

POLITECHNIKA WROCŁAWSKA
WYDZIAŁ ELEKTRONIKI, FOTONIKI I MIKROSYSTEMÓW

KIERUNEK: Automatyka i Robotyka (AiR)

**PRACA DYPLOMOWA
INŻYNIERSKA**

Symulacja sterowalnej bionicznej protezy
realizującej wybrane rodzaje chwytów

Simulation of a controllable bionic prosthesis
implementing selected types of grasps

AUTOR:
Nikodem Iwin

PROMOTOR:
Dr inż. Janusz Jakubiak

Spis treści

1 Wstęp	5
2 Przegląd środowisk symulacyjnych	7
2.1 Co to jest środowisko symulacyjne	7
2.2 Środowiska symulacyjne	7
2.3 Wybór odpowiedniego środowiska	11
2.4 Nvidia Isaac Sim	16
3 Bioniczna proteza	20
3.1 Anatomia ludzkiej kończyny górnej	20
3.2 Struktura kinematyczna ręki	24
3.3 Znaczenie wymiarów ręki oraz predyspozycji człowieka względem ruchliwości i siły chwytu	32
3.4 Bioniczna proteza w porównaniu do ludzkiej ręki	32
3.5 Analiza wymagań ruchowych protezy	33
4 Model protezy	40
5 Sterowanie	47
6 Symulacje	51
6.1 Otwarta oraz zrelaksowana dłoń	51
6.2 Zaciśnięta pięść	53
6.3 Chwyt pęsetowy	54
6.4 Pełny chwyt hakowy	56
6.5 Chwyt kulisty	57
7 Podsumowanie	59
Bibliografia	61
A Załączona płyta CD	65

Spis rysunków

2.1	Przykład ruchomych walców sprawdzający wymagania środowiska	14
2.2	Symulacja ruchomych walców oddziaływających na siebie	15
2.3	Sprawdzenie poprawności zaimplementowanej dynamiki ciała sztywnego, kolizji oraz grawitacji	15
2.4	Środowisko i interfejs Omniverse (z materiałów Nvidia [1])	16
2.5	Architektura systemu [1]	17
2.6	Domyślne okno programu po uruchomieniu	18
2.7	Przykładowe okno ustawień sceny	18
3.1	Schemat kończyny górnej [2]	20
3.2	Anatomia ręki [3]	21
3.3	Stawy palców [4]	22
3.4	Przedstawienie ruchów w obrębie ręki [5]	23
3.5	Przedstawienie ruchów kciuka [5]	23
3.6	Model ręki o 24 DoF reprezentujący przeguby i ramiona, nałożony na ludzką rękę [6]	25
3.7	Model ręki o 24 DoF reprezentujący matematyczną interpretację, nałożony na ludzką rękę [6]	25
3.8	Kinematyka odwrotna [6]	27
3.9	Przypadek redundantny modelu palców, gdy $\theta_{MCPp/o} = 0$ [6]	28
3.10	Płaszczyzny chwytu: a) <i>Pad Opposition</i> , b) <i>Palm Opposition</i> , c) <i>Side Opposition</i> , d) Układ współrzędnych w odniesieniu do dloni [7]	34
3.11	Skategoryzowane chwytu na podstawie połączonej klasyfikacji [7]	35
3.12	Rysunek zrelaksowanej dłoni [8]	36
3.13	Rysunek pięści z różnych perspektyw [8]	36
3.14	Rysunek pięści z przemieszczeniem powodowanym przez siłę [8]	37
3.15	Zdjęcie chwytu pęsetowego o zgiętych palcach dłoni [9]	37
3.16	Rysunek chwytu pęsetowego o delikatnie zgiętych palcach [10]	38
3.17	Rysunek chwytu hakowego [11]	38
3.18	Rysunek chwytu kulistego [11]	39
4.1	Zestawienie zdjęć modelu z różnych perspektyw w domyślnej konfiguracji	40
4.2	Zdjęcie modelu z siatką kolizji (różowe oznaczenia) i oznaczonymi przegubami (niebieskie oznaczenia)	41
4.3	Zdjęcie drzewa modelu	42
4.4	Zdjęcie przykładowego okna przegubu rotacyjnego na podstawie przegubu FFJ1	44
4.5	Zdjęcie przykładowego okna napędu przegubu rotacyjnego na postawie przegubu FFJ1	45
4.6	Zdjęcie importowania modelu poprzez URDF Importer	46

5.1	Zdjęcie drzewa grafów sterowania	49
5.2	Zdjęcie przykładowego sterowania na podstawie grafu <i>ffknuckle</i>	49
6.1	Zdjęcie modelu w konfiguracji domyślnej	51
6.2	Zdjęcie zrelaksowanej ręki Bebionic [12]	52
6.3	Zdjęcie modelu reprezentującego zrelaksowaną protezę	52
6.4	Zdjęcie pięści robotycznej ręki Shadow Dexterous Hand E [13]	53
6.5	Zdjęcie modelu w trakcie zaciskania pięści	53
6.6	Zdjęcie modelu zaciśniętej pięści	54
6.7	Zdjęcie chwytu pęsetowego zrealizowanego przez protezę Bebionic [12]	54
6.8	Zdjęcie modelu w trakcie realizacji chwytu pęsetowego	55
6.9	Zdjęcie modelu w momencie chwytu pęsetowego	55
6.10	Zdjęcie modelu w trakcie realizacji chwytu hakowego	56
6.11	Zdjęcie modelu w momencie chwytu hakowego	56
6.12	Chwycona piłka tenisowa [14]	57
6.13	Zdjęcie modelu w trakcie realizacji chwytu kulistego	57
6.14	Zdjęcie modelu w momencie chwytu kulistego	58

Spis tabel

2.1	Tabela porównująca wcześniej wymienione środowiska do postawionych kryteriów	13
3.1	Parametry D-H dla palców	26
3.2	Parametry D-H dla kciuka	27
3.3	Zakres ruchowy stawów każdego z palców [6]	31
4.1	Tabela zakresów ruchów przegubów modelu	43
5.1	Sterowanie ręką przy pomocy klawiatury	48

Rozdział 1

Wstęp

Robotyka towarzyszy ludzkości w każdej przestrzeni codziennego funkcjonowania. Spotykamy się z nią w przemyśle, w gospodarstwach domowych, w kosmosie oraz na ulicy. Stała się ona częścią naszego życia. Dzięki rozwojowi medycyny i bioniki, „część naszego życia” nabiera nowego znaczenia, gdyż coraz częściej leczy się ludzi przy pomocy osiągnięć robotyki. Nie tylko za pomocą robotów medycznych takich jak „Da Vinci” czy innego rodzaju aparaty diagnostyczno-medyczne, ale są to również wszechzupy, protezy i inne rozwiązania pozwalające na walkę z przeciwnościami losu czy niepełnosprawnością.

Bioniczne protezy ułatwiają wykonywanie codziennych czynności i pozwalają wrócić do normalnego funkcjonowania osobom niepełnosprawnym, po wypadkach czy amputacjach. Dzięki postępowi w rozwoju bioniki i medycyny oraz różnego rodzaju dofinansowaniach sztuczne kończyny stają się coraz bardziej popularne, lecz wciąż najlepsze rozwiązania dużo kosztują i nie każdy może sobie na nie pozwolić. Ponadto ludzki organizm jest na tyle złożony, że odtworzenie jego zachowań i ruchów jest wielopłaszczyznowym problemem inżynieryjnym. Jednym z najtrudniejszych zagadnień jest zbudowanie protezy ręki, która umożliwi użytkownikowi poczucie naturalności i swobody w codziennym funkcjonowaniu sztucznej kończyny. Wynika to z faktu, iż ludzka dłoń, będąc stosunkowo małą i lekką posiada ogromne zdolności manipulacyjne, wraz z możliwością pracy przy dużych obciążeniach. Co więcej, jest narzędziem bardzo precyzyjnym, a przy tym elastycznym i wytrzymały. Obecnie nieprzerwanie poszukiwane są rozwiązania, które spełnią powyższe wymagania, przy czym będą dostępne dla każdego. Pracują nad nimi zarówno małe start-upy, jak i ogromne korporacje. W związku z wielopoziomową problematyką odtworzenia sterowalnej roboczynej dłoni i nieodłącznymi zawiłościami wymagane są ogromna wiedza, znajomość technologii oraz szeroki wachlarz narzędzi. Jednym z tych narzędzi są symulacje komputerowe.

Symulacje pozwalają na zwizualizowanie celu, do którego się dąży, zapoznanie się z problematyką projektu, oraz testowanie rozwiązań w trakcie trwania prac nad budową fizycznego urządzenia. Dzięki temu możliwe jest bezpieczne sprawdzanie właściwości proponowanych rozwiązań, oszczędzanie zasobów i ograniczenie strat związanych z możliwymi błędami. Informacje, które ze sobą niosą, ich praktyczność oraz opłacalność, sprawiają, że są one użyteczne dla większości projektów. Jednym z nich jest projekt protezy dloni, którego symulacje są niezwykle cennymi dla badania oraz opracowania najbardziej zbliżonego do rzeczywistego rozwiązania. Ta wiedza pozwala również rozwijać się innym dziedzinom np. fizjoterapeuti mogą lepiej rehabilitować, psychologowie mogą lepiej odczytywać emocje, projektanci mogą tworzyć ergonomiczne przedmioty, a programiści mogą rozróżnić czy czynność wykonuje robot, czy

człowiek.

W celu znalezienia odpowiedniego narzędzia do realizacji symulacji chwytu bionicznej protezy reki należy najpierw dokonać przeglądu istniejących rozwiązań. Każde środowisko pozwala na symulację, jednakże różnią się one swoim funkcjonalnością. Na rynku istnieje wiele programów pełniących zarazem odmienne, jak i te same funkcje, co wynika z ich głównego przeznaczenia. Jedne skupiają się na symulacji i programowaniu robotów, a inne opierają swoje zalety na aspekcie ładnej wizualizacji i dokładnej implementacji fizyki. Ze względu na szeroką gamę dostępnych rozwiązań, które różnią się czasem drobnostkami, bardzo ważnym elementem jest wybór odpowiedniego środowiska. W przypadku błędnej decyzji może się okazać, że dany projekt może nie być możliwy do zrealizowania zgodnie z założonymi celami.

Poniższa praca dyplomowa została napisana w celu przybliżenia problematyki symulacji bionicznej ręki. W tym celu zostanie wybrane środowisko symulacyjne oraz zostanie w nim zaimplementowany model protezy. Aby poprzeć model wiedzą teoretyczną, zostanie opracowany segment poświęcony anatomii ręki, jej reprezentacji kinematycznej oraz chwytach. Następnie na podstawie informacji o rzeczywistej ręce i modelu zostaną zaimplementowane algorytmy sterowania, które zostaną zweryfikowane przez symulacje komputerowe.

Pracę podzielono na rozdziały, które stopniowo wprowadzają w problematykę oraz rozwiązywanie tematu symulacji bionicznej protezy realizującej wybrane chwyty. W rozdziale 2 przedstawiono, czym jest środowisko symulacyjne, omówiono dostępne rozwiązania oraz proces wyboru odpowiedniego środowiska, a kończąc rozdział, opisano szerzej docelowy symulator. W kolejnym rozdziale dokonano analizy reki pod kątem matematycznym oraz kinematycznym i przedstawiono klasyfikację chwytów. Rozdział 4 przedstawia model protezy. Następnie w rozdziale 5 opisano zaimplementowane sterowanie. Przedostatni rozdział zawiera symulacje wybranych rodzajów chwytów, a zwieńczeniem pracy jest rozdział 6, w którym podsumowano osiągnięte rezultaty.

Rozdział 2

Przegląd środowisk symulacyjnych

Obecnie powszechnym podejściem przy opracowywaniu nowych urządzeń jest symulowanie projektowanych mechanizmów. Od symulacji wymaga się sprawdzenia, czy dane rozwiązanie spełnia założone cele, dlatego przygotowuje się modele, które optymalizują koszt i jakość informacji. W zależności od potrzeb korzysta się z różnych środowisk, a na ich podstawie można przygotowywać, zarówno sprzęt, jak i oprogramowanie. W poniższym rozdziale zostały przedstawione dostępne rozwiązania, ich wady i zalety, a na końcu przedstawiono i omówiono wybrane środowisko, które wykorzystano do w dalszej części pracy.

2.1 Co to jest środowisko symulacyjne

W pracy definiujemy środowisko symulacyjne jako oprogramowanie komputerowe pozwalające na przedstawienie ruchu modelu oraz jego interakcji z zaprojektowanym otoczeniem. Ich użycie umożliwia przeprowadzanie licznych testów, na podstawie których można wyciągnąć rozmaite wnioski. W odniesieniu do symulacji bionicznej protezy można uzyskać między innymi informacje o trajektorii ruchu, siłach panujących w układzie, czy o odczytanych wartościach czujników. Dzięki symulacji możliwe jest zaprojektowanie i sprawdzenie, czy proteza powstała w planowany sposób będzie funkcjonować poprawnie zgodnie z jej przeznaczeniem. Pozwala to na wczesne wykrycie błędów oraz diagnozę czy przyjęte założenia, mające miejsce w trakcie projektu, są właściwe. Dodatkowo często w systemie można poprawić model oraz sprawdzić jak będzie się zachowywał po zmianach.

2.2 Środowiska symulacyjne

Na rynku istnieje wiele środowisk symulacyjnych. Każde powstało z myślą o konkretnym przeznaczeniu i zawiera różne funkcjonalności. Można wyróżnić między innymi środowiska edukacyjne, środowiska przemysłowe oraz środowiska o zastosowaniu ogólnym.

2.2.1 Środowiska edukacyjne

Środowiska edukacyjne powstały z myślą o poszerzaniu wiedzy z zakresu międu innymi robotyki, mechaniki, fizyki czy wytrzymałości materiałów. Zazwyczaj ich interfejs jest przejrzysty i prosty w obsłudze. Posiadają również bogatą bibliotekę gotowych modeli, z których można uczyć się różnych zagadnień z wyżej wymienionych

dziedzin. Ich głównymi wadami są często brak możliwości rozwoju przygotowanych projektów, brak możliwości implementacji własnych rozwiązań lub ograniczona komunikacja z zewnętrznym oprogramowaniem.

- Robot Virtual Worlds [15] - oprogramowanie stworzone z myślą o nauce programowania robotów dla dzieci i uczniów na różnym poziomie edukacji. Zaletą jest nauka programowania różnych robotów bez wymogu posiadania ich rzeczywistych wersji. Oprogramowanie obsługuje między innymi roboty VEX i LEGO. Środowisko jest płatne, ale możliwe jest pobranie 60-dniowej wersji bezpłatnej.
- Virtual Robotics Toolkit [16] - środowisko do symulacji i programowania robotów LEGO. Środowisko pozwala na tworzenie rozmaitych robotów LEGO bez wymogu posiadania klocków potrzebnych do ich budowy, a następnie na testowanie ich w różnych warunkach przygotowanych przez twórców. Tak jak Robot Virtual Worlds jest to środowisko płatne.

Środowiska edukacyjne są przejrzystym narzędziem do edukowania, ale ze względu na swoje ograniczenia nie są najlepsze do symulowania złożonych projektów realizujących inne zagadnienia niż założone przez twórców.

2.2.2 Środowiska przemysłowe

Środowiska przemysłowe powstały z myślą o symulacji rzeczywistych rozwiązań dostarczanych przez firmy, często z zakresu przemysłu. Jednym z ich głównych celów jest programowanie robotów w trybie offline/online. Poprzez wierne odwzorowanie rzeczywistych manipulatorów oraz innych rozwiązań można dokładnie zaplanować ruch robotów i ustawić ich parametry, zanim zostanie uruchomiona ich praca, a następnie zdalnie je serwisować pod kątem oprogramowania. Ponieważ są to narzędzia dostarczane przez firmy i przeznaczone do konkretnego zastosowania, ich głównym założeniem jest obsługa urządzeń danego producenta, co często stanowi przeszkodę dla rozwoju projektu spoza branży lub wręcz uniemożliwia jego zrealizowanie.

- ABB Robot Studio [17] - oprogramowanie pozwalające na sterowanie robotami stworzone przez firmę ABB. Poprzez identyczne odwzorowanie maszyn oraz możliwość ich programowania jest częstym narzędziem wybieranym do odwzorowywania linii produkcyjnych. Oprogramowanie oparte jest na płatnej licencji z możliwością pobrania trzydziestodniowej darmowej wersji próbnej.
- Robotmaster [18] - środowisko służące programowaniu robotów offline, symulacji i generowaniu kodu. Firma Hypertherm Associates wyróżnia zalety oprogramowania między innymi w wyznaczaniu dokładnej trajektorii pracy manipulatora, edycji symulacji przy pomocy „clik and drag”, automatyzacji optymalizacji ruchu robota oraz kompatybilności oprogramowania z wieloma innymi producentami.
- Fanuc Roboguide [19] - oprogramowanie podobne do ABB Robotic Studio, które zostało stworzone przez firmę Fanuc dla ich maszyn. Tak jak oprogramowanie firmy ABB jest narzędziem płatnym z darmową wersją próbną. Poprzez użycie specjalnych rozszerzeń środowisko pozwala na symulację między innymi paletyzacji, spawania, malowania oraz szybkiego tzw. „pick and place”.

- RoboDK [20] - środowisko służące programowaniu robotów offline. Jest to proste i skuteczne narzędzie, które zawiera szeroką gamę urządzeń wykorzystywanych w przemyśle. Producent przedstawia obsługę programu w pięciu krokach: wybór robota, wybranie jego efektora, wgranie modelu 3D, symulacja toru ruchu, generowanie oprogramowania. Jak większość środowisk przemysłowych jest to oprogramowanie płatne z darmową wersją próbną.
- KUKA.Sim [21] - oprogramowanie dostarczone przez firmę KUKA charakteryzuje się modułowością, dzięki której można efektywnie i elastycznie pracować na środowisku. Poprzez dodatki takie jak KUKA.Sim Modeling, KUKA.Sim Connectivity i KUKA.Sim ArcWelding można wyposażyć środowisko w potrzebne oprogramowanie, nie będąc zmuszonym do zakupu wszystkiego co oferuje firma. Również jest to oprogramowanie płatne z darmową wersją próbną.

Na podstawie przedstawionych środowisk widać, że są to narzędzia praktyczne, których głównym zadaniem jest programowanie robotów przemysłowych oraz ich obsługa pod kątem oprogramowania. Ze względu na swoją specjalizację nie posiadają rozbudowanych mechanizmów związanych z fizyką, modelowaniem, czy wizualizacją.

2.2.3 Środowiska o zastosowaniu ogólnym

Środowiska o zastosowaniu ogólnym są optymalne dla szerokiej gamy zastosowań. Pomimo braku specjalistycznej obsługi takiej, jaką posiadają środowiska przemysłowe, są wszechstronne rozwinięte i pozwalają na najładniejsze wizualizacje. Większość z nich ma możliwość złożonego modelowania oraz posiada obsługę programów zewnętrznych, przy pomocy których można np. sterować modelem. Ponadto pozwalają na symulowanie różnych warunków fizycznych oraz na przedstawienie różnego rodzaju treści między innymi manipulatorów, robotów mobilnych, pojazdów, czy sztucznych kończyn. W związku z ich szerokim zastosowaniem istnieje wiele dokumentacji i materiałów pomocniczych tworzonych przez twórców oraz społeczność, które ułatwiają ich obsługę. Dodatkowo warto wspomnieć, że wiele z nich opiera się na darmowej dystrybucji lub bezpłatnej wersji edukacyjnej.

- MSC Adams [22] - narzędzie stworzone przede wszystkim w celu dokładnych symulacji i obliczeń dynamiki wieloobiektowej. Dzięki licznym modułom można rozszerzać środowisko o dodatkowe opcje jak np. modelowanie poprzez moduł Adams Modeler. Dzięki zaimplementowanym algorytmom obliczeniowym simulator dokładnie odwzorowuje działanie fizyki i chętnie jest wykorzystywany przez różne firmy, które za jego pośrednictwem mogą ustalać między innymi interakcje między obiektami, wywierane siły, czy ich sterowność. Jest to narzędzie płatne, ale ma ograniczoną wersję studencką.
- Nvidia Isaac Sim [23] - stosunkowo nowe oprogramowanie służące symulacji robotyki i generowaniu syntetycznych danych. Narzędzie pozwala również na opracowywanie i testowanie robotów, i układów opartych na sztucznej inteligencji. W związku z zastosowaniem technologii opartej na ray-tracing'u środowisko wymaga mocniejszych procesorów i kart graficznych, co rekompensuje dokładnym odwzorowaniem fizyki i fotorealistyczną wizualizacją.
- Microsoft Robotics Developer Studio [24] - niewspierane od 2012 roku środowisko do kontroli robotów i ich symulacji, wykorzystywane zarówno w celach

edukacyjnych, jak i komercyjnych. Docelowym systemem, na którym działało środowisko był Windows 7. Narzędzie posiada bogatą bibliotekę popularnych wtedy robotów i scen do ich wizualizacji. W związku ze wbudowanym systemem modelowania obiektów 3D narzędzie może być wciąż wykorzystywane, lecz ze względu na brak aktualizacji może być problematyczne.

- Gazebo [25] - popularne środowisko z bogatą dokumentacją i rozwiniętą społeczeństwem. Ze względu na swoje możliwości i zastosowania jest szczególnie popularne wśród użytkujących ROS'a. Środowisko jest stale rozwijane od 2002 roku i wraz z rozwojem technologii wyposażane jest w kolejne rozszerzenie, ułatwienia i możliwości, a dzięki charakterowi typu Open Source stale gromadzi nowych użytkowników. Gazebo oparte jest na następujących zasadach: brak specjalizacji, stabilność, aktualność, modularność, rozszerzalność, elastyczność, utrzymanie, przenośność. W wyniku powyższych stanowi obecnie jedno z najlepszych rozwiązań darmowych i długotrwałych.
- MORSE [26] - jest to ogólny symulator o zastosowaniu akademickim stworzony przez francuskie LAAS-CNRS. Jest wykorzystywany w celu symulacji 3D różnej wielkości środowisk wraz z autonomicznymi robotami przy pomocy skryptów napisanych w Pythonie. Jego niewątpliwą zaletą jest możliwość pracy z poziomu wiersza poleceń. Jest to darmowe oprogramowanie przygotowane do pracy na systemie Linux, chociaż trwają prace nad wersjami na inne systemy operacyjne. Dzięki współpracom z licznymi ośrodkami akademickimi środowisko jest ciągle rozszerzane i aktualizowane, a nowe rozwiązania są wprowadzane bez większych opóźnień. Należy jednak zwrócić uwagę, iż nie posiada ono graficznego interfejsu i nie jest to dokładny symulator.
- CoppeliaSim [27] - kiedyś zwany V-Rep to robotyczny symulator z zintegrowanym środowiskiem, w którym można sterować każdym z obiektów z osobna. Ze względu na możliwość sterowania np. poprzez skrypty, API czy ROSa oraz wsparcie dla różnych języków programowania takich jak np. C/C++, Python, Lua czy nawet Matlab, jest wykorzystywane jako wszechstronne narzędzie. Przykładowo bada się i rozwija zagadnienia związane z algorytmami sterowania, automatyzacją i symulacją procesów, nadzorowaniem ruchu, jak również weryfikacją mechanizmów bezpieczeństwa. Środowisko posiada bogatą dokumentację oraz darmową wersję edukacyjną.
- Webots [28] - oprogramowanie typu Open Source do symulowania robotów. Jest wykorzystywane zarówno przez środowisko biznesowe jak np. Toyota lub Huawei, ale również przez środowiska akademickie. Jest narzędziem wieloplatformowym używanym od 1998 roku, a wyróżnia ją możliwość eksportowania symulacji jako pliku wideo, interaktywnej sceny w języku HTML lub transmisji na żywo do innej przeglądarki.
- GraspIt! [29] - symulator do testowania i badania robotycznych chwytów. Środowisko umożliwia wgrywanie obiektów do sceny oraz interakcję z nią. Oprogramowanie zawiera bogatą bibliotekę chwytów oraz robotycznych modeli rąk. Pozwala na dokładne planowanie ruchu, dynamiki oraz badanie zachowywania się modeli. Symulator posiada wsparcie dla różnych czujników oraz specjalistycznych ruchów dloni. Środowisko oparte jest na licencji GNU General Public License i może być używane między systemami MS Windows oraz Ubuntu Linux.

Środowiska o zastosowaniu ogólnym sprawdzają się w przypadku projektów z różnych dziedzin, realizujących zdywersyfikowane zadania i problemy. Można za ich pomocą symulować jednakowo zagadnienia z branży przemysłowej, hobystyczne projekty oraz projekty naukowe. Dodając do nich wszechstronności bogate dokumentacje z przykładami, są one ciekawym rozwiązaniem dla każdego.

2.3 Wybór odpowiedniego środowiska

Aby znaleźć odpowiednie środowisko, które pozwoli na symulacje bionicznej prototypy dloni, przeprowadzono trzy stopniową selekcję.

