

POLITECHNIKA WROCŁAWSKA
WYDZIAŁ ELEKTRONIKI

ROZPRAWA DOKTORSKA

System sterowania robota społecznego

mgr inż. Jan Kędziński

Promotor:
Prof. dr hab. inż. Krzysztof Tchoń

Oliwierowi

*Pragnę serdecznie podziękować promot-
rowi, Panu **Profesorowi Krzysztofowi
Tchoniowi**, za cały poświęcony mi czas, za
inspirację prowadzonych badań oraz pomoc
w tworzeniu niniejszej pracy. Chciałem rów-
nież podziękować przyjaciołom oraz kolegom
z Zakładu Podstaw Cybernetyki i Robotyki za
pomoc w realizacji badań. Chciałbym także
podziękować Pani mgr Marcie Zagdańskiej
za nieocenione wsparcie w zakresie psycholo-
gicznych aspektów prowadzonych badań.*

Spis treści

1	Wstęp	5
1.1	Koncepcja robota społecznego	9
1.1.1	Interakcja międzyludzka	9
1.1.2	Znaczenie twarzy w procesie interakcji	9
1.1.3	Modele emocji i teoria sztucznego umysłu	9
1.1.4	Urzeczywistnienie	12
1.1.5	Definicje i klasyfikacja	14
1.1.6	Interakcja człowiek-robot	18
1.2	Współczesny stan wiedzy	19
1.2.1	Przegląd istniejących konstrukcji	19
1.2.2	Obszary prowadzonych badań HRI	19
1.2.3	Techniki przeprowadzania eksperymentów HRI	20
2	Cel pracy	23
3	Robot FLASH implementacja sprzętowa	29
3.1	Ogólne parametry techniczne	29
3.2	Platforma jezdna	30
3.3	Korpus wraz z ramionami	35
3.4	Głowa EMYS	39
4	Platforma balansująca – model i algorytmy sterowania	43
4.1	Model dynamiki robota balansującego	44
4.1.1	Kinematyka	45
4.1.2	Dynamika	45
4.2	Sterowanie liniowe	48
4.2.1	Wyznaczenie przybliżenia liniowego w punkcie równowagi	48
4.2.2	Koncepcja sterownika liniowego	49
4.2.3	Sterownik minimalno-kwadratowy (LQR)	50
4.2.4	Sterownik rozmieszczający bieguny (PP)	51
4.2.5	Sterownik PD	51
4.2.6	Zastosowanie liniowego sterownika do modelu nieliniowego	52
4.3	Sterowanie nieliniowe – metoda linearyzacji	53
4.3.1	Śledzenie pozycji robota	53
4.3.2	Śledzenie trajektorii orientacji i odchylenia korpusu od pionu	55
4.4	Sterowanie nieliniowe – metoda endogenicznej przestrzeni konfiguracyjnej	56
4.4.1	Sterowanie z ograniczeniami	60
4.4.2	Sterowanie z priorytetami	61
4.4.3	Sterowanie z podziałem na kroki	63

4.5	Sterowanie nieliniowe – zadanie sterowania optymalnego	67
4.5.1	Planowanie ruchu w otwartej pętli	67
4.5.2	Planowanie ruchu w zamkniętej pętli	69
4.6	Pomiar odchylenia kąтового w robocie balansującym	70
4.6.1	Wyznaczenie odchylenia podstawie pomiaru przyspieszeń	72
4.6.2	Filtr Kalmana – wprowadzenie	73
4.6.3	Rozszerzony filtr Kalmana (EKF)	75
4.6.4	Bezśladowy filtr Kalmana (UKF)	77
5	System sterowania robota FLASH	81
5.1	Koncepcja	81
5.2	Warstwa integrująca	83
5.3	Warstwa najniższa - moduły oprogramowania	84
5.3.1	Moduł nawigacyjny	84
5.3.2	Moduł sterowania napędami	84
5.3.3	Moduły przetwarzania dźwięku	85
5.3.4	Moduły przetwarzania obrazu	85
5.3.5	Moduł przetwarzania 3D	86
5.3.6	Moduły odpowiadające za uczenie się	87
5.3.7	Pozostałe moduły	88
5.4	Warstwa pośrednia - implementacja kompetencji	88
5.4.1	Kompetencje grupy audio	89
5.4.2	Kompetencje grupy video	89
5.4.3	Kompetencje grupy ml (<i>machine learning</i>)	90
5.4.4	Kompetencje grupy body (platforma)	91
5.4.5	Kompetencje grupy arm i hand	91
5.4.6	Kompetencje grupy head	93
5.5	Warstwa zarządcy kompetencji	98
5.6	Interfejs programistyczny	100
5.6.1	Separatory (;) ()	100
5.6.2	Separatory (,) (&)	100
5.6.3	Pętla warunkowa (whenever)	101
5.6.4	Wystąpienie (at) i obsługa zdarzeń (Event)	101
5.6.5	Warunek blokujący (waituntil)	102
5.6.6	Pozostałe instrukcje	103
5.6.7	Etykietowanie kodu (Tag)	103
5.6.8	Generatory trajektorii	103
5.6.9	Drzewiasta struktura (robot)	103
5.7	Warstwa najwyższa	105
5.7.1	Zdalne sterowanie	105
5.7.2	Automat skończony	105
5.7.3	Zaawansowane oprogramowanie	106
6	Weryfikacja eksperymentalna	111
6.1	Eksperyment HRI z głową EMYS	111
6.1.1	Opis eksperymentu	112
6.1.2	Analiza danych	113
6.1.3	Wyniki	113
6.2	Eksperyment HRI z robotem FLASH	115

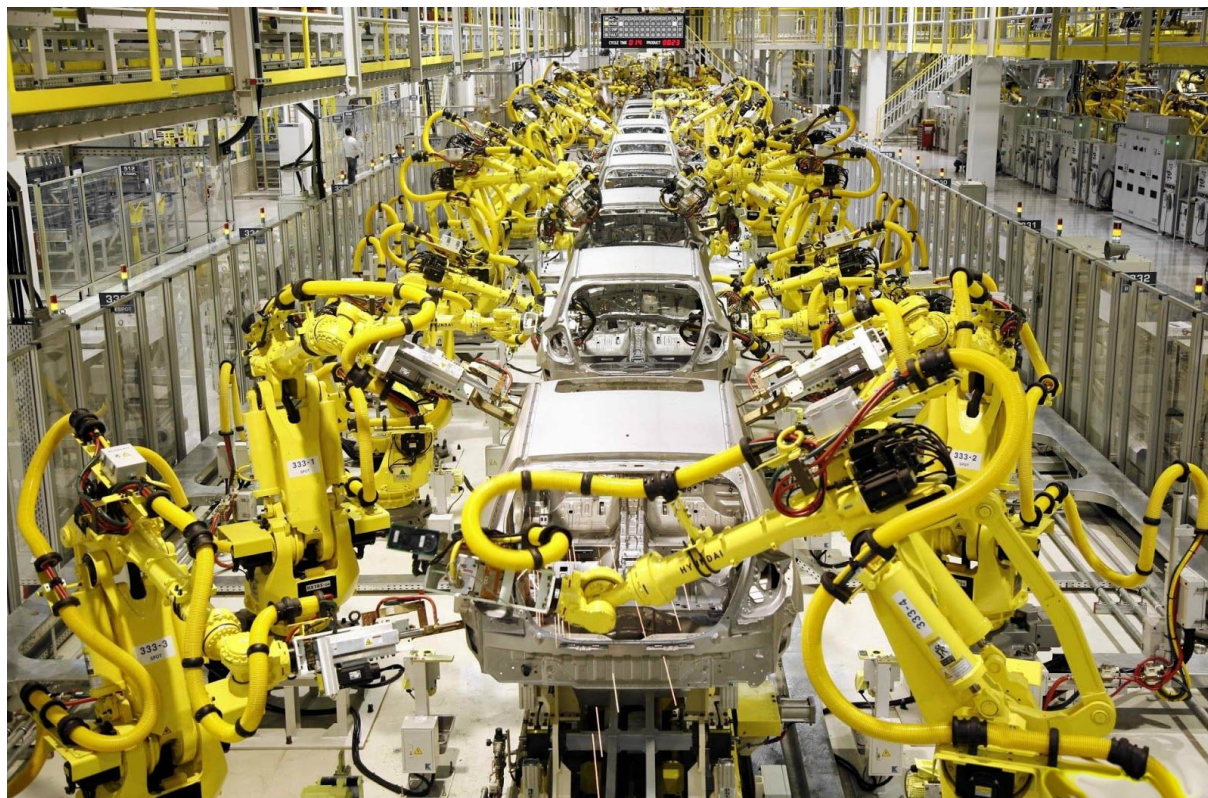
6.2.1	Opis eksperymentu	116
6.2.2	Analiza danych	116
6.2.3	Wyniki analizy danych okulograficznych	117
6.2.4	Wyniki analizy danych ankietowych	120
6.2.5	Wyniki analizy wywiadów pogłębionych	125
7	Uwagi końcowe	127
7.1	Dalsze plany rozwojowe	128
7.2	Obszary możliwego wykorzystania robota FLASH i jego głowy EMYS . . .	130
7.3	Aspekty dotyczące komercjalizacji opracowanego stanowiska badawczego .	132
7.4	Słowo końcowe	132

Rozdział 1

Wstęp

Historia pokazuje, że zawsze byliśmy zależni od maszyn, by przetrwać. Wygląda na to, że los jest ironiczny. – Morfeusz, „Matrix”, reż. L. i A. Wachowscy.

Maszyny od wieków towarzyszą ludziom w ich codziennych czynnościach. Służą nam w domu, pracy i w podróży. Nie zawsze jesteśmy świadomi tego, kiedy i gdzie pracują, abyśmy mogli odpoczywać. Z powodzeniem można stwierdzić, że ich obecność jest dziś konieczna do tego, żeby człowiek mógł prawidłowo funkcjonować. Dostarczają energii do naszych domów, pozwalają się przemieszczać i wytwarzać niezbędne do przeżycia dobra. W robotyce maszyny te, to niezawodne automaty wspomagające procesy produkcyjne. Projektowane i budowane z myślą o wykonywaniu konkretnych zadań, pozwalają na zwiększenie wydajności fabryk, czy też uzyskanie lepszej jakości produktów. Inteligentne, uniwersalne, zaprojektowane do automatycznego wykonywania wielu zadań roboty przemysłowe zrewolucjonizowały przemysł na całym świecie.



Rysunek 1.1: Roboty przemysłowe w trakcie czynności spawalniczych

Postępowa rola automatyzmu polega na tym, że dzięki niemu, pewne cenne procesy odbywają się same, bez kierowniczej interwencji podmiotu działającego, który jest odciążony od pewnych robót i może skupić cały swój świadomy wysiłek na czymś innym. „Traktat o dobrej robocie”, T. Kotarbiński

Producentom działającym w różnych gałęziach przemysłu nie wystarcza już zwiększenie produkcji i jakości wyrobów, muszą także obniżyć koszty ich wytwarzania. Wykonywanie manualnie prostych i powtarzalnych operacji przestało być opłacalne. Szacuje się, że obecnie na świecie pracuje ponad milion robotów przemysłowych, a ich liczba wciąż rośnie. Trudno sobie dziś wyobrazić proces produkcyjny, który nie mógłby być zautomatyzowany.

Robot – urządzenie zastępujące człowieka przy wykonywaniu niektórych czynności. „Uniwersalny słownik języka polskiego”, PWN.

Nie pojawiła się do tej pory jasna i powszechnie zaakceptowana definicja robotów. Istnieje jednak definicja robotyki, z której wynika czym jest robot – nauka o inteligentnym wykorzystaniu percepcji do działania [29]. Większości z nas, robot kojarzy się z humanoidalną konstrukcją typu android. Co gorsza, utarło się w nas przekonanie, że tego typu jednostki mogą stwarzać zagrożenie dla społeczeństwa. Ludzie przestali, a być może nawet nigdy nie zaczęli, postrzegać automaty jako bezduszne urządzenia mechaniczne, lecz jako twory mogące w przyszłości zagrozić ludzkości. Z punktu widzenia różnych definicji, roboty cechuje przede wszystkim: zastępowanie człowieka, autonomiczność, „inteligencja”, mobilność, manipulacja, zastosowanie w przemyśle. Rozsądek podpowiada, że cechy te mają wydzźwięk pozytywny i mogą stanowić pewną formę wsparcia dla naszego gatunku. Skąd zatem biorą się w nas tak skrajne skojarzenia?

Jam jest robot hartowany, zdalnie prądem sterowany, nitowany z każdej strony, wyklepany, uzwojony,... „Bajki robotów”, S. Lem

Niezwykłe dynamiczny rozwój technologii znalazł odzwierciedlenie w literaturze oraz filmach science-fiction. Opowiadania Isaaca Asimova, sztuka Karel Čapeka czy też słynna seria filmów Gwiezdne Wojny, którą zapoczątkował George Lucas na dobre ukształtowały w nas wyobrażenie o inteligentnej, obdarzonej ogromną siłą fizyczną, sztucznej formie życia, jaką jest robot.

*Jesteś tylko maszyną. Imitacją życia. Czy robot potrafi napisać symfonię?
Potrafi przemienić płótno w arcydzieło? – detektyw Spooner
A pan potrafi? – robot Sonny, „Ja, robot”, reż. A. Proyas*

Artyści kreują roboty na wzór ludzkiej istoty, wyposażają w uczucia, sny, oraz zmysły jak dotąd zarezerwowane jedynie dla ludzi. Niestety, zwykle biorą one udział w konfliktach pomiędzy maszyną a człowiekiem, nierzadko są również ich powodem. Początkowo są budowane z zamiarem, aby współdziałać z człowiekiem. Jednak ich systemy sterowania wymykają się spod kontroli i przekształcają robota w tyrana pragnącego ludzkiej zagłady. Nie są nas w stanie uchronić trzy prawa robotyki zaproponowane przez Isaaca Asimova. Na wielkim ekranie wielokrotnie przewijają się scenariusze, w których nagle przestają działać wszystkie telefony i Internet. Z nieba spadają samoloty, a pozbawione kierowców samochody rozpoczynają polowanie na ludzi. Coś przejmuje kontrolę nad komputerami, maszynami, elektrowniami i obroną wojskową. Cała technologia zwraca się przeciwko nam. Pojawia się nieznany dotąd wróg, którego sami stworzyliśmy. Takie wizje znamy z trylogii filmów Matrix, czy z serii filmów Terminator.



Rysunek 1.2: Roboty z filmów „Ja, robot”, „Terminator”, „Gwiezdne wojny”

Nie istnieje jeszcze robot, który potrafi założyć poszewkę na poduszkę. – M. Minsky

W rzeczywistości, jesteśmy dalecy od posiadania tak zaawansowanych technologii. Dotąd na świecie powstały nieliczne prototypy, wyposażone w sztuczne umysły na poziomie dwuletniego dziecka, które nie są w stanie nam w żaden sposób zagrozić. Choć posiadają sztuczną inteligencję, to nie potrafią wykonywać wielu podstawowych czynności. Zatem, w jakim celu buduje się tego typu konstrukcje? Czy nie wystarczy, że ponad milion robotów w zamkniętych fabrykach na całym świecie produkuje dla nas dzemy, szczoteczki do zębów, łożyska kulkowe, skarpetki, i wiele innych rzeczy? Faktem jest, że inteligentne maszyny na dobre zadomowiły się w naszych mieszkaniach. Producenci sprzętu AGD uczestniczą w wyścigu technologicznym twierdząc, że ich urządzenia są coraz sprytniejsze. Pralki decydują, jaki program prania i oszczędzania wody wybrać dla zawartości bębna. Piekarniki dostosowują tryb pieczenia do zaproponowanej potrawy, a odkurzacze zdolne są do samodzielnego odkurzenia naszego salonu. Nie są to jednak roboty, jakie przedstawiono powyżej, a już na pewno w żaden sposób nie przypominają żywej istoty.

*Elektryczne zwierzęta też mają swoje życie. Bez względu na to, jak jest nędzne.
„Czy androidy marzą o elektrycznych owcach?”, P. K. Dick*

Ludzkość zmierza w kierunku zrozumienia i zaakceptowania technologii. Pragniemy aby była ona, w naturalny dla nas sposób, intuicyjna i łatwa w obsłudze. Z tego powodu mamy tendencję do antropomorfizowania naszych komputerów, samochodów, i innych gadżetów [176]. Z drugiej strony, liczne badania marketingowe pokazują, że ludzie szybko przywiązują się do przedmiotów codziennego użytku [146, 193]. Mają również skłonność do przywiązania się do przedmiotów nie pełniących funkcji użytkowych, tylko i wyłącznie ze względu na ich walory estetyczne lub sentymtalne [194]. Istnieją zatem powody, dla których warto projektować i konstruować urządzenia społecznie interaktywne. Rozwijając je tak, aby mogły być nie tylko użyteczne, ale zdolne do wzajemnego oddziaływania z człowiekiem.

Ubuntu to esencja bycia człowiekiem. Jestem człowiekiem, bo przynależę do społeczeństwa i moje człowieczeństwo jest powiązane z Twoim – biskup D. Tutu.

Wprowadzenie robotów do środowiska człowieka nie tylko wiąże się z rozwojem technologii, w tym układów sterowania, rozwiązań mechatronicznych, czy sensorycznych. To przede wszystkim nauka o znaczeniu człowieczeństwa, o kulturze, komunikacji i stosunkach międzyludzkich. Silnie dotyka takich pojęć jak osobowość, tożsamość, emocje, empatia i wielu innych stanowiących podstawy nauk psychologiczno-społecznych [34]. Niezwykle ważnymi aspektami związanymi z projektowaniem i budową robotów do pracy w środowisku ludzkim stanowią komunikacja oraz percepcja. W tym kontekście, percepcja robota



Rysunek 1.3: Roboty Kismet, Nao, Asimo

skupia się głównie na człowieku, a człowieka na robocie i jego zachowaniu. Naturalne dla nas środki komunikacji, werbalne jak i niewerbalne, stanowią dziś ogromne wyzwanie nowej dziedzinie robotyki – robotyki społecznej.

Im bardziej zaawansowane technicznie (doskonalsze!) medium, tym bardziej prymitywne, błahe i bezużyteczne wiadomości są przy jego pomocy przekazywane – S. Lem

Konstruktorzy z różnych ośrodków badawczych pracują nad urzeczywistnieniami powyższych koncepcji. Rozwijają przy tym nie tylko możliwości manipulacyjne czy mobilne, lecz także pracują nad mechanizmami zdolnymi do wyrażania gestów i emocji, systemami dialogowymi, percepcją, w tym utrzymywanie kontaktu wzrokowego, śledzeniem człowieka, rozpoznawaniem gestów oraz zdolnościami do empatii. Sztuczny umysł powinien dostarczać odpowiednich mechanizmów pozwalających robotowi na interpretację zachowań człowieka i adekwatne podejmowanie zadań, planowanie działań na podstawie budowanego modelu człowieka. Fizyczne urzeczywistnienie powinno odzwierciedlać faktyczne umiejętności robota. Gdy dojdzie do zachwiania tej równowagi, robot może być nie w pełni zaakceptowany lub nawet wywołać odrazę.

*Zabaweczki. Żywe lalki. Oczywiście wszyscy wiedzą, po co je skonstruowano.
Mają skraść wasze serca. Mające zastąpić wam dzieci...
– konferansjer, „A.I.”, reż. S. Spielberg*

1.1 Koncepcja robota społecznego

1.1.1 Interakcja międzyludzka

U podstaw rozumienia koncepcji robota społecznego leży silnie zakorzeniona w nas potrzeba interakcji międzyludzkiej. W tej kwestii neurologowie są zgodni, że ludzkie mózgi i umysły ukształtowane są tak, aby być w ciągłej interakcji z drugim człowiekiem i stale do niej dążyć. Nie tylko w sensie fizycznym, lecz także i psychologiczny wizerunek drugiej osoby może mieć wpływ na działanie naszego umysłu. W skutek czego poprzez interakcję wpływamy na swoje zachowanie. Definiując pojęcie interakcji, należy uwzględnić bliskie nam procesy, takie jak: percepcja i ekspresja, intencjonalność i perswazja, komunikacja werbalna i niewerbalna. Komunikacji werbalnej odpowiada naturalna forma prowadzenia dialogu, a niewerbalna jest to zdolność do wyrażania i postrzegania mimiki twarzy, przede wszystkim emocji. Interakcja między dwoma osobami jest silnie zależna od ich osobowości, pochodzenia, rozwoju, utartych schematów społecznych i stereotypów, stylu przywiązania oraz presji jakiej były poddawane [103]. Zjawisko interakcji i jej znaczenie dla procesów poznawczych przedstawił już w latach 30-tych radziecki psycholog Lew Wygotski [255]. Jej reguły oraz rodzaje w szerokim zakresie opisywali w latach 60-tych E. Goffman [90], J. Condon [45] oraz E. T. Hall [100]. Na potrzeby opracowania koncepcji robota społecznie interaktywnego, wielu naukowców posługuje się wyżej wymienionymi teoriami, z sukcesem adoptując je do opisu sytuacji, w których dochodzi do interakcji człowieka z robotem.

1.1.2 Znaczenie twarzy w procesie interakcji

Niezwykle ważną rolę w procesie interakcji odgrywa twarz. Jej szczególne znaczenie, m.in. dla mechanizmów identyfikacji, podkreślono w pracy [103]. Elementarne umiejętności rozpoznawania twarzy obserwuje się już u noworodków, co wskazuje, że człowiek rodzi się z wrodzonymi dyspozycjami, które doskonalą się w procesie rozwoju. Przetwarzanie tych informacji stało się możliwe dzięki wykształceniu wyspecjalizowanego systemu neuronalnego. Liczne badania pokazują, że dostrzegane w naszym otoczeniu twarze, na grafikach, zdjęciach, animacji i filmach, intensywnie skupiają na sobie wzrok widza. Zjawisko to zaobserwowano nawet na próbkach zawierających niezwykle atrakcyjne i interesujące treści. Dowiedziono także, że twarz przyciąga wzrok obserwatora, nawet jeśli ten jest pobudzany przez zupełnie inne bodźce. Przykładowe wyniki badań zaprezentowano na rysunku 1.4. Fakt ten regularnie wykorzystuje się komercyjnie, np. w marketingu. Twarz wypełnia werbalne i niewerbalne kanały komunikacji, jest pojemnym nośnikiem informacji o człowieku, o jego stanie emocjonalnym, osobowości, zdrowiu, cechach charakteru, pochodzeniu i wielu innych. Dzięki niej możliwe staje się przewidywanie zachowań człowieka [70]. Ponadto, mimika jest bezcenną wskazówką, pozwalającą właściwie zinterpretować przekaz werbalny. Wyraz oczu, układ warg czy zmarszczenie nosa to naładowane emocjonalnie sygnały umożliwiające międzygatunkową wymianę informacji. Ten pierwotny kod pozwala nam przewidzieć zarówno zachowanie innych osób, jak i zwierząt. Każdy z nas widząc odsłonięte kły i zmrużone oczy psa błyskawicznie cofa rękę, by ochronić się przed ugryzieniem.

1.1.3 Modele emocji i teoria sztucznego umysłu

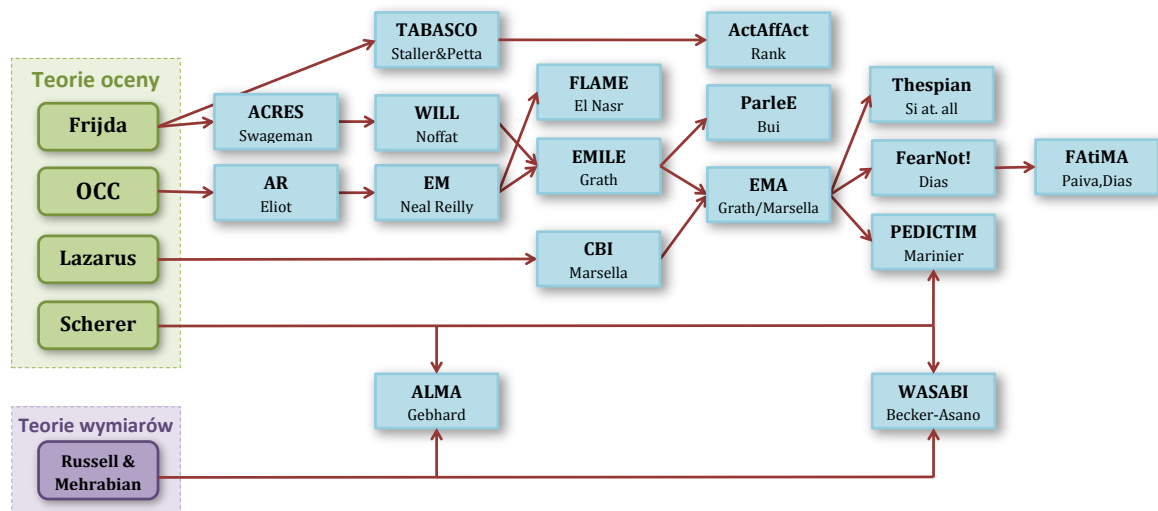
O sztucznej inteligencji naukowcy rozważali od początku drugiej połowy XX wieku. Jednak prawdziwy przełom nastąpił po roku 1950, kiedy Alan Turing jako pierwszy próbował



Rysunek 1.4: Przykładowe badania w postaci map cieplnych zaczerpnięto z CHL (www.eyetracker.com.pl)

formalnie zdefiniować, czym jest sztuczna inteligencja [249]. Przez kolejne dwadzieścia lat trwały intensywne prace nad SI. Chwilę później, Amerykanie Walter Pitts i Warren McCulloch przedstawiają załączki znanej dziś teorii sieci neuronowych. Rozwój technologii, a przede wszystkim rosnąca moc obliczeniowa komputerów, rozbudziły wielkie nadzieje na stworzenie prawdziwej SI. Niestety, ostatecznie zdano sobie sprawę z szeregu ograniczeń. Stwierdzono, że do osiągnięcia widocznych rezultatów potrzebna będzie całkowita zmiana myślenia. Na przełomie lat siedemdziesiątych i osiemdziesiątych, pewne ograniczenia udało się obejść tworząc systemy eksperckie. Pojawienie się języka PROLOG [185], którego twórcą jest Alain Colmerauer, pozwoliło na automatyczną analizę języka naturalnego. Systemy eksperckie, dostatecznie sprawnie potrafiły emulować procesy decyzyjne człowieka, będącego ekspertem w danej dziedzinie. Dziś dysponujemy maszynami o ogromnej mocy obliczeniowej i zasobach pamięci. To, co czterdzieści lat temu uchodziło za ograniczenie, dziś wydaje się nie istnieć. Jednakże, trudno mówić o istnieniu prawdziwej sztucznej inteligencji. Inteligentne budynki i autonomiczne samochody działają dzięki zaawansowanemu oprogramowaniu, które analizują dane z ogromnej ilości skomplikowanych sensorów. Słynny specjalista od SI Marvin Minsky stwierdził, że problemem obecnej sztucznej inteligencji jest skupianie się na pojedynczych problemach, w celu ich rozwiązania. Dlatego tworzenie systemów sztucznej inteligencji skupiających się na zbiorach specjalistycznych reguł jest błędnym podejściem. Potrzebne są systemy posiadające ogólną wiedzę o świecie, wraz z regułami rozumowania na podstawie tej wiedzy. Istnieją setki opracowań, na podstawie których można wnioskować o teorii sztucznego umysłu. Trudno je wszystkie wymienić, trudno nawet wskazać te przełomowe. Warto natomiast skupić się na teoriach poświęconych próbom stworzenia sztucznej osobowości.

Fizyczne urzeczywistnienie, w którym zostanie osadzony sztuczny umysł to tzw. autonomiczny agent. Postrzega on środowisko i, oddziałując na nie w czasie, realizuje przy tym własne cele, tak aby wpłynąć na to, co odbierze w przyszłości [81]. W literaturze z dziedziny robotyki spotyka się pojęcie agenta społecznie inteligentnego (*SIA - Socially Intelligent Agent*) [199, 50, 5]. Pod tym pojęciem kryje się obiekt, który okazuje pewne elementy ludzkiej inteligencji. Mowa tu nawet o najprostszych zachowaniach społecznych.



Rysunek 1.5: Rozwój modeli umysłu afektywnego (na podstawie pracy [166])

Markowanie osobowości, emocji, empatii, prowadzenie konwersacji w oparciu o komunikację werbalną i niewerbalną, wymaga posiadania wbudowanych mechanizmów reagowania i działania emocjonalnego. Ten istotny fakt determinuje jego afektywny charakter oraz konieczność podejmowania interakcji. Powinien pozwalać na naturalną reakcję na bodźce, dokonywać prawidłowej percepcji człowieka, komunikować się w naturalny dla nas sposób. Komunikaty niewerbalne, takie jak gestykulacja, mimika, kontakt fizyczny, dystans przestrzenny, pozycja ciała służą przekazywaniu znaczeń, ilustrują treść wypowiedzi, regulują przebieg rozmowy i stosunki między rozmówcami, pozwalają dostosować się do sytuacji. Wspomniane już zjawisko interakcji międzyludzkiej daje się stymulować przy pomocy robotów lub postaci animowanych, spełniających pewne założenia. Ich sztuczne umysły powinny być zdolne do tworzenia modelu człowieka posiadającego stan emocjonalny, temperament, charakter, osobowość, mentalność, przekonania i wiele innych. Stworzenie takiego systemu brzmi nieco utopijnie, ale w dziedzinie kreowania umysłów afektywnych naukowcom udaje się osiągać pewne sukcesy, o których można przeczytać m.in. w książkach [247], [166] oraz w pracy [155].

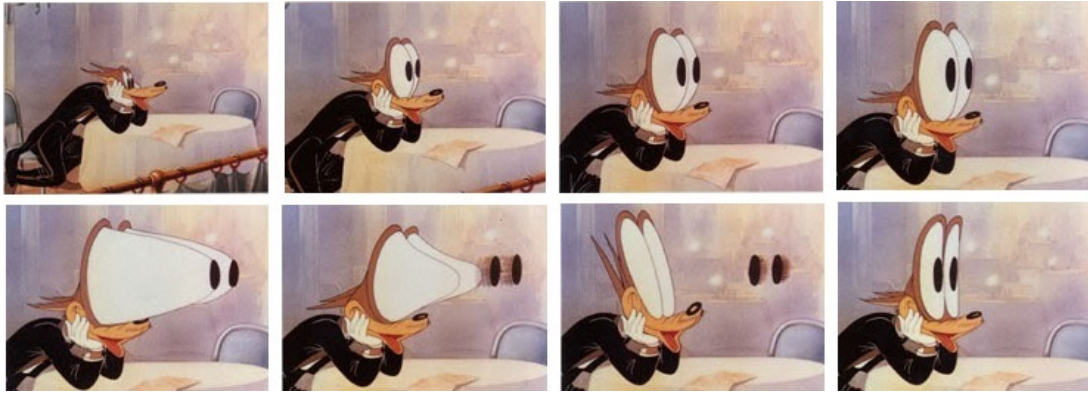
W dziedzinie modelowania emocji, dominują obecnie teorie oceny, być może dlatego, że zainteresowani są nimi głównie twórcy sztucznej inteligencji. Wg tej teorii, emocje wynikają z poszczególnych ocen dotyczących relacji między zdarzeniami, a indywidualnymi przekonaniami, intencjami, pragnieniami, czasami są określane jako relacja człowiek-otoczenie. Jedną z najpopularniejszych jest teoria Ortony'ego, Clore'ego i Collinsa (OCC) [191], która została stworzona specjalnie z myślą o wykorzystaniu w systemach sztucznej inteligencji. W teorii tej emocje definiowane są jako reakcja na zdarzenia. Do powstania obliczeniowych modeli emocji przyczyniły się także teoria Rosemana [221, 220] opis procesów Scherera [224] oraz teoria Frijda [84]. Stworzenie syntetycznych postaci (avatarów) było możliwe także dzięki teorii poznawczej Lazarusa [230, 151] oraz teorii modelowania emocji PSI [65, 16, 18] wraz z jej implementacją MicroPSI [15, 105]. Nieco inne podejście do modelowania emocji przedstawiają teorie wymiarów, które zakładają, że emocje nie powinny mieć reprezentacji dyskretnych, ale być rozpięte na przestrzeniach dwu, trój lub czterowymiarowych. Twórcy tych teorii twierdzą, że dyskretne kategoryzowanie emocji (strach, radość, smutek,...) ma niewiele wspólnego z rzeczywistością, gdyż mózg nie jest

podzielony na obszary, które odpowiadają za poszczególne emocje. Teoria ta, pod względem złożoności, jest stosunkowo prosta i często wykorzystywana w animacji i systemach rozpoznawania emocji. Wiele z nich bazuje na opracowaniach Mehrabiana i Russella [173]. Na szczególną uwagę zasługują modele PAD (Pleasure, Arousal, Dominance) [20], stworzony głównie z myślą o wykorzystaniu w trakcie komunikacji niewerbalnej oraz zbliżony ideologicznie ASV (Arousal, Valence, Stance) [34, 177]. Ciekawą koncepcję trójwymiarowej przestrzeni przedstawił Plutchnik w pracy [202]. Przestrzeń tę, przyrównał do przestrzeni kolorów HSV. Znaczący wkład w dziedzinie modelowania emocji mieli również Ekman i Friesen [68], którzy opracowali system kodowania wyrazów twarzy oraz zestaw podstawowych emocji.

Zastosowanie powyższych teorii, rozwój gier komputerowych [112] oraz potrzeba wyrażania nastroju animowanych postaci [197] dały początek istnieniu sztucznych umysłów, zdolnych do naturalnego generowania emocji. Niektóre z nich, takie jak system FLAME [72] do reprezentacji emocji stosuje logikę rozmytą. EMA [94, 95, 165] to bardzo popularny system bazujący na teoriach Smitha i Lazarusa. Do szczegółowej predykcji, jak ludzie oceniają i radzą sobie z emocjonalnymi sytuacjami EMA używa koncepcji Gratcha i Marsella [95]. Implementacje WASABI [21] oraz ALMA [88] wykorzystują zarówno teorie OCC oraz PAD. ALMA potrafi symulować emocje, nastrój oraz osobowość agenta. Ostatni system to FATiMA [60, 62, 14], który jest wciąż rozwijany, bazuje na projekcie FearNOT! [61] oraz EMA. U jego podstaw leżą kognitywna teoria OCC, teoria Lazarusa oraz PSI. Przedstawione systemy oraz ich chronologiczny proces powstawania przedstawiono na rysunku 1.5.

1.1.4 Urzeczywistnienie

Podjmując się próby scharakteryzowania fizycznego urzeczywistnienia dla robota społecznego, należy wziąć pod uwagę wszystkie wyżej wymienione aspekty związane z podejmowaniem interakcji międzyludzkiej. Chcąc zaprojektować i wykonać ciało sztucznego agenta, powinno się mieć na uwadze to, że do pewnego stopnia jest to próba stworzenia sztucznego człowieka. Na tym polu znaczącego postępu dokonali projektanci grafiki komputerowej oraz twórcy kreskówek [163, 130]. Słynni animatorzy Disneya, Frank Thomas i Ollie Johnston opracowali zbiór [242] podstawowych reguł, jakimi powinno się kierować projektując postacie animowane. W pracy tej wprowadzili pojęcie wiarygodności (*believability*), które należy rozumieć jako adekwatne do ludzkiej percepcji zachowanie wyglądu oraz zachowań robota. Obserwator powinien być w stanie dostrzec w nim elementy ludzkich zachowań, przejawów osobowości, stanu psychicznego oraz przewidzieć jego postępowanie. Mistrzami antropomorfizowania natury, a nawet nieożywionych obiektów są nie tylko animatorzy Disneya, ale także graficy z w wytwórni Pixar Animation Studios – producenci filmów takich, jak „Toy Story”, „Auta”, „Wall-E” i wielu innych. Tex Avery pracujący przez wiele lat dla wytwórni Warner Bros, twórca Królika Bugsa i Kaczora Duffiego, również stworzył pewne wytyczne [39], którymi kierują się twórcy kreskówek. To właśnie jego unikalnemu stylowi zawdzięczamy wyskakujące z orbit oczy (rysunek 1.6), dziwaczne rozciągnięcia i transformacje przeczące prawom fizyki oraz popularną do dziś przesadną ekspresję. Równie ważną postacią był amerykański artysta, plastyk i lalkarz Jim Henson twórca głównych postaci występujących w serialach telewizyjnych „Muppet Show” oraz „Ulica Sezamkowa”. Jego dzieła charakteryzują się ograniczoną ilością środków wyrazu. Często posiadają one nieruchome oczy oraz proste, pozbawione kącików usta, które mogą się jedynie zamykać i otwierać. Niemniej jednak, animatorzy teatralni poprzez odpowiednią postawę ciała oraz dynamikę ruchu, bez problemu wyrażają dowolne emo-

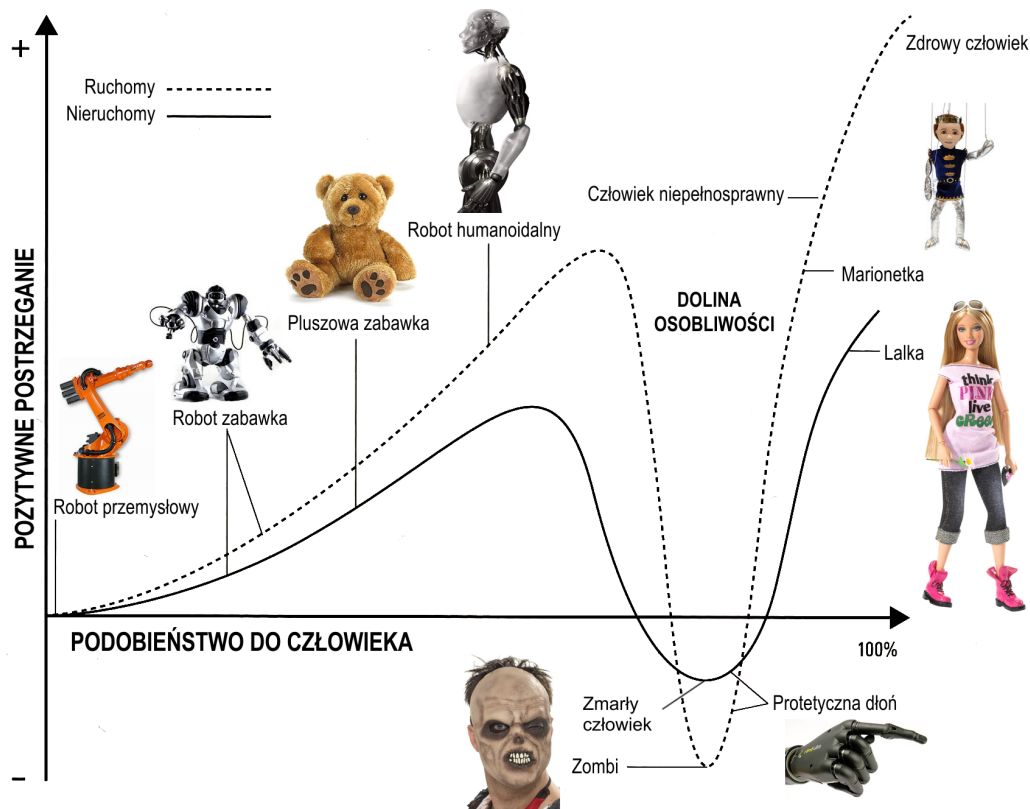


Rysunek 1.6: Słynny wytrzeszcz oczu Texa Averego

cje [212]. Wiele z tych założeń zostało w pełni zaadoptowanych na potrzeby opracowania wytycznych, jak projektowanie wyglądu, wizerunku oraz ruchu robotów społecznych.

Istnieje wiele powodów, by fizyczne konstrukcje były inspirowane naturą, aby tworzyć roboty humanoidalne lub zoomorficzne. Ich twarze powinny posiadać ludzkie elementy (oczy, usta, uszy, . . .), być zdolne do wyrażania gestów, emocji i mówić ludzkim lub możliwym do zrozumienia przez człowieka głosem. Roboty powinny poruszać się i z wyglądu przypominać postać człowieka [49]. Cechy te pozwalają na stymulowanie i podtrzymanie naturalnie zachodzących interakcji człowieka z robotem. W pracy [268] wykazano, jak bardzo wygląd robota wpływa na jego postrzeganie. Autorzy prac [266, 89, 142] podkreślają, że warto projektować roboty w kierunku konstrukcji humanoidalnych, unikać rozwiązań typowo mechanoidalnych. Z kolei, w pracy [105] zbadano jak wygląd robota wpływa na ludzi wykonujących różne zadania wspólnie z robotem. Autorzy tych prac wskazują, że użytkownicy robotów mechanoidalnych traktują je jak sprzęt domowego użytku. Oczekiwania wobec tej grupy robotów są na ogół niewielkie. Jednak równie niedobrze jest, gdy robot jest zbyt podobny do człowieka. Powstające na świecie androidy, do złudzenia przypominają ludzką postać. Wobec takich konstrukcji bardzo szybko rosną oczekiwania. Niestety ich sposób poruszania się oraz zaimplementowane umiejętności mocno rozczarowują. Zachowanie robota jest dalekie od tego, jak to robi człowiek. Tego typu konstrukcja zniechęca do podejmowania interakcji, a nawet budzi w użytkownikach lęk.

Zjawisko to przewidział już w 1970 roku Masahiro Mori. Opublikował swój słynny dziś artykuł w mało znanym japońskim czasopiśmie [179]. Na początku nie wzbudzał dużego zainteresowania, jednak po kilku latach zainteresowali się nim naukowcy, zajmujący się robotyką oraz graficy komputerowi. Został wielokrotnie przetłumaczony z języka japońskiego, ale jego pełne angielskie tłumaczenie wraz z autoryzacją autora pojawiło się w 2012 w magazynie [159]. Mori w swojej pracy przedstawił wykres zależności podobieństwa robota do człowieka, do jego postrzegania (rysunek 1.7). Początkowo funkcja ta jest zdecydowanie rosnąca. Jednak gdy podobieństwo staje się zbyt duże, gwałtownie spada pozytywne postrzeganie robota, by na koniec osiągnąć poziom maksymalny. Spadek ten, autor nazwał uncanny valley, czyli dolina osobliwości, tłumaczona też przez grafików jako dolina niesamowitości. Ponadto Mori wskazał na podobną zależność dla robotów będących w ruchu. Wykres ten charakteryzuje się większą dynamiką. Dolinę osobliwości autor tłumaczy jako postrzeganie robota jak niepełnosprawnego, upośledzonego lub nawet zmarłego człowieka. Po śmierci stajemy się zimni, nieprzyjemni w dotyku, pozbawieni ruchu, a twarz staje się blada. Prawdopodobnie przyczyną tego zjawiska jest nasz instynkt samozachowawczy, który chroni nas od niebezpieczeństwa. Wykres ten wzbudził pewne kontrowersje, gdyż kilku naukowcom udało się zaprzeczyć prawej stronie wykresu



Rysunek 1.7: Dolina osobliwości

Moriego. Robotycy Hanson [102] czy Bethal i Murphy [23] twierdzą, że nawet mecha-noidalne konstrukcje pozbawione humanoidalnej estetyki, także mogą podlegać zjawisku uncanny. Może to wynikać ze sposobu ich poruszania się, który również powinien być projektowany w możliwie przyjazny dla człowieka sposób. Goetz w pracy [89] podkreśla, że wygląd robota musi być także dostosowany do zadań, jakie będzie wykonywał.

Podsumowując, bezpieczniej jest projektować i wykonywać roboty humanoidalne o karykaturalnych kształtach, inspirować się postaciami animowanymi lub zwierzętami. Podo-bieństwo do człowieka nie jest wskazane, jeśli nie zostanie zachowana spójność wyglądu, umiejętności i zachowania. Pewną klasyfikację oraz wskazówki na temat projektowania i konstruowania sztucznych twarzy zawarto w pracy [63]. Niewątpliwie cenną książką jest opracowanie poświęcone konstrukcji robota Kismet [34]. Wiele aspektów budowy i klasyfikacji robotów interaktywnych poruszono w pracy [80]. Na temat unikania doliny osobliwości można przeczytać w pracy [261].

1.1.5 Definicje i klasyfikacja

Jak dotąd, w niniejszej pracy posługiwano się terminem robot interaktywny. Termin ten został po raz pierwszy użyty w artykule [80] nie po to, aby stworzyć nową podgrupę robotów społecznych, lecz aby podkreślić, że interakcja społeczna odgrywa w nich kluczową rolę. Definicja robota społecznego nie została ostatecznie sprecyzowana. Toczą się dyskusje nad tym, jakie cechy i umiejętności powinien on posiadać. Jedne z pierwszych sformułowañ na temat robotów serwisowych, współpracujących z ludźmi pojawiły się w pracy [75]. Autorzy opisują roboty jako inteligentne, programowalne urządzenia, zdolne do odczuwania i działania z korzyścią dla człowieka lub w celu poprawy jego wydajności. Niestety

definicja ta nie podkreśla znaczenia interakcji robot-człowiek. Takie roboty jak Sony Aibo [3], iCat [114] czy Nao [182] nie zostały stworzone w celu zwiększenia produktywności człowieka, a są klasyfikowane jako roboty społeczne.

Początkowo termin robot społeczny odnoszono do systemów wielorobotowych, których działanie było inspirowane zachowaniem zwierząt stadnych jak insekty, ptaki, ryby. Są to proste, nieskomplikowane roboty komunikujące się ze sobą i poruszające na wspólnym obszarze, zdolne do realizowania trudnych zadań, które dla jednego robota mogą być niewykonalne [26]. Do określania robotów antropomorficznych i zdolnych do interakcji z ludźmi używano sformułowania – robot towarzyski. Jednakże, nawet zindywidualizowane społeczeństwo, w tym ludzie, potrafią tworzyć grupy, sojusze, relacje i sieci społeczne. Z czasem termin robot społeczny stał się bliższym konstrukjom typu robot towarzyski. Obecnie określenie robot towarzyski oznacza podgrupę robotów społecznych.

Definicja 1 (Breazeal)[34] *Roboty społeczne to konstrukcje społecznie inteligentne, zdolne do nawiązywania interakcji i współdziałania z ludźmi, a także komunikowania się w naturalny dla człowieka sposób. Powinny rozumieć siebie jak i ludzi w kategoriach społecznych. Być postrzegane przez ludzi w taki sam sposób, w jaki one mogą postrzegać człowieka. Muszą być w stanie uczyć się, nabierać doświadczenie, a także dzielić się wiedzą z innymi. Celem ich działania jest zaprzyjaźnienie się z ludźmi.*

Niewątpliwie, twórczynią robotyki społecznej jest Cynthia Breazeal. Pionierka wskazała główne cechy, jakimi powinien charakteryzować się robot społeczny, wytyczyła kierunki rozwoju i jest współautorką kilku robotycznych konstrukcji. W książce [34] Breazeal podaje powyższą definicję, w której silnie podkreśla społeczne aspekty i sens konstruowania robotów społecznych. W artykule [30] autorka dokonuje ich podziału na cztery klasy:

1. **Roboty społecznie sugestywne** – są to konstrukcje, w których wykorzystuje się ludzką skłonność do antropomorfizowania technologii i przedmiotów, mogą to być, np. zabawki wyposażone w antropomorficzne elementy, do ich działania nie stosuje się żadnych modeli poznania społecznego.
2. **Roboty społecznie komunikatywne** (społeczne interfejsy) – są to roboty, których działanie opiera się na ich wizerunku, naturalnych dla człowieka kanałach komunikacji werbalnej i niewerbalnej, można je rozumieć jako społeczny interfejs w postaci robota przewodnika po muzeum, w przypadku tej klasy stosuje się proste modele poznania społecznego.
3. **Roboty społecznie wrażliwe** (otwarte) – są to roboty, które posiadają nie tylko umiejętności naturalnego komunikowania się, ale także czerpią korzyści z interakcji. Niezwykle popularne w tej klasie jest stosowanie mechanizmów uczenia się (np. poprzez naśladowanie). Interakcja z robotami tej klasy jest bogatsza i wymaga od robota zwiększonych możliwości percepcji. Ich działanie wykorzystuje średnio rozbudowane modele poznania społecznego.
4. **Roboty towarzyskie** – są to roboty, których działanie nie jest ukierunkowane jedynie na człowieka, ale także na samych siebie. Posiadają motywację oraz cele, są zdolne do budowania modelu działania człowieka. Wymagają rozbudowanych modeli poznania społecznego.

Definicja 2 (Dautenhahn&Billard)[53] *Roboty społeczne to urzeczywistnieni agenci, będący częścią grupy heterogenicznej: społeczności robotów lub/i ludzi. Posiadają*

zdolność do wzajemnego rozpoznawania się i angażowania w interakcje społeczne, nabywają doświadczenie na podstawie własnych interpretacji postrzeganego świata, komunikują się i uczą od siebie.

W pracy [80] powołano się na powyższą definicję wprowadzając pojęcie robota interaktywnego, które nie stanowi nowej grupy robotów społecznych. Autorzy jedynie precyzują, jakie umiejętności powinien posiadać robot społeczny:

- wyrażanie i rozpoznawanie emocji,
- prowadzenie dialogu,
- uczenie się i rozpoznawanie innych agentów,
- nawiązywanie i podtrzymywanie relacji społecznych,
- używanie naturalnych sygnałów (spojrzenie, gesty,...),
- posiadanie charakteru i osobowości,
- posiadanie zdolności uczenia się i rozwijania społecznych kompetencji.

Przedstawiony przez Breazeal podział robotów społecznych, autorzy powyższych prac rozbudowali o kolejne trzy klasy:

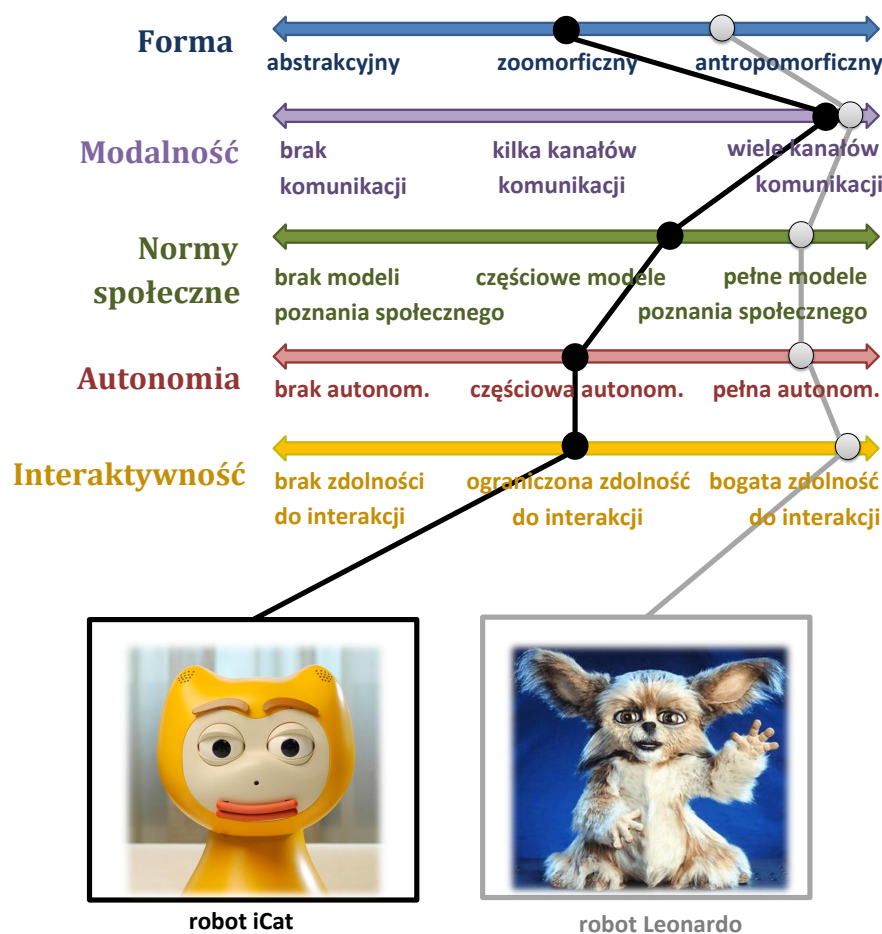
5. **Roboty społecznie osadzone** – roboty działające w środowisku społecznym, zdolne do postrzegania i odróżniania agentów od innych obiektów oraz interakcji z nimi.
6. **Roboty społecznie wbudowane** – roboty działające w środowisku społecznym, zdolne do interakcji z innymi agentami oraz ludźmi. Ich działanie jest sprzężone poprzez środowisko, w którym są osadzone i są w niewielkim stopniu świadome ludzkich struktur interakcyjnych.
7. **Roboty społecznie inteligentne** – roboty, których działanie ma cechy ludzkiej inteligencji oraz bazuje na rozbudowanych modelach ludzkiego poznania i społecznych kompetencjach.

***Definicja 3 (Bartneck&Forlizzi)**[19] Roboty społeczne to autonomiczne lub pół-autonomiczne jednostki zdolne do interakcji i komunikacji z ludźmi, zachowujące przy tym wszelkie normy społeczne zgodne z oczekiwaniami człowieka, z którym robot podejmuje interakcję.*

Przytoczona definicja jako jedyna wskazuje wprost na potrzebę autonomiczności w tego typu konstrukcjach. Pełna lub częściowa autonomia jest konieczna, gdyż zdalnie sterowany robot nie podejmuje samodzielnie decyzji, jest jedynie „przedłużeniem rąk człowieka”. Autorzy twierdzą, że agent musi posiadać fizyczne urzeczywistnienie, zatem wirtualne postaci (np. animacje komputerowe) nie podlegają tej definicji. Wyjątkiem jest robot, wyposażony w ekran, na którym wyświetlana jest graficzna postać głowy. W [19] zaproponowano niezwykle interesujący klasyfikator robotów społecznych. Bartneck i Forlizzi dokonują oceny robota w pięciu kategoriach (przykład na rysunku 1.8). Są to kolejno:

1. **Forma** – wzornictwo, materiał oraz elementy budowy mówią o tym, jakie zdolności behawioralne posiada robot (abstrakcyjny-zoomorficzny-antropomorficzny).

2. **Modalność** – definiuje liczbę kanałów komunikacji, są to, np. wzrokowy, głosowy, dotykowy, węchowy, a nawet smakowy (brak-jedno-dwu-wielokanałowa komunikacja).
3. **Normy społeczne** – w psychologii zauważono, że grupa w której przebywa człowiek ma znaczny wpływ na jego zachowanie. Uwarunkowanie społeczne, w tym interakcja międzyludzka, mogą być przeniesione na poziom interakcji robot-człowiek. W tym celu niezbędne są modele poznania społecznego (brak-częściowe-pełne modele).
4. **Autonomia** – definiuje, czy robot jest zdolny do samodzielnego podejmowania decyzji, bez ingerencji człowieka (brak-częściowa-pełna autonomia)
5. **Interaktywność** – wykazywanie zdolności wzajemnego oddziaływania robota i człowieka na podstawie teorii atrybucji (brak-częściowa-pełna zdolność do interakcji)



Rysunek 1.8: Klasyfikacja robotów iCat i Leonardo, na podstawie pracy [19]

Naukowcy twierdzą, że nie jest możliwe stworzenie jednego robota, który będzie się podobał każdemu użytkownikowi. Klienci kupują produkty różnych marek zwracając uwagę na ich wzornictwo i fizyczne parametry. Na ich wybór mają również wpływ upodobania, gusta, przyzwyczajenia, klienci kierują się także rozsądkiem i ekonomią. Poszukują samochodów lub telefonów komórkowych o odpowiednich parametrach, wyposażonych w wybrany zestaw funkcji. Zupełnie innymi cechami kierujemy się przy wyborze zwierząt

domowych, np. psa czy kota. Interesuje nas rasa, którą cechuje charakter, temperament, zdolność do uczenia się i bycia posłusznym. Powyższe motywy mogą znaleźć zastosowanie w odniesieniu do robotów społecznych. Przypuszczalnie, przyszli nabywcy tych urządzeń przy wyborze będą się kierować umiejętnościami robota oraz możliwością przystosowania go do pracy w miejscu zamieszkania. Ponadto, będą wymagali, aby robot podobnie jak zwierzę domowe, był zdolny do uczenia się zachowań właściciela. W wyniku wzajemnego oddziaływania robot powinien przejść proces podobny do „wychowania” lub „wyszkolenia”, a z czasem poznać preferencje właściciela i odpowiednio na niego reagować. Konstrukcje spełniające powyższe założenia określa się mianem *robota osobistego* [51].

