definovány jiným způsobem, viz. rov. (75). Výchozí množinu elementárních řídicích členů tvoří celkem 14 instancí z nichž bylo vyvozeno 24 kompozitních řídicích členů. Výsledný model chování zahrnuje 32 různých stavů. Parametr učení a srážkový faktor byly ponechány stejné. Velikost okamžité odměny byla přidělována opět z intervalu $\langle -1, 0 \rangle$ podle velikosti posuvu ve směru longitudinální osy robotu, viz. rov. (84). Přechody mezi stavy pod vlivem aktivací

řídících členů jsou zobrazeny na Obr. 36. Výslednou chůzi lze označit jako cyklickou, vlnitou, plazivou a nespojitou. Činitel využití pro nohy 0,3 nabývá hodnoty $\beta_{cv} = 86,5\%$, což odpovídá plazivé chůzi, a pro nohy 1,2 je $\beta_{cv} = 74\%$, což odpovídá tzv. rychlejší chůzi. Hodnoty tohoto ukazatele se liší, protože pohyb předních noh je realizován řídicím členem postoje, který manipuluje nohou jiným způsobem než řídicí člen kontaktu, který je používán pro změnu konfigurace noh zadních. Chůzi lze posoudit také pomocí diagramu chůze, viz. Obr. 37, a přiloženého videozáznamu, viz. příloha A.

Při řešení úlohy chůze po nakloněné rovině byl zvolen jiný postup, aby bylo možné posoudit adaptivní schopnost algoritmu Q-učení. Proti úloze chůze po vodorovné rovině se tato úloha odlišuje pouze změnou prostředí. Řídicí jednotka přitom nemá k dispozici žádnou informaci o okolí robotu. Při řešení této úlohy, proto byly ponechány všechny parametry týkající se reprezentace stavu, množiny řídicích členů, modelu chování i učení nezměněny. Pouze parametr učení α byl nastaven na hodnotu 0.1, aby mohla být naučená chůze postupně změněna, bude-li to zapotřebí. Výsledný styl chůze je velice podobný chůzi uplatňované v předchozí úloze. Nelze však již hovořit o cyklické chůzi. Proces konvergence některých řídicích členů vedl i při splnění cíle na odlišné konfigurace, které měli za následek opuštění hlavního cyklu chůze uplatňované v předchozí úloze. Algoritmus Q-učení byl však schopen najít optimální aktivace i pro nově vzniklé stavy. Výslednou chůzi je možné považovat vyhovující, neboť dosahuje dostatečně velkých posuvů v longitudinálním směru. Záznam chůze je uveden na videozáznamu v příloze A.

Uvedené verifikační experimenty prokázaly schopnost učícího algoritmu nalézt vyhovující vzájemně odlišné styly chůze při řešení různých úloh. Rovněž prokázaly vhodnost diskretizace spojitého stavového prostoru a ověřily funkčnost koordinačního schématu architektury DEDS. Experiment provedený na nakloněné rovině prokázal schopnost navrženého řešení adaptovat se v případě změny okolního prostředí. Představené řešení může být dále vylepšeno zjištěním pravděpodobností přechodů pod vlivem aktivací zákonů řízení. Tímto způsobem je možné odstranit nedeterminismus stavového automatu prostřední vrstvy řídicí jednotky. Díky tomu je následně možné použít i offline Q-učení a urychlit tak proces učení.

Další výhodou představuje možnost přechodu do módu řízení pomocí operátora. Tohoto způsobu lze snadno dosáhnout vypnutím nejvyšší vrstvy architektury řízení, která provádí naučený výběr z množiny dostupných zákonů řízení. Výběr akce řízení je ponechán operátorovi, přičemž aktivace zákonů řízení, které by mohli vést do nepřipustného stavu robotu, jsou automaticky vyřazovány. Operátor proto může vybírat pouze z množiny přípustných zákonů řízení a není zatěžován přílišnými detaily, které jsou řešeny na nejnižší vrstvě architektury.

Přestože je předložené řešení řídicí jednotky vztaženo ke konkrétní robotické platformě, popsany způsob návrhu je univerzální a může být s minimem nutných změn upraven pro jiný typ kráčivého robotu. Navíc předložené řešení může najít uplatnění nejen v mobilní robotice, ale také například při řešení pohybu svářečích či montážních robotů. V další práci by se proto autor práce chtěl zaměřit právě na využití uvedeného přístupu pro průmyslové manipulátory.