2.3.1 Selekcja wstępna

Pierwszym etapem selekcji odpowiedniego środowiska był przegląd literatury oraz określenie listy kryteriów dla finalnego rozwiązania. Lista wymagań postawiona środowiskom wygląda następująco:

1. Poruszanie modelem zgodnie z jego fizycznymi możliwościami.
2. Wyposażenie modelu w czujniki dotyku.
3. Poruszanie i wpływanie na inne obiekty zgodnie z fizyką.
4. Tworzenie i importowanie modeli trójwymiarowych.
5. Modelowanie elementów otoczenia.
6. Poruszanie modelem zgodnie z jego kinematyką.
7. Możliwość komunikacji zwrotnej z symulacją.
8. Pokazywanie oraz działanie siłami.
9. Możliwość komunikacji zewnętrznej między innymi z zewnętrznymi programami w celu sterowania, np. przy pomocy konsoli.
10. Możliwość sterowania wewnątrz środowiska w czasie rzeczywistym.
11. Posiadanie dobrej dokumentacji stworzonej przez twórców oprogramowania lub utworzoną przez społeczność.
12. Posiadanie przejrzystego i prostego interfejsu.
13. Posiadanie czytelnej i ładnej wizualizacji.
14. Dostępność darmowego dystrybucji albo wersji edukacyjnej na systemie Linux lub Windows 10.

2.3.2 Selekcja porównawcza

Następnie po przeglądzie literatury oraz utworzeniu listy wymagań opracowano tabelę 2.1, w której zawarto podsumowanie kryteriów spełnionych przez dane środowisko. Poprzez porównanie między sobą możliwości programów na podstawie spełnionych kryteriów, wyznaczono 3 interesujące rozwiązania. Były to Isaac Sim, Gazebo oraz MORSE, które następnie poddano selekcji końcowej, podczas której rozpatrywano je pod kątem realizacji projektu bionicznej ręki.

Warto wspomnieć, dlaczego środowisko GraspIt! pomimo bycia środowiskiem przeznaczonym dla wizualizacji robotycznych rąk i chwytów zostało wykluczone. Chociaż spełnia dużą liczbę kryteriów, ma bardzo rozbudowany system obsługi chwytów oraz bogatą bibliotekę robotycznych dloni, jednak środowisko nie jest wyposażone w najładniejszą wizualizację, posiada mało przykładów użycia i instrukcji stworzonych przez społeczność, które mogłyby być pomocne w trakcie tworzenia i obsługi programu oraz nie jest aktywnie rozwijane.

2. Przegląd środowisk symulacyjnych

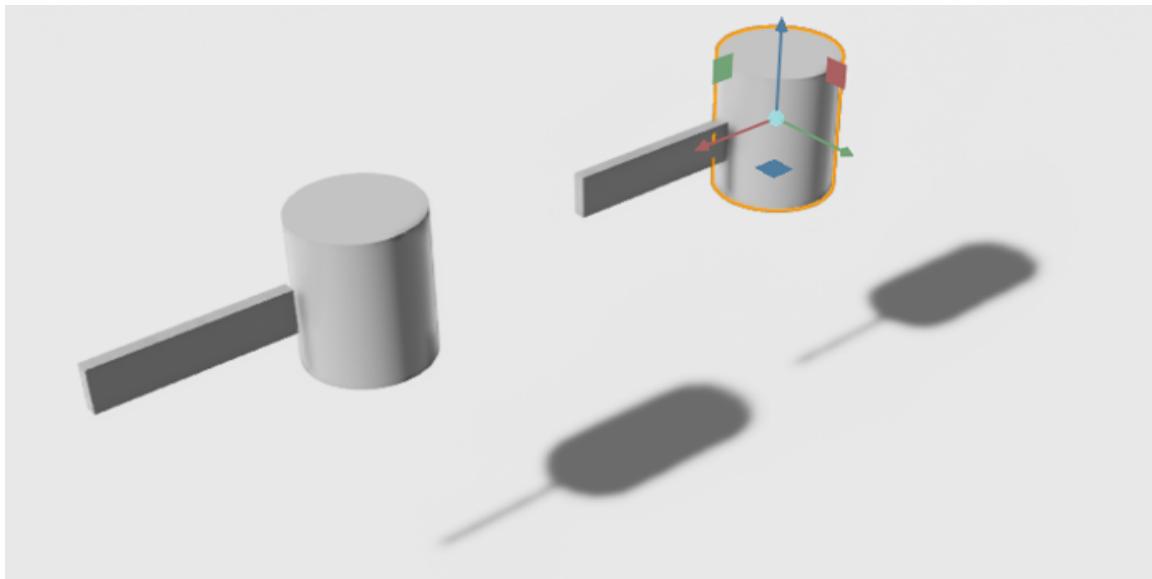
Tabela 2.1 Tabela porównująca wcześniejszymione środowiska do postawionych kryteriów

Środowisko	Spełnione Kryteria	Uwagi
Robot Virtual Worlds	1,6,12,14 1,6,12,14	Obydwia środowiska nie spełniają wymagań.
Virtual Robotics Toolkit	1,4,5,6,7,9,11,12,13,14 1,4,5,6,7,9,11,12,13,14	Są to rozwiązania płatne, a ich wersje edukacyjne i darmowe są ograniczone czasowo oraz nie posiadają wszystkich możliwości. Nie można w nich sterować w czasie rzeczywistym oraz nie zawsze jest zachowana pełna fizyka świata.
ABB Robot Studio	1,4,5,6,7,9,11,12,13,14 1,4,5,6,7,9,11,12,13,14	
Robotmaster	1,4,5,6,7,9,11,12,13,14 1,4,5,6,7,9,11,12,13,14	
Fanuc Roboguide	1,4,5,6,7,9,11,12,13,14 1,4,5,6,7,9,11,12,13,14	
RoboDK	1,4,5,6,7,9,11,12,13,14 1,4,5,6,7,9,11,12,13,14	
KUKA.Sim	1,4,5,6,7,9,11,12,13,14 1,2,3,4,5,6,7,8,9,11,12,13,14	Nie można sterować modelem w czasie rzeczywistym.
MSC Adams	1,2,3,4,5,6,7,8,9,11,12,13,14 1,4,5,6,7,8,9,12	Jest to przestarzałe środowisko z ostatnią dziającą dystrybucją na systemie Windows 7.
Microsoft Robotics Developer Studio	1,2,3,4,5,6,7,8,9,10,11,12,13,14 1,2,3,4,5,6,7,8,9,10,11,12,13,14	
Nvidia Isaac Sim	1,2,3,4,5,6,7,8,9,10,11,12,13,14 1,2,3,4,5,6,7,8,9,10,11,12,13,14	
Gazebo	1,2,3,4,5,6,7,8,9,10,11,12,13,14 1,2,3,4,5,6,7,8,9,10,11,12,13,14	
MORSE	1,2,3,4,5,6,7,8,9,10,11,12,13,14 1,2,3,4,5,6,7,8,9,11,12,13,14	
CoppeliaSim	1,2,3,4,5,6,7,8,9,11,12,13,14 1,2,3,4,5,6,7,8,9,11,12,13,14	Nie można sterować modelem w czasie rzeczywistym.
Webots	1,2,3,4,5,6,7,8,9,11,12,13,14 1,2,3,4,5,6,7,8,9,11,12,13,14	
GraspIt!	1,2,3,4,5,6,7,8,9,11,12,13,14 1,2,3,4,5,6,7,8,9,11,12,13,14	

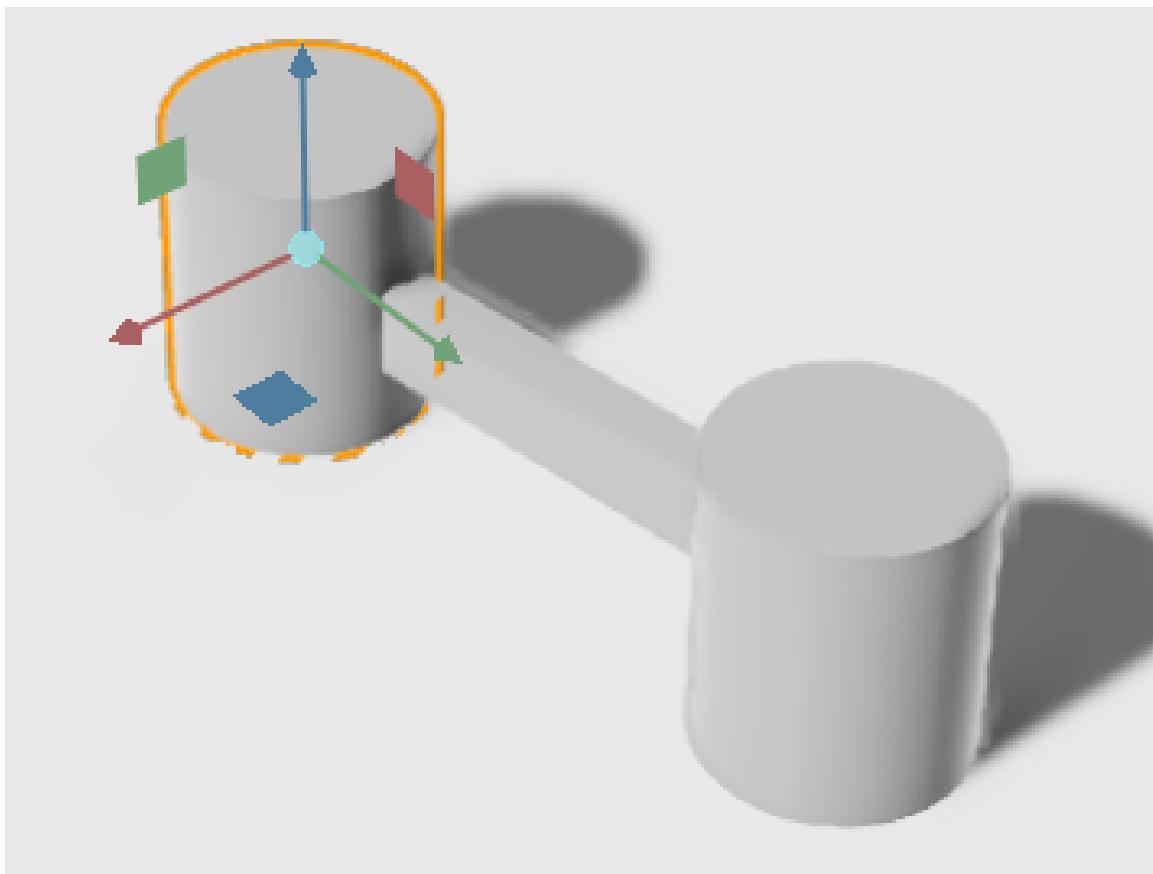
2.3.3 Selekcja końcowa

W końcowej selekcji byłybrane pod uwagę 3 środowiska o zastosowaniu ogólnym: MORSE, Isaac Sim oraz Gazebo. Zgodnie z ich dokumentacjami, znalezionymi materiałami pomocniczymi oraz informacjami od użytkowników wynikało, że środowiska nadają się w zupełności do celu realizacji bionicznej protezy dłoni. MORSE, Gazebo oraz Isaac Sim są darmowymi dystrybucjami. Każde z nich pozwala na modelowanie lub importowanie obiektów 3D oraz pozwala na komunikację z zewnętrznym oprogramowaniem. Jednakże, ponieważ symulacja bionicznej protezy zakłada dużą dokładność, MORSE został odrzucony, gdyż na jego oficjalnej stronie można przeczytać, że nie jest to dokładny fizyczny symulator [26]. Z tego powodu pozostały 2 środowiska Isaac Sim oraz Gazebo, które mają podobne możliwości. Obydwa obsługują format URDF(ang. *Unified Robot Description Format*) [30], w którym wykonany jest model protezy. Posiadają rozbudowane dokumentacje i jedyną znaczącą różnicą jest wygląd środowiska oraz symulacji.

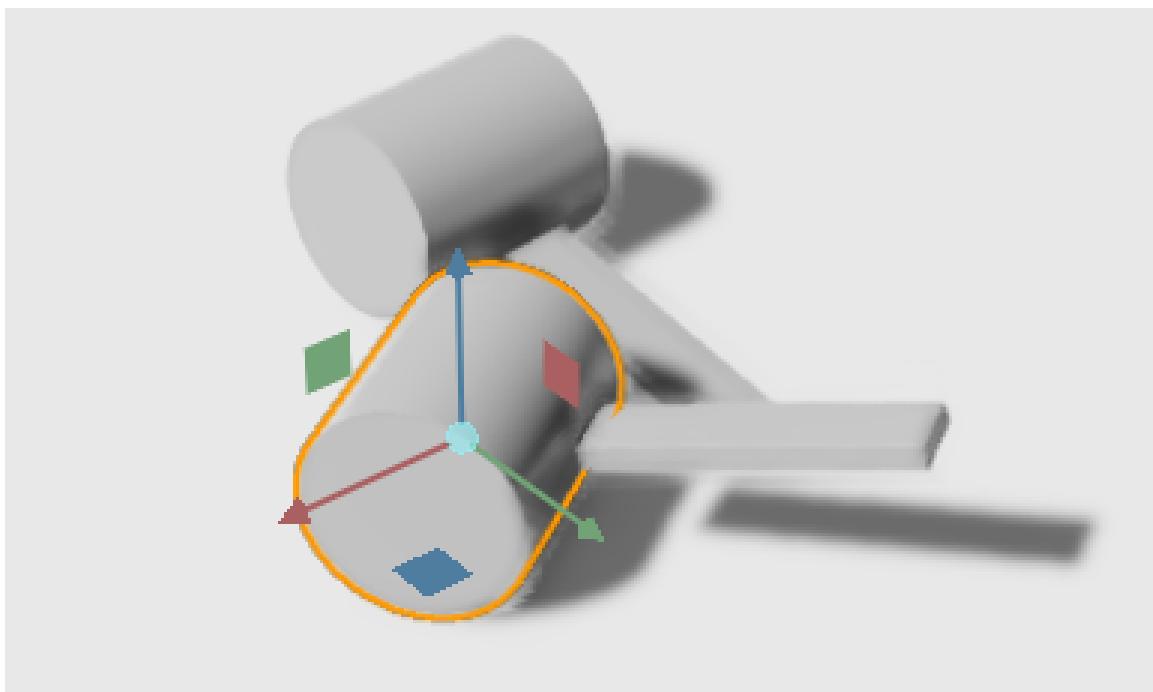
Isaac Sim jest narzędziem nowoczesnym, dokładnym obliczeniowo i posiadającym schludny interfejs. Warto zaznaczyć, że posiada również bogatą w przykłady instrukcję dostarczoną od twórców i społeczności [1]. W celu upewnienia się, czy środowisko zawiera wszystkie potrzebne funkcjonalności do realizacji pracy, opracowano prostą scenę z obiekty (rysunek 2.1). Na jej podstawie sprawdzono symulację ruchu oraz zachowanie fizyki ciał sztywnych, fizyki kolizji oraz siły grawitacji. Walce z prostopadłościanami zostały utworzone jako obiekty 3D, które połączono i wprawiono w ruch. Ustawiono również parametry dynamiki ciała sztywnego oraz kolizji. Na rysunku 2.2 sprawdzono, czy ciała mogą wywierać na siebie wpływ, co zostało potwierdzone, gdyż po zetknięciu się prostopadłościanów obydwa walce zmieniły razem swój ruch, zgodnie z przewidywaniami. Sprawdzenie fizyki odbyło się poprzez poruszenie ich zewnętrznym ruchem (*Shift* i *LPM*) co zostało zobrazowane na rysunku 2.3. Na podstawie przeprowadzonych symulacji stwierdzono, że Isaac Sim spełnia wszystkie założenia i został wybrany jako narzędzie do symulacji bionicznej protezy ręki.



Rysunek 2.1 Przykład ruchomych walców sprawdzający wymagania środowiska



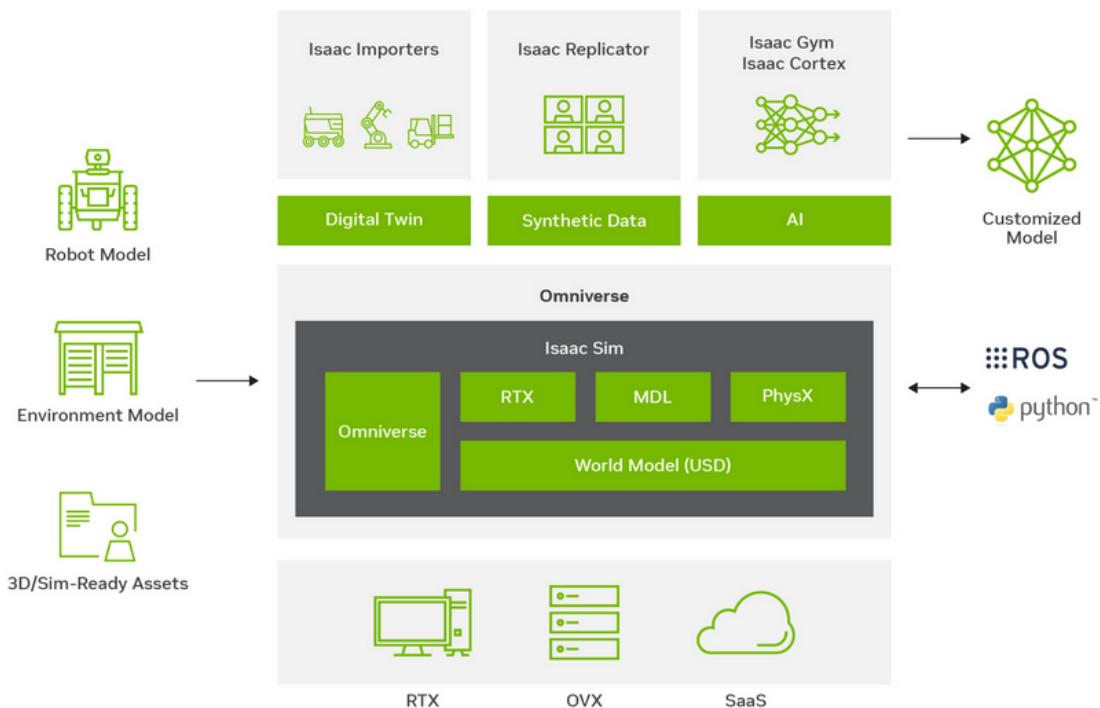
Rysunek 2.2 Symulacja ruchomych walców oddziaływających na siebie



Rysunek 2.3 Sprawdzenie poprawności zaimplementowanej dynamiki ciała sztywnego, kolizji oraz grawitacji

2.4 Nvidia Isaac Sim

Nvidia Isaac Sim to precyzyjne narzędzie do modelowania i symulacji robotów, zarządzania nimi oraz generowania danych. Środowisko wyróżnia się na tle pozostałych zaawansowanymi możliwościami fotorealistycznej wizualizacji, modularną architekturą systemu, bezproblemową łącznością i współpracy między różnymi komputerami, intuicyjną obsługą oraz możliwością wpływania na obiektyw w trakcie symulacji w czasie rzeczywistym [23]. Jak można przeczytać na głównej stronie producenta, jest to narzędzie stosowane głównie w celach: wizualizacji i omawiania zagadnienia manipulacji, symulacji nawigacji robotów, koordynowania robotów wraz z ich cechami oraz scenami. Isaac Sim jest częścią wciąż rozbudowującego się zestawu narzędzi *Omniverse*, który ma umożliwiać pracę nad jak największą liczbą zagadnień z zakresu robotyki i symulacji komputerowych (rysunek 2.4). Architektura systemu została pokazana na rysunku 2.5.



Rysunek 2.4 Środowisko i interfejs Omniverse (z materiałów Nvidia [1])



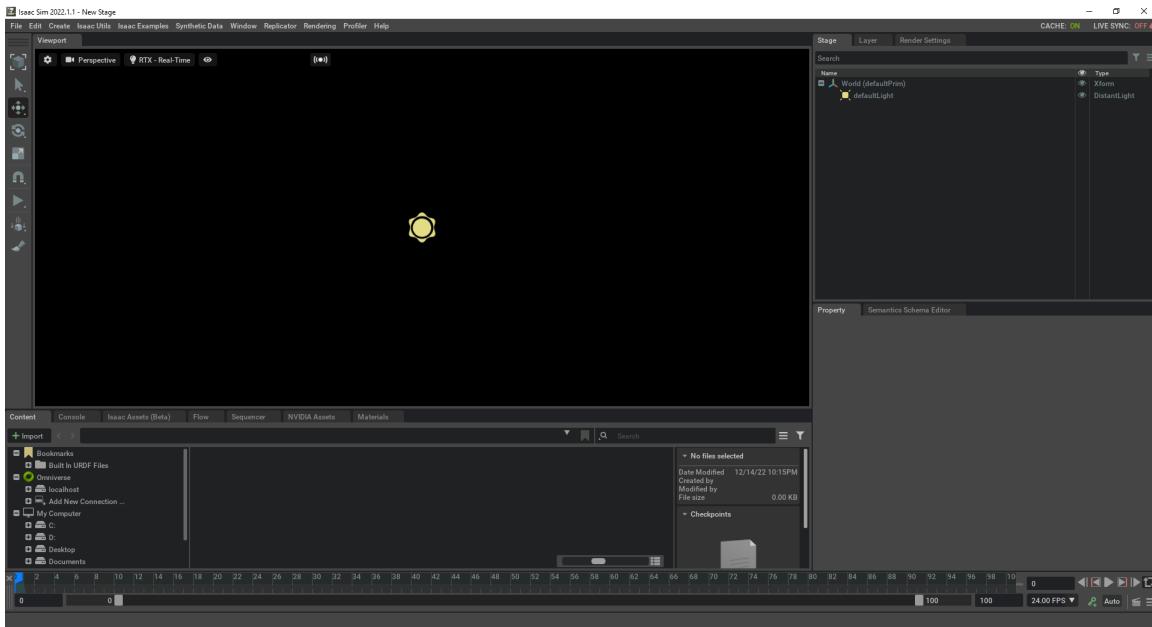
Rysunek 2.5 Architektura systemu [1]

Nvidia zaleca korzystanie z nowoczesnych kart graficznych opartych na swoich procesorach [31], co może być problematyczne dla osób nieposiadających nowoczesnych i wydajnych podzespołów komputerowych. Jednakże twórcy umożliwiają również rozwiązanie opartego na pracy w chmurze. Dodatkowo warto wspomnieć, że dzięki współpracy z innymi narzędziami jak np. Isaac Gym (odpowiedzialne głównie za trenowanie sztucznej inteligencji), Isaac Sim jest narzędziem mającym różnorodne zastosowania w wielu dziedzinach.

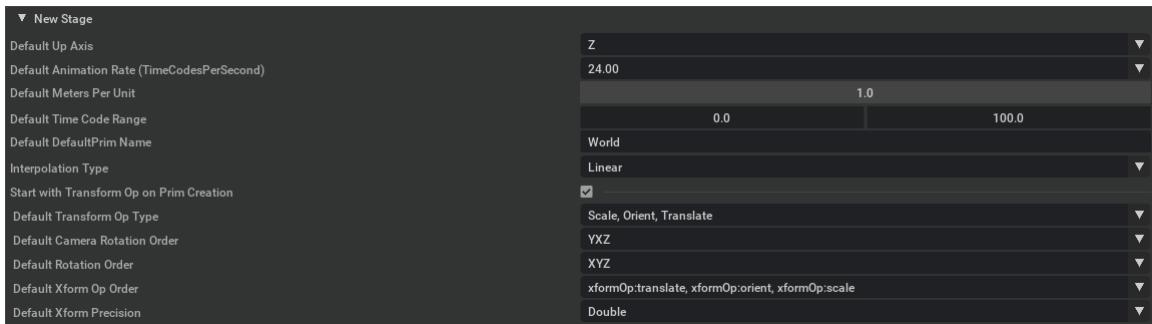
W trakcie tej pracy wykorzystano Isaac Sim w celu zainportowania modelu robotypycznej ręki, przygotowania sceny oraz napisania jej sterowania. Poniżej w kolejnych podrozdziałach przedstawiono i omówiono możliwości, które zostały wykorzystane przy realizacji pracy.

2.4.1 Ustawienia środowiska i główna zawartość

Po uruchomieniu programu wyskakuje domyślne okno środowiska przedstawione na rysunku 2.6. Program pozwala na ustawienie i sprawdzenie właściwości sceny. Są to między innymi oś determinująca wysokość jednostki, które są wykorzystywane na scenie (domyślnie są to metry), kolejność rotacji (domyślnie X, Y i Z). Przykładowe okno parametrów zostało pokazane na rysunku 2.7. Środowisko zawiera swobodną manipulację kamery, korzystanie z 3 domyślnych punktów widoku (z góry, z przodu i z prawej). Dodatkowo można tworzyć własne perspektywy, które można zapisywać, a następnie przełączać między nimi w dowolnym momencie. Tak samo istnieje możliwość tworzenia licznych okien z różnymi widokami, które pokazują bieżący obraz. Isaac Sim zawiera również bogatą bibliotekę przykładów robotów, scen czy obiektów dekoracyjnych, które można wgrać poprzez przeciągnięcie ich z odpowiedniej zakładki. Symulator zawiera również konsolę oraz pozwala na nagrywanie symulacji, a następnie ich odtwarzanie. Ponadto, podstawowe funkcje pakietu mogą zostać rozszerzone przez instalację paczek spośród licznej bazy rozszerzeń, które można pobrać i zainstalować dzięki w celu rozszerzenia środowiska o kolejne funkcje.



