

UNIVERZITA KOMENSKÉHO V BRATISLAVE
FAKULTA MATEMATIKY, FYZIKY A INFORMATIKY

TESTOVANIE REALITNEJ MEDZERY U
HUMANOIDNÉHO ROBOTA
BAKALÁRSKA PRÁCA

2024

JURAJ GAVURA

UNIVERZITA KOMENSKÉHO V BRATISLAVE
FAKULTA MATEMATIKY, FYZIKY A INFORMATIKY

TESTOVANIE REALITNEJ MEDZERY U
HUMANOIDNÉHO ROBOTA
BAKALÁRSKA PRÁCA

Študijný program: Aplikovaná informatika
Študijný odbor: Informatika
Školiace pracovisko: Katedra aplikovanej informatiky
Školiteľ: prof. Ing. Igor Farkaš, Dr.
Konzultant: Mgr. Michal Vavrečka, PhD.

Bratislava, 2024
Juraj Gavura



ZADANIE ZÁVEREČNEJ PRÁCE

Meno a priezvisko študenta: Juraj Gavura
Študijný program: aplikovaná informatika (Jednoodborové štúdium, bakalársky I. st., denná forma)
Študijný odbor: informatika
Typ záverečnej práce: bakalárska
Jazyk záverečnej práce: slovenský
Sekundárny jazyk: anglický

Názov: Testovanie realitnej medzery u humanoidného robota
Testing the reality gap in sim-to-real transfer with a humanoid robot

Anotácia: Simulátory sú široko používané v robotickej komunite, pretože umožňujú rýchle a lacné prototypovanie systémov v reálnom svete bez potreby fyzického prístupu k hardvéru. Avšak kvôli existujúcim nezrovnalostiam medzi simuláciami a reálnym svetom (realitná medzera) predstavujú problémy pre priamy prenos vytvoreného matematického modelu zo simulátora do fyzického robota.

Cieľ: 1. Vyberte si vhodný robotický simulátor pre humanoidného robota NICO a otestujte jeho funkčnosť. 2. Navrhňte jednoduchú motorickú úlohu s jedným ramenom smerujúcim na rôzne body na virtuálnej horizontálnej dotykovej obrazovke umiestnenej pred robotom a zaznamenajte 2D súradnice pozícií, ktorých sa dotkne ukazovák robota. 3. Zopakujte rovnakú úlohu s fyzickým NICO a skutočnou dotykovou obrazovkou. 4. Kvantifikujte chybu spôsobenej rozdielom medzi realitami a zamyslite sa nad možnosťami jej minimalizácie.

Literatúra: Collins J. (2019). Quantifying the Reality Gap in Robotic Manipulation Tasks. IEEE Int. Conf. on Robotics and Automation
Kerzel M. et al. (2017). NICO — Neuro-inspired companion: A developmental humanoid robot platform for multimodal interaction. In IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)
Salvato E. et al. (2021). Crossing the Reality Gap: A Survey on Sim-to-Real Transferability of Robot Controllers in Reinforcement Learning. IEEE Access.

Vedúci: prof. Ing. Igor Farkaš, Dr.
Konzultant: Mgr. Michal Vavrečka, PhD.
Katedra: FMFI.KAI - Katedra aplikovanej informatiky
Vedúci katedry: doc. RNDr. Tatiana Jajcayová, PhD.

Dátum zadania: 04.10.2023

Dátum schválenia: 28.10.2023

doc. RNDr. Damas Gruska, PhD.
garant študijného programu

Abstrakt

S každým novým rokom rastie dopyt po rôznych druhoch robotov. Pre rýchlejšie a jednoduchšie tréningovanie robotov sa využívajú rôzne simulátory. Aj keď su simulácie v dnešnej dobe veľmi pokročilé, vždy nastávajú určité nezrovnalosti a nepresnosti medzi prostredím vo virtuálnej realite a prostredím vo fyzickej realite. Tieto chyby nazývame vo svete robotiky rozdiely v realite. Cieľom tejto práce je vybrať vhodný simulátor na simuláciu humanoidného robota NICO a následný prechod do fyzickej reality. Následne sa pokúsiť identifikovať a kvantifikovať rozdiely, ktoré nastávajú pri tomto prechode. Podarilo sa nám namerať odchýlky v dvojrozmernom priestore, rozdiely polôh stupňov voľnosti a časov vykonania pohybu. V závere sme sa zamysleli nad možnou minimalizáciou odmeraného rozdielu.

Kľúčové slová: simulácia, realitná medzera, humanoidný robot NICO

Abstract

With each new year, the demand for different kinds of robots grows. Various simulators are being used to train robots faster and easier. Although simulations are very advanced nowadays, there are always some discrepancies and inaccuracies between the virtual reality environment and the physical reality environment. In the world of robotics, these errors are called reality gap. The aim of this work is to select a suitable simulator to simulate the humanoid robot NICO and then transition to physical reality. Then to try to identify and quantify the differences that occur during this transition. We were able to measure the variations in two-dimensional space, the differences in degrees-of-freedom positions, and motion execution durations. We conclude by reflecting on the possible minimization of the measured difference.

Keywords: simulation, reality gap, humanoid robot NICO

Obsah

Úvod	1
1 Dostupná kvantifikácia rozdielu v realite	3
1.1 Identifikácia	3
1.2 Kvantifikácia	3
1.2.1 Testy a výsledky	3
1.2.2 Zhrnutie	4
2 Riešenie problému rozdielu v realite	5
2.1 Známe metódy	5
2.1.1 Randomizácia domény	5
2.1.2 Adverzariálne učenie posilňovaním	5
2.1.3 Optimalizácia simulovaných prostredí	6
2.1.4 Transferové učenie	6
2.1.5 Použitie viacerých simulátorov súčasne	6
2.2 Nové metódy	7
2.2.1 Ladenie simulátoru	7
2.2.2 Diferencovateľná fyzika	7
3 Kinematika	9
3.1 Robotické rameno	9
3.2 Dopredná kinematika	10
3.3 Inverzná kinematika	10
4 Humanoidný robot	13
4.1 Robot NICO	13
4.2 Simulátor myGym	14
5 Návrh úlohy	17
5.1 Skúmané hodnoty	17
5.2 Pohyb robota	17
5.3 Experimenty	18

6 Implementácia	19
6.1 Obmedzenia	19
6.2 Prechody medzi sústavami	21
6.3 Hľadanie hodnoty na osi z	21
6.3.1 Meranie rozdielu	21
6.3.2 Riešenie	22
6.4 Iné implementácie	24
7 Výsledky	25
7.1 Rovnaký bod	25
7.2 Ciele v línii	25
7.3 Pokrytie celej plochy	28
7.4 Náhodné body	29
Záver	33

Zoznam obrázkov

1.1	Výsledky finálneho testu obsahujúceho interakciu robotického ramena s kockou v práci od Collins a kol. [5].	4
3.1	Vizualizácia doprednej a inverznej kinematiky v práci od Yonezawa a kol. [19].	11
4.1	Reálny robot NICO vo Fakulte Matematiky, Fyziky a Informatiky v Bratislave s dotykovou obrazovkou a nádstavcom na ukazovák (koncovom efektore), ktorý umožňuje interakciu s tabletom.	15
4.2	Robot NICO v simulácii simulátoru myGym, už s predpripraveným prostredím.	16
6.1	Uhlový rozsah šiestich stupňov voľnosti pravého ramena, ktoré NICO používa pri experimentoch. Os x robota NICO je os y na dotykovej obrazovke a naopak. Takže keď má kĺb v zátvorke x , tak sa v našom systéme otáča okolo osi x a nie y . Pre robota je to os y	20
6.2	Vizualizácia dosiahnuteľnej plochy pravého robotického ramena, testovacej plochy a kvadrantov.	20
6.3	Namerané hodnoty na osi z v simulácii v časoch, kedy sa koncový efektor fyzického robota NICO jemne dotýkal obrazovky v príslušnom bode x , y . NICO Arm predstavuje polohu základne pravého ramena fyzického robota.	22
7.1	Výsledky prvého experimentu - tri rovnaké ciele - čas pohybu: 1 sekunda	26
7.2	Výsledky druhého experimentu - časť s vertikálnymi líniami - čas pohybu: 1 sekunda	26
7.3	Výsledky druhého experimentu - časť s horizontálnymi líniami - čas pohybu: 1 sekunda	27
7.4	Výsledky druhého experimentu - časť s diagonálnymi líniami - čas pohybu: 1 sekunda	28
7.5	Výsledky tretieho experimentu - celé pokrytie - čas pohybu: 1 sekunda	28

7.6	Výsledky štvrtého experimentu - náhodné ciele - porovnanie odchýliek medzi cieľovými bodmi a zásahmi	29
7.7	Výsledky štvrtého experimentu - náhodné ciele - porovnanie odchýliek medzi jednotlivými kĺbmi robotického ramena	30
7.8	Výsledky štvrtého experimentu - náhodné ciele - porovnanie času odozvy robota NICO	31
7.9	Výsledky štvrtého experimentu - náhodné ciele - porovnanie času trvania pohybu robotického ramena	32

Úvod

S každým novým rokom rastie dopyt po rôznych druhoch robotov. Či už ide o robotické rameno pracujúce s objektami, robot, ktorý sa potrebuje pohybovať v ťažkom teréne, alebo robot, ktorý využíva vizuál z kamery a musí rýchlo rozoznať na čo sa pozerá. Predtým ako každý robot začne vykonávať svoje zadanie, musí prejsť určitou učiacou fázou. Jednou zo základných techník na cvičenie robota je učenie posilňovaním (angl. reinforcement learning). Toto učenie umožňuje vytvárať ovládače, ktoré sa dokážu naučiť optimálne správanie robota interakciou s okolím. Toto správanie, ktoré môžeme považovať ako stratégiu (angl. policy), je učené procesom pokus a omyl [16]. Z tohto dôvodu je tréning robota priamo vo fyzickej realite náročné časovo aj finančne, keďže je veľmi jednoduché robota počas tréningu poškodiť [11]. Preto je potrebné nájsť lepší spôsob na prácu s robotom vo vývoji.

Pre tento účel nám slúži simulácia. Simulácia je široko využívaná vo svete robotiky vďaka tomu, že umožňuje roboty trénovať lacno, bez potreby prístupu k hardvéru a efektívne, keďže dokáže jednoducho generovať množstvo syntetických dát, ktoré podstatne zrýchlia proces učenia. Medzi ďalšie výhody simulácie patrí možnosť vykonávania viacerých inštancií simulácie paralelne, a to všetko bez nutnosti ľudskej kontroly. Bohužiaľ, simulovanie učenia vo virtuálnej realite má aj svoje nevýhody. Z dôvodu nutnej aproximácie a abstrakcie niektorých fyzických javov, vznikajú nezrovnalosti medzi virtuálnou realitou v simulácii a fyzickou realitou. Simulácie tiež zvyknú vytvárať situácie, ktoré nemôžu nastať v našej realite. Tieto vlastnosti simulácie môžu značne komplikovať prestup zo simulácie do fyzickej reality. Tento problém vo svete robotiky nazývame rozdiel v realite (angl. reality gap) [5]. Iný spôsob kontrolovania robota je priamym manuálnym ovládaním. Napríklad jednoduché natiahnutie ruky za predmetom s využitím inverznej kinematiky. Aj tu je možné si pomôcť simulátorom, kde môžeme dopredu vidieť dôsledky vykonaných akcií, a teda aj pri takomto ovládaní vieme sledovať rôzne rozdiely v realite.

Neobyčajnosť a rozsiahlosť tohto javu je z môjho pohľadu veľmi sympatická a zaujímavá. Z tohto dôvodu som si vybral ponoriť sa do tejto témy hlbšie a vyskúšať si prechody medzi realitami na vlastnej koži. Správanie a ovládanie robotov ma vždy fascinovalo a vedel som si predstaviť, koľko práce stojí za správnym fungovaním akéhokoľvek prístroju takéhoto druhu. Aj toto malo za následok, že som neváhal pri možnosti

pracovania s reálnym humanoidným robotom.

Cieľom tejto práce je nájsť vhodný simulátor pre prechod z virtuálnej reality do fyzickej reality, ktorý vie pracovať s humanoidným robotom NICO. Ďalej identifikovať a kvantifikovať rozdiel v realite pri práci s týmto robotom - teda nájsť a pokúsiť sa odmerať odlišnosti medzi robotom NICO v simulácii a skutočným robotom, ktorý sa nachádza na fakulte matematiky, fyziky a informatiky v Bratislave. Napokon posledným cieľom tejto práce je zamyslieť sa nad príčinami nameraných rozdielov v realitách a možnými úpravami, či vylepšeniami prechodu medzi simuláciou a fyzickou realitou.