W literaturze można również spotkać określenie *robot towarzyszący* [56]. Jest to rodzaj robota asystującego w środowisku domowym. Stanowi połączenie cech robota społecznego i robota osobistego. Następującą definicję można również zaadoptować do robotów towarzyszących w podróży.

Robot towarzyszący jest to robot, zdolny do działania w środowisku domowym (w podróży). Potrafi wykonywać w nim różne zadania wspomagające domowników (podróżników) w różnych okolicznościach. Posiada społeczne umiejętności i jest zdolny do interakcji z ludźmi w akceptowalny dla właściciela sposób [52].

Rozwój robotyki w kierunku wskazanym przez wymienione w niniejszym rozdziale definicje wymaga wykorzystania szerokiej i interdyscyplinarnej wiedzy z obszaru: psychologii, socjologii, etologii, sztucznej inteligencji, robotyki, informatyki, mechatroniki, teorii gier, ekonomii, ergonomii, sztuki i wielu innych.

1.1.6 Interakcja człowiek-robot

Human-Robot Interaction (HRI) to stosunkowo młoda dziedzina nauki, której celem jest zdobywanie wiedzy na temat społecznych zachowań, zachodzących w trakcie kontaktu człowieka z robotem [52, 260]. U jej podstaw leży głęboko zakorzeniona nauka o interakcji międzyludzkiej. Wyróżnia się trzy trendy rozwoju dziedziny HRI:

- **Zorientowany na robota.** Zakłada się, że niezbędny jest przede wszystkim rozwój konstrukcji robotycznej jako samodzielnej jednostki realizującej swoje cele w oparciu o „własną sztuczną motywację”. Robot powinien posiadać umiejętność przetrwania i być zdolnym do zaspokajania ludzkich potrzeb interakcji. Głównymi tematami badań jest rozwój architektury sterowania, układów sensorycznych i motorycznych, a także implementacja sztucznych emocji.
- **Zorientowany na człowieka.** Podstawowym zadaniem jest rozwój robota w celu poprawy jego postrzegania przez człowieka. Pracuje się zarówno nad wyglądem robota, jak i jego zachowaniem. Wykorzystuje się fakt, że ludzie mają skłonność do antropomorfizowania maszyn i przywiązywania się do nich.
- **Zorientowany na umiejętności poznawcze robota.** motywem przewodnim jest tematyka sztucznej inteligencji, uczenia się, budowania inteligentnych maszyn i systemów decyzyjnych zdolnych do samodzielnego rozwiązywania problemów.

1.2 Współczesny stan wiedzy

1.2.1 Przegląd istniejących konstrukcji

W ciągu ostatniego dziesięciolecia na świecie powstało kilkadziesiąt konstrukcji robotów społecznych, głównie opracowanych z myślą o badaniach w dziedzinie HRI. Znakomita ich większość to prototypy, zaprojektowane i skonstruowane na uniwersytetach oraz w instytucjach badawczych w Japonii, Korei i Stanach Zjednoczonych. Niektóre z projektów zostały zarzucone i nie są już rozwijane, nieliczne są dostępne w postaci produktów komercyjnych. Wybrane z nich warto wymienić z zastosowaniem następującego podziału. Zaawansowane humanoidy: ASIMO [10], Albert Hubo [111], HRP-4 [110], MAHRU/AHRA [164], REEM-C [211], WABIAN-2R i KOBIAN [263], Humanoid [246], SURENA II [236], Kenta/Kotaro /Kojiro [123]. Humanoidalne roboty kołowe: TWENDY-ONE [250], ARMAR-III [8], Assistant Robot [123], REEM-H2 [211], TUBO [248], TIRO [244], FURO [86], Wakamaru [259], EMIEW [74], Care-O-bot 3 [40], TPR-Robina [246], M1 Mobile Manipulator [174], NEXI [203]. Humanoidalne korpusy - roboty stacjonarne pozbawione możliwości przemieszczania się: WE-4RII [263], SIMON [41], Elu2-RT2 [73], Robonaut [218], iCub [115], RoboThespian RT3 [219], DOMO [64], CRONOS [47]. Proste, interaktywne konstrukcje często wykorzystywane w badaniach HRI: Keepon [138, 148], NAO [182], iCat [114], PARO [196], AIBO [3], Pleo [201].

1.2.2 Obszary prowadzonych badań HRI

Analiza literatury wykazała, że roboty społeczne są testowane w interakcjach z ludźmi. Jednym z ważnych kierunków badań HRI są eksperymenty, w których konstrukcje takie jak Keepon, NAO, Kaspar [131], IROMEC [116], Zeno [269] nawiązywały i podtrzymywały interakcję z dziećmi. Liczne publikacje [216, 113, 77, 258, 215, 149, 228, 245], opisują funkcjonowanie tego typu urządzeń w kontaktach z dziećmi i młodzieżą z autyzmem lub zaburzeniami ze spektrum autyzmu oraz z dziećmi cierpiącymi na zespół nadpobudliwości psychoruchowej z deficytem uwagi (ADHD) [83]. Celem tych eksperymentów było zweryfikowanie możliwości zastosowania robota, jako trenera funkcji poznawczych oraz kompetencji społecznych i komunikacyjnych. Większość badań w tym obszarze miała charakter obserwacyjny i jakościowy (próby od kilku do kilkudziesięciu dzieci). Ich rezultaty dowodzą, iż ograniczenia techniczne w działaniu samych konstrukcji robotycznych, na dzień dzisiejszy uniemożliwiają zastosowanie ich na szerszą skalę, np. w środowisku domowym, w ośrodkach terapeutycznych, domach opieki czy szkołach. Jednocześnie autorzy publikacji podkreślają, że roboty społeczne mają potencjał do tego, by w przyszłości stać się użytecznymi narzędziami terapeutycznymi. Roboty społeczne były również testowane w charakterze wsparcia terapeutycznego dla pacjentów przebywających w szpitalach pediatrycznych. Wykazano, że zoomorficzne konstrukcje mogą przyczynić się do redukcji lęku i subiektywnie odczuwanego bólu u dzieci w wieku 6-16 lat. Warto przy tym podkreślić, że efekt ten był znacznie silniejszy, jeśli w zajęciach z robotem uczestniczyli również rodzice. Rezultat ten daje nadzieję, że w przyszłości roboty społeczne mogą być również zaangażowane do prowadzenia wywiadów diagnostycznych i innych zajęć w instytucjach opiekuńczych i placówkach pomocy społecznej.

Eksperymenty z robotami społecznymi były również ukierunkowane na ocenę ich przydatności w kontekście edukacyjnym [187, 237, 118, 222]. Najczęściej wykorzystywane w tym celu są roboty NAO oraz iCat. Wykazano, między innymi, że sposób zachowania się dzieci w czasie wywiadu prowadzonego przez roboty jest niemal identyczny, jak

zachowanie uczniów, którzy uczestniczą w rozmowach z nauczycielami. Ich obecność działała motywująco dzięki czemu uzyskuje się pozytywne efekty uczenia się. Przeprowadzono również badania [141, 121], w których zwrócono uwagę na fakt, że roboty społeczne mają wysoką skuteczność w motywowaniu dzieci do zdrowego żywienia i wykonywania ćwiczeń. Brak jest jednak przeprowadzonych na szeroką skalę badań weryfikujących użyteczność robota w charakterze nauczyciela lub trenera.

Innym obiecującym obszarem badań jest wykorzystanie robotów towarzyszących w terapii i wsparciu osób starszych oraz doświadczających objawów choroby Alzheimera, demencji i łagodnych zaburzeń funkcji poznawczych [257, 256, 101, 254, 186]. Znaczący udział w tego typu badaniach mają roboty PARO oraz AIBO. Okazuje się, że osoby starsze, w szczególności mające wcześniej kontakt z nowymi technologiami, a także doświadczające wspomnianych wyżej objawów w niewielkim nasileniu, chętnie akceptują obecność robotów zoomorficznych w swoim otoczeniu. Dotychczas nie udało się wykazać, iż roboty mogą przyczynić się do mierzalnej poprawy funkcji poznawczych, np. pamięci. Jednak ich obecność bezsprzecznie wpływa pozytywnie na nastrój badanych.

Większość publikacji naukowych z dziedziny robotyki zaznacza, że humanoidalne maszyny mają znaczny potencjał dalszego rozwoju. Ograniczeniem wszystkich opisywanych badań jest to, że były one często prowadzone na nielicznych próbach oraz obarczone pewnymi błędami metodologicznymi. Przyczyną tego stanu rzeczy jest nierzadko wyraźny rozdział, między naukami społecznymi a technicznymi. Te pierwsze koncentrują się na aspektach metodologicznych i realizacyjnych. Natomiast inżynierowie skupiają się wokół rozwiązań konstrukcyjnych, zagadnień sterowania, systemów lokomocji, wizji i układów sensorycznych. Brak współpracy i wymiany wiedzy jest poważnym ograniczeniem w implementacji robotów społecznych w realnych sytuacjach zadaniowych.

1.2.3 Techniki przeprowadzania eksperymentów HRI

W dziedzinie robotyki społecznej stosuje się wiele powszechnie znanych socjologiczno-psychologicznych technik badawczych. Najczęściej doprowadza się do krótkoterminowych, trwających od kilku do kilkunastu minut, kontaktów robota z człowiekiem. W przypadku eksperymentów typu długoterminowych dochodzi do serii spotkań z robotem, które trwają od 10 do nawet 30 dni. Dane zbiera się w oparciu o podstawowe metody i techniki badawcze:

- **PAPI** (*Paper And Pencil Interview*) – tradycyjna metoda badań ilościowych bazująca na papierowej wersji kwestionariusza, którego wypełnienie odbywa się przed lub po eksperymencie z robotem.
- **IDI** (*Individual in-Depth Interview*) indywidualny wywiad pogłębiony – badanie jakościowe polegające na indywidualnej, wnikliwej rozmowie z uczestnikiem eksperymentu. Nie rzadko, jest źródłem wielu informacji, których nie sposób przewidzieć w prostych formularzach ankietowych. Kluczową kwestią w tej metodzie jest wzbudzenie motywacji osoby uczestniczącej w badaniu do szczerych, otwartych i możliwie jak najpełniejszych odpowiedzi.
- **FGI** (*Focused Group Interview*) zogniskowany wywiad grupowy – jest to dyskusja prowadzona przez moderatora z grupą uczestników eksperymentu, którzy są zwykle dobierani zgodnie z kryteriami wynikającymi z celów badania. Wykorzystywana jest w eksperymentach grupowych, w których do zabawy z robotem angażuje się grupę, np. dzieci bawiące się z robotem w czasie lekcji lub przerwy obiadowej.

- **Analiza nagrań video** – technika dokonywania obserwacji, którą wykonują powołani sędziowie kompetentni (osoby przeszkolone w sposobie definiowania ocenianego zjawiska). W trakcie przeglądania nagrań wideo z przebiegu eksperymentu, ocenia się poziom zaangażowania uczestnika w interakcję z robotem, np. postawa ciała, czas trwania kontaktu wzrokowego, czas po którym badany udziela odpowiedzi, liczbę werbalnych i niewerbalnych kontaktów, wyrażane emocje, odporność na bodźce zakłócające eksperyment, itp.
- **Analiza danych procesowych** – w zależności od celu badań wykorzystuje się dane, które są zbierane w trakcie eksperymentu i przechowywane w pamięci robota. Pochodzące z układów percepcji, mogą być wykorzystane, np. w procesie uczenia się. W takiej sytuacji ich mierzalnym efektem może być poziom uzyskanej wiedzy na temat kolorów, kształtów, preferencji proksemicznych użytkownika, itp.

Systemy sterowania robotami społecznymi realizuje się w oparciu o wielowarstwową architekturę, w której najwyższy poziom reprezentowany jest przez sztuczny umysł. Efektem jego działania jest okazywanie wiarygodnych, ludzkich zachowań. Toczy się dyskusja na temat poziomu autonomii robota w trakcie przeprowadzania eksperymentów HRI. Faktem jest, że wszelkie próby uzyskania autonomicznej pracy robota wprowadzają dodatkową zmienną losową narażając eksperyment na niepowodzenie [162]. Jednakże, pozbawiając robota autonomii zatracamy możliwość dostrzeżenia aspektów dotyczących prac badawczych zorientowanych na rozwój robotów oraz zweryfikowanie ich obecnych możliwości [33]. Działanie robota, które bazuje na jego zdolnościach percepcyjnych, pozwala zaobserwować zjawiska zachodzące w trakcie, np. ich błędnego działania. Zawodne, nie odporne na zmianę barwy głosu, systemy rozpoznawania mowy czy też działający na podstawie kanału wizyjnego, systemy detekcji człowieka wciąż wymagają prowadzenia intensywnych prac nad ich doskonaleniem. Znacznie lepszym pomysłem okazuje się projektowanie takich scenariuszy eksperymentów, w których to robot jest ich moderatorem. Podtrzymanie autonomii nie rzadko wymusza ograniczenie się do sytuacji, w których człowiek odpowiada jedynie „tak” lub „nie”, czy też wykonuje, oczekiwany przez robota gest. Takie podejście wciąż gwarantuje pojemne źródło informacji na temat zjawisk zachodzących w trakcie interakcji człowieka z robotem, a co najważniejsze pozwala na dalsze pokonywanie barier technologicznych.

W szczególnych przypadkach dopuszcza się technikę zdalnego sterowania. Często, z powodu ograniczeń technologicznych, metoda *Wizard of Oz* [169, 139] jest jedyną możliwością przeprowadzenia eksperymentów. O przykładach i efektach jej zastosowania można przeczytać w pracach [4, 109, 233]. Należy jednak pamiętać, że z punktu widzenia definicji, przedmiotem badań tą metodą może być, np. wygląd robota, a nie jego zachowanie.

Rozdział 2

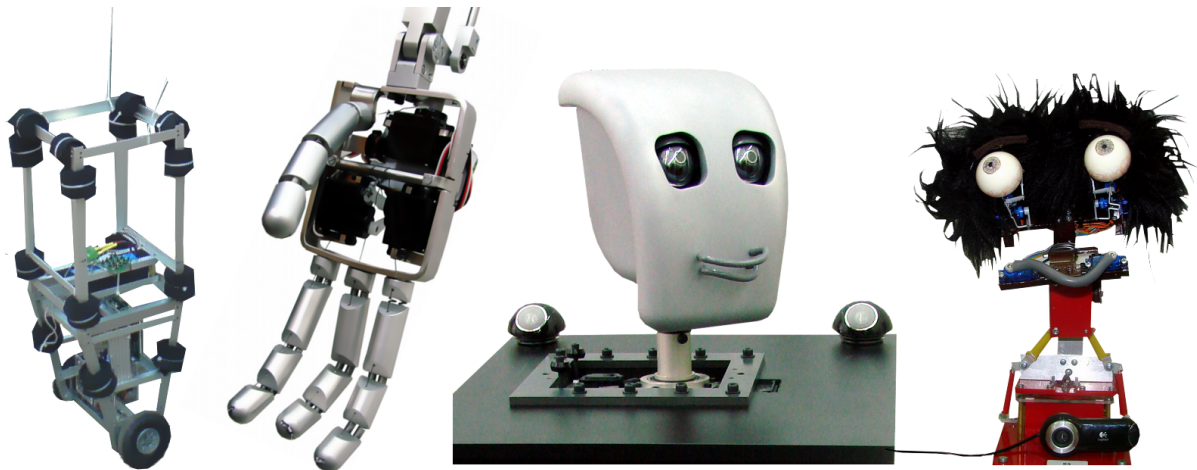
Cel pracy

Rynek robotów przemysłowych wymusił wypracowanie pewnych standardów dotyczących ich budowy i wyposażenia. Wygląd, interfejsy oraz metody programowania robotów przemysłowych są do siebie bardzo zbliżone, choć z przyczyn oczywistych nie kompatybilne. Powodem tego zjawiska jest stawianie jednakowych wymagań, np. manipulatorom przemysłowym. Ponadto, przywiązuje się dużą uwagę do jakości wykonywania zadań przez roboty. Dąży się do wytworzenia precyzyjnych układów manipulacyjnych, złożonych systemów nawigacyjnych oraz inteligentnych modułów kontroli i testu. Działania te zmierzają w kierunku przyspieszenia procesów produkcyjnych oraz osiągnięcia lepszej jakości wytwarzanych produktów. Niezawodność tego typu maszyn jest jednym z najważniejszych kryteriów oceny. Producenci robotów przemysłowych dostarczają dedykowane dla konkretnej marki systemy sterowania. Ich funkcjonalność i języki programowania są niekiedy bardzo podobne, lecz w żaden sposób nie przenośne. Zamknięta architektura systemu sterowania i ograniczony dostęp do dokumentacji uniemożliwiają stosowanie własnych algorytmów sterowania. Istnieją jednak otwarte projekty programistyczne, które wykorzystują warstwę sprzętową robota przemysłowego i pozwalają na implementowanie własnych sterowników. Na szczególną uwagę zasługują projekty OROCOS [36] i MROC++ [270, 271].

W przypadku innych działów robotyki standardy ciągle są kształtowane i podlegają ewolucji. Zjawisko to jest szczególnie widoczne w przypadku robotyki społecznej, która jest relatywnie nowym działem robotyki. Zważywszy na obszar potencjalnych zastosowań robotów społecznych i zakres prowadzonych badań, które dotyczą głównie interakcji pomiędzy człowiekiem a robotem, należało zdefiniować cele prac badawczo-rozwojowych, których efektem będzie nie tylko sprawnie działające stanowisko laboratoryjne, lecz także sposób jego oddziaływania na człowieka. Aktualnie roboty społeczne nie są dostępne komercyjnie, stanowią najczęściej prototypowe konstrukcje laboratoryjne. Ich systemy sterowania to zwykle dedykowane i często nieudokumentowane oprogramowanie. Niemniej jednak, w ostatnich latach powstało kilka otwartych projektów programistycznych z myślą o wykorzystaniu w robotyce, które z powodzeniem mogą być znaleźć istotne zastosowanie w robotyce społecznej. Do najpopularniejszych należą: dynamicznie rozwijający się ROS (*Robot Operating System*) [206] i YARP (*Yet Another Robot Platform*) [175], współpracujący z nimi Gostai Urbi [252], a także popularne systemy nawigacji: Player [44] i Karto [129]. Nie istnieje żadne gotowe oprogramowanie, które traktuje problem sterowania robotem społecznym w sposób kompleksowy.

Przedstawione w niniejszej rozprawie prace badawcze były prowadzone w ramach europejskiego projektu LIREC (*Living with Robots and Interactive Companions*) [156], który sfinansowano z 7 Programu Ramowego. Jednym z istotnych celów projektu było opracowanie technologii budowy robotów towarzyszących człowiekowi, zdolnych do operowania w

środowisku ludzkim i nawiązywania z ludźmi długoterminowych relacji. Przeprowadzone w pierwszej fazie projektu rozpoznanie w dziedzinie dostępnych rozwiązań [55] i szereg eksperymentów w małej skali z wykorzystaniem pre-prototypów (rysunek 2.1), takich jak platforma balansująca Cosmos [133], ramię oraz dłoń [54] zainstalowane na platformie PeopleBot [2], a także głowy Romek i Samuel [37]), w znacznym stopniu ukierunkowały dalsze prace badawcze. W trakcie prac nad projektem podjęto decyzję o budowie ro-



Rysunek 2.1: Pre-prototypy robota FLASH: od lewej Cosmos, dłoń, Samuel i Romek

bota, który miałby pozwolić na demonstrację opracowanych technologii w rzeczywistych warunkach operacyjnych. W pierwszej fazie trwania projektu LIREC opracowano szereg wytycznych, na podstawie których robot miał umożliwić realizację podstawowych scenariuszy badawczych. Powinien być, przede wszystkim, zdolny do nawiązania relacji społecznej z człowiekiem, przez dostosowanie swojego zachowania do jego preferencji [145] oraz zachowywać się w sposób akceptowalny przez człowieka [51, 25]. Precyzja, powtarzalność i niezawodność nie odgrywały tak wielkiej roli, jak to ma miejsce w przypadku zastosowań przemysłowych. Opracowując koncepcję robota należało uwzględnić możliwe wszystkie przedstawione w rozdziale pierwszym, aspekty związane z projektowaniem, budowaniem i sterowaniem robotem społecznym. Podczas jego budowy brano pod uwagę nie tylko aspekty technologiczne, ale również psychologiczne i społeczne. Zdecydowano o tym, aby wygląd robota posiadał elementy antropomorficzne, dzięki którym robot będzie łatwiej podtrzymywał interakcję z ludźmi. Ponadto, ekspresywna twarz i ręce zdolne do gestykulacji miały zwiększyć stopień akceptacji robota i umożliwić nawiązywanie długoterminowej relacji robot-człowiek. Zaobserwowano także, że platforma balansująca posiada lepsze własności jezdne niż statyczne podwozie, a jej drobne ruchy, wynikające z zaburzeń stanu równowagi, mają niezwykle pozytywny wpływ na sposób postrzegania robota. Ważne również było, aby cała konstrukcja miała charakter modułowy i umożliwiła dalszą rozbudowę poszczególnych składników robota oraz ich szybki serwis.

Ostatecznie, doprowadziło to do powstania stanowiska badawczego w postaci robota FLASH (*Flexible Lirec Autonomous Social Helper*) [134]. Jego konstrukcja mechaniczna składa się z dwukołowej, balansującej platformy mobilnej, ekspresyjnej głowy złożonej z trzech talerzy i z dwóch rąk zakończonych zręcznymi dłońmi o strukturze kinematycznej przypominającej ramiona i ręce człowieka. Podczas projektowania rozmiarów i wyglądu robota, brano pod uwagę jego walory użytkowe. Przyjęte wymiary, wygląd oraz jego możliwości są kompromisem między oczekiwaniami względem konstrukcji i dostępną technologią a wynikami aktualnie prowadzonych badań, dotyczących wpływu wyglądu robota

na stopień jego akceptacji przez człowieka. System sterowania robota zbudowano w oparciu o paradygmat architektury trójwarstwowej [35, 87, 12]. Wytworzone oprogramowanie składa się z modułów umożliwiających dostęp do warstwy sprzętowej i odpowiadających za przetwarzanie danych sensorycznych. Operowanie robota według zaplanowanych scenariuszy badawczych wymagało opracowania zestawu kompetencji pozwalających, m. in. na nawigowanie w naturalnym środowisku człowieka, wykonywanie prostych czynności usługowych, gestykulację, komunikację werbalną i niewerbalną, a zwłaszcza ekspresję emocji. W celu efektywnego wykorzystania zasobów sprzętowych należało zastosować mechanizmy programowe pozwalające na synchroniczną i równoległą pracę poszczególnych składników robota. Naturalnie, projektując każdy scenariusz zachowań robota, należało zadbać o afektywną część sztucznego umysłu, gdyż aspekt ten jest ważnym składnikiem procesu decyzyjnego robota społecznego. To na podstawie zdarzeń zewnętrznych, wytyczonych celów, aktualnego stanu emocjonalnego i stanu świata są generowane akcje realizowane następnie na poziomie kompetencji. Weryfikacji opracowanych technologii dokonano w eksperymentach HRI.

Robot FLASH został skonstruowany w Zakładzie Podstaw Cybernetyki i Robotyki pod kierunkiem prof. dr hab. inż. Krzysztofa Tchonia. Prace nad robotem odbywały się w zespole 9-osobowym przy współpracy z trzema zewnętrznymi biurami projektowymi. Autor pracy był jednym z głównych twórców pre-prototypów platformy balansującej Cosmos oraz głów Romek i Samuel oraz ich systemów sterowania. Wkład autora w powstanie robota FLASH polegał na współudziale w opracowaniu koncepcji konstrukcji mechanicznej na podstawie której zlecono wykonanie projektów mechanicznych oraz graficznych. Na bazie tych opracowań wykonano większość detali przy użyciu maszyn CNC oraz z wykorzystaniem technik szybkiego prototypownia (druku 3D). Autor wykonał montaż poszczególnych mechanizmów napędowych oraz zintegrował główne komponenty robota (platforma, korpus, głowa). Naturalnie, pozostali członkowie zespołu mieli również znaczący udział we wszystkich wyżej wymienionych zadaniach. W ramach niniejszej rozprawy autor opracował i wykonał system sterowania robota FLASH. Szczegółowe zadania rozwiązane w rozprawie obejmują:

- wyprowadzenie modelu kinematyki i dynamiki platformy robota balansującego,
- zaproponowanie liniowych algorytmów sterowania ruchu i balansowania platformy typu LQR, PP i PD, wykorzystujących przybliżenie liniowe modelu wzdłuż trajektorii wyznaczonej przez podsystem nawigacji,
- zaproponowanie algorytmów sterowania platformy wykorzystujących własność różniczkowej płaskości modelu platformy,
- zaproponowanie jakobianowych algorytmów planowania ruchu platformy z ograniczeniami i podziałem na podzadania metodą endogenicznej przestrzeni konfiguracyjnej,
- zastosowanie algorytmu sterowania optymalnego i sterowania predykcyjnego do śledzenia trajektorii platformy,
- przeprowadzenie badań symulacyjnych i ocena zaproponowanych algorytmów,
- implementację w robocie FLASH algorytmu LQR i algorytmu typu PD do sterowania ruchu i balansowania platformy,

- implementację filtra Kalmana do fuzji pomiarów z żyroskopu i akcelerometrów w celu wyznaczenia kąta odchylenia korpusu robota,
- opracowanie i wykonanie niskopoziomowego sterownika platformy wraz z układami peryferyjnymi i implementację systemu komunikacji zgodnej systemami Adept AR-COS,
- opracowanie i wykonanie niskopoziomowego sterownika głowy EMYS robota FLASH i implementację protokołu komunikacji zgodnego z protokołem Dynamixel,
- dobór układów napędowych platformy i głowy oraz opracowanie i wykonanie systemu zasilania robota,
- integrację niskopoziomowych sterowników platformy, ramion, dłoni i głowy, i zapewnienie komunikacji między nimi,
- opracowanie koncepcji struktury systemu sterowania w oparciu o paradygmat architektury trójwarstwowej,
- zastosowanie platformy programistycznej Gostai Urbi o w systemie sterowania robota we wszystkich trzech warstwach jego architektury,
- implementację zestawu dynamicznych bibliotek (UObject) do obsługi warstwy sprzętowej robota: sensorów, napędów, kamer, mikrofonu, etc.,
- implementację zestawu kompetencji robota niezbędnych do operowania w trakcie interakcji człowiek-robot, w tym śledzenia człowieka, gestykulacji, wyrażania emocji, generowania i rozpoznawania mowy oraz nawigacji typu SLAM,
- przygotowanie robota do pracy autonomicznej i implementację scenariuszy eksperymentów HRI,
- przeprowadzenie eksperymentów HRI z udziałem robota FLASH i głowy EMYS pozwalających na weryfikację konstrukcji i systemu sterowania robota.

Praca pokazuje, że współczesny poziom rozwoju teorii, techniki i technologii sterowania pozwala na budowę systemu sterowania robota społecznego. Własności systemu sterowania będącego przedmiotem niniejszej rozprawy można podsumować w następujący sposób:

- Sposób sterowania robota FLASH uwzględnia wyniki aktualnie prowadzonych badań, dotyczących wpływu wyglądu i zachowania robota na stopień jego akceptacji przez człowieka. Robot posiada elementy antropomorficzne, dzięki którym łatwiej podtrzymuje interakcję z ludźmi. Balansująca platforma, ekspresywna twarz i ręce zdolne do gestykulacji zwiększają stopień akceptacji robota.
- System sterowania robota jest kompletny, spójny i wykorzystuje efektywnie zasoby sprzętowe konstrukcji dzięki zastosowaniu platformy programistycznej Gostai Urbi. Platforma ta dostarcza mechanizmów pozwalających na synchroniczną i równoległą pracę komponentów robota (układ jezdny, ramiona, dłonie, głowa) w czasie rzeczywistym.

-
- Zaproponowany zestaw oprogramowania umożliwia realizację ludzkich zachowań robota, podejmowanie interakcji z człowiekiem, zachowywanie się w sposób wiarygodny i akceptowalny przez niego. Robot używa w tym celu gestykulacji, ekspresji emocji, komunikacji werbalnej i niewerbalnej, a dzięki systemom nawigacji może operować w naturalnym środowisku człowieka.
 - System sterowania robota udostępnia zestaw kompetencji, który pozwala na budowanie rozbudowanych scenariuszy eksperymentów HRI mających na celu zgłębianie wiedzy na temat dziedziny robotyki społecznej.

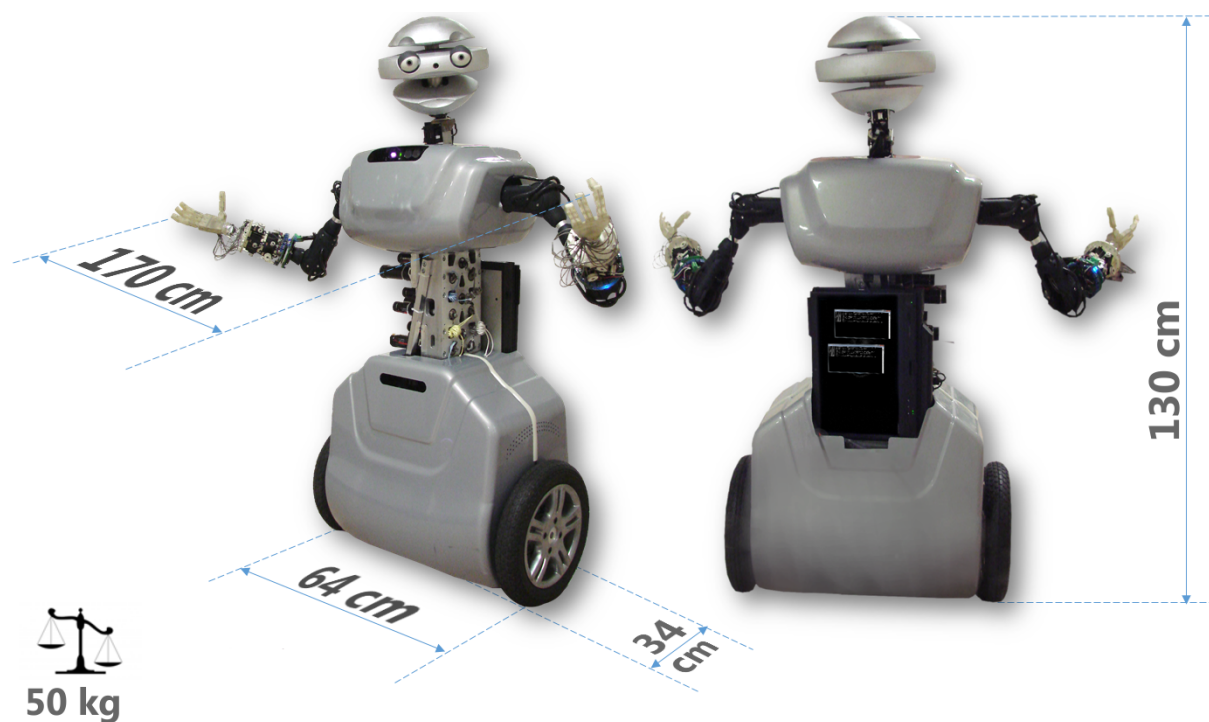
Warto dodać, że Politechnika Wrocławska jest jedynym z trzech ośrodków zajmujących się robotyką społeczną w Polsce. Badania nad robotami społecznymi były prowadzone na Politechnice Łódzkiej, gdzie powstał interaktywny robot Tepson [92, 91] i na Politechnice Gdańskiej, gdzie opracowano xEmotion – obliczeniowy model emocji dedykowany dla inteligentnych systemów decyzyjnych [147].

Rozdział 3

Robot FLASH implementacja sprzętowa

Projekt konstrukcji mechanicznej platformy oraz korpusu został opracowany w ramach projektu LIREC przy współpracy z biurem projektowym TF Construction sp. j. Projekt mechaniczny głowy oraz dłoni opracowano wraz z firmą Cad-Mech Sp. z o. o. Projekt graficzny obudów robota oraz kształtu głowy wykonano przy współpracy z Wrocławską Akademią Sztuk Pięknych.

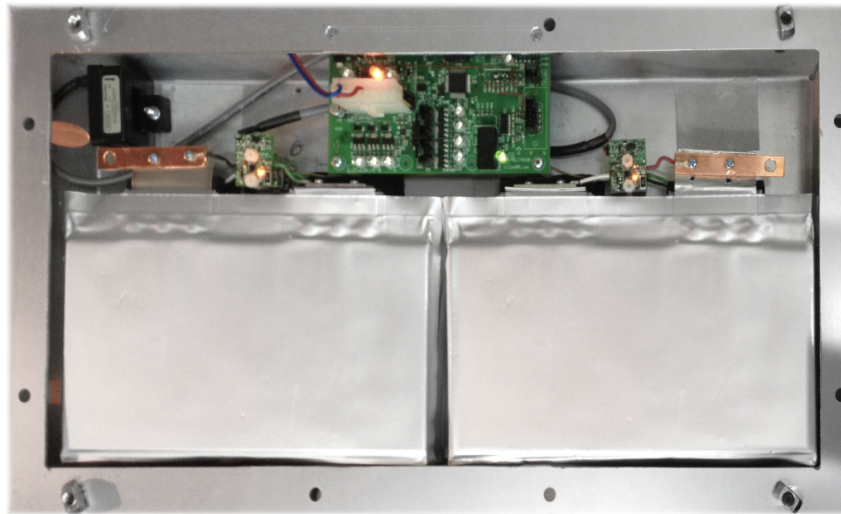
3.1 Ogólne parametry techniczne



Rysunek 3.1: Widok robota FLASH

Robot FLASH porusza się na platformie balansującej, wyposażony jest w ekspresywną głowę EMYS [136], korpus i dwa ramiona zakończone zręcznymi dłońmi. Wysokość robota wynosi 130cm, szerokość ze złożonymi ramionami 64cm, po rozłożeniu wzrasta do

ponad 170cm, długość 35cm, a jego waga to 50kg. Widok robota zaprezentowano na rysunku 3.1. Trzon układu sensorycznego robota stanowią: optyczny czujnik przestrzenny KINECT, skaner laserowy HOKUYO URG-04LX-UG01 oraz kamera kolorowa Quick-Cam Sphere AF firmy Logitech. System zasilania został zbudowany na bazie baterii akumulatorów firmy Kokam, o pojemności 16Ah i napięciu zasilania 42V (rysunek 3.2). Przy takiej konfiguracji robot może pracować bez przerwy przez 2-4h, w zależności od intensywności wykonywanych ruchów ramionami. Strukturę układu zasilania komponentów robota zaprezentowano na rysunku 3.3. Obecnie system sterowania robota zarządzany jest przez komputer pokładowy klasy laptop w konfiguracji Core i7-3250M 2.9 GHz, 16GB RAM, 4MB Cache, 180GB SSD, który może działać pod kontrolą systemu operacyjnego Linux Ubuntu lub Windows Embedded. Strukturę robota FLASH przedstawiono na rysunku 3.3.

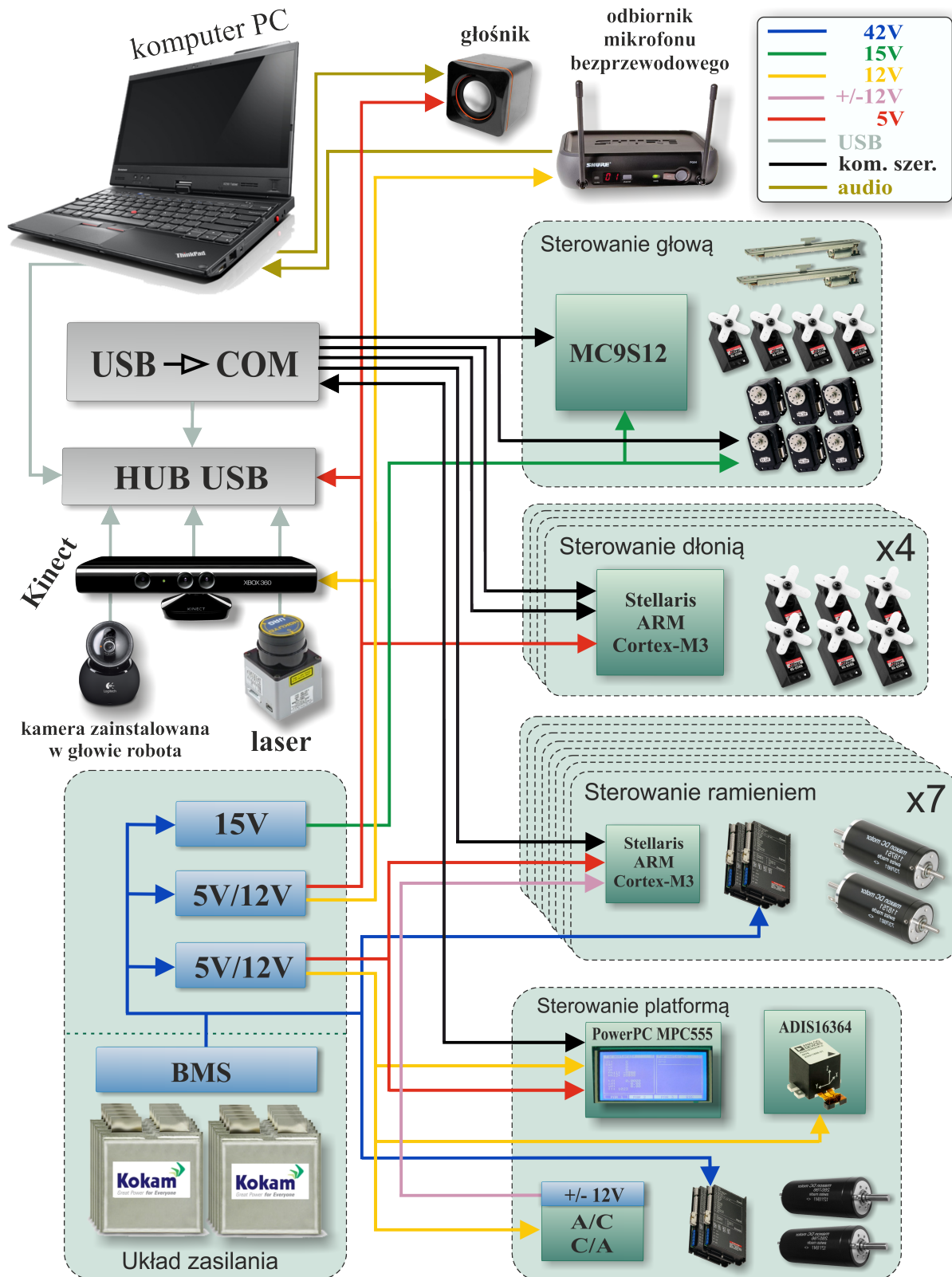


Rysunek 3.2: Bateria akumulatorów zainstalowana w dolnej części platformy jezdnej

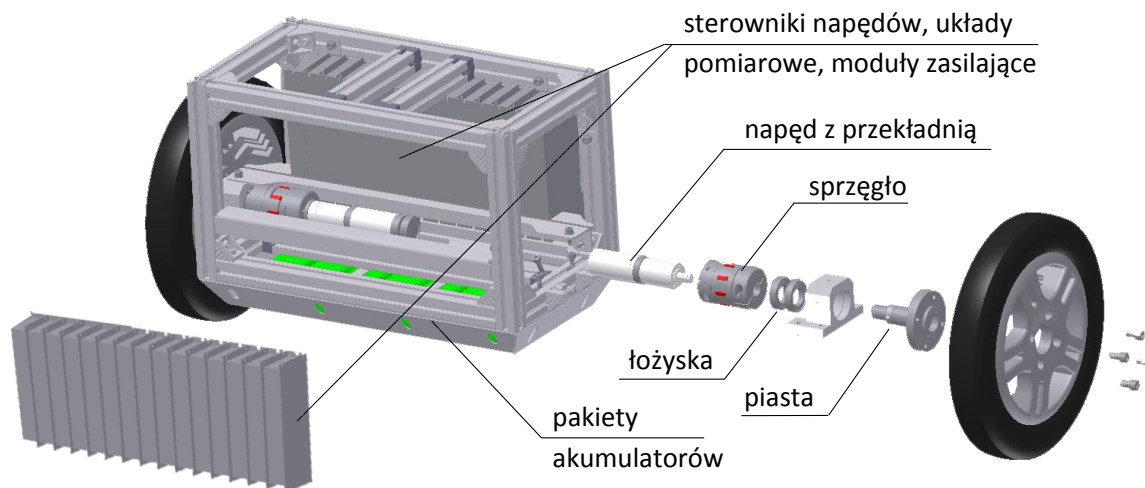
3.2 Platforma jezdna

Platforma ma charakter modułowy, a jej konstrukcję przedstawiono na rysunku 3.4. Szczegółowe informacje na temat platformy jezdnej można znaleźć w raporcie [54]. Większość elementów została zaprojektowana i wykonana od podstaw, gdyż zastosowanie gotowych rozwiązań nie było możliwe lub bardzo kosztowne. Do budowy konstrukcji nośnej zastosowano profile aluminiowe firmy BOSCH-REXROTH, które umożliwiają dalszą rozbudowę i łatwy montaż nieprzewidzianych komponentów. Całość porusza się na dwóch pompowanych kołach o średnicy 32cm firmy ROLKO. Nad napędami w dwóch rzędach zainstalowano: sterowniki platformy, sterowniki ramion, układy pomiarowe oraz moduły zasilaczy. Dolna część stanowi misa, w której umieszczono pakiety akumulatorów. Koła napędzane są dwoma szczotkowymi silnikami prądu stałego RE-35 o mocy 90W firmy MAXON, poprzez przekładnię o przełożeniu 66:1. W efekcie, pojedynczy zespół napędowy pozwala na uzyskanie maksymalnego momentu 50Nm. Na silnikach fabrycznie zainstalowano kodery kwadraturowe o rozdzielczości 1024 impulsy/kanal. Napęd na koła został przeniesiony poprzez wysokiej jakości sprzęgła firmy GERWAH.

Sterowanie ruchem platformy odbywa się poprzez system nawigacji, który do planowania ścieżki wykorzystuje dane o stanie platformy, pomiary ze skanera laserowego



Rysunek 3.3: Układ zasilania komponentów robota FLASH



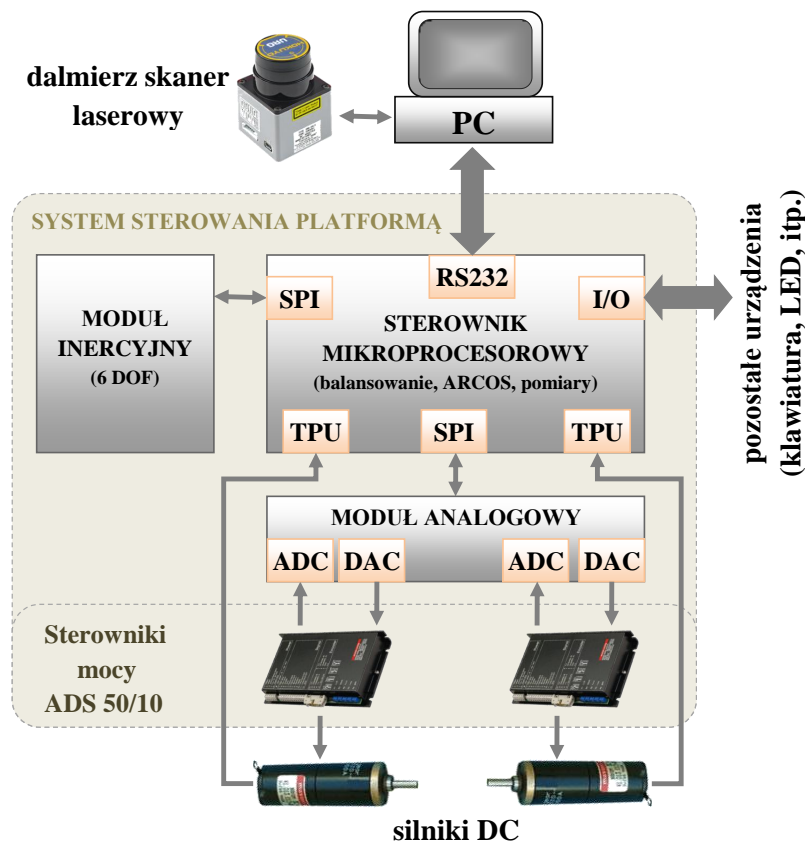
Rysunek 3.4: Konstrukcja mechaniczna platformy

(rysunek 3.6(f)) oraz informacje z mapy obszaru, po którym się porusza. System ten pracuje na komputerze pokładowym. Sterownik niskopoziomowy (rysunek 3.6(a)) odpowiada za utrzymywanie robota w równowadze, realizowanie zadanych prędkości, monitorowanie układów zasilania oraz stanów napędów. Komunikacja z warstwą wyższą odbywa się poprzez przewodowy interfejs RS232 lub bezprzewodowy Bluetooth. Głównym elementem sterownika jest minimoduł phyCORE-MPC555 firmy PHYTEC (rysunek 3.6(b)) wyposażony w mikrokontroler PowerPC MPC555 firmy Freescale [180]. Układ ten posiada 64-bitową jednostkę FPU (*Float Pointing Unit*) oraz dwie jednostki TPU (*Time Processor Unit*), które zawierają własny procesor i pozwalają odciążyć CPU od częstych, powtarzalnych operacji związanych z funkcjami czasowo-licznikowymi. Ponadto sterownik wyposażono w wyświetlacz dotykowy o rozdzielczości 240x128 pikseli, który umożliwia łatwe strojenie algorytmów balansowania oraz monitorowanie pracy platformy.

Sygnały generowane w sterowniku, trafiają do bloku przetwarzania sygnałów analogowych, wymaganych przez sterowniki mocy ADS 50/10 firmy MAXON Motor [210] (rysunek 3.6(e)). Pracują one w trybie sterowania prądowego, który pozwala na zadawanie momentu generowanego przez silnik. Wartości te, zadaje się poprzez analogowe napięcie $\pm 10V$. Sygnały te generowane są przez precyzyjne przetworniki cyfrowo-analogowe DAC714 firmy Burr Brown. Sterowniki silników, pozwalają także na monitorowanie aktualnego stanu napędów, np. aktualną wartość prądu płynącego przez cewki silnika, prędkość obrotowa silnika. Wartości, te w postaci analogowego sygnału są mierzone przez przetwornik analogowo-cyfrowy MAX1270 firmy MAXIM. Blok przetwarzania sygnałów analogowych (rysunek 3.6(c)) posiada izolację galwaniczną. Do pomiaru odchylenia od pionu użyto zintegrowanego czujnika inercyjnego ADIS16364 firmy Analog Devices (rysunek 3.6(g)). Czujnik ten, w swej strukturze, zawiera trzy akcelerometry i trzy żyroskopy, po jednym dla każdej osi. Jego zaletą tego jest to, że jest on skompensowany w zakresie temperatur $-20 \div 70^{\circ}C$, skalibrowany w zerze, posiada odpowiedni zestaw filtrów oraz wewnętrzny przetwornik A/C. Strukturę systemu sterowania platformy przedstawiono na rysunku 3.5.

Ponadto, zaprojektowano i wykonano dedykowane moduły przetwornic niskonapięciowych. Każdy moduł dostarcza pięciu napięć, trzy po 5V i dwa po 12V (rysunek 3.6(d)).

Oprogramowanie sterownika składa się z ośmiu głównych bloków. Większość z nich



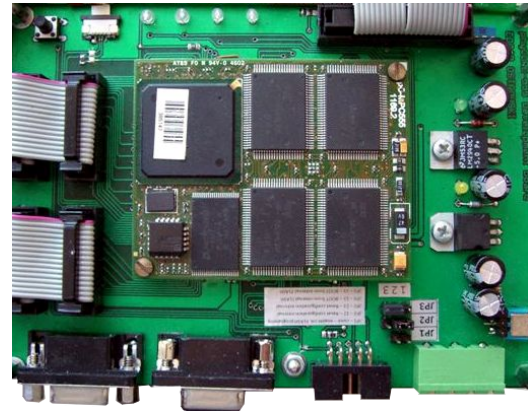
Rysunek 3.5: Schemat blokowy systemu sterowania platformy

ściśle współpracuje z peryferiami mikrokontrolera, a niektóre odpowiadają jedynie za realizację obliczeń. Najbardziej czasochłonne są obliczenia algorytmu realizującego balansowanie platformy oraz filtrację danych sensorycznych. Do wyznaczenia odchylenia od pionu, wykorzystuje się pomiary przyspieszeń w dwóch osiach oraz pomiar prędkości kątowej w jednej osi. Fuzji zebranych informacji dokonuje się w sposób efektywny, przy użyciu standardowego algorytmu filtracji Kalmana. Wynikiem jego pracy jest pomiar odchylenia korpusu od pionu. Informacja ta jest wykorzystywana w trakcie przemieszczania robota i utrzymywania jego równowagi. Zastosowano liniowy algorytm balansowania wykorzystujący przybliżenie liniowe dynamiki układu.

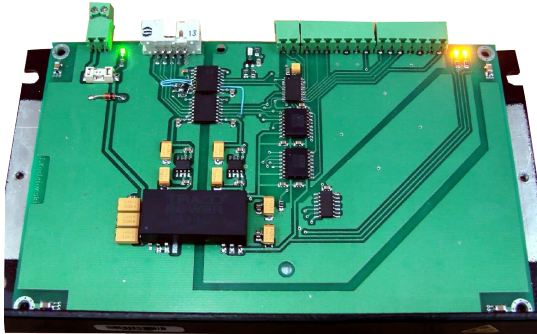
Kolejnym, ważnym zadaniem systemu sterowania platformy jest realizacja komunikacji zgodnej z systemem, ARCOS (*Advanced Robotics Control and Operations*) jaki pracuje na platformach Pioneer 3-DX, -AT, Performance PeopleBot lub PowerBot firmy Adept Mobile Robots [2]. System ten dostarcza informacji o ruchu platformy, stanie baterii i danych sensorycznych, takich jak czujniki obrotów kół, dalmierzy sonarowych oraz podczerwieni, czujników inercyjnych, zderzaków i wiele innych. Pozwala także na konfigurację parametrów robota, np. maksymalną prędkość, maksymalne przyspieszenie, dopuszczalne przemieszczenie, poziomy alarmów, liczbę dostępnych dalmierzy i wiele innych. Dzięki temu, możliwe jest sterowanie nawigacją robota z użyciem oprogramowania Aria [2] lub Player [200]. Schemat blokowy oprogramowania przedstawiono na rysunku 3.7. Na rysunku 3.8 zaprezentowano wyniki pomiarów zebranych w czasie jazdy platformy.