Rysunek 2.6 Domyślne okno programu po uruchomieniu



Rysunek 2.7 Przykładowe okno ustawień sceny

2.4.2 Tworzenie obiektów

Program pozwala na modelowanie różnego rodzaju obiektów, oświetleń, efektów dźwiękowych i wizualnych, fizyk oraz wielu innych. Podczas pracy korzystałem z modelowania 3D, dodawania oświetlenia oraz fizyki sceny i obiektów.

Modelowane bryły mogą przyjmować różne kształty między innymi stożka, sześcianu, czy sfery. Wybrany obiekt wyświetlany jest na scenie oraz dodawany jest do jej listy elementów w oknie *stage*. Modele można przesuwać, obracać oraz ustawać wymiar w każdej z 3 osi. Można również nadać jej właściwości fizyczne, poprzez wybranie interesującej z listy. W przypadku symulacji bionicznej protezy były to fizyka ciała sztywnego oraz fizyka kolizji. Dodatkowo utworzyć własne właściwości fizyczne, w których znajduje się również możliwość wyboru koloru obiektu, któremu ją otrzyma. Oprócz modelowania zwykłych brył można również modelować bryły, które służą za źródło światła. Taki model oprócz podstawowych wymienionych własności może również oświetlać scenę, której parametry można zmieniać np. barwa światła, nasycenie, odległość, na którą świeci.

Generowana automatycznie zawiera podstawową fizykę grawitacji oraz podstawowe światło. Jednakże nie posiada ona fizycznej płaszczyzny, na której można by symulować własności fizyczne modeli. Aby ją utworzyć, należy wybrać z okna fizyki

Ground Plane. Dodatkowo można dodać lub zmienić fizykę sceny w celu symulacji innych warunków.

2.4.3 Możliwe opcje sterowania

Isaac Sim udostępnia rozmaite formy sterowania, dzięki którym każdy użytkownik powinien znaleźć coś dla siebie.

Pierwszym sposobem jest sterowanie poprzez skrypty i fragmenty kodu napisane w języku Python. Przy pomocy prostych skryptów można wpływać na obiekty w symulacji.

Drugim sposobem jest nawigacja przy pomocy rozszerzenia REPL (Read-Evaluate-Print loop). Jest to powłoka programistyczna pozwalająca na interpretację kawałków kodu oraz interaktywne sprawdzanie ich treści w trakcie symulacji. Przykładowymi takimi rozszerzeniami są IPython i Jupyter Notebook. Dzięki temu środowisko Isaac Sim jest tylko narzędziem do symulacji, a całość tworzenia oraz wykonywania kolejnych czynności odbywa się z poziomu napisanego i przygotowanego kodu. Należy zaznaczyć, że to rozszerzenie jest przeznaczone tylko dla wersji na systemie Linux.

Trzecim sposobem sterowania w środowisku są nieprzetworzone API USD. Jak zostało wspomniane, Isaac Sim jest częścią Nvidia Omniverse, które jest oparte na formacie USD (frameworku do obsługi danych 3D). Tworząc sceny oraz wszystkie właściwości dla symulacji, tworzone są równolegle skrypty, które są wczytywane i uruchamiane wraz z symulacją. Oznacza to, że można sterować symulacją poprzez uruchomienie domyślnej sceny i wgranie interesującego pliku do okna *Script Editor*. Po uruchomieniu symulacji wgrywany jest załączony skrypt i odtwarzana jest symulacja. Jednakże ten sposób sterowania wymaga dużej wiedzy, która może stanowić barierę dla niezaawansowanych użytkowników.

Czwartym sposobem sterowania jest Isaac Sim Core API. Jest to uproszczona wersja sterowania poprzez nieprzetworzone API USD wymienione w poprzednim akapicie. Jest to wygodny sposób sterowania dla początkujących. Isaac Sim posiada wbudowane podstawowe interfejsy API, które pozwalają na uproszczenie niektórych czynności, dzięki czemu nie trzeba pisać całości API.

Piątym sposobem sterowania jest kontrola symulacji poprzez OmniGraphy. OmniGraphy to środowisko programowania wizualnego oprogramowania Omniverse. Programowanie odbywa się poprzez łączenie ze sobą odpowiednich kafelków o konkretnych właściwościach. Jest to rozwiązanie, które pozwala w łatwy sposób manipulować systemem oraz importować do niego własne potrzebne funkcjonalności. OmniGraphy umożliwiają również korzystanie z ROS i ROS2, wykorzystanie różnych czujników, sterowników, urządzeń wejścia/wyjścia i interfejsu użytkownika zapewniając wydajność obliczeniową. Ich niewątpliwą zaletą jest prostota implementacji wynikającą z intuicyjnego graficznego interfejsu.

Na potrzeby symulacji bionicznej protezy zdecydowano wykorzystać ostatnią z metod, OmniGraphy, ze względu na prostotę implementacji, bogatą liczbę dostępnych przykładów oraz możliwość sterowania w czasie rzeczywistym. Sposób implementacji algorytmów został opisany w rozdziale Sterowanie.

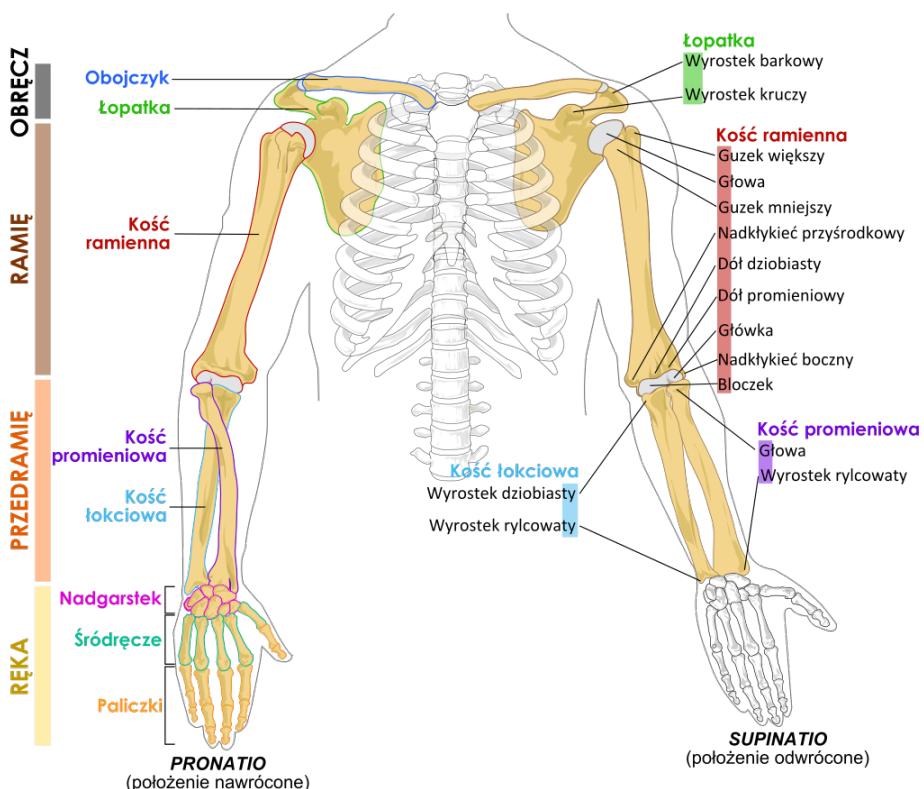
Rozdział 3

Bioniczna proteza

Praca ma na celu przybliżenie problematyki bionicznej protezy od stawu łokciowego do dłoni włącznie. Protezę można podzielić na dwie części. Pierwsza część wykonuje tak zwane makroruchy, za które odpowiadają łokieć oraz nadgarstek. Służą one ustawnieniu ręki do wykonania chwytu lub gestu. Aby zrealizować zamierzoną czynność np. złapać przedmiot, wykorzystuje się tak zwane mikroruchy wykonywane przez dłoń i palce. Mikroruchy są o wiele trudniejszym zagadnieniem i w związku z tym one były głównym zagadnieniem pracy, a symulacji została poddana proteza ręki.

3.1 Anatomia ludzkiej kończyny górnej

Poniższy podrozdział warto rozpocząć od rysunku 3.1 obrazującego część ludzkiego szkieletu, a dokładniej kończynę górną.

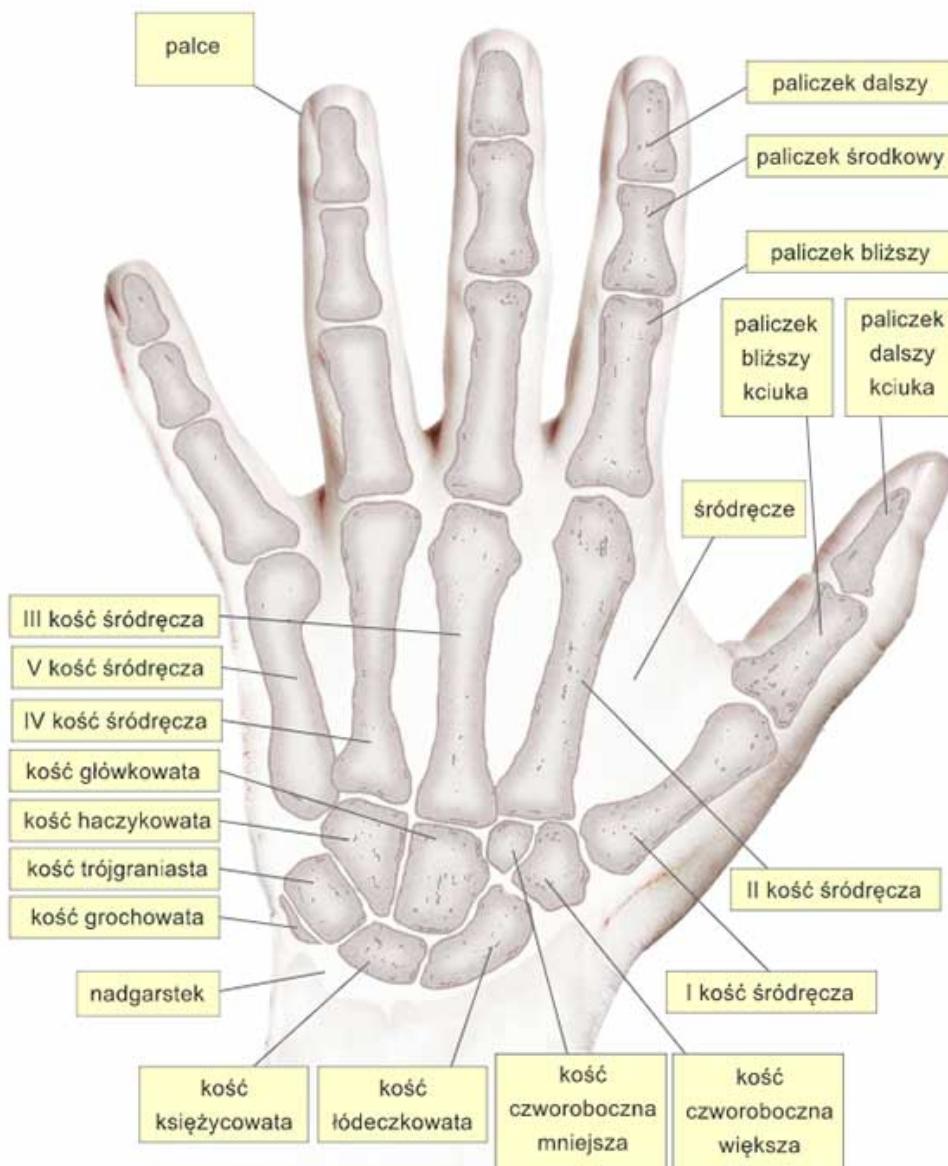


Rysunek 3.1 Schemat kończyny górnej [2]

Jak pokazano na rysunku 3.1 na kończynę górną składają się kolejno:

- Obręcz - obojczyk i łopatka,
- Ramię - kość ramienna,
- Przedramię - kość promieniowa oraz łokciowa,
- Ręka - nadgarstek, śródręcze i paliczki.

W kontekście tej pracy skupiono się na przedramieniu oraz samej ręce. Przedramię rozpoczyna się stawem łokciowym, następnie umiejscowione są kości promieniowa oraz łokciowa, które kończą się na nadgarstku. Jak napisano wyżej, przedramię pełni funkcje makroruchu. Za mikroruchy odpowiada dalsza część kończyny - ręka - którą tworzą nadgarstek, dłoń i palce. Na rysunku 3.2 pokazano anatomię ręki.



Rysunek 3.2 Anatomia ręki [3]

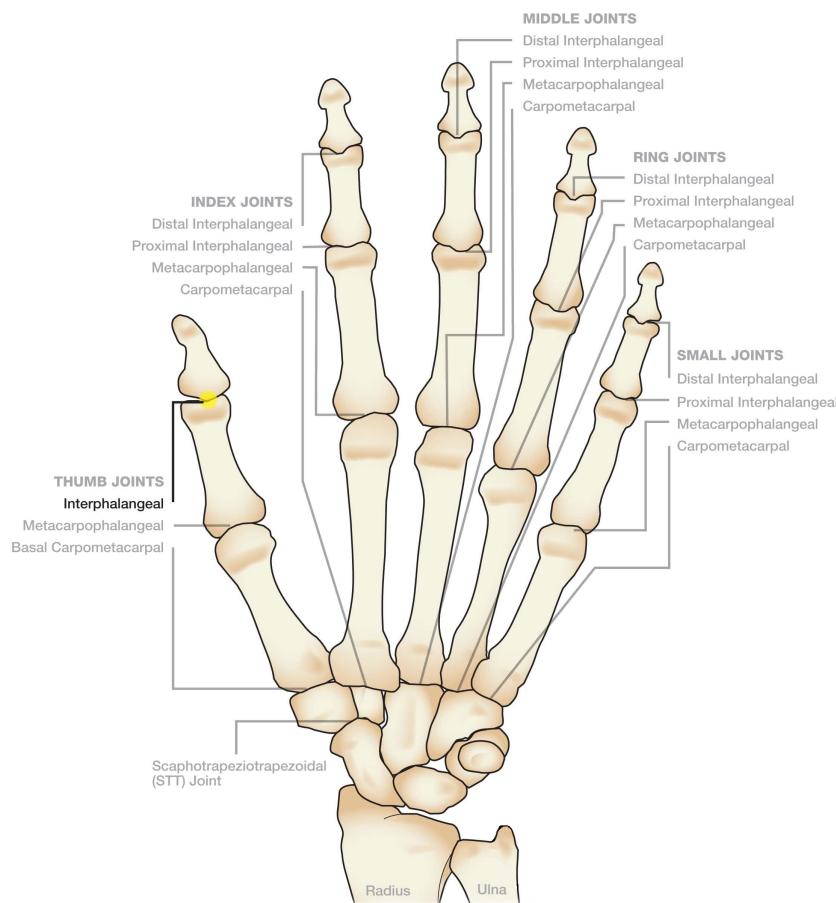
Dłoń tworzą kości śródłędzca (ang. *Metacarpal phalanx*) wraz ze stawami i mięśniami. Na palce składają się stawy, mięśnie oraz kolejne paliczki:

- Paliczek bliższy (ang. *Proximal phalanx*),
- Paliczek środkowy (ang. *Middle phalanx*), który nie występuje w przypadku kciuka,
- Paliczek dalszy (ang. *Distal phalanx*).

Stawy palców zostały przedstawione na rysunku 3.3. Mały, serdeczny, środkowy oraz wskazujący palec posiadają po 4 stawy (od nadgarstka w stronę końca palca):

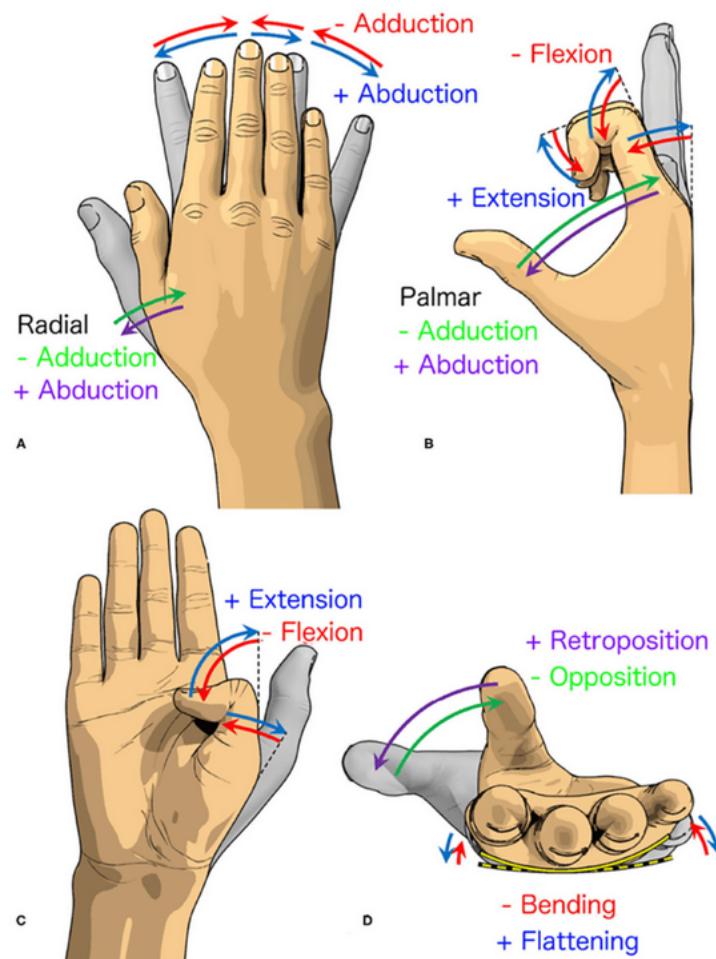
1. staw nadgarstkowo-śródłędzczny (ang. *Carpometacarpal Joint (CMC)*),
2. staw śródłędzno-paliczkowy (ang. *Metacarpophalangeal Joint (MCP)*),
3. staw międzymaliczkowy bliższy (ang. *Proximal Interphalangeal (PIP)*),
4. staw międzymaliczkowy dalszy (ang. *Distal Interphalangeal (DIP)*),

Kciuk ze względu na swoją budowę nie posiada rozgraniczenia na stawy międzymaliczkowy bliższy i dalszy. Zastępuje je **staw międzymaliczkowy** (ang. *Interphalangeal Joint (IP)*). Pozostałe dwa stawy są takie same jak w przypadku pozostałych palców. Czasami staw nadgarstkowo-śródłędzny kciuka jest opisywany jako **TMC** (ang. *Trapeziometacarpal Joint*).

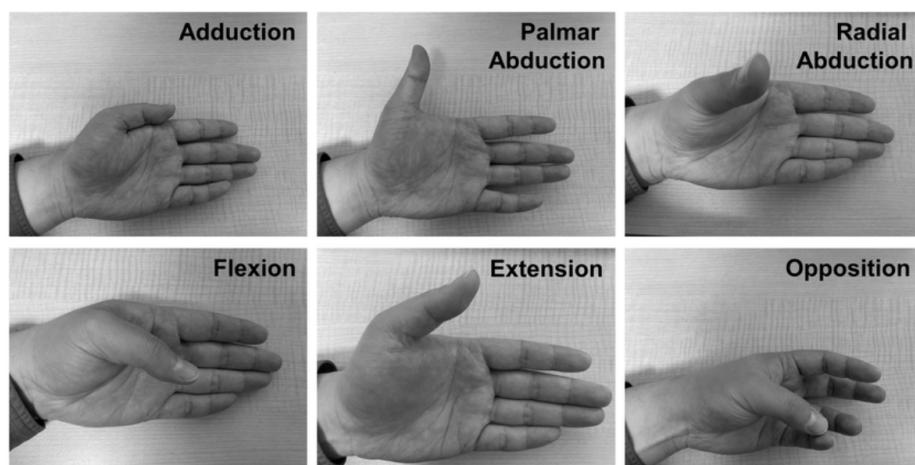


Rysunek 3.3 Stawy palców [4]

Ludzka ręka może realizować antagonistyczne ruchy w płaszczyźnie dłoni, takie jak odwodzenie/przywodzenie (ang. *abduction/adduction*), prostowanie/zginanie (ang. *extension/flexion*), unoszenie/obniżanie (ang. *bending/flattening*) oraz nawracanie/odwracanie (ang. *opposition/retroposition*). Ruchy te zaprezentowano na rysunkach 3.4 i 3.5.



Rysunek 3.4 Przedstawienie ruchów w obrębie ręki [5]



Rysunek 3.5 Przedstawienie ruchów kciuka [5]

3.2 Struktura kinematyczna ręki

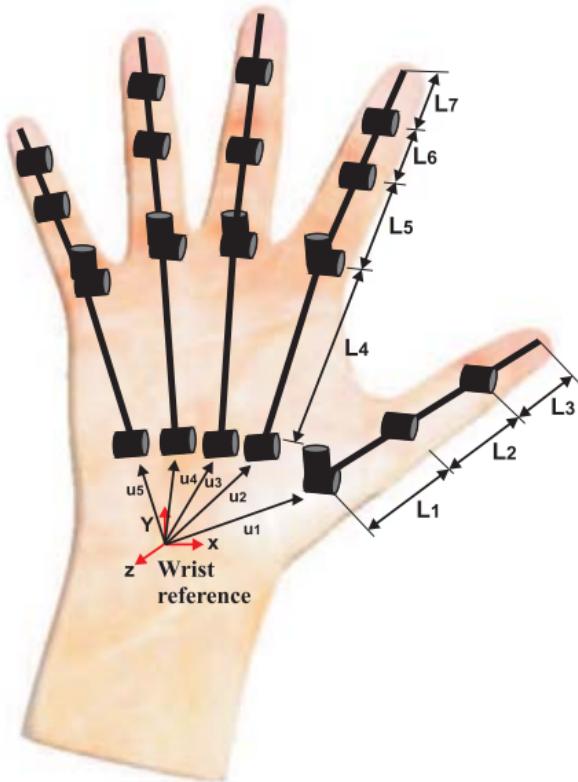
Poniższy podrozdział został opracowany na podstawie materiału konferencyjnego „Efficient human hand kinematics for manipulation tasks” autorów Salvador Cobos, Manuel Ferre, M.A. Sanchez Uran, Javier Ortego oraz Cesar Pena [6].

Za realistyczny kinematyczny model ręki uznaje się model o 24 stopniach swobody (DoF) oparty na ludzkim szkielecie. Jest to złoty środek między złożonością i realizmem. Model oparty jest na 19 ramionach reprezentujących ludzkie kości i przegubach odwzorowujących stawy. Poniżej przedstawiono dwie kinematyki, jedną dla kciuka, a drugą dla pozostałych palców, gdyż ich zasada działania jest zbliżona i główną różnicę są długości paliczków. Mały, serdeczny, środkowy oraz wskazujący palec mają po 5 DoF i tworzone są przez 4 ramiona (kości) i 4 przeguby (stawy). Miejsca zgięć są modelowane przez przeguby o odpowiedniej liczbie stopni swobody. Staw nadgarstkowo-śródręczny (CMC) posiada 1 DoF, staw śródręczno-paliczkowy (MCP) ma 2 DoF, stawy międzypaliczkowy bliższy (PIP) i międzypaliczkowy dalszy (DIP) mają po 1 DoF. Kciuk modelowany jest poprzez 3 ramiona (kości), 3 przeguby (stawy) i posiada 4 DoF. Staw nadgarstkowo-śródręczny kciuka (TMC) ma 2 DoF, a stawy śródręczno-paliczkowe (MCP) i międzypaliczkowy (IP) mają po 1 DoF.