V kapitole 1 a 2 sa pozeráme na už dostupné pokusy o kvantifikáciu tohto problému, výsledky a odporúčané riešenia, respektíve spôsoby minimalizácie nájdených rozdielov. V kapitole 3 hovoríme o doprednej a inverznej kinematike, ktoré sú kľúčové pre náš konkrétny výskum. Humanoidné roboty, robot NICO konkrétne a simulátor využívaný pre prácu s týmto robotom je popísaný v kapitole 4. V kapitole 5 je popísaná metodika skúmania a návrh experimentov. Nakoniec nasleduje implementácia týchto metód v kapitole 6 a výsledky experimentov so zamyslením sa nad ich možnými vylepšeniami v kapitole 7.

Kapitola 1

Dostupná kvantifikácia rozdielu v realite

1.1 Identifikácia

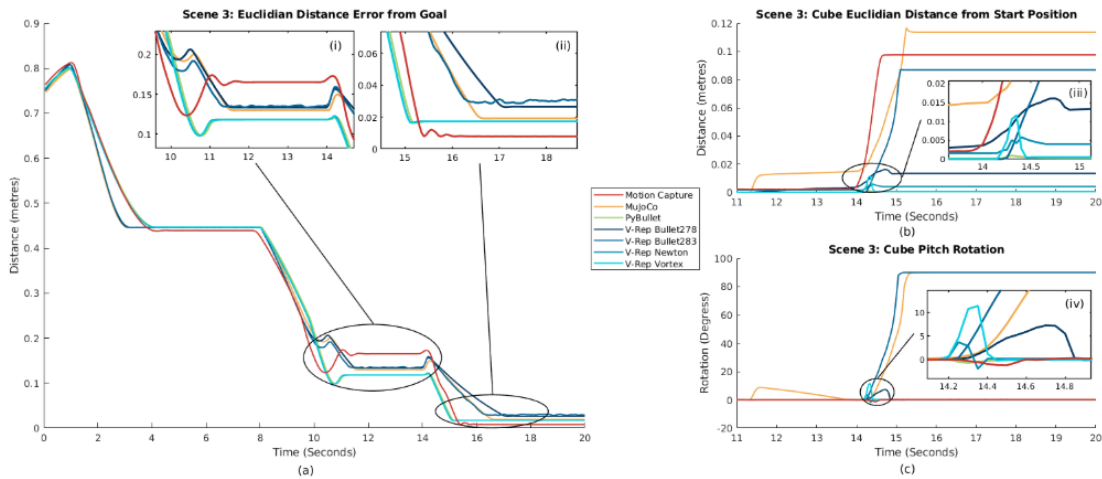
Skôr ako je možné kvantifikovať rozdiel v realite, je potrebné zistiť ako identifikujeme, že je rozdiel v realite vôbec prítomný. Predstavíme si priebeh učenia robota pomocou simulácie. Najprv sa v simulácii trénuje ovládač jednou z metód učenia a následne sa vo virtuálnej realite aj otestuje. Ak robot cieľ nesplní, ovládač sa trénuje ďalej. Ak naopak robot cieľ splní, test sa zopakuje na reálnom robotovi vo fyzickej realite. Ak nájdeme medzi výkonom robota v simulácii a výkonom robota vo fyzickej realite nejaké odlišnosti, nastal problém s prestupom do fyzickej reality a teda ide o rozdiel v realite [16]. Teraz keď sme identifikovali rozdiel v realite, môžeme sa ho pokúsiť nejakým spôsobom kvantifikovať.

1.2 Kvantifikácia

Ako uvádzajú Collins a kol. [5] vo svojej práci, táto kvantifikácia je možná do určitej miery. Otestovali najznámejšie simulátory s rôznymi fyzikálnymi motormi na sade troch testov pre prácu s robotickým ramenom Kinova Mico2 6DOF a porovnali pohyb v simulácii s pohybom v reálnom svete vyrátaním chyby pomocou euklidovskej vzdialenosti v závislosti od času. Ako simulátory si vybrali V-Rep, MuJoCo a PyBullet. Fyzikálne motory, ktoré tieto simulátory používajú sú Bullet, ODE, Vortex, Newton a MuJoCo. Testy obsahujú pohyb jednotlivých kĺbov ramena a interakciu s kockou.

1.2.1 Testy a výsledky

Prvý test zahŕňal jednoduchú rotáciu jedného kĺbu ramena o 100 stupňov za 6 sekúnd. Najlepšie si viedli simulátory PyBullet a V-Rep s fyzikálnymi motormi Vortex a Newton



Obr. 1.1: Výsledky finálneho testu obsahujúceho interakciu robotického ramena s kockou v práci od Collins a kol. [5].

napriek tomu, že v týchto simulátoroch rameno neoscilovalo tak, ako vo fyzickej realite. Fyzikálny motor Bullet v simulátore V-Rep osciloval rovnako ako rameno vo fyzickej realite, ale prišiel do finálnej pozície o sekundu skôr.

V druhom teste sa striedavo rotovali dva kĺby ramena a hral v ňom veľkú úlohu pohyb proti gravitácii. Fyzickú realitu najlepšie kopíroval V-Rep s motormi Newton a Vortex a tesne za nimi PyBullet. Ostatné motory výrazne zaostávali a žiadnemu sa nepodarilo imitovať pohyb reálneho robota presne.

Tretí, finálny a najkomplexnejší test obsahoval interakciu s kockou. Táto interakcia spočívala v nastavení ramena do polohy vedľa kocky a následné jemné potlačenie kocky, teda tento finálny test zahŕňal pohyb viacerých kĺbov ramena. Žiadnemu simulátoru sa nepodarilo kopírovať pohyb kocky dokonalo. Najbližšie bol PyBullet, no iba z dôvodu, že kontakt s kockou nezaregistroval vôbec, rameno vošlo do kocky. MuJoCo a V-Rep s motorom Bullet boli najlepšie čo sa týka pohybu kocky, no zarotovali kocku o 90 stupňov, pričom v reálnom svete sa kocka zarotovala minimálne, detaily je možné vidieť na obrázku 1.1.

1.2.2 Zhrnutie

Z testov môžeme vidieť, že existujú značné rozdiely medzi jednotlivými simulátormi a fyzikálnymi motormi. Niektoré vynikali v jednom type scenáru, iné zase v druhom. Teda ideálne by bolo, keby bolo možné využiť jeden motor na jednu časť pohybu a druhý na inú. No je očividné, že simulácia interakcie dvoch pevných objektov potrebuje výrazné zlepšenie.

Kapitola 2

Riešenie problému rozdielu v realite

2.1 Známe metódy

Problém rozdielu v realite vo svete robotiky existuje už desiatky rokov a preto poznáme nemalý počet pokusov o riešenie tohto problému. [16] a tiež [5] vo svojich prácach uvádzajú niekoľko potenciálnych riešení.

2.1.1 Randomizácia domény

Jedna z najpoužívanějších metód je randomizácia domény (angl. domain randomisation), ktorá už v minulosti mala pomerne dobré výsledky. V tomto prípade je trénovaný model vystavený náhodným vstupom a tým získa ovládač väčšiu robustnosť. Teda cieľom je, aby robot dokázal splniť svoju úlohu napriek rôznym prekážkam. Môžeme si to predstaviť na trénovaní robotického videnia, kde by model prijímal vstup ako náhodné hodnoty napríklad farby, tieňovania alebo pozície kamery, no stále by vedel rozoznať hľadaný objekt [5]. Mnoho ďalších metód na riešenie rozdielu v realite využíva náhodný výber. Adaptačná sieť s náhodným výberom (angl. randomised-to-canonical adaptation network) je spôsob randomizácie, pri ktorom sa pozorovania zozbierané vo fyzickej doméne spolu s pozorovaniami zo simulovanej domény zobrazia do spoločnej domény, ktorú nazývame kanonická doména [16]. Na podobnom princípe využívania dát zo simulácie spolu s dátami z reálneho sveta fungujú metódy ako adaptácia domény (angl. domain adaptation) a generatívne adverzariálne siete (angl. generative adversarial networks) [5].

2.1.2 Adverzariálne učenie posilňovaním

Inou metódou riešiacou problém rozdielu v realite je adverzariálne učenie posilňovaním (angl. adversarial reinforcement learning). Pri tomto spôsobe učenia sa trénujú dva ovládače, jeden sa nazýva protagonista a druhý antagonista. Úlohou druhého ovládača

je generovať stále zložitejšie a zložitejšie modely na zvládnutie cieľu robota. Na túto metódu nadväzuje algoritmus nazývaný robustné adverzariálne učenie posilňovaním. V tomto prípade je dodatočná spoľahlivosť zaručená tým, že antagonista navyše vnáša do modelu prostredia destabilizujúce perturbácie [16].

2.1.3 Optimalizácia simulovaných prostredí

Na rozdiel od randomizácie, optimalizácia simulovaných prostredí je technika, v ktorej je cieľom lepšie emulovať fyzickú realitu. Samozrejme, aby bolo možné simuláciu optimalizovať, je vyžadované väčšie množstvo dát z reálneho sveta. Tým pádom je táto metóda špecifická pre každého robota a aplikáciu [5]. Doteraz spomenuté techniky redukovania rozdielu v realite spočívali v tom, že v simulácii naučené ovládače bolo možné priamo a jednoducho previezť do reálneho robota.

2.1.4 Transferové učenie

Na problém rozdielu v realite sa z iného uhla pozerá technika nazývaná transferové učenie (angl. transfer learning). Oproti technikám, ktoré sa snažia riešiť rozdiel v realite, cieľom transferového učenia je vyhnúť sa výskytu tohto rozdielu prostredníctvom dvoch súbežných alebo po sebe idúcich fáz tréningu. Prvá tréningová fáza prebieha v simulácii, druhá vo fyzickej realite. Tento spôsob vieme použiť na klasické učenie posilňovaním, teda učenie pomocou metódy pokus a omyl. Poznatky naučené v simulácii vieme využiť v nasledujúcej fáze tréningu v realite, a aj pri neúspešnom pokuse je možné recyklovať nejaké množstvo týchto poznatkov. Tým vieme výrazne znížiť čas potrebný na tréning. Transferové učenie sa delí podľa viacerých vlastností. Jedna z nich je smer prenosu, jednosmerné smeruje len zo simulácie do fyzickej reality, zatiaľ čo obojsmerné prenáša informácie aj opačným smerom. S tým súvisí ďalšia vlastnosť a to počet prenosov, ten môže byť jeden, čo je charakteristické pre jednosmerné transferové učenie, alebo môže prenos nastať viackrát, čo sa využíva pri obojsmernom type. Treťou dôležitou vlastnosťou je druh informácií, ktoré sú prenášané. Môže ísť o určité parametre alebo aj o celé stratégie. Na redukcii rozdielu v realite je špeciálne praktický obojsmerný typ transferového učenia, pri ktorom si simulátor prevezme dáta z reálneho sveta a tým vie realistickejšie prispôbiť prostredie simulácie k reálnemu svetu. Týmto zredukuje odlišnosti medzi virtuálnou a fyzickou realitou, a teda aj samotný rozdiel v realite [16].

2.1.5 Použitie viacerých simulátorov súčasne

Poslednou metódou, ktorú spomenieme je použitie viacerých simulátorov na prekonanie skreslenia jedného simulátoru. Simulátory sú odlišné v rôznych vlastnostiach, preto je možné využiť jeden ako hlavný simulátor a potom na úspešné ovládače použiť

druhý simulátor, ktorý odstráni ovládače, využívané na nerealistické scenáre. Rovnako je možnosťou využiť niekoľko fyzikálnych motorov namiesto jedného [5]. Napriek veľkému počtu rôznych techník na riešenie problému rozdielu v realite, je tento problém do veľkej miery stále prítomný. Dokonca ani nevieme určiť, ktorá z týchto techník je najefektívnejšia. Ako bolo už vyššie spomínané, potrebovali by sme nejaký spôsob ako kvantifikovať rozdiel v realite.

2.2 Nové metódy

2.2.1 Ladenie simulátoru

Collins a kol. [3] ukázali, že rozdiel v realite sa dá do veľkej miery redukovať prostredníctvom ladenia simulátoru, teda úpravy určitých parametrov podľa špecifického scenáru a robota. Táto technika obsahuje štyri fázy učenia. Prvou je kolekcia dát z reálneho sveta. Druhou fázou je samotná simulácia činnosti robota. V testovaní použili simulátory Py-Bullet a V-Rep, obidva používajú fyzikálny motor Bullet. Tieto simulátory ponúkajú širokú škálu parametrov, ktoré je možné nastavovať, preto sú vhodné pre túto metódu. Treťou fázou je selekcia parametrov simulátora. Štvrtou a poslednou fázou je optimalizácia simulátoru nastavením parametrov vybraných v tretej fáze. Na optimalizáciu využili algoritmus nazývaný diferenciálna evolúcia implementovaný v optimalizačnom module SciPy, pretože umožňuje optimalizovať veľké množstvo parametrov súčasne. Pri testovaní optimalizácie zistili, že najdôležitejšie parametre sú časový krok simulácie, laterálne trenie objektov a kĺbová rýchlosť. Táto technika vykazuje dobré výsledky aj pri interakcii s objektami.