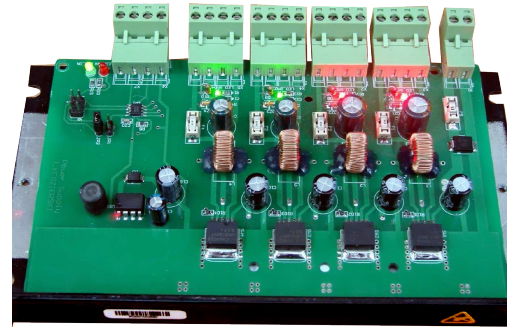
(a) Sterownik platformy jezdnej



(b) Jednostka centralna platformy jezdnej



(c) Układ generowania i przetwarzania sygnałów analogowych



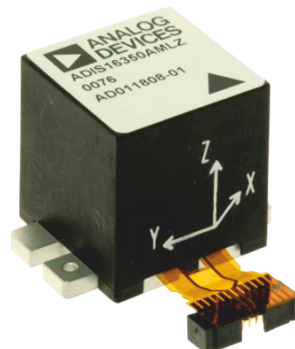
(d) Zespół przetwornic niskonapięciowych



(e) Sterownik mocy napędów platformy i ramion

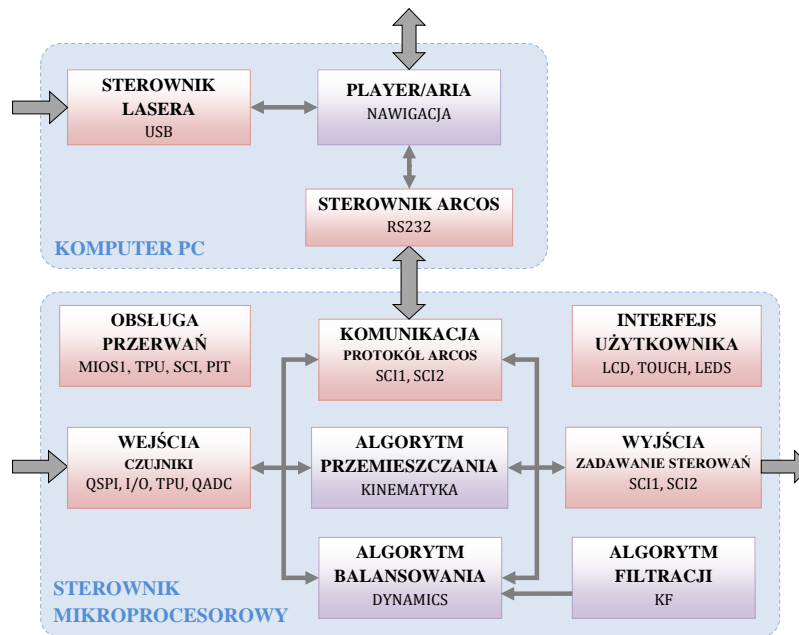


(f) Skaner laserowy



(g) Czujnik inercyjny ADIS16364

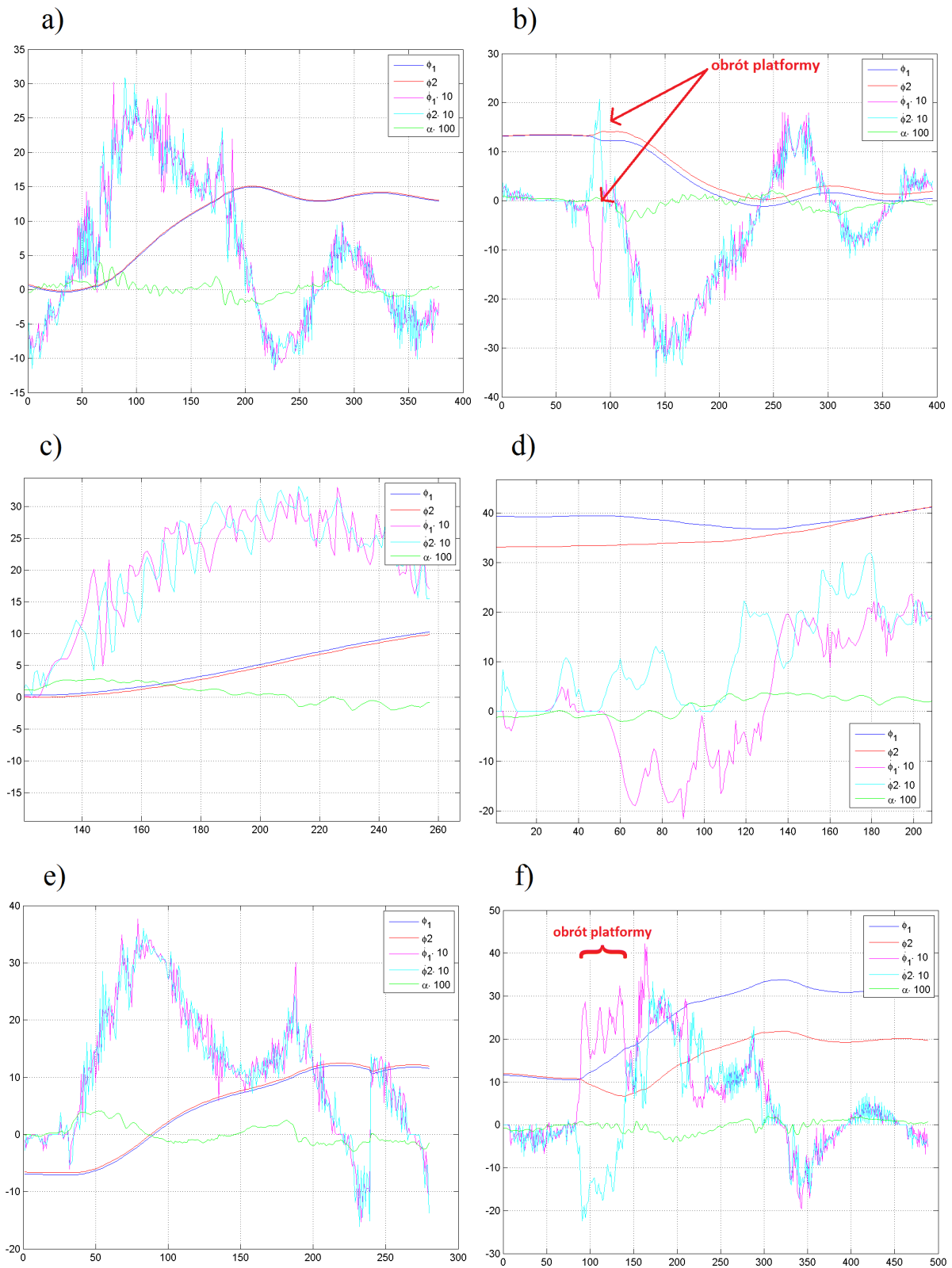
Rysunek 3.6: Elementy składowe platformy jezdnej



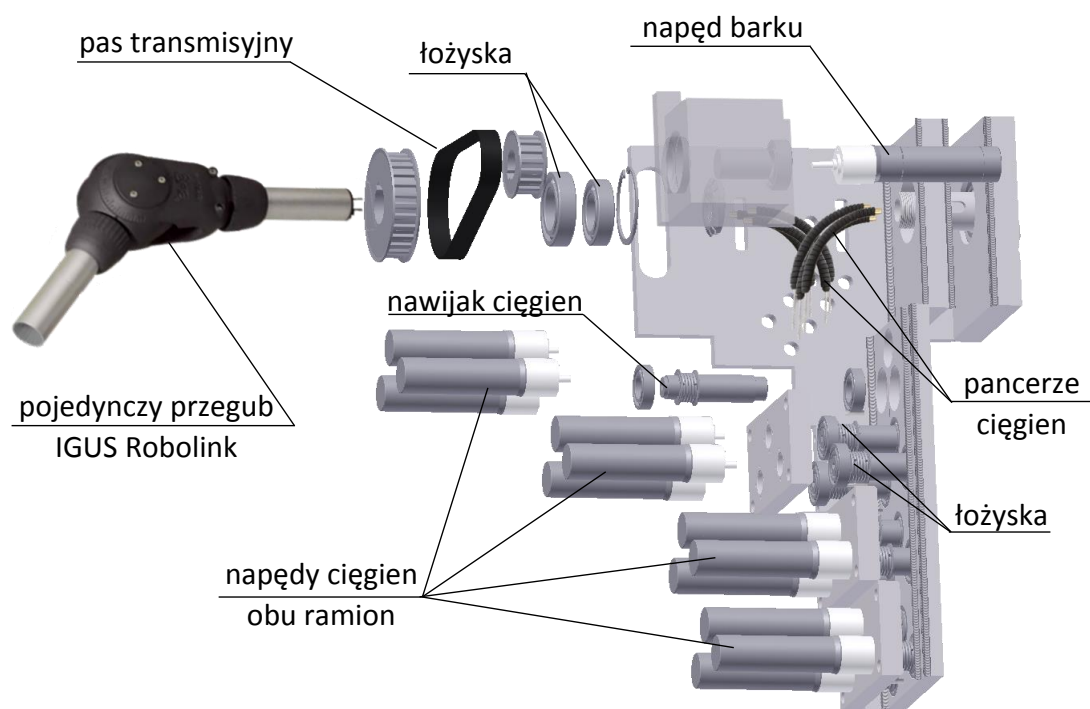
Rysunek 3.7: Schemat blokowy oprogramowania sterującego platformą

3.3 Korpus wraz z ramionami

Układ napędowy ramion w całości został osadzony w korpusie robota, który zaprezentowano na rysunku 3.9. Strukturę kinematyczną rąk robota przedstawiono na rysunku 3.10a. Szczegółowe informacje na temat budowy korpusu, ramion oraz dłoni można znaleźć w pracach [134, 27]. Można zauważyć, że struktury kinematycznej ręki jest zbliżona do ręki człowieka, posiada bowiem trzy stopnie swobody w barku q_1, q_2, q_3 , łokieć o jednym stopniu swobody q_4 , jeden stopień swobody q_5 pomiędzy nadgarstkiem a łokciem i nadgarstek o dwóch stopniach swobody q_6, q_7 . Ręce robota zostały zbudowane w oparciu o przeguby Robolink firmy IGUS [217]. Są to elementy charakteryzujące się prostą, zwartą konstrukcją o niewielkich wymiarach i masie, które można łatwo wzajemnie łączyć za pośrednictwem dedykowanych, lekkich rurek z tworzywa sztucznego bądź aluminium. Przeguby Robolink posiadają dwa stopnie swobody: zginanie i obrót wzdłuż głównej osi. Są one napędzane za pośrednictwem wytrzymałych linek z tworzywa sztucznego – dwie linki na każdy stopień swobody, łącznie cztery na przegub. W każdej ręce znajdują się dwa przeguby Robolink, z którymi związane są następujące stopnie swobody: q_2 (zginanie), q_3 (obróć), q_4 (zginanie) i q_5 (obróć). Budowa nadgarstka opiera się na przegubie kulowym, w którym zablokowano ruch rotacyjny wzdłuż osi ręki. W wyniku tego, uzyskano przegub o dwóch stopniach swobody: odwodzenie q_6 i zginanie q_7 . Ich napęd również odbywa się za pośrednictwem linek z tworzywa sztucznego – łącznie cztery linki. Silniki umieszczone są w korpusie robota, linki prowadzone są do silnika w dedykowanych pancierzach. Pancerze opierają się na przelotowej śrubie naciągowej umożliwiającej regulację napięcia linek. Przegub q_1 , związany z ruchem obrotowym ręki względem korpusu, jest napędzany za pośrednictwem przekładni pasowej, bez wykorzystania linek. Momenty napędowe silników, ich moce i zastosowane przekładnie zostały dobrane odpowiednio do zapotrzebowania na generowaną siłę, prędkość maksymalną oraz występujące opory ruchu. Przeguby q_3 są napędzane szczotkowymi silnikami prądu stałego RE-25 o mocy 20W firmy MAXON,



Rysunek 3.8: Pomiary stanu platformy podczas wykonywania różnych ruchów. a) przejazd i zatrzymanie, b) obrót, powrót i zatrzymanie, c) rozpędzanie platformy, d) obrót platformy i rozpędzanie, e) przejazd i zatrzymanie, f) obrót i przejazd.

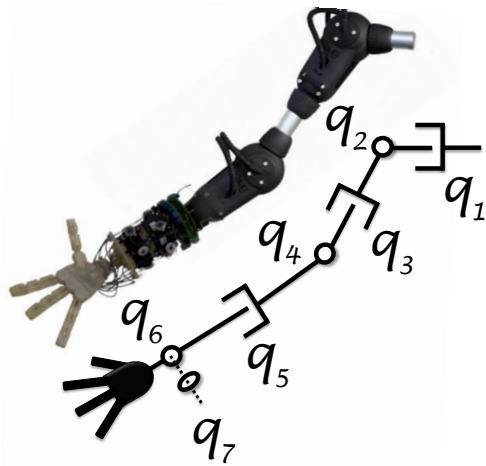


Rysunek 3.9: Konstrukcja mechaniczna korpusu robota

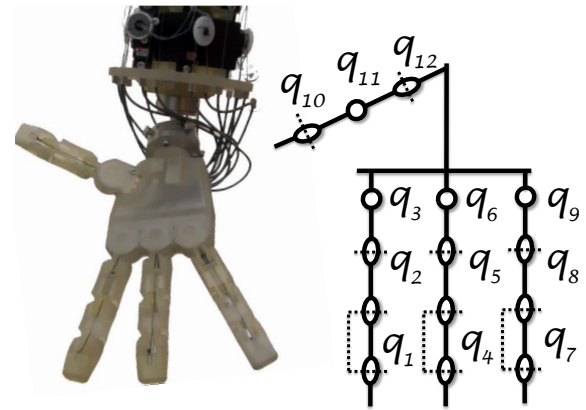
poprzez przekładnię o przełożeniu 190:1. Przeguby q_6 i q_7 są napędzane takimi samymi silnikami poprzez przekładnię o przełożeniu 132:1. Pozwala to na uzyskanie momentów, odpowiednio dla zestawienia, 34Nm oraz 24Nm. Pozostałe przeguby ramion q_1 , q_2 , q_4 , q_5 napędzane są silnikami RE-30 o mocy 60W poprzez przekładnię 103:1 co pozwala na wygenerowanie momentu równego 67Nm. Na wszystkich napędach fabrycznie zainstalowano kodery kwadraturowe o rozdzielczości 256 impulsy/kanal.

Struktura kinematyczna dłoni robota została przedstawiona na rysunku 3.10b. Dłoń składa się z trzech identycznych palców: wskazującego, środkowego i serdecznego oraz jednego przeciwstawnego palca pełniącego rolę kciuka. Palce wskazujący, środkowy i serdeczny posiadają po cztery przeguby obrotowe. Przeguby łączące paliczek dalszy ze środkowym i środkowy z paliczkiem bliższym są sprzężone, z tego względu przeguby te reprezentuje tylko jeden stopień swobody, odpowiednio: palec wskazujący q_1 , palec środkowy q_4 i palec serdeczny q_7 . Paliczek bliższy jest połączony z dłonią za pośrednictwem przegubu o dwóch stopniach swobody, umożliwiających zginanie (q_2 , q_5 , q_8) i odwodzenie (q_3 , q_6 , q_9). Kciuk jest zbudowany w nieco odmienny sposób. Posiada trzy przeguby obrotowe umożliwiające, odpowiednio: ruch do wnętrza dłoni q_{12} , odwodzenie q_{11} , zginanie q_{10} . Łącznie każda dłoń posiada dwanaście stopni swobody. Pomijając brak małego palca, funkcjonalnie struktura kinematyczna dłoni jest zbliżona do dłoni ludzkiej. Dłoń została wykonana w technologii szybkiego prototypowania MJM. W przypadku przegubów związanych z odwodzeniem palców, tj. q_3 , q_6 , q_9 , q_{11} oraz przegubu q_{12} , siła napędowa przenoszona jest w pełni za pośrednictwem linek – dwie linki na przegub. Pozostałe przeguby q_1 , q_2 , q_4 , q_5 , q_7 , q_8 i q_{10} są napędzane w odmienny sposób. W każdym palcu zamontowano sprężyny zegarkowe, generujące siłę potrzebną do jego wyprostowania. Zginanie odbywa się w standardowy sposób, siła generowana przez napędy, przenoszona jest do przegubu za pośrednictwem linek – jedna linka na każdy przegub. Dłoń o podobnej konstrukcji została zaprezentowana w pracy [24]. Napęd dłoni stanowi dwanaście zmodyfikowanych serwonapędów.

a



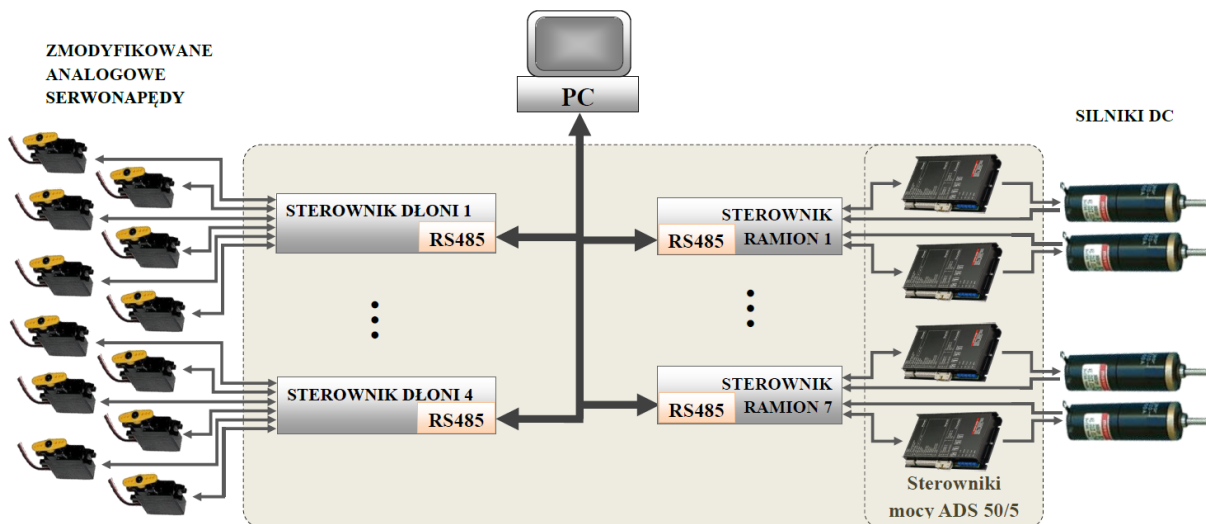
b



Rysunek 3.10: Struktura kinematyczna a – ręki, b – dłoni robota FLASH

prędkości modelarskich HS-82 MG firmy Hitec [106]. Pojedynczy napęd pozwala na wygenerowanie momentu równego 0,03Nm. Serwonapędy umieszczone są w specjalnej ramie z tworzywa sztucznego, wykonanej w technologii szybkiego prototypowania SLS. Rama zamontowana jest w przedramieniu robota. Linki prowadzone są do poszczególnych silników w dedykowanych pancierzach. Pancerze, od strony ramy nośnej, opierają się na przelotowych śrubach służących do regulacji naciągu linek.

Struktura systemu sterowania rąk i dłoni została przedstawiona na rysunku 3.11. System ten kontroluje ruch poszczególnych przegubów rąk i dłoni, został zbudowany



Rysunek 3.11: Schemat blokowy systemu sterowania rąk i dłoni robota FLASH

w oparciu o niskopoziomowe, rozproszone sterowniki ruchu własnej konstrukcji. Zostały one zbudowane z użyciem mikrokontrolera *Stellaris* firmy Texas Instruments z rdzeniem ARM Cortex-M3 [234]. Zadaniem sterowników ruchu jest regulacja pozycji, prędkości i/lub momentu napędowego. Komunikują się one z komputerem nadrzędnym za pośrednictwem wybranego interfejsu komunikacyjnego, obecnie jest to interfejs RS485 z protokołem Dynamixel [66]. W robocie FLASH zastosowano dwa typy sterowników ruchu:



Rysunek 3.12: Leonardo – jeden z bohaterów serialu *Wojownicze Żółwie Ninja* oraz głowa EMYS

dwukanałowy z wejściami/wyjściami analogowymi, przystosowany do pracy ze stopniami mocy ADS 50/5 firmy Maxon Motor [210] i sześciokanałowy, kompaktowy, ze zintegrowanymi stopniami mocy, o niewielkich gabarytach i wadze, przystosowany do pracy ze zmodyfikowanymi serwami modelarskimi. Modyfikacja serw modelarskich, polega na usunięciu z nich elektroniki sterującej i podłączeniu silnika oraz czujnika pozycji bezpośrednio do sterownika ruchu. Na komputerze nadrzędnym uruchamiana jest kompetencja gestykulacji, które zadaje poszczególnym sterownikom pożądane parametry ruchu.

3.4 Głowa EMYS

Przed przystąpieniem do budowy głowy robota FLASH, opracowano kilkanaście koncepcji kształtu oraz sposobów rozmieszczenia ruchomych elementów twarzy, takich jak oczy, brwi, powieki, usta, itd. Ostatecznie zdecydowano, aby głowa składała się z trzech zainstalowanych równolegle nad sobą dysków. Inspiracją tego pomysłu byli fikcyjni bohaterowie komiksu, serialu animowanego oraz filmów fabularnych *Wojownicze Żółwie Ninja* (Rysunek 3.12) [144].

Dwa ruchome dyski (górny i dolny) stanowią środki wyrazu robota. Dolny element imituje ruch szczęki, górny ruch brwi i marszczenie czoła. Pod względem wyrażania emocji czy gestów, oczy wraz z brwiami i powiekami należy traktować jako całość. Powieki i brwi stanowią jeden detal, poruszający się w dwóch osiach: obrót na boki oraz zasłonięcie oka. Obrót oczu wraz z powiekami na boki powoduje wzmocnienie generowanych emocji. Przykładowo, wyrażanie smutku odbywa się poprzez obrócenie ich na zewnątrz, natomiast złość daje się wyrazić poprzez obrót do wewnątrz. Dodatkowo, oczy wyposażono w ruch translacyjny, pozwalający na ich wysunięcie na kilka centymetrów. Ruch ten dostarcza dodatkowego środka wyrazu, jak zaskoczenie, często spotykanego w kreskówkach. Przykładowe miny takie jak zniesmaczenie, strach, zaskoczenie, złość, radość smutek zaprezentowano na rysunku 3.13.

Głowa posiada 11 stopni swobody (3 – szyja, 2 – elementy głowy, 2 – oczy, 4 – powieki). Rozmieszczenie przegubów zaprezentowano na rysunku 3.14. Gwałtowne ruchy masy w górnej części korpusu mają niepożądany wpływ na zachowanie całej platformy. Z tego względu, konstrukcja mechaniczna głowy została wykonana z lekkich, aluminiowych detali. Elementy zewnętrzne zostały wydrukowane przy pomocy metod szybkiego prototypowania 3D, wykorzystujących technologię SLS. Szkielet głowy stanowi element rurowy, do którego przymocowano dolny i górny dysk. Całość została osadzona na szyi, składa-



(a) Złość



(b) Zniesmaczenie



(c) Strach



(d) Radość



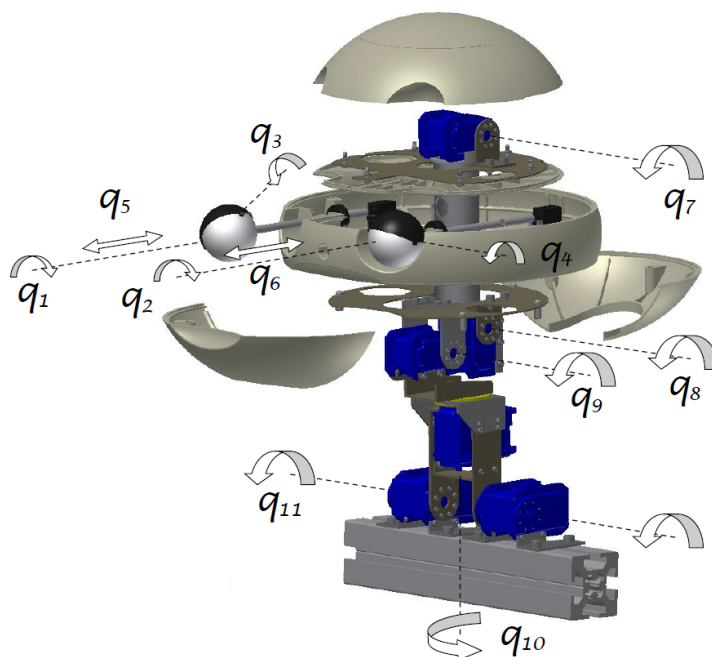
(e) Smutek



(f) Zaskoczenie

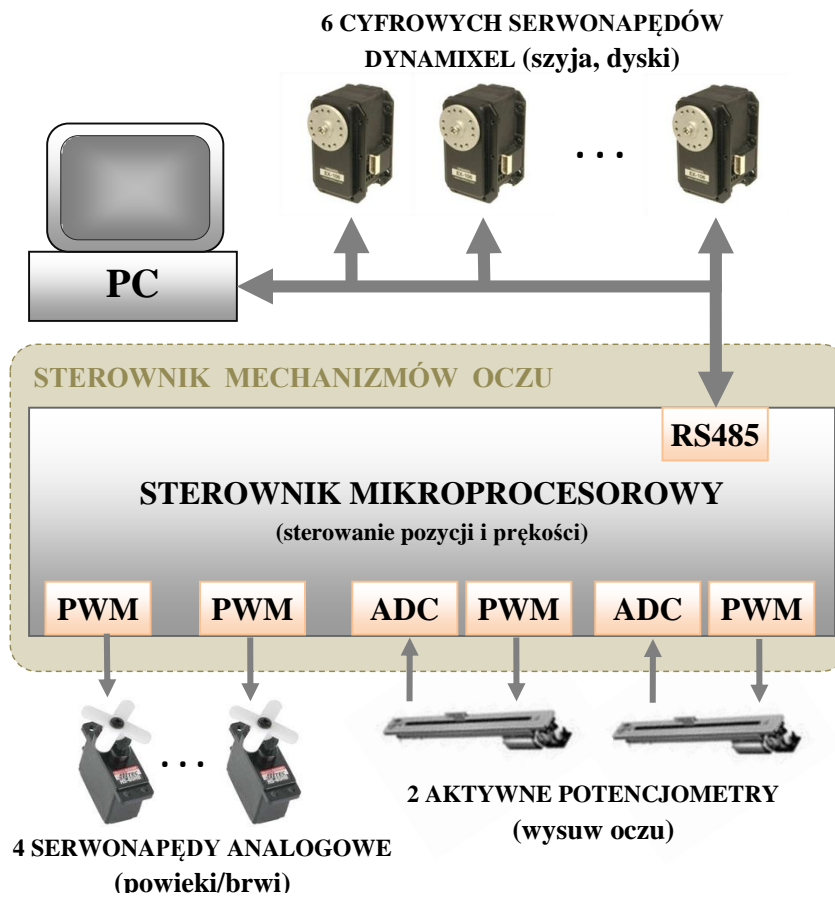
Rysunek 3.13: Przykładowe wyrazy emocji głowy EMYS

jącej się z napędów oraz drobnych elementów aluminiowych. Szyja powinna umożliwiać naturalne wodzenie oczami za rozmówcą, rozglądanie się, pozwalać na gest skinienia, a nawet potrząsania głową. Ruch szyi powinien być przede wszystkim płynny. W konstrukcji

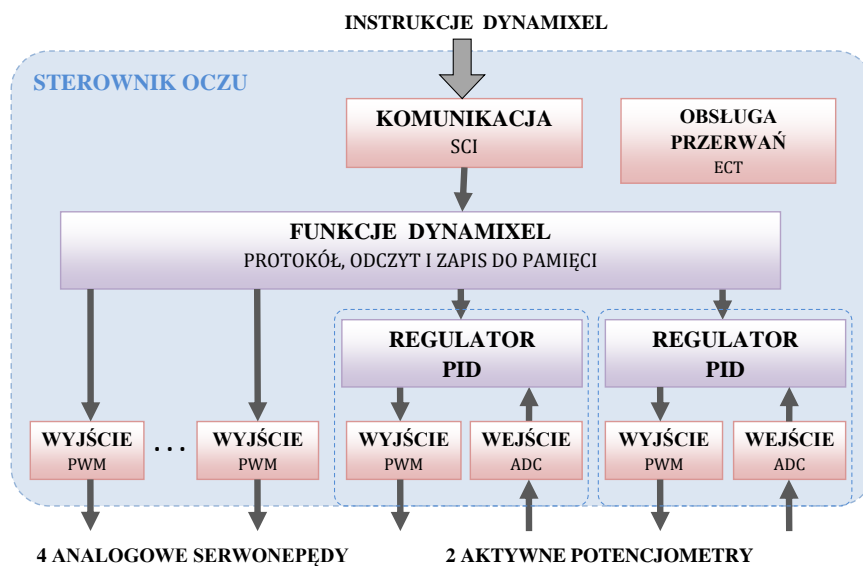


Rysunek 3.14: Rozmieszczenie przegubów głowy

szyi zainstalowano cztery, wysokiej jakości, serwonapędy Dynamixel firmy Robotis. Dwa z nich odpowiadają za pochylanie głowy i są to modele RX-64 generujące moment 7,6Nm. Napędy te pracują synchronicznie w tej samej osi, co dostarcza dwukrotnie większego momentu w tym przegubie. Kolejny identyczny napęd odpowiada za obracanie głowy. Czwarty, odpowiedzialny za kiwanie głową, to nieco słabszy model RX-28 generujący moment 3,7Nm. Takie same napędy zostały użyte do wprawiania w ruch górnego i dolnego talerza głowy. Rozdzielczość serwonapędów Dynamixel wynosi 1024 impulsy przy zmianie kąta o 300° . Do komunikacji wykorzystują asynchroniczną transmisję szeregową. Każdy z nich ma przypisany własny, konfigurowalny numer ID, który determinuje urządzenie docelowe. Każdy mechanizm umożliwia konfigurację jego podstawowych parametrów ruchu, takich jak maksymalny moment obrotowy, prędkość obrotowa lub zakresy wychYLENIA. Możliwe jest także odczytywanie aktualnej informacji o stanie napędu (temperatura, wystąpienie przeciążenia lub przegrzania, napięcie zasilania układu). Ruch oczu i powiek powinien być bardzo szybki. W tym przypadku użyto serwonapędów klasy mikro HS-65HB firmy Hitec, używanych powszechnie w modelarstwie. Moment generowany przez ten napęd wynosi 0,18Nm. Do wysuwania oczu niezbędne było użycie napędów liniowych. W tym celu zainstalowano aktywne potencjometry suwakowe firmy Alps stosowane w profesjonalnym sprzęcie audio (suwak potencjometru napędzany jest mikrosilnikiem elektrycznym). Jednostkę centralną stanowi mikrokontroler HC9S12A64 firmy Freescale, który generuje sygnały sterujące napędami modelarskimi, a także realizuje algorytm regulacji PID do poruszania aktywnymi suwakami. Od strony komputera, sterownik ten jest interpretowany jako sześć niezależnych serwonapędów Dynamixel. Strukturę systemu sterowania zaprezentowano na rysunku 3.15, a schemat blokowy oprogramowania niskopoziomowego na rysunku 3.16.



Rysunek 3.15: Schemat blokowy systemu sterowania napędami



Rysunek 3.16: Schemat blokowy oprogramowania sterującego mechanizmem oczu

Rozdział 4

Platforma balansująca – model i algorytmy sterowania

Jak wspomniano w rozdziale 2, platforma balansująca posiada lepsze własności jezdne niż statyczne podwozie. Ponadto, jej sposób poruszania się ma pozytywny wpływ na postrzeganie robota przez człowieka. Ten istotny fakt brano również pod uwagę w trakcie projektowania sterownika platformy. Powinien on nie tylko stabilnie utrzymywać korpus w równowadze, lecz także pozwolić systemowi nawigacji na przemieszczanie platformy, w tym pokonywanie nierówności powierzchni mieszkalnych, jak np. progi. Co więcej, w fizycznym kontakcie z użytkownikiem robot powinien być podatny na kontakt fizyczny z człowiekiem. W literaturze, tego typu konstrukcje często są nazywane mobilnym odwróconym wahadłem z dwukołowym, różnicowym układem napędowym [150, 235, 93].

Do poprawnej pracy robota niezbędne jest wyposażenie go w układ pomiarowy, który dostarczy informacji o prędkościach obydwu kół i o odchyleniu korpusu od pionu. W dziedzinie sensorów można obecnie zaobserwować rozwój miniaturowych, czujników elektronicznych, takich jak żyroskopy, akcelerometry, inklinometry oraz magnetometry. Często w swej strukturze zawierają one zintegrowane układy filtracji i komunikacji. Odpowiednio dobrany zestaw czujników dostarcza wielowymiarowej informacji, która jest wynikiem pomiarów różnych wielkości fizycznych, np. prędkości kątowej i przyspieszenia liniowego. W poszukiwaniu bezwzględnego odchylenia korpusu od pionu, stosuje się algorytmy fuzji i filtracji w postaci estymatorów stanu, których efektem pracy jest odsumiony i stabilny pomiar.

Pierwszych doświadczeń z obiektem rzeczywistym nabrano budując robota Cosmos [133]. Po wielu próbach przeniesienia liniowych sterowników uzyskanych w trakcie symulacji na obiekt rzeczywisty udało się doprowadzić jedynie do tego, że konstrukcja utrzymywała równowagę, a układ nie wzbudzał się. Możliwe było również sterowanie orientacją platformy. Niestety, niska jakość napędów przybliżony model matematyczny robota, a także niekompletny algorytm sterowania, nie pozwoliły na realizację ruchu wzdłużnego. Dodanie wejść związanych z jazdą w przód powodowało, że ruch platformy stawał się niestabilny, a sterownik wzbudzał się, co w ostateczności doprowadzało do upadku robota. Zastosowane niskiej jakości czujniki inercyjne były nieskompensowane temperaturowo, co wymuszało wykonywanie częstych kalibracji jednostki pomiarowej. Wyjścia pomiarowe mogły być obciążone zakłóceniami typu nie-gaussowskiego. Efektem zastosowania standardowego filtra Kalmana (KF) mogło być uzyskanie błędnej informacji o odchyleniu korpusu względem siły grawitacji.

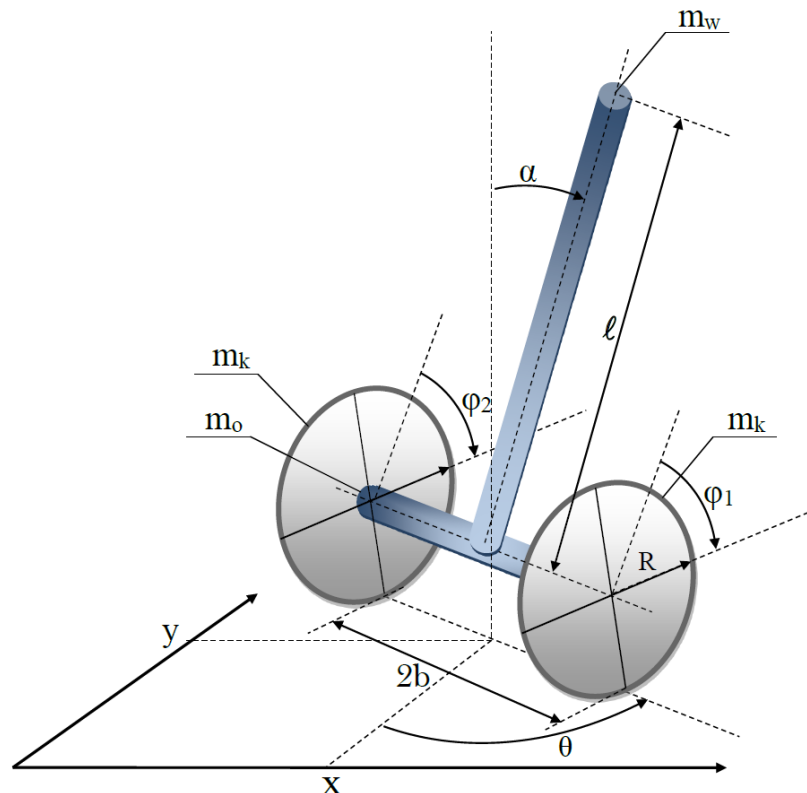
Z tych powodów zdecydowano, aby na nowo wyznaczyć model dynamiki i kinematyki robota [122] oraz przeprowadzić szereg symulacji, na podstawie których możliwe będzie po-

nowne opracowanie zarówno liniowych jak i sterowników nieliniowych. Ponadto, zbadano własności bezśladowego filtra Kalmana (UKF), który może być stosowany w nieliniowych systemach, których wejścia i wyjścia obciążone są zakłóceniami typu nie-gaussowskiego.

Wszystkie obliczenia numeryczne i symulacje przeprowadzono w środowisku MATLAB [168], natomiast do obliczeń symbolicznych wykorzystano oprogramowanie Mathematica [167]. Ponadto, do rozwiązywania problemów optymalnego sterowania zastosowano narzędzie ACADO Toolkit [108, 107, 7].

4.1 Model dynamiki robota balansującego

Robot balansujący będzie traktowany jako dwukołowa, nieholonomiczna platforma klasy 2.0, napędzana względem pionowo utrzymywanej masy. Masa ta zainstalowana jest na osi, w której osadzono napędy kół. Przyjęto założenie, że zarówno oś, jak i wahadło mają postać pręta jednorodnego, a koła zamodelowano jako dwa identyczne cienkie dyski. Wyznaczony, w oparciu o zasadę d'Alemberta [240], model dynamiki i kinematyki ma postać afinicznego układu sterowania z ośmiowymiarową przestrzenią stanu i dwoma wejściami sterującymi, odpowiadającymi momentom napędowym kół. Tego typu obiekty w literaturze nazywane są układami niedosterowanymi (*underactuated*) [207, 232, 57, 241] lub układami z deficytem napędów. Schemat robota, zawierający definicję jego współrzędnych kinematycznych oraz parametrów geometrycznych i dynamicznych, przedstawia rysunek 4.1.



Rysunek 4.1: Schematyczny rysunek robota balansującego

4.1.1 Kinematyka

Robot został opisany we współrzędnych uogólnionych $q = (x, y, \theta, \phi_1, \phi_2, \alpha) \in \mathcal{R}^6$, gdzie (x, y) są współrzędnymi środka osi robota, kąt θ jest jego orientacją, kąty ϕ_1 i ϕ_2 oznaczają obroty kół, a kąt α opisuje odchylenie korpusu od pionu. O nieholonomiczności robota decydują przyjęte założenia o braku poślizgów bocznych i wzdłużnych. Założenia te pozwalają na uzyskanie trzech ograniczeń w postaci Pffafa

$$A(q)\dot{q} = \begin{bmatrix} \sin \theta & -\cos \theta & 0 & 0 & 0 & 0 \\ \cos \theta & \sin \theta & b & -R & 0 & -R \\ \cos \theta & \sin \theta & b & 0 & -R & -R \end{bmatrix} \begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \\ \dot{\phi}_1 \\ \dot{\phi}_2 \\ \dot{\alpha} \end{pmatrix} = 0.$$

Z warunku Pffafa $A(q)\dot{q} = 0$ wynika, że prędkości kół i wahadła należą do jądra macierzy $A(q)$, czyli $\dot{q} \in \text{Ker}A(q)$. Należy znaleźć wektory rozpinające to jądro, czyli

$$\dot{q} = G(q)\eta,$$

gdzie $\eta \in \mathcal{R}^k$, $k = n - l$, n -liczba zmiennych stanu, l -liczba ograniczeń. Macierz $G(q)$ spełnia zatem warunek

$$A(q)G(q) = 0$$

i można ją wybrać w następujący sposób

$$G(q) = \begin{bmatrix} \cos \theta & \cos \theta & \cos \theta \\ \sin \theta & \sin \theta & \sin \theta \\ \frac{1}{b} & -\frac{1}{b} & 0 \\ \frac{2}{R} & 0 & 0 \\ 0 & \frac{2}{R} & 0 \\ 0 & 0 & \frac{1}{R} \end{bmatrix}, \quad \eta = \begin{pmatrix} \eta_1 \\ \eta_2 \\ \eta_2 \end{pmatrix}. \quad (4.1)$$

Uzyskano model kinematyki robota, gdzie współrzędne pomocnicze η_1 oraz η_2 mają sens przeskalowanych prędkości kół, a η_3 przeskalowanej prędkości wahadła.

4.1.2 Dynamika

W celu wyznaczenia kompletnego modelu dynamiki robota należy wyznaczyć energię kinetyczną K korpusu, osi oraz kół, a także energię potencjalną V wahadła. Następnie, przy pomocy wyznaczonych energii, należy zapisać równanie Lagrangea układu swobodnego

$$L(q, \dot{q}) = K(q, \dot{q}) - V(q),$$

w którym energię kinetyczną K można przedstawić w kwadratowej formie

$$K = \frac{1}{2}\dot{q}^T Q(q)\dot{q}.$$

Z diagonalną macierzą bezwładności

$$Q(q) = \begin{bmatrix} Q_{11} & 0 & Q_{13} & 0 & 0 & Q_{16} \\ 0 & Q_{22} & Q_{23} & 0 & 0 & Q_{26} \\ Q_{31} & Q_{32} & Q_{33} & 0 & 0 & 0 \\ 0 & 0 & 0 & Q_{44} & 0 & Q_{46} \\ 0 & 0 & 0 & 0 & Q_{55} & Q_{56} \\ Q_{61} & Q_{62} & 0 & Q_{64} & Q_{65} & Q_{66} \end{bmatrix},$$

gdzie poszczególne elementy tej macierzy są równe

$$\begin{aligned}
Q_{11} &= Q_{22} = m_w + 2m_k + m_o, \\
Q_{33} &= m_w l^2 \sin^2 \alpha + 2m_k b^2 + m_k R^2 + \frac{1}{3} m_o b^2 \\
Q_{44} &= Q_{55} = m_k R^2, \\
Q_{66} &= 2R^2 m_k + m_w l^2, \\
Q_{13} &= Q_{31} = -m_w l \sin \alpha \sin \theta, \\
Q_{23} &= Q_{32} = m_w l \sin \alpha \cos \theta, \\
Q_{16} &= Q_{61} = M_w l \cos \alpha \cos \theta, \\
Q_{26} &= Q_{62} = M_w l \cos \alpha \sin \theta, \\
Q_{46} &= Q_{56} = Q_{64} = Q_{65} = M_k R^2.
\end{aligned}$$

Wyznaczona energia potencjalna ma postać

$$V(q) = m_w l g \cos \alpha.$$

Korzystając z macierzy inercji $Q(q)$ oraz wzoru na symbole Christoffela ,

$$c_{ij}^k(q) = \frac{1}{2} \left(\frac{\partial Q_{ik}(q)}{\partial q_j} + \frac{\partial Q_{jk}(q)}{\partial q_i} - \frac{\partial Q_{ij}(q)}{\partial q_k} \right)$$

można wyznaczyć macierz sił odśrodkowych i Coriolisa

$$C_{kj}(q, \dot{q}) = \sum_{i=1}^6 c_{ij}^k(q) \dot{q}_i$$

Na podstawie wyznaczonej energii potencjalnej uzyskano wektor sił grawitacji $D(q)$, jaki działa na masę wahadła. Stosując zasadę d'Alamberta, dla układów z ograniczeniami nieholonomicznymi, otrzymano model dynamiki robota we współrzędnych uogólnionych

$$Q(q)\ddot{q} + C(q, \dot{q})\dot{q} + D(q) = F + Pu, \quad (4.2)$$

gdzie wektor $F = A^T(q)\lambda$, $\lambda \in \mathcal{R}^3$ to siły więzów wymuszające spełnienie ograniczeń nieholonomicznych, które nie wykonują pracy na dopuszczalnych trajektorjach układu. Macierz wejścia P ma postać

$$P = k_m \begin{bmatrix} 0_{3 \times 2} \\ I_2 \\ 0_{1 \times 2} \end{bmatrix}, \quad u = \begin{pmatrix} u_1 \\ u_2 \end{pmatrix}.$$

gdzie k_m to stosunek wytworzonego momentu do prądu płynącego przez cewki silnika, a u to wartość zadanego prądu.

Wyprowadzony model dynamiki robota 4.2 we współrzędnych uogólnionych można wyrazić w postaci modelu we współrzędnych pomocniczych [171] eliminując przy tym mnożniki Lagrange'a λ . Korzystając z zależności $\dot{q} = G(q)\eta$ wyznaczono $\ddot{q} = \dot{G}(q)\eta + G(q)\dot{\eta}$ i dokonując ich podstawienia do 4.2 uzyskano model dynamiki we współrzędnych pomocniczych

$$\begin{cases} \dot{q} = G(q)\eta, \\ M(q)\dot{\eta} + N(q, \eta) = Bu, \end{cases} \quad (4.3)$$

gdzie $M(q) = G^T(q)Q(q)G(q)$ oznacza zredukowaną macierz inercji, $N(q, \eta)$ jest wektorem sił odśrodkowych, Coriolisa i grawitacji, a B to stała macierz wejścia.

W modelu zaniedbano tarcie oraz przyjęto napęd typu bezpośredniego (*direct drive*). Wykorzystując powyższy formalizm lagranżowski i uwzględniając ograniczenia fazowe, związane z brakiem poślizgu kół, wyznaczono następujące elementy macierzy modelu dynamiki robota balansującego: dla macierzy bezwładności $M(q) = [M_{ij}]$

$$\begin{aligned} M_{11} &= M_{22} = m_w + 8m_k + \frac{4}{3}m_o + \frac{1}{b^2}(m_w l^2 \sin^2 \alpha + m_k R^2), \\ M_{12} &= M_{21} = m_w + \frac{2}{3}m_o - \frac{1}{b^2}(m_w l^2 \sin^2 \alpha + m_k R^2), \\ M_{13} &= M_{23} = M_{31} = M_{32} = m_w + 4m_k + m_o + \frac{l}{R}m_w \cos \alpha, \\ M_{33} &= m_w + 4m_k + m_o + \frac{2l}{R}m_w \cos \alpha + \frac{l^2}{R^2}m_w, \end{aligned}$$

dla wektora $N(q, \eta) = (N_1, N_2, N_3)^T$

$$\begin{aligned} N_1 &= -\frac{l}{b^2}m_w \sin \alpha (2\eta_2 + \eta_3)(\eta_2 - \eta_1) - \frac{2l^2}{b^2 R}m_w \sin \alpha \cos \alpha (\eta_2 - \eta_1)\eta_3 \\ &\quad - \frac{l}{R^2}m_w \sin \alpha \eta_3^2, \\ N_2 &= \frac{l}{b^2}m_w \sin \alpha (2\eta_1 + \eta_3)(\eta_2 - \eta_1) + \frac{2l^2}{b^2 R}m_w \sin \alpha \cos \alpha (\eta_2 - \eta_1)\eta_3 \\ &\quad - \frac{l}{R^2}m_w \sin \alpha \eta_3^2, \\ N_3 &= -\frac{l}{b^2 R}m_w \sin \alpha (\eta_2 - \eta_1)^2 (R + l \cos \alpha) - \frac{l}{R^2}m_w \sin \alpha \eta_3^2 - \frac{l}{R}m_w g \sin \alpha \end{aligned}$$

oraz dla macierzy wejścia

$$B = \frac{2k_m}{R} \begin{bmatrix} I_2 \\ 0 \end{bmatrix}.$$

Przy wyprowadzaniu modelu przyjęto, że koła mają masy m_k równomiernie rozłożone na swoich obwodach, a ich promienie wynoszą R . Oś łącząca koła ma masę m_o i zerowy moment bezwładności względem osi obrotu kół. Korpus ma masę m_w rozłożoną równomiernie wzdłuż długości l i jest położony symetrycznie w odległości b od każdego z kół. Na potrzeby symulacji i dalszych rozważań teoretycznych przyjęto następujące wartości:

oznaczenie wartość jednostka opis

m_k	1	kg	masa koła
m_w	5	kg	masa wahadła
m_o	10	kg	masa osi
R	0,12	m	promień koła
l	1,5	m	długość wahadła
b	0,3	m	odległość koła od środka platformy
k_m	2,6	N·m/A	współczynnik wytworzonego momentu
g	9,81	m/s ²	przyspieszenie ziemskie

W dalszej części pracy zostaną zaprezentowane algorytmy sterowania robotem balansującym. Zostaną omówione i zbadane symulacyjnie liniowe i nieliniowe metody sterowania.

4.2 Sterowanie liniowe

Układy typu odwrócone wahadło mobilne, w pewnym zakresie ich pracy, można sterować przy pomocy sterowników liniowych. Algorytm ze sprzężeniem zwrotnym wyznacza się na podstawie przybliżenia liniowego układu w punkcie równowagi. Popularne i proste algorytmy PD, a także algorytmy wyznaczone metodą LQR lub PP łatwo implementuje się w praktyce.

4.2.1 Wyznaczenie przybliżenia liniowego w punkcie równowagi

Konstrukcja algorytmów liniowych jest oparta na pierwszej metodzie Lapunowa [127] i wykorzystuje metody projektowania liniowych sterowników rozmieszczających bieguny oraz minimalno-kwadratowych [76]. Do zaprojektowania sterowników niezbędne jest znalezienie przybliżenia liniowego w pobliżu punkcie równowagi układu postaci

$$\dot{x} = f(x) + g(x)u. \quad (4.4)$$

Zapiszmy ponownie model kinematyki i dynamiki robota balansującego

$$\begin{cases} \dot{q} = G(q)\eta, \\ \dot{\eta} = -M^{-1}(q)N(q, \eta) + M^{-1}(q)Bu, \end{cases} \quad (4.5)$$

Zauważmy, że struktura tego modelu pozwala wyodrębnić 6-wymiarowy podsystem $x = (\phi_1, \phi_2, \alpha, \eta_1, \eta_2, \eta_3) \in \mathcal{R}^6$ opisany równaniami

$$\dot{x} = \begin{pmatrix} \mathcal{U}\eta \\ \dot{\eta} \end{pmatrix}, \quad \text{gdzie} \quad \mathcal{U} = \frac{1}{R} \begin{bmatrix} 2 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad (4.6)$$

który determinuje zachowanie całości.

Przybliżenie liniowe powyższego układu zostanie wyliczone w pobliżu trajektorii odniesienia $x_r(t) = (\phi_{1r}(t), \phi_{2r}(t), 0, \eta_{1r}(t), \eta_{2r}(t), 0)$. Jest ona wyliczana na podstawie prędkości liniowej $V_{lin} = const$ i kątovej $V_{rot} = const$ platformy, zadanych z systemu nawigacji. Równania elementów składowych trajektorii referencyjnej mają postać

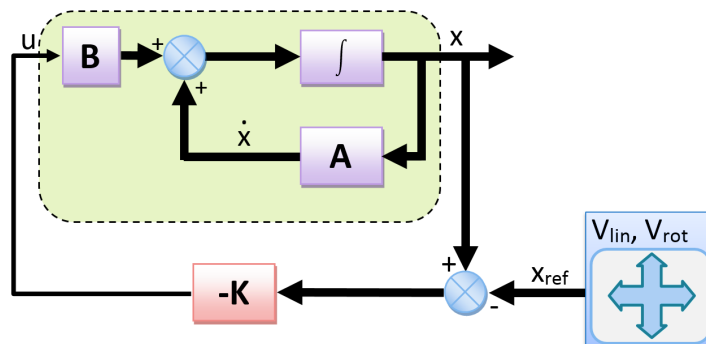
$$\begin{aligned} \eta_{1r}(t) &= \frac{1}{2}R\dot{\phi}_{1r}(t) = \frac{1}{2}R(V_{lin}\frac{1}{R} - V_{rot}\frac{b}{R}) = \frac{1}{2}(V_{lin} - bV_{rot}), \\ \eta_{2r}(t) &= \frac{1}{2}R\dot{\phi}_{2r}(t) = \frac{1}{2}(V_{lin} + bV_{rot}), \\ \phi_{1r}(t) &= \frac{1}{R}(V_{lin} - bV_{rot})t, \quad \phi_{1r}(0) = 0, \\ \phi_{2r}(t) &= \frac{1}{R}(V_{lin} + bV_{rot})t, \quad \phi_{2r}(0) = 0. \end{aligned}$$

Na podstawie (4.5) oraz (4.6) zapisujemy układ w postaci (4.4)

$$f(x) = \begin{pmatrix} \mathcal{U}\eta \\ -M^{-1}(q)N(q, \eta) \end{pmatrix} + \begin{bmatrix} 0 \\ -M^{-1}(q)B \end{bmatrix} u, \quad (4.7)$$

dla którego zostanie wyznaczone przybliżenie liniowe. Biorąc pod uwagę, że $u_r = 0$ przybliżenie liniowe równania (4.7)

$$\dot{x} - \dot{x}_r = A(x_r)(x - x_r) + B(x_r)u, \quad \text{gdzie} \quad A(x_r) = \left. \frac{\partial f(x)}{\partial x} \right|_{x=x_r}, \quad B = g(x_r). \quad (4.8)$$



Rysunek 4.2: Schemat blokowy sterownika dla liniowej wersji systemu

Otrzymane przybliżenie liniowe $\dot{e} = Ae + Bu$ błędu śledzenia trajektorii odniesienia $e = x - x_r$ nie zależy od zadanej prędkości liniowej V_{lin} . Ma to sens fizyczny, gdyż w układzie inercyjnym, który poruszają się ze stałą prędkością nie działają na niego żadne siły. Macierz A jest zależna od prędkości kątowej V_{rot} . Można jednak przyjąć, że jeżeli układ (4.8) jest sterowalny dla $V_{rot} = 0$, własność ta się zachowa przy małych V_{rot} .

Przy założeniu, że $V_{rot} = 0$ i po podstawieniu danych liczbowych, macierze A i B wyglądają następująco

$$A = \begin{bmatrix} 0 & 0 & 0 & 16.67 & 0 & 0 \\ 0 & 0 & 0 & 0 & 16.67 & 0 \\ 0 & 0 & 0 & 0 & 0 & 8.33 \\ 0 & 0 & -2.28 & 0 & 0 & 0 \\ 0 & 0 & -2.28 & 0 & 0 & 0 \\ 0 & 0 & 1.06 & 0 & 0 & 0 \end{bmatrix} \quad B = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 2.36 & -0.53 \\ -0.53 & 2.36 \\ -0.16 & -0.16 \end{bmatrix}$$

Otrzymany liniowy układ sterowania sprawdzono pod względem stabilności i sterowalności. Z pierwszej metody Lapunowa [127] możemy wnioskować o stabilności układu nieliniowego na podstawie liniowego przybliżenia układu w pobliżu stanu równowagi. Wyznaczone wartości własne $\lambda_1 = 0, \lambda_2 = 0, \lambda_3 = 0, \lambda_4 = 0, \lambda_5 = 2.97, \lambda_6 = -2.97$ macierzy A układu liniowego są wielokrotne i mają nieujemne części rzeczywiste. Oznacza, to że układ (4.8) jest niestabilny. Sterowalność układu sprawdzono wyliczając macierz sterowalności Kalmana [127] $\Omega = [B; AB; \dots; A^5B]$. Na podstawie wyliczenia $rank(\Omega) = 6$ można wnioskować, że system jest sterowalny.

4.2.2 Koncepcja sterownika liniowego

Na potrzeby sterowania robotem balansującym zaprojektowano sterownik liniowy korzystając z dwóch metod. Pierwszy z nich to sterownik minimalno-kwadratowy, drugi to sterownik rozmieszczający bieguny. Teoria regulacji korzysta ze sprzężenia zwrotnego od stanu. Wektor stanu przemnożony przez odpowiednią macierz wzmocnień K , podawany jest na wejście systemu, jak pokazano na rysunku 4.2. Prawo sterowania ma postać

$$u = -Ke.$$

Przybliżenie liniowe robota balansującego ma postać (4.8)

$$\dot{e} = Ae + Bu.$$

Stosując powyższe prawo sterowania do modelu (4.8) otrzymano

$$\dot{e} = (A - BK)e = \tilde{A}e,$$

Aby układ z zamkniętą pętlą sprzężenia zwrotnego był stabilny, musi być spełniony warunek $\forall_i \operatorname{Re} \lambda_i(\tilde{A}) < 0$.

4.2.3 Sterownik minimalno-kwadratowy (LQR)

Jednym z najczęściej używanych sterowników liniowych jest LQR (*Linear-quadratic regulator*) [96, 231]. Jest on łatwy do zastosowania w praktyce, zwłaszcza kiedy wszystkie zmienne stanu są mierzalne. Podczas sterowania zakładamy, że system liniowy znajduje się w pobliżu równowagi gdyż dla takich warunków zostało wyznaczone przybliżenie liniowe.

Należy wyznaczyć sterowanie u , które minimalizuje kwadratowy wskaźnik jakości:

$$J = \frac{1}{2} \int_0^{\infty} (e^T(t)Qe(t) + u^T(t)Ru(t)) dt,$$

gdzie Q i R są stałymi, diagonalnymi nastawami regulatora. W praktyce dobiera się wartości elementów macierzy Q i R w następujący sposób:

$$Q_{ii} = 1/\max. \text{ akceptowalna wartość } [e_i^2],$$

$$R_{ii} = 1/\max. \text{ akceptowalna wartość } [u_i^2].$$

Prawo sterowania w sterowniku LQR ma postać

$$u = -R^{-1}B^T P e,$$

gdzie P uzyskuje się poprzez rozwiązanie następującego równania Riccatiego

$$0 = PA + A^T P - PBR^{-1}B^T P + Q.$$

Macierz wzmocnień K opisana jest równaniem

$$K = R^{-1}B^T P$$

i określa w całości postać sprzężenia zwrotnego dla opisywanego systemu. Do wyznaczenia macierzy K użyto środowiska Matlab i dostępnej w nim funkcji lqr

$$[K, P, ev] = lqr(A, B, Q, R),$$

gdzie

- K – macierz wzmocnień sterownika,
- P – macierz Lapunowa,
- ev – wektor wartości własnych układu zamkniętego.

Dla macierzy nastaw $Q = I_6$ oraz $R = I_2$ otrzymano

$$K = \begin{bmatrix} 0 & -1 & -43.03 & -7.61 & -11.15 & -148.19 \\ -1 & 0 & -43.03 & -11.15 & -7.61 & -148.19 \end{bmatrix}.$$

Wartości własne układu z zamkniętą pętlą sprzężenia mają części rzeczywiste mniejsze od zera, zatem układ ze sprzężeniem zwrotnym jest stabilny. Zaobserwowano, że wraz ze wzrostem współczynników macierzy Q malał czas powrotu do stanu równowagi, ale wzrastało przeregulowanie. Wynik symulacji dla parametrów $Q = I_6$ i $R = I_2$ pokazano na rysunku 4.5a, a dla zwiększonych parametrów $Q_{11} = Q_{22} = Q_{33} = 100$ na rysunku 4.5b. Dla obu przypadków przyjęto warunek początkowy $x(0) = [0, 0, \frac{\pi}{6}, 0, 0, 0]^T$.

4.2.4 Sterownik rozmieszczający bieguny (PP)

Kolejnym, równie popularnym podejściem do projektowania sterownika liniowego jest metoda rozmieszczania biegunów PP (*Pole-Placement*) [1, 11, 158]. Metoda sterowania polega na wyznaczeniu takiej macierzy K , aby wynikowa macierz \tilde{A} , równa

$$\tilde{A} = A - BK$$

miała wartości własne spełniające warunek $\forall_i \operatorname{Re} \lambda_i(\tilde{A}) < 0$. Ich optymalne wartości wylicza się na podstawie uogólnionej formuły Ackermanna [265]. Macierz K wyznaczono w środowisku Matlab przy użyciu funkcji *place*

$$K = \text{place}(A, B, p),$$

gdzie p to wektor biegunów $\lambda_i(\tilde{A})$. Jeżeli celem jest szybka odpowiedź, to bieguny powinny leżeć daleko od osi urojonej płaszczyzny zespolonej. Trzeba pamiętać, że szybka odpowiedź na zadane wymuszenie będzie wymagać silnych napędów robota. Przykład wyliczenia macierzy sprzężenia wykonano dla następujących biegunów

$$\lambda_1 = -1, \quad \lambda_2 = -2, \quad \lambda_3 = -3, \quad \lambda_4 = -4, \quad \lambda_5 = -5, \quad \lambda_6 = -6.$$

Stabilność systemu będzie zagwarantowana tak długo, jak długo część rzeczywista $\lambda_i(\tilde{A})$ będzie mniejsza od zera. Macierz wzmocnień K dla powyższych parametrów ma postać

$$K = \begin{bmatrix} 0.06 & -0.19 & -20.78 & -1.61 & -4.18 & -68.78 \\ -0.45 & -0.04 & -34.16 & -7.89 & -4.76 & -116.56 \end{bmatrix}.$$

Zaobserwowano, że wraz z oddalaniem biegunów od osi urojonej malał czas powrotu do stanu równowagi. Wzrastało także przeregulowanie. Wynik symulacji dla parametrów $\lambda = \{-1, -2, -3, -4, -5, -6\}$ zaprezentowano na rysunku 4.5c, dla $\lambda = \{-3, -4, -5, -6, -7, -8\}$ na rysunku 4.5d. Dla obu przypadków przyjęto warunek początkowy $x(0) = [0, 0, \frac{\pi}{6}, 0, 0, 0]^T$.