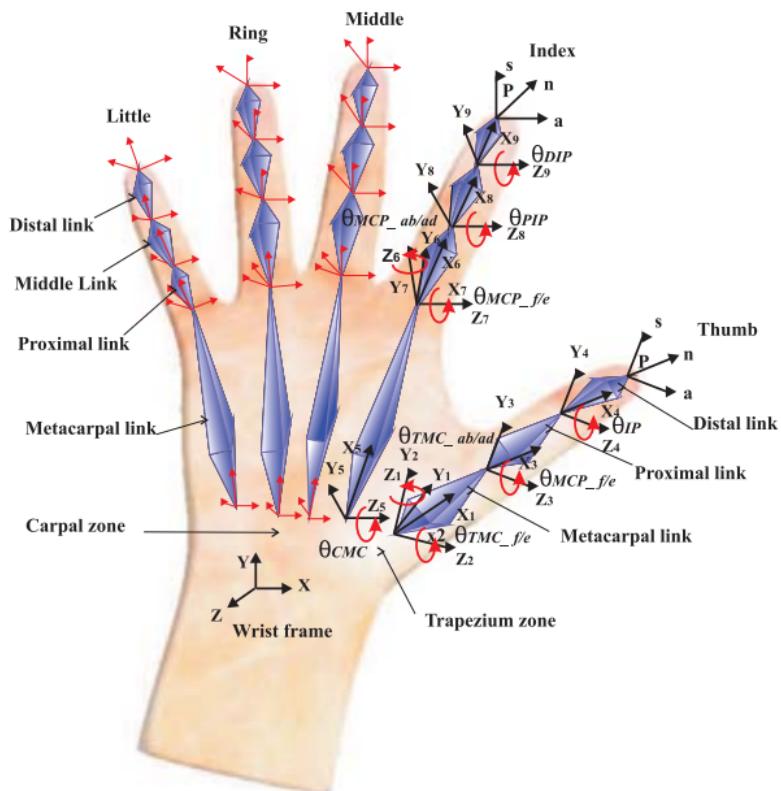
Na rysunkach 3.6 i 3.7 pokazano reprezentacje modelu ręki, który cechują:

- obsługa 4 stopni swobody kciuka,
- uruchamianie przegub MCP na podstawie przegubu CMC,
- odwodzenie/przywodzenie przegubu MCP ma pierwszeństwo względem zgięcia przegubu, co pozwala na dokładniejsze symulowanie przemieszczenia,
- CMC pozwala na deformację dłoni w przypadku chwytu okrągłych obiektów

Dzięki zastosowaniu modelu opartego na ludzkiej anatomii oraz uwzględnieniu powyższych cech uzyskało się model odwzorowujący w dużym stopni realistyczne ruchy i gesty. Na jego podstawie przedstawiono kinematyki kciuka i palców.



Rysunek 3.6 Model ręki o 24 DoF reprezentujący przeguby i ramiona, nałożony na ludzką rękę [6]



Rysunek 3.7 Model ręki o 24 DoF reprezentujący matematyczną interpretację, nałożony na ludzką rękę [6]

3.2.1 Kinematyka Prosta

Kinematyka dloni opisuje położenia i orientacje w przestrzeni końcówek palców względem układu odniesienia, jakim jest nadgarstek (rysunek 3.7) i pozwala je odwzorować w trakcie symulacji. Tabela 3.1 przedstawia parametry Denavita-Hartenberga (D-H) dla modelu palców.

Tabela 3.1 Parametry D-H dla palców

Przegub	kąt θ_i	długość d_i	długość a_i	kąt α_i
1	θ_{CMC}	0	L_4	$\pi/2$
2	$\theta_{MCPp/o}$	0	0	$-\pi/2$
3	$\theta_{MCPz/r}$	0	L_5	0
4	θ_{PIP}	0	L_6	0
5	θ_{DIP}	0	L_7	0

Jak napisano wcześniej wszystkie palce, z wyjątkiem kciuka, składają się z 4 kości, a ich długości przyporządkowane są długościom łączeń w łańcuchach kinematycznych. Palce wykonują ruchy antagonistyczne tzn. **przywodzenie/odwodzenie (p/o)** oraz **zginanie/prostowanie (z/pr)**. W modelu ręki przegub MCP ma 2 DoF, czyli realizuje te ruchy. Pozostałe przeguby (CMC, PIP i DIP) mogą się tylko zginać i prostować w stosunku do powierzchni dloni. Równania (3.1) i (3.2) przedstawiają końcowe równania opisujące przekształcenia układu współrzędnych z nadgarstka do końca palca.

$$p_i = {}^0_{-1}T(u_i){}^0_5T(\theta_j)i \quad (3.1)$$

$$p_i = {}^0_{-1}T(u_i){}^0_1T(\theta_{CMC})iT(\theta_{MCPp/o})iT(\theta_{MCPz/pr})iT(\theta_{PIP})iT(\theta_{DIP})i \quad (3.2)$$

Oznaczenia dla powyższych równań są następujące:

- p_i - przedstawienie pozycji i orientacji palca
- ${}^0_{-1}T(u_i)i$ - reprezentacja odległości między nadgarstkami, a punktem referencyjnym palca
- ${}^0_5T(\theta_j)i$ - macierz transformacji pomiędzy i-tym punktem referencyjnym, a jego palcem. Macierz zawiera w sobie macierze biorące udział w przekształcaniu dla każdego stawu (θ_{CMC} , $\theta_{MCPp/o}$, $\theta_{MCPz/pr}$, θ_{PIP} i θ_{DIP})
- i - oznaczenie palca (wskazujący, środkowy, serdeczny, mały)
- j - oznaczenie przegubu i jego zakresu ruchowego (θ_{CMC} , $\theta_{MCPp/o}$, $\theta_{MCPz/pr}$, θ_{PIP} i θ_{DIP})

Dla kciuka musiał powstać drugi model kinematyki ze względu na występowanie jednego paliczka mniej. Kciuk modelują 4 kości (czworoboczna, śródręczna, paliczki bliższy i dalszy) oraz 3 stawy (TMC, MCP i IP). W przypadku tego palca to **staw nadgarstkowo-śródręczny (TMC)** posiada 2 DoF, czyli realizuje **przywodzenie/odwodzenie (p/o)** oraz **zginanie/prostowanie (z/pr)**, a pozostałe stawy posiadają tylko 1 DoF. W tabeli 3.2 przedstawiono parametry Denavita-Hartenberga (D-H) dla modelu kciuka, a równania (3.3) i (3.4) prezentują jego kinematykę.

Tabela 3.2 Parametry D-H dla kciuka

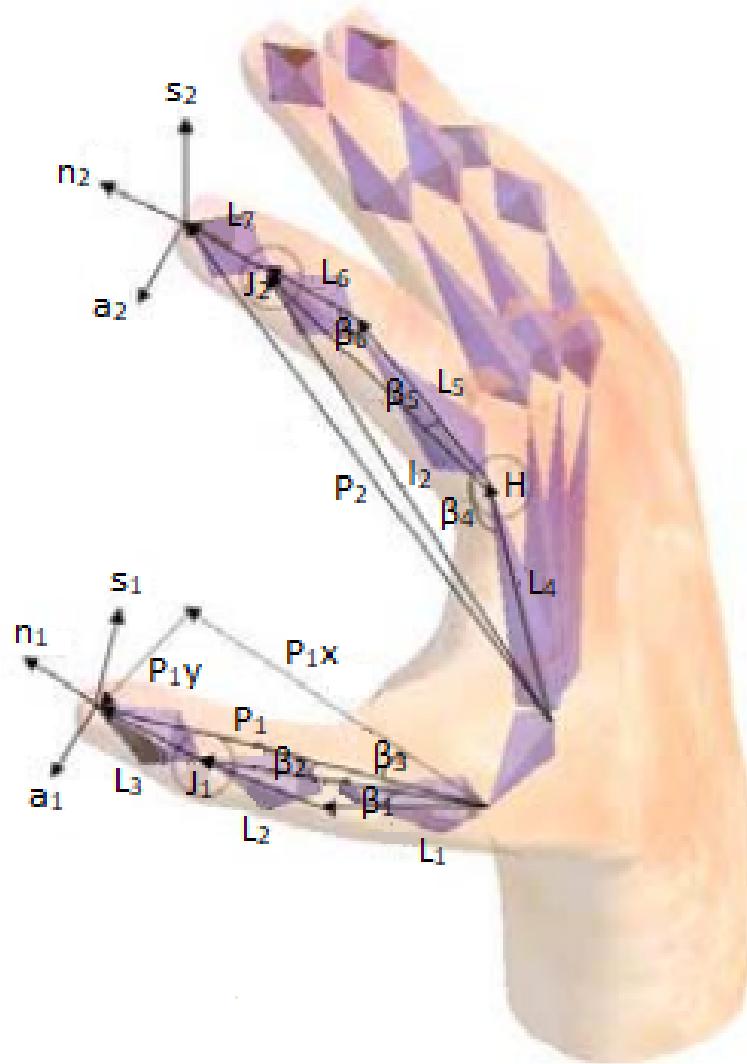
Przegub	kąt θ_i	długość d_i	długość a_i	kąt α_i
1	$\theta_{TMCp/o}$	0	0	$\pi/2$
2	$\theta_{TMCz/pr}$	0	L_1	0
3	$\theta_{MCPz/pr}$	0	L_2	0
4	θ_{IP}	0	L_3	0

$$p_{kciuk} = {}^0_{-1} T(u_{kciuk}) {}^0_4 T(\theta_j) kciuk \quad (3.3)$$

$$p_{kciuk} = {}^0_{-1} T(u_{kciuk}) {}^0_1 T(\theta_{kciuk MCPp/o}) {}^1_2 T(\theta_{TMCz/pr}) {}^2_3 T(\theta_{MCP}) {}^3_4 T(\theta_{IP}) \quad (3.4)$$

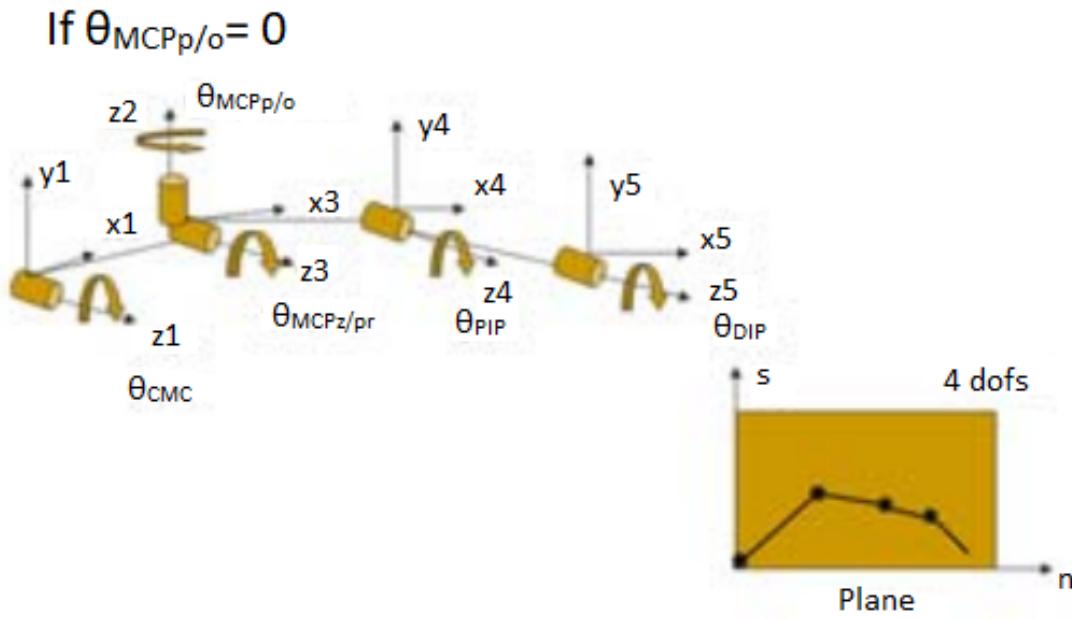
Oznaczenia są analogiczne jak dla równania kinematyki palców.

3.2.2 Kinematyka odwrotna



Rysunek 3.8 Kinematyka odwrotna [6]

Kinematyka odwrotna została wyliczona za pomocą metody geometrycznej ze stosunku trójkątów (rysunek 3.8) i rozwiązuje problem płaszczyzn, w których wykonywany jest ruch. W zależności od układu odniesienia w danym przegubie ruch może być wykonywany w wartościach dodatnich i ujemnych. Wynika to ze zginania, prostowania, przywodzenia, odwodzenia oraz przypadków redundantnych, które są generowane, gdy wartość przywodzenia i odwodzenia stawu MCP wynosi 0. Istnieją również bardziej złożone przypadki, gdy wartość odwodzenia będzie zbyt duża. Łanuch kinematyczny dla modelu palców jest najbardziej stabilny wtedy, gdy występują w nim same przywodzenia.



Rysunek 3.9 Przypadek redundantny modelu palców, gdy $\theta_{MCPp/o} = 0$ [6]

Model kinematyczny ręki jest modelem redundantnym, oznacza to że liczba stopni swobody modelu jest większa niż przestrzeń, w której się porusza [32]. By rozwiązać problem nadsterowności można stosować różne metody iteracyjne. W tym przypadku zastosowano metodę Newtona-Raphsona opartą na jakobianach, którą wspiera się ograniczeniami mającymi na celu zwiększenie zbieżności rozwiązań.

Aby wyliczyć kinematykę odwrotną dla modelu palców należało najpierw wyznaczyć orientację i pozycję koniuszka palca, wobec którego liczona była kinematyka. Pierwszy stopień swobody dla stawu CMC reprezentowany jest poprzez (3.5).

$$\theta_{CMC} = \arctan\left(\frac{a_{2y}}{a_{2x}}\right) \quad (3.5)$$

Tak samo wyliczany jest staw MCPp/o (3.6).

$$\theta_{MCPp/o} = \arctan\left(\frac{a_{2x}}{a_{2z} \cos(\theta_{CMC})}\right) \quad (3.6)$$

Zależność między stawami DIP, PIP oraz MCP(z/pr) opisuje zależność geometryczną, przedstawiona na rysunku 3.8. Wektor H (składowe H_x , H_y , H_z) wyliczony jest przy pomocy długości kości śródrcza oraz informacji ze stawu CMC (3.7).

$$H_x = L_4 c \cos(\theta_{CMC}); H_y = L_4 \sin(\theta_{CMC}); H_z = 0 \quad (3.7)$$

Punkt J2 określono jako różnicę odległości końca palca względem początku (P2) i iloczynu długości kości paliczka dalszego oraz wektora n7 (3.8).

$$J_2 = P_2 - [L_7 \cdot n_7] \quad (3.8)$$

Znając J2 oraz H wyliczono wektory u, r2, r3 (3.9) oraz kąty β_4 i β_5 (3.10).

$$u = J_2 - H; \quad r_2 = \text{norm}(J_2); \quad r_3 = \text{norm}(u); \quad (3.9)$$

$$\beta_4 = \arccos\left(\frac{L_4^2 + r_3^2 - r_2^2}{2L_4r_3}\right); \quad \beta_5 = \arccos\left(\frac{L_5^2 + r_3^2 - L_6^2}{2L_5r_3}\right) \quad (3.10)$$

Kąty β_4 i β_5 pozwalają określić zgięcie i prostowanie przegubu MCP, co pokazuje (3.11).

$$\theta_{MCPz/pr} = \pi - \beta_5 - \beta_4 \quad (3.11)$$

Natomiast prostowanie i przeprost stawu MCP przedstawia (3.12).

$$\theta_{MCP(przeprost)} = \beta_4 - \pi - \beta_5 \quad (3.12)$$

Staw PIP reprezentuje (3.14), które otrzymano jako różnice π oraz wyliczonej w (3.13) β_6 :

$$\beta_6 = \arccos\left(\frac{L_6^2 + L_5^2 - r_3^2}{2L_6L_5}\right) \quad (3.13)$$

$$\theta_{PIP} = \pi - \beta_6 \quad (3.14)$$

Ostatni przegub DIP wyznacza (3.17), które jest funkcją $\arctan 2$ z wyliczonych k_1 i k_2 ((3.15) i (3.16)).

$$\begin{aligned} k_1 = & ((c_2c_4c_3c_1 - c_4s_1s_3 - c_2s_4c_1s_3 - s_4c_3s_1)nx + \\ & + (c_2c_4c_3s_1 + c_4c_1s_3 - c_2s_4s_1s_3 + s_4c_3c_1)ny + \\ & + ((s_4s_3 - c_4c_3)s_2)nz \end{aligned} \quad (3.15)$$

$$\begin{aligned} k_2 = & (-s_4c_2c_1c_3 + s_4s_1s_3 - c_4c_1c_2s_3 - c_4c_3s_1)nx + \\ & + (-s_4c_2s_1c_3 - s_4c_1s_3 - c_4s_1c_2s_3 + c_4c_3c_1)ny + \\ & + ((c_4s_3 + s_4c_3)s_2)nz \end{aligned} \quad (3.16)$$

$$\theta_{DIP} = \arctan 2[k_1, k_2] \quad (3.17)$$

Podsumowując powyższe obliczenia i porządkując kolejność stawów, kinematyka odwrotna dla każdego stawu wygląda następująco ((3.18), (3.19), (3.20), (3.21) i (3.22)):

$$\theta_{CMC} = \arctan\left(\frac{a_{2y}}{a_{2x}}\right) \quad (3.18)$$

$$\theta_{MCPp/o} = \arctan\left(\frac{a_{2x}}{a_{2z} \cos(\theta_{CMC})}\right) \quad (3.19)$$

$$\theta_{MCPz/pr} = \pi - \beta_5 - \beta_4 \quad (3.20)$$

$$\theta_{PIP} = \pi - \beta_6 \quad (3.21)$$

$$\theta_{DIP} = \arctan 2[k_1, k_2] \quad (3.22)$$

Kinematyka odwrotna modelu kciuka różni się tak samo, jak proste zadanie kinematyki wspomniane wcześniej, ale proces jest podobny jak w przypadku modelu palca.

Równania (3.23) i (3.24) przedstawia przywodzenie/odwodzenie oraz zginanie/prostowanie stawu TMC.

$$\theta_{TMCp/o} = \arctan\left[\frac{P_{1y}}{P_{1x}}\right] \quad (3.23)$$

$$\theta_{TMCz/pr} = \arctan 2[x_6, x_5] \quad (3.24)$$

$$\begin{aligned} x_5 &= \frac{J_{1z}x_4 + J_{xy}x_3}{x_3^2 + x_4^2} \quad \text{i} \quad x_6 = \frac{J_{1z}x_3 - J_{xy}x_4}{x_3^2 + x_4^2} \\ x_4 &= L_2x_2J_{xy} = \sqrt{J_{1x}^2 + J_{1y}^2} \quad \text{i} \quad x_3 = (L_1 + (L_2x_1)) \end{aligned}$$

Staw MCP reprezentuje (3.25).

$$\theta_{MCP} = \arctan 2(x_2, x_1) \quad (3.25)$$

$$\begin{aligned} x1 &= \frac{r_1 - L_1^2 - L_2^2}{2L_1L_2} \quad \text{i} \quad x2 = \sqrt{1 - x_1^2} \\ r_1 &= \text{norm}(J_1J_1 = P_1 - (L_3 \cdot n_1)) \end{aligned}$$

Staw IP został opisany poprzez (3.26).

$$\theta_{IP} = \theta - \theta_{TMCz/pr} - \theta_{MCP} \quad (3.26)$$

$$\theta = \arctan 2[n_{1z}, \sqrt{n_{1x}^2 + n_{1y}^2}]$$

3.2.3 Ograniczenia ruchowe palców

Palce posiadają ograniczenia ruchowe, które dzielą się na międzypalcowe i wewnętrzpalcowe. Ograniczenia międzypalcowe mówią o oddziaływaniu na siebie kolejnych palców, czyli np. o maksymalnym wychyleniu jednego palca względem drugiego. Ograniczenia wewnętrzpalcowe przedstawiają zakres ruchowy pomiędzy paliczkami i ścięgnami. Dzięki tym więzom możemy wykonywać naturalne ruchy dynamiczne i statyczne oraz gestykulować w kontrolowany sposób. Zakres motoryki ręki zależy jednak od wielu czynników między innymi budowy, długości kości, mięśni czy innych zagadnień z zakresu biomechaniki [33] [34]. W trakcie poruszania palcami wykonywane są ruchy aktywne oraz pasywne (bierne). Ruch aktywny jest powodowany mięśniami i ścięgnami, natomiast ruch pasywny jest powodowany przez czynnik zewnętrzny np. dotyk rehabilitanta i nie jest wynikiem wysiłku. W tabeli 3.3 przedstawiono zakresy ruchowe stawów każdego z palców wyrażone w stopniach [°], które uzyskano przy pomocy specjalnej rękawicy wyposażonej w czujniki - *Cyberglove* [6].

Tabela 3.3 Zakres ruchowy stawów każdego z palców [6]

Palec	Zginanie	Prostowanie	Przywodzenie/ Odwodzenie
Kciuk			
Łódeczkowo-czworoboczny (STM)	0°	0°	0°
Nadgarstkowo-śródręczny kciuka (TMC)	50°- 90°	15°	45°- 60°
Śródręczno-paliczkowe(MCP)	75°- 80°	0°	0°
Międzypaliczkowy(IP)	75°- 80°	5°- 10°	0°
Wskazujący			
Nadgarstkowo-śródręczny (CMC)	5°	0°	0°
Śródręczno-paliczkowy (MCP)	90°	30°- 40°	60°
Międzypaliczkowy bliższy(PIP)	110°	0°	0°
Międzypaliczkowy dalszy (DIP)	80°- 90°	5°	0°
Środkowy			
Nadgarstkowo-śródręczny (CMC)	5°	0°	0°
Śródręczno-paliczkowy (MCP)	90°	30°- 40°	45°
Międzypaliczkowy bliższy(PIP)	110°	0°	0°
Międzypaliczkowy dalszy (DIP)	80°- 90°	5°	0°
Serdeczny			
Nadgarstkowo-śródręczny (CMC)	10°	0°	0°
Śródręczno-paliczkowy (MCP)	90°	30°- 40°	45°
Międzypaliczkowy bliższy(PIP)	120°	0°	0°
Międzypaliczkowy dalszy (DIP)	80°- 90°	5°	0°
Mały			
Nadgarstkowo-śródręczny (CMC)	15°	0°	0°
Śródręczno-paliczkowy (MCP)	90°	30°- 40°	50°
Międzypaliczkowy bliższy(PIP)	135°	0°	0°
Międzypaliczkowy dalszy (DIP)	90°	5°	0°

W tabeli 3.3 zostały uwzględnione ograniczenia statyczne palców oraz zdefiniowane nowe ograniczenia międzypalcowe i wewnętrzpalcowe mające na celu poprawienie i urealnienie motorki modeli palców i kciuka. Ograniczenia międzypalcowe poprawiły głównie wpływ ściegien środkowego, serdecznego i małego palca. Natomiast ograniczenia wewnętrzpalcowe poprawiły śledzenie trajektorii palców podczas chwytów kulistych i walcowatych. Na ich podstawie zostały opracowane kolejne dwa modele matematyczne, które upraszczają równania ręki dla tych ruchów [6].

Podsumowując ten podrozdział, opracowano rzeczywisty model kinematyczny ręki, który umożliwia rozwiązywanie zadania prostego i odwrotnego kinematyki. Zostały również wyznaczone zakresy ruchowe stawów. Dzięki zaimplementowaniu mechanizmów na bazie ludzkiej anatomii model pozwala na porównanie rzeczywistej ręki względem projektowanej protezy.

3.3 Znaczenie wymiarów ręki oraz predyspozycji człowieka względem ruchliwości i siły chwytu

Wcześniej wspomniano, że model kinematyczny ręki zależy między innymi od dłużności kości. Istnieje wiele badań na temat klasyfikacji długości [35] rąk oraz ich wpływu na predyspozycje, jak i siłę chwytu [33] [34]. Badania udowodniły, że średnio mężczyźni mają większe ręce, co wynika z biologii. We wszystkich opracowaniach wyniki są podobne, a różnice między nimi mają małe wartości. Wynika to z anatomii próbki, na której były przeprowadzone badania. Jednakże wszystkie opracowania potwierdzają, że siła zacisku dłoni, kciuka oraz palców jest proporcjonalna do ich wielkości. Wraz ze wzrostem długości kości palców rośnie siła chwytu, a najbardziej znaczące są rozstaw palców, rozstaw całej ręki oraz długość palca wskazującego. Te parametry stanowią podstawę do określenia predyspozycji do silnego chwytu, jednakże ważne są również zdolności indywidualne jednostki oraz stan, w jakim jest ręka. Sportowcy uprawiający koszykówkę, piłkę ręczną, judo czy inne sporty chwytne wykazują większą sprawność i siłę chwytu w porównaniu do osób nieuprawiających sportów. Porównując zdolności chwytne ludzi po udarze potwierdzono, że kończyna chora ma siłę ścisku mniejszą nawet o połowę wartości względem kończyny zdrowej. Oznacza to, że oprócz budowy anatomicznej ważną rolę odgrywa zdrowie [33].

3.4 Bioniczna proteza w porównaniu do ludzkiej ręki

Zazwyczaj tworząc antropomorficzną bioniczną protezę zakłada się, że ta nie będzie się wyróżniać od ludzkiego ciała. Będzie niezauważalna dla użytkownika i ograniczy jego dyskomfort. Dlatego przed inżynierami stoi zadanie opracowania modelu, który będzie wytrzymały i jak najbardziej sprawny, zachowując podobną masę, wielkość i czucie co ludzka ręka. W tym podrozdziale zostało omówione porównanie bionicznych protez względem ręki, które podsumowano w pracy „Performance Characteristics of Anthropomorphic Prosthetic Hands” [36] oraz porównano jej wyniki z bieżącym rozwiązaniem Bebionic Hand EQD [12].

Ludzka ręka waży około 400 gramów, jednakże ze względu na mocowanie protezy oraz brak naturalnego szkieletu, mięśni i ścięgien protezy o takiej masie uznawane są za zbyt ciężkie, a według użytkowników jest to jeden z dominujących aspektów. Tworzone rozwiązania muszą zatem ważyć mniej niż 400 gramów jako całość, gdzie na tę masę składają się między innymi materiał, z którego jest wykonana proteza, masa czujników, waga napędów i zasilania. Jednakże obecne rozwiązania pokazują, że proteza ważącą więcej, bo około 616 gramów znajduje swoje zastosowanie w rzeczywistości, dzięki odpowiedniemu sposobie montażu oraz faktu, iż ludzkie ciało z czasem przyzwyczaja się do obciążenia. W przypadku wspomnianej protezy po 3-6 miesiącach człowiek jest w pełni zintegrowany z protezą i obsługuje ją jak własną rękę.