4.3 Přínos práce

4.3.1 Přínos teoretický

Hlavní teoretický přínos této práce spočívá ve vytvoření robustní architektury řízení kráčivého robotu, která umožňuje využití strojového učení pro nalezení různých stylů chůze. Za přínos lze dále považovat:

- zavedení vhodné diskretizace spojitého prostoru umožňující použití algoritmu Q-učení pro řízení kráčivého robotu
- navržení algoritmizace simultánních aktivací řídicích členů
- navržení řídicího členu kontaktu schopného činnosti ve strukturovaném prostředí
- zajištění reaktivního chování robotu na neznámé překážky v jeho bezprostřední blízkosti

4.3.2 Přínos praktický

Za praktické přínosy této práce lze považovat:

- vytvoření komplexního software projektu, který umožňuje provádět dynamické simulace robotických soustav v různých prostředích
- popsany způsob řízení robotu může být snadno využit i pro řízení jiných mechatronických soustav
- uvedení průběhu funkcí použitých řídicích členů pro různé konfigurace robotu
- vyhodnocení naučených stylů chůze pomocí standardizovaných ukazatelů

4.3.3 Přínos pedagogický

- v rámci řešení této disertační práce bylo vytvořeno několik bakalářských prací zabývajících se mobilní robotikou [33], [35]
- zveřejněné videozáznamy chůze robotu ukazují poutavou formou činnost strojového učení, čehož může být využito pro motivaci zájmu studentů o algoritmy umělé inteligence
- vytvořený software projekt umožní studentům se zájmem o robotiku jednoduchým způsobem navrhnout simulace vlastních robotických systémů

Literatura

- [1] Alexander R.M.: The gaits of bipedal and quadrupedal animals, *Int. J. Robotics Research* 3(2): 49-59, 1984.
- [2] Bar-Cohen Y.: Biologically Inspired Intelligent Robots Using Artificial Muscles, *International Conference on MEMS, NANO and Smart Systems (ICMENS'03)*, 2003.
- [3] Beer R.D., Quinn R.D., Chiel H.J., Ritzmann R.E.: Biologically inspired approaches to robotics: What can we learn from insects?, *Communications of the ACM*, Vol.40 No.3, March 1997.
- [4] Billard A., Ijspeert A.: Biologically Inspired Neural Controllers for Motor Control in a Quadruped Robot, *Int. Joint Conf. on Neural Networks (IJCNN'00)*, 2000.
- [5] Billard A.: Learning Motor Skills By Imitation: A Biologically Inspired Robotic Model, *Int. journal Cybernetics and Systems*, 32:155-193, 2001.
- [6] Bor K., Březina T.: Návrh a implementace kompozitního řídicího členu chůze 4-nohého robotu, *Diplomová práce, Vysoké Učení Technické v Brně, Fakulta Strojního Inženýrství*, 2004.
- [7] Brooks R.A.: A robot that walks; emergent behaviors from carefully evolved network. *Neural Computation*, 1(2): 355-363, 1989.
- [8] Connolly Ch., Grupen R.: *Applications of Harmonic Functions to Robotics*, Univ. of Massachusetts, Amherst, 1992.
- [9] Ehrenberger Z, Kolíbal Z.: *Průmyslové Roboty III*, Nakladatelství VUT Brno, Brno, 1993.
- [10] Grupen R.A.: *A Developmental Organization for Robot Behavior*, University of Massachusetts, Amherst, 2004.
- [11] Hildebrand M.: *Symmetrical gaits of horses*. Science, 1965.
- [12] Hardarsson F.: *Stability analysis and synthesis of statically balanced walking for quadruped robots*, Doctoral Thesis, Royal Institute of Technology, Stockholm, 2002.
- [13] Hoff J., Bekey G.A.: A cerebellar Approach to Adaptive Locomotion for Legged Robots, *Proc. Int. Symposium on Computational Intelligence in Robotics and Automation (CIRA'97)*, 1997.
- [14] Huber M., Grupen R.: *A Hybrid Discrete Event Dynamic Systems Approach to Robot Control*, Technical Report #96-43, Univ. of Massachusetts, 1996.
- [15] Huber M.: *A Hybrid Architecture for Adaptive Robot Control*. Ph.D. Dissertation, Univ. of Massachusetts, Amherst, 2000.
- [16] Jefferson A., Coelho Jr., Grupen R.: *Effective Multifingered Grasp Synthesis*, Technical Report, Univ. of Massachusetts, Amherst, 1994.