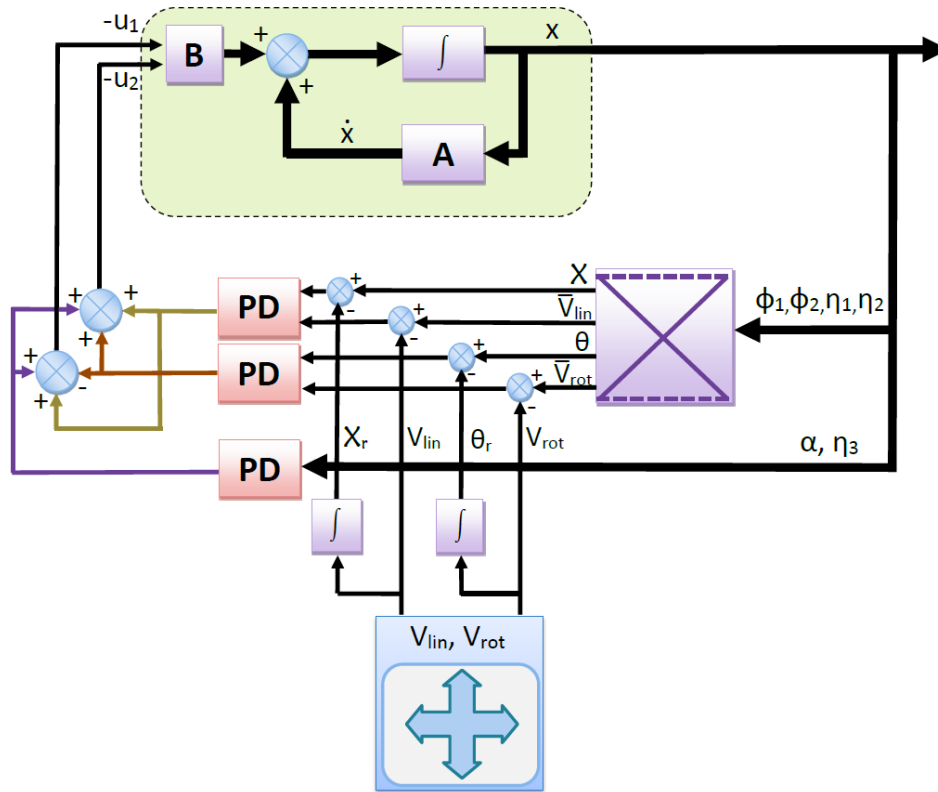
4.2.5 Sterownik PD

Metody LQR i PP pozwalają znaleźć optymalne nastawy regulacji pod warunkiem, że posiada się dostatecznie dobrze zidentyfikowany model robota. Jednak, gdy okaże się, że model nie był zbyt dokładnie wyznaczony, zmiana nastaw i próba ich doboru w sposób eksperymentalny jest trudna i wymaga dużego doświadczenia. Dobrym pomysłem jest przekształcenie wektora stanu x w ruch translacyjny i obrotowy platformy, a następnie zastosowanie trzech pętli regulacji PD dla ruchu translacyjnego, obrotowego i wychylenia wahadła. Dobór nastaw przebiega intuicyjnie, a strojenie eksperymentalne jest łatwiejsze. Schemat koncepcyjny sterownika pokazano na rysunku 4.3.

Dla tego przypadku definiujemy trzy sterowania. Pierwsze związane jest z ruchem wzdłużnym platformy

$$u_{lin} = P_{lin}(\mathcal{X} - \mathcal{X}_r) + D_{lin}(\tilde{V}_{lin} - V_{lin}),$$

gdzie $\mathcal{X} = R(\phi_1 + \phi_2)$ to droga jaką pokonuje platforma, $\mathcal{X}_r = (\phi_{1r} + \phi_{2r})$ to jej wartość odniesienia, \tilde{V}_{lin} to prędkość liniowa platformy, a V_{lin} to prędkość liniowa zadana z systemu nawigacji.



Rysunek 4.3: Schemat blokowy potrójnej pętli sterownika PD

Kolejne sterowanie reprezentuje ruch obrotowy platformy

$$u_{rot} = P_{rot}(\theta - \theta_r) + D_{rot}(\bar{V}_{rot} - V_{rot}),$$

gdzie θ_r to wartość referencyjna orientacji platformy, $\bar{V}_{rot} = \frac{1}{b}(\eta_2 - \eta_1)$ reprezentuje prędkość kątową platformy, a V_{rot} to prędkość kątowa zadana z systemu nawigacji.

Ostatnie sterowanie odpowiada za utrzymywanie wahadła w pionie i ma postać

$$u_{\alpha} = P_{\alpha}\alpha + D_{\alpha}\eta_3.$$

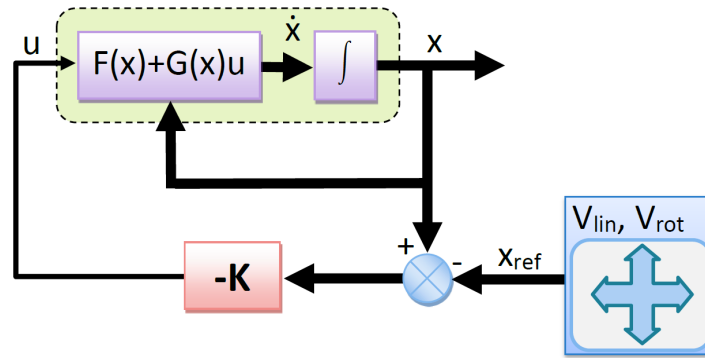
Ostatecznie wszystkie sterowania są sumowane i podawane na wejście układu z przeciwnym znakiem

$$u_1 = -(u_{lin} - u_{rot} + u_{\alpha}), \quad u_2 = -(u_{lin} + u_{rot} + u_{\alpha}).$$

Wynik symulacji dla eksperymentalnie dobranych nastaw przedstawiono na rysunkach 4.5e oraz 4.5f. Warunek początkowy przyjęto $x(0) = [0, 0, \frac{\pi}{6}, 0, 0, 0]^T$.

4.2.6 Zastosowanie liniowego sterownika do modelu nieliniowego

Po wyznaczeniu sterowników dla modeli liniowych, podjęto próbę przetestowania ich na nieliniowym modelu dynamiki robota. Oczekuje się, że przy niewielkim błędzie śledzenia trajektorii x_r , sterownik powinien pracować prawidłowo. Strukturę systemu przedstawiono na rysunku 4.4. Do przeprowadzenia symulacji wybrano po jednej, najszybciej działającej wersji sterownika układu liniowego zaprojektowanego metodą PP oraz LQR. Wyniki przedstawiono na rysunkach, 4.5g dla LQR oraz 4.5h dla PP. Dla obu przypadków przyjęto warunek początkowy $x(0) = [0, 0, \frac{\pi}{6}, 0, 0, 0]^T$.



Rysunek 4.4: Schemat blokowy systemu sterowania

4.3 Sterowanie nieliniowe – metoda linearyzacji

Sterowanie liniowe zapewnia jedynie stabilizację w pobliżu punktu równowagi. W przypadku robota balansującego, który jest obiektem nieliniowym, sterownik działa poprawnie jedynie przy niewielkich odchyleniach wahadła od pionu. W dalszej części pracy rozważany będzie pełny nieliniowy model matematyczny robota balansującego (patrz równanie (4.3)). W celu realizacji śledzenia trajektorii zadanej, np. ruch platformy na płaszczyźnie, zaproponowano nieliniowy sterownik bazujący na linearyzacji wyjść i odprężonym odwzorowaniu we-wy [241], w istocie wykorzystujący różniczkową płaszczyznę układu (4.3) [152]. Dzięki temu, możliwe jest zastosowanie liniowego regulatora [117] do nieliniowego systemu. Zastosowanie tego sterownika dla podobnego układu opisano w pracach [126, 125]. Idea polega na wyborze pary wyjść linearyzujących, następnie poprzez wielokrotne ich różniczkowanie i podstawienia, znalezieniu możliwości odsprężenia układu przy pomocy sprzężenia zwrotnego. Schemat blokowy układu zaprezentowano na rysunku 4.6. Poniżej przedstawiono dwa przypadki. W pierwszym z nich wybrano centralny punkt osi robota, jako jego pozycję w globalnym układzie współrzędnych. Drugie podejście polegało na wyborze kątów odchylenia wahadła i orientacji robota. W podejściu mieszanym, np. współrzędna położenia i kąt wahadła, nie udało się wyznaczyć odsprężenia z powodu trudnych do uniknięcia osobliwości. Sterownik ten został także opisany w pracy [137].

4.3.1 Śledzenie pozycji robota

Jak już wspomniano, dla pierwszego przypadku wybrano wyjścia płaskie, będące pozycją robota

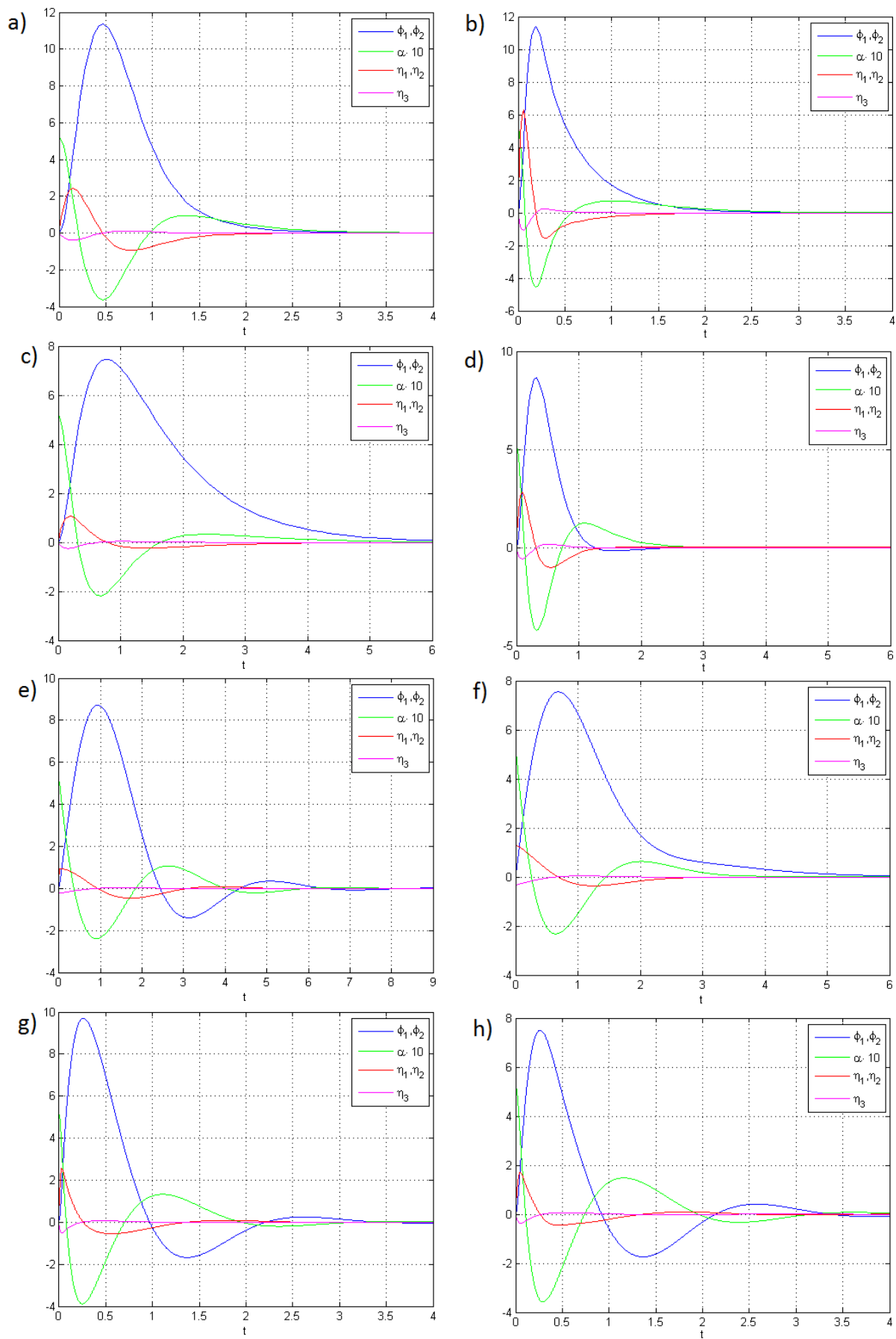
$$y_1 = x, \quad y_2 = y.$$

Zgodnie z procedurą wyznaczania algorytmu sterowania po dwukrotnym zróżniczkowaniu wyjść uzyskano $\ddot{y}_1(t)$, $\ddot{y}_2(t)$. Niestety, oba równania zależą od sumy sterowań $u_1(t) + u_2(t)$, co daje nie wystarczającą liczbę wejść, aby móc zastosować odsprężenie we-wy. W związku z tym wprowadzono nową zmienną stanu $v = u_1 + u_2$ oraz rozbudowano dynamikę układu (4.3) $\dot{v} = w_1$, gdzie w_1 oznacza nowe wejście sterujące. Różniczkując po raz trzeci wyjścia y_1, y_2 wzdłuż trajektorii układu (4.3) uzyskano następującą zależność we-wy

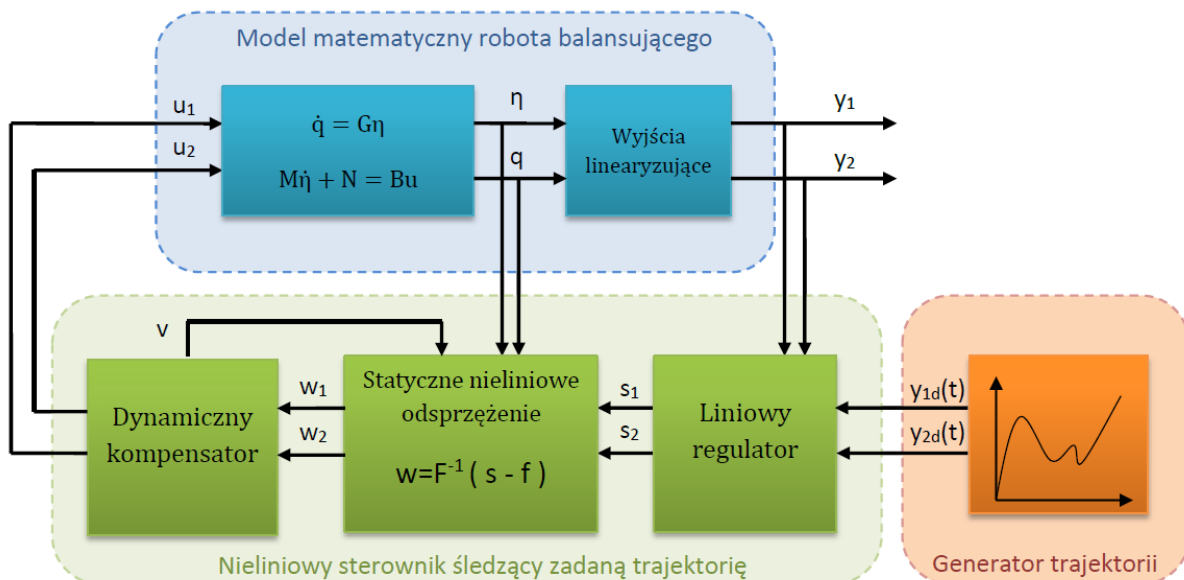
$$y^{(3)} = f(q, \eta, v) + F(q, \eta)w,$$

gdzie $w = (w_1, w_2)$, a $w_2 = u_2 - u_1$ to nowe wejście sterujące. Można teraz wyznaczyć sprzężenie zwrotne,

$$w = F^{-1}(q, \eta)(s - f(q, \eta, v)),$$



Rysunek 4.5: Wyniki symulacji dla sterowników liniowych



Rysunek 4.6: Schemat blokowy sterownika nieliniowego - metoda linearyzacji

które daje dynamiczną linearyzację we-wy

$$y^{(3)} = s.$$

postać funkcji $f(q, \eta, v)$ i $F(q, \eta)$ jest zbyt złożona, żeby ją tu przytoczyć, por. [125]. Warunkiem linearyzacji jest nieosobliwość macierzy F . Niestety, w powyższym układzie pojawia się osobliwość, a w celu jej uniknięcia należy przyjąć, że robot jest cały czas w ruchu $\eta_{123} \neq 0$. Tego ograniczenia można uniknąć stosując zabieg, który zaproponowano w pracy [126].

Śledzenie zadanej trajektorii $y_d(t)$ uzyskano poprzez liniowy sterownik

$$s(t) = y_d^{(3)} - K_2(\ddot{y}(t) - \ddot{y}_d(t)) - K_1(\dot{y}(t) - \dot{y}_d(t)) - K_0(y(t) - y_d(t)),$$

w którym macierze wzmocnienia należy tak dobrać, aby układ był asymptotycznie stabilny.

Wyniki symulacji śledzenia trajektorii $x_d(t) = 10 \sin(0.7t)$ i $y_d(t) = 10 \cos(0.7t)$, dla różnych punktów początkowych przedstawiono na rysunkach 4.7a i 4.7b. Wybór wyjść, jedynie w postaci położenia robota, z praktycznego punktu widzenia nie ma większego sensu. Na rysunkach 4.7c i 4.7d pokazano bowiem, że powyższy algorytm sterowania nie gwarantuje stabilizacji wahadła.

4.3.2 Śledzenie trajektorii orientacji i odchylenia korpusu od pionu

W kolejnym podejściu wybrano alternatywne wyjścia płaskie

$$y_1 = \theta, \quad y_2 = \alpha.$$

Po dwukrotnym zróżniczkowaniu wyjść uzyskano

$$\ddot{y} = f(q, \eta) + F(q, \eta)w,$$

gdzie $w = (w_1, w_2)$, a składowe wektora sterowań są równe $w_1 = u_1 + u_2$ i $w_2 = u_2 - u_1$. Z uwagi na złożoność funkcji $f(q, \eta)$ i $F(q, \eta)$ nie podano explicite ich postaci. Podobnie jak w poprzednim podejściu wyznaczono sprzężenie zwrotne

$$w = F^{-1}(q, \eta)(s - f(q, \eta)),$$

które daje statyczną linearyzację we-wy $\ddot{y} = s$. Śledzenie zadanej trajektorii $y_d(t)$ uzyskano poprzez liniowy sterownik

$$s(t) = \ddot{y}_d(t) - K_1(\dot{y}(t) - \dot{y}_d(t)) - K_0(y(t) - y_d(t)).$$

Warunkiem nieosobliwości macierzy F jest

$$m_w + 4m_k + m_o + \frac{m_w l \cos \alpha}{R} \neq 0.$$

i jest on łatwy do spełnienia.

Wyniki symulacji śledzenia trajektorii $\theta_d(t) = \pi$ i $\alpha_d(t) = 0.1 \cos(2t)$, przedstawiono na rysunkach 4.7e i 4.7f. Wybór wyjść w postaci kątów orientacji i odchylenia wahadła od pionu ma praktyczny sens, nie mniej jednak wciąż nie stanowi kompletnego rozwiązania zadania śledzenia. Na rysunku 4.7g pokazano bowiem, że układ ten nie utrzymuje platformy na pozycji. Położenia x i y nie są kontrolowane.

4.4 Sterowanie nieliniowe – metoda endogenicznej przestrzeni konfiguracyjnej

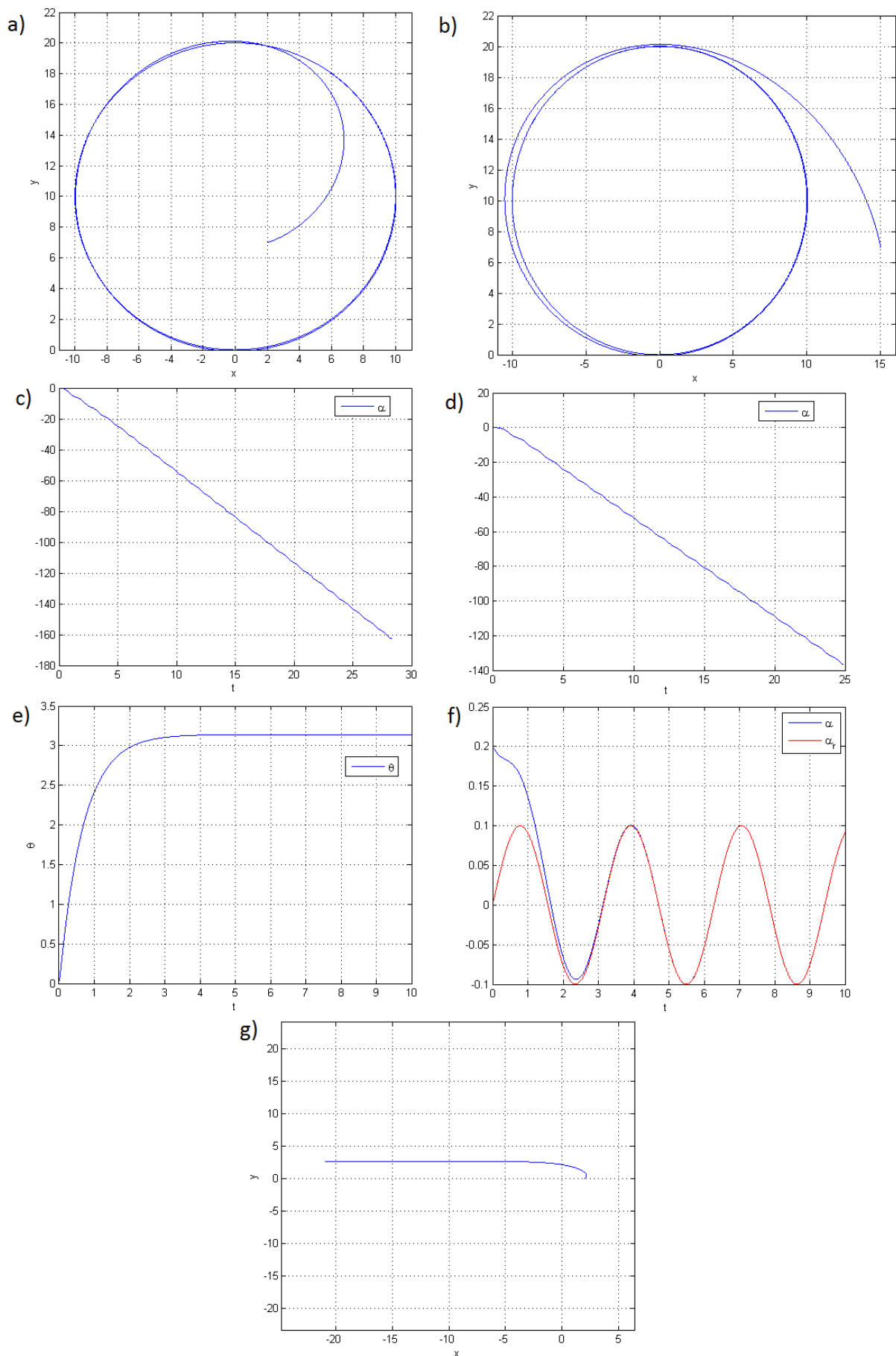
Jednym z kluczowych zagadnień przy sterowaniu manipulatorem lub manipulatorem mobilnym jest rozwiązanie zadania kinematyki odwrotnej i ostatecznie zaplanowanie jego trajektorii. Robot balansujący jest w pewnym sensie manipulatorem mobilnym z deficytem sterowań (dla wahadła). Podjęto próbę wyznaczenia sterowań platformą wykorzystując pełny, nieliniowy model matematyczny robota w oparciu o metodę endogenicznej przestrzeni konfiguracyjnej. W przeciwieństwie do poprzedniego podejścia, możliwe jest zaplanowanie takich sterowań, aby na ich końcu osiągnąć więcej niż dwie zadane składowe wektora stanu. Metoda została szczegółowo opisana w pracy [239], a jej różne modyfikacje i implementacje przedstawiono w pracach [120, 208]. Struktura jednego kroku, dla dyskretnej wersji algorytmu została zaprezentowana na rysunku 4.8. Poniżej znajduje się krótki opis metody.

Przyjmijmy, że matematyczny model robota balansującego zostanie wyrażony w postaci afinicznego układu z dryftem,

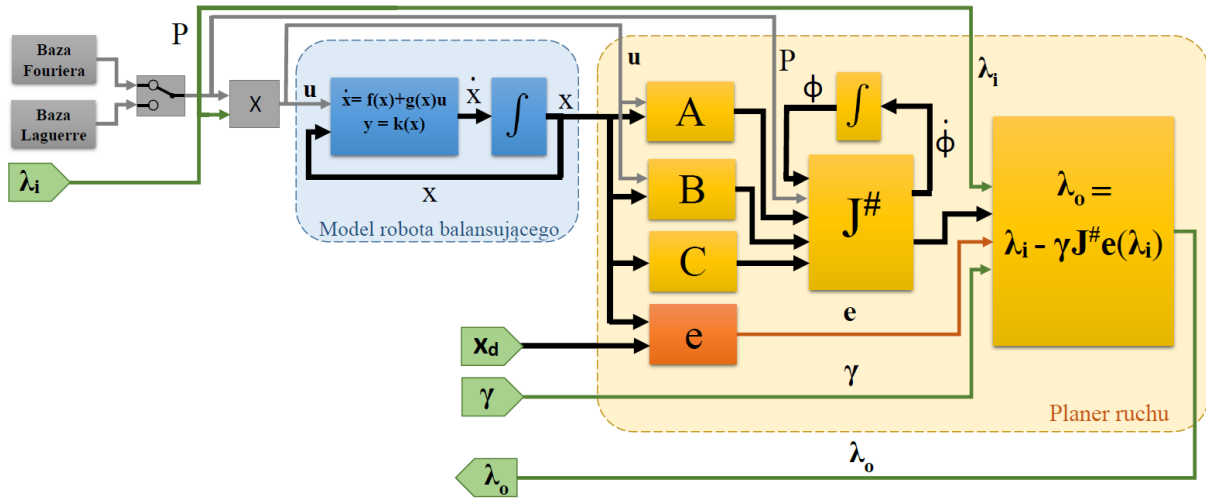
$$\begin{cases} \dot{x} = f(x) + g(x)u \\ y = k(x), \end{cases} \quad (4.9)$$

w którym wektor stanu $x \in \mathcal{R}^n$, wejście układu $u \in \mathcal{R}^m$, a wyjście $y \in \mathcal{R}^r$. Dla zadanego sterowania $u(\cdot) \in \mathcal{X} = L_m^2[0, T]$ oraz punktu początkowego x_0 uzyskuje się trajektorię $x(t) = \varphi_{x_0, t}(u(\cdot))$. Wynik sterowania działającego na pewnym interwale $[0, T]$ może być zapisany jako odwzorowanie końcowe

$$K_{x_0, T}(u(\cdot)) = y(T) = k(\varphi_{x_0, T}(u(\cdot))).$$



Rysunek 4.7: Wyniki symulacji dla sterowników ze zlinearyzowanymi wyjściami



Rysunek 4.8: Struktura pojedynczego kroku dla dyskretnej wersji algorytmu bazującego na metodzie endogenicznej przestrzeni konfiguracyjnej

Przestrzeń liniowa \mathcal{X} będzie w dalszej części nazywana endogeniczną przestrzenią konfiguracyjną. Zadaniem planowania ruchu jest znaleźć taką funkcję sterującą $u(\cdot)$, by na określonym horyzoncie T osiągnąć położenie zadane y_d , tzn.

$$y_d = K_{x_0, T}(u(\cdot)).$$

Korzystając z metody homotopii definiujemy krzywą sterowań $u_\theta(\cdot)$, $\theta \in \mathcal{R}$, taką by błąd $e(\theta) = K_{x_0, T}(u_\theta(\cdot)) - y_d$ spełniał równanie

$$\frac{de(\theta)}{d\theta} = -\gamma e(\theta), \quad \gamma > 0,$$

gdzie γ to współczynnik tłumienia. Po wstawieniu błędu do powyższego równania, otrzymujemy równanie Ważewskiego-Dawidenki

$$J_{x_0, T}(u_\theta(\cdot)) \frac{du_\theta(\cdot)}{d\theta} = -\gamma e(\theta), \quad (4.10)$$

w którego rozwiązaniu przychodzi nam z pomocą umiejętność wyznaczenia analitycznego Jakobianu. Dla systemu (4.9) jakobian analityczny obliczamy w następujący sposób [239]

$$J_{x_0, T}(u(\cdot))v(\cdot) = C(T) \int_0^T \Phi(T, t) B(t) v(t) dt$$

$$A(t) = \frac{\partial(f(x(t)) + g(x(t))u(t))}{\partial x}, \quad B(t) = g(x(t)), \quad C(t) = \frac{\partial k(x(t))}{\partial x},$$

gdzie $\Phi(T, t)$ oznacza macierz fundamentalną uzyskaną z równania $\frac{\partial \Phi(t, s)}{\partial t} = A(t)\Phi(t, s)$, $\Phi(s, s) = I_n$. Równanie (4.10) przekształcamy w celu wyliczenia $\frac{du_\theta(\cdot)}{d\theta}$ i uzyskujemy

$$\frac{du_\theta(\cdot)}{d\theta} = -\gamma J_{x_0, T}^\#(u_\theta(\cdot))(K_{x_0, T}(u_\theta(\cdot)) - y_d). \quad (4.11)$$

$J_{x_0, T}^\#(u(\cdot))$ oznacza pewną prawostronną odwrotność Jakobianu, na przykład z odwrotności Moore'a i Penrose'a

$$J_{x_0, T}^\#(u(\cdot)) = J_{x_0, T}^*(u(\cdot))(J_{x_0, T}(u(\cdot))J_{x_0, T}^*(u(\cdot)))^{-1},$$

gdzie

$$J_{x_0, T}^*(u(\cdot)) : \mathcal{R}^r \rightarrow \mathcal{X}, \quad \left(J_{x_0, T}^*(u(\cdot))\eta \right) (t) = B^T(t)\Phi(T, t)C^T(t)\eta, \quad \eta \in \mathcal{R}^r.$$

Według metody endogenicznej przestrzeni konfiguracyjnej, jako rozwiązanie zadania planowania należy przyjąć granicę

$$u_d(t) = \lim_{\theta \rightarrow +\infty} u_\theta(t),$$

dla której uzyskuje się rozwiązanie $y(T) = y_d$ układu (4.9).

Z praktycznego punktu widzenia, implementując przedstawiony powyżej algorytm, łatwiej jest operować na skończone-wymiarowej reprezentacji endogenicznej przestrzeni konfiguracyjnej. W tym celu można przekształcić algorytmu z postaci ciągłej na jej dyskretny odpowiednik. Należy przyjąć pewną ortogonalną bazę $L_m^2[0, T]$, z funkcjami bazowymi $\varphi_1(t), \varphi_2(t), \dots$, a następnie sterowania $u(t)$ wyrazić w postaci skończonego szeregu funkcji bazowych z odpowiednimi współczynnikami wagowymi. Dzięki tej operacji uzyskamy reprezentację endogenicznej przestrzeni konfiguracyjnej z ograniczonym pasmem

$$u(t) \rightarrow P(t)\lambda,$$

gdzie $\lambda \in \mathcal{R}^s$ jest wektorem zawierającym współczynniki szeregu, $s = \sum_{i=1}^m s_i$, s_i jest długością rozwinięcia i -tej składowej sterowania, a $P(t)$ jest macierzą

$$P(t) = \begin{bmatrix} \varphi_1(t) & \varphi_2(t) & \cdots & \varphi_{s_i}(t) & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & & \vdots & \vdots & \vdots & & \vdots \\ 0 & 0 & \cdots & 0 & \varphi_1(t) & \varphi_2(t) & \cdots & \varphi_{s_m}(t) \end{bmatrix}$$

Ustalając więc s -wymiarową parametryzację przestrzeni sterowań otrzymujemy skończone wymiarową endogeniczną przestrzeń konfiguracyjną. Ostatecznie, rozwiązanie zadania otrzymujemy za pomocą algorytmu typu Jakobianu pseudoodwrotnego z ograniczonym pasmem, w postaci dyskretnej, określonego równaniem

$$\lambda_{\theta+1} = \lambda_\theta - \gamma \tilde{J}_{x_0, T}^\#(\lambda_\theta) e(\lambda_\theta),$$

gdzie

$$\begin{aligned} \tilde{J}_{x_0, T}^\#(\lambda_\theta) &= \tilde{J}_{x_0, T}^T(\lambda_\theta) (\tilde{J}_{x_0, T}(\lambda_\theta) \tilde{J}_{x_0, T}^T(\lambda_\theta))^{-1}, \\ \tilde{J}_{x_0, T}(\lambda_\theta) &= C_{\lambda_\theta}(T) \int_0^T \Phi_{\lambda_\theta}(T, t) B_{\lambda_\theta}(t) P(t) dt \end{aligned}$$

W badaniach posługiwano się dwoma rodzajami baz. Pierwszy zestaw funkcji, to baza Fouriera

$$u_i(t) = \lambda_{i0} + \sum_{j=1}^{\frac{s_i-1}{2}} \left(\lambda_{i2j-1} \sin \frac{2\pi jt}{T} + \lambda_{i2j} \cos \frac{2\pi jt}{T} \right),$$

druga baza to zestaw funkcji Laguerre'a

$$u_i(t) = \lambda_{i0} + \sum_{j=1}^{s_i-1} \left(\lambda_{ij} \frac{e^t}{j!} \frac{d^j}{dt^j} (e^{-t} t^j) \right).$$

W dalszej części pracy układ (4.3), z którego wyodrębniono 7-wymiarowy podsystem $x = (x, y, \theta, \eta_1, \eta_2, \eta_3, \alpha) \in \mathcal{R}^7$ rozważany będzie jako

$$\begin{cases} \dot{x} &= f(x) + g(x)u \\ y &= k(x), \end{cases} \quad (4.12)$$

4.4.1 Sterowanie z ograniczeniami

Podstawowy algorytm planowania ruchu znajduje rozwiązanie na określonym horyzoncie doprowadza wyjście układu do punktu zadanego. Jednakże nie pozwala na kontrolę stanu w trakcie działania wyliczonych sterowań. W przypadku robota balansującego, celem jest nie tylko doprowadzenie platformy do punktu zadanego, ale zadbanie o to, aby korpus znajdował się w okolicy pionu. Uwzględnienie tego warunku uzyskuje się przez wprowadzenie do układu ograniczeń. Problemem sterowania z ograniczeniami zajmowano się w pracy [119]. Rozważano włączenie do systemu zarówno ograniczeń dla niektórych zmiennych, jak i dla sterowań.

Przyjmijmy, że dla robota balansującego chcemy ograniczyć wychylenie wahadła

$$\alpha_{min} \leq \alpha(t) \leq \alpha_{max}.$$

Powyższą nierówność należy przedstawić w postaci funkcji kary $(x)_+ = \max\{x, 0\}$, która będzie się uaktywniać tylko przy przekroczeniu wartości granicznych. W pracy [43] pokazano, że tego typu funkcję można aproksymować funkcją gładką.

$$(x)_+ \cong p(x, \mu) = x + \frac{1}{\mu} \ln(1 + e^{-\mu x}),$$

gdzie $\mu > 0$ jest parametrem. Stosując zaproponowane przybliżenie funkcji kary, można rozbudować układ 4.12 o dodatkową zmienną stanu oraz dodatkowe wyjście. Rozszerzony układ ma postać

$$\begin{cases} \dot{x} &= f(x) + g(x)u \\ \dot{x}_{n+1} &= p(\alpha - \alpha_{max}, \mu) + p(-\alpha + \alpha_{min}, \mu) \\ y &= k(x) \\ y_{r+1} &= x_{n+1}. \end{cases} \quad (4.13)$$

Zadanie planowania ruchu pozostaje bez zmian. Należy znaleźć taką funkcję sterowań $u(\cdot)$, by na określonym horyzoncie T osiągnąć położenie zadane $z_d = (y_d, 0)$. Niestety, zadanie to nie może być rozwiązane przy pomocy standardowego algorytmu jakobianowego z powodu osobliwości jakobianu. W pracy [119] zaproponowano, aby funkcję kary dla systemu (4.13) zmodyfikować dodając element regularyzujący, np. α^2 . Ostatecznie, uzyskany układ zregularyzowany ma postać

$$\begin{cases} \dot{x} &= f(x) + g(x)u \\ \dot{x}_{n+1} &= p(\alpha - \alpha_{max}, \mu) + p(-\alpha + \alpha_{min}, \mu) + \alpha^2 \\ y &= k(x) \\ y_{r+1} &= x_{n+1}. \end{cases} \quad (4.14)$$

Przebieg algorytmu pozostaje bez zmian, z wyjątkiem wyliczania jakobianu. Jakobian wyznaczamy na podstawie układu zregularyzowanego (4.14), natomiast błąd wyznaczamy z układu rozszerzonego (4.13). Taki algorytm planowania ruchu jest typu Jakobianu zaburzonego.

Symulacje układu przedstawiono na rysunku 4.9. Pierwsze cztery wykresy (a, b, c, d) prezentują przypadek, w którym zadano położenie platformy $x = 3$, $y = 3$ na horyzoncie czasowym $T = 4s$. Kolejne cztery rysunki (e, f, g, h) przedstawiają wynik symulacji, w którym punktem docelowym na horyzoncie czasowym $T = 2s$ było położenie platformy $x = 2$, $y = 2$ oraz kąt wahadła $\alpha = 0$. W obydwu przypadkach jako stan początkowy przyjęto $x_0 = (0, 0, 0, 0, 0, 0, 0, 0)$. Ponadto, jako ograniczenie kąta wychylenia wahadła zadano

$-0.4 \leq \alpha(t) \leq 0.4$. Przestrzeń sterowań wybrano typu Fouriera z pobudzeniem początkowym $\lambda_1 = (0.01, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0)$, $\lambda_2 = (0.011, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0)$, współczynnik $\gamma = 0.1$.

Zaobserwowano, że w pierwszym przypadku, pod koniec trwania symulacji wahadło gwałtownie opada. Algorytm utrzymał kąt korpusu w zadanym ograniczeniu, jednakże można przypuszczać, że tuż po upływie czasu T robot się przewróci. Drugi przypadek pokazuje, że żądanie osiągnięcia zerowego położenia wahadła zdecydowanie poprawia sytuację. Okazuje się, że w tym przypadku ograniczenie nie zostanie naruszone, gdyż żądanie osiągnięcia kąta $\alpha = 0$ powoduje, że układ jest daleki od ograniczeń. Przy dłuższych czasach symulacji wystąpiły problemy ze zbieżnością algorytmu.

4.4.2 Sterowanie z priorytetami

Kolejnym wariantem sterowania metodą endogenicznej przestrzeni konfiguracyjnej jest wprowadzenie priorytetów rozwiązywanych zadań. Implementację oraz wyjaśnienie modyfikacji standardowego algorytmu zaprezentowano w pracy [208]. Jak już wspomniano, w przypadku robota balansującego, celem jest nie tylko doprowadzenie platformy do punktu zadanego, ale zadbanie o to, aby korpus znajdował się w okolicy pionu. W związku z powyższym, zaproponowano dwa zadania, z których pierwsze doprowadza robota do zadanego położenia na płaszczyźnie, a drugie minimalizuje kwadrat odchylenia korpusu od pionu

$$\begin{aligned} S_1 : & \quad x(0) \xrightarrow{u([0,T])} x_T, \\ S_2 : & \quad \int_0^T \alpha^2(t) dt \xrightarrow{u([0,T])} \min. \end{aligned}$$

Zadanie z niższym priorytetem będzie realizowane tylko wtedy, gdy zadanie z wyższym priorytetem zostanie rozwiązane. Całka z drugiego zadania powinna mieć jak najmniejszą wartość, bez wpływu na zadanie główne. Aby zadanie S_2 mogło być realizowane, należy podobnie jak w poprzednim przykładzie, rozbudować system 4.9 o dodatkową zmienną stanu oraz dodatkowe wyjście. Rozszerzony system ma postać

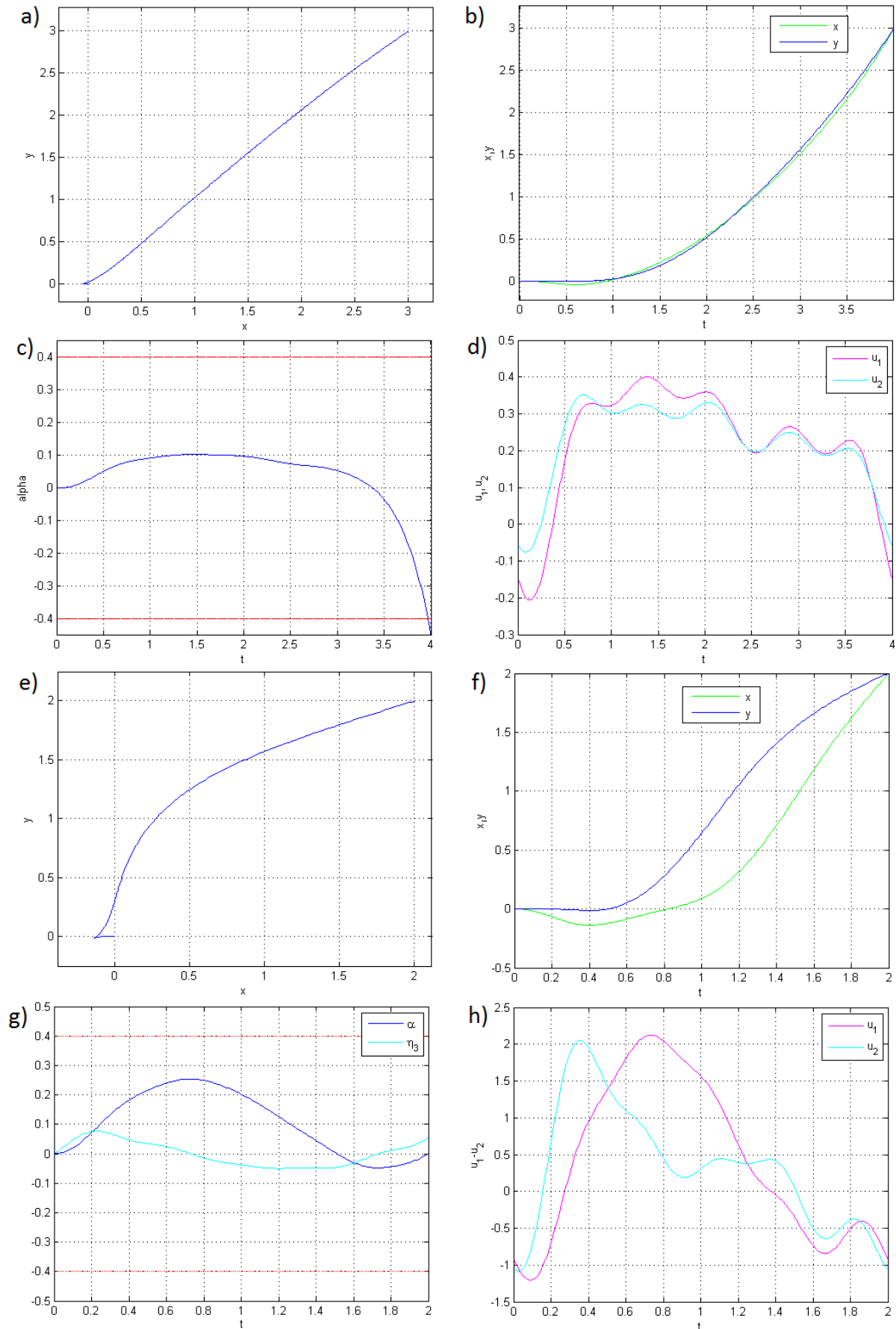
$$\begin{cases} \dot{x} &= f(x) + g(x)u \\ \dot{x}_{n+1} &= \alpha^2 \\ y &= k(x) \\ y_{r+1} &= x_{n+1} \end{cases} \quad (4.15)$$

Implementacja algorytmu przebiega podobnie jak w przypadku standardowej wersji, jednakże obliczeń jacobianu i błędów dokonujemy dla każdego zadania osobno. Dla skończonej wymiarowej reprezentacji endogenicznej przestrzeni konfiguracyjnej $u_\lambda(t) = P(t)\lambda$ mamy kolejno dla głównego zadania

$$\begin{aligned} \tilde{J}_{1,T}(\lambda) &= C_1 \int_0^T \Phi_\lambda(T, t) B_\lambda(t) P_\lambda(t) dt, \\ \tilde{e}_1(\lambda) &= y(T) - y_d \end{aligned}$$

oraz dla zadania o niższym priorytecie

$$\begin{aligned} \tilde{J}_{2,T}(\lambda) &= \int_0^T \int_0^t \alpha_\lambda(t) e_7^T \Phi_\lambda(t, s) B_\lambda(s) P_\lambda(s) ds dt, \\ \tilde{e}_2(\lambda) &= x_{r+1}(T), \end{aligned}$$



Rysunek 4.9: Wyniki symulacji algorytmów planowania ruchu metodą endogeniczną przestrzeni konfiguracyjnej z ograniczeniami

gdzie C_1 to macierz wyjścia wybierająca elementy wektora stanu. Dla powyższych obliczamy pseudoodwrotności $\tilde{J}_{1,T}^\#(\lambda)$ oraz $\tilde{J}_{2,T}^\#(\lambda)$.

Ostatecznie, można zapisać dyskretną formę algorytmu z podziałem na dwa zadania, z zachowaniem ich priorytetów

$$\lambda_{\theta+1} = \lambda_\theta - \gamma_1 \tilde{J}_{1,T}^\#(\lambda_\theta) \tilde{e}_1(\lambda_\theta) - \gamma_2 \tilde{D}_{1,T}(\lambda_\theta) \tilde{J}_{2,T}^\#(\lambda_\theta) \tilde{e}_2(\lambda_\theta),$$

gdzie $\tilde{D}_{1,T} = I - \tilde{J}_{1,T}^\# \tilde{J}_{1,T}$ to skończenie wymiarowa projekcja na $\ker \tilde{J}_{1,T}$. Dla każdego podzadania dobiera się współczynnik tłumienia błędu γ .

Symulacje algorytmu przedstawiono na rysunku 4.10. Pierwsze cztery wykresy (a, b, c, d) prezentują przypadek, w którym zadano położenie platformy $x = 3$, $y = 3$, $\alpha = 0$ na horyzoncie czasowym $T = 3s$. Kolejne cztery rysunki (e, f, g, h) przedstawiają wynik symulacji, w którym punktem docelowym, na horyzoncie czasowym, był cały wektor stanu. Zażądano, aby robot nie tylko dojechał do pozycji zadanej, lecz także zatrzymał się w niej. Punkt $x_d = (2, 2, \frac{\pi}{2}, 0, 0, 0, 0, 0, 0, 0)$ miał być osiągnięty na horyzoncie czasowym $T = 2s$. W obydwu przypadkach jako stan początkowy przyjęto $x_0 = (0, 0, 0, 0, 0, 0, 0, 0, 0, 0)$. Przestrzeń sterowań wybrano typu Fouriera z pobudzeniem początkowym $\lambda_1 = (0.01, 0, 0, 0, 0, 0, 0, 0, 0, 0)$, $\lambda_2 = (0.011, 0, 0, 0, 0, 0, 0, 0, 0, 0)$ oraz współczynnikami $\gamma_1 = 0.1$ i $\gamma_2 = 10$.

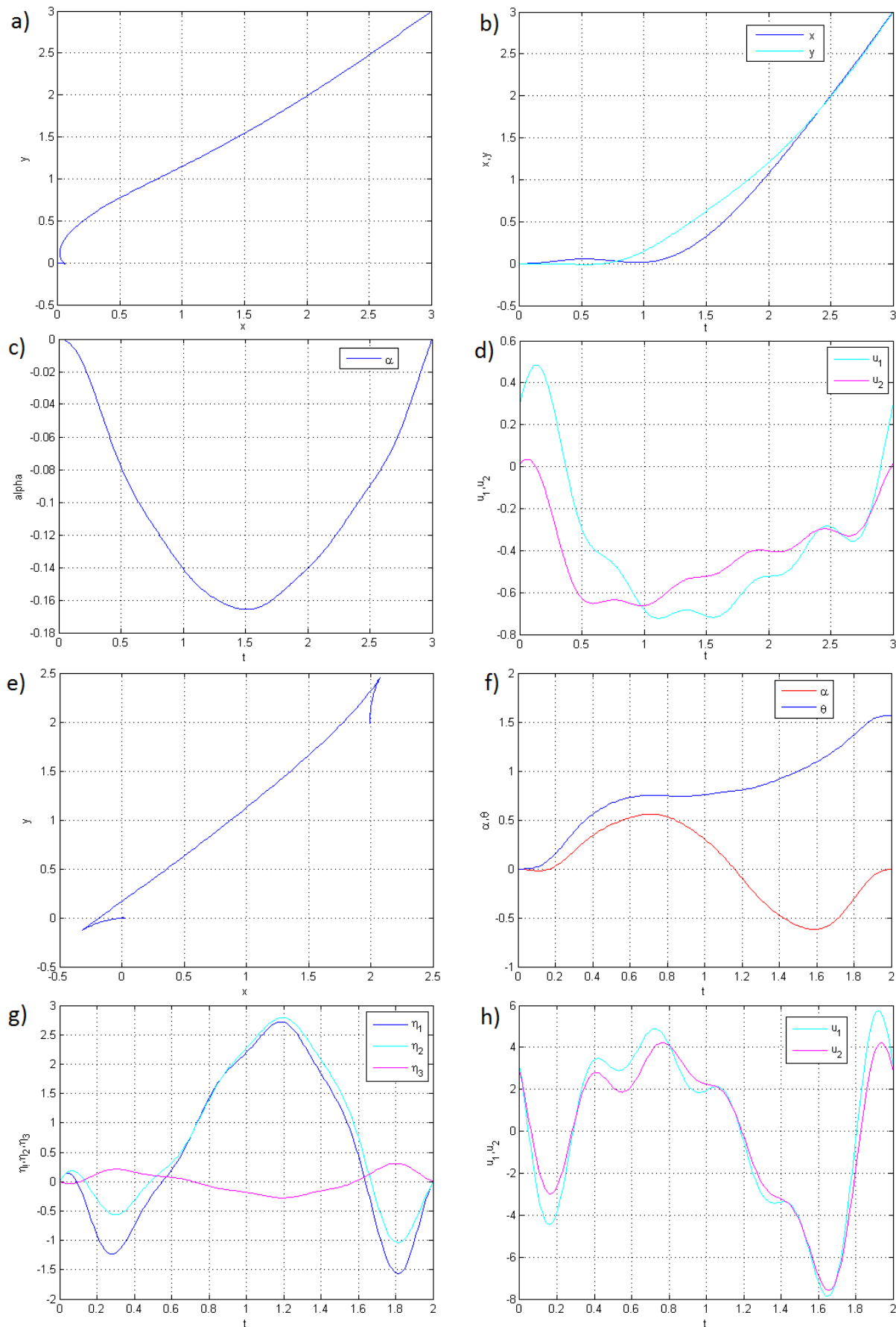
Zaobserwowano, że zgodnie z przyjętym celem, w obydwu przypadkach algorytm utrzymał korpus robota w pionie, doprowadzając platformę do punku zadanego. Oznacza to, że można robotem dojechać do zadanego punktu z określonymi prędkościami lub nawet go w nim zatrzymać. Podobnie jak w poprzednim podrozdziale, wystąpiły problemy ze zbieżnością algorytmu przy dłuższych horyzontach sterowania.

4.4.3 Sterowanie z podziałem na kroki

W obydwu poprzednich podejściach, ze względu na lokalny charakter algorytmu oraz złożony model dynamiki obiektu, nie było możliwe znalezienie rozwiązań dla horyzontów czasowych dłuższych niż 2-3s. Zaobserwowano jednak, że istnieje możliwość zażądania, aby algorytm wyznaczył takie sterowanie, aby osiągnąć cały zadany wektor stanu $y = x$. Oznacza to, że dowolnie długi horyzont czasowy można podzielić na krótkie odcinki czasowe, dla których algorytm będzie działał prawidłowo. Należy przy tym zadbać o to, aby między poszczególnymi krokami była zachowana ciągłość, zarówno dla stanu jak i sterowań. Pierwszy warunek jest łatwy do spełnienia, stan końcowy dla każdego kroku należy ustawić jako początkowy dla kolejnego. W celu uniknięcia skoków sterowań, które mogą być trudne do realizacji przez rzeczywisty napęd, rozbudowano układ w taki sposób, aby sterowania były zadawane w formie pierwszej pochodnej. Dotychczasowe wejście włączono do wektora stanu i wprowadzono nową wartość sterującą s . Modyfikacja ta dotyczy układu 4.15 i algorytmu z priorytetami.

$$\begin{cases} \dot{x} &= f(x) + g(x)u \\ \dot{u}_1 &= s_1 \\ \dot{u}_2 &= s_2 \\ y &= x \end{cases} \implies \begin{cases} \dot{z} &= F(z) + G(z)s \\ y &= k(z) \end{cases}, \quad z = (x, u)^T.$$

Ważne znaczenie ma sposób automatycznego doboru punktów pośrednich. Algorytm zaprezentowany w podrozdziale 4.4.2 pozwala na znajdowanie rozwiązania dla całego wektora stanu. Dzięki temu, można tak dobrać punkty styku poszczególnych kroków, aby w każdym z nich platforma nie zmieniała swojej prędkości, a dopiero po przejściu przez



Rysunek 4.10: Wyniki symulacji algorytmów planowania ruchu metodą endogenicznej przestrzeni konfiguracyjnej z priorytetami

wszystkie punkty pośrednie, zatrzymywała się na pozycji zadanej. Dla realizacji tej idei zaproponowano prostą funkcję, która wylicza punkty pośrednie jedynie na podstawie zadanej prędkości liniowej platformy i punktu docelowego. Naturalnie, czas trwania jednego kroku nie może być dłuższy niż $T = 2s$.