Kolejnym aspektem jest długość ręki. Każdy człowiek jest inny i tak samo wymiary rąk są różne. Jednakże jak zostało wspomniane wyżej, istnieją uśrednione wielkości, na podstawie których producenci tworzą swoje protezy. Zgodnie z atlasem miar [35] średnia długość ręki, od nadgarstka do końca palca środkowego, to 15,9 cm, a szerokość ręki, od końca dłoni przy małym palcu do końca szerokości kciuka leżącego wzdłuż dłoni, wynosi 9,25 cm. Jednakże producenci często tworzą swoje rozwiązania w różnej wielkości. Na rynku istnieje duży wybór protez - od większych po mniejsze. Dzięki temu każdy, zarówno dzieci, jak i dorosli, znajdują coś dla siebie.

Tworząc protezę, czyli narzędzie, które ma być wykorzystywane codziennie, zakłada się, że będzie to narzędzie wytrzymałe realizujące powtarzalne ruchy. Ludzka dominująca ręka wykonuje około od 2500 do 3000 ruchów chwytnych w ciągu 8 godzin. Badania przeprowadzone w 2005 roku przedstawiły, że proteza powinna wytrzymać minimum 300 000 ruchów, co daje niecały rok pracy, zanim układy elektromechaniczne ulegną uszkodzeniu. Dzięki rozwojowi technologii i stosowaniu się do zaleceń producenta życie protezy zakładane jest na przynajmniej 5 lat. Wraz z postępem rozwinięły się również zdolności chwytnie protez, które umożliwiają chwytność zbliżoną do ludzkiej ręki, a siła chwytu działa na tej samej zasadzie tzn. jest proporcjonalna do wielkości ręki/protezy. Jednak ze względu na możliwość przypadkowego uszkodzenia obiektu lub obsługi specyficznych przedmiotów jak np. ekran dotykowy smartfona, firmy wykorzystują specjalne materiały na palcach dłoni lub całe rękawiczki, które nakładane są na protezę [37].

3.5 Analiza wymagań ruchowych protezy

Ludzka ręka wykonuje dziennie tysiące ruchów w różnych konfiguracjach i tego oczekujemy od protez. Jednakże ze względu na ich ogólną gamę ruchów oraz skomplikowanie stworzenie uniwersalnej, posiadającej wszystkie możliwości ludzkiej dłoni protezy jest na obecne warunki zadaniem niemożliwym. Wykorzystując klasyfikację chwytów, modele matematyczne oraz różnorakie rozwiązania inżynieryjne np. symulacje, stworzono protezy, które mogą realizować dużą część codziennych chwytów i gestów [12]. Są to między innymi otwarta dłoń, zaciśnięta pięść, chwyty pęsetowy, hakowy, czy kulisty. Jednakże, żeby móc je opracować najpierw trzeba było poznać matematykę stojącą za nimi, ich cel, któremu służą oraz ich funkcjonalności opisywane między innymi przez taksonomię chwytów. Taka klasyfikacja została opracowana w artykule „The GRASP Taxonomy of Human Grasp Types” [7] autorów Thomas Feix, Javier Romero, Heinz-Bodo Schmiedmayer, Aaron M. Dollar i Danica Kragic, na podstawie którego opisano ten podrozdział.

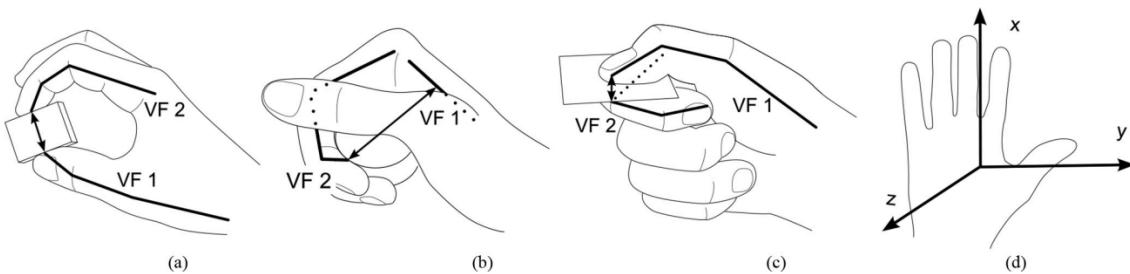
Każdy ruch można rozpatrzyć w 4 kategoriach:

1. Charakter chwytu: silny, pośredni i precyzyjny
2. Płaszczyzna chwytu
3. Liczba wirtualnych palców (ang *Virtual Finger*)
4. Ułożenie kciuka

3.5.1 Klasyfikacja chwytu ze względu na jego charakter

Wszystkie wykonywane ruchy ręką mają swój określony cel, który określany jest poprzez zapotrzebowanie na siłę i precyzję. Charakter siłowy chwytu można tłumaczyć jako używanie dużej siły, która blokuje chwytyany obiekt. Dzięki temu przedmiot nie może wykonać samoistnie żadnego przemieszczenia ani rotacji, a aby go przemieścić należy poruszyć więcej niż ręką np. poruszyć ramieniem. Przeciwieństwem charakteru siłowego jest charakter precyzyjny, który pozwala na wykonywanie ruchów przy pomocy ręki, bez udziału pozostałych części kończyny górnej. Charakter pośredni jest połączeniem obydwu charakterów w mniej więcej podobnych proporcjach, co pozwala na dokładniejsze zróżnicowanie chwytów.

3.5.2 Klasyfikacja chwytu ze względu na jego płaszczyznę



Rysunek 3.10 Płaszczyzny chwytu: a) *Pad Opposition*, b) *Palm Opposition*, c) *Side Opposition*, d) Układ współrzędnych w odniesieniu do dłoni [7]

Każdy ruch dłoni występuje w jednej z trzech osi trójwymiarowego układu współrzędnych. Jej przypisanie zależy od kierunku dominującej siły, która działa na chwytyany obiekt.

- Na rysunku 3.10(a) pokazano tzw. *Pad Opposition* oznacza to, że siła jest generowana równolegle do powierzchni dłoni. Sytuacja pokazuje, że siła wywierana jest na obiekt między kciukiem i palcem wskazującym, czyli w przybliżeniu równolegle do dłoni (zgodnie z osią X pokazaną na rysunku 3.10(d))).
- Rysunek 3.10(b) obrazuje sytuację *Palm Opposition*, czyli kiedy siła jest generowana prostopadle do dłoni. W tym przypadku siła wywierana jest na obiekt między końcami palców małego, serdecznego, środkowego i wskazującego, a powierzchnią dłoni, czyli prostopadle do niej (zgodnie z osią Z pokazaną na rysunku 3.10(d))).
- Przykład (c) z rysunku 3.10 przedstawia *Side Opposition*. Siła jest generowana wzdłuż dłoni, co potwierdza ilustracja, na której siła wywierana jest na obiekt między kciukiem i palcem wskazującym, czyli wzdłuż do dłoni (zgodnie z osią Y pokazaną na rysunku 3.10(d))).

3.5.3 Klasyfikacja chwytu ze względu na liczbę wirtualnych palców *Virtual Finger*

Virtual finger to podział chwytów ze względu na liczbę palców ułożonych w ten sam sposób i działających z siłą w podobnym kierunku. Ten podział pozwala na uproszczenie chwytu jako ruchu danej liczby **wirtualnych palców (VF)**, zamiast obliczania wszystkich palców. Na rysunku 3.10(a) widać, że na przedmiot działają dwa wirtualne palce. Jeden tworzy kciuk (VF1), a drugi pozostałe palce (VF2). Na rysunku 3.10(b) również ma miejsce podobna sytuacja. Kciuk nie wpływa na obiekt i naciskają na niego jedynie dłoń (VF1) i końcówki pozostałych palców (VF2). Na rysunku 3.10(c) VF2 tworzy kciuk, a VF2 tworzą pozostałe palce.

3.5.4 Klasyfikacja chwytów ze względu na ułożenie kciuka

Kciuk jest przeciwny względem pozostałych palców, co umożliwia zaawansowane chwyty. Jak wspomniano w podrozdziale 3.2 staw nadgarstkowo-śródłęczny kciuka (TMC) może wykonywać ruchy odwodzenia i przywodzenia. Na tej podstawie można podzielić chwyty ze względu na stan, w jakim jest kciuk - kciuk odwiedziony (ang. *thumb abducted*) lub kciuk przywiedziony (ang. *thumb adducted*).

3.5.5 Połączona taksonomia chwytów

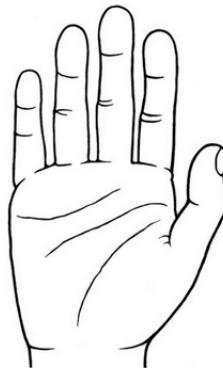
Połączenie powyższych klasyfikacji pozwala dostarczyć dużo informacji z jednego chwytu. Znane są jego charakter, ułożenie dloni względem dominującej siły oraz orientacja kciuka. Ponadto wiadomo jak można uprościć konkretny chwyt w odniesieniu do liczby wirtualnych palców. Na rysunku 3.11 przedstawiano połączoną taksonomię kilku chwytów.

Opp: VF:	Power					Intermediate			Precision										
	Palm		Pad			Side		Pad			Side								
	3-5	2-5	2	2-3	2-4	2-5	2	3	3-4	2	2-3	2-4	2-5	3					
Thumb Abducted	 1: Large Diameter	 2: Small Diameter	 3: Medium Wrap	 10: Power Disk	 11: Power Sphere		 31: Ring	 28: Sphere Finger	 18: Extension Type	 26: Sphere 4-Finger	 19: Distal Type	 23: Adduction Grip	 21: Tripod Variation		 9: Palmar Pinch	 8: Prismatic 2 Finger	 7: Prismatic 3 Finger	 6: Prismatic 4 Finger	 20: Writing Tripod
Thumb Adducted	 17: Index Finger Extension	 4: Adducted Thumb					 5: Light Tool			 15: Fixed Hook		 16: Lateral	 29: Stick	 25: Lateral Tripod		 22: Parallel Extension			
												 32: Ventral							

Rysunek 3.11 Skategoryzowane chwyty na podstawie połączonej klasyfikacji [7]

3.5.6 Zrelaksowna dłoń

Zrelaksowana dłoń jest jednym z ważniejszych jak nie najważniejszy gest. Jest to podstawowe ułożenie ręki, które wykonują mięśnie, gdy ręka nie pełni żadnej czynności. Jednocześnie o ustawienie stanowi podstawę dla kolejnych ruchów. W trakcie tego gestu wszystkie palce są delikatnie zgięte. Kciuk jest w pozycji bocznej i zgina się delikatnie w kierunku dłoni. Taką rękę przedstawia rysunek 3.12. W zagadnieniu protez jest to o tyle istotny gest, iż, poprawnie zrealizowany zwiększa poczucie komfortu u użytkownika. Ułatwia również korzystanie z innych gestów, gdyż po otrzymaniu sygnału pozwala od razu przejść do innych chwytów np. haka [12].



Rysunek 3.12 Rysunek zrelaksowanej dłoni [8]

3.5.7 Zaciśnięta pięść

Mocno zaciśnięta pięść jest gestem pokazującym bardzo dużą sprawność ręki. Poprawnie wykonana pozwala na zabezpieczenie ręki przed siłą przekazywaną i wywieraną na palce w trakcie kontaktu z obiektem. Chroni zarówno kości przed połamaniem, jak i stawy przed zwichnięciem, co nie oznacza, że tym nie może stać się krzywda. Poprawnie zaciśniętą pięść cechują schowane do środka dłoni złączone palce, gdzie paliczek dalszy próbuje dotknąć paliczek bliższy, a stawy śródręczno-paliczkowe są zgięte pod kątem około 90° (rysunek 3.13). Kciuk jest przywiedziony i paliczek dalszy naciska na paliczek środkowy środkowego palca. Na rysunku 3.14 zostało pokazane przemieszczenie palców w związku z siłą.



Rysunek 3.13 Rysunek pięści z różnych perspektyw [8]

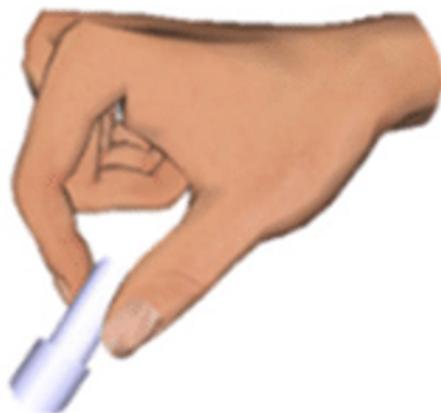


Rysunek 3.14 Rysunek pięści z przemieszczeniem powodowanym przez siłę [8]

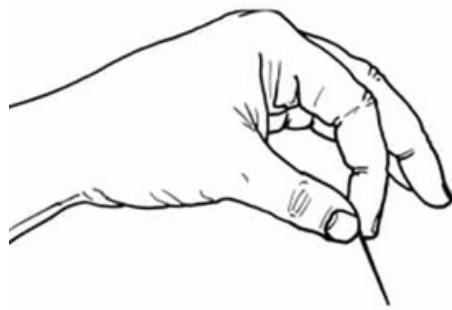
Ze względu na symulację bionicznej protezy ręki jest to istotny gest, który pokazuje możliwości ruchowe wszystkich przegubów, oraz sprawdza, czy dobrane długości ramion reprezentujących kości są odpowiednie i pozwalają na prawidłowe złożenie pięści.

3.5.8 Chwyt pęsetowy

Chwyt pęsetowy (ang. *pinch grasp*) jest chwytem z kategorii precyzyjnych, o orientacji dłoni typu Pad Opposition, podczas którego kciuk jest odwiedziony. W życiu codziennym używany jest między innymi w celu łapania małych rzeczy takich jak np. śrubki. Istnieje wiele rodzajów chwytu pęsetowego [38] [7], które odróżniają orientacja palców niebiorących udziału w chwycie oraz stopień wyprostowania palca chwytyjącego i kciuka. W tej pracy skupiono się na pokazaniu oraz symulowaniu chwytu pęsetowego o 2 VF, wykorzystującego palec serdeczny oraz kciuk. Podczas ruchu reszta palców jest delikatnie zaciśnięta do środka dłoni. Celem chwytu jest złapanie obiektu końcami palców. Na rysunkach 3.15 i 3.16 zostały przedstawione poglądowe chwyty.



Rysunek 3.15 Zdjęcie chwytu pęsetowego o zgiętych palcach dloni [9]

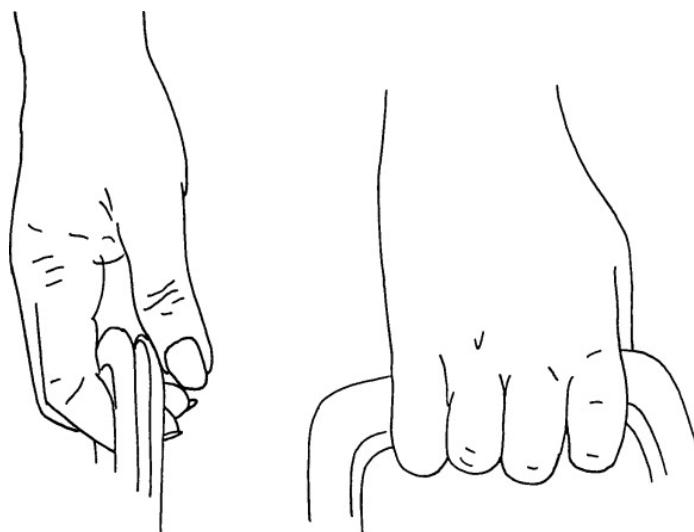


Rysunek 3.16 Rysunek chwytu pęsetowego o delikatnie zgiętych palcach [10]

Pod kątem symulacji gest sprawdza sterowalność kciuka w odwiedzeniu oraz jego współpracę z palcem wskazującym. Sprawdza on również bioniczną protezę pod kątem precyzyjnych ruchów.

3.5.9 Chwyt hakowy

Chwyt hakowy różni się w zależności od sposobu chwytania oraz rozmiarów obiektu. Ma on charakter siłowy, posiada 2-5VF i odbywa się w orientacji Palm Opposition. W trakcie codziennego funkcjonowania chwyt wykorzystywany jest między innymi do trzymania się poręczy, czy utrzymania torby z zakupami 3.17. Warto zauważyć, że chwyt jest zbliżony do chwytu cylindrycznego, który ma podobny charakter i często są uznawane za ten sam. Jednakże chwyt hakowy wyróżnia możliwość trzymania przedmiotu w samych palcach bez dotykania dłoni, jak na przykładzie *Fixed Hook* (15) z rysunku 3.11 *Fixed Hook*.



Rysunek 3.17 Rysunek chwytu hakowego [11]

Dla symulacji wybrano pewnych chwyt hakowy z przywiedzionym kciukiem. Na potrzeby symulacji stworzono walec średniej wielkości, który symbolizuje uchwyt w tramwaju. W porównaniu do poprzednich chwytów wykazuje on prace kciuka w innej płaszczyźnie. Pozwala również na podkreślenia chwytność palców podczas chwytu siłowego o orientacji Palm Opposition oraz na sprawdzenie zachowywania się modelu przy fizycznym chwycie przedmiotu.

3.5.10 Chwyt kulisty

Chwyt kulisty tak jak hakowy zależy od rozmiarów oraz sposobu chwytu okrągłego przedmiotu. Może być realizowany zarówno jak chwyt precyzyjny, siłowy, w różnych płaszczyznach oraz przy użyciu różnej liczby palców. Najtrudniejszymi przypadkami dla bionicznych protez są chwyty wymagające obsługi wszystkich palców, należących do chwytów z zakresu 5VF. Problematyczne jest obsłużenie 5 palców jednocześnie i poprawne ułożenie dloni względem chwytanego obiektu, tak by ten był pewnie złapany. Na rysunku 3.18 pokazano przykładowy chwyt.



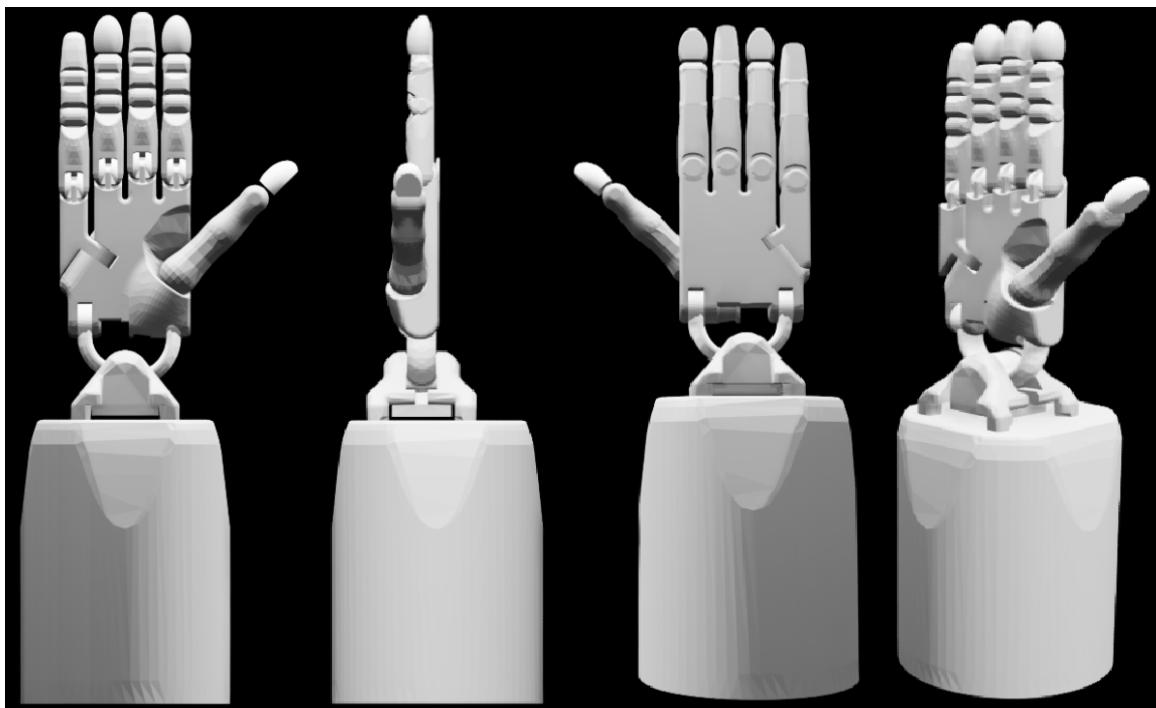
Rysunek 3.18 Rysunek chwytu kulistego [11]

Symulowanie chwytu o 5VF jest niezwykle cenne, gdyż pozwala na sprawdzenie jakości chwytu przez palce oraz wymodelowania powierzchni dłoni. Względem pochodzących symulacji wyróżnia się potrzebą rozszerzenia wszystkich palców, dostosowania powierzchni dłoni oraz odpowiedniego ułożenia kciuka, by ten stanowił kontrolę dla palców.

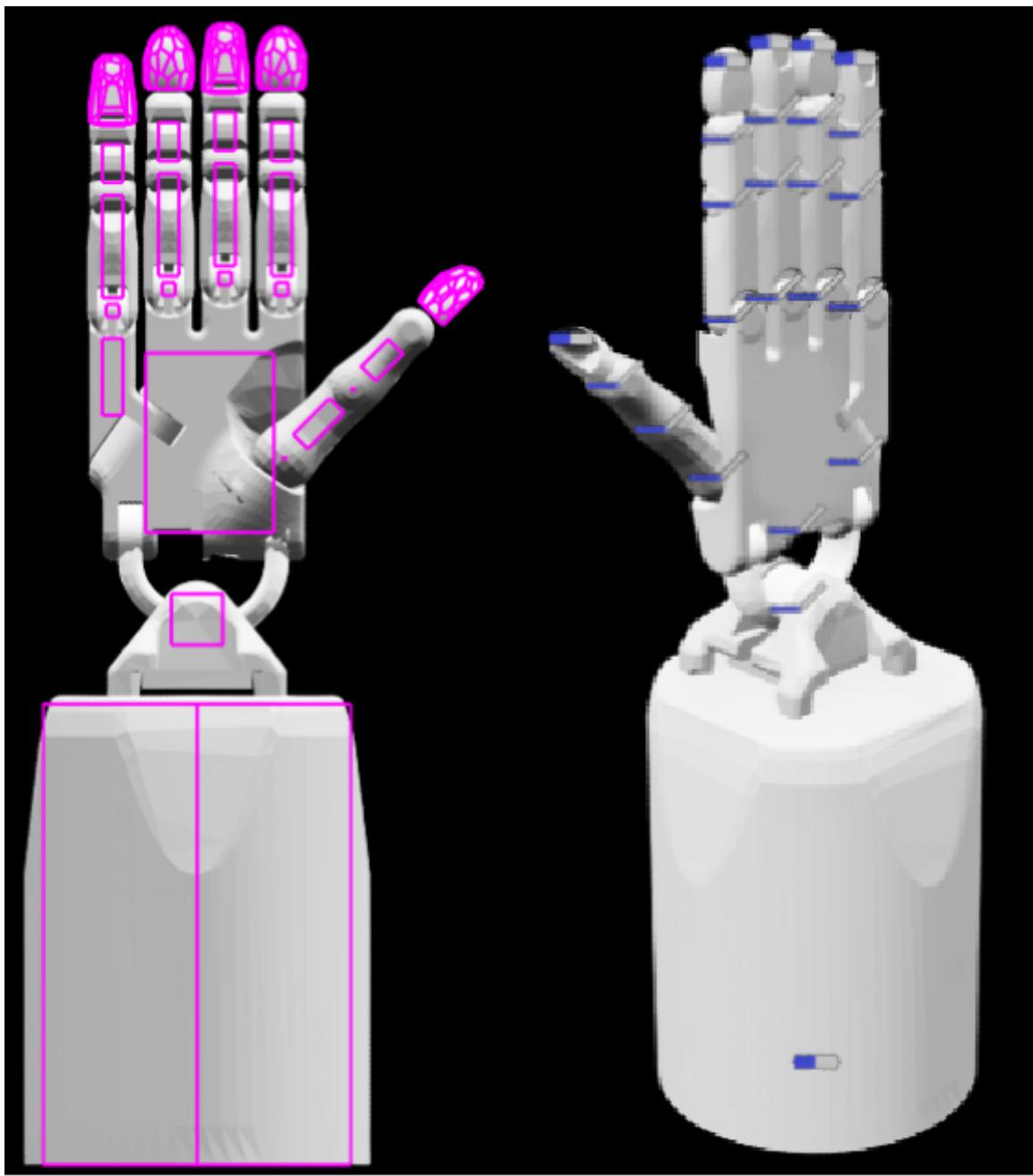
Rozdział 4

Model protezy

W ramach symulacji bionicznej protezy potrzebny był sprawdzony poprawny model, który gwarantuje prawidłowe zachowanie się w trakcie ruchu. W związku z tym wybrano popularny model robotycznej ręki Shadow Dexterous Hand w wersji Hand E (prawa ręka) opracowany i upubliczniony [39] przez firmę Shadow Robot na platformie GitHub [40]. Pomimo tego, że nie modeluje faktycznej protezy, ten model jest opracowany na podobieństwo ludzkiej ręki z uwzględnieniem anatomii, a jego budowa pozwala na zaawansowane ruchy małego palca i części dłoni, do której jest mocowany. Ze względu na architekturę, zakres ruchowy oraz format URDF, wymagany przez Isaac Sim, Shadow Dexterous Hand spełnia wymagania w celu przybliżenia problematyki symulacji bionicznej protezy. Pomimo swojej złożoności importowany model ma mniej stopni swobody (22 DoF) [13], niż wcześniej przedstawiona reprezentacja matematyczna rzeczywistej ręki (24DoF), ale zachowuje się podobnie do niej. Na rysunkach 4.1 i 4.2 pokazano zdjęcia modelu z różnych perspektyw oraz z włączoną wizualizacją siatki kolizji i oznaczonymi przegubami.