2.2.2 Diferencovateľná fyzika

S narastajúcim počtom dostupných knižníc pre automatickú diferenciáciu prichádzajú do popredia simulátory, ktoré využívajú diferencovateľné fyzikálne motory. Tieto simulátory majú v robotike veľký potenciál, pretože sľubujú optimalizáciu simulačných parametrov, riadiacich vstupov a morfológie agenta [4].

Collins a kol. [4] navrhli novú revolučnú metódu na riešenie problému rozdielu v realite, ktorá využíva tieto knižnice. Nazvali ju RealityGrad. Pozostáva zo štyroch fáz, podobne ako ladenie simulátoru (časť 2.2.1). Prvou fázou je kolekcia optimálnych trajektórií v simulácii, ktorá bude použitá ako množina dát v nasledujúcej fáze. Druhou fázou je tréning stratégií a treťou je následné zavedenie týchto stratégií, teda ich použitie na robotovi v reálnom svete. Štvrtou a najdôležitejšou fázou je systémová identifikácia. V nej sa dáta nazbierané v tretej fáze využívajú na zlepšenie presnosti simulátoru pomocou knižnice pre automatickú diferenciáciu. Tieto knižnice umožňujú

túto optimalizáciu simulátoru spraviť časovo efektívne, vyžadujú ale väčšie množstvo pamäte RAM. Pre ešte presnejšie výsledky sa tieto štyri spomenuté fázy vykonajú niekoľko krát po sebe, s tým, že prvá iterácia berie dáta z reálneho sveta a každá ďalšia berie už optimalizované dáta predošlej iterácie. Tým je zaručené neustále spresňovanie simulátoru, a teda aj násobné zredukovanie rozdielu v realite.

Kapitola 3

Kinematika

V teórii klasickej mechaniky je kinematika náuka o mechanickom pohybe telies. Skúma zmeny vzájomných polôh telies v čase a priestore. Priestorom chápeme euklidovský priestor, ktorý je spojitý a nekonečný. Medzi hlavné predmety skúmania patrí rýchlosť, zrýchlenie, poloha a dráha pohybu. Pohyb je jedna z hlavných vecí, na ktoré sú vo svojej podstate robotické mechanizmy určené, preto je kinematika najzákladnejším hľadiskom návrhu, analýzy, riadenia a simulácie robotov. V robotike sa kinematika zaoberá pohybom telies v mechanizmoch bez ohľadu na sily, ktoré sú príčinou tohto pohybu. Na riešenie rôznych problémov kinematiky sa komunity zaoberajúce sa robotikou zameriavajú na efektívne uplatňovanie rôznych reprezentácií orientácie, polohy a ich derivácií vzhľadom na čas [18]. Medzi tieto základné problémy patrí dopredná a inverzná kinematika, ktoré využívame pri pohybe robotickým ramenom v tejto práci.

3.1 Robotické rameno

Robotické rameno je jedno z najbežnejších typov manipulátora, to znamená, že to je mechanické zariadenie používané na manipuláciu s objektami bez priameho fyzického kontaktu operátora. Je zložené spravidla z niekoľkých pevných častí, ktoré sú spojené kĺbmi. Takto za sebou sériovo zapojené časti a kĺby spolu vytvárajú kinematické zreťazenie. Toto zreťazenie je na jednom konci spojené buď s pevnou základňou alebo s telom robota. Na pracovanie s predmetmi využíva robotické rameno efektor, ktorý sa nachádza na druhom konci reťazca. Kĺby takéhoto robotického ramena charakterizujeme pomocou stupňov voľnosti (angl. degrees of freedom), ktoré súvisia so štruktúrou a mechanizmom spojenia. V trojdimenzionálnom priestore poznáme tri posuvné a tri rotačné stupne voľnosti. Pre každú os sa jeden posúva po tejto osi a jeden otáča okolo nej. Je zvykom každému kĺbu inkorporovať iba jeden stupeň voľnosti za účelom prostoty [14].

3.2 Dopredná kinematika

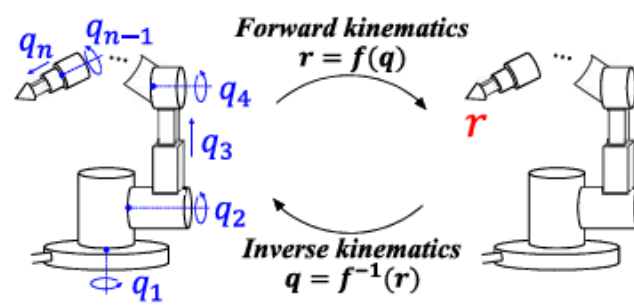
Problém doprednej kinematiky opisuje nájdenie polohy a orientácie koncového efektora vzhľadom na základňu na základe polohy všetkých kĺbov a hodnôt všetkých geometrických parametrov spojenia, teda geometriej štruktúry robotického mechanizmu. Počet stupňov voľnosti mechanizmu sa rovná počtu polôh kĺbov zreťazenia. Kĺby majú namontované snímače, ktorými vieme zistiť hodnotu stupňa voľnosti prislúchajúcu konkrétnemu spojeniu, no často potrebujeme vypočítať polohu kĺbov vzhľadom na pevný rám. Preto je pre prácu s robotom problém doprednej kinematiky kritický. V praxi sa tento problém rieši výpočtom transformácie medzi koncovým efektorm a fixovanou základňou. Teda polohu efektora získame kombináciou transformácií susedných článkov reťazca. Ide o prechod z uhlového priestoru kĺbov manipulátora do karteziánskeho priestoru polohy koncového efektora [17].

3.3 Inverzná kinematika

Problém inverznej kinematiky je opačný k problému doprednej kinematiky. Pre manipulátor so sériovým reťazcom je to nájdenie hodnôt kĺbov na základe polohy a orientácie koncového efektora vzhľadom na základňu a geometrickú štruktúru zreťazenia. Táto úloha sa rieši vo svete robotiky už niekoľko desaťročí, pretože narušenie od doprednej kinematiky, je tento problém nedeterministický. Pre jednu a tú istú polohu a orientáciu efektora spravidla existuje niekoľko kombinácií transformácií susedných častí spojenia, preto je výpočet časovo značne náročnejší. Ide o konverziu polohy a orientácie koncového efektora manipulátora do priestoru kĺbov [20].

Počas dlhodobého skúmania tohto problému sme spoznali nemalý počet riešení problému inverznej kinematiky. V základe sa delia na dva druhy. Analitický a numerický. Analitické riešenie využíva výraz v uzavretom tvare, ako vstup berie polohu koncového efektora a ako výstup poskytuje vektor polôh kĺbov. Toto riešenie ale v mnoho prípadoch nie je možné aplikovať, keďže tento problém má často nekonečne veľa riešení. V tom prípade je potrebné použiť numerický spôsob alebo iteratívny, ktorý pri hľadaní približného riešenia postupne optimalizuje výsledok každou iteráciou. Medzi známe numericé metódy patrí Jacobiho metóda alebo Levenberg-Marquardtova metóda [20].

V obrázku 3.1 vidíme vizualizáciu prechodu z priestoru kĺbov do priestoru efektora a naopak. Označenie r predstavuje polohu koncového efektora manipulátora v karteziánskej sústave, zatiaľ čo q predstavuje vektor polôh všetkých kĺbov v zreťazení od fixnej základne po koncový efektor. Funkcia f vyjadruje doprednú kinematiku, teda ako vstup dostáva vektor polôh kĺbov a na výstupe vracia polohu efektora. Označenie f^{-1} predstavuje opačné riešenie, teda inverznú kinematiku.



Obr. 3.1: Vizualizácia doprednej a inverznej kinematiky v práci od Yonezawa a kol. [19].

Kapitola 4

Humanoidný robot

Robot, ktorý simuluje ľudské telo sa nazýva humanoidný robot. Konštrukcia takéhoto robota má kopírovať anatómiu ľudského tela za účelom interakcie s ľudskými nástrojmi, nadobudnutia pocitu komfortnosti pri práci s ľuďmi alebo iné. Niektoré humanoidné roboty napodobňujú len časť tela, napríklad od pása nahor. Pri experimentoch zahŕňajúcich interakciu s ľuďmi sú obzvlášť dôležité črty tváre, ako ústa alebo oči. Spodná časť tela v tomto prípade väčšinou potrebná nie je [15].

Tvorba humanoidných robotov sa vyvíja už desaťročia. Prvé zmienky humanoidného staticky a dynamicky vyváženého robota sú zo šesťdesiatych rokov dvadsiateho storočia. Ide o robota s názvom WABOT, ktorého autor bol Ichiro Kato z Japonska. WABOT bol schopný komunikovať s človekom v japončine a merať vzdialenosti pomocou vonkajších receptorov. Pomocou dolných končatín vedel chodiť a pomocou horných končatín dokázal prenášať predmety za pomoci hmatových senzorov. Tento úspech súvisí s teóriou stability nulového bodu, ktorej predstaviteľ bol srbský inžinier Miodir Vukobratović v rovnakom období [6].

V dnešnej dobe sú humanoidné roboty využívané na široké spektrum úloh. V zdravotníctve asistujú napríklad pri operáciach, zmiernovaní bolesti, liečbe chorôb alebo fyzioterapii. V oblasti vzdelávania vedia priniesť priaznivé výsledky, vďaka schopnosti logicky uvažovať a analyzovať situácie, ktoré môžu podporiť ľudské učenie. Sociálne alebo asistenčné roboty majú využitie hlavne medzi staršími a zdravotne postihnutými ľuďmi. Ich úlohou je v tomto prípade robiť spoločníka v domácnosti, zabávať a hrať hry. Autistické deti preferujú spoločnosť humanoidných robotov, pretože sú predvídateľnejší ako ľudia [15].

4.1 Robot NICO

S účelom vykonať všetky ciele tejto práce sme pre naše experimenty vybrali humanoidného robota s názvom NICO. Toto pomenovanie je v angličtine skratka pre neuro-

inšpirovaného spoločníka (angl. Neuro-Inspired COmpanion). Tento stredne veľký vývojový robot bol skonštruovaný pre interakciu s ľuďmi formou spoločníka v prirodzenom prostredí. Dokáže sa učiť zo skúseností a inštrukcií a tiež sa používa na neurokognitívny výskum [13]. Softvér pre prácu s NICOm bol dizajnovaný skupinou Knowledge Technology v Univerzite Hamburgu a je plne dostupný pre verejnosť spolu so zdrojovými kódmi [10].