Na początek, na podstawie zadanej prędkości V_{lin} i punktu docelowego $[x_d, y_d, \theta_d]$ wyliczana jest droga oraz czas potrzebny do jej przebycia

$$d = \sqrt{(x_d - x_0)^2 + (y_d - y_0)^2}, \quad t = \frac{d}{V_{lin}}.$$

Następnie, wyliczany jest kierunek (orientacja) θ_{pos} dla punktów pośrednich. Zakłada się, że platforma jedzie możliwie najkrótszą drogą do punktu docelowego

$$\theta_{pos} = atan2(x_d - x_0, y_d - y_0).$$

Wyliczany jest także czas trwania jednego kroku Δt i liczba kroków n zaokrąglona w górę

$$\Delta t = \frac{t}{n}, \quad n = \lceil \Delta t \rceil.$$

Ostatecznie, punkty startowy, pośrednie oraz końcowy są wyliczane wg formuły

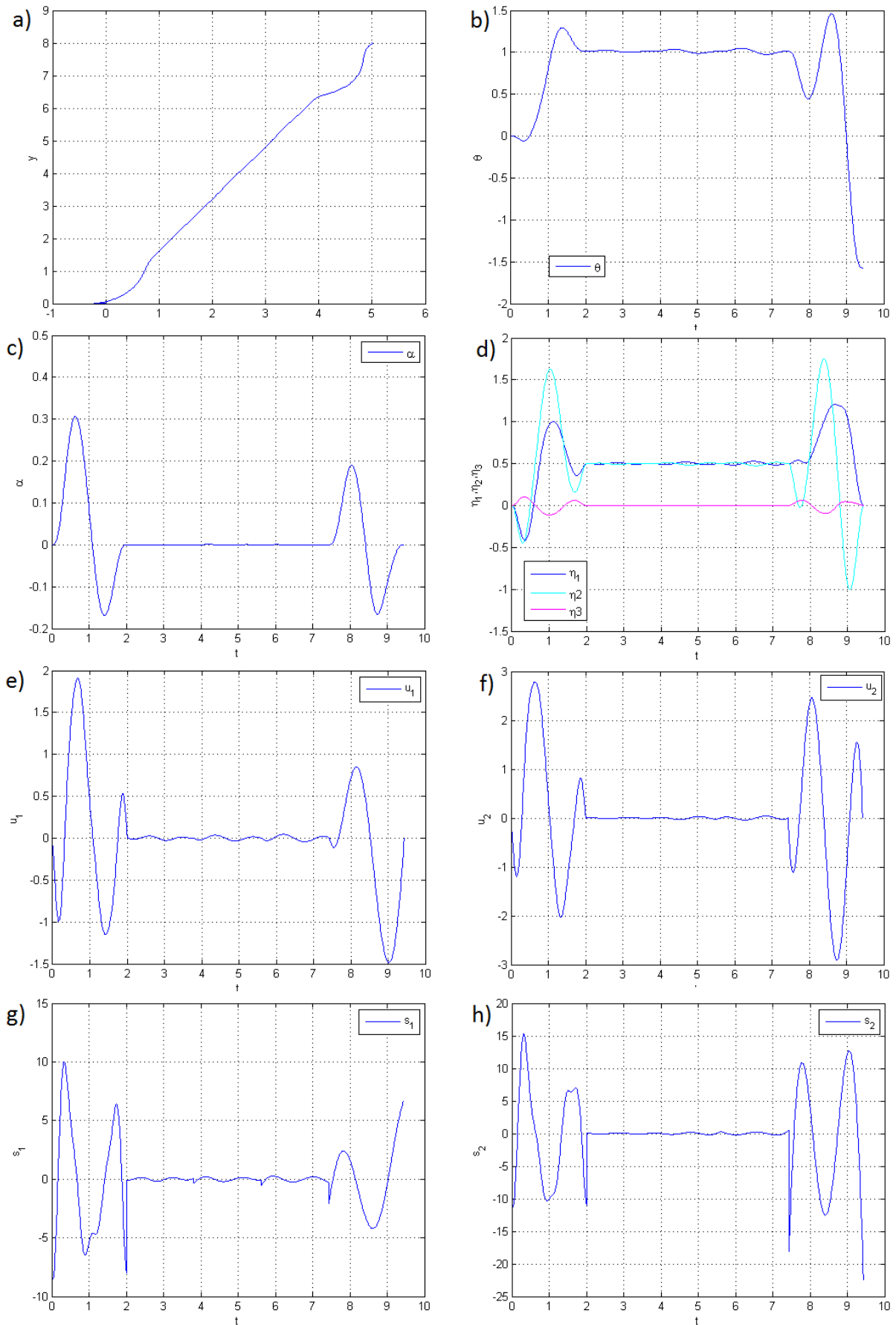
$$\begin{aligned} z_0 &= [x_0, y_0, \theta_0, 0, 0, 0, 0, 0, 0, 0], \\ \forall_{i=1}^{n-1} z_i &= \left[x_0 + \frac{i\Delta T}{t}(x_d - x_0), y_0 + \frac{i\Delta T}{t}(y_d - y_0), \theta_{pos}, \frac{V_{lin}}{2}, \frac{V_{lin}}{2}, 0, 0, 0, 0, 0 \right], \\ z_n &= [x_d, y_d, \theta_d, 0, 0, 0, 0, 0, 0, 0]. \end{aligned}$$

Symulacje układu przedstawiono na rysunku 4.11. Platforma z punktu $z_0 = (0, 0, 0, 0, 0, 0, 0, 0, 0, 0)$ dojechała do $z_d = (5, 8, -\frac{\pi}{2}, 0, 0, 0, 0, 0, 0, 0)$, z prędkością $1m/s$. Dla $T = 2s$ otrzymano następujące punkty pośrednie

$$\begin{aligned} t &= 9,43s \\ n &= 5 \\ \Delta T &= 1,81s \\ z_0 &= [0, 0, 0, 0, 0, 0, 0, 0, 0, 0] \\ z_1 &= [0.96, 1.54, 1.01, 0.50, 0.50, 0, 0, 0, 0, 0] \\ z_2 &= [1.92, 3.07, 1.01, 0.50, 0.50, 0, 0, 0, 0, 0] \\ z_3 &= [2.88, 4.61, 1.01, 0.50, 0.50, 0, 0, 0, 0, 0] \\ z_4 &= [3.84, 6.14, 1.01, 0.50, 0.50, 0, 0, 0, 0, 0] \\ z_5 &= [5.00, 8.00, -1.57, 0, 0, 0, 0, 0, 0, 0] \end{aligned}$$

Przestrzeń sterowań wybrano typu Fouriera, z pobudzeniem początkowym $\lambda_1 = (0.1, 0, 0, 0, 0, 0, 0, 0, 0, 0)$, $\lambda_2 = (0.11, 0, 0, 0, 0, 0, 0, 0, 0, 0)$ oraz współczynnikami $\gamma_1 = 0.1$ oraz $\gamma_2 = 10$.

Zaobserwowano, że zgodnie z przyjętym celem, algorytm utrzymał korpus robota w pionie, doprowadzając platformę do punktu zadanego. Zastosowanie podziału na kroki, rozwiązało problem sterowania robotem na dłuższym horyzoncie. Planując ruch robota, zgodnie z przygotowaną mapą pomieszczenia, można wyznaczać punkty w dogodny sposób, uwzględniając przy tym, np. naniesione przeszkody. Krokowa praca pozwala także na lepszą kontrolę stanu, między punktem startowym a docelowym.



Rysunek 4.11: Wyniki symulacji algorytmów planowania ruchu metodą endogeniczną z podziałem na kroki

4.5 Sterowanie nieliniowe – zadanie sterowania optymalnego

4.5.1 Planowanie ruchu w otwartej pętli

Problem planowania ruchu dla systemu

$$\dot{x} = f(x) + g(x)u \quad (4.16)$$

może być również sformułowany jako problem optymalnego sterowania OCP (*Optimal Control Problem*), tzn. w czasie T znaleźć takie $u(\cdot) \in \mathcal{X}$, aby zminimalizować funkcję

$$J(u(\cdot)) = \int_0^T L(x(t), u(t), T) dt + M(x(T), T),$$

w której L to funkcja Lagrange'a, natomiast M to funkcja Mayera, które w szczególnym przypadku mogą mieć postać

$$\begin{aligned} L(t) &= x^T(t)Qx(t) + u(t)^T Ru(t), \\ M(T) &= x^T(T)Px(T), \end{aligned}$$

gdzie $Q = Q^T \geq 0$, $R = R^T \geq 0$ i $P = P^T \geq 0$ są macierzami wzmocnień. Optymalnego sterowania szukano przy pomocy narzędzi ACADO Toolkit [108, 107, 7]. Oprogramowanie to pozwala także na szukanie rozwiązań dla układów z ograniczeniami. W przypadku robota balansującego, ograniczenie wychylenia korpusu jest niezbędne.

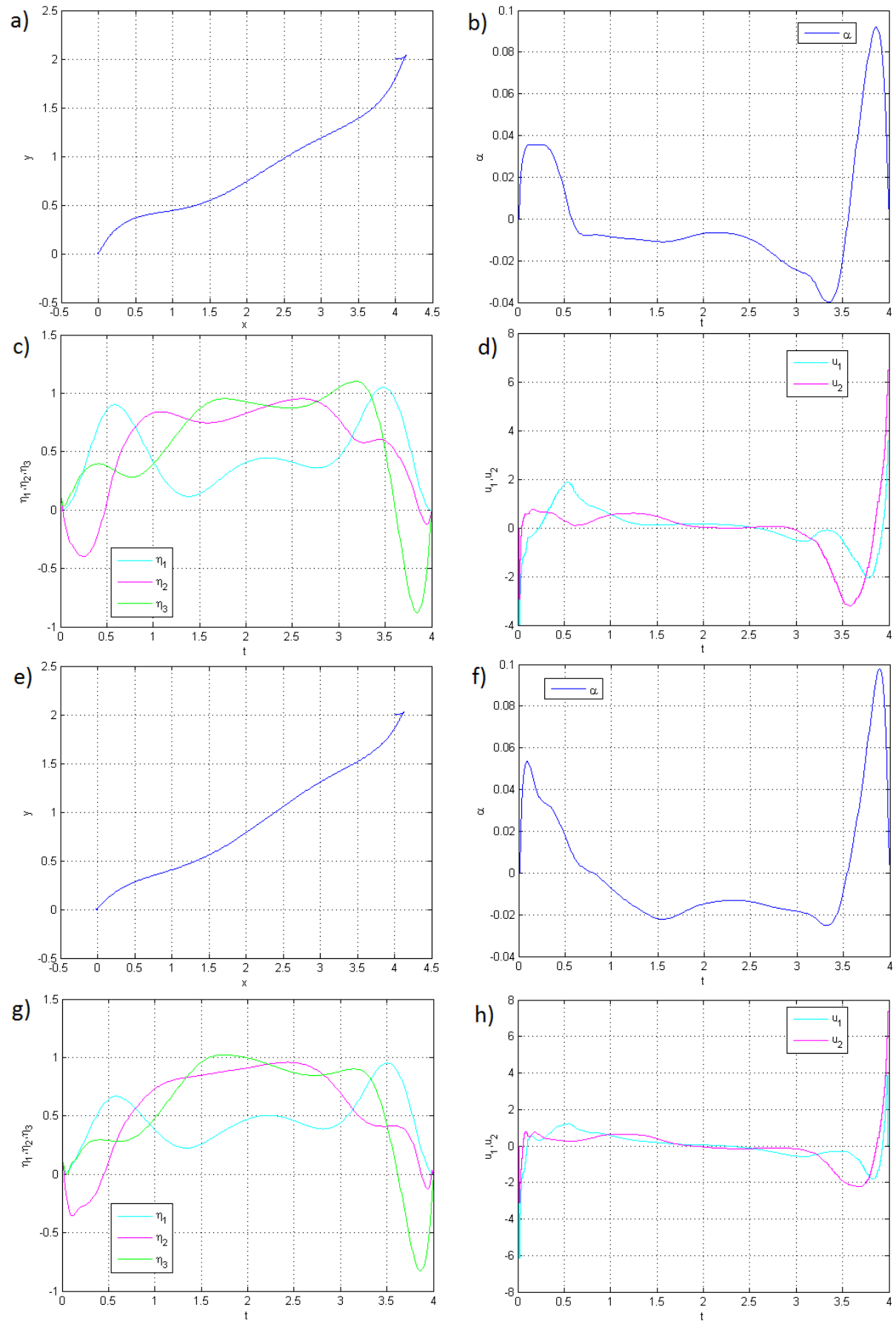
Podobnie jak dla metody przestrzeni endogenicznej przyjmijmy układ (4.3), z którego wyodrębniono 7-wymiarowy podsystem $x = (x, y, \theta, \eta_1, \eta_2, \eta_3, \alpha) \in \mathcal{R}^7$ i opisano w postaci (4.16).

Pierwszej symulacji dla tego układu dokonano minimalizując jedynie funkcję Mayera, tzn. żądano aby układ został jedynie doprowadzony w pobliże punktu zadanego, bez względu na długość drogi, jaką pokona oraz ile zużyje przy tym energii. Na rysunku 4.12 pierwsze cztery wykresy (a,b,c,d) przedstawiają wyniki symulacji dla następujących parametrów $x_0 = (0, 0, 0, 0.1, 0.1, 0, 0)$ oraz $x_d = (4, 2, 0, 0, 0, 0, 0)$. Na wychylenie korpusu nałożono ograniczenie $-0.5 \leq \alpha \leq 0.5$. Przyjęto horyzont czasowy $T = 4s$. Na potrzeby symulacji konieczne było nadanie niezerowych prędkości początkowych kół. Ponieważ ACADO rozwiązuje zadanie w sposób dyskretny, przyjęto stały krok symulacji równy 0.01s.

Kolejną symulację przeprowadzono minimalizując wyrażenie Lagrange'a, tzn. szukano takich sterowań, aby doprowadzić robota w pobliże punktu zadanego, przy czym jednocześnie minimalizując w czasie pokonywania drogi użytą energię oraz wychylenia wahadła wg

$$L(x(t), u(t)) = \alpha(t)^2 + u(t)^T u(t).$$

Wyniki symulacji, dla identycznych warunków początkowych i końcowych ograniczeń oraz parametrów czasowych, zamieszczono na wykresach (e,f,g,h). Porównując wykresy 4.12 d) i 4.12 h) zaobserwowano, że zadziałało kryterium minimalizacji zużycia energii, co wskazuje na poprawną realizację założeń.



Rysunek 4.12: Wyniki symulacji algorytmów optymalnego sterowania

4.5.2 Planowanie ruchu w zamkniętej pętli

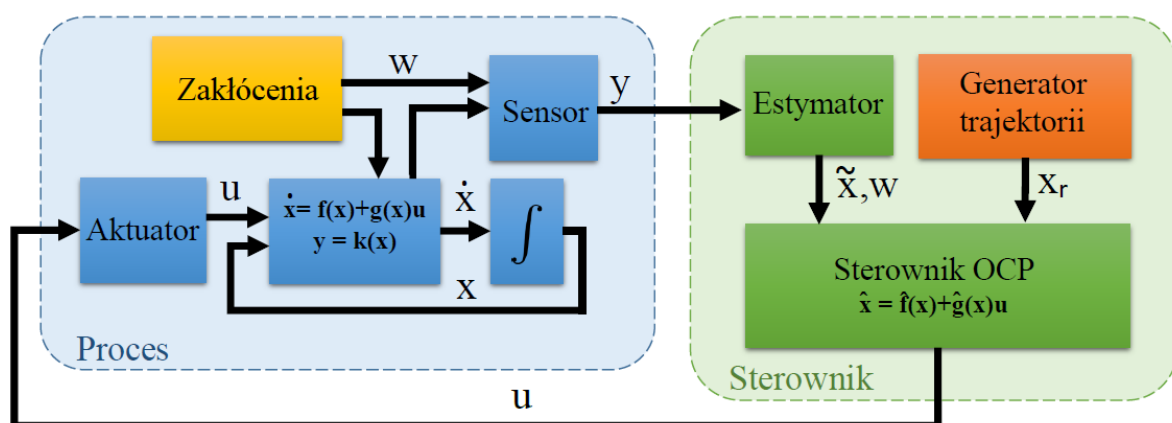
Wszystkie do tej pory omówione przykłady planowania ruchu działają w pętli otwartej. Może się okazać, że nie posiadamy dobrze zidentyfikowanego modelu matematycznego robota i wyliczone optymalne sterowanie nie spełni swojej roli. W przypadku tradycyjnych (statycznych) układów jezdnych skutkiem niedokładnie wyznaczonego sterowania, będzie zwiększenie się błędu osiągnięcia celu. W przypadku robota balansującego, może to doprowadzić do przewrócenia się robota i jego uszkodzenia.

Skutecznym remedium może okazać się zastosowanie algorytmu sterowania predykcyjnego MCP (*Model Predictive Control*) [209, 38, 160], który znakomicie sprawdza się w praktyce [238]. Algorytmy predykcyjne są przykładem zaawansowanych metod sterowania. Ich działanie opiera się na wykorzystaniu wiedzy o przyszłym zachowaniu się wielkości regulowanej, w celu wyznaczenia optymalnych wartości sterujących. Przewidując przyszłe wartości regulowane, korzysta się z modelu matematycznego sterowanego obiektu, wcześniejszych wartości sygnału sterującego oraz wartości sygnałów zakłócających.

Celem sterowania predykcyjnego jest zapewnienie, by trajektoria stanu $x(t)$ układu $x = (x, y, \theta, \eta_1, \eta_2, \eta_3, \alpha) \in \mathcal{R}^7$ w postaci (4.16) była jak najbardziej zbliżona do sygnału odniesienia $x_r(t)$. Podobnie, jak w przypadku otwartej pętli minimalizuje się kwadratowy wskaźnik jakości

$$J(u(\cdot)) = \int_0^T ((x(t) - x_r(t))^T Q (x(t) - x_r(t)) + (u(t) - u_r(t))^T R (u(t) - u_r(t))) dt + (x(T) - x_r(T))^T P (x(T) - x_r(T)),$$

gdzie $Q = Q^T \geq 0$, $R = R^T \geq 0$ i $P = P^T \geq 0$ są macierzami wzmocnień, a u_r to sterowanie dla którego uzyskujemy trajektorię odniesienia $\dot{x}_r = f(x_r) + g(x_r)u_r$. Działanie algorytmu można podzielić na trzy fazy. Pierwsza z nich to faza pomiaru lub estymacji stanu. W kolejnej, dla skończonego czasu T_p , zwanego czasem predykcji, rozwiązuje się zadanie optymalnego sterowania (OCP), na podstawie aktualnego stanu oraz trajektorii odniesienia. Ostatnia faza aplikuje wyznaczone sygnały sterujące na krótkim odcinku czasowym T_s , zwanym horyzontem sterowania; musi być spełniony warunek $T_s < T_p$. Schemat blokowy układu sterowania został przedstawiony na rysunku 4.13.



Rysunek 4.13: Struktura układu sterowania metodą predykcyjną

Przy doborze horyzontu predykcji i sterowania należy przyjąć pewien kompromis. W przypadku nieznanego modelu powinno się skracać horyzont sterowania. Niestety, dzieje się to kosztem utraty zysków z optymalnego sterowania. Jeśli posiadamy bardzo

dokładny model, a na układ regulacji działają małe zakłócenia, to można również stosować pełny wyznaczony ciąg sterowań, a uaktualnianie parametrów modelu i kolejne wyznaczenie ciągu przyszłych wartości sygnału sterującego przeprowadzić co horyzont predykcji.

Badania symulacyjne przeprowadzono dla następujących parametrów: stan początkowy $x_0 = (0, 0, 0, 0, 0, 0, 0)$. Na wychylenie korpusu nałożono ograniczenie $-0.5 \leq \alpha \leq 0.5$. Jako trajektorię odniesienia zaproponowano kwadrat o boku $4m$ realizowany w czasie $28s$. Horyzont predykcji ustawiono na $4s$ z podziałem czasowym $0.05s$. Horyzont sterowania wynosi dokładnie tyle, ile jeden krok $0.05s$. Podsumowując, algorytm optymalizacji szuka rozwiązania dla 80 kroków, po czym pierwszy z nich zadaje jako sterowanie na obiekt rzeczywisty. Wyniki symulacji zaprezentowano na rysunku 4.14 (a,b,c,d).

W celu sprawdzenia odporności sterownika na nieznaną model dynamiki robota, wykonano kolejną serię symulacji, wprowadzając zaburzenie do obiektu sterowanego. Zmiana polegała na zwiększeniu masy osi z $10kg$ na $15kg$ oraz zmiany masy wahadła z $5kg$ na $20kg$. Pozostałe parametry pozostają identyczne jak w pierwszej symulacji. Jest to bardzo duża zmiana własności fizycznych obiektu. W takiej sytuacji, żaden sterownik pracujący w otwartej pętli nie utrzyma robota w pozycji pionowej. Wyniki symulacji znajdują się na rysunku 4.14 (e,f,g,h).

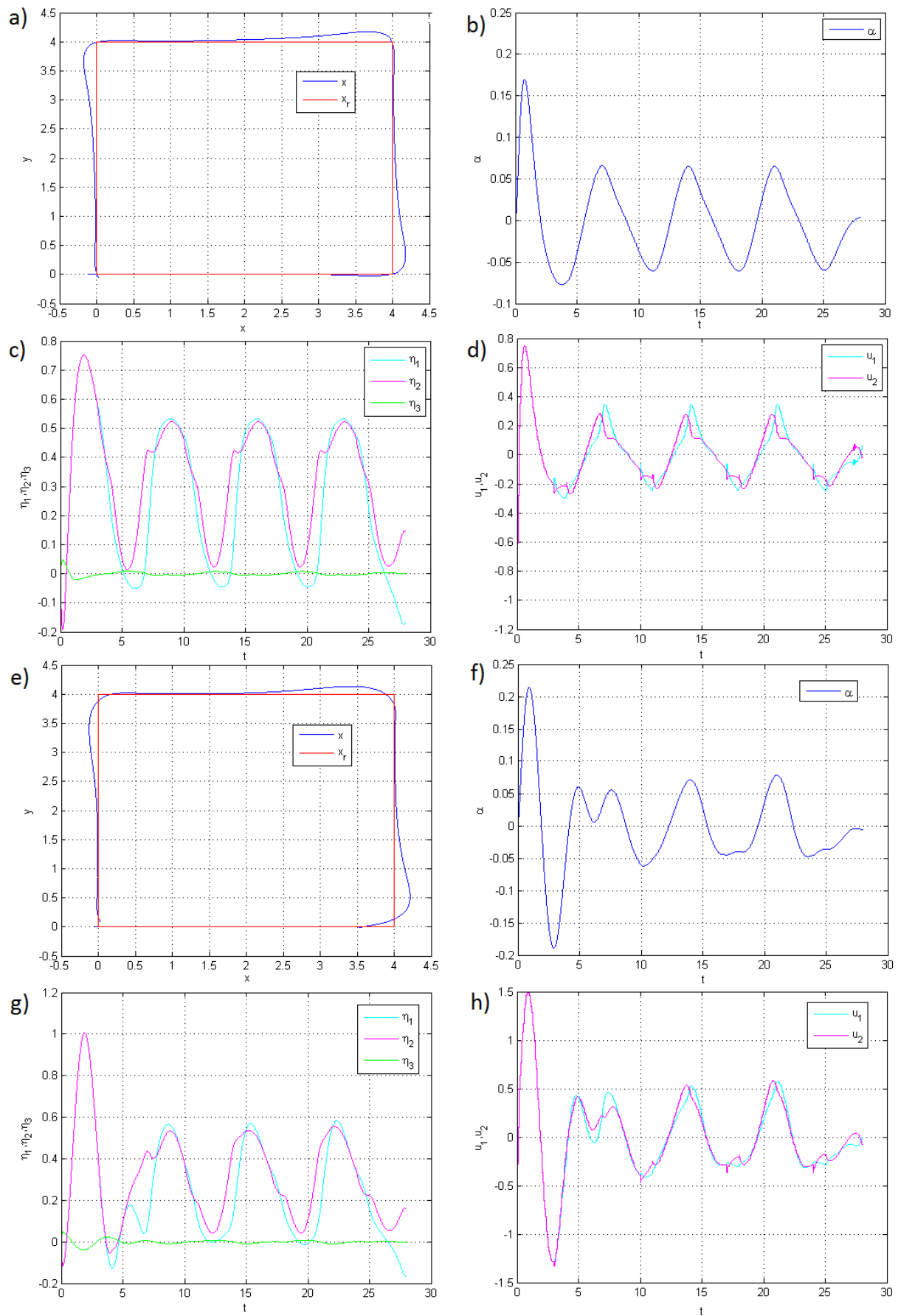
Zaobserwowano, że sterowanie predykcyjne doskonale radziło sobie nawet z dużą nieznaną model oraz niepewnością pomiaru stanu (wyjścia). Ten rodzaj sterowania można rekomendować do zastosowania w przypadku platformy balansującej.

4.6 Pomiar odchylenia kąтового w robocie balansującym

Robot balansujący różni się nieco budową od klasycznego odwróconego wahadła mobilnego. Różnica polega głównie na tym, że robot balansujący nie posiada stałego punktu odniesienia (brak poruszającej się na płaszczyźnie platformy). W związku z tym, nie jest możliwe zastosowanie klasycznego czujnika obrotu dokonującego pomiaru odchylenia wahadła względem platformy. Pomiarów odchylenia całego korpusu dokonuje się przy pomocy czujników inercyjnych (inklinometry, akcelerometry, żyroskopy). Niestety, żaden z sensorów nie dostarcza dostatecznie dokładnej informacji o odchyleniu kątowym. W praktyce, wykorzystuje się kilka czujników dla jednego systemu. Najczęściej są to inklinometr lub akcelerometr i żyroskop. Skutecznej fuzji danych pochodzących z tych sensorów dokonuje się przy pomocy filtru Kalmana [264, 184, 170, 132].

Akcelerometry mierzą przyspieszenia, w tym także statyczne, np. pochodzące od siły grawitacji. Zdolność tę wykorzystuje się do wyznaczania orientacji obiektów w przestrzeni. Przy budowie systemu sterowania robota FLASH, podjęto próbę stworzenia jednostki inercyjnej, zbudowanej z czujników przyspieszenia oraz żyroskopów. Ponieważ dostępne akcelerometry charakteryzują się znacznymi szumami sygnału na wyjściu zaleca się zastosowanie dodatkowego czujnika, np. żyroskopu. Czujniki żyroskopowe dostarczają informacji o prędkości obrotowej. Ich wadą jest to, że z czasem ulegają dryfowi (wynika to z ich konstrukcji wewnętrznej), pomiary stają się więc błędne. Ponadto, aby uzyskać bezwzględny pomiar odchylenia, niezbędne jest zastosowanie numerycznego całkowania, które również powoduje utratę informacji o wartości początkowej. Filtr Kalmana skutecznie niweluje niepożądane zjawisko dryfu oraz dokonuje przy tym fuzji pomiarów, dostarczając dokładnej i odszumionej informacji o odchyleniu (obrocie).

Coraz częściej można znaleźć gotowe IMU (*Inertial Measurement Unit*) [267, 190],



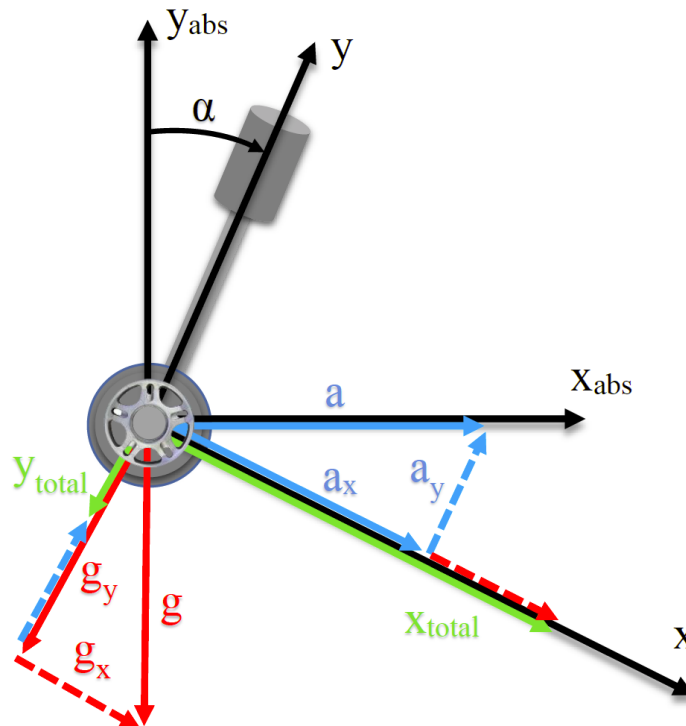
Rysunek 4.14: Wyniki symulacji algorytmów predykcyjnych

które są wyposażone w zintegrowane, szybkie procesory sygnałowe dostarczające gotowe, przetworzone dane. Nie mniej jednak, wyjścia tych jednostek pomiarowych również ulegają dryftowi.

4.6.1 Wyznaczenie odchylenia podstawie pomiaru przyspieszeń

Akcelerometry dostarczają informacji o kierunku działania siły, w tym statycznej siły grawitacji. Pomiaru odchylenia wahadła od kierunku działania siły grawitacji, można dokonać przy pomocy dwóch zainstalowanych prostopadle względem siebie akcelerometrów.

Ponieważ robot znajduje się w ciągłym ruchu, odczytane z osi poziomej i pionowej wartości przyspieszeń będą składowymi siły grawitacji oraz przyspieszeń platformy. Należy ten fakt uwzględnić w trakcie obliczeń trygonometrycznych. Jedną z metod ilustruje rysunek 4.15



Rysunek 4.15: Uzyskanie kąta odchylenia od pionu na podstawie pomiarów z akcelerometrów

Po odczytaniu przyspieszeń z obu osi, dokonuje się następujących obliczeń

$$a = \pm \sqrt{a_x^2 + a_y^2 - 1},$$

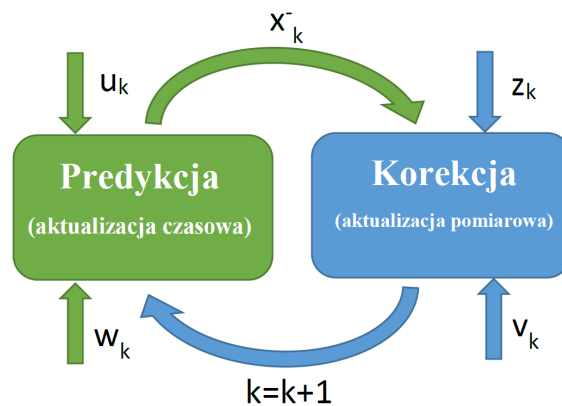
Następnie dokonuje się obliczeń

- dla $a > 0$ $\alpha = \arcsin\left(\frac{a_x - a_y \cdot a}{a_x^2 + a_y^2}\right)$
- dla $a < 0$ $\alpha = \arcsin\left(\frac{a_x + a_y \cdot a}{a_x^2 + a_y^2}\right)$
- dla $a = 0$ $\alpha = \arcsin(a_x)$

Wszystkie wartości przyspieszeń są znormalizowane do wielkości przyspieszenia grawitacyjnego ($g=1$). Jak widać z powyższych równań, kłopotliwe jest wyliczenie kierunku przyspieszenia robota a . Można go oszacować znając aktualne wartości sterowań napędów lub na podstawie przyspieszenia obu kół $sgn(a) = sgn(\ddot{\phi}_1 + \ddot{\phi}_2 - 2\ddot{\alpha})$, gdzie $\ddot{\alpha}$ to zróżniczkowana wartość pomiaru z żyroskopu.

4.6.2 Filtr Kalmana – wprowadzenie

Filtr Kalmana [264, 184, 170] jest często wykorzystywany w robotyce, gdzie układy percepcji otoczenia odgrywają istotną rolę. Są one często jedynym źródłem informacji o środowisku, w którym znajduje się robot. Posługując się tym algorytmem można wyznaczyć pomiarowo niedostępne zmienne, jedynie na podstawie znajomości modelu matematycznego obiektu i bieżących wartości wielkości pomiarowo dostępnych. Algorytm jest prosty i nie wymaga czasochłonnych obliczeń, dlatego może być zaimplementowany nawet na niezbyt wydajnych dolnej warstwy systemu sterowania.



Rysunek 4.16: Cykl dyskretnego filtra Kalmana

Filtr Kalmana jest dwufazowym rekursywnym algorytmem filtracji, w którym pierwsza faza nazywana jest predykcją lub aktualizacją czasową. Druga faza nazywa się korekcją lub aktualizacją pomiarową (Rysunek 4.16). W trakcie predykcji, bazując na stanie z poprzedniego kroku oraz wejścia (jeśli występuje), wyznacza się estymowaną wartość stanu \hat{x} oraz jego macierz kowariancji i są to wartości a priori. Pomiar z w drugiej fazie jest pewną formą sprzężenia zwrotnego. Na jego podstawie dokonuje się wyznaczenia wartości a posteriori dla stanu i jego macierzy kowariancji.

Proces i układ pomiarowy opisuje się przy pomocy liniowych modeli matematycznych

$$\begin{aligned}x_k &= Ax_{k-1} + Bu_{k-1} + Gw_{k-1}, \\z_k &= Hx_k + v_k.\end{aligned}$$

Pierwsze równanie różnicowe jest to model procesu, pokazujący powiązanie aktualnego stanu x_k z poprzednim x_{k-1} przy sterowaniu u_{k-1} i szumie w_{k-1} będącym szumem procesu. Drugie równanie to model pomiaru z_k , gdzie v_k to zakłócenia pomiaru. Zarówno w_k jak i v_k reprezentują biały szum Gaussa.

$$p(w) \sim N(0, Q), \quad p(v) \sim N(0, R).$$

Zapis prawdopodobieństw warunkowych dla powyższych modeli, będzie wyglądał następująco

$$p(x_k|x_{k-1}) \sim N(Ax_{k-1} + Bu_k, Q),$$

$$p(z_k|x_k) \sim N(Hx_k, R).$$

Teraz zdefiniujmy błędy szacowania, w których \hat{x}^- to estymowany stan a priori uzyskany z procesu, a \hat{x} to estymowany stan a posteriori uwzględniający pomiar z_k .

$$e_k^- = x_k - \hat{x}_k^-,$$

$$e_k = x_k - \hat{x}_k,$$

gdzie e_k^- to błąd a priori, a e_k to błąd a posteriori. Są to różnice pomiędzy rzeczywistym stanem, a wartością estymowaną. W praktyce, rzeczywiste wartości stanu x_k nie są znane. Macierze kowariancji mają postać

$$P_k^- = E[e_k^-, e_k^-],$$

$$P_k = E[e_k, e_k],$$

gdzie P_k^- to macierz kowariancji a priori, a P_k to macierz kowariancji a posteriori. Wzmocnienie Kalmana K_k jest rodzajem wagi, z jaką wpłynie faza korekcji na estymowany stan

$$K_k = P_k^- H^T (H P_k^- H^T + R)^{-1}.$$

Równanie, które mówi jakie jest optymalne skorygowanie prognozy w chwili k , bazujące na wszystkich dotychczasowych pomiarach będzie miało postać

$$\hat{x}_k = \hat{x}_k^- + K_k(z_k - H\hat{x}_k^-),$$

gdzie z_k to pomiar, a różnica $(z_k - H\hat{x}_k^-)$ nazywa się aktualizacją pomiarową. Pozostaje jeszcze skorygować macierz kowariancji

$$P_k = (I - K_k H) P_k^-,$$

gdzie I oznacza macierz jednostkową. Ostatecznie, algorytm ma postać

PREDYKCJA
$\hat{x}_k^- = Ax_{k-1} + Bu_{k-1}$
$P_k^- = AP_{k-1}A^T + Q$
KOREKCJA
$K_k = P_k^- H^T (H P_k^- H^T + R)^{-1}$
$\hat{x}_k = \hat{x}_k^- + K_k(z_k - H\hat{x}_k^-)$
$P_k = (I - K_k H) P_k^-$

Zakładamy, że równanie układu pomiarowego dla inklinometru zbudowanego z akcelerometrów (patrz podrozdział 4.6.1) oraz żyroskopu ma postać

$$\begin{aligned}\alpha_k &= \alpha_{k-1} + (\omega_{k-1} - g_{bias}) dt + w_{k-1} \\ \omega_k &= \omega_{k-1} - g_{bias} \\ z_k &= \alpha_k + v_k,\end{aligned}\tag{4.17}$$

gdzie α_k jest odchyleniem kątowym, ω_k jest prędkością kątową, g_{bias} oznacza dryf żyroskopu, w_k to szum żyroskopu, v_k to szum akcelerometrów, a dt to okres między pomiarami. Pomiar prędkości kątowej ω_k , odczytany z żyroskopu, uwzględniony zostanie już w fazie predykcji jako wymuszenie u_k , a pomiar odchylenia uzyskany z akcelerometrów w fazie korekcji. Na podstawie założonego opisu układu pomiarowego można utworzyć wektor stanu $x = (\alpha, \omega, g_{bias})$ i zapisać macierze

$$A = \begin{bmatrix} 1 & 0 & -dt \\ 0 & 0 & -1 \\ 0 & 0 & 1 \end{bmatrix},$$

$$B = \begin{bmatrix} dt \\ 1 \\ 0 \end{bmatrix}$$

oraz wyjście filtru

$$H = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}.$$

Macierz R odnosi się do szumu akcelerometru i ma wymiar 1×1 , macierz kowariancji Q ma wymiar 3×3

$$Q = \begin{bmatrix} Q_{11} & 0 & 0 \\ 0 & Q_{22} & 0 \\ 0 & 0 & Q_{33} \end{bmatrix}.$$

Zarówno Q i R należy wyznaczyć eksperymentalnie. Wymiar macierzy K jest 1×3 . Macierz G nie jest potrzebna do implementacji algorytmu.

Wyniki pracy, standardowego filtru Kalmana przedstawiono na pierwszych czterech wykresach na rysunku 4.18 odpowiednio dla parametrów a) $R = 1$, $Q = I_3$, b) $R = 10$, $Q = I_3 \cdot 0.1$ c) $R = 100$, $Q = I_3 \cdot 0.01$ d) $R = 1000$, $Q = I_3 \cdot 0.001$.

4.6.3 Rozszerzony filtr Kalmana (EKF)

Standardowy filtr Kalmana stosuje się w dyskretnych procesach do optymalnej estymacji stanu. Zarówno proces, jak i związek pomiędzy pomiarem i procesem, opisuje się liniowymi modelami. W przypadku nieliniowych równań stanu można skorzystać z tzw. rozszerzonego filtra Kalmana (*Extended Kalman Filter*), w którym równania stanu są zlinearyzowane poprzez zastąpienie nieliniowych funkcji ich rozwinięciami w szereg Taylora. Zakłada się, że nieliniowy układ równań ma postać

$$\begin{aligned}x_k &= f(x_{k-1}, u_{k-1}, w_{k-1}), \\ z_k &= h(x_k, v_k),\end{aligned}$$

gdzie funkcja przejścia stanu f , zależy od sterowania u_{k-1} i szumu procesu w_{k-1} . Funkcja h_k jest funkcją wyjścia opisująca zależność między stanem a pomiarem i jego błędem v_k . Zarówno w_k jak i v_k reprezentują białe szumy Gaussa.

W praktyce nie są znane wartości szumów w_k i v_k w każdym kroku. Można zatem przybliżyć stan i pomiar nie uwzględniając szumów

$$\begin{aligned}\tilde{x}_k &= f(x_{k-1}, u_{k-1}, 0), \\ \tilde{z}_k &= h(x_k, 0).\end{aligned}$$

Zlinearyzowane równania układu mają postać

$$\begin{aligned}x_k &\approx \tilde{x}_k + A^J(x_{k-1} - \hat{x}_{k-1}) + W^J w_{k-1}, \\ z_k &\approx \tilde{z}_k + H^J(x_k - \tilde{x}_k) + V^J v_k,\end{aligned}$$

gdzie

- x_k i z_k – rzeczywiste, aktualne wektory stanu i pomiaru,
- \tilde{x}_k i \tilde{z}_k – przybliżone wartości stanu i pomiaru,
- \hat{x}_k – oszacowany stan a posteriori (2 faza),
- w_k i v_k – szum procesu i pomiaru ($N(0, Q), N(0, R)$),

a macierze

$$\begin{aligned}A^J &= \frac{\partial f}{\partial x}(\hat{x}_{k-1}, u_{k-1}, 0), & W^J &= \frac{\partial f}{\partial w}(\hat{x}_{k-1}, u_{k-1}, 0), \\ H^J &= \frac{\partial h}{\partial x}(\tilde{x}_k, 0), & V^J &= \frac{\partial h}{\partial v}(\tilde{x}_k, 0).\end{aligned}$$

Wzmocnienie K_k oblicza się w sposób standardowy. Jakobiany A^J, W^J, H^J, V^J należy wyznaczyć w każdym cyklu działania algorytmu. Dlatego stosujemy notację wyglądającą A_k, W_k, H_k, V_k . Często w praktyce, jako W_k, V_k przyjmuje się macierz jednostkową I , więc Q i R traktuje się jak w przypadku zwykłego KF, a w każdym kroku oblicza się jedynie A_t i H_k . Ostatecznie, algorytm ma postać

PREDYKCJA
$\hat{x}_k^- = f(\hat{x}_{k-1}, u_{k-1}, 0)$ $P_k^- = A_k P_{k-1} A_k^T + W_k Q_{k-1} W_k^T$
KOREKCJA
$K_k = P_k^- H_k^T (H_k P_k^- H_k^T + V_k R_k V_k^T)^{-1}$ $\hat{x}_k = \hat{x}_k^- + K_k (z_k - h(\hat{x}_k^-, 0))$ $P_k = (I - K_k H_k) P_k^-$

Największą wadą EKF jest to, że rozkład zmiennej losowej, po przejściu przez nieliniowe transformacje, nie będzie już normalny. EKF jest prostym estymatorem stanu typu ad hoc, opartym na linearyzacji równań układu. Podejście to może być stosowane tylko w procesach, których zakłócenia mają charakter gaussowski. Ponadto, występujące nieliniowości nie mogą być zbyt duże, gdyż algorytm bazuje na przybliżeniu liniowym. W przypadku skomplikowanych układów, EKF może nie dać zadowalającego rozwiązania. Ponadto, podejście to wymaga ciągłego wyliczania macierzy przybliżenia liniowego, co jest czasochłonne i nie zawsze możliwe do wykonania w czasie rzeczywistym.

4.6.4 Bezśladowy filtr Kalmana (UKF)

Bezśladowy filtr Kalmana (*Unscented Kalman Filter*) został zaprezentowany w pracach [124, 262, 128] jako alternatywa dla EKF. Jego główną zaletą jest zdecydowanie lepsza dokładność oraz możliwość zastosowania do dowolnych układów nieliniowych. Algorytm nie linearyzuje modeli procesów i pomiarów, ale operuje na parametrach statystycznych poddanych pewnym przekształceniom nieliniowym. Podobnie jak w EKF, stan jest również reprezentowany przez zmienną losową typu Gaussa, ale określoną przez minimalny zestaw ściśle dobranych próbek. Próbkę te pozwalają na precyzyjne uzyskanie wartości oczekiwanej oraz kowariancji zmiennej losowej. Kiedy zmienna losowa jest poddawana transformacjom nieliniowym, algorytm jest w stanie wychwycić jej parametry z dokładnością do trzeciego rzędu, dla zmiennej z rozkładem Gaussa oraz drugiego rzędu dla zakłóceń typu nie-gaussowskiego. Złożoność obliczeniowa jest podobna lub mniejsza niż EKF i nie wymaga liczenia jakobianów ani hessianów. Ponadto, nie wymaga tak dużej ilości próbek jak, np. filtry cząsteczkowe. Podstawą działania UKF są przekształcenia bezśladowe [251].

Przekształcenia bezśladowe, to jedna z metod analizy statystycznej, pozwalająca na wyznaczenie wartości średniej oraz kowariancji zmiennej losowej x o wymiarze L poddanej nieliniowej transformacji

$$y = f(x) \quad (4.18)$$

Dla zmiennej x możemy zapisać jej wartość oczekiwaną \bar{x} oraz kowariancję P_x . W celu przeprowadzenia obliczeń statystycznych tworzy się zbiór (macierz) \mathcal{X} o wymiarze $2L + 1$ wektorów \mathcal{X}_i , wraz z odpowiadającymi im wartościami wagowymi W_i . Wektory te są nazywane wektorami sigma i tworzone wg reguły

$$\begin{aligned} \mathcal{X}_0 &= \bar{x}, \\ \mathcal{X}_i &= \bar{x} + \left(\sqrt{(L + \lambda)P_x} \right)_i, & i = 1, \dots, L \\ \mathcal{X}_i &= \bar{x} - \left(\sqrt{(L + \lambda)P_x} \right)_{i-L}. & i = L, \dots, 2L, \end{aligned}$$

gdzie $\left(\sqrt{(L + \lambda)P_x} \right)_i$ oznacza i -ty wiersz pierwiastka z macierzy $(L + \lambda)P_x$. Odpowiadające wektorom sigma wartości wagowe wyliczamy wg

$$\begin{aligned} W_0^{(m)} &= \lambda / (L + \lambda), \\ W_0^{(c)} &= \lambda / (L + \lambda) + (1 - \alpha^2 + \beta), \\ W_i^{(m)} = W_i^{(c)} &= \lambda / \{2(L + \lambda)\}, & i = 1, \dots, 2L, \end{aligned}$$

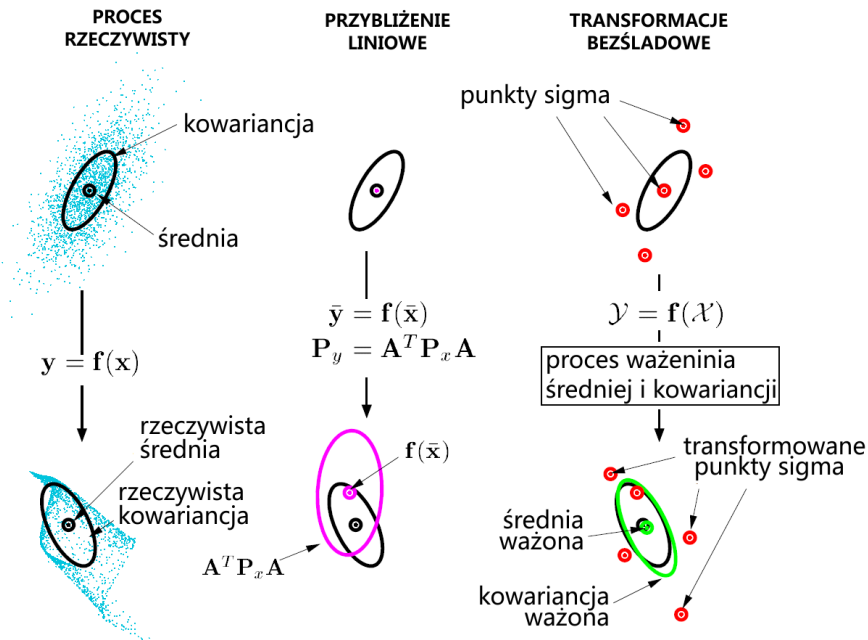
gdzie $\lambda = \alpha^2(L + \kappa) - L$ jest parametrem skalującym. Współczynnik α decyduje o rozrzucie punktów sigma wokół \bar{x} ; zwykle przyjmuje się małą wartość rzędu 0.001. Współczynnik κ to również parametr skalujący, zwykle przyjmuje się 0. Ostatni wprowadzony parametr to β , który zawiera wiedzę a priori o rozkładzie zmiennej losowej x (dla rozkładów gaussowskich optymalną wartością jest 2).

Następnie wektory te są przekształcane przez nieliniową funkcję transformacji (4.18) rozważanego układu

$$\mathcal{Y}_i = f(\mathcal{X}_i). \quad i = 1, \dots, 2L.$$

Wartość oczekiwana i kowariancja dla y uzyskanego a posteriori są przybliżana wagami W_i wg wzorów

$$\begin{aligned} \bar{y} &\approx \sum_{i=0}^{2L} W_i^{(m)} \mathcal{Y}_i, \\ P_y &\approx \sum_{i=0}^{2L} W_i^{(c)} \{\mathcal{Y}_i - \bar{y}\} \{\mathcal{Y}_i - \bar{y}\}^T. \end{aligned}$$



Rysunek 4.17: Porównanie procesu propagacji zmiennej losowej dla procesu bazującego na przybliżeniu liniowym układu oraz transformacji bezśladowych (rysunek zaczerpnięty z pracy [262]).

Na rysunku 4.17 przedstawiono porównanie procesu propagacji zmiennej losowej dla przybliżenia liniowego i transformacji bezśladowych.

Przedstawiony aparat matematyczny zastosowano do opisu rekursywnej estymacji stanu. Stan będący zmienną losową można rozumieć jako powiązanie rzeczywistego stanu procesu oraz zakłóceń. Rekursywny estymator stanu w postaci bezśladowego filtra Kalmana można zapisać w postaci układu równań,

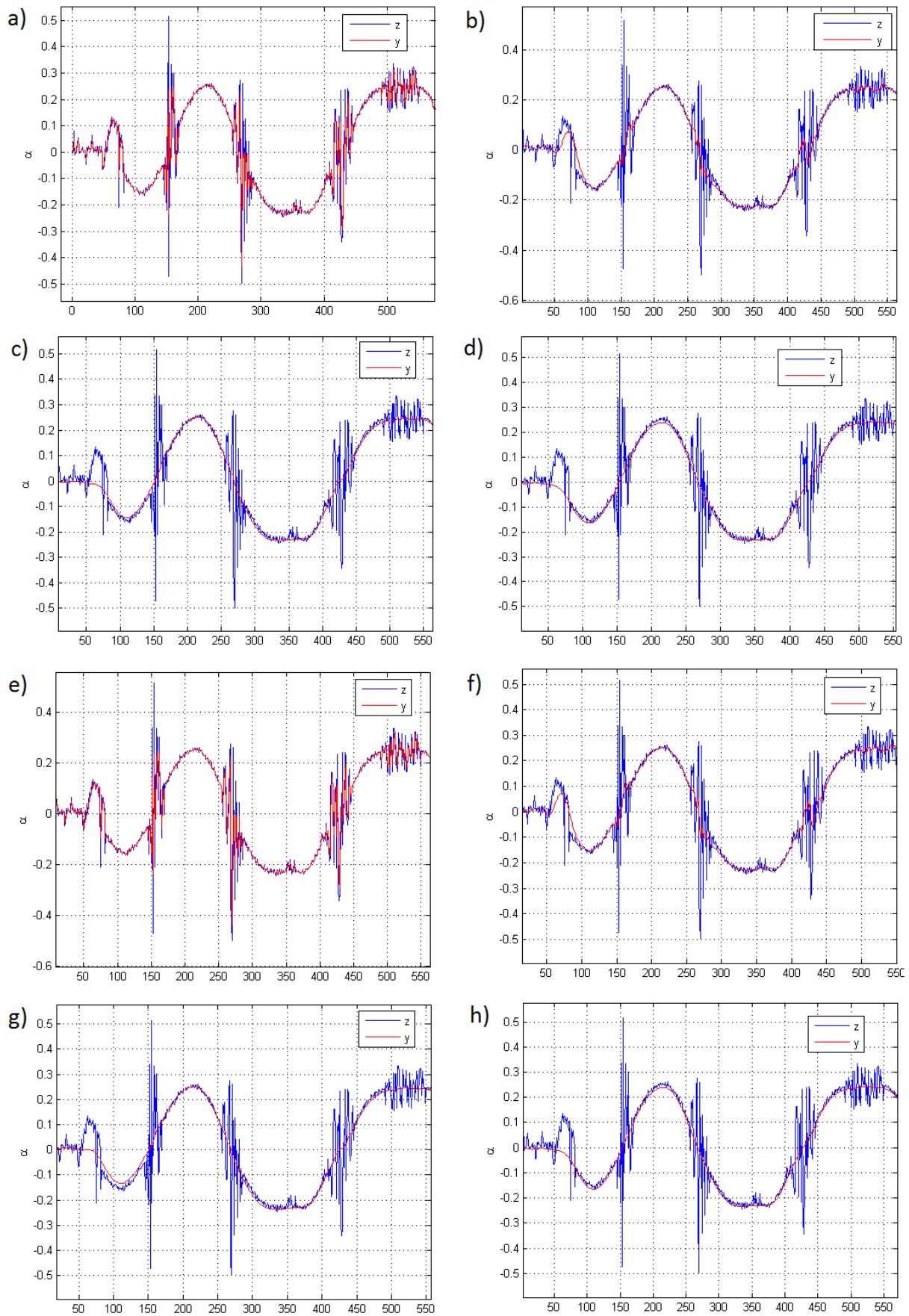
PREDYKCJA
$\mathcal{X}_{k k-1} = F[\mathcal{X}_{k-1}]$
$\hat{x}_k^- = \sum_{i=0}^{2L} W_i^{(m)} \mathcal{X}_{i,k k-1}$
$P_{x_k}^- = \sum_{i=0}^{2L} W_i^{(c)} [\mathcal{X}_{i,k k-1} - \hat{x}_k^-][\mathcal{X}_{i,k k-1} - \hat{x}_k^-]^T + Q$
$\mathcal{Y}_{k k-1} = H[\mathcal{X}_{k k-1}]$
$\hat{y}_k = \sum_{i=0}^{2L} W_i^{(m)} \mathcal{Y}_{i,k k-1}$
$P_{y_k} = \sum_{i=0}^{2L} W_i^{(c)} [\mathcal{Y}_{i,k k-1} - \hat{y}_k][\mathcal{Y}_{i,k k-1} - \hat{y}_k]^T + R$

KOREKCJA
$P_{x_k y_k} = \sum_{i=0}^{2L} W_i^{(c)} [\mathcal{X}_{i,k k-1} - \hat{x}_k^-] [\mathcal{Y}_{i,k k-1} - \hat{y}_k]^T$
$\mathcal{K} = P_{x_k y_k} P_{y_k}^{-1}$
$\hat{x}_k = \hat{x}_k^- + \mathcal{K}(z_k - \hat{y}_k)$
$P_{x_k} = P_{x_k}^- - \mathcal{K} P_{y_k} \mathcal{K}^T$

w których Q oznacza macierz kowariancji procesu, a R jest macierzą kowariancji pomiaru. Wspólne dla obydwu faz wyliczane są wektory sigma wg formuły

$$\mathcal{X}_{k-1} = \left[\hat{x}_{k-1}, \hat{x}_{k-1} \pm \left(\sqrt{(L + \lambda) P_{k-1}} \right)_i \right] \quad i = 1, \dots, 2L.$$

Ten sam zestaw rzeczywistych danych pomiarowych, który zastosowano dla standardowego filtra Kalmana przebadano używając bezśladowego filtra Kalmana dla układu (4.17). Wyniki zaprezentowano na rysunku 4.18, odpowiednio dla parametrów e) $R = 1$, $Q = I_3$, f) $R = 10$, $Q = I_3 \cdot 0.1$ g) $R = 100$, $Q = I_3 \cdot 0.01$ h) $R = 1000$, $Q = I_3 \cdot 0.001$. Zaobserwowano, że uzyskane wykresy pokrywają się. Oznacza to, że dla układów liniowych bezśladowy filtr Kalmana, zadziałał identycznie jak klasyczny filtr Kalmana. Sugeruje to, że szumy układu pomiarowego są typu gaussowskiego. Można zatem przyjąć, że zaimplementowany na platformie klasyczny filtr Kalmana działa prawidłowo.



Rysunek 4.18: Wyniki pracy klasycznego filtra Kalmana i bezszładowego filtra Kalmana

Rozdział 5

System sterowania robota FLASH

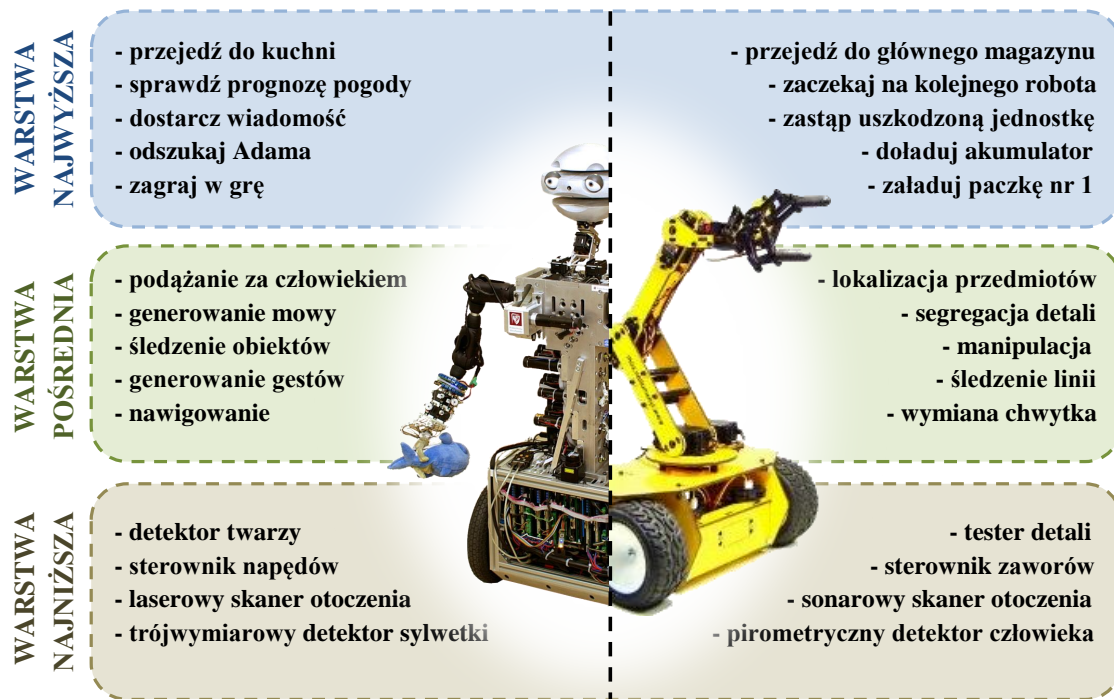
System sterowania robota FLASH stworzono w oparciu o zewnętrzne oprogramowanie oraz zestaw wybranych bibliotek. Dostarcza on mechanizmów umożliwiających wykorzystanie wszystkich komponentów robota FLASH. System zapewnia pełną synchronizację ruchów platformy, ramion oraz dłoni. Do realizacji percepcji, komunikacji i interakcji, robot został wyposażony w systemy audio-wideo, które przy pomocy kamer, głośników i mikrofonów. Zadbano o to, aby dane napływające z układów sensorycznych, przetwarzane były w czasie rzeczywistym, w efekcie czego uzyskano wiarygodne i interaktywne zachowanie robota. Zastosowany zestaw narzędzi programistycznych pozwala na nieograniczoną rozbudowę systemu. W relatywnie krótkim czasie można stworzyć dowolny moduł dostarczający brakujących umiejętności. System zrealizowano w ramach paradygmatu trójwarstwowej architektury sterowania.