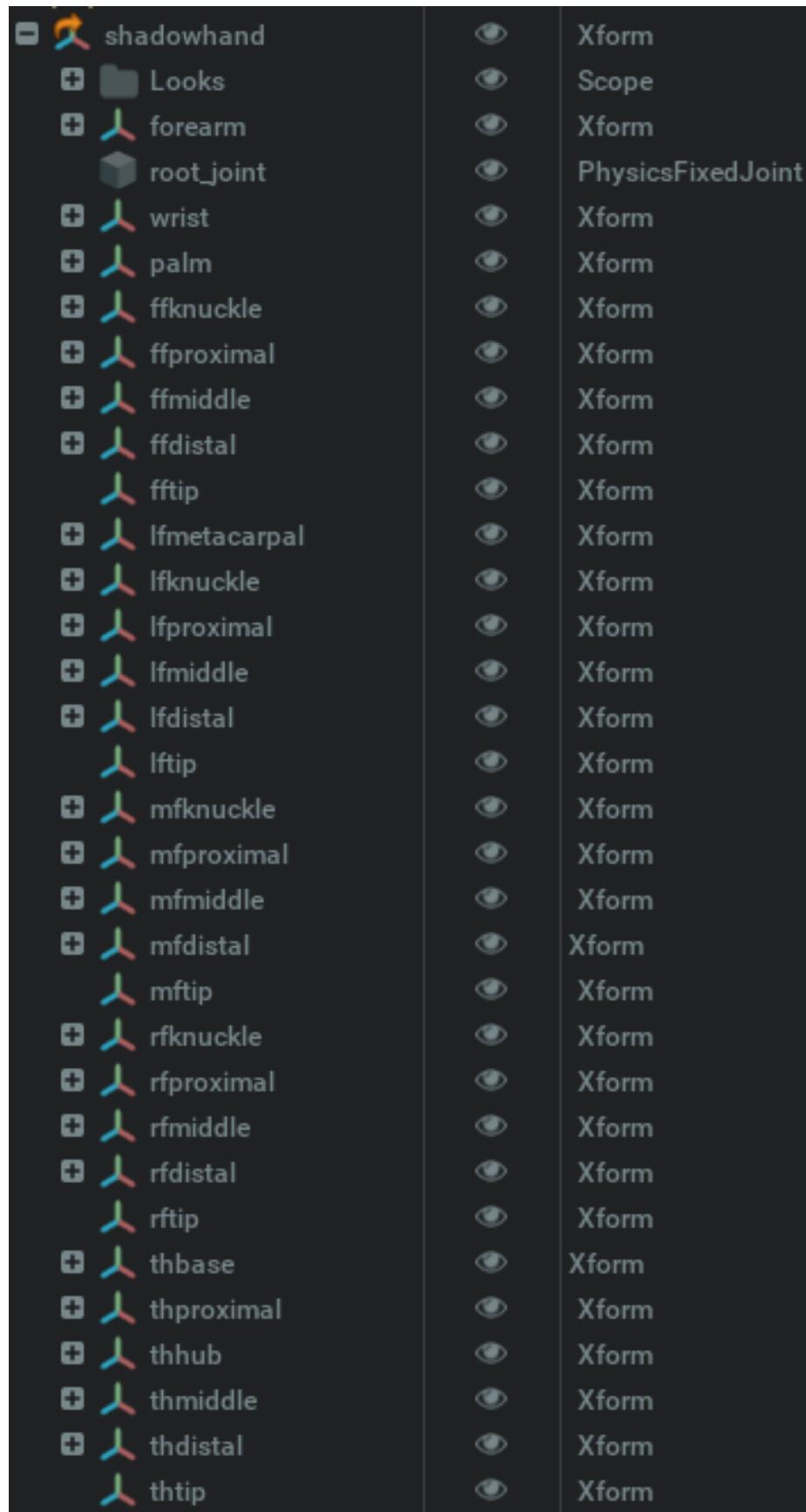


Rysunek 4.1 Zestawienie zdjęć modelu z różnych perspektyw w domyślnej konfiguracji



Rysunek 4.2 Zdjęcie modelu z siatką kolizji (różowe oznaczenia) i oznaczonymi przegubami (niebieskie oznaczenia)

Model ręki składa się z kawałka przedramienia, nadgarstka o 2 DoF, dłoni, palców o 3 DoF każdy oraz kciuka o 5 DoF. W tabeli 4.1 opisano zakresy ruchowe każdego przegubu zgodnie z ich umiejscowieniem w drzewie, przedstawionym na rysunku 4.3. W nawiasach wymieniono część, w której znajduje się dany przegub. Skróty są zakodowane jako WR-Wrist, LF-Little Finger, RF-Ring Finger, MF-Middle Finger, FF-Fore Finger, TH-Thumb, J-Joint, liczba oznacza który to przegub od nadgarstka do końca ręki (np. WRJ1 - Wrist Joint 1).



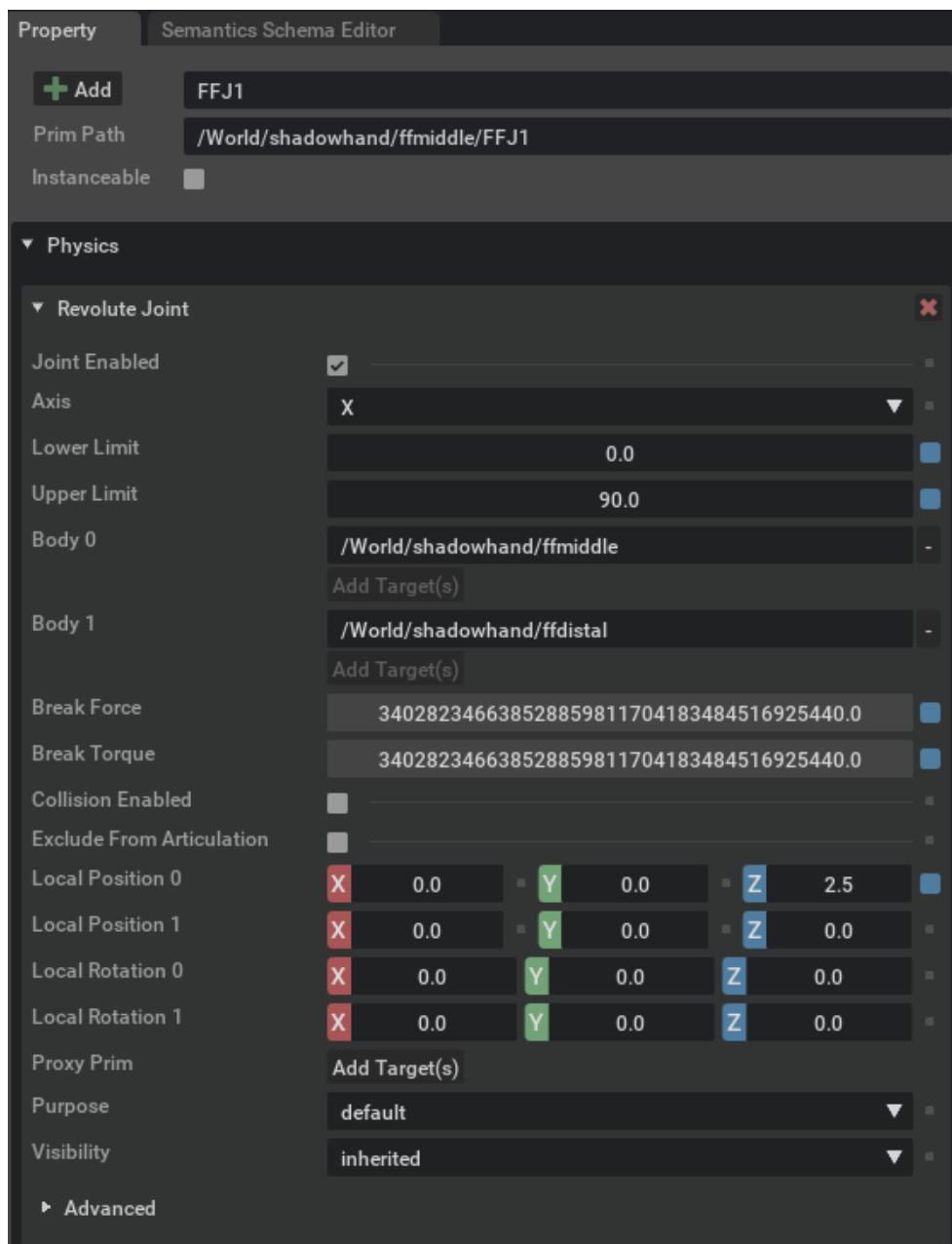
Rysunek 4.3 Zdjęcie drzewa modelu

Tabela 4.1 Tabela zakresów ruchów przegubów modelu

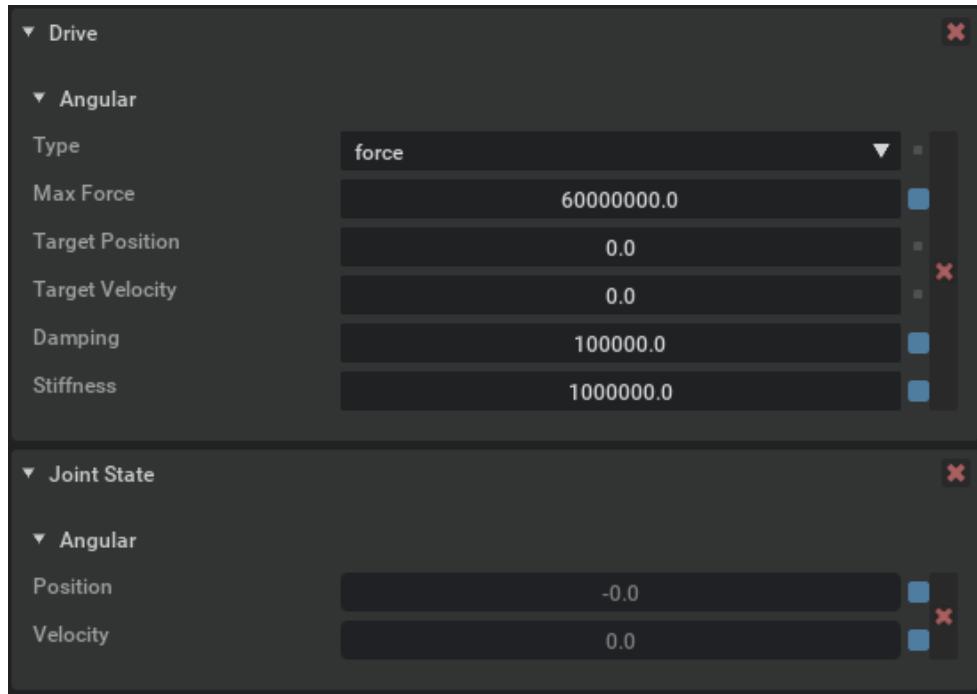
Przegub	Zakres[°]	
	Min	Max
Nadgarstek		
WRJ1 (wrist)	-45	30
WRJ2 (forearm)	-30	10
Dłoń		
LFJ5 (palm)	0	40
RFJ4 (palm)	-25	25
MFJ4 (palm)	-25	25
FFJ4 (palm)	-25	25
THJ5 (palm)	-60	60
Palec mały		
LFJ4 (lfmetacarpal)	-25	25
LFJ3 (lfknuckle)	0	90
LFJ2 (lfproximal)	0	90
LFJ1 (lfmiddle)	0	90
LFTip (lfdsital)	0	0
Palec serdeczny		
RFJ3 (rfknuckle)	0	90
RFJ2 (rfproximal)	0	90
RFJ1 (rfmiddle)	0	90
RFTip (rfdsital)	0	0
Palec środkowy		
MFJ3 (mfknuckle)	0	90
MFJ2 (mfproximal)	0	90
MFJ1 (mfmiddle)	0	90
MFTip (mfdsital)	0	0
Palec wskazujący		
FFJ3 (ffknuckle)	0	90
FFJ2 (ffproximal)	0	90
FFJ1 (ffmiddle)	0	90
FFTip (ffdsital)	0	0
Kciuk		
THJ4 (thbase)	0	75
THJ3 (throximal)	-15	15
THJ2 (thhub)	-30	30
THJ1 (thmiddle)	0	90
THTip (thdistal)	0	0

Aby model zachowywał się zgodnie z prawami fizyki oraz zaimplementowaną fizyką sceny (grawitacją), została dodana fizyka brył sztywnych z obecnością kolizji (ang. *Rigid Body with Colliders Preset*) dla każdego elementu. Za poruszanie częściami odpowiadają konkretne przeguby o zaimplementowanych napędach ang. *drive*. Nałożone

są one na przeguby rotacyjne (ang. *PhysicsRevoluteJoint*, które pozwalają na rotacje wokół 1 osi. Każdy przegub łączy dwa ciała, posiada zakres ruchu 4.1 oraz określone maksymalne siły, które może wytrzymać, zanim zostanie uszkodzony. Opis takiego przegubu został przedstawiony na rysunku 4.4. Napędy mogą być typu siłowego (ang. *force*) lub przyspieszeń (ang. *acceleration*). Określa je również maksymalna siła, którą mogą przyjąć jako sterowanie. Przeguby mogą być regulowane poprzez przekazywanie położenia (ang. *target position*) lub prędkości (ang. *target velocity*). Natomiast parametry tłumienia (ang. *Damping*) i sztywności (ang. *stiffness*) odpowiadają za kontrolę nad sterowaniem. Zakłada się, że sterując przy pomocy położenia powinno się mieć wysoką sztywność i odpowiednio małe lub zerowe tłumienie, a przy sterowaniu prędkością odwrotnie [1]. Na potrzeby stworzonego sterowania oraz zwiększenia kontroli ustawiono wysoką sztywność modelu, co zostało wytłumaczone w rozdziale sterowanie. Własności napędu przedstawiono na rysunku 4.5.

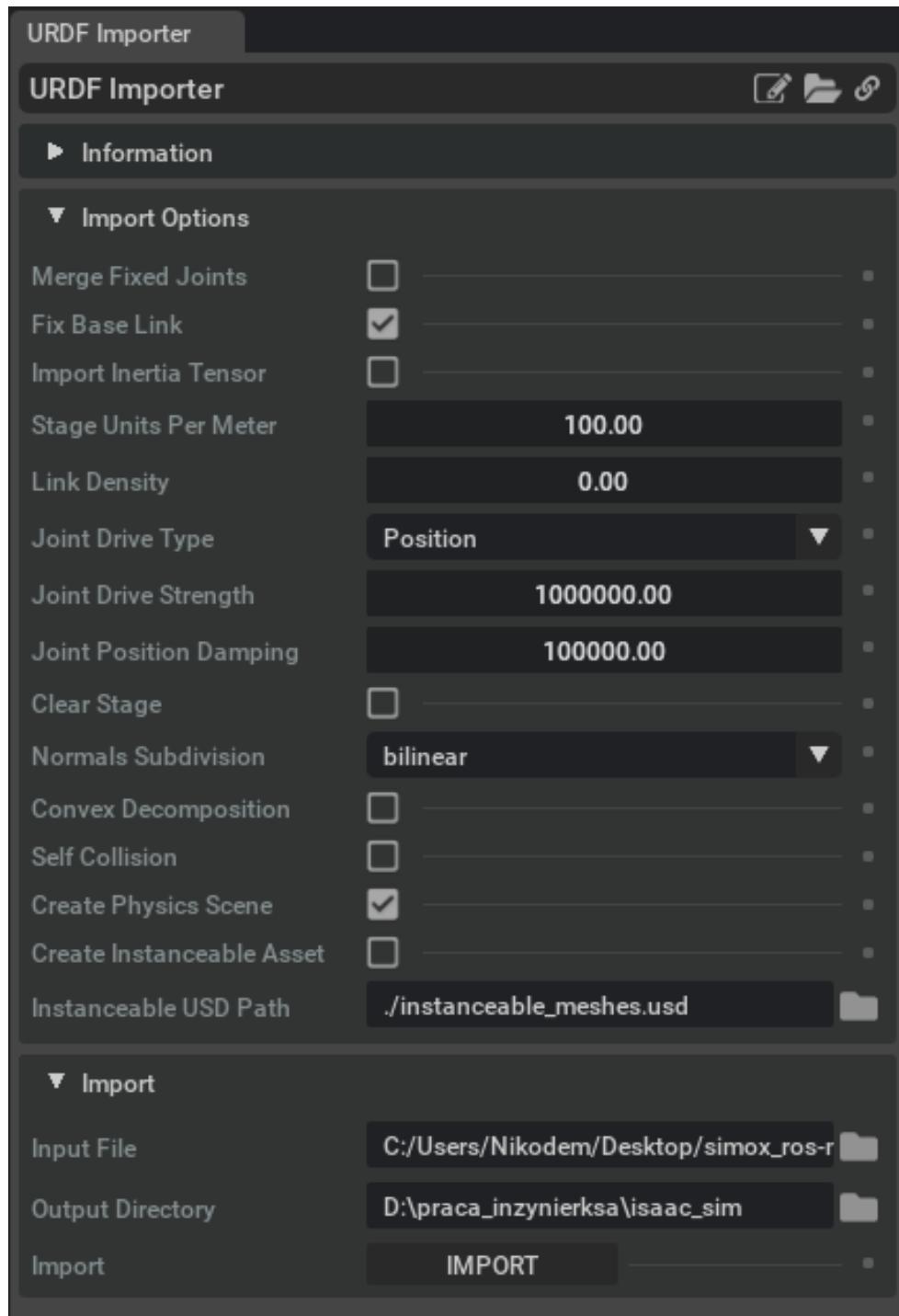


Rysunek 4.4 Zdjęcie przykładowego okna przegubu rotacyjnego na podstawie przegubu FFJ1



Rysunek 4.5 Zdjęcie przykładowego okna napędu przegubu rotacyjnego na postawie przegubu FFJ1

Aby zaimportować model użyto rozszerzenia do środowiska URDF Importer. W oknie importowania można określić liczbę jednostek względem metra, które zostało wyznaczone w liczbie 100, by model był większy i łatwiejszy do oglądania w trakcie symulacji. Ten zabieg nie mógłby być użyty w przypadku nauki sztucznej inteligencji czy obsługi maszyny. Model został sztucznie powiększony 100 razy, co w świecie rzeczywistym mogłoby skutkować niepoprawną pracą lub uszkodzeniem. Dodatkowo w oknie importowania można ustawić wartości sztywności oraz tłumienia napędu. Przykładowe okno importowania zostało pokazane na rysunku 4.6.



Rysunek 4.6 Zdjęcie importowania modelu poprzez URDF Importer

Rozdział 5

Sterowanie

Posiadając środowisko oraz model protezy, należało opracować jej sterowanie. Isaac Sim pozwala na zimplementowanie rozmaitych rozwiązań i ich kombinacji w celu sterowania modelami oraz sceną. Ze względu na założoną możliwość sterownia w czasie rzeczywistym, w trakcie symulacji wybrano sterowanie za pomocą OmniGraphu, który umożliwia implementację kontrolera. Proteza może wykonywać różne ruchy i pozwala na poruszanie każdym ze stawów z osobna. Dlatego zostało opracowane specjalne sterowanie pozwalające na kontrolę wszystkich 24 zamodelowanych przegubów.

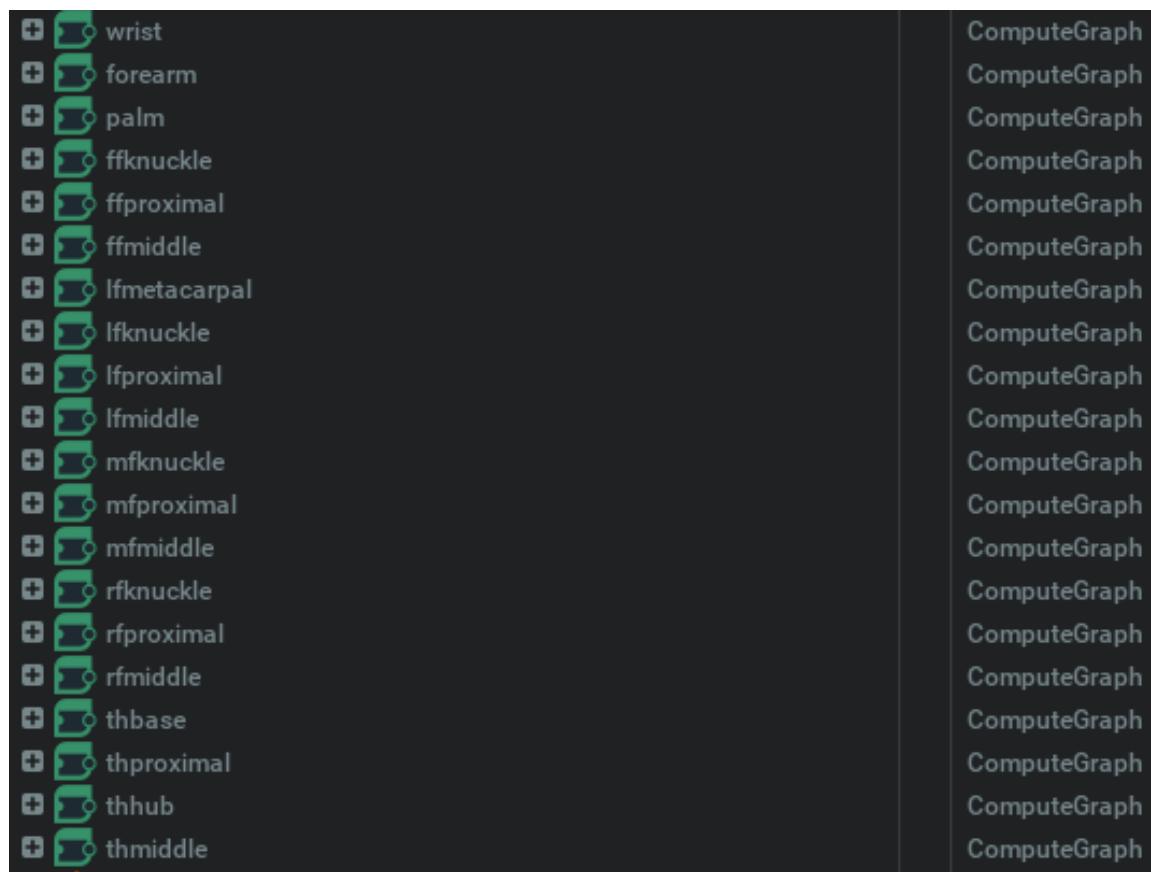
Ze względu na zakresy ruchowe oraz dużą liczbę możliwych wykonywanych ruchów sterowanie przy pomocy popularnych kontrolerów mogłoby być trudne. Klawiatura posiada o wiele więcej przycisków i idealnie nadaje się do obsługi wszystkich napędów. W tabeli 5.1 przedstawiono listę klawiszy sterowania, przegub, którym sterują oraz czy charakter jest dodatni, czy ujemny („+” oznacza zwiększenie wartości przegubu, a „-” zmniejszanie). Należy zaznaczyć, że używane klawisze nie pełnią funkcji przycisków funkcyjnych (oprócz F, który przybliża lub oddala kamerę i G, który wyłącza/włącza siatkę). Do pełnej obsługi wymagana jest również klawiatura numeryczna, która obsługuje przywodzenie/odwodzenie palców oraz orientację części dloni małego palca. Ułożenie klawiszy do obsługi palców zostało oparte na zasadzie segmentów oraz przeciwnych wartości. Klawisze odpowiadające za zwiększenie lub zmniejszanie znajdują się pod sobą np. A/Z. Przez segmenty rozumie się, że klawiaturę można podzielić na sterowanie kolejnymi palcami np. sterowanie palcem wskazującym odbywa się poprzez klawisze A/Z, S/X i D/C.