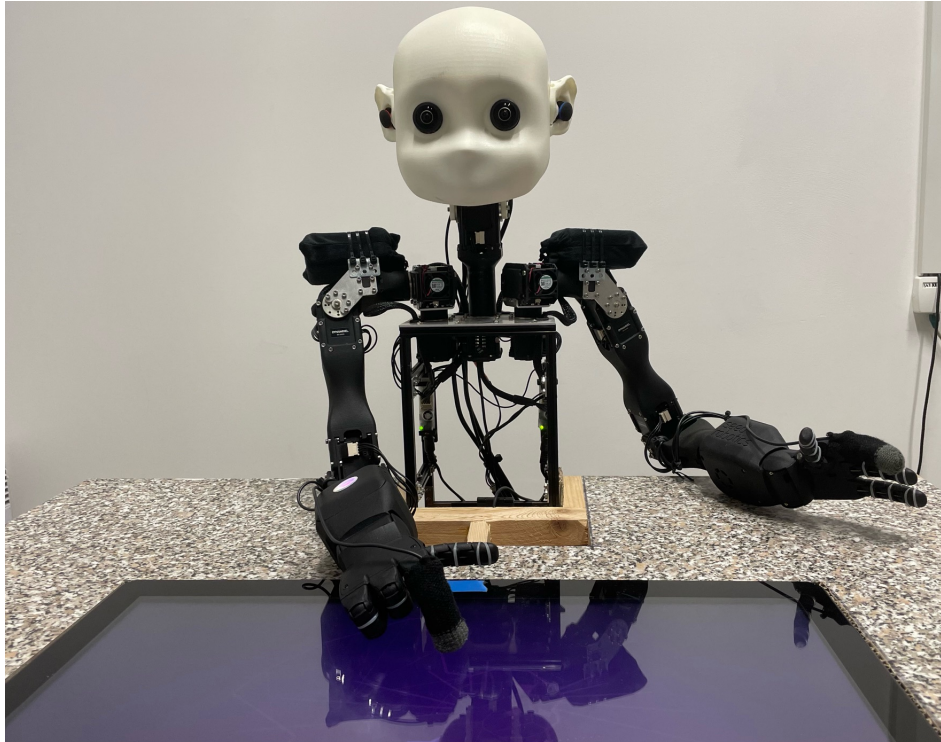
NICO sa pýši schopnosťou meniť výrazy tváre. Obsahuje LED matice na miestach úst a obočí, to mu umožňuje vyjadriť množstvo emócií. Naučené výrazy je možné doplniť novými vlastnými dizajnami. Ďalej ponúka videozáznam so snímkovou frekvenciou 60 snímkou za sekundu pri 4K rozlíšení. To vďaka dvom kvalitným kamerám na mieste očí. Stereofónne nahrávanie zvuku priamo do počítača umožňujú dva mikrofóny, jeden pre každé ucho. RH4D manipulátory dokážu pokročilo uchopovať a manipulovať s objektami. V neposlednom rade prichádza tento robot s 10 stupňami voľnosti v trupe a 22 stupňami voľnosti v rukách [9]. Robot NICO, ktorého máme k dispozícii, má pred sebou dotykovú obrazovku s rozlíšením 1920 na 1080 pixelov (4.1) a na ukazováku, ktorý slúži ako koncový efektor ramena, má omotaný alobal spájkovaný s drôtom, ktorý je omotávaný okolo jeho predlaktia. Na tomto alobale má natiahnutý odstrihnutý kus rukavice určenej na prácu s mobilným telefónom v zime. Drôt prináša elektrický náboj, zatiaľ čo alobal zabezpečuje dostatočnú jemnosť dotyku. Spolu s rukavicou to imituje dotyk ľudského prsta, ktorý dotyková obrazovka vie zachytiť.

Toto všetko a skutočnosť, že všetky moduly humanoidného robota NICO obsahujú svoje vlastné USB rozhranie, prispieva k jeho veľkej flexibilitate. Z týchto dôvodov je to najvhodnejší kandidát pre náš experiment kvantifikácie rozdielu v realite.

4.2 Simulátor myGym

Pri výbere vhodného simulátoru pre náš experiment sme hľadali simulátor s podporou pre robota NICO a s jednoduchým prechodom zo simulácie do reality. Naša prvá voľba bol simulátor Unity. V ňom sme mali prístup ku pomerne kvalitne simulovanej verzii robota NICO. No zdrojové kódy k tomuto simulátoru neboli voľne dostupné, takže flexibilita úprav bola nižšia. Taktiež je dôležitý fakt, že neponúkal možnosť prechodu z virtuálnej reality, teda simulácie, do fyzickej reality, čo je nevyhnutné pre naše experimenty.

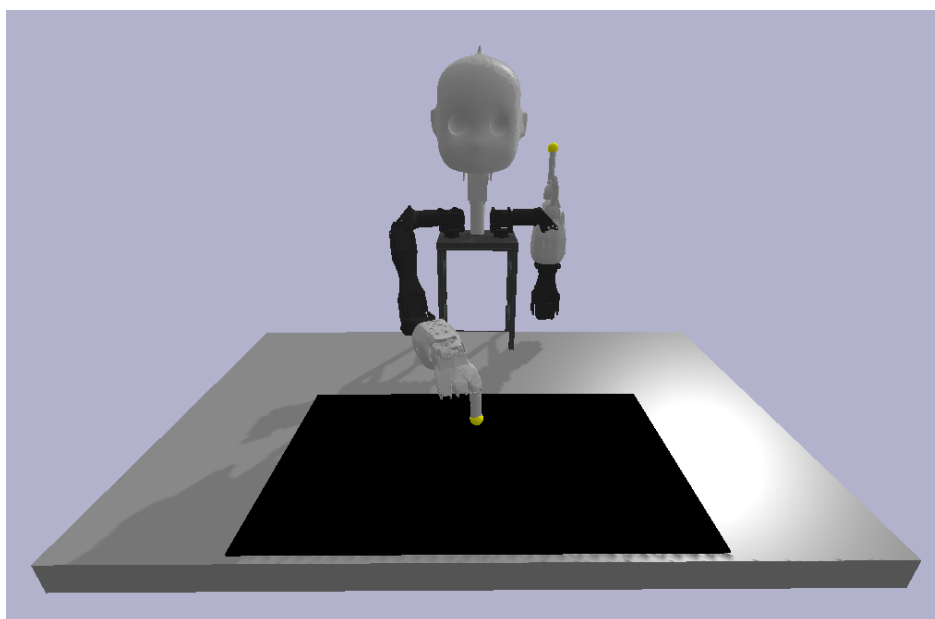
Preto sme pre našu prácu vybrali simulátor myGym. Tento simulátor bol síce vo vývoji, no narozdiel od simulátoru Unity, ponúkal plne dostupné zdrojové kódy k softvéru [7]. To znamená, že bol mimoriadne flexibilný v oblasti variácie experimentov a parametrov. Taktiež v ňom bolo možné veľmi jednoducho prechádzať zo simulácie do fyzickej reality [8] a bol v ňom už pripravený simulovaný humanoidný robot NICO



Obr. 4.1: Reálny robot NICO vo Fakulte Matematiky, Fyziky a Informatiky v Bratislave s dotykovou obrazovkou a nástavcom na ukazovák (koncovom efektore), ktorý umožňuje interakciu s tabletom.

(Obr. 4.2). Tým spĺňal všetky požiadavky pre vhodný simulátor na vykonanie každého z cieľov tejto práce.

Po krátkom zoznamovaní sa so softvérom myGym, sme vyskúšali jeho základné funkcionality. Išlo hlavne o rôzne pohyby s ramenom robota NICO, ktoré sa vzťahuje k našej práci. So simulátorom sa pracovalo príjemne, hlavne vďaka tomu, že sme mali prístup ku všetkým zdrojovým kódom, ktorých nebolo veľké množstvo, keďže bol simulátor v tomto čase stále vo vývoji. Aj pre túto jednoduchosť softvéru fungovala simulácia v tomto prostredí výborne a bolo možné túto simuláciu ľubovoľne upraviť podľa našich predstáv.



Obr. 4.2: Robot NICO v simulácii simulátoru myGym, už s predpripraveným prostredím.

Kapitola 5

Návrh úlohy

Druhým cieľom našej práce bol návrh jednoduchej motorickej úlohy pre rameno robota NICO tak, aby bolo možné porovnať simuláciu s fyzickou realitou.

5.1 Skúmané hodnoty

Čo teda konkrétne vieme my skúmať pri prechode medzi realitami? Základným parametrom, ktorý môžeme merať je poloha objektov v čase. Polohou chápeme karteziánske súradnice v dvojrozmernom alebo trojrozmernom priestore. V simulácii máme prístup k polohám všetkých objektov v scéne. Vo fyzickej realite sme použili na zistenie polohy v dvojrozmernom priestore dotykovú obrazovku, ktorá sa nachádza pred robotom NICO. Druhou veličinou, ktorú vieme zmerať vo fyzickej realite, je stupeň voľnosti. Motory nášho humanoidného robota NICO ponúkajú túto informáciu, teda vieme si vypýtať hodnotu akéhokoľvek stupňa voľnosti v tele robota v akomkoľvek čase. V simulátore túto informáciu poznáme samozrejme tiež. Poslednými parametrami, ktoré môžeme porovnávať sú čas odozvy motorov a dĺžka trvania pohybu. Čas odozvy je rozdiel medzi časom, kedy sme poslali motorom príkaz na vykonanie pohybu a časom, kedy sa začali hýbať. Dĺžka trvania pohybu je uplynulý čas medzi začiatkom pohybu a dorazením všetkých motorov do požadovanej polohy. Bohužiaľ, čas nám poskytuje kód, ktorý ovláda robota, takže sa nemôžeme spoliehať na úplnú presnosť týchto hodnôt. Teda v skutočnosti musíme rátať s tým, že v kóde môžu prebiehať nejake výpočty a porovnania, ktoré môžu ovplyvniť túto informáciu. No bavíme sa o minimálnych rozdieloch.

5.2 Pohyb robota

Ako úlohu, ktorú má robot NICO vykonávať pri meraní sme vybrali jednoduché natiahnutie ramena za cieľom (angl. reach). Ciele boli umiestnené na dotykovej obrazovke,

aby sme mohli prečítať polohu koncového efektora vo fyzickej realite po vykonaní pohybu. Tento pohyb nám poskytoval všetky potrebné informácie pre porovnanie realít. Pre ovládanie pohybu ramena sme vybrali inverznú kinematiku, ktorej výpočet ponúkal simulátor myGym. Inverzná kinematika rieši presne takýto typ úlohy, kedy poznáme polohu koncového efektora a potrebujeme zistiť polohy stupňov voľnosti v manipulátore. Teda poznáme polohu cieľa a potrebujeme zistiť, aké hodnoty máme poslať do motorov, aby koncový efektor skončil po vykonaní pohybu v polohe tohto cieľa. Preto je tento prístup vhodný pre našu úlohu. Prechod zo simulácie do fyzickej reality prebiehal v rovnaký čas ako simulácia v myGym, takže sme mohli zároveň sledovať pohyby virtuálneho robota aj fyzického. Stačí zapnúť funkciu, ktorá posiela príkazy reálnemu robotovi.

5.3 Experimenty

Pre väčší objem získaných dát a možné porovnania výsledkov sme vykonali viac typov experimentov. Všetky experimenty používali iba natiiahnutie ruky za dotykovou obrazovkou, no menili sme polohy cieľov a dĺžku času vykonávaného pohybu. Prvý experiment pozostával z troch cieľov rovnomerne rozmiestnených po ploche obrazovky. Jeden vľavo, jeden vpravo a jeden v strede. Do každého z troch bodov sme nechali robota trafiť sa tridsať krát. Účel tohto experimentu bolo odmeranie konzistentnosti riešenia, teda či a poprípade ako veľmi sa navzájom líšia riešenia s rovnakým vstupom. Druhý experiment obsahoval postupnosti cieľov v rade, buď vertikálne, horizontálne alebo diagonálne línie. Účel druhého experimentu bolo sledovanie schopnosti robota nasledovať rovnú čiaru. Vizualizáciou zásahou sme túto štatistiku mohli dobre sledovať. Ciele v treťom experimente boli pravidelne rozmiestnené po celej testovacej ploche obrazovky. Tu sme podobne ako pri druhom experimente sledovali kam sa robot na obrazovke triafa, no tento krát sme mohli vyčítať z vizualizácie, či vzdialenosť a smer od základne robotického ramena má vplyv na vychýlenie od požadovaného cieľa a či uvidíme nejakú pravidelnosť. Pre posledný a hlavný experiment sme si rozdelili testovaciu plochu obrazovky na štyri kvadranty. Potom sme robotovi dávali tridsať náhodných cieľov buď po celej ploche alebo iba v jednom zo štyroch kvadrantov. Týchto päť iterácií sme vykonali raz s dĺžkou trvania pohybu jedna sekunda a druhý krát s dĺžkou tri sekundy. Z týchto desiatich menších experimentov sme potom mohli vytvoriť rôzne štatistiky. Odchýlky v dvojrozmernom priestore, odchýlky stupňov voľnosti, čas odozvy a čas vykonania pohybu. Tieto všetky namerané hodnoty sme mohli v závislosti od vzdialenosti a smeru od základne ramena a dĺžky času vykonania pohybu na vstupe. Pri tomto experimente nás už nezaujímala vizualizácia zásahov ako v predošlých testoch, iba porovnanie štatistík, teda kvantifikácia prípadného rozdielu v realite.