Stworzono kompletne oprogramowanie, które jest łatwe w instalacji, konfiguracji oraz użytkowaniu, nawet przez mniej doświadczonych programistów robotów. Dzięki elastycznej strukturze systemu, jego działanie przetestowano na kilku konstrukcjach robotów (Pioneer [2], PeopleBot [2], FLASH [136], Romek [37], Samuel [37], EMYS [136]). Przygotowany system pozwala na dowolne zestawianie ze sobą, np. platforma mobilna-głowa, ramiona-dłonie lub ramiona-głowa. Można zatem zastosować go dla zestawu konstrukcji np. Pioneer-Romek, PeopleBot-EMYS, itd. Do uruchomienia systemu wymagana jest jego poprawna instalacja oraz przygotowanie pliku konfiguracyjnego, w którym ustala się jakie komponenty systemu będą wykorzystywane oraz jakie będą ich parametry pracy. Przygotowane oprogramowanie jest dostępne dla systemu Windows oraz Linux. System sterowania robota FLASH opisano także w pracy [135], w której podkreślono otwarty charakter wytworzonego oprogramowania.

5.1 Koncepcja

Mówiąc o architekturze sterowania robotem, często wskazuje się na jej trójwarstwowy charakter [35, 87, 12]. Podział ten stosuje się ze względu na zakres działań poszczególnych komponentów. Systemy, począwszy od tych najprostszycy, a skończywszy na złożonych strukturalnie, składają się zwykle z modułów realizujących pewne funkcje w zakresie obsługi zasobów sprzętowych, ich funkcjonalnego wykorzystania i realizacji celów [12]. Moduły te są ulokowane w poszczególnych warstwach i mogą wykorzystywać współdzielone zasoby.

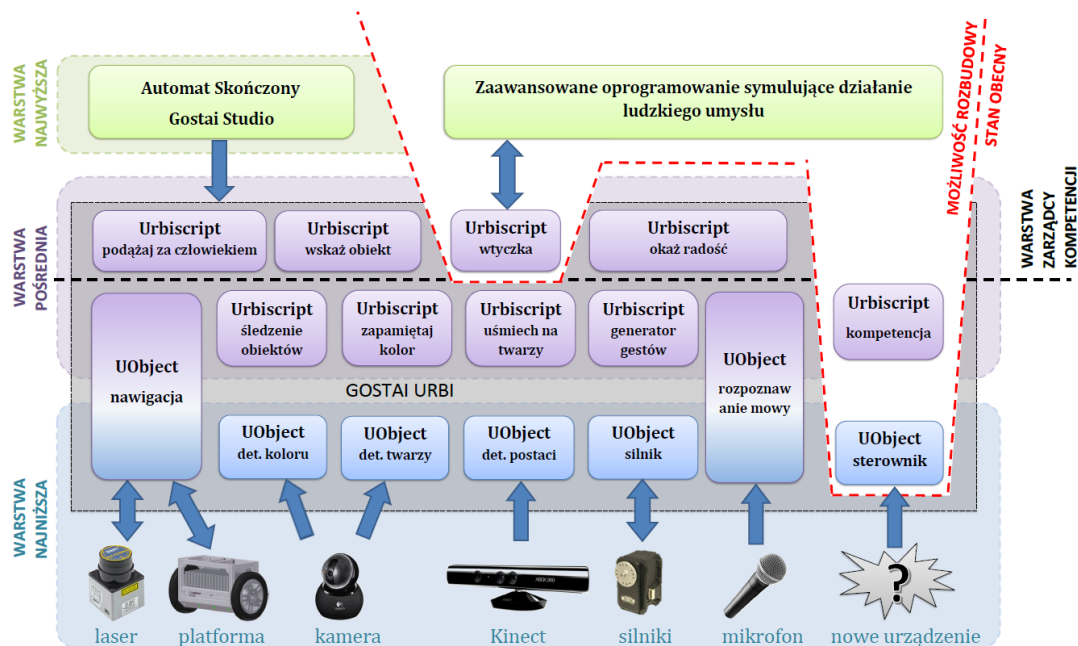
W ramach prac koncepcyjnych zmierzających do opracowania architektury systemu sterowania, podjęto próbę zaadaptowania wspomnianego podziału zadań dla robota spo-



Rysunek 5.1: Przykładowa realizacja zadań w trójwarstwowym systemie sterowania

łeczny. Na rysunku 5.1 zilustrowano architekturę trójwarstwową na przykładzie dwóch robotów: robota społecznego i manipulatora mobilnego. Warstwa najniższa obejmuje sterowanie robota mające bezpośredni dostęp do sprzętu. Odpowiedzialna jest ona za sterowanie napędami, śledzenie trajektorii, pobieranie i przetwarzanie obrazów z kamery oraz ostateczne dostarczenie przetworzonych informacji do warstwy pośredniej. Gdy nie wymaga się dużych mocy obliczeniowych, warstwa najniższa często jest realizowana z użyciem prostych układów mikroprocesorowych. Warstwa pośrednia odpowiada za funkcje robota i realizację poszczególnych kompetencji. To ich obecność, decyduje jakie zadania robot będzie w stanie wykonać. W przedstawionym przykładzie zarówno robot społeczny, jak i platforma mobilna posiadają manipulator. Obydwie konstrukcje powinny zatem posiadać kompetencję manipulacji. Ze względu na ich odmienne zastosowanie, pierwszy z nich może być także wyposażony w kompetencję wyrażania gestów, natomiast drugi, np. w możliwość spawania. Wspomniane kompetencje będą wykorzystywać te same zasoby sprzętowe czyli manipulator. Warstwa najwyższa obejmuje umysł robota. W zależności od potrzeb umysł może być zrealizowany w postaci automatu skończonego, systemu decyzyjnego, algorytmu szeregującego zadania lub z użyciem rozbudowanego oprogramowania symulującego działanie ludzkiego umysłu. Warstwa ta nie zawsze musi być w pełni autonomiczna, w niektórych przypadkach może być wspomagana przez człowieka.

Przedstawiona koncepcja to jedynie podział logiczny, który z różnych przyczyn nie zawsze musi być sztywno odzwierciedlony w strukturze oprogramowania, czy realizacji sprzętowej. Może się zdarzyć, że jeden moduł będzie pobierał z warstwy najniższej dane z dalmierzy, przetwarzał je i przesyłając sterowania do platformy, realizował kompletną kompetencję nawigacji. Z kolei, detektor gestów bazujący na obrazie z kamer stereowizyjnych i wymaga dużych mocy obliczeniowych może pracować na komputerze zdalnym, który realizuje także warstwę najwyższą.



Rysunek 5.2: Struktura oprogramowania

5.2 Warstwa integrująca

Jako główne narzędzie integrujące poszczególne moduły oprogramowania użyto platformy programistycznej Urbi firmy Gostai [252]. Oprogramowanie to integruje i zapewnia komunikację na dwóch najniższych poziomach trójwarstwowej architektury. Pozwala na dynamiczne ładowanie modułów nazywanych *UObject* oraz zapewnia pełną kontrolę ich pracy. Strukturę oprogramowania pokazano na rysunku 5.2.

Moduły *UObject* tworzy się po to, aby umożliwić dostęp do warstwy sprzętowej robota, a także w celu włączenia do systemu zewnętrznego oprogramowania dostępnego w postaci bibliotek. *UObject* mogą pracować na komputerze pokładowym lub być uruchamiane w trybie zdalnym, na wysokowydajnych jednostkach obliczeniowych. Dzięki takiemu mechanizmowi, możliwe jest zrealizowanie wielu zadań jednocześnie, nawet jeśli są to pracochłonne procesy przetwarzania obrazu. Wtyczki *UObject* tworzy się w języku C++, a ich niskopoziomowa implementacja pozwala na optymalne wykorzystanie zasobów sprzętowych. W zależności od potrzeb, poszczególne bloki systemu mogą pracować w sposób synchroniczny lub asynchroniczny. Decyduje o tym programista na poziomie języka skryptowego. Kontrola dostępu do zasobów sprzętowych i pamięciowych jest wykonywana przez Urbi automatycznie.

Urbi dostarcza także skryptowy język programowania robotów *urbiscript*, który jest zorientowany na równoległe zarządzanie zdarzeniami oraz synchronizowanie poszczególnych komponentów systemu sterowania. Jest on wykorzystywany głównie w warstwie pośredniej. Zadaniem tej warstwy jest wykonywanie poleceń umysłu, czyli współbieżne, synchroniczne uruchamianie poszczególnych kompetencji, a także informowanie warstwy wyższej o powodzeniu wykonanego zadania. O tym, jakie komponenty biorą udział w wykonywaniu każdego zadania, decyduje odpowiednio skonfigurowana warstwa zarządcy kompetencji. Konfiguruje się go dla konkretnego fizycznego zestawienia (np. platforma+głowa+Kinect). Zarządca kompetencji realizuje polecenia warstwy najwyższej, wykorzystując dostępne dla danego urzeczywistnienia kompetencje. Na przykład, otrzymanie instrukcji „Podążaj za człowiekiem” będzie skutkowało zlokalizowaniem człowieka na podstawie systemu wizyj-

nego, uruchomieniem nawigacji i podążanie za nim. Ponadto, zgodnie z definicją robota społecznego, powinien on wykonywać polecenie, okazując przy tym swój stan emocjonalny przy pomocy dostępnych środków wyrazu.

Poniżej, przedstawiono pokrótce niektóre moduły oraz opisano sposoby implementacji wybranych kompetencji. Ich wybór był podyktowany scenariuszami zaplanowanych eksperymentów HRI. Pełna dokumentacja przedstawianego, oprogramowania znajduje się w pod adresem [79].

5.3 Warstwa najniższa - moduły oprogramowania

Dla robota FLASH napisano zestaw modułów, który wykorzystuje zbiór popularnych bibliotek i sterowników pozwalających na użycie wszystkich zainstalowanych w robocie komponentów, takich jak silniki, dalmierze, kamery, itd. W trakcie realizacji zadbane o przenośny charakter wytworzonego oprogramowania. Wyjątek stanowią moduły korzystające z bibliotek dostępnych dla jednego konkretnego systemu operacyjnego. W takim przypadku zaproponowano alternatywny moduł dla pozostałych systemów operacyjnych.

5.3.1 Moduł nawigacyjny

Jak wspomniano w rozdziale 3.2, oprogramowanie sterownika platformy balansującej zostało napisane w taki sposób, aby zachować kompatybilność z produktami firmy Mobile-Robots [2], np. Pioneer, PeopleBot, PowerBot, itd. Zatem, moduł stworzony do sterowania platformą mobilną robota FLASH może być wykorzystany z każdą z wymienionych konstrukcji. Moduł ten, nazwany **UAria** bazuje na otwartych narzędziach programistycznych *ARIA* [6] dostarczonych przez producenta. UAria zapewnia także obsługę skanerów laserowych, takich jak Sick LMS1xx, LMS2xx oraz Hokuyo URG 04-LX. Ten ostatni skaner zainstalowano w robocie FLASH. Zaimplementowano także rozszerzoną wersję modułu nawigacji, który bazuje na komercyjnych bibliotekach *ARNL* [9]. Zwiększone możliwości dotyczą pełnej implementacji systemu SLAM (*Simultaneous Localization And Mapping*). Poniżej zaprezentowano kilka wybranych funkcji, wywołanych na poziomie skryptowym.

Przykład:

```
UAria.setTransSpeed(speed);           // zadaj prędkość liniową platformy
UAria.setRotSpeed(speed);             // zadaj prędkość kątową platformy
...
x = UAria.getX();                     // odczytaj pozycję x uzyskaną na podstawie odometrii
y = UAria.getY();                     // odczytaj pozycję y uzyskaną na podstawie odometrii
...
[val1,val2,...] = UAria.getSonars();   // pobierz pomiary z sonarów
[val1,val2,...] = UAria.getLaser();    // pobierz pomiary ze skanera laserowego
```

5.3.2 Moduł sterowania napędami

Do komunikacji z niskopoziomowymi sterownikami napędów stworzono moduł **UDynamixel**. Komunikuje się on z napędami rąk, dłoni oraz głowy poprzez interfejs RS485. Wszystkie sterowniki w procesie komunikacji wykorzystują protokół, jaki obowiązuje w serwo napędach Dynamixel firmy Robotis [66]. Każdy napęd ma przypisany własny, konfigurowalny numer ID, który determinuje urządzenie docelowe. Możliwa jest konfiguracja podstawowych parametrów ruchu, takich jak maksymalny moment obrotowy, prędkość obrotowa, zakresy wychYLENIA i wiele więcej. UDynamixel umożliwia także odczytywanie

aktualnej informacji o stanie napędu (temperatura, wystąpienie przeciążenia lub przegrzania, napięcie zasilania układu). Wszystkie sterowniki podzielono na cztery grupy: głowa, ramiona, lewa i prawa dłoń. Dzięki temu, poszczególne magistrale nie są przeciążone informacjami napływającymi z generatorów trajektorii. Za wyliczanie i wysyłanie pozycji odpowiadają kompetencje gestykulacji, wyrażania emocji oraz mowy. Opis kompetencji znajduje się w dalszej części pracy. Poniżej zaprezentowano kilka wybranych funkcji modułu.

Przykład:

```
UDynamixel.SetMaxTorque(ID,torque);      // ustaw maksymalny moment obrotowy
UDynamixel.SetCWAngleLimit(ID,angle);    // ustaw prawostronny limit obrotu
...
UDynamixel.SetMovingSpeed(ID,speed);     // ustaw prędkość obrotową
UDynamixel.SetGoalPosition(ID,position); // ustaw pozycję docelową
```

5.3.3 Moduły przetwarzania dźwięku

Funkcje związane z generowaniem i przetwarzaniem dźwięku, są oparte na oprogramowaniu Loquendo [157], Microsoft SAPI [223] oraz bibliotekach SDL [226] i SFML [227]. Na podstawie każdej z nich stworzono oddzielny moduł odpowiedzialny za syntezę i rozpoznawanie mowy, a także pozwalający na odtwarzanie i nagrywanie dźwięków. Są to kolejno **USpeech**, **URecog**, **UPlayer** oraz **URecord**.

MS SAPI jest dostarczone bezpłatnie z każdym aktualnym systemem operacyjnym Windows. Na szczególną uwagę zasługuje tu dobrze działający system rozpoznawania mowy, który wymaga przygotowania odpowiednich słowników wyrażen gramatycznych. Niestety, oprogramowanie to nie pracuje pod kontrolą systemu Linux. Komercyjne oprogramowanie Loquendo zawiera wysokiej jakości system syntezy mowy. Pozwala on nie tylko na generowanie sekwencji w wielu językach, lecz także modyfikację parametrów wypowiedzi. Zarówno SAPI, jak i Loquendo umożliwiają, na podstawie syntezy mowy, generowanie znaczników VISEM, które opisano w dalszej części pracy.

Biblioteki SDL oraz SFML pozwoliły na stworzenie prostych odtwarzaczy i rejestratorów dźwięku. Są one przydatne do odtwarzania sentencji dźwiękowych przygotowywanych off-line. Wszystkie moduły przetwarzania dźwięku, uzupełniając się wzajemnie, realizują jeden z najważniejszych kanałów komunikacji międzyludzkiej, jakim jest mowa i dźwięk. Poniżej przedstawiono kilka przykładów użycia modułów.

Przykład:

```
USpeech.speak("Hello world");           // wypowiedz kwestię "Hello world"
phrase = URecog.result;                  // odczytaj rozpoznaną frazę
...
UPlayer.play("sound.wav");               // odtwórz plik dźwiękowy
```

5.3.4 Moduły przetwarzania obrazu

Jest to największa grupa modułów, jaką dysponuje system sterowania robota FLASH. Większość z nich opiera się na otwartym i popularnym zestawie bibliotek OpenCV [189, 28].

Najprostsze moduły, jak **UCamera**, umożliwiają dostęp do obrazu z kamer i zapewniają jego wstępną obróbkę. Kolejne moduły to głównie detektory, np. koloru, ruchu, obiektów czy elementów twarzy. Detektor obiektów **UObjectDetector** bazuje na algorytmach klasyfikacji Haara, dzięki któremu możliwe jest wykrycie na obrazie twarzy. **UColorDetector** jest wykorzystywany jako prosty detektor kolorów. Jego działanie jest



Rysunek 5.3: Efekt pracy niektórych modułów przetwarzania obrazu

dość mocno uzależnione od oświetlania. Moduł detekcji ruchu **UMoveDetector** bazuje na metodzie ekstrakcji ruchu pikseli na płaszczyźnie obrazu. Jest bardzo przydatny, jeżeli w procesie wykrywania człowieka, wykorzystuje się statyczną kamerę, np. urządzenie Kinect. W połączeniu z detekcją źródła dźwięku, może on służyć jako lokalizator aktywnych obszarów wokół robota. Interesujących danych na temat wykrytej twarzy człowieka dostarcza moduł **UFacetDetector**, który został zaimplementowany na bazie biblioteki FacET [181]. Moduł wylicza zestaw parametrów wykrytej na obrazie ludzkiej twarzy, które mogą być wykorzystane w procesie rozpoznawania emocji.

Ostatni moduł, **UImageTool**, udostępnia zestaw narzędzi do przetwarzania obrazu na poziomie skryptowym. Dzięki niemu można dokonać, między innymi, konwersji kolorów do innej przestrzeni, kadrowania obrazu, operacji morfologicznych, progowania, rysowania i wielu innych. Na rysunku 5.3 zaprezentowano kilka przykładów działania wymienionych detektorów (obiektów, kolorów, ruchu i parametrów twarzy). Poniżej zamieszczono przykładowe wywołania niektórych funkcji.

Przykład:

```

UCamera.GetImage(); // pobierz obraz z kamery
UObjectDetector.setImage(UCamera.image); // przełącz obraz do detektora obiektów
image = UObjectDetector.object; // pobierz obraz wykrytej twarzy
...
UImageTool.setImage(UCamera.image); // przełącz obraz do modułu
UImageTool.imgMedianBlur(parametr); // zastosuj filtr medianowy na przekazanym obrazie
[R,G,B] = UImageTool.getPixelValue(x,y); // pobierz wartość RGB wskazanego piksela

```

5.3.5 Moduł przetwarzania 3D

Układy sensoryczne, bazujące na skomplikowanych systemach stereo-wizyjnych do niedawna były wykorzystywane jedynie w przemyśle. Ceny tych urządzeń są bardzo wysokie. Rynek interaktywnych gier komputerowych dostarczył kilka tanich rozwiązań. Efektem ich

działania jest chmura punktów będąca trójwymiarową reprezentacją sceny. Przestrzenne czujniki serii Kinect i produkty firmy Asus doskonale sprawdzają się jako detektory głębi obrazu. Dane te, poddane analizie, umożliwiają zidentyfikowanie w przestrzeni charakterystycznych obiektów, np. sylwetki człowieka.

Firma Microsoft jako pierwsza dostarczyła wspomnianej technologii i obecnie zapewnia jeden z najbogatszych zestawów narzędzi programistycznych, które pozwalają na detekcję nie tylko człowieka, lecz także jego części ciała, gestów oraz emocji. Spośród innych otwartych projektów wykorzystujących czujniki przestrzenne na szczególną uwagę zasługuje platforma programistyczna OpenNI. Oprogramowanie to stanowi warstwę pośrednią, integrującą wiele dostępnych projektów programistycznych. Dzięki niej zunifikowane dane mogą być wykorzystywane przez oprogramowanie, dedykowanemu konkretnemu zastosowaniu, np. detekcji gestów, języka migowego, trójwymiarowej rekonstrukcji twarzy oraz mapy pomieszczenia. Jednym z najpopularniejszych jest zestaw bezpłatnych lecz nie otwartych bibliotek NITE, których dostawcą jest firma PrimeSense [205] – producent technologii którą wykorzystuje Microsoft oraz Asus. Zarówno OpenNI, jak i NITE zostały wykorzystane do stworzenia modułu **UKinectOpenNI**, który pozwala na wykrycie sylwetki człowieka, wyróżnienie jego kończyn, jak również otrzymanie informacji o odległościach wybranych elementów znajdujących się na przetwarzanym obrazie.

Moduł ten w połączeniu z innymi, np. UImageTool dostarcza wielu dodatkowych możliwości, między innymi śledzenia obiektu trzymanego w dłoni, jego klasyfikacji i detekcji koloru. Biblioteka ta, do niedawna była w pełni przenośna pomiędzy systemami operacyjnymi. Przykład działania modułu zaprezentowano na rysunku 5.4. Poniżej zamieszczono przykładowe wywołania niektórych funkcji modułu.

Przykład:

```
image = UKinectOpenNI.image; // pobierz obraz RGB
image = UKinectOpenNI.depth; // pobierz obraz głębi
...
[user1,user2,...] = UKinectOpenNI.getUsersID(); // podaj listę wykrytych sylwetek
[x,y,z] = UKinectOpenNI.getJointPosition(user1, leftHand); // podaj poz. lewej dłoni pierwszej sylwetki
```

5.3.6 Moduły odpowiadające za uczenie się

System sterowania został również wyposażony w moduły realizujące procesy uczenia się. Są one oparte głównie na algorytmach zawartych w bibliotece OpenCV i mogą zostać wykorzystane do zbierania oraz klasyfikowania informacji o otoczeniu i użytkownikach robota. Moduł **UKMeans** wykorzystuje algorytm K-centroidów (K-średnich) i może posłużyć, np. do uczenia robota kolorów. Zapamiętuje on dane w postaci liczb oraz przypisanych im etykiet. Kolejny moduł, **UEigenfaces**, bazuje na algorytmie twarzy własnych. Potrafi on rozpoznać człowieka na podstawie wyuczonej wcześniej bazy danych (próbek twarzy). Przykładowe wykorzystanie modułów uczenia się zaprezentowano poniżej.

Przykład:

```
UEigenfaces.train(objectDetector.object, "jan"); //zapamiętaj obraz jako twarz Jana
...
name = UEigenfaces.find(objectDetector.object); // sprawdź twarz znajduje się na podanym obrazie
UKNearest.train([132,86,12], "czerwony"); // zapamiętaj podane dane jako kolor czerwony
...
color = UKNearest.find([101,45,71],parameter); // sprawdź jaki kolor reprezentują podane dane
```



Rysunek 5.4: Efekt pracy modułu UKinectOpenNI

5.3.7 Pozostałe moduły

Opracowano także kilka dodatkowych modułów, które wspomagają pracę i programowanie robota. Moduł **UImageDisplay** służy do wyświetlania przetwarzanych obrazów. Zwykle pracuje w trybie zdalnym, na komputerze który jest wykorzystywany do celów monitoringu i zdalnego uruchamiania. Moduł **UJoystick** bazuje na bibliotece SDL i zapewnia możliwość zdalnego sterowania, np. poprzez bezprzewodowy kontroler Sony SIXAXIS. Moduł **USerial** umożliwia wykorzystanie portów szeregowych, zapewnia komunikację oraz konwersję danych do dogodnych dla programisty formatów (string, byte, vector).

W trakcie dwóch lat użytkowania robota FLASH powstało znacznie więcej modułów, których nie wymieniono w niniejszej pracy. Niektóre z nich są już nieaktualne i zostały zastąpione alternatywnym rozwiązaniem, które dostarcza większych możliwości. Pozostałe nie zostały jeszcze dostatecznie przebadane i przygotowane do pełnej implementacji.

5.4 Warstwa pośrednia - implementacja kompetencji

W warstwie pośredniej powinny znaleźć się wszystkie funkcje, które realizują kompetencje robota wraz z systemem zarządzania nimi. Istotne jest, aby wszystkie zadania wykonywane były synchronicznie. Gest ręką, któremu towarzyszy obrót platformy, powinien wykonać się w odpowiednich przedziałach czasowych. Ma to szczególne znaczenie, np. przy generacji mowy. Każdy użyty w systemie syntezy mowy generuje znaczniki, które informują o kształcie ust jaki powinien towarzyszyć wypowiedzianym dźwiękom. Położenia szczęki robota muszą nadążać za wypowiedzianą kwestią.

Zadaniem warstwy pośredniej jest wykonywanie poleceń umysłu, w tym generowania adekwatnych zachowań robota. O tym, jakie komponenty biorą udział w wykonywaniu

każdego zadania, decyduje odpowiednio skonfigurowana warstwa zarządcy kompetencji. Przygotowany zestaw zachowań powinien być sparametryzowany w taki sposób, aby możliwe było jego łatwe dopasowanie do aktualnej sytuacji i konfiguracji. Posługując się przykładem generatora mowy, parametrami wywołania powinny być nie tylko tekst, ale i czas wypowiedzi, poziom oraz barwa, np. na podstawie stanu emocjonalnego robota. To co zostanie udostępnione w warstwie pośredniej, będzie wykorzystywane przez oprogramowanie symulujące ludzki umysł. Im bogatszy repertuar dostępnych kompetencji i zachowań ekspresywnych, tym ciekawsze scenariusze będą możliwe do realizacji.

Niektóre kompetencje w całości zrealizowano na poziomie modułów *UObject*, inne zostały uzupełnione skryptami napisanymi w języku *urbiscript*, a jeszcze inne zaimplementowano w całości na poziomie skryptów. Aby ułatwić korzystanie z nich oraz zachować modułowy charakter systemu, pogrupowano je według funkcji, jaką realizują. Kompetencje dotyczące głowy zgromadzono w strukturze **head**, funkcje wideo w strukturze **video**, itd. Ostatecznie, utworzono siedem grup kompetencji, które opisano w dalszej części pracy. Jak już wspomniano, system sterowania robotem FLASH ma charakter modułowy. Dzięki temu, skrypty sterujące, np. głową, mogą być wykorzystane, z zachowaniem jej pełnej funkcjonalności, na innej platformie jezdnej, np. PeopleBot. Przewidziano także możliwość szybkiego odłączania brakujących lub uszkodzonych komponentów robota. Każdą z warstw można łatwo rozbudować poprzez dołączenie nowych modułów i napisanie wykorzystujących je skryptów. *UObjecty* oraz skrypty są ładowane do silnika *urbi-engine* w trakcie uruchamiania systemu.

5.4.1 Kompetencje grupy audio

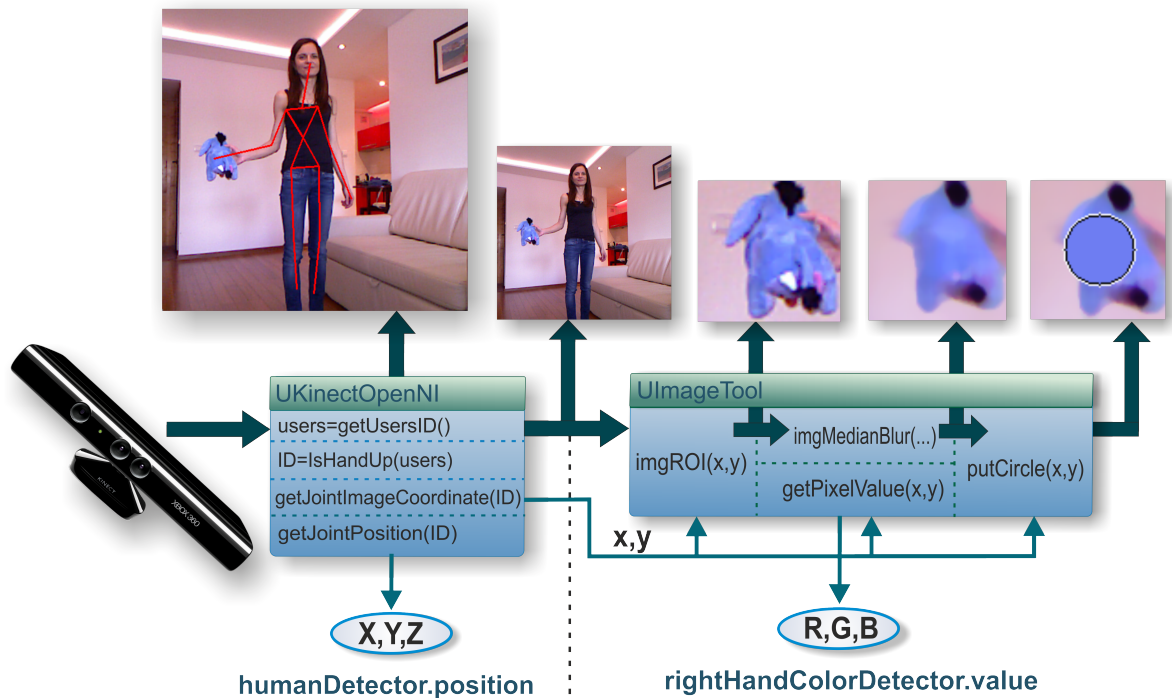
Audio jest przykładem grupy, w której wszystkie składowe kompetencje zostały zrealizowane na poziomie modułów *UObject*. Dostęp do poszczególnych funkcji bibliotek *UPlayer*, *USpeech* oraz *URecog* odbywa się poprzez kopię struktur danych, np. zamiast *USpeech.isSpeaking* wywołuje się funkcje *audio.speech.isSpeaking*. Przykłady wywołań zaprezentowano poniżej.

Przykład:

```
audio.speech.speak("Hello world"); // wypowiedz kwestię "Hello world"
phrase = audio.recognition.result; // odczytaj rozpoznaną frazę
...
audio.player.play("sound.wav"); // odtwórz plik dźwiękowy
```

5.4.2 Kompetencje grupy video

Grupa video, wykorzystuje wszystkie moduły odpowiedzialne za pobieranie obrazu z kamery, obsługę czujnika przestrzennego Kinect, przetwarzanie obrazu, a także detekcję obiektów, kolorów, ruchu oraz parametrów twarzy. Ich wspólne wykorzystanie pozwoliło na stworzenie kompetencji, które odgrywają kluczową rolę w procesie HRI. Podobnie jak w grupie audio, kompetencje dotyczące prostej detekcji są pełną kopią funkcji modułu *UObject*. Dodano jedynie opcję wyboru źródła obrazu (kamera lub Kinect) oraz możliwość ich selektywnego włączenia lub wyłączenia. Znacząco został rozbudowany moduł przetwarzania danych z urządzenia Kinect. W przypadku wykrycia wielu osób, system skupia się na tej, która ostatnia podniosła rękę (gest machania). Ponadto, stworzono kompetencje śledzenia obiektów trzymanyh przez człowieka w dłoni oraz odczytanie ich kolorów. Na rysunku 5.5 zilustrowano przykładowy przepływ danych oraz współpracę między modułami. Przykład wywołania niektórych kompetencji zaprezentowano poniżej.



Rysunek 5.5: Uproszczony schemat przepływu danych i współpraca pomiędzy modułami wideo

Przykład:

```

video.visionRun(); // uruchom system wizyjny robota
video.objectDetector1.source = "camera"; // ustaw kamerę, jako źródło obrazu pierwszego det. obiektów
video.objectDetector1.enabled = true; // uruchom pierwszy detektor obiektów
...
video.humanDetector.enabled = true; // uruchom detektor sylwetki
[x,y,z] = video.humanDetector.position; // pobierz położenie sylwetki w przestrzeni
...
video.rightHandColorDetector.enabled = true; // uruchom det. koloru obiektu trzymanego w prawej dłoni
[x,y,z] = video.rightHandColorDetector.position; // pobierz położenie prawej dłoni w przestrzeni
[R,G,B] = video.rightHandColorDetector.value; // pobierz war. koloru ob. trzymanego w prawej dłoni

```

5.4.3 Kompetencje grupy ml (*machine learning*)

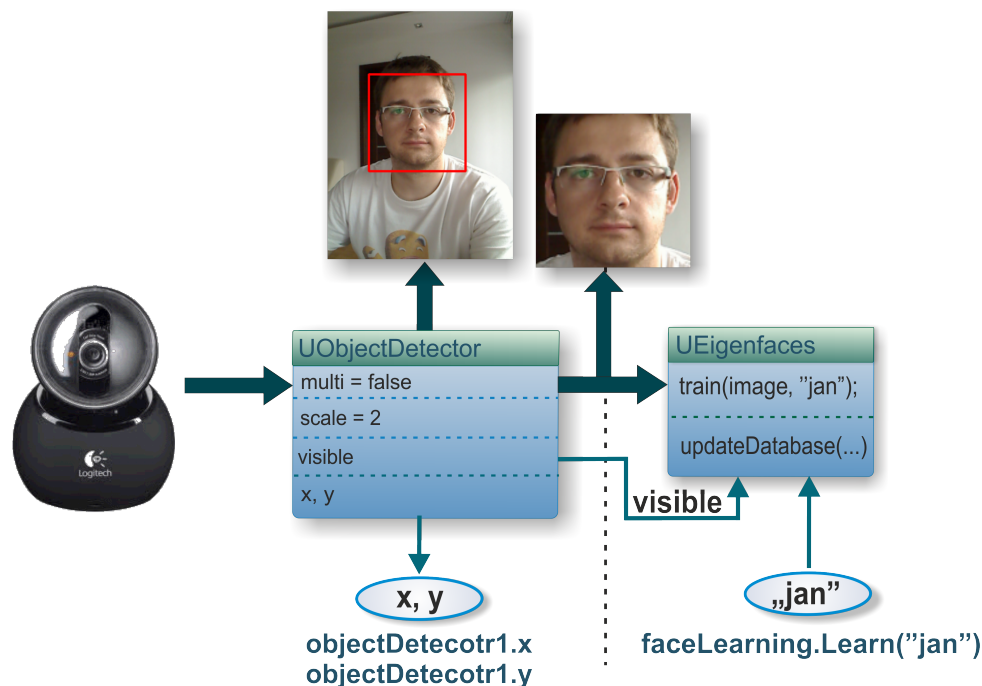
Jest to grupa kompetencji odpowiedzialnych za procesy uczenia się. Obecnie w systemie znajdują się tylko dwa moduły realizujące procesy uczenia. Jeden z nich został wykorzystany do nauki kolorów obiektów trzymany przez człowieka w dłoni oraz kolorów jego ubrań. Próbkę danych są dostarczane przez grupę kompetencji **video**. Drugi moduł uczenia się, wykorzystywany jest przez kompetencja rozpoznawania użytkownika. Pobiera ona dane z grupy **video**, a konkretnie z detektora obiektów (twarzy). Przewidziano także możliwość zapisu i wczytania obydwu wyuczonych baz. Na rysunku 5.6 zilustrowano przykładowy przepływ danych i współpracę między modułami. Poniżej zamieszczono przykład wywołania niektórych funkcji.

Przykład:

```

ml.colorLearning.LearnFromLeftHand("red"); // zapamiętaj kolor obiektu trzymanego w prawej dłoni
// jako kolor czerwony
color = ml.colorLearning.CheckFromLeftHand(); // sprawdź kolor obiektu trzymanego w prawej dłoni
...
ml.faceLearning.Learn("jan"); // zapamiętaj twarz Jana
name = ml.faceLearning.Check(); // sprawdź wykrytą twarz
...

```

Rysunek 5.6: Uproszczony schemat przepływu danych i współpraca pomiędzy modułami wideo

```
m1.Save(fileName, colors_base, faces_base); // zapisz wyuczone bazy danych
m1.Load(fileName, colors_base, faces_base); // wczytaj wyuczone bazy danych
```

5.4.4 Kompetencje grupy body (platforma)

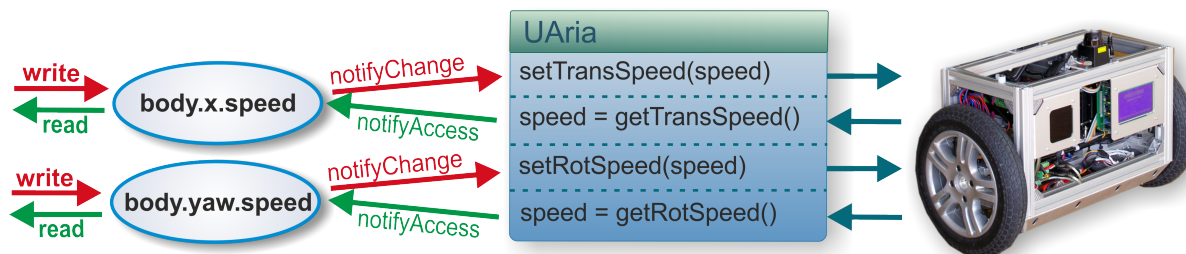
Grupa ta w całości bazuje na module `UAria`, jednakże nie wykorzystuje jedynie kopii struktur danych, jak ma to miejsce w przypadku grupy kompetencji audio. Dostęp do poszczególnych funkcji modułu odbywa się poprzez mechanizm notyfikacji kontrolujący dostęp do zmiennych (slotów) obiektu. Mechanizm ten jest częścią języka *urbiscript* i polega na wyborze odpowiedniej funkcji w czasie odczytu zmiennej i innej funkcji w czasie jej zapisu. W trakcie zapisu, np. zmiennej `speed`, zadajemy wartość z jaką ma się poruszać platforma, a w trakcie odczytu otrzymamy jej rzeczywistą prędkość. Mechanizm ten zilustrowano na rysunku 5.7. Poniżej zaprezentowano kilka przykładowych wywołań.

Przykład:

```
body.x.speed = 300; // zadaj prędkość liniową platformy 300mm/s
yawSpeed = body.yaw.speed; // odczytaj prędkość kątową platformy
...
batteryLevel = body.battery.voltage; // odczytaj poziom baterii
safetyDistance = body.laser.getClosest(-20,20); // sprawdź najkrótszy dystans od przeszkody
// w zakresie kąta widzenia lasera od -20 do 20 st.
```

5.4.5 Kompetencje grupy arm i hand

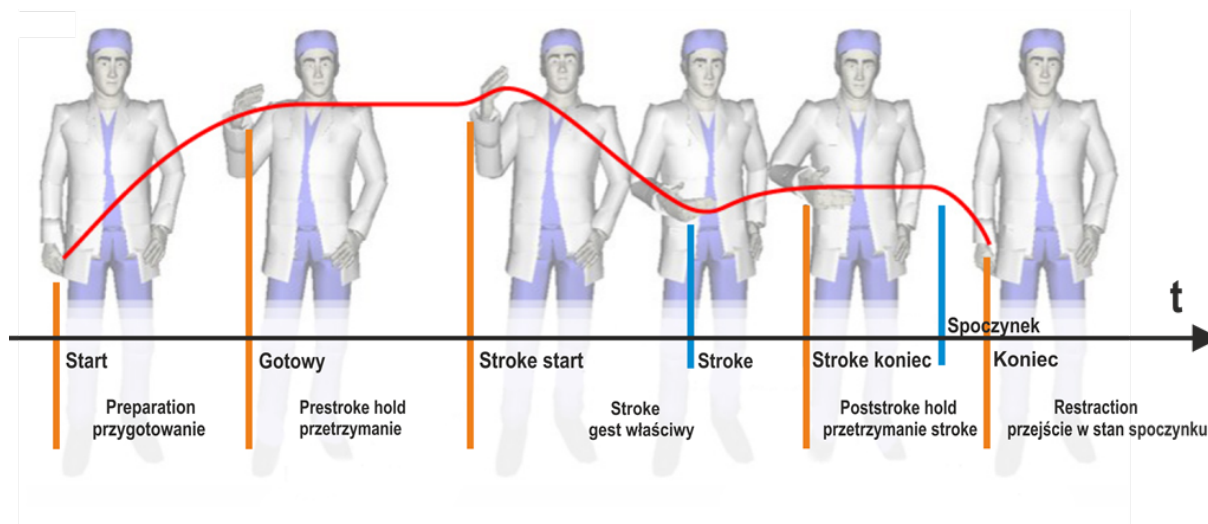
Naukowcy od lat podkreślają, jak znaczącą rolę w procesie komunikacji werbalnej i niwerbalnej mają gesty [140, 172]. Oznacza to konieczność zaimplementowania w robocie społecznym generatora gestów, który by odpowiadał wskazanym w powyższych pracach teoriom. W przypadku robota FLASH ważne było to, aby wykorzystywał możliwości obu ramion robota, a zarazem uwzględniał ich ograniczenia. Pierwsze podejście do sposobu



Rysunek 5.7: System notyfikacji zmiennych platformy mobilnej

sterowania pojedynczym ramieniem podjęto w pracy [188]. Rozważano zastosowanie dostępnego oprogramowania w postaci silnika gestów Greta [104], przeznaczonego dla wirtualnych awatarów. Na szczególną uwagę zasługuje wbudowany w Greta system opisu zachowań BML (*Behavior Markup Language*) oraz parametryzowany interpolator trajektorii. Później zrezygnowano z tego podejścia i zastosowano jednolity mechanizm bazujący na generatorach trajektorii, które dostarcza środowisko Urbi.

Zgodnie z teorią McNeilla [172], każdy gest definiuje się poprzez zbiór punktów synchronizujących, które dzielą cykl gestu na pięć faz. Są to kolejno: przygotowanie, przetrzymanie, gest właściwy, przetrzymanie gestu właściwego oraz przejście w stan spoczynku. Fazy te zaprezentowano na rysunku 5.8.

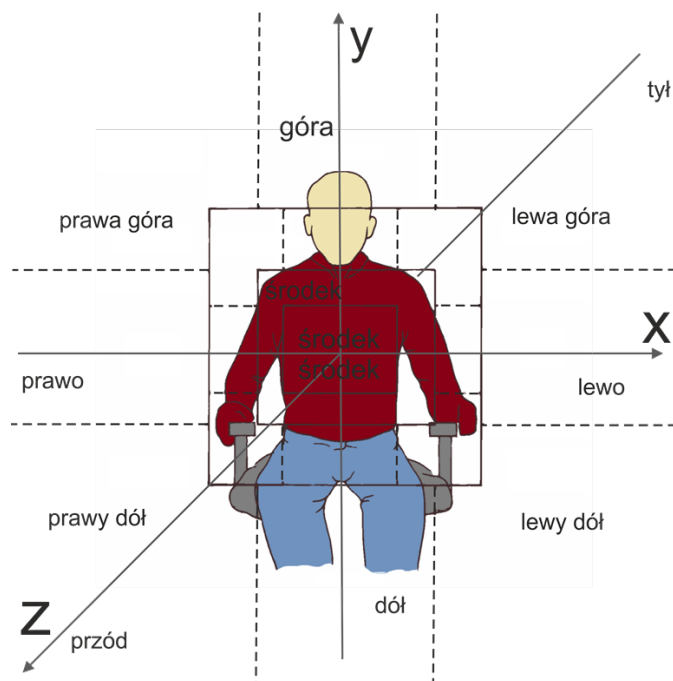


Rysunek 5.8: Fazy gestu

Każdy gest musi obowiązkowo zawierać fazę *Stroke*, która towarzyszy wzmocnieniu wypowiedzianej frazy [172]. Punkty synchronizujące są opisane poprzez kształt i położenie dłoni względem ciała, a także orientację nadgarstka. Do opisu gestów posłużono się przestrzenią zaproponowaną przez McNeilla [172], którą zaprezentowano na rysunku 5.9.

Kształt dłoni może być zdefiniowany przy użyciu hamburskiej notacji nazw symboli języka migowego HamNoSym [204]. Jednakże, w systemie sterowania robotem FLASH przyjęto intuicyjne nazwy kształtów, np. OK, Point, Open, Close, Peace, itd... Orientację nadgarstka (położenie przegubu kulowego) określono przy pomocy dwóch kątów.

Zbiór gestów opisany za pomocą położenia w przestrzeni stanowi bazę, którą można swobodnie wykorzystywać w trakcie mowy, przemieszczania się lub w celu wskazania jakiegoś miejsca. Każdy ruch jest sparametryzowany czasem jego wykonania. Na podstawie tego parametru, generatory trajektorii wysyłają pozycję do każdego napędu. Do genero-



Rysunek 5.9: Obszar gestykulacji

wania ruchów w poszczególnych przegubach zastosowano gotowe generatory trajektorii typu *smooth*, jakie dostarcza środowisko Urbi. Więcej o generatorach można znaleźć w dalszej części pracy oraz w dokumentacji [253].

Wadą przyjętego rozwiązania jest brak płynnego przejścia między fazami gestu. Obecnie, na styku ruchów ramię musi być płynnie zatrzymane i ponownie płynnie rozpędzone (brak interpolacji). Okazało się jednak, że w przypadku gestów rąk, nie stanowi to znaczącego pogorszenia w odbiorze. Większość gestów jakie wykonują ludzie, posiadają fazy zatrzymania. Znacznie gorzej wygląda to w przypadku ruchów głowy. Przykład wywołania niektórych gestów zaprezentowano poniżej.

Przykład:

```

arm.MoveForward(3s);           // ustaw ramiona w przednim obszarze w czasie 3s
arm.UpFront(3s);              // ustaw ramiona w przednim, górnym obszarze w czasie 3s
arm.DownSide(5s);            // ustaw ramiona w bocznym dolnym obszarze w czasie 5s
...
arm.hand.Point(1s);           // ustaw wskazujący kształt dłoni w czasie 1s
arm.hand.Open(maxIntensity,1s); // otwórz maksymalnie dłoń w czasie 1s
arm.hand.FingerClose(fingerNo1,maxIntensity,0.5s); // zamknij maksymalnie pierwszy palec w czasie 0.5s

```

5.4.6 Kompetencje grupy head

Warstwę kompetencji dla głowy zrealizowano posługując się, między innymi, klasyfikacją FACS *Facial Action Coding System* [69, 68]. System FACS był wielokrotnie stosowany w robotyce [22, 198]. W klasyfikacji tej wykorzystuje się pojęcie „jednostki czynnościowej” AU *Action Units*. Jest ona rozumiana jako izolowany i niepodzielny ruch mimiczny możliwy do świadomego wykonania przez człowieka oraz wywołujący obserwowalne zmiany w wyglądzie twarzy. Klasyfikacja FACS opisuje 44 jednostki AU. Wykorzystywana jest często przez psychologów oraz grafików komputerowych, zarówno do generowania jak i do rozpoznawania emocji. Ekman i Friesen w swoich pracach [68, 71] opisują sześć głównych i najczęściej rozpatrywanych emocji, są to: radość, smutek, zaskoczenie, zniesmaczenie,

złość i strach. Ten zestaw podstawowych emocji zaimplementowano i przebadano na wielu robotycznych konstrukcjach [31, 80, 32].

Przedstawioną metodykę opisu wyrazów twarzy zastosowano do robota FLASH, a konkretnie głowy EMYS [136]. Zagadnieniem tym zajmowano się także w pracy [212]. W tym celu każdy AU należy wyrazić przy pomocy ruchomych detali twarzy robota, obu dysków oraz przegubów szyi. Jednym z największych problemów był brak odpowiedniej ilości ruchomych elementów. Ostatecznie, pomimo tak ograniczonych możliwości udało się zdefiniować 16 jednostek czynnościowych. Jak łatwo zauważyć, niektóre AU jak pochylenie lub przekręcenie szyi, a także mrugnięcie okiem, były łatwe do wyrażenia. Inne należało wyrazić w sposób pośredni, np. wysunięcie gałek ocznych na zewnątrz jest odpowiednikiem szeroko otwartych oczu, a podniesienie górnego dysku imituje uniesione brwi.

Zestaw możliwych do wykonania AU zebrano w tabeli 5.1, a ich wygląd zaprezentowano na rysunku 5.10. W tym kontekście, definicje neutralnego wyrazu twarzy i sześciu podstawowych emocji mają postać:

- neutralna – AU(0),
- złość – AU(2+4+43+57),
- zniesmaczenie – AU(4+17+43),
- strach – AU(1+2+17+58),
- radość – AU(1+2+25),
- smutek – AU(1+4+54),
- zaskoczenie – AU(1+2+5+25+58).

Ruch poszczególnych elementów, podobnie jak w przypadku ramion, sparametryzowano czasem. Ponadto, zgodnie z systemem FACS, dodano także intensywność wykonywanych ruchów, którą z zakresu $[0, 1]$ przeskalowano do $[0, 10]$. Ostatecznie, interfejs warstwy kompetencji zapewnia dostęp do głowy zarówno na poziomie pojedynczych AU, jak i kompletnych zachowań ekspresywnych.

Równie ważną kwestią, dotyczącą sterowania głowy było zaimplementowanie systemu imitacji mowy. Najczęściej techniką stosowaną w animacji jest wizualizacja fonemów [225], które są najmniejszą jednostką mowy rozróżnialną dla użytkowników danego języka. Wizualizacji dokonuje się przez charakterystyczne ułożenie ust nazywane visemami [78, 243, 17]. Efektem ich mapowania fonemów jeden do jeden jest uzyskanie 52 visemów. Jednakże, ze względu na niewielkie różnice między nimi, w praktyce stosuje się zredukowaną listę reprezentacji dźwięków [178, 192]. Nie ma zgodności co do tego, jaka jest ich minimalna liczba. Zwykle zależy to od modelu rozważanej postaci (postać animowana, lalka lub robot) oraz języka. Animatorzy Disneya zaproponowali grupę kanoniczną 12 visemów, na podstawie której tworzyli filmy animowane. W synteźatorach mowy najczęściej spotykaną jest zaproponowany przez Microsoft zestaw 21 visemów, który bazuje na disneyowskiej propozycji. Każdy visem opisuje para parametrów: numer ID i czas trwania. W tabeli 5.2 znajduje się wykaz wszystkich 21 znaczników. Na rysunku 5.11 zaprezentowano przykładowe pogrupowanie podobnych visemów

Głowa EMYS nie posiada ust, jedynie dolny dysk pozwala na imitację ruchu szczęki. Podjęto próbę zdefiniowania niewielkiej liczby visemów, zauważalnie różniących się od siebie. Po analizie wybrano trzy, względem neutralnego odchylenia dolnego dysku. Do każdego z nich przypisano, pasujące pod względem wizualnym, kształty ust z grupy 21



Rysunek 5.10: Jednostki czynnościowe AU dla robota EMYS

AU	Ruch	Przegub
AU(1+2)	do góry	górny dysk
AU(4)	do dołu	
AU(43)	zamknij oczy	powieka
AU(45)	mruganie	
AU(46)	„puszczenie” oka	
AU(1)	przekręć na zewnątrz	brew
AU(2)	przekręć do wewnątrz	
AU(5)	wysuń	wysunięcie oczu
AU(17)	do góry	dolny dysk
AU(25)	do dołu	
AU(51)	przekręć w lewo	panoramowanie szyją
AU(52)	przekręć w prawo	
AU(53)	przechyl do tyłu	kiwanie szyją
AU(54)	przechyl do przodu	
AU(57)	przybliż	kiwanie szyją i kiwanie głową
AU(58)	oddal	

Tabela. 5.1: AU dla robota EMYS przypisane do ruchów w poszczególnych przegubach

wisemów (patrz tabela 5.3). Przegub q_8 oznacza położenie dolnego dysku (patrz rysunek 3.14). Na rysunku 5.12 zilustrowano przepływ danych między modułami, jaki towarzyszy mowie robota.





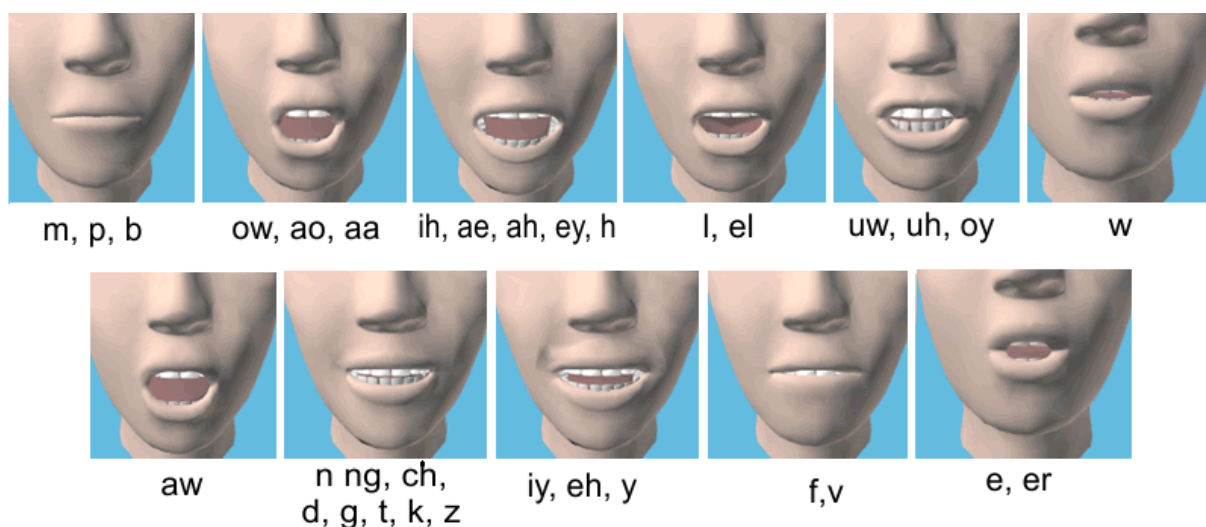
Wizualizacja	Wisemy	Opis
	0, 8	dysk na pozycji neutralnej $q_8 = N$
	2, 3, 5, 9, 13, 14	dysk na pozycji $q_8 = N - 9^\circ$
	1, 4, 7, 11, 12	dysk na pozycji $q_8 = N - 15^\circ$
	6, 10, 15, 16, 17, 18, 19, 10, 21	dysk na pozycji $q_8 = N + 6^\circ$

Tabela. 5.3: Położenie dolnego dysku robota EMYS dla 21 wisemów podzielonych na 4 grupy

Wydawać by się mogło, że w przypadku braku ust zastosowanie systemu wisemów

Visem	Fonem	Visem	Fonem
0	cisza	11	ay
1	ae, ax, ah	12	h
2	aa	13	r
3	ao	14	l
4	ey, eh, uh	15	s, z
5	er	16	sh, ch, jh, zh
6	y, iy, ih, ix	17	th, dh
7	w, uw	18	f, v
8	ow	19	d, t, n
9	aw	20	k, g, ng
10	oy	21	p, b, m

Tabela. 5.2: Wykaz 21 visemów



Rysunek 5.11: Pogrupowane znaczniki visems

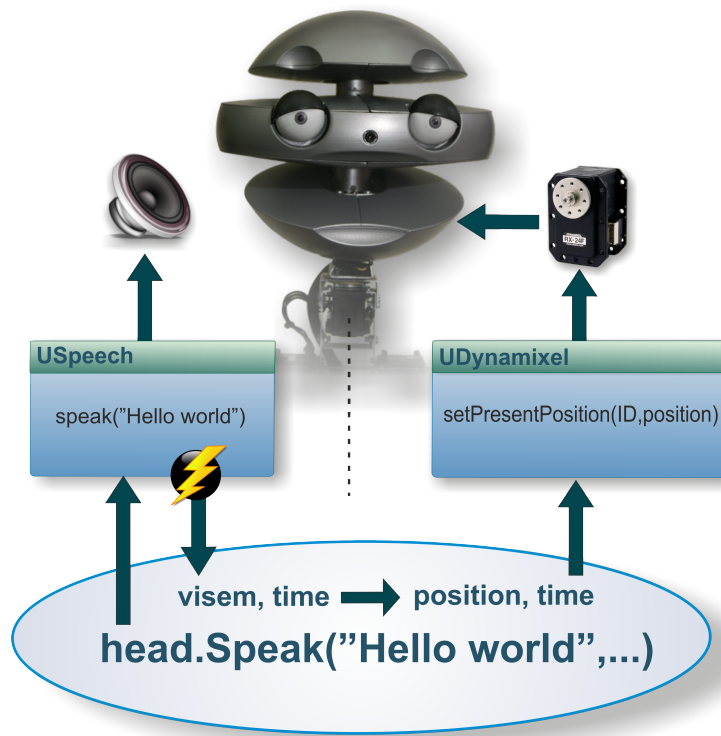
nie ma sensu. Przeprowadzono porównanie powyższego rozwiązania z parametryzowanym amplitudą oraz prędkością, cyklicznym ruchem dysku góra-dół. Okazało się, że pomimo zastosowania tak niewielkiej ilości visemów uzyskano nieporównywalnie lepszy efekt. Synchroniczna imitacja mowy była bardzo dobrze postrzegana, co dowodzi sensowności przyjętego założenia. Poniżej zaprezentowano kilka przykładowych wywołań funkcji kompetencji głowy.