Sterowanie przegubem opiera się na nadawaniu mu prędkości, która zmienia jego orientację. W trakcie opracowywania rozwiązania problematyczne okazało się zatrzymywanie protezy w miejscu, gdy żaden z przycisków nie jest naciśnięty. Gdy przegub miał mniejszą sztywność, a jego odnotowana prędkość była różna od 0, to kontynuował ruch w czasie. Aby przegub poruszał się podczas naciśnięcia klawisza na klawiaturze, zwiększo jego sztywność. Końcowym efektem jest ruch danego stawu w momencie naciskania klawisza, a gdy ten nie jest wciśnięty przegub zachowuje swoje obecne położenie.

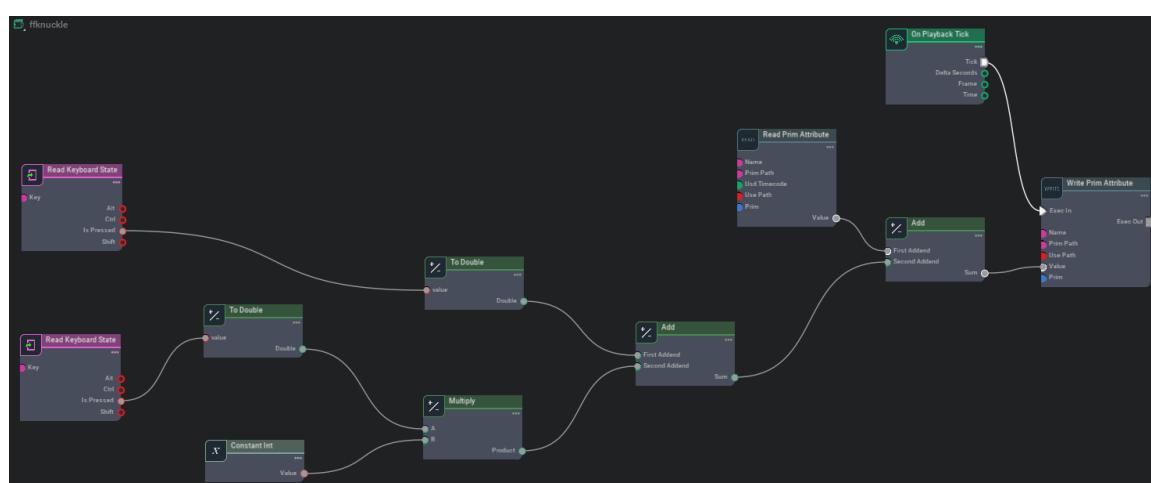
Tabela 5.1 Sterowanie ręką przy pomocy klawiatury

	Klawisz	Przegub	Ruch
Nadgarstek	LeftShift	WRJ1	+
	CapsLock	WRJ1	-
	LeftAlt	WRJ2	+
	LeftControl	WRJ2	-
Dłonί	NumpadAdd	LFJ5	+
	NumpadEnter	LFJ5	-
	Numpad6	RJF4	+
	Numpad9	RJF4	-
	Numpad8	MFJ4	+
	Numpad5	MFJ4	-
	Numpad7	FFJ4	+
	Numpad4	FFJ4	-
	Numpad1	THJ5	+
	Numpad0	THJ5	-
Palec Mały	1	LFJ4	+
	Q	LFJ4	-
	2	LFJ3	+
	W	LFJ3	-
	3	LFJ2	+
	E	LFJ2	-
	4	LFJ1	+
	R	LFJ1	-
Palec serdeczny	5	RFJ3	+
	T	RFJ3	-
	6	RFJ2	+
	Y	RFJ2	-
	7	RFJ1	+
	U	RFJ1	-
Palec środkowy	8	MFJ3	+
	I	MFJ3	-
	9	MFJ2	+
	O	MFJ2	-
	0	MFJ1	+
	P	MFJ1	-
Palec wskazujący	A	FFJ3	+
	Z	FFJ3	-
	S	FFJ2	+
	X	FFJ2	-
	D	FFJ1	+
	C	FFJ1	-
Kciuk	F	THJ4	+
	V	THJ4	-
	G	THJ3	+
	B	THJ3	-
	H	THJ2	+
	N	THJ2	-
	J	THJ1	+
	M	THJ1	-

Opracowane sterowanie oparte jest na tzw. *Action Graph*. W celu zapewnienia przejrzystości rozwiązania stworzono liczne grafy. Każdy odpowiada za kontrolę przegubu z konkretnej części z drzewa modelu 4.3 co zostało zobrazowane na rysunku 5.1. Grafy powstały w sposób analogiczny, gdyż warunki określone są jedynie przez napędy oraz własności przegubów. Przykładowy graf pokazano na rysunku 5.2.



Rysunek 5.1 Zdjęcie drzewa grafów sterowania



Rysunek 5.2 Zdjęcie przykładowego sterowania na podstawie grafu *ffknuckle*

Po uruchomieniu symulacji zostaje aktywowany *Node* pozwalający na przekazywanie wartości prędkości do przegubu. Naciskając odpowiedni klawisz odczytywana jest wartość 1 lub -1. Następnie sprawdzane jest, czy obydwa przyciski nie są naciśnięte oraz sumowana jest ich wartość wypadkowa. Potem ta jest sumowana wraz z odczytaną aktualną wartością *drive:angular:physics:target:Velocity* danego przegubu i wysyłana jest ona do przegubu. Dzięki takiej implementacji możliwa jest precyzyjna, chociaż powolna zmiana położenia przegubu i dokładne symulowanie ruchów ręki. Ze względu na równoczesną pracę wszystkich grafów w trakcie symulacji można ruszać różnymi elementami ręki jednocześnie. Ograniczenie stanowi ilość aktywnie odczytywanych jednocześnie klawiszy przez klawiaturę. Jeżeli jakiś przegub osiągnie swój limit to nadawanie mu kolejnych prędkości powoduje zwiększenie *Target Velocity* na przegubie, ale nie powoduje on dodatkowych zmian, czy niekontrolowanych zachowań między elementami modelu.

Oprócz powyższego zalecanego sterowania istnieje możliwość sterowania poprzez wpisywanie wartości do samego napędu. Jest to najprostsza forma sterowania, która istnieje od razu po dodaniu napędu na przegub. Jest to przydatna funkcja, aby szybko wyzerowywać wartości przegubów lub sprawdzić konkretne ustawienia bez konieczności doprowadzenia ręki do konkretnego stanu od stanu początkowego.

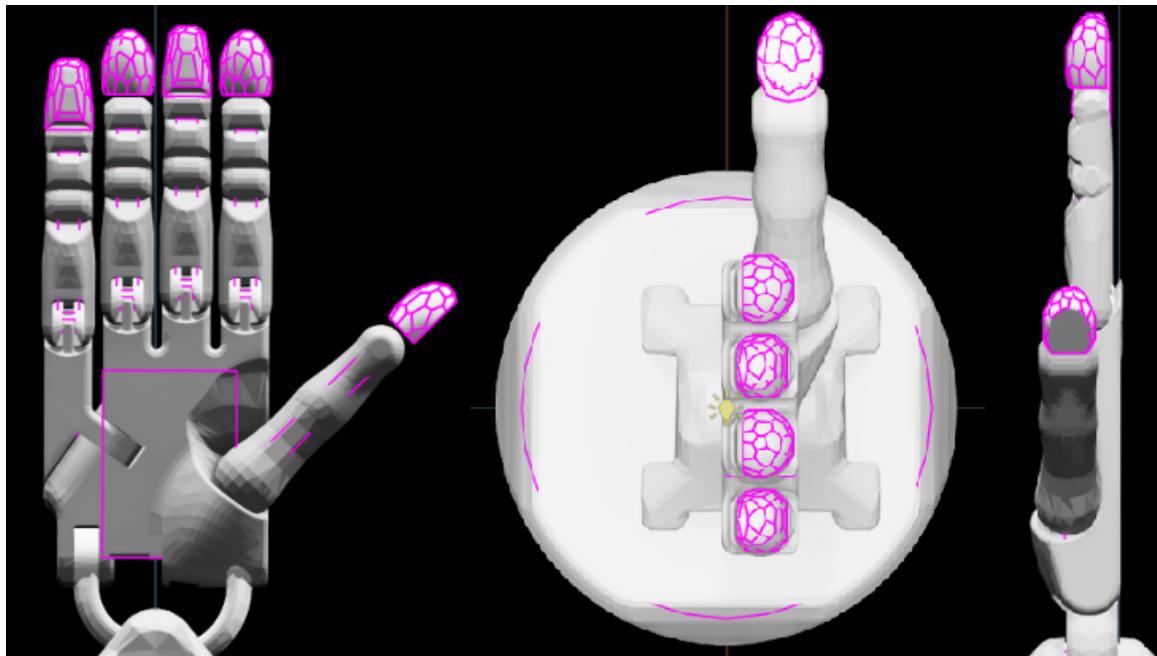
Rozdział 6

Symulacje

Symulacje chwytów rozpoczynały się od pozycji domyślnej przedstawionej w poprzednim rozdziale. Po każdorazowej symulacji wszystkie wartości przegubów były zerowane poprzez ponowne wczytanie modelu. Sterowanie odbywało się zgodnie z omówionym sposobem za pomocą klawiatury.

6.1 Otwarta oraz zrelaksowana dłoń

Otwarta dłoń przedstawiona na rysunku 6.1, pełni funkcję domyślnej konfiguracji protezy i od niej zaczynało się symulowanie kolejnych gestów. W trakcie tej symulacji wartości pozycji oraz prędkości na napędach wynoszą 0.

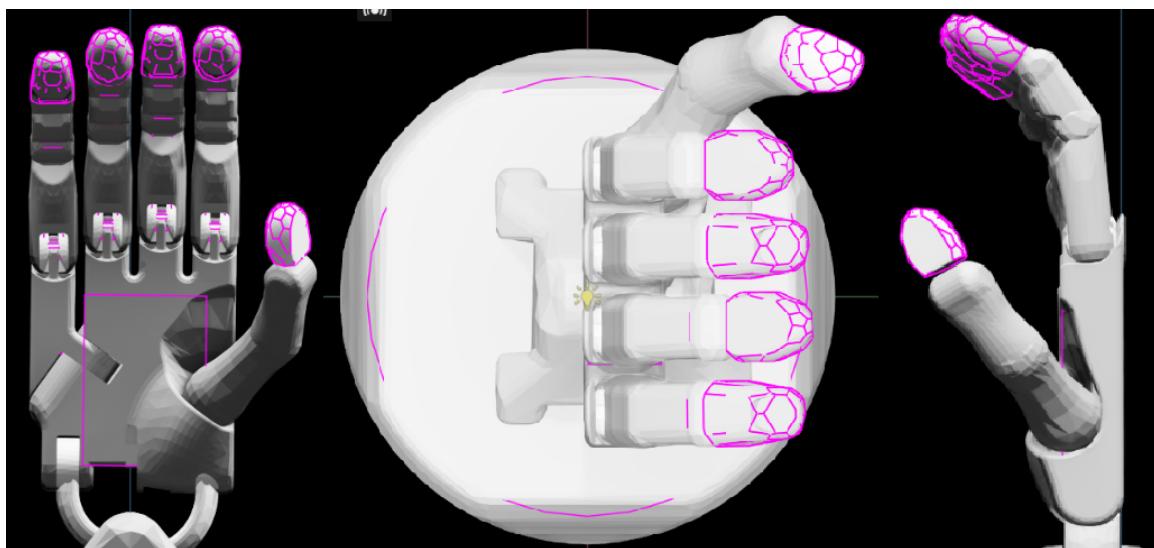


Rysunek 6.1 Zdjęcie modelu w konfiguracji domyślnej

Bioniczna proteza w spoczynku przyjmuje formę zrelaksowanej ręki, którą charakteryzuje zginający się do dłoni kciuk w pozycji bocznej oraz pozostałe lekko zgięte palce. Jest to ważny gest, gdyż, stanowi o wygodzie użytkownika. Zarówno sprawia wrażenie naturalności, jak i umożliwia szybkie przejście do innych chwytów jak np. walcowy czy pięści. Na rysunku 6.2 przedstawiono zdjęcie z instrukcji rzeczywistej protezy ręki [12], następnie na rysunku 6.3 pokazano zdjęcia symulowanego modelu z różnych perspektyw. Pierwszym krokiem symulacji były drobne zgięcia w poszczególnych stawach palców, a następnie wzorując się na rzeczywistej ręce oraz załączony zdjęciu stosowane protezy 6.2 na odpowiednim wygięciu kciuka.



Rysunek 6.2 Zdjęcie zrelaksowanej ręki Bebionic [12]



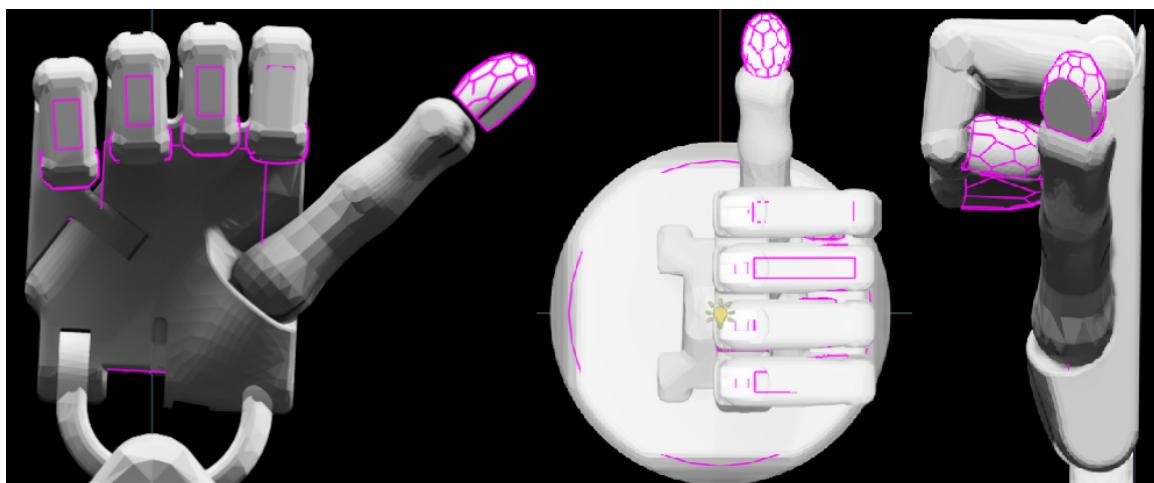
Rysunek 6.3 Zdjęcie modelu reprezentującego zrelaksowaną protezę

6.2 Zaciśnięta pieść

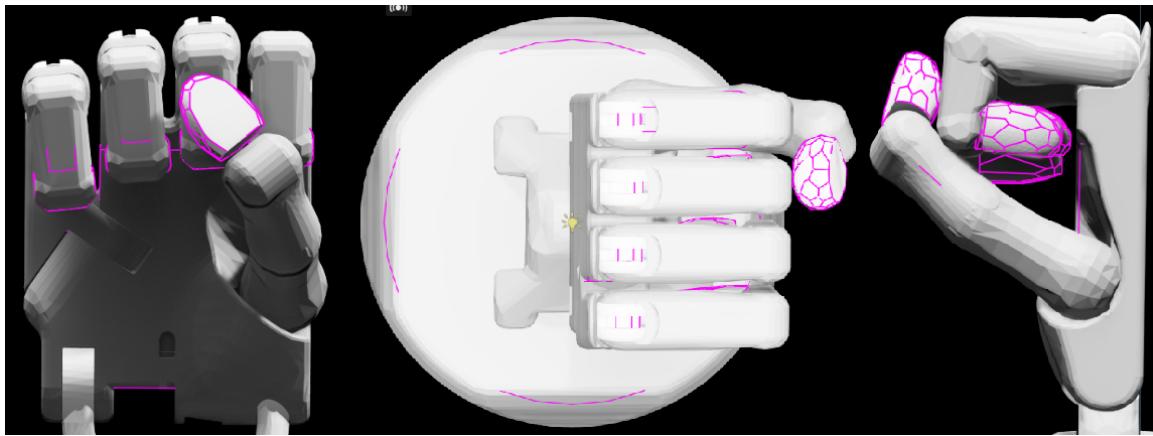
Zaciśnięcie pięści jest niezwykle trudnym gestem do wykonania ze względu na wymiary protezy oraz odpowiednie ustawienia kciuka. W trakcie przeprowadzonych symulacji udało się uzyskać efekt podobny do prezentowanego przez firmę Shadow Robot (rysunek 6.4). Na kolejnych rysunkach 6.5 i 6.6 pokazano zdjęcia zacieśniana pięści oraz końcowy gest. Symulacje rozpoczęto od konfiguracji domyślnej oraz maksymalnego zgęśczenia palców. Następnie zajęto się ruszaniem kciuka, by ten znalazł się na poziomie zgiętych palców. Starano się by paliczek dalszy kciuka naciskał na środkowy palec, co osiągnięto i przedstawiono niżej.



Rysunek 6.4 Zdjęcie pięści robotycznej ręki Shadow Dexterous Hand E [13]



Rysunek 6.5 Zdjęcie modelu w trakcie zaciskania pięści



Rysunek 6.6 Zdjęcie modelu zaciśniętej pięści

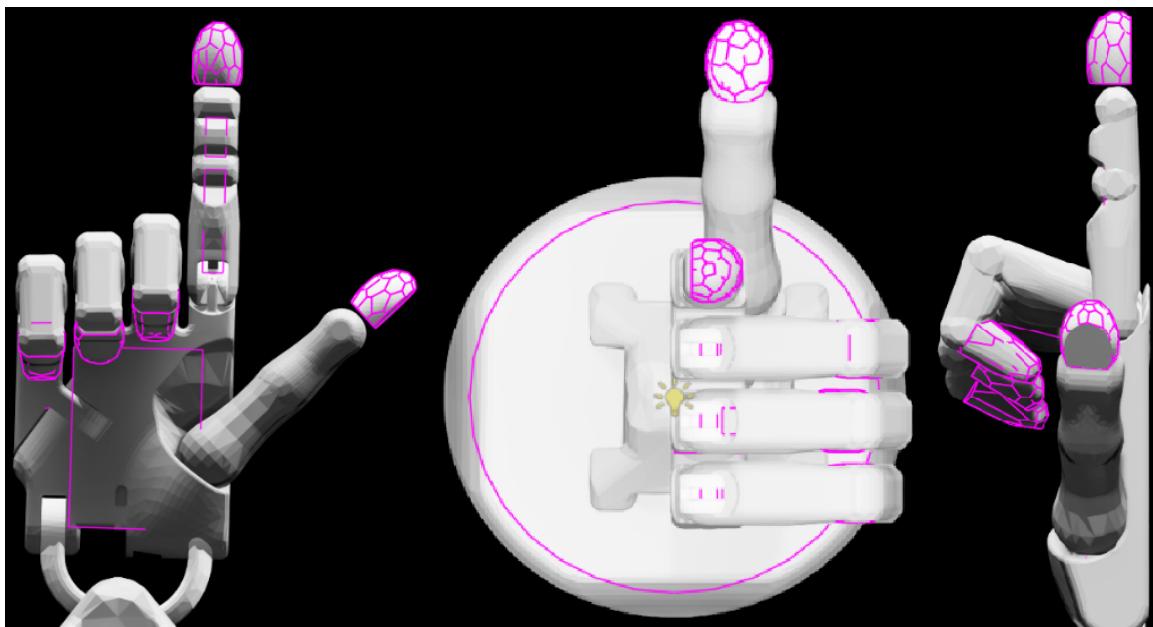
Porównując otrzymany chwyt do przedstawionego przez firmę Shadow Robot, można uznać, że gest został wykonany poprawnie. Obydwa ułożenia wyglądają podobnie, jednakże nie są to idealne ludzkie pięści. W trakcie symulacji kciuk nie pokrywa palców i nie stanowi pełnego zabezpieczenia.

6.3 Chwyt pęsetowy

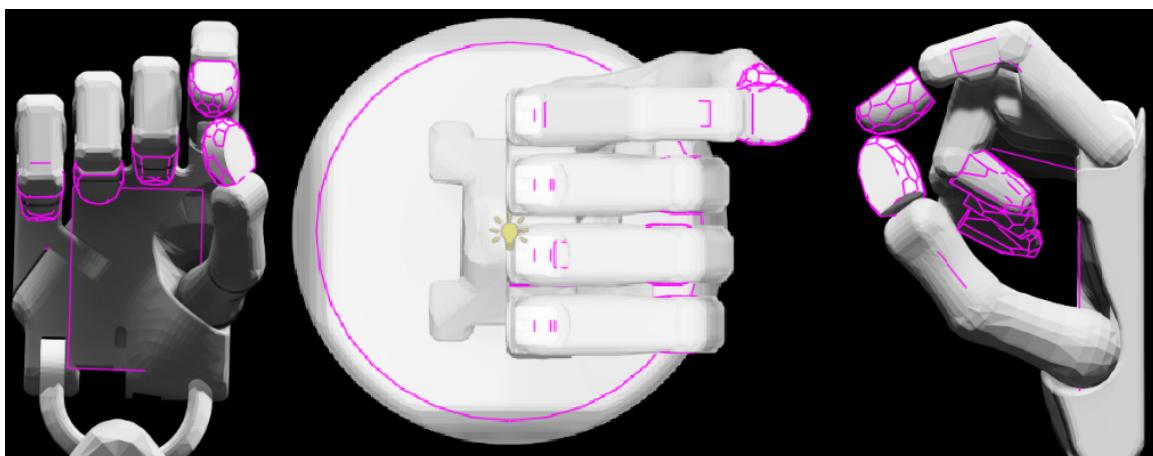
Chwyt pęsetowy jest podstawowym chwytem wykonywanym w życiu codziennym. Dla symulacji bionicznej protezy jest o tyle istotny, gdyż wymaga kontroli kciuka oraz palca wskazującego, nie dopuszczając by pozostałe palce ingerowały w chwyt. Takim chwytom można podnosić zarówno drobne obiekty jak np. monety oraz większe lekkie przedmioty. Dla symulacji wybrano chwyt pęsetowy ze zgiętymi palcem małym, serdecznym i środkowym. Na początku symulacji zgięto palec mały, serdeczny oraz środkowy tak by nie przeszkadzały w trakcie chwytu, ale by nie były maksymalnie zgięte jak w przypadku pięści (rysunek 6.8). Następnie rozpoczęto zginanie palca wskazującego oraz kciuka, tak by stykały się swoimi opuszkami w oddalonym od reszty palców punkcie (rysunek 6.9). Symulację przeprowadzono na podobieństwo rzeczywistego chwytu protezy (rysunek 6.7) oraz chwytu ludzkiej ręki.



Rysunek 6.7 Zdjęcie chwytu pęsetowego zrealizowanego przez protezę Bebionic [12]



Rysunek 6.8 Zdjęcie modelu w trakcie realizacji chwytu pęsetowego

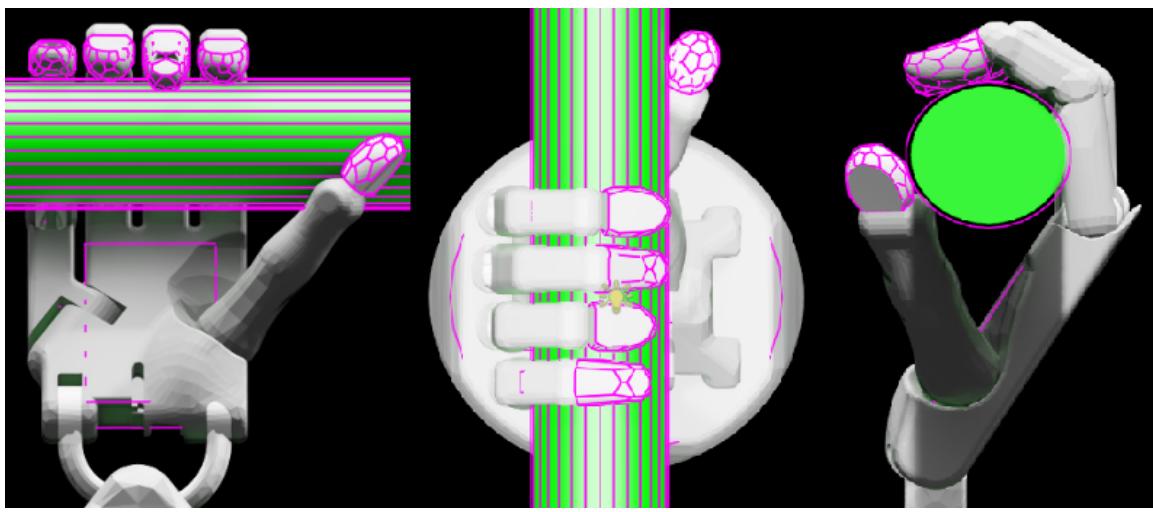


Rysunek 6.9 Zdjęcie modelu w momencie chwytu pęsetowego

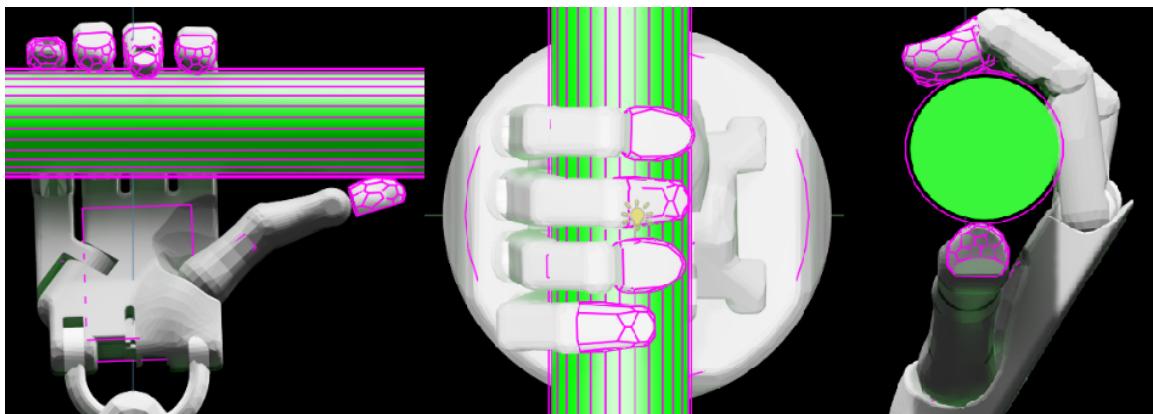
Chwyt został zrealizowany poprawnie. Końcówki palca wskazującego oraz kciuka zetknęły się poza zakresem pozostałych palców, które nie ingerowały w chwyt i pozostały swobodne tak jak w przypadku protezy oraz rzeczywistej ręki.