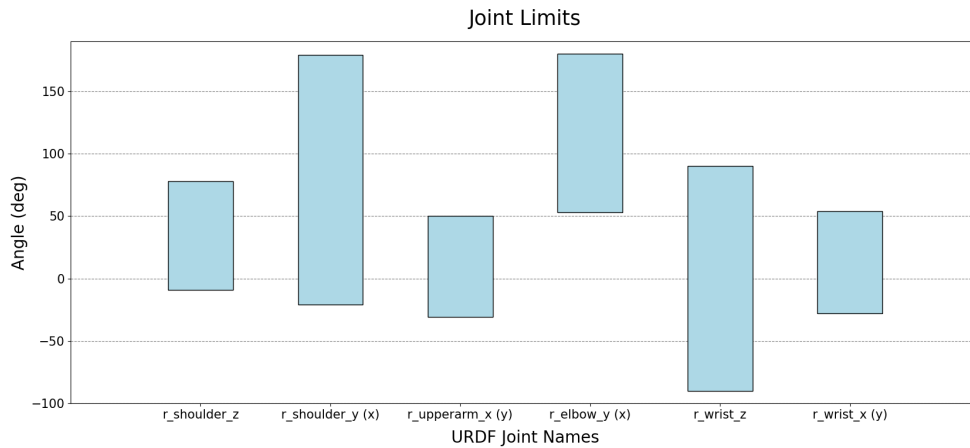
Kapitola 6

Implementácia

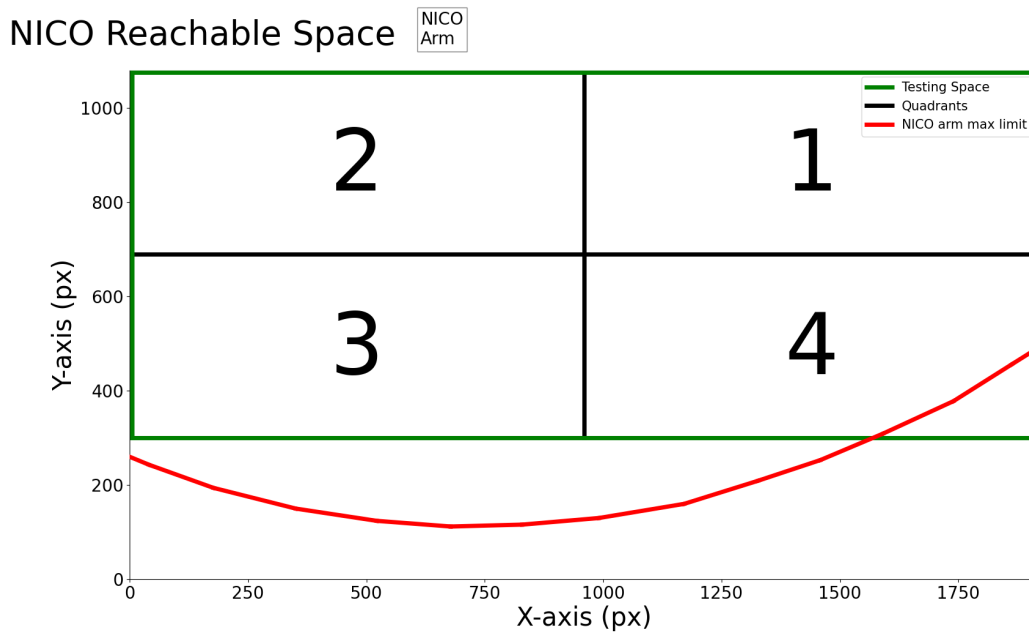
6.1 Obmedzenia

Pri práci s akýmkoľvek robotom musíme dbať na určité obmedzenia a limity. Jedným z dôležitých obmedzení vo svete robotiky sú uhlové limity alebo rozsahy kĺbov. Náš robot NICO má dvadsaťdva stupňov voľnosti, no keďže na vykonanie všetkých experimentov nám stačí jedno rameno, tak sme sa pozreli len na stupne voľnosti pravého ramena. Robotické rameno nášho humanoidného robota NICO má desať kĺbov, teda desať stupňov voľnosti. No pre náš účel klikať ukazovák na dotykovú obrazovku sme zafixovali kĺby prstov v takej polohe, aby prišiel koncový efektor ramena do kontaktu s obrazovkou vždy ako prvý a smerom zhora nadol. Teda potrebovali sme udržať orientáciu efektora v smere kolmom na obrazovku. To sa podarilo zafixovaním posledných štyroch stupňov voľnosti manipulátora v pozíciách, ktoré simulujú ukazovanie prstom spolu s úpravou počiatočnej pozície ramena tak, aby už na začiatku efektor smeroval na obrazovku. Takže nám ostalo šesť stupňov voľnosti, ktoré môžeme meniť pri pohybe s ramenom. Iné kĺby sme nepoužívali. URDF názvy týchto šiestich stupňov voľnosti spolu s rozsahmi ich uhlov vidíme v obrázku 6.1.

Druhým obmedzením, s ktorým musíme rátať v našom testovacom prostredí je dosah robotického ramena. Robot NICO s pravým ramenom a so zafixovanými kĺbmi prstov nedočiahne na celú plochu dotykovej obrazovky. Preto sme skrátili testovaciu plochu na ose y tak, aby sme minimalizovali šancu, že dostane na vstupe cieľ, na ktorý nedokáže dosiahnuť. Pre možnosť tvorenia lepších štatistík sme rozdelili túto zmenšenú testovaciu plochu na štyri kvadranty. Vizualizácia kvadrantov a dosiahnuteľnej plochy robota NICO je v obrázku 6.2.



Obr. 6.1: Uhlový rozsah šiestich stupňov voľnosti pravého ramena, ktoré NICO používa pri experimentoch. Os x robota NICO je os y na dotykovej obrazovke a naopak. Takže keď má kĺb v zátvorke x , tak sa v našom systéme otáča okolo osi x a nie y . Pre robota je to os y .



Obr. 6.2: Vizualizácia dosiahnuteľnej plochy pravého robotického ramena, testovacej plochy a kvadrantov.

6.2 Prechody medzi sústavami

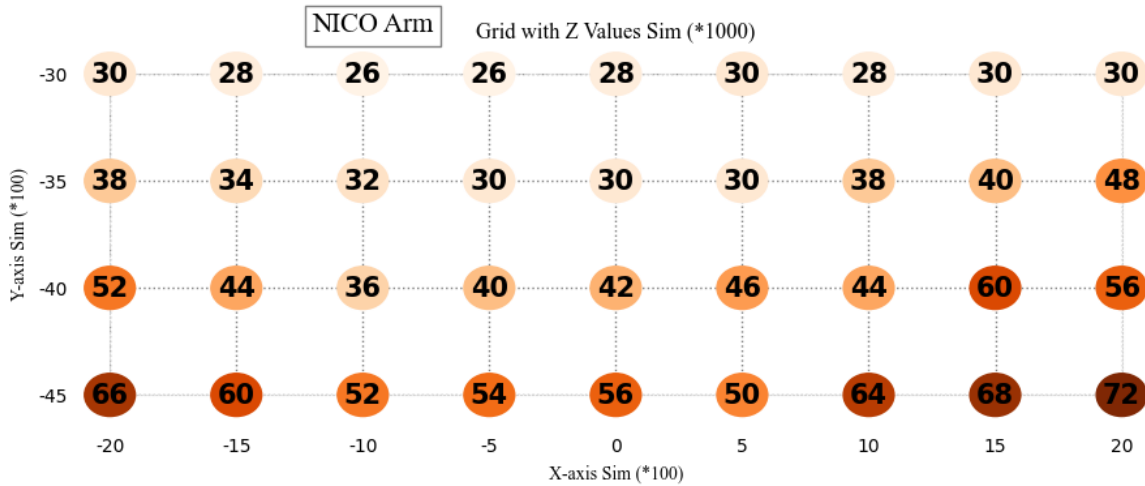
Pre správnu interpretáciu našich výsledkov sme si museli dať pozor na rozdiely sústav v karteziánskom priestore. Z dotykovej obrazovky dostávame dvojicu pixelov, reprezentujúcu polohy na osiach x a y , zatiaľ čo simulácia má svoju vlastnú reprezentáciu týchto hodnôt. Dokonca os x v simulácii a pre reálneho robota je os y na obrazovke. A os x na obrazovke je os y v simulácii. Potrebovali sme preto spraviť metódy, ktoré konvertujú súradnice z obrazovky do priestoru v simulátore a opačne. Pre lepšiu vizualizáciu sme sa pri kreslení grafov rozhodli používať reprezentáciu dotykovej obrazovky. Taktiež by sme radi vyjadrili vzdialenosti na dotykovej obrazovke v centimetroch a nie len v pixeloch. O to sa stará metóda na konverziu z pixelov na centimetre.

6.3 Hľadanie hodnoty na osi z

S dbaním na obmedzenia robota NICO a s funkčnými metódami na konverziu dát sme mohli prvýkrát vyskúšať ovládať reálneho robota NICO vo fyzickej realite. Pohyby simulovaného robota napodobňoval fyzický NICO pekne, no hneď sme si mohli všimnúť nepresnosti medzi simuláciou a realitou. Keď sme chceli aby prišiel s koncovým efektorom na úroveň dotykovej obrazovky a dotkol sa nej, tak buď zastavil pohyb nad obrazovkou alebo zatlačil príliš silno na obrazovku. Takto nebolo možné vykonať žiadne experimenty, keďže potrebujeme aby sa NICO vždy alebo aspoň skoro vždy dotkol obrazovky, nech môžeme zaznamenať polohu zásahu a porovnať s cieľom. Pre pokračovanie v práci bolo nevyhnutné vyriešiť tento problém.

6.3.1 Meranie rozdielu

Vyzeralo to tak, že výška, v ktorej skončí koncový efektor robotického ramena sa mení v závislosti od hodnôt x a y cieľového bodu. Napadlo nám, že môžeme zapísať tieto výšky pre pravidelné rozmiestnenie bodov na testovacej ploche obrazovky a tým zistíme, ako sa mení výška s meniacou sa polohou cieľového bodu. Odmerať ako vysoko je koncový efektor od obrazovky by sme vedeli, no v opačných prípadoch, kedy NICO vidí cieľ pod skutočnou úrovňou obrazovky a začne tlačiť do nej, nevieme zmerať ako nízko ten bod vidí. Preto sme otočili tento problém. Nepýtali sme sa otázku, ako vysoko je efektor reálneho robota, keď efektor simulovaného robota je presne na obrazovke, ale pýtali sme sa, ako vysoko je efektor simulovaného robota, keď sa fyzické rameno jemne dotkne obrazovky. Takže sme si pripravili maticu súradníc x a y tridsiatichšiestich bodov v testovacom priestore a pre každý bod sme postupne púšťali koncový efektor fyzického robota NICO smerom nadol, až kým nenastal dotyk s obrazovkou. Vtedy sme si zapísali z -ovú hodnotu koncového efektora v simulácii a priradili ju ku bodu v matici. Túto



Obr. 6.3: Namerané hodnoty na osi z v simulácii v časoch, kedy sa koncový efektor fyzického robota NICO jemne dotýkal obrazovky v príslušnom bode x , y . NICO Arm predstavuje polohu základne pravého ramena fyzického robota.

maticu budeme ďalej nazývať kalibračnou maticou. Výsledok merania môžeme vidieť v obrázku 6.3. Na prvý pohľad si všimneme, že namerané hodnoty nie sú dokonalo pravidelné, no vieme tam nájsť určité trendy. S rastúcou vzdialenosťou fyzického efektora od základne fyzického ramena rastie aj hodnota na z -ovej osi koncového efektora v simulácii. Takže by sme mohli napísať funkciu ktorá aproximuje z -ovú hodnotu čisto iba na základe vzdialenosti efektora od základne, ale bolo by to veľmi nepresné. Lepším riešením nám prišlo aproximovať hľadanú hodnotu za použitia okolitých bodov. Teda s použitím nameraných hodnôt. Model neurónovej siete by vedel riešiť takýto problém, no nakoniec sme sa rozhodli ísť analytickou cestou.