- [17] Ishiguro A., Watanabe Y., Uchikawa Y.: An Immunological Approach to Dynamic Behavior Control for Autonomous Mobile Robots, Proc. Int. Conf. on Intelligent Robots and Systems (IROS '95), 1995.
- [18] Jeong K., Yang T., Oh J.: A study on the support pattern of a quadruped walking robot for aperiodic motion, In. Proc. IROS, str. 308-313, Pittsburgh, PA, 1995.
- [19] Kopinec M.: Návrh a realizace senzorické soustavy náklonu kráčivého mobilního robota, diplomová práce, Ústav mechaniky těles, mechatroniky a biomechaniky, Fakulta strojního inženýrství, VUT v Brně, 2004.
- [20] Krček P., Dvořák J.: Plánování Cesty Autonomního Lokomočního Robotu na Základě Strojového Učení, disertační práce, Vysoké Učení Technické v Brně, Brno, 2010.
- [21] Krejsa J., Věchet S.: Rapidly Exploring Random Trees Used for Mobile Robots Path Planning. Inženýrská Mechanika 12 (4) (2005) pp. 231-238, 2005.
- [22] MacDonald W.S.: Legged Locomotion over Irregular Terrain using the Control Basis Approach, Master Project, University Of Massachusetts, Amherst, 1996.
- [23] MacDonald W.: Design and implementation of four-legged walking robot, BS thesis, ECE dept., MIT, Amherst, 1994.
- [24] McGhee R.B., Frank A. A.: On the stability properties of quadruped creeping gaits, Mathematical Biosciences, 1968.
- [25] Mařík, V. a kol.: Umělá inteligence 1, Academia, Praha, 1993.
- [26] Mochida T., Ishiguro A., Aoki T., Uchikawa Y.: Behavior arbitration for autonomous mobile robots using emotion mechanisms, Proc. Int. Conf. on Intelligent Robots and Systems (IROS '95), 1995.
- [27] Ondroušek V.: Návrh a implementace koordinačního mechanismu řídících členů chůze čtyřnohého robota, diplomová práce, Vysoké Učení Technické v Brně, Fakulta Strojního Inženýrství, 2004.
- [28] Pettersson L., Jansson K., Rehbinder H., Wikander J.: Behavior-based Control of a Four Legged Walking Robot, Department of Machine Design, Royal Institute of Technology, Sweden, 1998.
- [29] Quinn R. D., Nelson G. M., Bachmann R. J., Kingsley D. A., Offi J., Ritzmann R. E.: Insect Designs for Improved Robot Mobility, 4th International Conference On Climbing and Walking Robots, Berns and Dillmann eds., 69-76, 2001.
- [30] Rai L., Kang S.J., Rim K.: Dynamic Skill-Based Software Architecture for Locomotion Control in Shape Reconfigurable Mobile Robots, Proc. IEEE Workshop on Software Technologies for Future Embedded and Ubiquitous Systems and 2nd Int. Workshop on Collaborative Computing, Integration, and Assurance (SEUS-WCCIA'06), 2006.
- [31] Ridderström CH.: Legged locomotion: Balance, control and tools - from equation to action, Royal Institute of Technology, Stockholm, 2003.