6.4 Pełny chwyt hakowy

Chwyt hakowy jest jednym z częściej używanych podczas dnia np. trzymania ucha kubka, trzymania poręczy w tramwaju, czy noszenia zakupów. Dla symulacji zdecydowano się na wersję z przywiedzionym kciukiem, aby zweryfikować jego zakres ruchowy. W celu pokazania chwytności symulowanej bionicznej protezy reki stworzono walec, któremu nadano fizykę kolizji. Dzięki temu stanowi on oparcie i blokadę dla modelu ręki, która go chwytała. W trakcie symulacji najpierw odchylono nadgarstek oraz kciuk i ustawiły cylinder w dłoni. Następnie zacieśniano palce do momentu, aż napotkały opór ze względu na wykrycie kolizji z ustawnionym obiektem (rysunek 6.10). Końcowym etapem symulacji było przywiedzenie kciuka, tak by złapał wał wzdłuż jego długości (rysunek 6.11).



Rysunek 6.10 Zdjęcie modelu w trakcie realizacji chwytu hakowego



Rysunek 6.11 Zdjęcie modelu w momencie chwytu hakowego

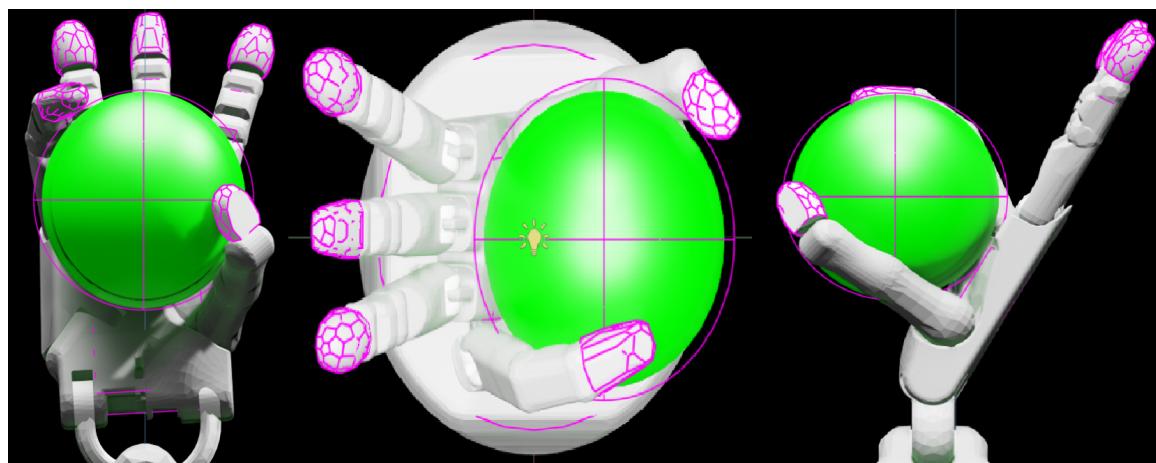
Symulacja przebiegła poprawnie i udało się przedstawić pełne przywiedzenie kciuka. Symulowany ruch jest pełnym chwytu hakowym, gdyż rurka nie jest w całej dloni, a tylko w płaszczyźnie palców. Taki sam chwyt hakowy zawarto również na rysunku 3.11. Symulacja pokazała zablokowanie się palców na obiekcie oraz ustawnienie kciuka w innej pozycji, niż symulowano wcześniej. Kciuk jest przywiedziony i wyprowadzony w prawie maksymalnym zakresie, podczas gdy paliczek dalszy ma kontakt z cylindrem.

6.5 Chwyt kulisty

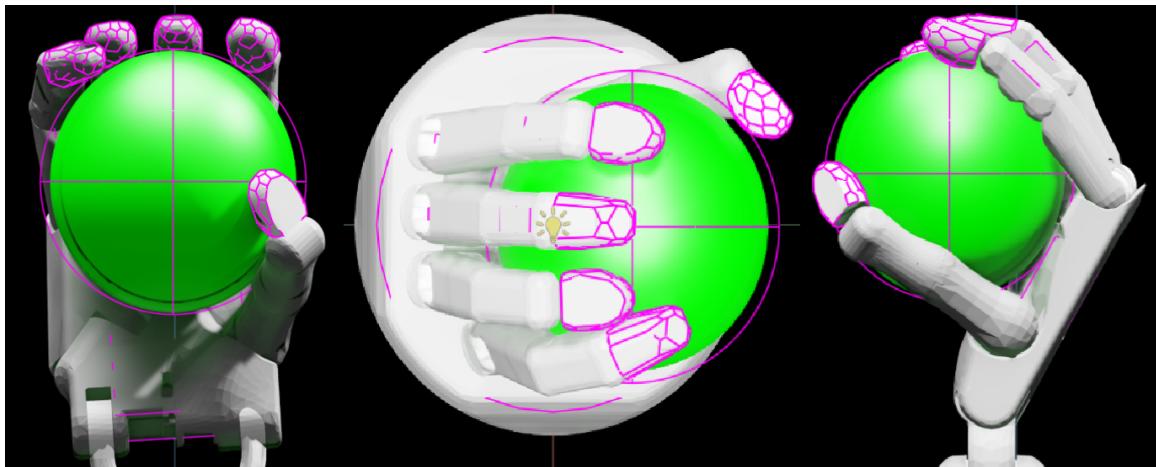
Chwyt kulisty jest jednym z najbardziej zaawansowanych chwytów ze względu na liczbę VF, która wynosi od 2 do 5. Jego wersja zależy również od wielkości trzymanego przedmiotu. Aby zaprezentować jeden z najtrudniejszych przypadków wybrano przykład o 5VF. Na potrzebę symulacji stworzono kulę, z fizyką kolizji o średnicy pozwalającej na objęcie większością dłoni. Jej rozmiar był wzorowany na piłce tenisowej. Symulacje rozpoczęto od odchylenia nadgarstka i przystawieniu piłki do dłoni. Następnie kolejno przyblizono do niej dłoń przy pomocy nadgarstka oraz rozstawiono odpowiednio szeroko wszystkie palce. Ustawiono również kciuk oraz wygięto dłoń w płaszczyźnie małego palca. Zaciśnięto mały palec oraz kciuk czego efekt widać na rysunku 6.13. Pozostało tylko złapać piłkę resztą palców, co zobrazowano na rysunku 6.14.



Rysunek 6.12 Chwycona piłka tenisowa [14]



Rysunek 6.13 Zdjęcie modelu w trakcie realizacji chwytu kulistego



Rysunek 6.14 Zdjęcie modelu w momencie chwytu kulistego

Symulacja przebiegła prawidłowo. Wszystkie palce zablokowały się na powierzchni kuli, obejmując ją w większości niczym piłkę do tenisa. W trakcie symulacji wykorzystano wszystkie przeguby rotacyjne, czego wynikiem jest chwyt przypominający chwyt piłki tenisowej przez rzeczywistą rękę (rysunek 6.12).

Rozdział 7

Podsumowanie

Proteza ma za zadanie zastąpić brakującą część człowieka. Zrealizowanie tego w pełni jest obecnie niemożliwe. Natomiast możliwe jest opracowanie bionicznej ręki, która pozwoli użytkownikowi na wykonywanie jak największej liczby czynności, dzięki bogatej bibliotece chwytów. Pomimo zaawansowanego postępu w dziedzinie bioniki i szeroko rozumianej inżynierii nie istnieje urządzenie idealne. Każde rozwiązanie jest podatne na uszkodzenia, a żadne nie jest całkowicie samonaprawialne. Nawet najlepsze do tej pory protezy należy ładować do poprawnego i jak najdłuższego funkcjonowania. Ludzka ręka wykonuje dziennie tysiące ruchów w różnych konfiguracjach i zazwyczaj służy przez całe ludzkie życie. Podczas gdy rozwiązania elektryczno-mechaniczne często mają określony czas sprawności. Jednakże można przybliżać się do osiągnięcia przełomu, a podstawowym narzędziem są symulacje komputerowe.

Na rynku istnieje ogrom środowisk symulacyjnych, które mają swoje wady i zalety. Każdy z osobna znajdzie narzędzie, które będzie jej bardziej odpowiadać. W przypadku symulacji bionicznej protezy ręki, po gruntownej analizie zdecydowano, że będzie to Isaac Sim. Jest to symulator o zastosowaniu ogólnym, ale ponieważ ma dokładny system fizyki oraz ogromny wachlarz narzędzi takich jak tworzenie i manipulowanie obiektami, czy implementacja różnego rodzaju algorytmów sterowania, jest narzędziem optymalnym do celu tej pracy.

Aby móc zacząć mówić o symulacji czegokolwiek, najpierw trzeba posiadać, chociażby podstawową wiedzę z danego zakresu. W przypadku symulacji bionicznej protezy ręki przeanalizowano anatomię rzeczywistej ręki. Przedstawiono również jej matematyczny model, za pomocą którego obliczono kinematykę prostą i odwrotną. Dodatkowo opisano zmierzone zakresy ruchowe palców. Znając anatomiczny oraz model kinematyczny ręki pozostało ustalić rodzaje chwytów do symulacji. W tym celu omówiono taksonomię ruchów ze względu na charakter, orientację dloni, ilość wirtualnych palców oraz pozycję kciuka. Spośród różnych chwytów zdecydowano się na symulacje codziennych chwytów, gdzie każda kolejna wizualizacja pokazywała inny aspekt modelu ręki.

W celu posiadania pewności o zachowaniu się protezy w czasie symulacji zaimplementowano model ręki Shadow Dexterous Hand E. Pomimo posiadania mniejszej liczby stopni swobody niż model kinematyczny rzeczywistej ręki, Shadow Dexterous Hand E pozwala na podobny zakres ruchów. Ponadto jego budowa oparta jest na ludzkiej anatomicznej oraz dodatkowo zaimplementowany jest mechanizm wyginania płaszczyzny dłoni przy małym palcu. Zakresy ruchowe kolejnych stawów również są zbliżone do rzeczywistej ręki.

Dla modelu opracowano sterowanie. Podstawowym wymaganiem była możliwość kontroli w trakcie symulacji, co sugerowało opracowanie jakiejś formy kontrolera. Ze

względzie na dużą liczbę sterowalnych przegubów oraz na zakres ruchu w osiach dodatkowych i ujemnych zdecydowano się na opracowanie sterowania przy pomocy klawiatury, której liczba klawiszy wystarcza do obsługi całej ręki. W związku z tym opracowano sterowanie przy użyciu interaktywnych grafów, które są wewnętrznym sposobem programowania w pakiecie narzędzi Omnisverse. Dla przejrzystości implementacji opracowano tyle sterowań, ile jest części z przegubami rotacyjnymi w drzewie modelu. Aby zwiększyć responsywność symulowanej protezy, zwiększo sztywność stawów, co spowodowało że ta rusza się, gdy wcisnięty jest klawisz.

Finalnie zostało tylko zasymulować wybrane rodzaje chwytów. Pierwszym z nich był gest zrelaksowanej dłoni. Jest to jeden z najważniejszych ruchów, gdyż stanowi w ogromnym stopniu o naturalności protezy. Jest to zarazem najprostszy z gestów, gdyż wymaga małego zgęcia w przegubach palców i kciuka. W przypadku symulacji udało się osiągnąć efekt podobny jak do rzeczywistej protezy marki Bebionic 6.2, która jest komercyjnym stosowanym rozwiązaniem.

Kolejnym gestem była zaciśnięta pięść. Dla symulowania protez ręki jest to niezwykle ważny gest, który weryfikuje czy zostały dobrane odpowiednie wymiary ramion (kości) oraz czy przeguby (stawy) mają wystarczające zakresy ruchu. Jak widać obydwa przedstawione wcześniej rysunki 6.6 i 6.4 są podobne. Symulowana pięść jest podobna do tej ze zdjęcia producenta 6.4. Jednakże przekładając je na rzeczywisty chwyt, nie spełnia on całkowicie swoich założeń. Pomimo odpowiedniego zgęcia palców do środka, symulowana ręka nie była w stanie nałożyć kciuka na palce, by ten je zablokował i pełnił funkcję wsparcia. Wynikać to mogło zza krótkich ramion budujących kciuk, odrobinę za małych zakresów ruchowych stawów oraz możliwych błędów w trakcie sterowania, gdyż każdy błąd w odchyleniu przegubu mógł zwiększać błąd położenia kciuka względem zamierzonego ułożenia. To również potwierdza, dlaczego rozwiązania rzeczywiste są dodatkowo kalibrowane przez specjalistów. Pomimo tego w trakcie symulacji pięści pokazano, ruchliwość wszystkich przegubów palców oraz kontrolę nad kciukiem w stanie odwiedzionym.

Następnie symulowany był chwyt pęsetowy, który jest chwytem precyzyjnym. Jest to jeden z podstawowych ruchów, służący głównie podnoszeniu drobnych przedmiotów przy pomocy kciuka oraz palca wskazującego. Ze względu na swój charakter oraz wymóg kontroli kciuka jest to istotny chwyt względem protezy. W trakcie symulacji udało się osiągnąć zamierzone cele. W chwycie brały udział tylko palec wskazujący oraz odwiedzony kciuk, które stykały się paliczkami dalszymi. Pozostałe palce były delikatnie zgęte, by nie ingerować w chwyt. Warto podkreślić, że chwyt był podobny do chwytu wykonywanego przez protezę Bebionic 6.7.

Czwartym symulowanym chwytem był pełen chwyt hakowy. Opiera się on na złapaniu przedmiotu w płaszczyźnie palców. Zdecydowano się na wariant z przywiedzionym kciukiem, by sprawdzić jego sprawność w takiej konfiguracji. Do tego celu stworzono model cylindra o małej średnicy, przedstawiającego rurkę, oraz dodano mu fizykę kolizji, by mógł wejść w interakcję z symulowaną protezą. Symulację chwytu rozpoczęto od wygięcia kciuka, wychylenia nadgarstka oraz włożeniu poprzez translację obiektu rurki, by ta znajdowała się w obrębie palców. Następnie przybliżono do niej dłoń z palcami i rozpoczęto zacieśnianie palców, aż te napotkały opór wywarty przez kolizję z rurką. Posiadając chwycony wał przez palce, ruszano kciukiem, by ten złapał cylinder wzduż jego długości. Jak widać na końcowym efekcie (rysunek 6.11), udało się doprowadzić kciuk do pozycji przywiedzionej. Należy jednak zauważyć, że kciuk jest w pozycji nienaturalnej - przeproście. Jest to celowe, gdyż pokazuje prawidłowość w zakresach stawów kciuka w innej pozycji niż odwiedziona, z którą wykonuje się najczęściej czynności w życiu codziennym. Ponadto chwyt był podobny do tego

umieszczonego na rysunku 3.11 w podrozdziale o połączonej klasyfikacji chwytów.

Piątą, a zarazem finalną symulacją była symulacja chwytu kulistego. Na potrzeby tej symulacji stworzono kulę z fizyką kolizji, tka jak poprzednio symulowany walec. Kula wzorowana była na piłce tenisowej, a zamierzony chwyt został zrealizowany przy użyciu wszystkich 5 palców. Początek symulacji odbył się podobnie jak w przypadku chwytu hakowego. Został odgięty nadgarstek, który następnie przybliżono do ustawnionej kuli. Potem odgięto dłoń w pozycji małego palca oraz rozsunieto odpowiednio wszystkie palce. Kolejno zaciśnięto kciuk oraz mały palec do momentu blokady powodowanej przez kolizję ze sferą. Finalnie zaciśnięto ostatnie 3 palce. Porównując otrzymany wynik (rysunek 6.14) z rzeczywistymi chwytami (rysunek 6.12), stwierdzono że symulacja wyszła poprawnie. To potwierdziło, że symulowana ręka wykonuje poprawnie zgięcia przegubów mimo odchylenia palców oraz dloni.

Podsumowując całość pracy, stwierdzono że znaleziono, wybrano i opracowano odpowiednie narzędzia w celu przybliżenia problematyki symulacji bionicznej protezie ręki. Przedstawiono anatomiczne rzeczywistej reki oraz jej kinematyczny odpowiednik. W środowisku symulacyjnym opracowano i wysterowano model protezy, którego końcowym efektem było 5 symulacji chwytów i gestów, gdzie każda wnosiła wartość dodaną i sprawdziła model pod innym katem. Potwierdziło to również możliwość symulowania i sterowania protezą w środowisku Isaac Sim. Pracę można rozwiniąć pod wieloma aspektami, np. rozwinać model o dodanie czujników pomiarowych, by odczytać jakie siły działają w trakcie chwytu, opracować różne inne algorytmy sterowania, czy skrypty symulacyjne w celu nauki ruchów przez sztuczną inteligencję lub zaimplementować inne modele, by porównać ich chwytność oraz ruchliwość.

Bibliografia

- [1] Nvidia. Omniverse robotics documentation. https://docs.omniverse.nvidia.com/app_isaacsim/app_isaacsim/overview.html#.
- [2] Wikipedia. Kończyna góra. https://pl.wikipedia.org/wiki/Ko%C5%84czyna_g%C3%B3rna.
- [3] Media Nauka. Kości ręki. <https://www.medianauka.pl/kosci-reki>.
- [4] American Society for Surgery of the Hand. Body anatomy: Upper extremity joints. <https://www.assh.org/handcare/safety/joints>.
- [5] John-John Cabibihan, Farah Alkhatib, Mohammed Mudassir, Osama Sam Al-Kwif, Khaled Diab, and Elsadig Mahdi. Suitability of the openly accessible 3d printed prosthetic hands for war-wounded children. *Frontiers in Robotics and AI*, 7:594196, 01 2021.
- [6] Salvador Cobos, Manuel Ferre, M.A. Sanchez Uran, Javier Ortego, and Cesar Peña. Efficient human hand kinematics for manipulation tasks. In *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2246–2251, 2008.
- [7] Thomas Feix, Javier Romero, Heinz-Bodo Schmiedmayer, Aaron M. Dollar, and Danica Kragic. The grasp taxonomy of human grasp types. *IEEE Transactions on Human-Machine Systems*, 46(1):66–77, 2016.
- [8] Michael H. Morgan and David R. Carrier. Protective buttressing of the human fist and the evolution of hominin hands. *Journal of Experimental Biology*, 216(2):236–244, 01 2013.
- [9] Andreas Geiger, Elisabeth Brandenburg, and Rainer Stark. Natural virtual reality user interface to define assembly sequences for digital human models. *Applied System Innovation*, 3:15, 03 2020.
- [10] ORTHOFIXAR. Hand function tests. <https://orthofixar.com/special-test/hand-function-tests/>.
- [11] Nigel Palastanga, Derek Field, and Roger Soames. The upper limb. In *Anatomy and Human Movement*, chapter 4, pages 65–294. Butterworth-Heinemann, 2 edition, 1994.
- [12] ottobock. Bebionic hand eqd. <https://www.ottobock.com/en-us/product/8E70>.
- [13] Shadow Robot Company. Shadow robot. <https://www.shadowrobot.com/>.

- [14] Gautam Sadarangani and Carlo Menon. A preliminary investigation on the utility of temporal features of force myography in the two-class problem of grasp vs. no-grasp in the presence of upper-extremity movements. *BioMedical Engineering OnLine*, 16, 05 2017.
- [15] Robomatter Inc. Robot virtual worlds. <https://www.robotvirtualworlds.com/>.
- [16] Cogmation Robotics. Virtual robotics toolkit. <https://www.virtualroboticstoolkit.com/>.
- [17] ABB. Abb robotstudio. <https://new.abb.com/products/robotics/robotstudio>.
- [18] Hypertherm Associates. Robotmaster. <https://www.robotmaster.com/en>.
- [19] Fanuc. Fanuc roboguide. <https://www.fanuc.eu/pl/pl/roboty/akcesoria/roboguide>.
- [20] RoboDK Inc. Robodk. <https://robodk.com/simulation>.
- [21] KUKA. Kuka.sim. <https://www.kuka.com/pl-pl/produkty-i-us%C5%82ugi/systemy-robot%C3%B3w/oprogramowanie/planowanie-projektowanie-serwis-bezpiecze%C5%84stwo/kuka,-d-,sim>.
- [22] Hexagon AB. Msc adams. <https://hexagon.com/products/product-groups/computer-aided-engineering-software/adams>.
- [23] Nvidia. Isaac sim. <https://developer.nvidia.com/isaac-sim>.
- [24] Microsoft. Microsoft robotics developer studio. https://en.wikipedia.org/wiki/Microsoft_Robotics_Developer_Studio.
- [25] Open Robotics. Gazebo. <https://gazebosim.org/home>.
- [26] ONERA, LAAS-CNRS, and ISAE-SUPAERO. Morse. https://www.openrobots.org/morse/doc/stable/what_is_morse.html.
- [27] Coppelia Robotics, Ltd. Coppeliasim. <https://www.coppeliarobotics.com/>.
- [28] Cyberbotics Ltd. Webots. <https://cyberbotics.com/#features>.
- [29] Columbia University Robotics Group. Graspit! <https://graspit-simulator.github.io/>.
- [30] Open Robotics. urdf - ros wiki. <http://wiki.ros.org/urdf>.
- [31] Nvidia. Isaac sim. https://docs.omniverse.nvidia.com/app_isaacsim/app_isaacsim/requirements.html.
- [32] Krzysztof Tchoń, Robert Muszyński, Alicja Mazur, Ignacy Dulęba, and Robert Hossa. *Manipulatory i roboty mobilne*. Akademicka Oficyna Wydawnicza PLJ, 2000.
- [33] Maria Łopatka, Agata Guzik-Kopyto, Robert Michnik, and Wiesław Rycerski. Badania antropometryczne kończyny górnej oraz pomiar siły ścisku dloni i kciuka. *Aktualne Problemy Biomechaniki*, 6, 2012.

- [34] Ali Fallahi and Ali Jadidian. The effect of hand dimensions, hand shape and some anthropometric characteristics on handgrip strength in male grip athletes and non-athletes. *Journal of Human Kinetics*, 29(2011):151–159, 2011.
- [35] A. Gedliczka and P. Pochopień. *Atlas miar człowieka: dane do projektowania i oceny ergonomicznej : antropometria, biomechanika, przestrzeń pracy, wymiary bezpieczeństwa*. CIOP, 2001.
- [36] Joseph T. Belter and Aaron M. Dollar. Performance characteristics of anthropomorphic prosthetic hands. In *2011 IEEE International Conference on Rehabilitation Robotics*, pages 1–7, 2011.
- [37] ottobock. Bebionic glove. <https://shop.ottobock.us/Prosthetics/Upper-Limb-Prosthetics/bebionic/bebionic-Hands-%26-Gloves/bebionic-Black-Glove/p/8S710>.
- [38] Christoph Borst, Max Fischer, and G. Hirzinger. Calculating hand configurations for precision and pinch grasps. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, volume 2, pages 1553 – 1559 vol.2, 02 2002.
- [39] Shadow Robot Company. Dexterous hand documentation. https://dexterous-hand.readthedocs.io/en/latest/user_guide/3_robot_xacros.html.
- [40] Shadow Robot Company. Simox integration with ros. https://github.com/shadow-robot/simox_ros/.

Dodatek A

Załączona płyta CD

Do niniejszej pracy została załączona płyta CD zawierająca kopię pracy dyplomowej w formacie PDF oraz przeprowadzone symulacje w formacie USD.

W12N_252928_2023_praca_inżynierska.pdf - kopia pracy inżynierskiej

sterowalna_proteza.usd - symulacja protezy ręki w konfiguracji domyślnej

relaks.usd - symulacja zrelaksowanej ręki

piesc.usd - symulacja pięści

pesetowy.usd - symulacja chwytu pesetowego

hakowy.usd - symulacja chwytu hakowego

kulistu.usd - symulacja chwytu kulistego

proba.usd - symulacja walców stworzona w celu sprawdzenia środowiska