6.3.2 Riešenie

Pre čo najpresnejšiu aproximáciu z -ovej hodnoty, používa naše riešenie všetky štyri okolité body, ktoré tvoria rohy najmenšieho štvorca okolo hľadaného bodu. Na získanie z -ovej hodnoty touto aproximáciou potrebujeme mať k dispozícii kalibračnú maticu a hodnoty x a y bodu, ktorého z hľadáme. Najprv nájdeme medzi ktorými bodmi sa cieľový bod nachádza, respektíve v ktorom štvorci v kalibračnej matici a potom na základe z -ových hodnôt rohov tohto štvorca aproximujeme hľadanú z -ovú hodnotu. Nezávislé premenné a funkcie, ktoré využívame pri výpočte sú:

$G(x, y)$... funkcia, ktorá na vstupe berie x -ovú a y -ovú súradnicu, a ak sa v kalibračnej matici nachádza bod s týmito súradnicami, tak vráti hodnotu z tohto bodu

x ... x -ová súradnicová hodnota hľadaného bodu

y ... y -ová súradnicová hodnota hľadaného bodu

Závislé premenné, ktoré využívame pri výpočte sú:

d_v ... vertikálna vzdialenosť dvoch susedných bodov s rovnakými y -ovými hodnotami v kalibračnej matici

d_h ... vertikálna vzdialenosť dvoch susedných bodov s rovnakými x -ovými hodnotami v kalibračnej matici

x_c ... x -ová hodnota ľavého dolného rohu štvorca, v ktorom sa nachádza hľadaný bod

y_c ... y -ová hodnota ľavého dolného rohu štvorca, v ktorom sa nachádza hľadaný bod

z_l ... z -ová hodnota bodu, ktorý má rovnakú hodnotu y ako hľadaný bod a leží medzi ľavým dolným a ľavým horným rohom štvorca, v ktorom sa nachádza hľadaný bod

z_r ... z -ová hodnota bodu, ktorý má rovnakú hodnotu y ako hľadaný bod a leží medzi pravým dolným a pravým horným rohom štvorca, v ktorom sa nachádza hľadaný bod

z ... hľadaná hodnota

Výpočet:

$$\begin{aligned} x_c &= \lfloor \frac{x}{d_h} \rfloor \cdot d_h \\ y_c &= \lfloor \frac{y}{d_v} \rfloor \cdot d_v \end{aligned} \quad (6.1)$$

Takto sme našli x -ovú a y -ovú hodnotu ľavého dolného rohu štvorca, v ktorom sa nachádza hľadaný bod.

$$z_l = G(x_c, y_c) + \frac{y - y_c}{d_v} \cdot (G(x_c, y_c + d_v) - G(x_c, y_c)) \quad (6.2)$$

V tomto kroku sme zobrali z -ovú hodnotu ľavého dolného bodu štvorca a pripočítali k nej rozdiel z -ovej hodnoty ľavého dolného rohu a ľavého horného rohu štvorca vynásobený pomerom y -ových vzdialeností medzi ľavým dolným rohom a hľadaným bodom

a medzi ľavým dolným rohom a ľavým horným rohom. Tým sme získali aproximáciu z -ovej hodnoty bodu naľavo od hľadaného bodu.

$$z_r = G(x_c + d_h, y_c) + \frac{y - y_c}{d_v} \cdot (G(x_c + d_h, y_c + d_v) - G(x_c + d_h, y_c)) \quad (6.3)$$

Obdobne ako v predošlom kroku, aj v tomto sme aproximovali z -ovú hodnotu, ale tento krát bodu napravo od hľadaného bodu.

$$z = z_l + \frac{x - x_c}{d_h} \cdot (z_r - z_l) \quad (6.4)$$

V poslednom kroku sme vykonali rovnakú aproximáciu ako v predošlých dvoch krokoch, no teraz na x -ovej osi. Medzi bodmi, ktorých z -ové hodnoty sme už aproximovali.

Týmto spôsobom sme vedeli získať hodnotu osi z na ľubovoľnom mieste v testovacej ploche obrazovky. Bod mimo kalibračnej matice sme takto aproximovať nevedeli, ale to v tejto práci nepotrebujeme. Vďaka tejto metóde aproximácie sme mohli vykonať všetky pripravené experimenty na reálnom robotovi NICO.

6.4 Iné implementácie

Na spracovanie dotykov obrazovky pri robotovi NICO bol pripravený skript TouchAgent.py [2] využívajúci knižnicu Pygame [1]. No tento softvér je spravený pre operačný systém windows a simulátor myGym beží v operačnom systéme Linux. Takže bolo potrebné prepísať časti kódu, ktoré boli špecifické pre Windows na kód, ktorý podporuje Linux.

Pre vykonanie každého z experimentov sme museli upraviť zdrojový kód simulátora myGym a na spracovanie všetkých typov dát sme použili knižnicu matplotlib [12].

Kapitola 7

Výsledky

V tejto práci máme dva druhy vizualizácie výsledkov. Jeden obsahuje ciele so zásahmi a poprípadne malé množstvo štatistík v rohu. Tento typ vizualizácie je vhodný pre presné pozorovania chýb v zásahoch v závislosti od vzdialenosti a smeru od základne robotického ramena, ktorá je v týchto grafoch znázornená štvorčekom s popisom NICO Arm. Druhý typ vizualizácie výsledkov sú boxploty. Tie nám ponúkajú detailnejšiu štatistiku rozdielov v realitách.

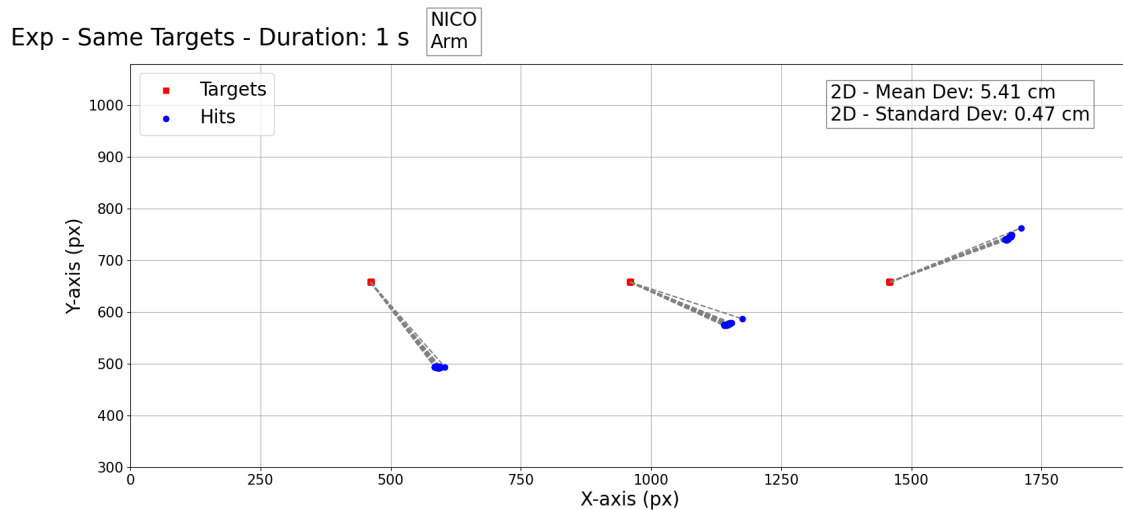
7.1 Rovnaký bod

V prvom experimente sme si vytvorili tri ciele. Jeden na ľavej strane testovacej plochy, jeden na pravej a posledný v strede. Do každého z nich sme nechali robota trafiť sa tridsať krát pohybom, ktorý mal trvať jednu sekundu a zapísali si polohy zásahov. Chceli sme tu sledovať konzistenciu riešenia a prechodu medzi realitami. Vizualizácia výsledkov je v obrázku 7.1.

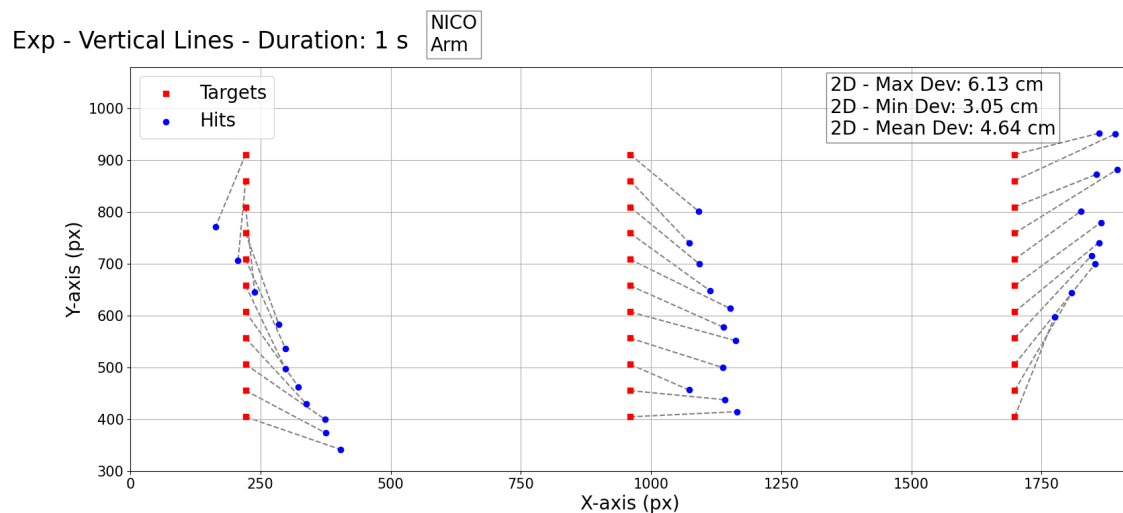
Hneď z prvého výsledku sme zistili mnoho zaujímavých informácií. Priemerná vzdialenosť cieľa od zásahu bola 5.41 centimetrov, čo je pomerne veľké číslo na tak jednoduchú úlohu. No cieľom tohto experimentu bolo skontrolovať konzistenciu rovnakého riešenia, teda ako sa líšia zásahy, ktoré dostanú rovnaký cieľ. A to podľa vizualizácie dopadlo dobre. Môžeme vidieť, že štandardná odchýlka v tomto experimente je len 0.47 centimetrov. Poslednú vec, ktorú si môžeme všimnúť je, že zásahy divergujú od cieľa vždy v nejakom smere. Začínáme mať podozrenie, že to súvisí s polohou základne robotického ramena.

7.2 Ciele v línii

Druhý experiment sa skladal z viacerých menších experimentov. V každom z nich boli ciele uložené za sebou v čiarach, robot NICO išiel do každého cieľa raz a pohyb ramena



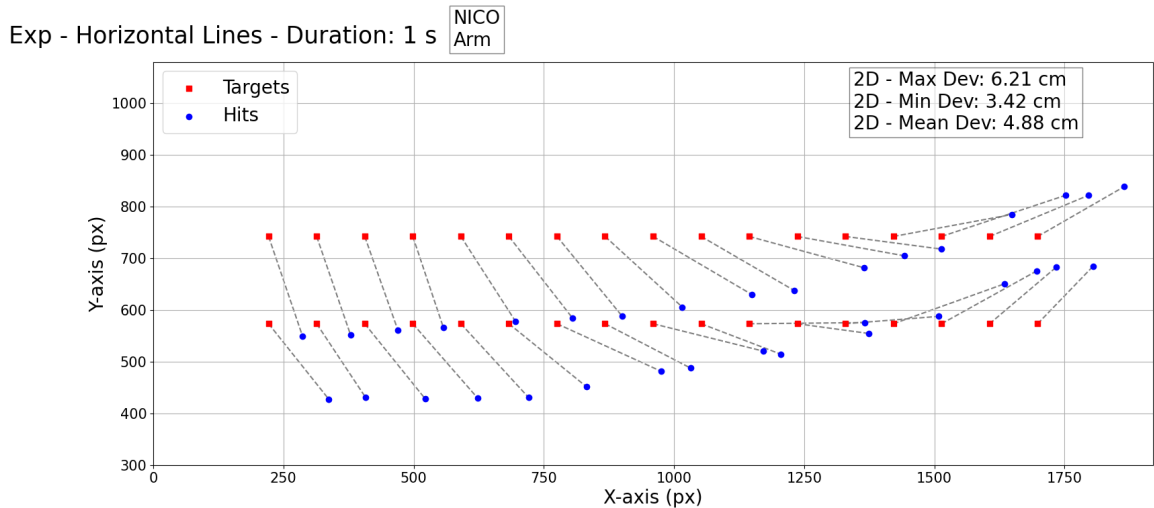
Obr. 7.1: Výsledky prvého experimentu - tri rovnaké ciele - čas pohybu: 1 sekunda



Obr. 7.2: Výsledky druhého experimentu - časť s vertikálnymi líniami - čas pohyb: 1 sekunda

mal trvať vždy jednu sekundu. Snažili sme sa zistiť do akej miery vie NICO nasledovať priamku, respektíve body na priamke.

Prvá časť tohto experimentu pozostávala z cieľov v troch vertikálnych líniah. V ľavej strane testovacej plochy, v pravej a v strednej. Vizualizácia tejto časti je v obrázku 7.2. Môžeme si tam všimnúť, že nasledovať body v priamej čiare sa robotovi nepodarilo. Na ľavej strane nasledoval diagonálnu líniu namiesto vertikálnej. V pravej a strednej časti plochy to bolo lepšie, no odchýlka zásahov od cieľov je stále veľká, v tomto prípade je priemer 4.64 centimetrov. Najväčšia nameraná vzdialenosť medzi cieľom a zásahom bolo 6.13 centimetrov a najmenšia 3.05 centimetrov. Rovnako ako v prvom experimente, vidíme, že v závislosti od polohy cieľa sa mení smer vychýlenia zásahu od cieľového bodu.