- [32] Řeřucha V.: Inteligentní řízení kráčejícího robota, Habilitační práce, Vojenská Akademie v Brně, 1997.
- [33] Seriš R.: Využití ODE pro Sestavení Dynamického Modelu Čtyřnohého Robotu, bakalářská práce, VUT v Brně, Brno, 2008.
- [34] Shavlik J., Dietterich, T.: Readings in Machine Learning, San Francisco, Morgan Kaufmann, pp. 267-283, 1990.
- [35] Schreiber P.: Návrh Automatického Generátoru Prostředí Pro Mobilní Robot, bakalářská práce, VUT v Brně, Brno, 2008.
- [36] Song S., Waldron K. J.: Machines that Walk., MIT Press, Cambridge, MA, 1989.
- [37] Spennenberg D. a kol., Aramies: A Four-Legged Climbing and Walking Robot, Faculty of Mathematics and Computer Science, University of Bremen, 2005.
- [38] Thrun S., Wolfram B., Dieter F.: A Real-Time Algorithm for Mobile Robot Mapping With Applications to Multi-Robot and 3D Mapping, proc. on IEEE International Conference on Robotics and Automation, San Francisco, 2000.
- [39] U.S.Army: Logistical vehicle off-road mobility. Fort Eustis, Va.: U.S. Army, Transportation Combat Developments Agency, feb 1967. (Project TCCO 62-5).
- [40] Věchet S., Krejsa J., Grepl R.: How to build a robot with no money, merkur, lego and old stepper motor, proc. of Engineering Mechanics 2005, Praha, Institute of Thermomechanics, Theoretical and Applied Mechanics, Academy of Sciences of Czech Republic, 2005, pp. 335-336, ISBN: 80-85918-93-5.
- [41] Věchet S., Krejsa J., Ondroušek V.: The Development of Autonomous Racing Robot Bender, časopis Inženýrská Mechanika, s. 277-287, 2007.
- [42] Wettergreen D., Pangels H., Bares J.: Behavior-based gait execution for the Dante II walking robot, Proc. Int. Conf. on Intelligent Robots and Systems (IROS '95), 1995.
- [43] Watkins, C. J. C. H.: Learning from delayed rewards, PhD thesis, University of Cambridge, Cambridge, England, 1989.
- [44] Witten, I. H.: An adaptive optimal controller for discrete-time Markov environments, Information and Control, 34:286-295, Academic Press, Londýn, 1977, ISSN 0019-9958.
- [45] Wolfram B., a kol.: Experiences with an interactive museum tour-guide robot, J. of Artificial Intelligence, 1999.

Internetové zdroje:

- [46] Dvořák J.: Přednášky předmětu Expertní systémy, dostupné z www.uai.fme.vutbr.cz/~jdvorak/vyuka/es/es.htm, cit. dne 5.3.2011.
- [47] Nilsson J.: Introduction to Machine Learning, Stanford University, 2005. Dostupné z <http://robotics.stanford.edu/~nilsson/MLBOOK.pdf>, cit. dne 2.4.2011.
- [48] Open Dynamics Engine, doztupné z www.ode.org, cit. dne 20.5.2011.
- [49] Robot EQ3-KT, fa Lynxmotion, dostupné z www.lynxmotion.com, cit. dne 12.10.2008.

- [50] Freese M.: Virtual Robot Experimentation Platform, dostupné z www.v-rep.eu, cit. dne 5.3.2011.
- [51] Woergoetter F., Porr B.: Reinforcement learning, Scholarpedia, 3(3): 1448, 2008, dostupné z www.scholarpedia.org/article/Reinforcement_learning, cit. dne 8.3.2011.
- [52] Palčák F.: Možnosti využití programu ADAMS pro studium, výuku i výzkum, odborná přednáška, FSI VUT, 21.3.2011, dostupné z mechatronika.fme.vutbr.cz/podporaodbornehovzdelavaniclenutymu/prednaskydetail/4, cit. dne 2.4.2011.
- [53] slovník cizích slov, dostupné z slovník-cizich-slov.abc.cz, cit. dne 21.8.2007.