Przykład:

```

head.BehaveJoy(maxIntensity,2s); // okaż radość z maksymalną intensywnością w czasie 2s
head.BehaveSuprise(maxIntensity,3s); // okaż zaskoczenie z maksymalną intensywnością w czasie 3s
...
head.ExpSmile(maxIntensity,2s); // uśmiechnij się z maksymalną intensywnością w czasie 3s
head.ExpAngry(maxIntensity,2s); // zezłość się z maksymalną intensywnością w czasie 2s
...
head.MoveUp(maxIntensity,2s); // spójrz maksymalnie do góry w czasie 2s
head.MoveLeft(maxIntensity,2s); // spójrz maksymalnie w lewo w czasie 2s
...
head.ActBlinking(10s, midIntensity); // uruchom mruganie oczami, ze średnią intensywnością, których
// losowo wyliczany, odstęp między mrugnięciami będzie wynosił
// maksymalnie 10s
head.Stop; // zatrzymaj ruch głowy
...

```



Rysunek 5.12: Uproszczony schemat przepływu danych oraz współpraca pomiędzy modułami mowy

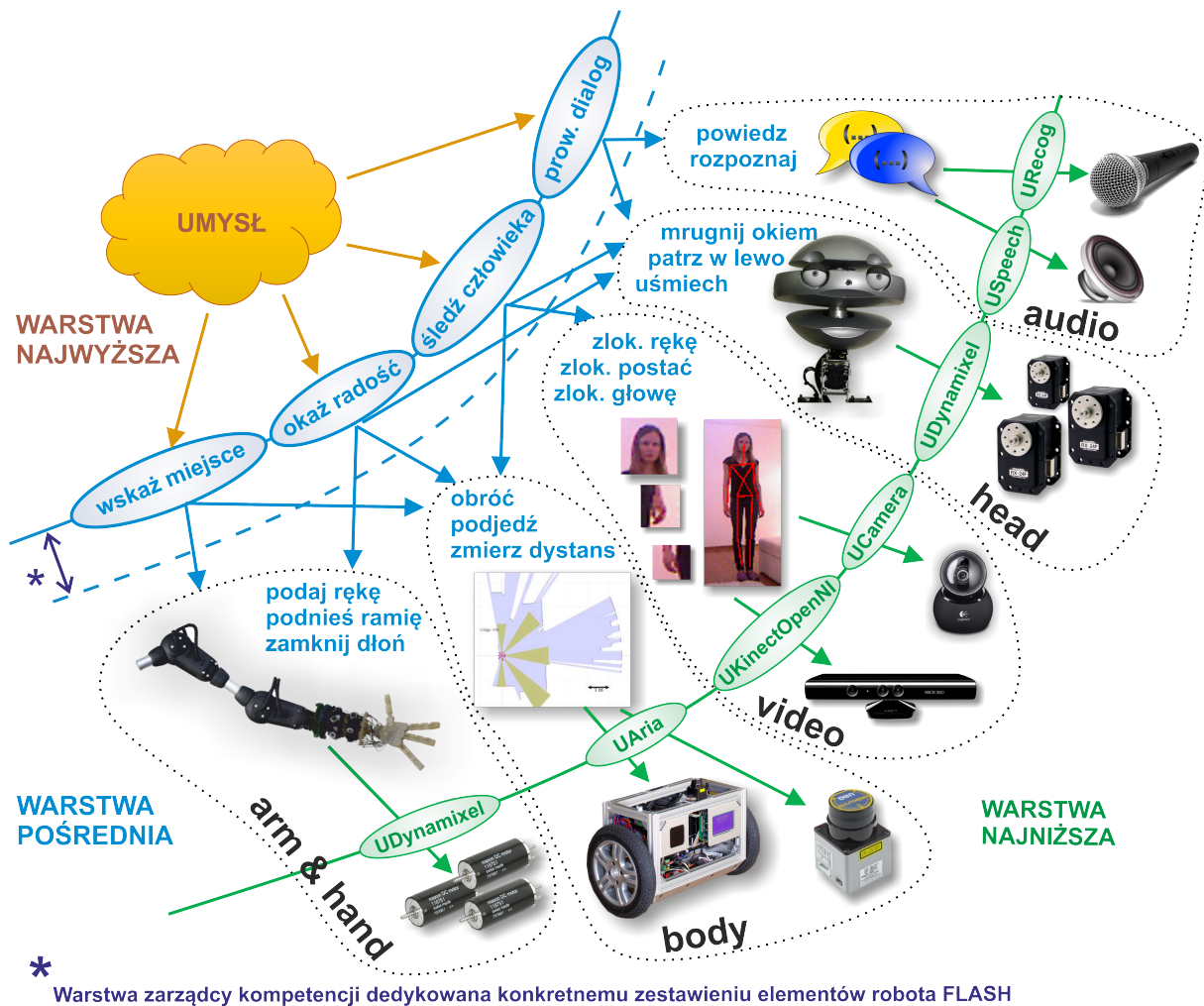
```
head.Say2Sapi("Hello world", maxIntensity); // używając MS SAPI wypowiedz kwestię "Hello world",
// zwizualizuj ją dynamicznym ruchem dolnego dysku
head.Say(fileName, maxIntensity); // wypowiedz kwestię zapisaną w pliku dźwiękowym, zwizualizuj
// ją dynamicznym ruchem dolnego dysku na podstawie visemów
// zapisanych w pliku tekstowym o tej samej nazwie
```

5.5 Warstwa zarządcy kompetencji

Zarządca kompetencji rezyduje na szczycie warstwy pośredniej. Stanowi on łącznik między abstrakcją sprzętu i jej niskopoziomowym systemem sterowania, a umysłem robota. Jego zadaniem jest dobieranie spośród dostępnych kompetencji tych, które są niezbędne do wykonania każdego zadania przewidzianego w scenariuszu eksperymentu. W zaprezentowanym systemie sterowania robotem konfigurację zarządcy należy przeprowadzić dla każdego robota (fizycznego zestawienia). Spowodowane jest to tym, że to samo zadanie może być wykonane przy użyciu różnych kompetencji, np. w trakcie „podążania za człowiekiem”, wykorzystuje się detektor twarzy. Bazuje on na obrazie z kamer RGB, ale jeśli jest dostępny Kinect, to również z detekcji osób na podstawie tego czujnika. Podobnie, wodzenie wzrokiem może być zrealizowane jedynie przy pomocy głowy, ale można włączyć do tego zadania również platformę, jeśli wchodzi w skład fizycznego zestawienia. Na rysunku 5.13 zilustrowano sposób wykorzystania kompetencji do realizacji zadań, zaplanowanych do wykonania przez warstwę umysłu. Zamieszczono także przykładową funkcję zarządcy kompetencji, realizującą wyraz zniesmaczenia.

Przykład:

```
function BehaveDisgust(time) // niech funkcja generująca zachowanie ekspresywne
{ // okazujące „zniesmaczenie” spowoduje
```

Rysunek 5.13: Wykorzystanie składników systemu przez umysł robota.

```

head.Say("I am disgusted", maxIntensity), // wypowiedzenie kwestii
head.BehaveDisgust(maxIntensity,time)& // wygenerowanie głową zniesmaczenia
arm.hand.MoveOpen(time,midIntensity)& // otwarcie dłoni
arm.DownSide(time)& // rozłożenie rąk
body.Turn(time,angle)& // oraz przekręcenie platformy
}

```

Dla robota FLASH przygotowano kilkadziesiąt podobnych konfiguracji. W zależności od eksperymentu, można w krótkim czasie dodać do zarządcy nowe funkcje. Niektóre z nich zaprezentowano poniżej.

Przykład:

```

TurnOpposite(time) // obróć platformę w kierunku, w którym aktualnie jest zwrócona głowa
BehaveComeOn(time) // wykonaj gest przywołania
BehavePoint(time) // wykonaj gest wskazania
...
BehaveGreetings(parametr1, parametr2,...) // wykonaj gest pozdrowienia
BehaveIntroduce() // przedstaw się
...
BehaveAngry(parametr) // okaż złość
BehaveDisgust(parametr) // okaż zniesmaczenie
BehaveSuprise(parametr) // okaż zaskoczenie
...
ActFollowPerson(parametr1, parametr2,...) // podążaj za człowiekiem
ActFollowRightHand(parametr1, parametr2,...) // podążaj za prawą ręką

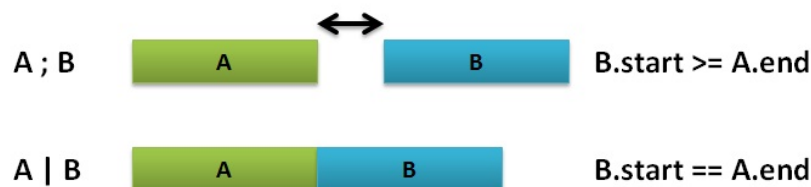
```

5.6 Interfejs programistyczny

Robota programuje się w języku skryptowym urbiscript, ładując instrukcje do silnika Urbi (Urbi Engine Runtime) poprzez dowolną aplikację kliencką. Urbiscript jest to dynamiczny język skryptowy, którego składnia opiera się na znanych językach programowania. Na szczególną uwagę zasługuje wbudowany w Urbi mechanizm szeregowania i zrównoleglania zadań (*orchestrator*). O tym, w jakim czasie i trybie mają być uruchamiane poszczególne zadania decyduje programista używając odpowiednich separatorów. Poniżej zaprezentowano wybrane instrukcje przedstawiające możliwości języka urbiscript. Szczegółowe informacje znajdują się w dokumentacji [253].

5.6.1 Separatory (;) (|)

Separatory te służą do szeregowania zadań. Różnica pomiędzy nimi polega na tym, że w przypadku separatora (|) planista zadań wykona je jedno po drugim. W przypadku separatora (;) pomiędzy tymi zadaniami może nastąpić przełączenie do innego zadania.



W przykładzie poniżej, robot mrugnie najpierw prawym, a potem lewym okiem.

Przykład:

```
head.RBlink() | head.LBlink(); // wykonaj kolejno instrukcje
```

5.6.2 Separatory (,) (&)

Separatory te pozwalają na zrównoleglanie zadań w sposób synchroniczny i asynchroniczny. W przypadku separatora (,) planer uruchamia zadanie w tle po czym przechodzi do kolejnego, drugi separator (&) wskazuje, że obydwa zadania mają się rozpocząć dokładnie w tym samym czasie.



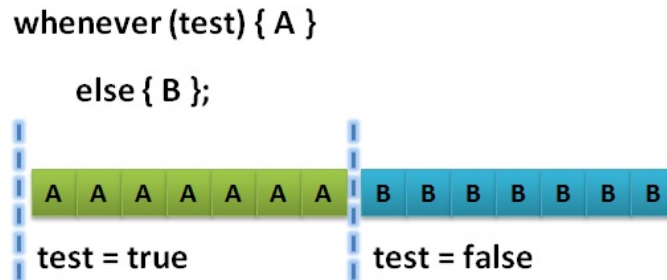
Poniżej zamieszczono przykład, w którym robot mrugnie obojgiem oczu w tym samym czasie.

Przykład:

```
head.RBlink() & head.LBlink(); // obie instrukcje wykonaj w tym samym czasie
```

5.6.3 Pętla warunkowa (whenever)

Pętla (**whenever**) pozwala na zapętlenie zadania w przypadku spełnienia warunku pętli oraz wykonywania innego, w przypadku gdy ten warunek nie jest spełniony



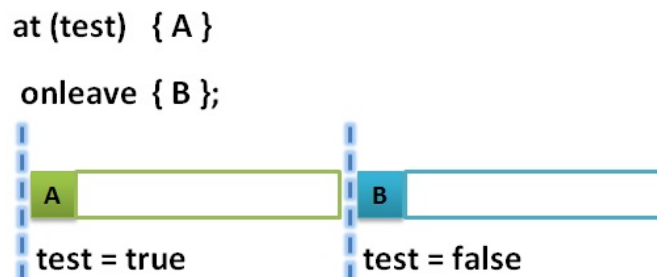
Poniżej zamieszczono przykład, którego efektem będzie mruganie oczami na widok twarzy człowieka, w przeciwnym wypadku robot okaże smutek.

Przykład:

```
whenever(video.objectDetector.visible) {
    head.RBlink() & head.LBlink() | sleep(10s); // mrugnij prawym okiem i czekaj 10s
} else {
    head.BehaveSad(); // okaż smutek
};
```

5.6.4 Wystąpienie (at) i obsługa zdarzeń (Event)

Wystąpienie (**at**) jest mechanizmem synchronizacji bardzo przydatnym w programowaniu robotów. Umożliwia on nie tylko obsługę wystąpień warunkowych, lecz także zdarzeń typu (**Event**). Ponadto, pozwala również na zastosowanie prostej filtracji (eliminowania drgań). Jest ona pomocna w przypadku rzeczywistych pomiarów, w których mogą występować krótkotrwałe niepożądane skoki wartości mierzonej. W poniższym przykładzie pokazano, że wystąpienie **at** zostanie obsłużone dopiero wtedy, gdy detektor obiektów wykryje obecność twarzy w sposób ciągły przez 2 sekundy.



Przykład:

```

at(video.colorDetector.visible ~ 2s) {           // ustaw zdarzenie 1
  head.BehaveJoy();                             // okaż radość
} onleave {
  head.BehaveNeutral();                         // okaż neutralne zachowanie
};

e = Event.new;                                  // zadeklaruj zdarzenie 2
at(e?){                                         // ustaw zdarzenie 2
  head.RBlink();                               // mrugnij okiem
};

e!;                                             // wywołanie zdarzenia 2

```

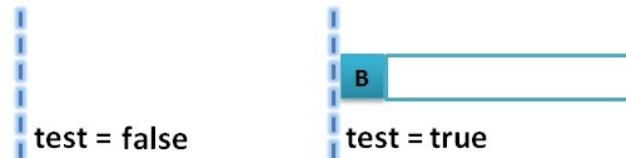
5.6.5 Warunek blokujący (waituntil)

Warunek blokujący pozwala na zatrzymanie wykonywanego kodu do momentu, aż zostanie on spełniony. Poniżej zaprezentowano przykład wykorzystania instrukcji (**waituntil**).

```

waituntil (test) ;
{ B };

```

**Przykład:**

```

waituntil (video.objectDetector.visible);       // czekaj, aż pojawi się twarz człowieka
head.BehaveJoy() & head.RBlink();              // okaż radość i mrugnij prawym okiem

```

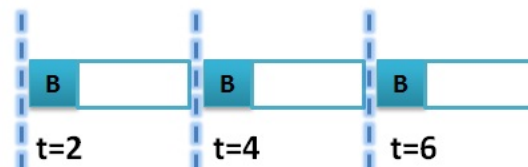
Cykliczne wywołanie (every)

Jest to mechanizm pozwalający na uruchamianie zadań o charakterze cyklicznym. Przykład wywołania mrugnięcia okiem co 2 sekundy zaprezentowano poniżej.

```

every (2s) ;
{ B };

```

**Przykład:**

```

every (2s) head.RBlink(), // mrugaj prawym okiem co 2s

```

5.6.6 Pozostałe instrukcje

Powyżej wymieniono tylko wybrane instrukcje, zwłaszcza te, które nie są spotykane w innych językach programowania. Naturalnie, urbiscript pozwala także na korzystanie z instrukcji znanych, np. z C++ (**for**, **while**, **if**, **loop**,...) oraz umożliwia tworzenie funkcji i klas (**class**, **function**, **slot**,...). Więcej informacji na ten temat można znaleźć w dokumentacji [253].

5.6.7 Etykietowanie kodu (Tag)

Równie istotna z punktu widzenia programowania, jest możliwość stosowania, tzw. etykiet (*Tag*) kodu. Oznacza to, że pewne zadania można grupować i wspólnie nimi zarządzać. Jest to własność przydatna przy implementacji, np. priorytetów zadań. Poniżej zaprezentowano przykład, w którym wywołanie zdarzenia spowoduje wstrzymanie uruchomionej w tle akcji mrugania okiem.

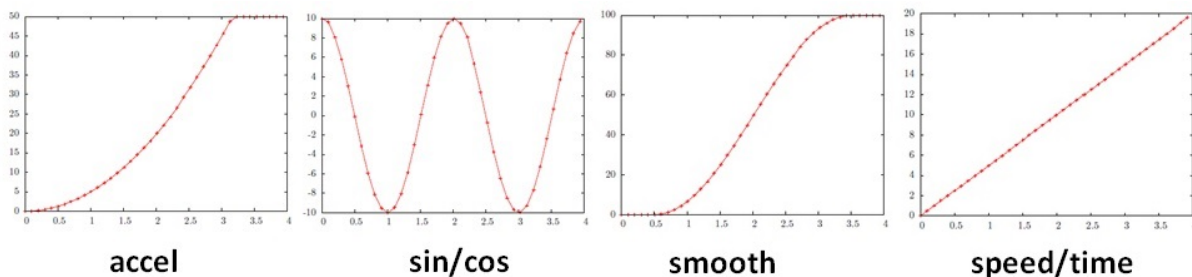
Przykład:

```
tag: every(2s) head.RBlink(),           // mrugaj prawym okiem co 2s

at(video.colorDetector.visible) {
  tag.pause;                             // zatrzymaj mruganie
} onleave {
  tag.resume;                             // wznów mruganie
};
```

5.6.8 Generatory trajektorii

Jest to niezwykle przydatny mechanizm do sterowania napędami robotów. Dzięki niemu możliwe jest uzyskanie płynnych ruchów, których podstawą czasu jest zegar systemowy. Generatorów trajektorii typu *smooth* użyto dla wszystkich napędów robota FLASH. Dostarczają one pozycje odniesienia dla niskopoziomowych sterowników głowy, ramion i dłoni. Poniżej zaprezentowano przebiegi wybranych generatorów oraz przykład użycia jednego z nich.

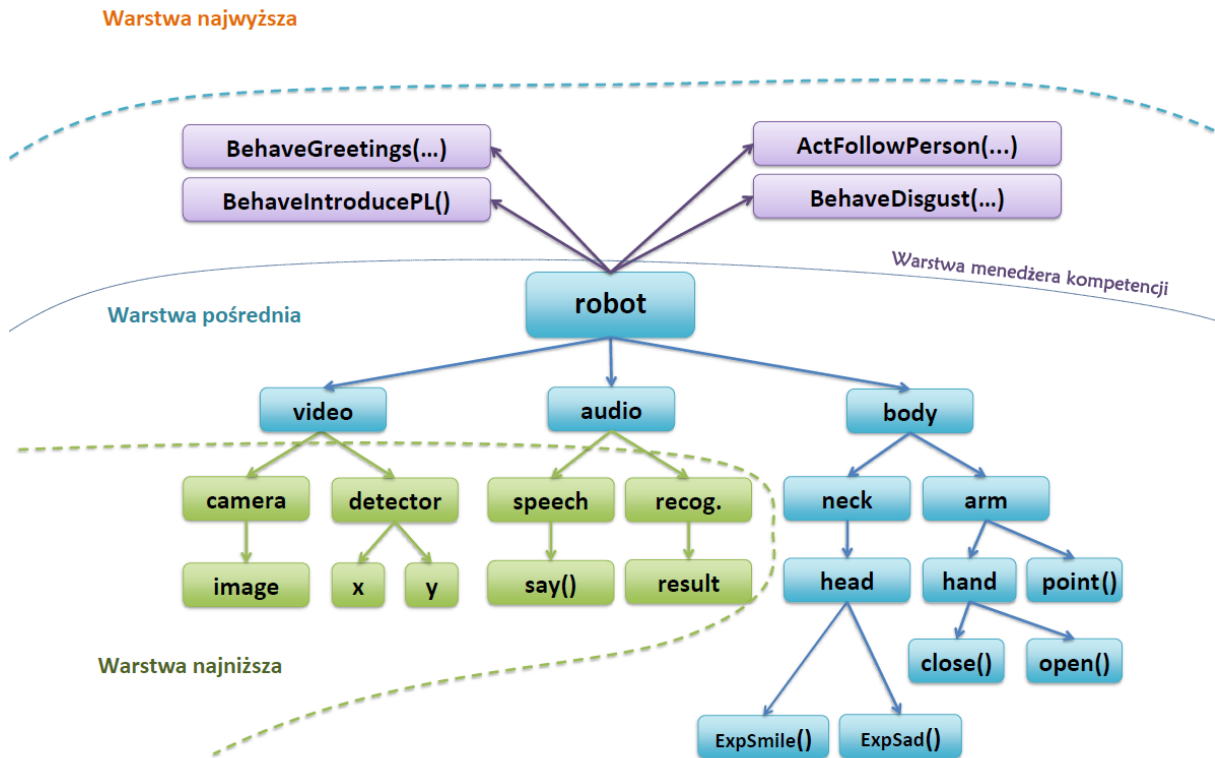


Przykład:

```
head.yaw = 0 sin:3s ampli:20,           // wygeneruj trajektorię sinusoidalną o amplitudzie 20 i okresie 3s
```

5.6.9 Drzewiasta struktura (robot)

Dostęp do poszczególnych kompetencji robota, funkcji i ich parametrów pracy możliwy jest poprzez strukturę **robot**. Na rysunku 5.14 przedstawiono niewielki jej fragment. Wybierając po kropce kolejne gałęzie przechodzi się do wybranych komponentów systemu. Poniżej zaprezentowano przykład poruszania się po strukturze **robot**.



Rysunek 5.14: Struktura funkcjonalna robota

Przykład:

```
robot.body.arm.hand.Open(1s);           // otwórz dłoń
robot.body.neck.head.BehaveJoy(3s);     // okaż głową radość
robot.body.x.speed=100;                 // zadaj prędkość liniową platformy
robot.video.camera.image;               // pobierz obraz z kamery
robot.audio.speechRecognition.result;    // pobierz rozpoznaną kwestię
```

Warto zwrócić uwagę, że interfejs stworzony dla robota FLASH dostarcza identycznych funkcji dla innego fizycznego zestawienia, bez względu na to, z jakich składników (urządzeń) się ono składa. Jeżeli wywołana zostanie funkcja

```
robot.body.neck.head.ExpSmile(3s);      // wygeneruj uśmiech w czasie 3s
```

to niezależnie od tego, jaki rodzaj głowy został użyty w danym fizycznym zestawieniu, w ciągu 3 sekund zostanie wygenerowany uśmiech. Innym przykładem może być wywołanie funkcji

```
robot.body.neck.head.Say("Hello world"); // wypowiedz kwestię
```

W przypadku robota Samuel [37], wypowiedź „Hello world” zostanie zilustrowana ruchem warg napędzanych czterema serwonapędami, a w przypadku robota EMYS jedynie ruchem dolnego dysku. Ponadto struktura *robot* umożliwia korzystanie z tzw. lokalizatorów. Efektem wywołania instrukcji

```
robot.body.arm.MoveUp(5s);              // podnieś ramiona w czasie 5s
```

będzie ruch obu ramion w górę w czasie 5s. Wywołanie instrukcji

```
robot.body.arm[left].MoveUp(5s);        // podnieś lewe ramię w czasie 5s
```

spowoduje podniesienie jedynie lewego ramienia. Dostęp do kompetencji również został zunifikowany. Wywołując instrukcję

```
robot.body.x.speed=300; // zadaj prędkość liniową platformy
```

nastąpi zadanie platformie mobilnej prędkości wzdłużnej równej 300mm/s, niezależnie od tego czy jest to platforma Pioneer czy FLASH oraz czy korzysta one z modułu nawigacji, bazującego na oprogramowaniu ARIA czy ARNL.

5.7 Warstwa najwyższa

W warstwie najwyższej systemu sterowania robota powinien znaleźć się system decyzyjny, z wbudowanym obliczeniowym modelem emocji. O teoriach afektywnego umysłu pisano już w rozdziale 1.1.3. Wskazano tam na ich właściwości oraz dokonano przeglądu dostępnego oprogramowania.

Umysł robota społecznego może być zrealizowany w postaci automatu skończonego, w którym symuluje się i upraszcza pewne mechanizmy, jakie zawiera obliczeniowy model emocji. Pozwala to na realizację prostych scenariuszy na potrzeby krótkich eksperymentów zachowując przy tym autonomiczny charakter pracy robota. Znacznie lepszym podejściem jest zastosowanie zaawansowanego oprogramowania, w którym emocje mają wpływ na podejmowane działania. U podstaw sztucznej osobowości leżą kognitywne teorie emocji oraz teorie poznawcze. Dla prostych scenariuszy, których czas trwania waha się od kilku do kilkunastu minut, konfigurowanie i programowanie tak rozbudowanego systemu może nie mieć sensu, gdyż nie zdąży ono w tak krótkim czasie wykorzystać swoich możliwości. W szczególnych przypadkach dopuszcza się realizację warstwy najwyższej w postaci stanowiska zdalnego sterowania. Problem autonomii robota został omówiony w rozdziale 1.2.3.

5.7.1 Zdalne sterowanie

Do realizacji zdalnego sterowania przygotowano zestaw skryptów wywołujących odpowiednie sekwencje ruchu. Dla wygody operatora stworzono moduł UObject, który zapewnia obsługę dowolnego kontrolera gier (joystick, pad, ballpad, itp.).

Najczęściej z robotem FLASH wykorzystywany był bezprzewodowy kontroler Sony SIXAXIS, który komunikuje się z komputerem pokładowym przy pomocy bezprzewodowego interfejsu bluetooth. Ponadto, przygotowano zestaw interfejsów graficznych, które pozwalają na ręczne sterowanie przy pomocy klawiatury oraz monitorowanie stanu robota. Panele zdalnego sterowania stworzono wykorzystując aplikację Gostai Lab. Przykładowe interfejsy pokazano na rysunku 5.15.

5.7.2 Automat skończony

Większość eksperymentów krótkoterminowych przeprowadzono przy użyciu oprogramowania Gostai Studio. Aplikacja ta jest dedykowana dla systemu Urbi i przeznaczona do implementacji skończonych automatów. Jej graficzny interfejs jest estetyczny, intuicyjny i łatwy w obsłudze. Tworzy się w nim stany, podstany oraz warunki przejścia pomiędzy nimi. Pozwala relatywnie szybko zaimplementować rozbudowane scenariusze, które nie wymagają symulacji długoterminowych procesów, jakie zachodzą w ludzkim umyśle. Nie mniej jednak, pewne mechanizmy umysłu są możliwe do zaimplementowania.

Uwzględnienie w automacie przestrzeni PAD (Pleasure, Arousal, Dominance) [20] czy ASV (Arousal, Valence, Stance) [34, 177] może mieć wpływ na sposób poruszania się robota, wybór zachowań ekspresywnych, a nawet na warunki przejścia. Na rysunku 5.16



Rysunek 5.15: Interfejsy graficzne dla robota FLASH oraz kontroler gier Sony SIXAXIS

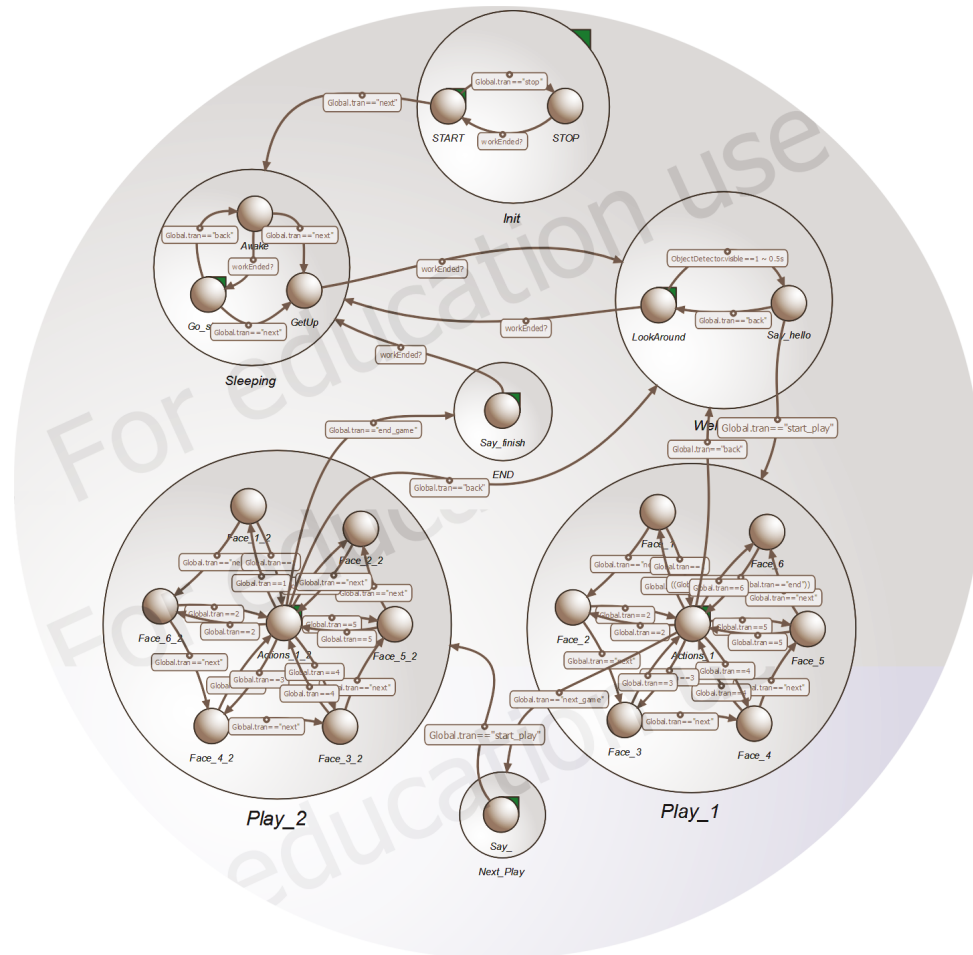
zaprezentowano przykładowy scenariusz eksperymentu zrealizowany z wykorzystaniem Gostai Studio.

5.7.3 Zaawansowane oprogramowanie

Jednym z ciekawszych projektów, dotyczących prac nad sztucznym umysłem, jest rozwijany przez Portugalski Instituto de Engenharia de Sistemas e Computadores - Investigaço e Desenvolvimento system FAtiMA (FearNot! Affective Mind Architecture) [60, 62, 14]. System ten został opracowany w celu stworzenia wiarygodnych syntetycznych postaci, w których emocje i osobowość są spójne. Postacie te muszą być zdolne do wyrażania emocji, a ich stan emocjonalny musi być odzwierciedlony w podejmowanych działaniach [61]. Koncepcja systemu opiera się na teorii oceny Orthony'ego, Clore'a Collinsa (OCC) [191]. Schemat architektury przedstawiono na rysunku 5.17. Wyróżnia się w nim dwa bloki generowania działań, są to warstwa reaktywna i warstwa deliberatywna.

W warstwie reaktywnej działania (akcje) nie podlegają procesom planowania, są podejmowane natychmiastowo na podstawie percepcji otoczenia i aktualnego stanu emocjonalnego. Warstwa ta posiada podsystem oceny, który po uwzględnieniu progów wpływa na aktualny stan emocjonalny. Ocena zdarzeń w warstwie reaktywnej następuje na podstawie predefiniowanych emocjonalnych reguł reakcji. Reguły te reprezentują zmienne ze zbioru oceny OCC, takie jak spełnienie oczekiwań (*events desirability*), spełnienie oczekiwań innych (*desirability for other*) i chwalebność działania (*praiseworthiness of the action*). Na podstawie tych zmiennych jest generowany szeroki zakres emocji, od radości do gniewu. Ponadto, w przypadku zgodności oceny potwierdzającej regułę zostanie podjęta akcja.

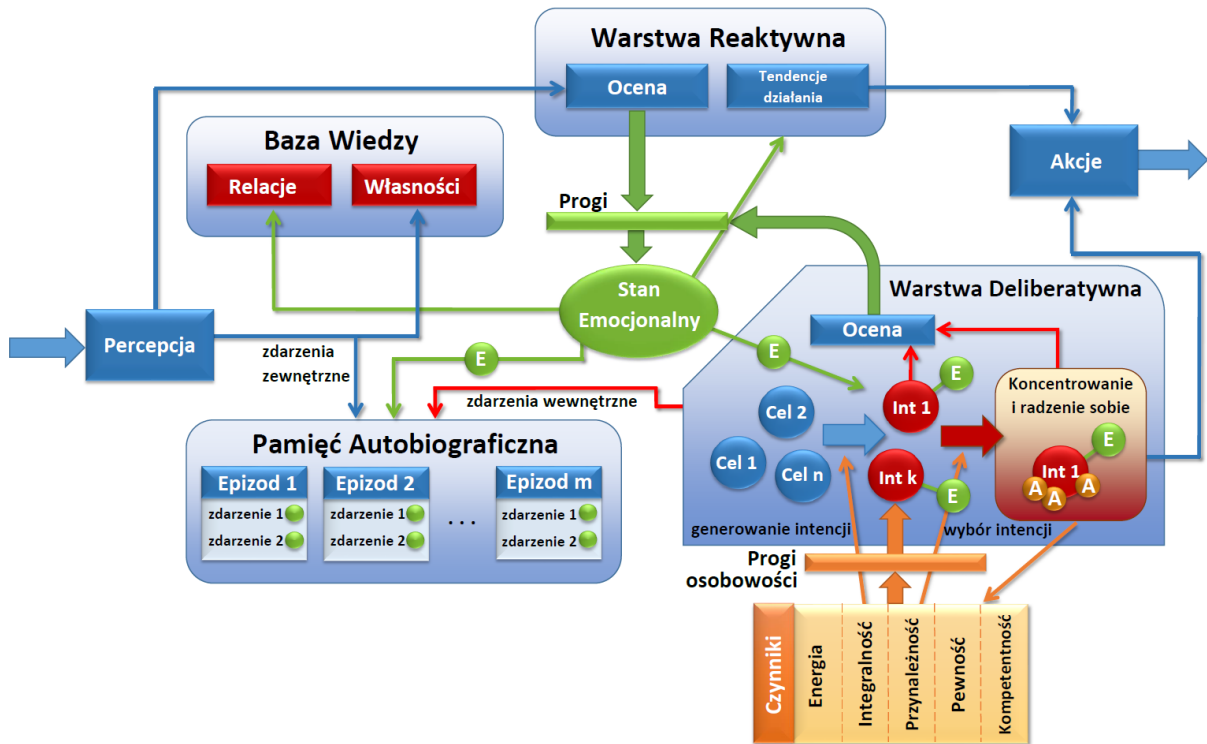
Warstwa deliberatywna obejmuje złożony i długotrwały proces rozumowania, który jest zorientowany na przyjęte cele i dostarcza planu działań. W trakcie napływających danych z systemów percepcji agent podejmuje cel, który staje się intencją (celem do realizacji). Emocje są generowane z uwzględnieniem obecnego stanu intencji, a raczej ich



Rysunek 5.16: Widok przykładowego automatu zaimplementowanego z użyciem oprogramowania Gostai Studio

prawdopodobieństwa sukcesu lub faktu osiągnięcia celu. Emocje, takie jak nadzieja i strach powstają na podstawie istotności wybranego celu. Intencja, która wywołuje najsilniejsze emocje, jest wybierana jako cel do realizacji. Dla wybranych intencji tworzony jest optymalny plan realizacji celu. Ponadto, agent posiada także tzw. cele własnego interesu, które opisują pewne czynniki, nazywane też potrzebami (*drives*). Są one zaczerpnięte z teorii modelowania emocji PSI i można je podzielić na trzy grupy:

- **Czynniki fizjologiczne** (energia, fizyczna integralność) – pierwszy czynnik reprezentuje potrzebę zdobywania niezbędnej do działania energii. Zwykle odpowiada odwrotnie proporcjonalnie do poziomu energii zgromadzonej w akumulatorach. Czynnik ten z czasem się zwiększa. Fizyczna integralność to potrzeba utrzymywania robota w dobrej kondycji. Czynnik ten może się zwiększyć, gdy robot uszkodzi sobie jakiś komponent.
- **Czynniki społeczne** (społeczna przynależność) – jest to czynnik, który podtrzymuje potrzebę nawiązywania kontaktów społecznych. Reprezentuje poczucie akceptacji przez społeczeństwo. Wzrasta, gdy agent odbiera sygnały, o jego wykluczeniu ze społeczeństwa.
- **Czynniki kognitywne** (pewność/niepewność, kompetentność) – pierwszy czynnik reprezentuje zdolność do przewidywania efektów podejmowanych celów. Wzrasta,



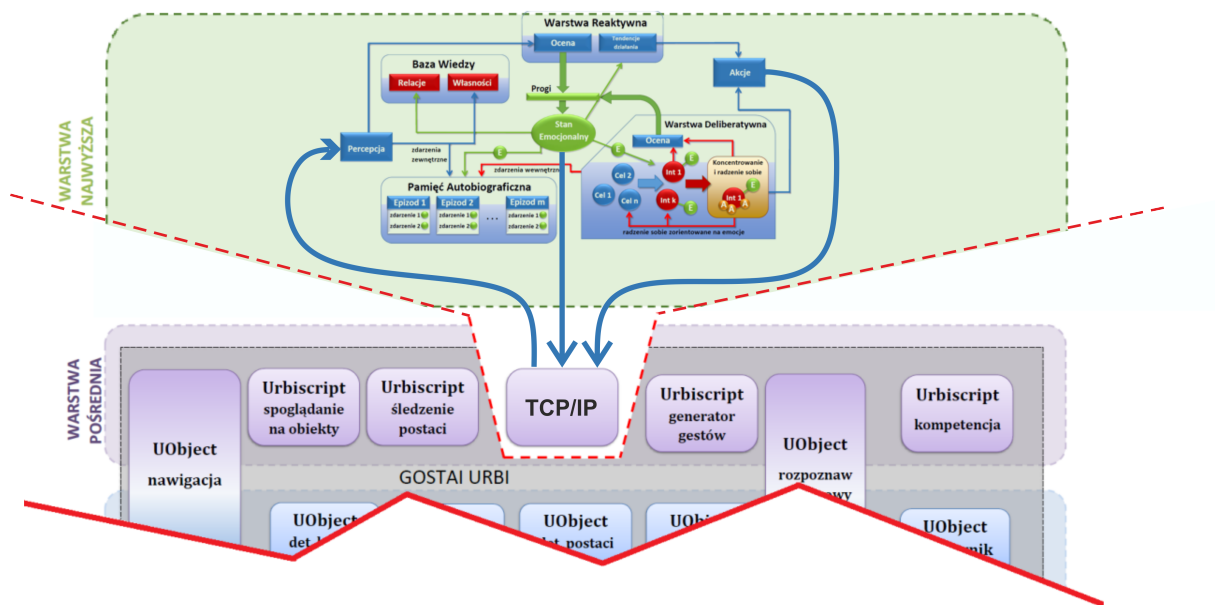
Rysunek 5.17: Schemat architektury FATiMA, stworzony na podstawie pracy [14]

gdy agent zdobywa wiedzę na temat środowiska. Gdy efekty działań nie są zgodne z przewidywaniami agenta, wzrasta poziom jego frustracji. Kompetentność odpowiada przewidywanym zdolnością do wykonywania zadań, w tym własnych celów. Czynnikiem ten wzrasta, gdy wykonywane zadania kończą się powodzeniem, maleje w przeciwnym wypadku.

Jeżeli aktywne cele stoją w sprzeczności z celami własnego interesu tworzy się tzw. zagrożenie, które wywołuje strach. Wybrany cel jest rezultatem dwóch typów procesów radzenia sobie (*coping*). Pierwszy jest zorientowany na planowanie i realizację działań, drugi na zmianę emocji. Ostateczny wybór działania (akcji) będzie zależał od założonej osobowości agenta.

Postrzegane zdarzenia zewnętrzne aktualizują bazę wiedzy i podlegają one ocenie przez obie warstwy, czego efektem jest zmiana stanu emocjonalnego. Zdarzenia oraz towarzyszące im emocje są zapamiętywane w pamięci autobiograficznej.

W celu przyłączenia powyższego oprogramowania FATiMA do systemu sterowania robota FLASH, który rezyduje na dwóch niższych warstwach architektury, stworzono dedykowany zestaw skryptów. Na poziomie kompetencji zapewnia on realizację (tłumaczenie) zadań oraz stanu emocjonalnego otrzymanych z warstwy umysłu. Pewnych modyfikacji, dotyczących realizacji połączenia sieciowego oraz implementacji protokołu komunikacji, dokonano również w oprogramowaniu FATiMA (Rysunek 5.18). Ostatecznie, zrealizowano krótkie demo, w którym robot wykonał kilka prostych zadań, mających na celu odnalezienia wskazanej osoby. Obecnie trwają prace nad dalszym rozwojem integracji oprogramowania FATiMA z systemem sterowania robota FLASH. Pozwoli to na wykorzystanie robota FLASH w długoterminowych badaniach z dziedziny HRI.



Rysunek 5.18: Schemat przyłączenia oprogramowania FATiMA do systemu Urbi

Rozdział 6

Weryfikacja eksperymentalna

Jednym z wielu zadań projektu LIREC było opracowanie i skonstruowanie prototypu stanowiska badawczego w celu eksperymentalnej weryfikacji wytworzonych w projekcie technologii. W czasie opracowywania poszczególnych elementów robota, z częścią z nich przeprowadzono eksperymenty w małej skali, które wskazały na ich wady konstrukcyjne i zmusiły do przeprojektowania konstrukcji. Platforma balansująca Cosmos [133], ramię oraz dłoń [54] zainstalowane na platformie PeopleBot [2], a także głowy Romek i Samuel [37]) przyczyniły się do powstania ostatecznych wersji elementów składowych robota FLASH. Na szczególną uwagę zasługuje oryginalna głowa typu EMYS [136], która może służyć jako samodzielne stanowisko badawcze, w postaci stołowego robota osobistego. Głowa EMYS została wykonana w kilku egzemplarzach i jest obecnie wykorzystywana w wielu eksperymentach prowadzonych w Polsce oraz przez zagranicznych partnerów projektu LIREC [136, 212, 213, 214, 58]. Są to badania głównie z obszaru HRI, ale także dotyczą własności konstrukcji i jej eksploatacji oraz weryfikują opracowaną w projekcie architekturę sterowania. Ponadto, poszukuje się nowych metod tworzenia zachowań ekspresywnych. Robot FLASH, został ukończony znacznie później niż głowa, gdyż dopiero pod koniec projektu (czerwiec 2012). Dotąd udało się z robotem przeprowadzić, dwa większe eksperymenty [13]. Ponadto, przygotowano dla robota FLASH wiele krótkich scenariuszy demonstracyjnych, które są wykorzystywane w trakcie pokazów, targów oraz imprez akademickich.

Poniżej opisano dwa eksperymenty, w których pierwszy wykonano z udziałem samej głowy EMYS, a drugi z kompletnym robotem FLASH. Oba eksperymenty stanowiły wyzwanie dla systemu sterowania robota i umożliwiły praktyczną weryfikację jego działania.

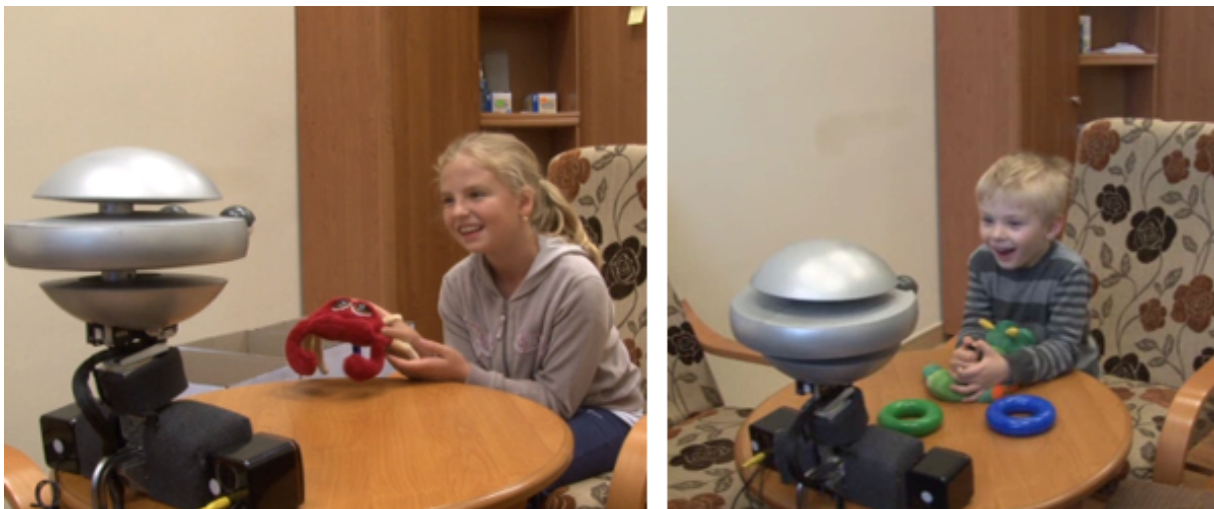
6.1 Eksperyment HRI z głową EMYS

Celem eksperymentu było zbadanie poziomu zaangażowania uczestnika w interakcję z robotem EMYS oraz zweryfikowanie poprawności rozpoznawania wyrażanych przez robota emocji. Eksperyment właściwy poprzedzono dwoma badaniami pilotażowymi, których celem było precyzyjne przygotowanie scenariusza, dzięki któremu robot uzyska maksymalne zaangażowanie badanego oraz czy, badany będzie wykazywał odpowiedni stopień rozpoznawalności zachowań robota. Badania zrealizowano w zespole pięcioosobowym przy współpracy z grupą psychologów z Otto-Friedrich-Universität w Bambergu. Szczegółowa analiza danych eksperymentalnych znajduje się w pracy [136]. Poniżej zamieszczono wybrane wyniki.

6.1.1 Opis eksperymentu

Eksperyment przeprowadzono w szkole podstawowej w okolicy Wrocławia, a do udziału w nim zaproszono dzieci w wieku 8-12 lat. Wybór ten był podyktowany kilkoma względami. Dzieci mają nieco niższy stopień rozpoznawalności emocji niż dorośli, co pozwoli na wnioskowanie, że jeśli dzieci postrzegają zachowanie robota jednoznacznie, to dorośli tym bardziej nie powinni mieć z tym problemu. Po drugie, wiadomo że dzieci w różnym wieku posiadają zróżnicowany poziom rozwoju funkcji poznawczych – dotyczy to również zdolności rozpoznawania emocji [98]. Jeżeli udałoby się to potwierdzić dla ekspresywnej głowy robota, stanowiłoby to uzasadnienie przyjętej metody badawczej. Po trzecie dzieci mają tendencje do ukrywania lub maskowania wyrażanych emocji w czasie interakcji. Zjawisko to może być zweryfikowane w przypadku interakcji z robotem.

W eksperymentach wzięła udział grupa dzieci, w skład której weszło 18 chłopców i 27 dziewczynek o średniej wieku 9,9 roku. Przez cały okres eksperymentu robot pracował w pełni autonomicznie. Scenariusz został podzielony na dwie gry. W pierwszej części (gra w przedrzeźnianie), robot pokazywał kolejno sześć podstawowych emocji (*złość, radość, zaskoczenie, strach, wstręt i smutek*). Następnie, po każdej z nich zachęcał do jej powtórzenia. W drugiej części scenariusza (gra w dopasowanie zabawki), robot ponownie pokazywał sześć min (w innej kolejności), ale tym razem po każdej prosił o wyciągnięcie kolorowej maskotki z podpisanego pudełka (*wesoły, smutny, zły, nie pasuje*). Zatem, trzy miny miały swój odpowiednik w zabawkach, a dla trzech pozostałych należało pokazać zabawkę z pudełka *nie pasuje*. Nazwę emocji *radość* celowo zmieniono na *wesoły*, aby ułatwić najmłodszym dzieciom jej identyfikację. Przebieg eksperymentu pokazano na rysunku 6.1.



Rysunek 6.1: Przebieg eksperymentu z głową EMYS

W trakcie eksperymentu, dzięki detektorowi twarzy (obiektów), robot cały czas utrzymywał kontakt wzrokowy z uczestnikiem. EMYS odróżniał zabawki na podstawie ich koloru, a w czasie rozpoznawania przyglądał się każdej z nich. W każdym pudełku znajdowały się zabawki jednakowego koloru. Po zaprezentowaniu zabawki, robot potwierdzał wybór i prosił o jej odłożenie na miejsce. Po każdej sesji dziecko oglądało nagranie wideo z przebiegu eksperymentu i jeszcze raz nazywało każdą minę. Jedna sesja trwała 5-8 minut, w czasie jej trwania był obecny jeden lub dwóch prowadzących. Badania odbywały się w pomieszczeniu szkolnego pedagoga, które większość dzieci dobrze znała i nie odczuwała w nim lęku.

6.1.2 Analiza danych

Przebieg interakcji z każdym dzieckiem został nagrany przy pomocy dwóch kamer wideo. Jedna z nich była skierowana na robota, a druga na uczestnika badania. W dniu poprzedzającym eksperyment, dzieci oceniały emocje wyrażane przez człowieka na podstawie 12 zdjęć zaczerpniętych z pracy [68] oraz wypełniły kwestionariusz „wielkiej piątki” (pięcioczynnikowy model osobowości) [161]. Teoria „wielkiej piątki” służy do pomiaru i opisu osobowości człowieka. Opracowane na jej podstawie kwestionariusze pozwalają, za pomocą zaledwie kilkunastu pytań, zdiagnozować u badanego cechy charakterystyczne dla danej osobowości. Pięcioczynnikowy model osobowości obejmuje: neurotyczność, ekstrawersję, otwartość, ugodowość oraz sumienność. Tuż po eksperymencie przeprowadzono z nimi wywiad. Na podstawie nagrania wideo oceniano zaangażowanie dziecka w interakcję z robotem. Zwracano uwagę na postawę ciała, liczbę werbalnych i niewerbalnych kontaktów, wyrażane emocje, kierunek spojrzeń oraz orientację ciała dziecka w kierunku robota i prowadzącego. Oceny tych danych dokonano z udziałem trzech sędziów kompetentnych [97]. Dodatkowo współczynniki danych na temat zaangażowania badanego w eksperyment uzyskiwano na podstawie silnego zaangażowania, a nie na podstawie jego „dobrego” samopoczucia. Negatywne wtedy, kiedy badany był rozkojarzony i spoglądał na prowadzącego lub w kierunku kamer. Ostatecznie zebrano kolejno pięć typów danych:

- kwestionariusz osobowości („wielka piątka”),
- ocena zdolności rozpoznawania emocji,
- nagranie wideo,
- ocena emocji robota w czasie eksperymentu,
- wywiad pogłębiony.

6.1.3 Wyniki

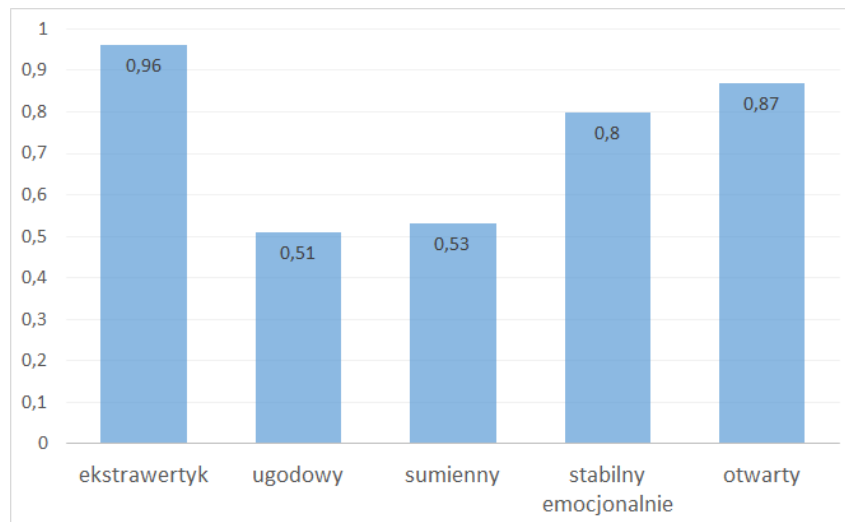
Poniżej zamieszczono wybrane wyniki badań, które przedstawiono w formie pytań i odpowiedzi.

Jak badany postrzega robota w czasie interakcji?

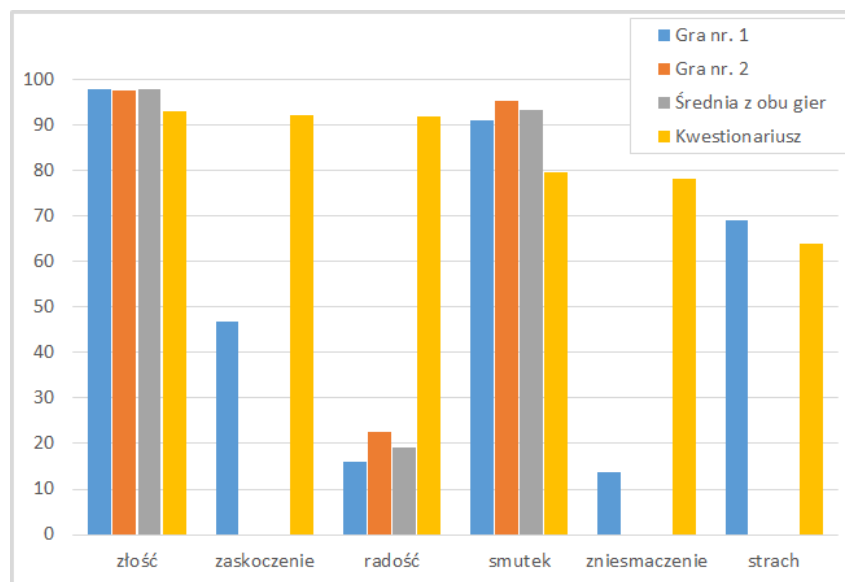
Bez względu na płeć, niemal wszyscy uczestnicy twierdzili, że robot jest płci męskiej, gdyż na to wskazywał jego głos. 96% uczestników zadeklarowało chęć ponownego spotkania się z robotem. Jedynie 33% odpowiedziało, że robot posiada emocje, a 21%, że ich nie ma. Pozostali nie byli pewni swojej odpowiedzi. Dzieci zostały poproszone, aby scharakteryzowały osobowość robota zgodnie z [46]. Na wykresie 6.2 zamieszczono wyniki. Znakomita większość badanych twierdzi, że robot posiada osobowość ekstrawertyka, jest otwarty oraz stabilny emocjonalnie.