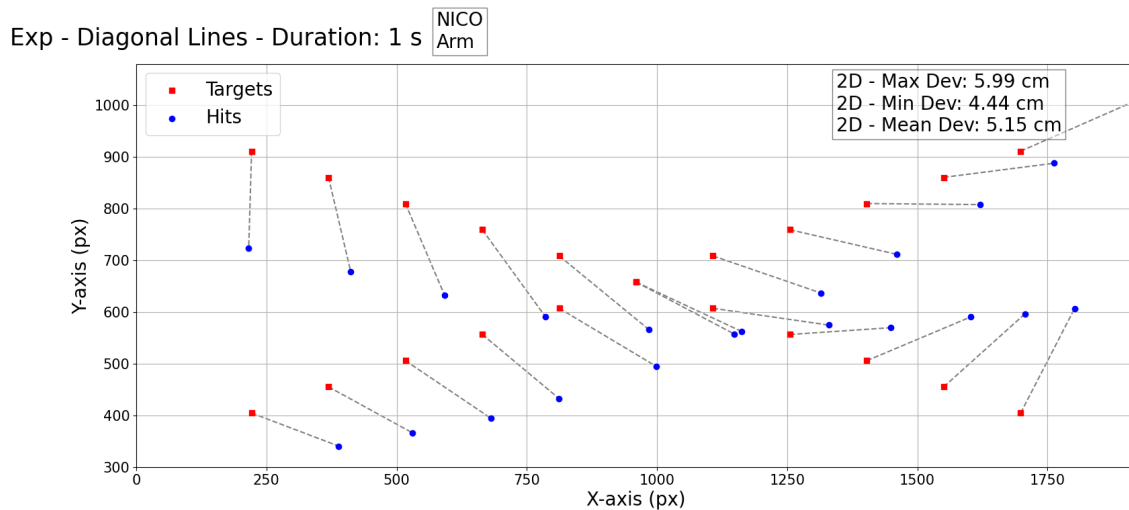


Obr. 7.3: Výsledky druhého experimentu - časť s horizontálnymi líniami - čas pohybu: 1 sekunda

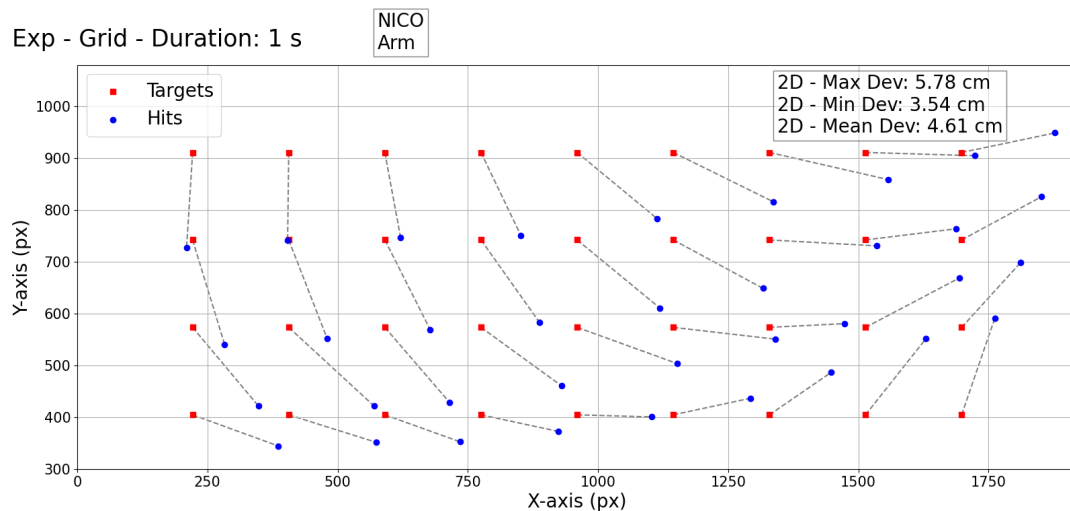
V druhej časti druhého experimentu sme sa pokúšali nasledovať dve horizontálne čiary. Jednu v hornej časti plochy, jednu v dolnej. Obrázok 7.3 ponúka vizualizáciu výsledkov. Veľkosť odchýliek zásahov od cieľových bodov ostáva v podobných číslach ako v predošliých experimentoch. Zaujímavé na tomto experimente je, že v ľavej polo-
vici testovacej plochy sa darilo robotovi ísť v priamom smere, samozrejme zásahy boli vychýlené od cieľov ale zdá sa, že viacej cieľov po sebe sa vychýľovalo veľmi podobným spôsobom. Prechodom do pravej časti plochy sa divergencie začali meniť, akokeby smerom od základne robotického ramena.

Tretia časť tohto experimentu obsahovala dve diagonálne línie, teda ciele rozmiestnené po diagonálnych líniiach. Vizualizáciu výsledkov vidíme v obrázku 7.4. Čo sa týka veľkostí odchýliek v tejto časti experimentu, môžeme si všimnúť menší rozsah hodnôt. Maximálna odchýlka je 5.99 centimetrov a minimálna je 4.44 centimetrov. Ich rozdiel je len 1.55 centimetra, zatiaľ čo v predošliých testovaniach sa tento rozdiel blížil k trom centimetrom. No priemer sa stále drží pri piatich centimetroch, konkrétne 5.15. Ďalej si môžeme z výsledkov tohto testu všimnúť, že sa tu robotovi darilo najlepšie nasledovať líniu. Pravdepodobne to bude spôsobené tým, že jedna diagonála je kolmá k polohe základni robotického ramena. V opačnej diagonále vidíme priamu linku zásahov len na pravej strane testovacej plochy. Tu môžeme uvažovať, že je to spôsobené čiastočným smerovaním línie cieľov ku základni.

V týchto troch častiach druhého experimentu sme si mohli jasne všimnúť trend divergencie bodu zásahu od cieľového bodu v závislosti od polohy cieľa.



Obr. 7.4: Výsledky druhého experimentu - časť s diagonálnymi líniami - čas pohybu: 1 sekunda

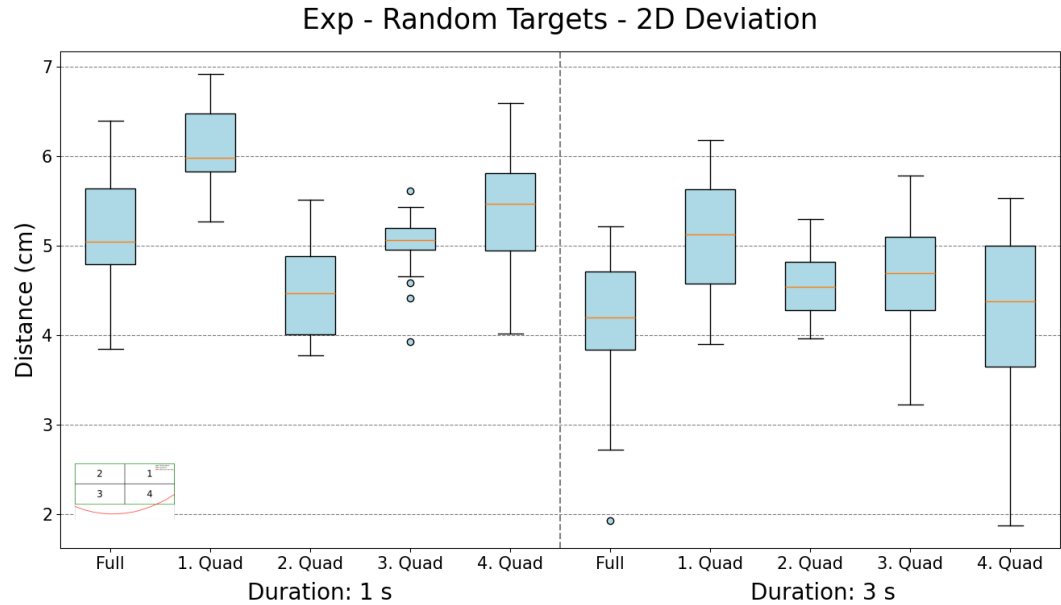


Obr. 7.5: Výsledky tretieho experimentu - celé pokrytie - čas pohybu: 1 sekunda

7.3 Pokrytie celej plochy

V treťom experimente sme dúfali, že charakterizujeme vychýlenia zásahov od cieľov tým, že pokryjeme celú testovaciu plochu. To sme spravili rovnomerným rozdelením tridsiatichšiestich bodov na ploche. Do každého bodu sa robot triafal jedenkrát a každý pohyb mal trvať rovnako ako v predošlých experimentoch jednu sekundu.

Výsledky tohto experimentu sú vizualizované v obrázku 7.5. Tento experiment dosiahol zatiaľ najmenšiu priemernú odchýlku, a to 4.61 centimetrov, no stále sa to drží hranice piatich centimetrov. Čo je ale zaujímavé na výsledkoch tohto experimentu je, že presne vidíme trend vychýlení polôh zásahov od polôh cieľov. Vyzerá to tak, že zásahy rotujú v smere opačnom k smeru hodinových ručičiek buď s rastúcou vzdialenosťou



Obr. 7.6: Výsledky štvrtého experimentu - náhodné ciele - porovnanie odchýliek medzi cieľovými bodmi a zásahmi

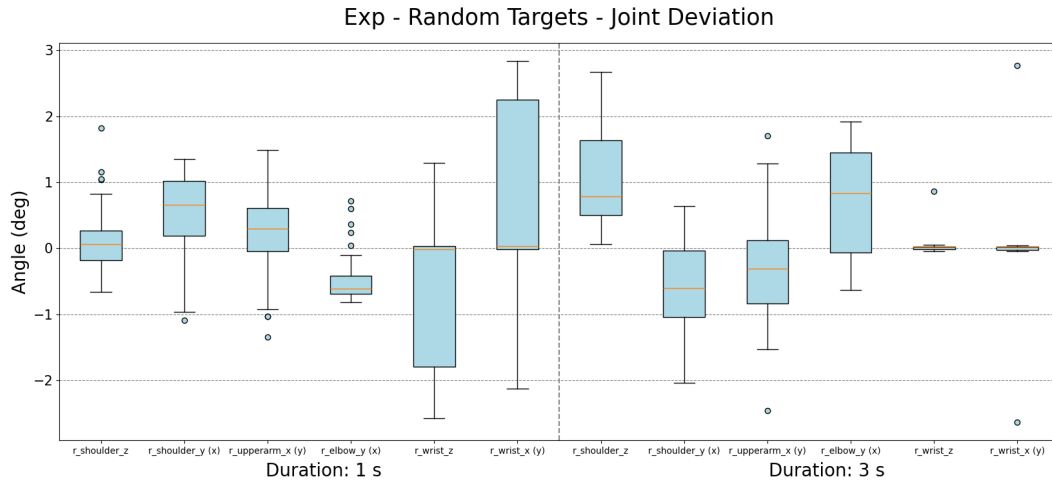
cieľového bodu od základne robotického ramena, alebo tento jav silnie s klesajúcou y -ovou hodnotou a rastúcou x -ovou hodnotou cieľového bodu súčasne.

Podľa výsledkov tohto experimentu by sme vedeli odhadnúť polohu zásahu koncového efektora robota NICO na základe polohy príslušného cieľového bodu podobne, ako sme to spravili pre výpočet z -ovej hodnoty v kalibračnej matici. Pravdepodobne by sme týmto spôsobom vedeli značne spresniť interakciu robota s dotykovou obrazovkou a tým minimalizovať rozdiel v realite.

7.4 Náhodné body

Štvrtý experiment je finálny a najrozsiahlejší experiment v našej práci. Pozostáva z iterácií generovania náhodných cieľov buď v celej ploche určenej na testovanie, alebo v rozmedzí jedného zo štyroch kvadrantov, na ktoré sme plochu rozdelili v podkapitole 6.1. A v tomto prípade sme týchto päť testovaní zopakovali druhýkrát so zmenou trvania pohybu z jednej sekundy na tri sekundy. Pri týchto testovaniach nás už nezaujímá vizualizácia zásahov v ploche, tému divergencie sme uzavreli v treťom experimente. Pri tomto experimente sme sa sústredili na štatistiky odchýliek zásahu od cieľa v dvojrozmernom priestore, uhlové odchýlky jednotlivých kĺbov manipulátora, čas odozvy robota NICO a čas vykonania pohybu. Tieto parametre sme porovnali medzi jednotlivými spomenutými variáciami generovania náhodných cieľov a času trvania pohybu.