Seznam obrázků

Obr. 1: Schématické naznačení os rotace	11
Obr. 2: Grafy vybraných pojmenovaných stylů chůze koně, [31].	14
Obr. 3: Symbolický popis řídicího členu, [15]	22
Obr. 4: Předloha simulačního experimentu, robot EQ3-KT [49]	38
Obr. 5: Schéma zavedených souřadných systémů robotu	39
Obr. 6: Schéma robotu, zobrazení v rovině těla robotu	40
Obr. 7: Souřadný systém spjatý s nohou 0 robotu	40
Obr. 8: Model robotu, pohled ve směru longitudinální osy $X^{(2)}$	41
Obr. 9: Model robotu, pohled ve směru osy $Z^{(2)}$	42
Obr. 10: Konfigurace robotu č. 7	43
Obr. 11: Konfigurace robotu č. 8	43
Obr. 12: Konfigurace robotu č. 11	43
Obr. 13: Potenciálová funkce řídicího členu postoje, nevhodná normalizace	47
Obr. 14: Potenciálová funkce řídicího členu postoje, konfigurace č. 1	48
Obr. 15 : Potenciálová funkce řídicího členu postoje, konfigurace č. 7	48
Obr. 16: Potenciálové funkce řídicího členu postoje pro různé výšky těla nad zemí	49
Obr. 17: Potenciálové funkce řídicího členu $\Phi_{P_{x,z}}^{0,1,2,3}$ pro konfiguraci č. 1	49
Obr. 18: Potenciálové funkce řídicího členu $\Phi_{P_{x,z}}^{0,1,2,3}$ pro konfiguraci č. 7	50
Obr. 19: Robot umístěný na nakloněné rovině	52
Obr. 20: Síly působící na robot v bodě došlapu N^0	52
Obr. 21: Potenciálová funkce řídicího členu $\Phi_{K_{x,z}}^{0,2,3}$ pro konfiguraci č. 11	54
Obr. 22: Potenciálová funkce řídicího členu $\Phi_{K_{x,z}}^{0,2,3}$ pro konfiguraci č. 8	55
Obr. 23: Potenciálová funkce řídicího členu $\Phi_{K_1}^{0,1,3}$	55
Obr. 24: Příklad vstupu programu, a) Xode formát prostředí b) přímá tvorba modelu	57
Obr. 25: Ukázka uložených informací o modelu chování systému	58
Obr. 26: Příklad textového výstupu během procesu učení	58
Obr. 27: Algoritmus sestavení model všech možných chování systému	61
Obr. 28: Algoritmus činnosti kompozitního řídicího členu	62
Obr. 29: Vliv aktivace $\Phi_{P_{\varphi}}^{0,1,2,3}$ v členitém terénu, výchozí a cílová konfigurace	64
Obr. 30: Výsledek konvergence instance $\Phi_{P_{x,z}}^{0,1,2,3}$, různé pohledy	65
Obr. 31: Vliv aktivace $\Phi_{P_2}^2$ v členitém terénu, výchozí a cílová konfigurace	65
Obr. 32: Vliv aktivace $\Phi_{K_{x,z}}^{1,2,3}$ v členitém terénu, výchozí a cílová konfigurace	66
Obr. 33: Reaktivní chování řídicího členu $\Phi_{K_2}^{0,2,3}$ při výskytu neznámé překážky	67
Obr. 34: Graf chůze v úloze rotace	73
Obr. 35: Přejechy mezi stavy robotu v úloze rotace	74
Obr. 36: Přejechy mezi stavy robotu v úloze dopředné chůze po rovině	78
Obr. 37: Graf chůze v úloze dopředné chůze po rovině	79
Obr. 38: Přejechy mezi stavy robotu v úloze dopředné chůze po nakloněné rovině	80

Seznam příloh

Příloha A

CD-ROM obsahující:

1) Zdrojové kódy projektu

\projekt\projekt zdrojove kody v1.13.zip

2) Videozáznamy provedených simulací ukazující chování robotu

\simulace zakladni\..

kontakt_noha_1.avi – řídicí člen kontaktu měnící konfiguraci nohy

kontakt_noha_2.avi – řídicí člen kontaktu měnící konfiguraci nohy, jiný pohled

kontakt_translace_1.avi – řídicí člen kontaktu posouvající tělo robotu

kontakt_translace_2.avi – řídicí člen kontaktu posouvající tělo robotu, jiný pohled

postoje_boceni_1.avi – řídicí člen postoje měnící úhel bočení těla robotu

postoje_boceni_2.avi – řídicí člen postoje měnící úhel bočení těla robotu, jiný pohled

postoje_noha_1.avi – řídicí člen postoje měnící konfiguraci nohy

postoje_noha_2.avi – řídicí člen postoje měnící konfiguraci nohy, jiný pohled

postoje_translace_1.avi – řídicí člen postoje posouvající tělo robotu

postoje_translace_2.avi – řídicí člen postoje posouvající tělo robotu, jiný pohled

trajektorie_boceni_1.avi – řídicí člen trajektorie měnící úhel bočení

trajektorie_boceni_2.avi – řídicí člen trajektorie měnící úhel bočení, jiný pohled

\simulace kompozitni\..

kontakt_noha_postoj_rot1.avi – ř.č. kontaktu nohy a ř.č. postoje měnící úhel bočení

kontakt_noha_postoj_rot2.avi – stejná aktivace, pohled shora

postoj_rotace_vyska.avi – ř.č. postoje měnící úhel bočení a výšku

postoj_vyska_noha.avi – ř.č. postoje měnící výšku a konfiguraci nohy

\simulace komplexni\..

rotace_rovina.avi – výsledná chůze robotu při řešení úlohy rotace

chuze_rovina.avi – výsledná chůze robotu po vodorovné rovině

chuze_sklon_1.avi – výsledná chůze robotu po nakloněné rovině

chuze_sklon_2.avi – výsledná chůze robotu po nakloněné rovině