Prvou a hlavnou štatistikou je odchýlka polôh zásahov od polôh cieľov. Graf s

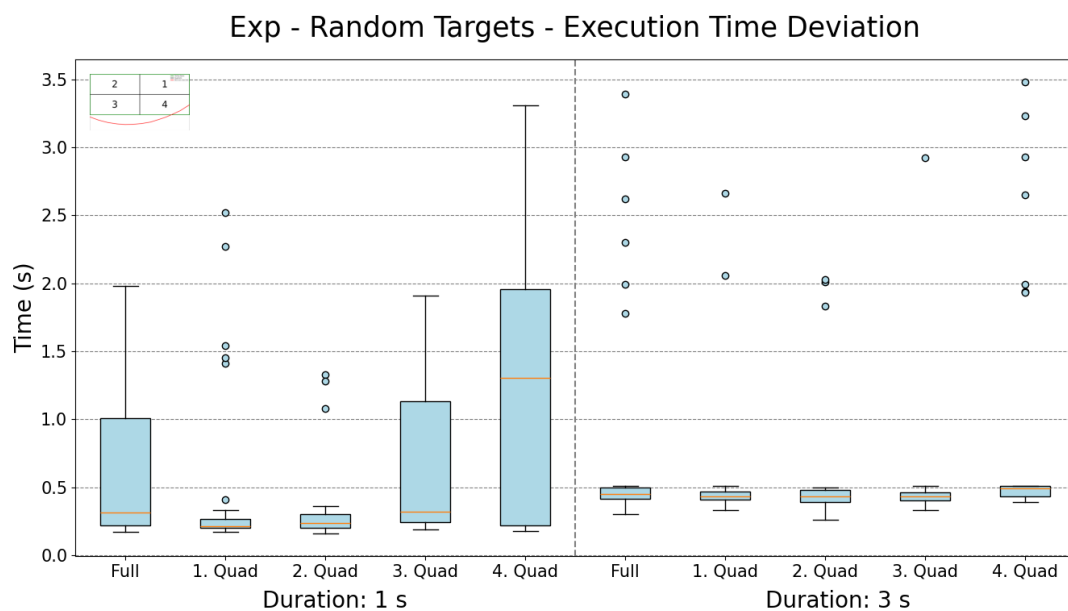


Obr. 7.7: Výsledky štvrtého experimentu - náhodné ciele - porovnanie odchýliek medzi jednotlivými kĺbmi robotického ramena

touto štatistikou je v obrázku 7.6. Vidíme, že medián sa drží podobne ako priemer v predošlých experimentoch okolo hodnoty 5 centimetrov. Z grafu vyčnieva veľkosť odchýliek v prvom kvadrante, je najväčšia aj v jednosekundovom pohybe, aj v trojsekundovom. Môže to byť spôsobené tým že na dosiahnutie cieľa nachádzajúceho sa v prvom kvadrante musel NICO najviac vytočiť rameno, a teda najviac zapojiť kĺby. Druhý zaujímavý poznatok, ktorý sme si odniesli z tohto grafu je, že odchýlky v pohyboch trvajúcich tri sekundy sú priemerne menšie ako odchýlky v pohyboch, ktoré mali trvať jednu sekundu. Myslíme si, že jedna sekunda je pomerne krátky čas na prekonanie niektorých vzdialenosí, ktoré môžu nastať medzi cieľom a počiatočnou polohou koncového efektora. Rameno v tomto mohlo nadobudnúť veľkú rýchlosť a keď zrazu zastaví na cieľovej polohe, zotrvačné sily posunú koncový efektor nežiadaným smerom. Jediný kvadrant, kde sú podobné odchýlky v oboch trvaniach je druhý kvadrant. To vychádza z našej teórie o zotrvačnej sile. Druhý kvadrant má NICO najbližšie, preto aj keď je čas na pohyb len jedna sekunda, nemusí vyvinúť takú vysokú rýchlosť.

Druhý graf 7.7 opisuje odchýlky uhlov kĺbov alebo stupňov voľnosti pre náhodné ciele v celej testovacej ploche. Na jednotlivé kvadranty sme túto štatistiku z dôvodu kompaktnosti nerozdeľovali. No porovnali sme pohyb s dĺžkou trvania jedna sekunda a pohyb s dĺžkou trvania tri sekundy. Mediány týchto odchýliek sa veľmi nelíšia od nuly. Najväčšie namerané odchýlky sú v okolí troch stupňov. Najviac z grafu vyčnieva kĺb `r_wrist_x`. Môže to byť spôsobené nárazom koncového efektora do obrazovky a tým nečakané posunutie tohto stupňa voľnosti. Pri pohybe s trvaním tri sekundy takýto náraz pravdepodobne pre pomalú rýchlosť nenastal, takže tam nevidíme takéto odchýlky.

Tretím grafom v tomto experimente je graf opisujúci čas odozvy robota (Obrázok 7.8). Teda uplynulý čas medzi časom, kedy sme poslali príkaz motorom na zmenu polohy



Obr. 7.9: Výsledky štvrtého experimentu - náhodné ciele - porovnanie času trvania pohybu robotického ramena

Záver

Táto práca obsahovala štyri dôležité ciele, ktoré boli stručne opísané v kapitole Úvod. Prvým, základným cieľom bol výber vhodného robotického simulátora pre prácu s humanoidným robotom NICO a otestovanie funkčnosti tohto simulátora. Výber softvéru pre simulovanie robota a krátke otestovanie jeho funkčnosti sme opísali v kapitole Simulátor myGym. S vybratým simulátorom sa mi pracovalo dobre, takže tento cieľ bol úspešne splnený.

Druhým cieľom našej práce bol návrh jednoduchej motorickej úlohy s jedným ramenom smerujúcim na rôzne body na virtuálnej horizontálnej dotykovej obrazovke umiestnenej pred robotom a následné vykonanie tejto úlohy v simulovanom prostredí. Popri tom bolo potrebné zaznamenávať dvojdimenzionálne súradnice pozícií, v ktorých sa dotkne ukazovák robota, teda koncový efektor manipulátora. Zistiť polohu efektora v simulátore nebol problém, keďže 3D polohy všetkých objektov v simulácii sú známe. Návrh úlohy bol opísaný v kapitole 5. Myslím, že sme vybrali vhodné úlohy pre náš účel. Zachovali si jednoduchosť, ale zároveň variáciu.

Tretím, náročnejším cieľom práce bolo zopakovanie navrhnutej úlohy s reálnym fyzickým robotom NICO a skutočnou dotykovou obrazovkou. Náročnejšie to bolo z dôvodu, že išlo o prechod zo simulácie do reality. Ten síce v simulátore myGym implementovaný bol, no ako sme opisovali v kapitole Testy a výsledky, pri tomto prechode je zvykom mať rozličné problémy. Zároveň je známe, že pri práci s robotmi sa taktiež pomerne často vyskytujú nemalé komplikácie. Problémy pri prechode medzi realitami a ich riešenia sú opísané v kapitole 6. Tento cieľ ešte samozrejme obsahoval ukládanie polôh dotykov koncového efektora ramena s obrazovkou pre následné porovnanie s hodnotami z druhého cieľu.

Štvrtý, posledný a najdôležitejší cieľ našej práce bolo ako aj kvantifikovať chybu spôsobenú rozdielom medzi realitami, teda virtuálnou realitou v simulácii a fyzickou realitou, tak aj následné zamyslenie sa nad možnosťami minimalizácie tejto chyby. Bez splnenia tohto cieľa by sme nemali žiadne výsledky, preto bol najdôležitejším cieľom práce. V poslednej kapitole našej práce 7 sme detailne spracovali a popísali výsledky experimentov, trendy, ktoré sme spozorovali a anomálie, ktoré nastali.

Všetky ciele našej práce sme splnili a hlavne sa nám do určitej miery podarilo kvantifikovať medzeru medzi virtuálnou realitou a fyzickou realitou na zaujímavom

humanoidnom robotovi NICO. Získali sme užitočné informácie, ktoré môžeme využiť pri ďalšej práci s týmto robotom.

Literatúra

- [1] Github pygame: Library for digital arts, games and music making. Dostupné 21.5.2024. URL: <https://github.com/pygame>.
- [2] AndyLucny. Github andylucny/nico/touch: Software for processing robot touches on a tablet. Dostupné 21.5.2024. URL: <https://github.com/andylucny/nico/tree/main/touch>.
- [3] Jack Collins, Ross Brown, Jurgen Leitner, and David Howard. Traversing the reality gap via simulator tuning, 2020. [arXiv:2003.01369](https://arxiv.org/abs/2003.01369).
- [4] Jack Collins, Ross Brown, Jürgen Leitner, and David Howard. Follow the gradient: Crossing the reality gap using differentiable physics (realitygrad), 2021. [arXiv:2109.04674](https://arxiv.org/abs/2109.04674).
- [5] Jack Collins, David Howard, and Jürgen Leitner. Quantifying the reality gap in robotic manipulation tasks, 2018. [arXiv:1811.01484](https://arxiv.org/abs/1811.01484).
- [6] Waseda University Humanoid Robotics Institute. Wabot - waseda robot-. Dostupné 7.5.2024. URL: https://www.humanoid.waseda.ac.jp/booklet/kato_2.html.
- [7] Incognite-Lab. Github - incognite-lab/mygym. Dostupné 15.5.2024. URL: <https://github.com/incognite-lab/myGym>.
- [8] Incognite-Lab. mygym's documentation. Dostupné 15.5.2024. URL: <https://mygym.readthedocs.io/en/latest/>.
- [9] University of Hamburg Knowledge Technology Group. Nico neuro-inspired companion — seed robotics. Dostupné 17.5.2024. URL: <https://www.seedrobotics.com/nico-robot>.
- [10] Knowledgetechnologyuhh. Github - knowledgetechnologyuhh/nico-software: Software to run the nico (neuro inspired companion) robot. Dostupné 17.5.2024. URL: <https://github.com/knowledgetechnologyuhh/NICO-software>.

- [11] Quentin Le Lidec, Wilson Jallet, Louis Montaut, Ivan Laptev, Cordelia Schmid, and Justin Carpentier. Contact models in robotics: a comparative analysis, 2023. [arXiv:2304.06372](https://arxiv.org/abs/2304.06372).
- [12] Matplotlib. Github - matplotlib/matplotlib: Plotting with python. Dostupné 21.5.2024. URL: <https://github.com/matplotlib/matplotlib>.
- [13] Hwei Geok Ng, Paul Anton, Marc Brügger, Nikhil Churamani, Erik Fließwasser, Thomas Hummel, Julius Mayer, Waleed Mustafa, Nguyn Linh Chi, Nguyen Quan, Marcus Soll, Sebastian Springenberg, Sascha Griffiths, Stefan Heinrich, Nicolás Navarro-Guerrero, Erik Strahl, Johannes Twiefel, Cornelius Weber, and Stefan Wermter. Hey robot, why don't you talk to me? 08 2017. doi:10.1109/ROMAN.2017.8172383.
- [14] Peter Pukančík. Riadiaci systém s inverznou kinematikou pre mobilné robotické rameno, 2012. Dostupné 4.5.2024: https://dai.fmph.uniba.sk/~petrovic/th12/Pukancik_Rameno.pdf.
- [15] Prasanna Rasal. Humanoid robotics. 9:128–130, 06 2021. doi:<https://doi.org/10.22214/ijraset.2021.35918>.
- [16] Erica Salvato, Gianfranco Fenu, Eric Medvet, and Felice Andrea Pellegrino. Crossing the reality gap: A survey on sim-to-real transferability of robot controllers in reinforcement learning. *IEEE Access*, 9:153171–153187, 2021. doi:10.1109/ACCESS.2021.3126658.
- [17] Tony Punnoose Valayil and Rose Shaji Augustine. Methods to solve forward kinematics of parallel and serial manipulators. *AIP Conference Proceedings*, 2670(1):030003, 12 2022. [arXiv:https://pubs.aip.org/aip/acp/article-pdf/doi/10.1063/5.0115314/16229983/030003_1_online.pdf](https://pubs.aip.org/aip/acp/article-pdf/doi/10.1063/5.0115314/16229983/030003_1_online.pdf), doi:10.1063/5.0115314.
- [18] Kenneth J. Waldron and James Schmiedeler. *Kinematics*, pages 11–36. Springer International Publishing, Cham, 2016. doi:10.1007/978-3-319-32552-1_2.
- [19] Ansei Yonezawa, Heisei Yonezawa, and Itsuro Kajiwara. Simple inverse kinematics computation considering joint motion efficiency. *IEEE Transactions on Cybernetics*, pages 1–12, 2024. doi:10.1109/TCYB.2024.3372989.
- [20] Chengyi Zhao, Yimin Wei, Junfeng Xiao, Yong Sun, Dongxing Zhang, Qiuquan Guo, and Jun Yang. Inverse kinematics solution and control method of 6-degree-of-freedom manipulator based on deep reinforcement learning. *Scientific Reports*, 14(1):12467, 2024. doi:<https://doi.org/10.1038/s41598-024-62948-6>.