Jak badany rozpoznaje i rozróżnia emocje robota?

W czasie eksperymentu robot odgrywał dwie gry, pokazując przy tym sześć emocji. Na wykresie 6.3 zamieszczono wyniki rozpoznawalności poszczególnych emocji w czasie gry pierwszej, drugiej, oraz na podstawie kwestionariusza. Należy zwrócić uwagę, że w drugiej grze *zaskoczenie*, *wstręt* i *strach* przypisane do zabawki *nie pasuje*.



Rysunek 6.2: Postrzeganie osobowości robota EMYS (od -1 = „zupełnie nie” do 1 = „zupełnie”)



Rysunek 6.3: Rozpoznawanie emocji

W przypadku emocji zaczerpniętych z bazy Ekmana, poziom ich rozpoznawalności był dość wysoki. Dzieci nieco gorzej radziły sobie z miną *strach*. Natomiast w przypadku emocji *złość* osiągnięto wysoki współczynnik rozpoznawalności. Można zatem przyjąć, że dane te są miarodajne. Literatura podaje, że dzieci najlepiej rozpoznają emocje *wesoły*, *zły* i *smutny* [98]. W przypadku robota EMYS poziom rozpoznania emocji *wesoły* był bardzo niski. Prawdopodobnie ma to związek z brakiem uniesionych kątek ust [48]. Mina *zniechęcenie* jest dość złożona i trudna w interpretacji. Miało to również odzwierciedlenie w przypadku przeprowadzonych badań z robotem. Ponadto, EMYS nie posiada nosa, który zwykle przy wyrażaniu tej emocji marszczy się [68]. Mina *smutek* była lepiej rozpoznawana kiedy wykonywał ją robot, niż w mina człowieka zamieszczona w kwestionariuszu. Można przypuszczać, że wynika to z tego, że robot posiada zabawny kształt i potrafi efektownie opuścić powieki. *Strach* był rozpoznany tylko trochę gorzej niż w przypadku emocji wyrażanych przez człowieka. Należy jednak pamiętać, że to była najsłabiej rozpoznawana mina także na podstawie kwestionariusza.

Jakie czynniki wpływają na poziom rozpoznawania emocji?

Na podstawie analizy wyników nie zaobserwowano, żeby płeć dziecka wpływała na poziom rozpoznawalności emocji robota. Z całą pewnością miał za to wpływ wiek uczestników. Im starsze dzieci, tym lepiej rozpoznawały emocje EMYSa. Co ciekawe, nie zaobserwowano tego zjawiska w przypadku kwestionariusza, ale trzeba podkreślić że poziom rozpoznawalności emocji był ogólnie wysoki. Nie dostrzeżono także wpływu stopnia zaangażowania w zabawę z robotem na trafność rozpoznanych emocji. Można przypuszczać, że trafność rozpoznawania zależy głównie od wyglądu robota. Osobowość badanego miała znaczenie jedynie u dzieci o osobowości ugodowej oraz stabilnych emocjonalnie. W przypadku pierwszej grupy, dzieci były bardziej skupione na sytuacji związanej z eksperymentem i trudniej im było się skupić na emocjach robota. Uczestnicy o osobowości stabilniej emocjonalnie poprawniej rozpoznawali emocje EMYSa. Być może wynika to ze zwiększonej wrażliwości dzieci, przez którą były bardziej podatne na ekspresję robota. Istotne było również to, że w czasie zabawy brakowało kontekstu wyrażanych emocji, co zdecydowanie utrudniało rozpoznawanie.

Czy zainteresowanie dzieci robotyką ma wpływ na interakcję i sposób postrzegania EMYSa?

Zaobserwowano pewien związek pomiędzy zainteresowaniem dzieci robotyką (robotami), a pozytywnym zaangażowaniem w zabawę z robotem. Podobną korelację uzyskano pomiędzy pytaniem „*Czy dobrze się bawiłeś/aś z robotem?*”, a ogólnym zaangażowaniem w zabawę z robotem. Uczestnicy eksperymentu interesujący się robotami byli bardziej zaintrygowani tym, co robi EMYS oraz wykazywali zwiększoną aktywność. Negatywną korelację, potwierdzającą powyższą obserwację, uzyskano między zadowoleniem z zabawy, a spoglądaniem na prowadzącego.

Czy istnieje wpływ osobowości dziecka na interakcję i sposób postrzegania robota?

Dzieci posiadające osobowość neurotyczną częściej spoglądały w kamerę. Mogło to być spowodowane tym, że denerwowały się i czuły niepewnie w czasie badań. Uczestnicy o otwartej i ugodowej osobowości wykazywali większe zadowolenie z udziału w eksperymencie oraz częściej deklarowali chęć ponownego spotkania robota. Dzieci otwarte chętniej udzielały odpowiedzi na pytania, co wydaje się być oczywiste. Co najważniejsze, osobowość dziecka nie wpływała znacząco na poziom zaangażowania w interakcję z robotem.

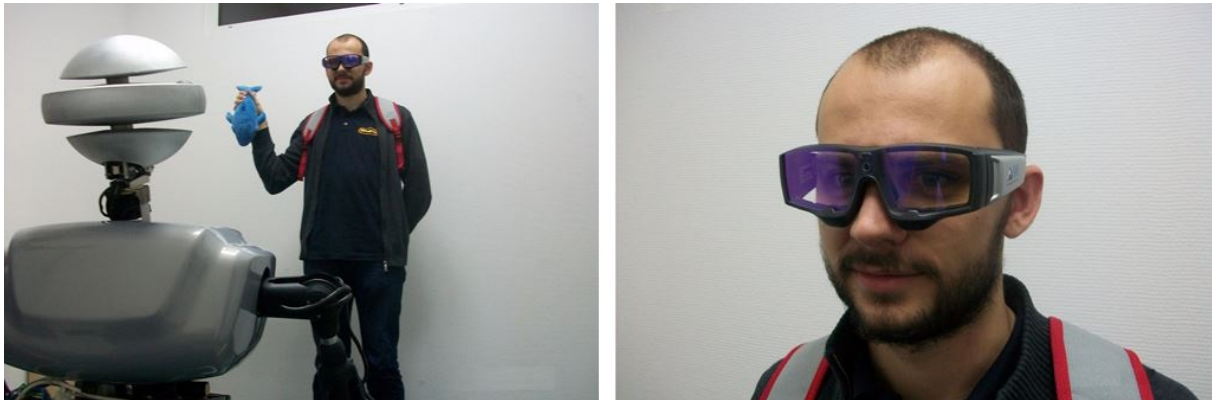
6.2 Eksperyment HRI z robotem FLASH

Celem przeprowadzonego eksperymentu była ocena, w jakim stopniu wygląd oraz natężenie reakcji emocjonalnych robota FLASH rzutują na sposób postrzegania go przez ludzi mających z nim kontakt. Badano, między innymi, utrzymywanie kontaktu wzrokowego, który ma szczególne znaczenie w procesie interakcji [103]. Eksperyment właściwy był poprzedzony serią badań pilotażowych, których celem było dopracowanie scenariusza zachowań robota oraz zapoznanie się zespołu badawczego z aparaturą pomiarową. Do badań zastosowano mobilny okulograf (eyetracker), dzięki któremu uzyskano klasyczne mapy ciepłe oraz wyznaczono i zbadano obszary zainteresowań. Badania zrealizowano w zespole pięcioosobowym, przy współpracy z grupą psychologów z korporacji Millward

Brown SMG KRC. Szczegółowa analiza danych znajduje się w raporcie [85] oraz pracy [67]. Poniżej opisano wybrane wyniki eksperymentu.

6.2.1 Opis eksperymentu

Eksperyment został przeprowadzony na Dworcu Głównym we Wrocławiu. Wybór lokalizacji był podyktowany koniecznością pozyskania do udziału w badaniu dużej liczby osób, o zróżnicowanym profilu demograficznym i społecznym. W eksperymencie wzięły udział osoby między 15 a 82 rokiem życia, przy czym średnia wieku respondentów wyniosła 26,9 lat. Przebadano 90 uczestników w czasie 4-6 minutowej interakcji z robotem, a także przeprowadzono 113 ankiet oraz wywiadów pogłębionych. Różnica wynika z tego, że niektóre nagrania uzyskane przy pomocy z okulografu musiały być odrzucone z powodu błędnej pracy urządzenia pomiarowego.



Rysunek 6.4: Przebieg eksperymentu z robotem FLASH

Badanie przeprowadzone z każdym respondentem składało się z dwóch części. W części pierwszej (sesja w okularach do Eye Trackingu) osoby uczestniczące w eksperymencie, miały bezpośredni kontakt z robotem. Zastosowano tu aparaturę firmy SensoMotoric Instruments [229] wraz z oprogramowaniem do analizy nagrań BeGaze. Podczas trwającej, około czterech minut interakcji FLASH witał się z respondentem, przedstawiał się, a następnie zapraszał do krótkiej zabawy. Zabawa polegała na tym, że robot prosił o pokazanie maskotki z pudełka, które znajdowało się na stanowisku. Następnie odpowiadał z czym mu się ona kojarzy, dlaczego ją lubi lub nie lubi i wyrażał przy tym adekwatną do wypowiedzi emocję. Robot rozróżniał maskotki na podstawie ich koloru. W czasie badań robot pracował w pełni autonomicznie. Przebieg eksperymentu pokazano na rysunku 6.4.

6.2.2 Analiza danych

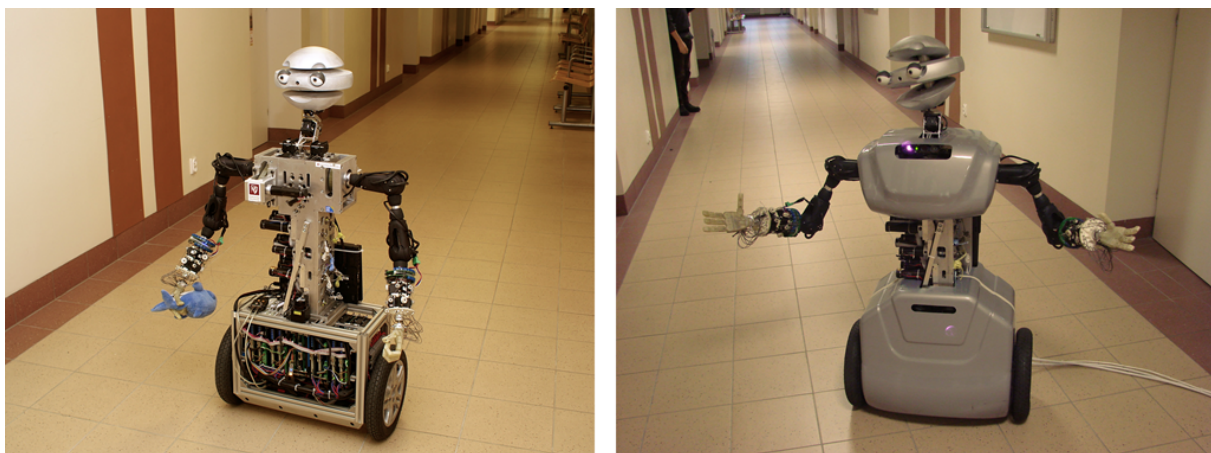
Metoda okulograficzna jest obecnie bardzo często wykorzystywana w badaniach marketingowych. Zasada działania eye-trackera, polega na rejestrowaniu obrazu, jaki widzi badany i ścieżki ruchu jego gałek ocznych. Dzięki temu uzyskuje się dane o tym, gdzie i na jak długo badany skupiał wzrok [59, 143, 154].

Bezpośrednio po zakończeniu sesji z robotem badanego zapraszano na wywiad kwestionariuszowy. Zastosowano klasyczną technikę zbierania danych ankietowych – PAPI (ang. Paper and Pencil Interview), opartą o osobistych wywiadach z respondentami. Ankieterzy odczytywali badanym pytania ankiety, a następnie kodowali ich odpowiedzi na specjalnych formularzach. W badaniu wykorzystano autorski kwestionariusz zawierający

pytania dotyczące cech demograficznych badanych, styczności z techniką, komputerem i grami komputerowymi. Pytano także o pojęcie sztucznej inteligencji i dziedziny Science Fiction oraz skojarzeń, opinii i odczuć wywołanych przez robota FLASH.

Cała próba została podzielona na trzy podgrupy zgodnie z następującym schematem badawczym:

- **robot aemocjonalny w obudowie** – robot nie wyrażał emocji, jedynie gestykował; na konstrukcję mechaniczną założono obudowę,
- **robot emocjonalny w obudowie** – robot wyrażał emocje i gestykował; na konstrukcję mechaniczną założono obudowę,
- **robot aemocjonalny bez obudowy** – robot wyrażał emocje i gestykował; na konstrukcję mechaniczną nie założono obudowy.



Rysunek 6.5: Robot FLASH, po prawej z obudową, z lewej ze zdjętymi osłonami

6.2.3 Wyniki analizy danych okulograficznych

O znaczeniu twarzy w procesie interakcji międzyludzkiej napisano na wstępie niniejszej pracy. Wypełnia ona werbalne i niewerbalne kanały komunikacji [70, 103]. Podobnych zjawisk, jakie towarzyszą kontaktom wzrokowym, poszukiwano w czasie interakcji człowieka z robotem FLASH. Wydaje się, że uzyskane efekty powinny być podobne do tych, jakie otrzymano z badań interakcji międzyludzkiej [82, 183, 99]. Jednakże dotychczas nie przeprowadzono żadnych badań HRI z wykorzystaniem okulo grafu. Jediną znaną autorowi pracą, która wykorzystuje opisywaną metodę w robotyce jest [195], z tą różnicą, że w badaniach użyto fotografii z wizerunkiem robota. Na podstawie pracy [42], przypuszcza się że odczytując emocje na twarzy innych ludzi potrzebujemy również informacji przestrzennych. Łatwiej jest nam rozpoznawać sygnały zakodowane w mimice wówczas, gdy możemy oglądać twarz w 3D. Sugeruje to, że prezentowanie bodźców wzrokowych w postaci filmów i fotografii, może istotnie zniekształcać wyniki badań z wykorzystaniem eye-trackingu.

Wyniki zamieszczono głównie w postaci, tzw. map cieplnych oraz map fiksacji wzroku. „Cieplejsza” barwa lub odsłonięty obszar oznaczają częstsze spoglądanie na dany obszar.

Czy badani skupiali wzrok na twarzy robota?

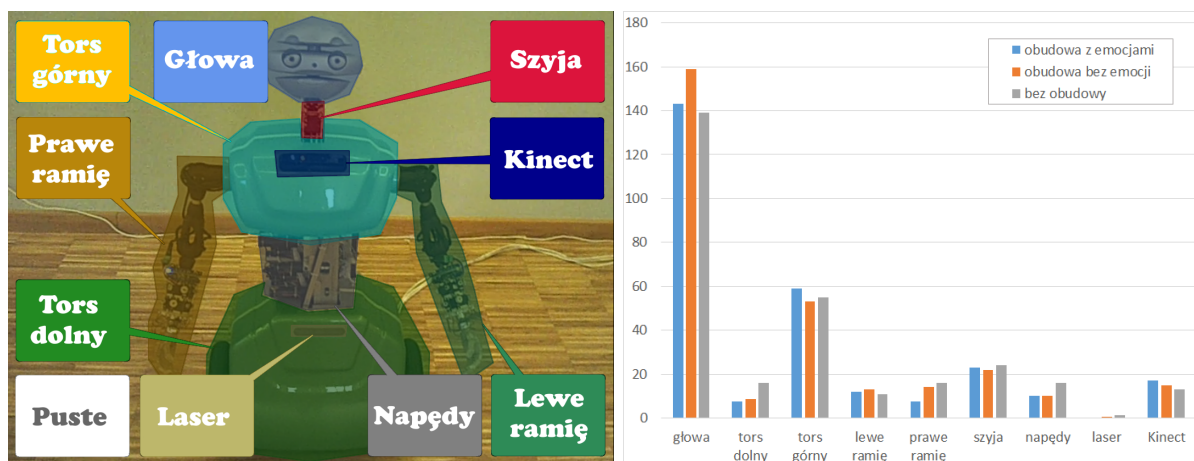
Badania jednoznacznie potwierdzają, że we wszystkich trzech grupach uczestnicy, przede wszystkim skupiali wzrok na twarzy robota. Potwierdza to fakt, że nie traktowali go jak maszyny wolnostojącej na hali produkcyjnej lub wspomagającej człowieka w trakcie wykonywania prac domowych. Zastanawiające jest, że pomimo widocznych elementów mechanicznych i elektronicznych, we wszystkich trzech grupach twarz robota była w centrum uwagi. Mapy ciepłne i mapy fiksacji zaprezentowano na rysunkach 6.6, 6.7, 6.8.

Jakie inne elementy robota przyciągały wzrok?

Okazało się, że równie często co na twarz badany spoglądał na szyję robota. Być może wynikało to z tego, że odsłonięte napędy szyi znajdowały się blisko twarzy. Inne antropomorficzne elementy, np. ramiona i dłonie, również mocno przykuwały uwagę uczestników eksperymentu. Niezwykle interesujące dla uczestników były również wszelkie świecące kontrolki, zarówno te znajdujące się przy czujniku Kinect w korpusie górnym (rysunki 6.6 i 6.7), jak i te przy pozbawionej obudowy platformie (rysunek 6.8). W przypadku serii badań z obudową, dużym zaskoczeniem okazał się zupełny brak zainteresowania badanych nie przykrytymi napędami, w środkowej części korpusu. Jedynie sporadycznie spoglądano na aluminiowy wspornik, który zaburzał nieco symetrię konstrukcji.

Czy przy braku obudowy, widoczne elementy mechanicznej części przyciągały wzrok?

Zaobserwowano, że brak obudowy ma niewielki wpływ na obszary zainteresowań. Nieznacznie zwiększyła się częstość spoglądania na platformę, napędy oraz czujnik laserowy. Jak już wspomniano, po zdjęciu obudowy widoczna się stała szybko mrugająca dioda LED sygnalizująca komunikację platformy z komputerem PC, na którą prawdopodobnie często spoglądano (patrz rysunek 6.8). Twarz wciąż była obszarem dominującym, co wskazuje na to, że brak obudowy nie zakłóca znacząco procesu interakcji, ma za to wpływ na postrzeganie robota, o czym napisano w dalszej części pracy. Na rysunku 6.9 zaprezentowano obszary ROI oraz fiksację wzroku na nich.



Rysunek 6.9: Obszary ROI (z lewej), wykres średniej liczby fiksacji wzroku na zadanych obszarach (po prawej).

Czy zachowanie robota wpływa na obszary zainteresowań?



Rysunek 6.6: Fiksacja wzroku na robocie z obudową wyrażającego emocje



Rysunek 6.7: Fiksacja wzroku na robocie z obudową nie wyrażającego emocji



Rysunek 6.8: Fiksacja wzroku na robocie bez obudowy wyrażającego emocje

W trakcie powitania badani zwracali uwagę zarówno na twarz, jak i na prawą dłoń, którą robot witał się z każdym uczestnikiem (rysunek 6.10). Z czasem, wzrastało zainteresowanie twarzą oraz drugą dłonią, co wydaje się być naturalne, gdyż w pierwszej fazie eksperymentu robot intensywnie gestykulował (rysunek 6.11). Pod koniec sesji (konkretnie, w trakcie pokazywania ostatniej zabawki) badani przyglądali się już niemal tylko twarzy (rysunek 6.12).

6.2.4 Wyniki analizy danych ankietowych

Przed omówieniem wyników uzyskanych na podstawie ankiet należy wspomnieć, że grupa badawcza nie była dostatecznie reprezentatywna. Próba nie była dobierana w pełni losowo. Mimo, iż liczebności we wszystkich trzech porównywanych grupach były wystarczające by zastosować podstawowe testy statystyczne, wyniki badania nie zawsze są istotne statystycznie. Zastrzeżenia dotyczą następujących wskaźników: wiek, wykształcenie, styczność z techniką oraz odczuwanie dyskomfortu. Niemal połowa respondentów mieściła się w przedziale wiekowym 20-29 lat. Załedwie co 10 osoba miała więcej niż 40 lat. Co piąta osoba była w wieku 15-19 lat. Tylko 14% badanych posiadała wykształcenie niższe niż średnie. Wskaźnik poziomu styczności z techniką i dziedziną Science Fiction został oszacowany na podstawie odpowiedzi respondentów na pytania dotyczące częstotliwości korzystania z komputera, grania w gry komputerowe, oglądania TV i seriali (w tym programów nadawanych w Internecie), liczby obejrzanych dotychczas filmów SF, a także znajomości pojęcia sztuczna inteligencja. Jedynie 13% badanych zostało zakwalifikowanych do grupy osób o niskiej styczności z techniką i dziedziną Science Fiction. Średnio co 9 osoba odczuwała dyskomfort w czasie interakcji z robotem FLASH. Pomimo niedostatecznie reprezentatywnej grupy, warto wskazać na pewne wyniki jako kierunki przyszłej eksploracji zagadnienia postrzegania robota społecznego przez osoby zaangażowane w interakcje z nim.

Poniżej zamieszczono wybrane wyniki badań, które przedstawiono w formie pytania i odpowiedzi.

Czy badany odczuwał dyskomfort w czasie interakcji z robotem?

Zdecydowana większość respondentów nie odczuwała dyskomfortu w obecności robota FLASH (wykres 6.13). Najbardziej niekomfortowo czuły się osoby z grupy, w której robot okazywał emocje, lecz nie miał obudowy. Różnica była jednak niewielka. Zaobserwowano, iż odczuwanie dyskomfortu w obecności robota, nie jest powiązane z wcześniejszą stycznością z techniką i dziedziną Science Fiction, płcią, wiekiem czy wykształceniem.

Jak respondenci scharakteryzowali FLASHa?

Respondenci wybierali z podanej listy 25 atrybutów opisujących robota FLASH, średnio 7,4 przymiotnika. Poszczególne grupy badanych różniły się między sobą w liczbie określeń, mianowicie

- Grupa z emocjonalnym robotem z obudową: 8,4;
- Grupa z emocjonalnym robotem z obudową: 7,3;
- Grupa z emocjonalnym robotem bez obudowy: 6,5.

Niemal połowa badanych (48,2%) przypisała robotowi jakieś dodatkowe (nie wymienione w formularzu ankiety) określenie. Średnio ankietowani podawali 0,8 dodatkowych cech.



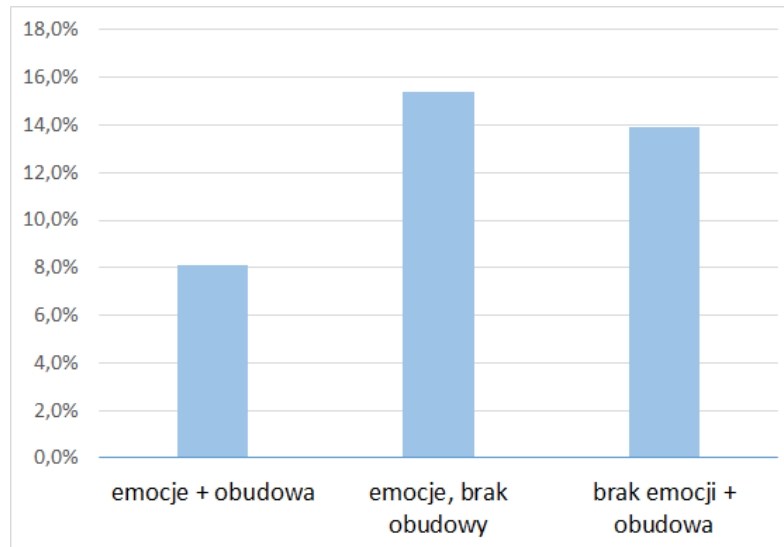
Rysunek 6.10: Fiksacja wzroku na robocie w trakcie powitania



Rysunek 6.11: Fiksacja wzroku na robocie w trakcie przedstawiania się

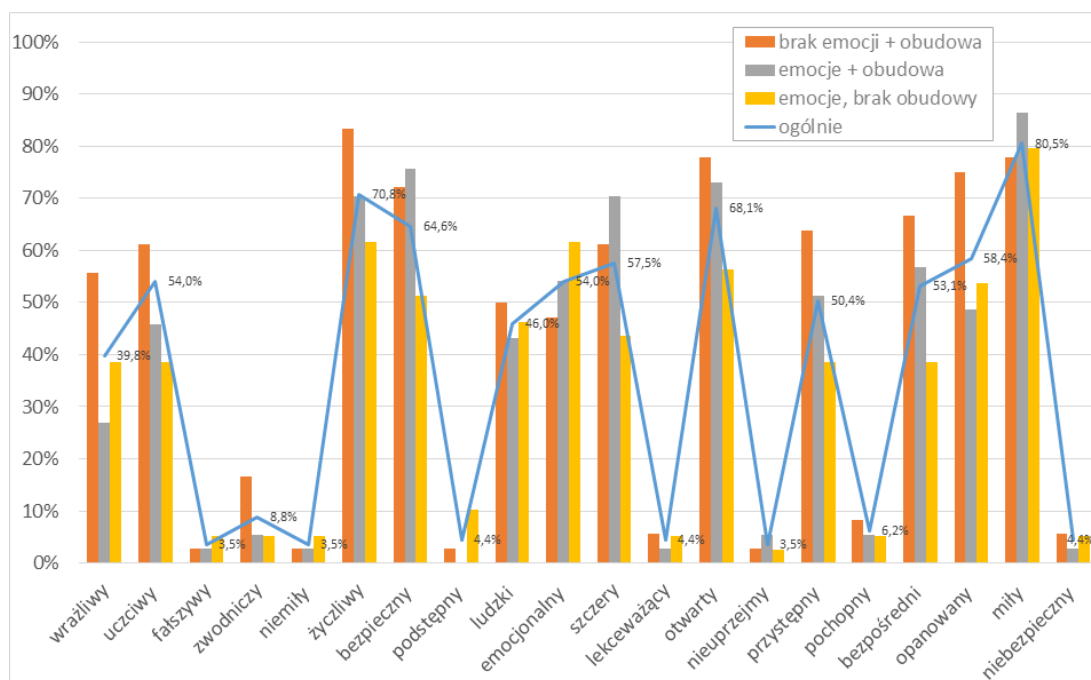


Rysunek 6.12: Fiksacja wzroku na robocie pod koniec sesji



Rysunek 6.13: Poziom odczuwania dyskomfortu w każdej z grup

Zdecydowana większość badanych była zgodna, iż FLASH jest *miły, życzliwy, otwarty i bezpieczny*. Ponad połowa respondentów określiła robota mianem *opanowanego, szczerego, uczciwego, emocjonalnego i bezpośredniego*. Zdaniem respondentów FLASH zdecydowanie nie jest *falszawy, niemiły i nieuprzejmy*. Jedynie pojedyncze osoby określiły robota mianem *podstępny, lekceważący i niebezpieczny*. Poszczególne atrybuty przypisywane przez respondentów robotowi FLASH zostały zaprezentowane na wykresie 6.14. Respondenci byli pytani o to, jakie inne (obok wymienionych w pytaniu) cechy można



Rysunek 6.14: Atrybuty przypisywane robotowi FLASH

przypisać robotowi. Wśród spontanicznie wymienianych atrybutów dominowały pozytywne. Robot budził przede wszystkim rozbawienie oraz ciekawość. Wydaje się, że zastrzeżenia wobec FLASHa dotyczą głównie tego, że jest nadmiernie schematyczny, nie dość twórczy, a także tworzący dystans i odpychający.

Jak zachowanie robota oraz jego urzeczywistnienie wpływały na postrzeganie robota?

Respondenci z grupy z aemocjonalnym robotem z obudową przypisywali robotowi więcej cech niż badani w pozostałych grupach. Różnice między porównywanymi grupami zaobserwowano w odniesieniu do dwóch atrybutów: *wrażliwy oraz bezpośredni*. Mianem *wrażliwego* najczęściej określano robota aemocjonalnego w obudowie. Tymczasem *bezpośredniość* najrzadziej przypisywano emocjonalnej maszynie bez obudowy. Okazujący emocje robot bez obudowy był, w porównaniu z aemocjonalnym i emocjonalnym w obudowie, rzadziej oceniany jako *bezpieczny i szczerzy*. Ekspresywność i obudowa skłaniały uczestników eksperymentu do rzadszego określania robota mianem *opanowanego*. *Uczciwość* przypisywana jest najczęściej robotowi aemocjonalnemu, najrzadziej pozbawionemu obudowy. Najwięcej dodatkowych określeń uzyskano w grupie mającej kontakt z emocjonalnym robotem w obudowie. Badani z tej grupy częściej widzieli FLASHa jako zabawnego i śmiesznego, a także sympatycznego i inteligentnego. W grupie z aemocjonalnym FLASHem w obudowie zwrócono szczególną uwagę na fakt, że robot jest szczerzy, prostolinijny, spokojny i cierpliwy lub okazujący emocje (tj. smutek lub radość). Ta ostatnia obserwacja prawdopodobnie wynika z faktu, że w związku z brakiem konkretnych i zmiennych reakcji emocjonalnych, badani przypisali robotowi „nastrój” (klasyfikując go jako smutny lub radosny). Reasumując, obudowa wydaje się mieć kluczowe znaczenie dla sposobu postrzegania FLASHa.

Jak wykształcenie, wiek, styczność z techniką oraz płeć badanego miały wpływ na postrzeganie robota?

Zaobserwowano, że osoby, które miały wcześniej styczność z techniką oraz dziedziną Science Fiction rzadziej przypisują robotowi *uczciwość*. Z kolei, w grupach o bardzo niskiej styczności z techniką i Science Fiction częściej określano robota mianem *ludzkiego i miłego*.

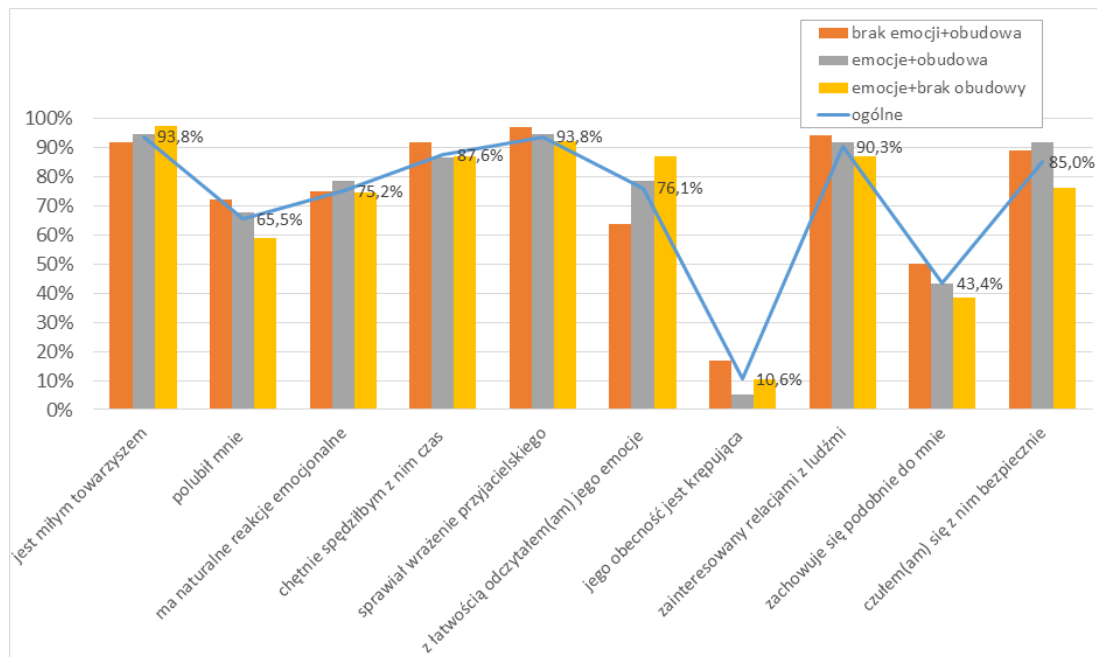
Mężczyźni wykazują nieco mniejsze zaufanie wobec robota społecznego. Częściej postrzegają FLASHa jako *falszywego*. Kobiety częściej widzą robota jako *sympatycznego, pomocnego, dowcipnego, który jest inteligentny, ale też nieco dziecinny*. Dla panów na pierwszy plan wysuwają się cechy, takie jak *schematyczność w działaniu, nowoczesność, funkcjonalność, chłód, nieludzki charakter, niedostateczna kreatywność, brak emocjonalności*. Ponadto, mężczyźni chętnie używają charakterystyk opisowych związanych z wyglądem. Kobiety wybierają silnie nacechowane afektywnie przymiotniki: *słodki, Super Facet, itp.*

Osoby powyżej 30 roku życia częściej deklarują, że robot jest *zwodniczy*. Ponadto, badani z tej grupy są też mniej skłonni nazywać FLASHa *miłym*. Statystycznie najwięcej atrybutów negatywnych, przypisywały robotowi osoby w wieku 40-59 lat. Wydaje się, że wpływ wieku na sposób postrzegania robota FLASH wymaga dalszych badań. Niezbędne jest zebranie danych ankietowych od większej liczby osób, które przekroczyły 40 rok życia.

Respondenci z wykształceniem podstawowym i zawodowym czują się w obecności robota *mniej bezpiecznie*. Ponadto, badani z wymienionych grup rzadziej oceniają FLASHa jako *przystępnego*. Rezultaty te mogą być przejawem lęku osób badanych przed nieznanym. Osoby lepiej wykształcone chętniej przypisują FLASHowi *wrażliwość*.

Jakie przekonania na temat robota wyrazili uczestnicy eksperymentu?

Niemal wszyscy badani byli przekonani, że robot FLASH z pewnością jest *miłym towarzyszem*, a także *sprawia wrażenie przyjacielskiego*. Niewiele mniej osób zgadza się



Rysunek 6.15: Przekonania na temat cech FLASHa

ze stwierdzeniem, iż robot jest zainteresowany nawiązywaniem relacji z ludźmi. Znaczny odsetek ankietowanych deklaruje chęć spędzenia z FLASHem więcej czasu. Być może u podłoża tych twierdzeń leży znaczny poziom bezpieczeństwa w obecności robota. Obecność FLASHa była kłopotliwa dla co 10-tego uczestnika eksperymentu. Warto również zaznaczyć, że ponad połowa respondentów nie identyfikuje się z robotem i jego zachowaniem. Pełniejszy obraz przekonań przedstawia wykres 6.15.

Jak poczucie dyskomfortu wpływa na przekonania na temat robota?

Osoby odczuwające dyskomfort w czasie interakcji z robotem rzadziej twierdzą, że jest on miłym towarzyszem i częściej są skłopotane jego obecnością. Badani z tej grupy negują podobieństwo zachowań FLASHa do własnego postępowania. Poczucie dyskomfortu nie ma związku z subiektywnym wrażeniem, że „robot mnie polubił”, a także z oceną naturalności jego reakcji emocjonalnych czy postrzeganiem jego zainteresowania nawiązywaniem kontaktu z ludźmi.

Jak wykształcenie, wiek, styczność z techniką oraz płeć badanego miały wpływ na przekonania na temat robota?

Mężczyźni częściej oceniają reakcje emocjonalne FLASHa jako nienaturalne. Osoby pomiędzy 50 a 59 rokiem życia częściej negują stwierdzenia dotyczące tego, że FLASH jest miłym towarzyszem. Badani z grupy 50+ częściej niż inni wątpią, by robot był zainteresowany nawiązywaniem relacji z ludźmi. Ankietowani, którzy nie przekroczyli 39 roku życia częściej niż starsi respondenci nie widzą podobieństw między zachowaniami robota a ich własnym postępowaniem. Najmłodszy respondenci (15-19 lat) oraz osoby z przedziału 50-59 lat najczęściej twierdzą, że nie czują się z FLASHem bezpiecznie.

Badani z wykształceniem zawodowym rzadziej twierdzili, że FLASH sprawia wrażenie przyjacielskiego. Osoby z tej grupy miały też największe problemy z odczytaniem reakcji emocjonalnych robota. Obecność FLASHa była mniej kłopotliwa dla osób ze średnim lub

wyższym wykształceniem. Jednocześnie osoby z tej grupy, były bardziej skłonne twierdzić, że FLASH nie był zainteresowany nawiązywaniem relacji z ludźmi. Osoby mające w przeszłości dużą lub bardzo dużą styczność z techniką i dziedziną SF częściej postrzegały reakcje emocjonalne FLASHa jako naturalne.

6.2.5 Wyniki analizy wywiadów pogłębianych

Czy robot posiada cechy ludzkie?

Uczestnicy eksperymentów, we wszystkich grupach badawczych, najczęściej opisywali dłonie robota. To właśnie dzięki nim przypisywali robotowi wiele cech ludzkich. Co więcej, w przypadku grupy z robotem aemocjonalnych, jako ludzki element konstrukcji, badani wskazywali jedynie dłonie. Wielu badanych dostrzegło cechy ludzkie robota w elementach i na podstawie ruchu twarzy. Robot był postrzegany jako emocjonalny, gdyż o jego twarzy mówiono „posiada mimikę”. Respondenci negowali ludzkie cechy robota najczęściej wtedy, gdy nie działał prawidłowo, a zwłaszcza wtedy, gdy zawodziły jego układy percepcji. Badani wyrazili zaskoczenie tym, że robot posiada przekonania i preferencje. Opinię tą wyrażali na podstawie wypowiedzi robota, np. „Nie lubię tej zabawki”, „Nigdy nie pozwalają mi się bawić tą zabawką”, „Uwielbiam tą zabawkę”, itd. Generalnie ta językowa konstrukcja wzbudzała pozytywne emocje. Przyczyniły się do tego także niektóre „ludzkie” potrzeby robota, jak „Muszę odpocząć”, „Muszę iść spać”.

Jak respondenci postrzegają robota?

Respondenci często wymieniali twarz robota, jako element silnie przykuwający wzrok. Cecha ta została potwierdzona badaniami z eye-trackera. Dla wielu z nich, ciekawy był również sposób poruszania się na platformie balansującej. Badani wyrazili opinie, że na podstawie znanych im filmom SF, robot FLASH wyglądem spełnia ich oczekiwania. Nie mniej jednak, wskazali także jego konkretne wady. Wymieniali, między innymi, że jest schematyczny, gubił uczestnika lub błędnie rozpoznawał zabawki. Miało to też pozytywny wydźwięk, ponieważ badani czuli się bezpiecznie z robotem, gdyż jak twierdzili „nie jest jeszcze zbyt skomplikowany”. Bezpośredniość robota budziła skrajne emocje. Niektórzy opisywali go jako rozkazującą i wydającą polecenia maszynę, inni doceniali klarowność wypowiedzi. Bardzo ważne było to, że robot wykazywał zachowania, świadczące o jego zdolnościach percepcyjnych (detekcja człowieka, rozpoznawania zabawek). Badani zwrócili uwagę, że w trakcie wyrażania emocji brakuje adekwatnej modulacji głosu. Pojawił się również wątek testowania robota. Może on mieć znaczenie w kontekście przyszłych kontaktów, np. jako możliwość sposobu na wspólne spędzanie czasu — szukanie błędów w zachowaniu, zaskakujących i nielogicznych reakcji, wprowadzanie go w konsternację, itd.

Czy dostrzeżono różnice postrzegania robota między poszczególnymi grupami?

Ciekawe okazało się, że konstrukcja pozbawiona obudowy częściej była określana mianem „zaawansowana technicznie”, natomiast o robocie z obudową częściej wyrażano się, że jest przyjacielski i deklarowano chęć zaprzyjaźnienia się z nim. W tej grupie doceniano także głos robota. W przypadku robota wyrażającego emocje, ich nazwanie nie sprawiało badanym większych trudności. Częściej używano spontanicznego określenia, że przypomina człowieka. W grupie z robotem aemocjonalnym, respondenci deklarowali równie

wysoką emocjonalność robota jak w pozostałych grupach, ale nie potrafili nazwać żadnej z emocji. Prawdopodobnie, ich ocena wynika jedynie z wypowiedzi FLASHa. Przykładem może być złość, która była przypisywana robotowi na podstawie treści wypowiedzi a nie mimiki. Przejawem emocji mogły też być deklaracje robota na temat tego, co lubi a czego nie lubi. W grupie tej, także rzadziej określano robota mianem ludzkiego, a częściej mówiono o nim jak o maszynie.

Rozdział 7

Uwagi końcowe

W pracy przedstawiono platformę badawczą w postaci robota społecznego FLASH. Zaprezentowano jego konstrukcję mechaniczną, system sterowania omówiono wyniki wybranych eksperymentów. Robot składa się z dwukołowej, balansującej platformy mobilnej, ekspresyjnej głowy złożonej z trzech talerzy oraz dwóch rąk zakończonych zręcznymi dłońmi przypominającej ludzkie odpowiedniki. Zaproponowano i przebadano kilka algorytmów sterowania platformą balansującą. Przeprowadzono symulacje z wykorzystaniem najprostszycy algorytmów, których zasada działania opiera się na przybliżeniu liniowym układu w stanie równowagi. Następnie zaproponowano kilka metod planowania ruchu i sterowania, wykorzystujących pełny nieliniowy model matematyczny robota. Najbardziej obiecującym podejściem jest metoda sterowania predykcyjnego, która cechuje duża odporność na niedokładność przybliżonego modelu matematycznego robota.

Całość jest zarządzana przez elastyczny i łatwy w rozbudowie system sterowania, który został opracowany w oparciu o trójwarstwową architekturę. Stworzono kompletne oprogramowanie, które jest łatwe w instalacji, konfiguracji oraz użytkowaniu, nawet przez mniej doświadczonych programistów robotów. Zastosowane oprogramowanie URBI dostarcza mechanizmów umożliwiających wykorzystanie wszystkich komponentów robota FLASH. Język skryptowy *urbiscript*, w którym programuje się robota jest zorientowany na równoległe zarządzanie zdarzeniami oraz synchronizowanie poszczególnych komponentów systemu sterowania. Zapewnia ono synchroniczną i współbieżną pracę platformy, ramion i dłoni. *Urbiscript* pozwala na szybkie tworzenie scenariuszy zachowań, co przekłada się na krótki czas przygotowywania eksperymentów HRI.

W systemie sterowania napisano zestaw modułów, który wykorzystuje zbiór popularnych bibliotek i sterowników pozwalających na użycie wszystkich zainstalowanych w robocie urządzeń, takich jak silniki, dalmierze, kamery, itd. Niektóre kompetencje robota w całości zrealizowano na poziomie modułów (dynamicznych bibliotek), inne zostały uzupełnione skryptami napisanymi w języku *urbiscript*, a jeszcze inne zaimplementowano w całości na poziomie skryptów. Przygotowany zestaw zachowań robota jest sparametryzowany w taki sposób, aby możliwe było jego łatwe dopasowanie do aktualnej sytuacji i konfiguracji. Dysponując bogatym zestawem zachowań robota można budować złożone scenariusze badawcze.

Ostatecznie, powstał robot zdolny do spójnych i ekspresywnych zachowań, który jest dobrze odbierany przez ludzi i może być uważany za prototyp robota towarzyszącego człowiekowi w jego codziennych czynnościach. Zaproponowany system sterowania robota został przebadany eksperymentalnie.

7.1 Dalsze plany rozwojowe

Opracowana platforma testowa nie stanowi rozwiązania ostatecznego. Wiele jej składników wymaga dalszej rozbudowy, przeprojektowania lub skonstruowania od początku. Niektóre z nich zostały opracowane w taki sposób, aby dostarczyć poszukiwanych rozwiązań w jak najkrótszym czasie, jednak niekoniecznie zachowując przy tym optymalność rozwiązania.

Uwagi dotyczące warstwy sprzętowej

Główny sterownik głowy został zaprojektowany tak, aby realizować proste sterowanie napędami oczu i emulować obecność napędów typu Dynamixel. Zabieg ten miał na celu zachowanie standardów komunikacji, jakie zostały narzucone przez system sterowania dyskami i szyją robota. Sterownik ten nie pozwala wykryć awarii napędów ani realizować dodatkowych funkcji monitorujących ich pracę.

Podczas pracy z głową robota dostrzeżono, że najslabszym punktem konstrukcji są mocno przeciążone napędy wysuwające oczy. Ich wymiana jest konieczna, aby zredukować liczbę awarii robota. Warto także rozważyć wymianę niektórych napędów szyi na silniejsze, gdyż zastosowane napędy pracują na granicy dopuszczalnych parametrów.

Głowa robota EMYS stanowi konstrukcję samodzielną, która może pracować niezależnie od robota. Obecnie prototyp nie posiada systemu mocowania w miejscu pracy. Zaprojektowanie uniwersalnej podstawy umożliwi zainstalowanie głowy w dowolnym miejscu, jak również montaż popularnych i dostępnych na rynku sensorów.

Warto rozważyć opracowanie koncepcji nowego napędu platformy balansującej. Mógłby on dostarczyć większej prędkości przemieszczania się i umożliwić zmniejszenie szerokości wózka. Druga z tych cech, jest kluczowa dla zapewnienia możliwości bezpiecznego przejeżdżania robota przez drzwi i inne wąskie przejścia. Zmniejszenie szerokości platformy mogłoby być uzyskane poprzez zastosowanie krótszych zespołów napędowych lub opracowanie koncepcji przeniesienia napędu. Pozwoliłoby to na nie współliniowe rozmieszczenie silników.

Obecnie komputerem sterującym robota FLASH jest laptop dokowany w stacji bazowej umiejscowionej na plecach robota. Takie położenie komputera jest ryzykowne i może doprowadzić do jego uszkodzenia. Ponadto, z powodu na stałe zainstalowanej stacji dokującej powstało ograniczenie do jednego producenta laptopów. Warto rozważyć zaprojektowanie pokładowej jednostki sterującej, opartej o przemysłowy komputer mający niski pobór energii, który będzie posiadać jedynie niezbędne dla robota komponenty.

Percepcja dźwięku (rozpoznawanie mowy) opiera się na zewnętrznych systemach mikrofonów przewodowych lub bezprzewodowych, podłączanych do komputera sterującego robotem. Warto rozważyć zastąpienie ich przez kierunkowe przetworniki wraz z dedykowanymi narzędziami programistycznymi pozwalającymi m.in. na odfiltrowanie zakłóceń lub rozpoznanie kierunku głosu.

Obecnie nawigacja robota FLASH bazuje na pomiarach z jednego dalmierza laserowego. Dalmierz ten jest umiejscowiony w przedniej części robota na wysokości około 40cm. Pozwala on na skuteczną i bezpieczną nawigację jedynie w ustrukturyzowanym środowisku o niewielkiej liczbie przeszkód. Rozwój urządzeń do nawigacji powinien uwzględnić rozbudowane sensory dedykowane systemom nawigacji – kamery ToF, skanery laserowe 3D i macierze sonarów.

Wadą obecnego korpusu wraz z ramionami jest brak możliwości pracy bez platformy, ponieważ wszystkie moduły sterujące ramion zostały umieszczone w platformie mobil-

nej. Dobrym pomysłem byłoby opracowanie mniejszych sterowników i umieszczenie ich w górnej części korpusu. Ponadto, konstrukcja mechaniczna powinna pozwalać na odłączanie ramion od platformy. Obecnie, z ramienia wyprowadzono liczne ciągnia, które są napędzane silnikami w korpusie.

Robot FLASH wyposażony jest w dłonie, które posiadają 12 stopni swobody i pozwalają na bardzo zaawansowaną gestykulację. Są jednak dość awaryjne i zbyt słabe, by pozwolić na manipulację przedmiotami. Wydaje się, że dostępne na rynku rozwiązania protetyczne z powodzeniem mogłyby być zaadoptowane do robota FLASH.

Ramię robota bazuje na gotowym produkcie Robolink firmy IGUS. Wadą tego rozwiązania jest, występowanie znacznego tarcia przy prowadzeniu linek, które jest przyczyną częstych awarii i ogranicza udźwig ramion. Ponadto, zainstalowany system sterowania ramionami nie wykorzystuje kodarów mierzących położenie i prędkość przegubów, niedostępnych w momencie opracowania sterowników przegubów. Pomiaru tego dokonuje się na osiach silników, co przy panujących w układzie znaczących elastycznościach nie zapewnia wymaganej precyzji ruchu. Przy projektowaniu nowych ramion, warto rozważyć umiejscowienie napędów w ramieniu i wprowadzenie, np. systemu szeregowej elastyczności oraz sterowania impedancyjnego.

Obecnie robot FLASH jest parkowany ręcznie. Konieczne jest zatem stworzenie automatycznego urządzenia dokowania robota (zapewniającego mu stabilność po wyłączeniu układów utrzymywania równowagi) i jego ładowania. W szczególności, należy opracować sposób przechwytywania korpusu po wyłączeniu sterownika platformy jezdnej.

Uwagi dotyczące system sterowania

Na podstawie przeprowadzonych symulacji różnych algorytmów sterowania i planowania ruchu platformy balansującej pokazano, że sterowanie liniowe jest wystarczające dla zadań realizowanych przez robota. Nie mniej jednak można oczekiwać, że stosując nieliniowy sterownik uzyska się lepszą jakość balansowania, zwłaszcza w trakcie przemieszczania się robota. W badaniach symulacyjnych najlepszy algorytm sterowania uzyskano stosując metodę sterowania predykcyjnego. Algorytm pracuje w pętli zamkniętej i gwarantuje dużą odporność na niedokładność modelu dynamiki robota. W zależności od zadań robota, warto rozważyć jego implementację.

Niewątpliwie najsłabszym punktem systemu sterowania jest brak dobrze zaimplementowanej i przetestowanej eksperymentalnie warstwy umysłu robota. Jak dotąd, posługiwano się jedynie automatem stanu oraz przygotowano proste demo z użyciem oprogramowania FAtiMA. Wskazane jest, aby prowadzić dalsze prace dotyczące implementacji architektury umysłu oraz jego integracji z dotychczas istniejącym systemem sterowania robota. Prace te, powinny objąć również modele pamięci i procesy uczenia się.

Równie ważne jest rozwinięcie listy kompetencji (umiejętności) robota. Efekt ten powinien być uzyskany zarówno poprzez dopracowanie już istniejących składników architektury sterowania, jak i dołączenie kolejnych bloków dostarczających nowych umiejętności. Szczególnie pożądane są nowe systemy syntezy i rozpoznawania mowy, systemy nawigacji, oraz oprogramowanie przetwarzające obraz z kamer i skanerów 3D. Istotne jest także opracowanie modułów rozpoznawania emocji użytkownika. Planuje się dołączenie bibliotek pozwalających na pobieranie informacji z Internetu, w tym z lokalnych portali informacyjnych i portali społecznościowych.

Uwagi dotyczące prowadzenia dalszych badań

Na podstawie uzyskanych wyników badań sformowano następujące rekomendacje dotyczące dalszych prac. Subiektywne poczucie dyskomfortu w relacji z robotem wymaga dalszych badań prowadzonych z udziałem większej liczby osób doświadczających nieprzyjemnych uczuć w wyniku kontaktu z maszyną. Poczucie dyskomfortu może być związane z ujawnianiem przez robota emocji takich jak złość czy smutek. Ponadto, przyczyną bardziej pozytywnego odbioru FLASHa emocjonalnego może być dysonans związany z faktem, iż respondenci nie są przyzwyczajeni do obcowania z maszyną zachowującą się jak człowiek. Oba wspomniane aspekty wymagają dalszej analizy. Wbrew przewidywaniom, aemocjonalność wydaje się być korzystną cechą robota. Spostrzeżenie to wymaga dalszej weryfikacji.

Potwierdzone zostały przewidywania dotyczące kluczowej roli obudowy. Robot bez obudowy stwarza znacznie większy dystans społeczny. Niezbędne jest także prowadzenie dalszych badań weryfikujących nastawienie różnych grup docelowych (w tym osób mniej biegłych technicznie, starszych i gorzej wykształconych) do robota FLASH. W obecnym badaniu reprezentacja wymienionych grup była niezbyt liczna. Wstępnie można zakładać, że we wspomnianych grupach docelowych konieczne będzie podjęcie dodatkowych działań osłabiających lęk wobec robota.

Istotne wydaje się przeprowadzenie eksperymentów umożliwiających udzielenie odpowiedzi na następujące cztery podstawowe grupy pytań badawczych:

- Czy i w jaki sposób rodzaj ujawnianych emocji (pozytywne, negatywne, mieszane) rzutuje na sposób postrzegania FLASHA?
- Czy aemocjonalność robota jest rzeczywiście korzystna? W jakich rodzajach interakcji? Z jakimi grupami docelowymi?
- Jaka jest „optymalna emocjonalność” robota (stosunek częstości przejawiania poszczególnych emocji i ich natężenia)?
- U kogo robot FLASH wywołuje dyskomfort? W jaki sposób można go zniwelować? Jakie elementy zachowania i wyglądu robota nasilają lub redukują dyskomfort odczuwany przez uczestnika interakcji?

Warto krytycznie zastanowić się nad obszarami dalszych badań. Rozważyć, czy nie wybiegają one za daleko w przyszłość. Stosując metody WoZ, nie ma sensu badać zachowań robota, które nie są w najbliższym czasie do osiągnięcia z powodu, np. braku technologii. Może okazać się, że kiedy technologia się pojawi, wyniki uzyskane przed laty będą już bezużyteczne. Zagrożenie to, szczególnie dotyczy młodego pokolenia, które coraz częściej przenosi swoje życie do „wirtualnego świata”.

7.2 Obszary możliwego wykorzystania robota FLASH i jego głowy EMYS

Robot FLASH powstał przede wszystkim w celu przeprowadzenia badań HRI w ramach projektu LIREC. Modułowa konstrukcja robota i jego elastyczny system sterowania umożliwiają dalsze prowadzenie badań i zgłębianie wiedzy na temat projektowania i sterowania tego typu konstrukcji. Z powodzeniem może być wykorzystywany w ośrodkach naukowo-badawczych w których prowadzi się badania z robotami społecznymi.

FLASH, po udoskonaleniu konstrukcji, może być wykorzystywany w wielu obszarach życia społeczno-gospodarczego, np.: w placówkach edukacyjnych, w prowadzeniu badań przez instytuty badań i opinii społecznej, w instytucjach leczniczych oraz w działaniach informacyjno-marketingowych podmiotów publicznych i prywatnych. We wszystkich tych dziedzinach istnieje bardzo duże zainteresowanie robotami społecznymi, które zastąpiłyby człowieka. Powody tego zainteresowania są różne:

- **Dla placówek edukacyjnych** – potrzeba stwarzania możliwości poznawania najnowszych technologii przez uczniów szkół podstawowych i średnich, zainteresowania nowymi technologiami uczestników zajęć w świetlicach, kółkach zainteresowań, dzieci o specjalnych potrzebach edukacyjnych i korzystających z zajęć reedukacyjnych, dzieci szczególnie uzdolnionych.
- **Dla instytutów badań i opinii społecznej** – wysokie koszty badań związane z koniecznością angażowania dużej liczby ankierów oraz chęć wyeliminowania, tzw. błędu ankierskiego (tj. zniekształceń danych związanych z cechami i zachowaniem ankiera). Roboty zastępujące ankiera, zmniejszą koszty i przyczynią się do poprawy jakości uzyskiwanych rezultatów. Należy jednak podkreślić, że jedynie maszyna łatwo nawiązująca interakcję z rozmówcą i wzbudzająca jego zainteresowanie, będzie w stanie z powodzeniem zbierać dane ankietowe.
- **Dla instytucji leczniczych** – „opiekowanie się” i dostarczanie rozrywki osobom 50+, pacjentom długoterminowo przebywającym w szpitalach, osobom z łagodnymi zaburzeniami funkcji poznawczych.
- **Dla działań informacyjnych i marketingowych** – tani i pewny sposób udzielania informacji klientom sklepów i punktów sprzedaży, uczestnikom targów i konferencji, osobom korzystającym z kiosków i punktów informacyjnych, np. w urzędach, na dworcach kolejowych, autobusowych i lotniczych, turystom.

Szczególnie obiecująco, jeśli chodzi o możliwość wykorzystania robota FLASH i głowy EMYS, wygląda zainteresowanie tymi robotami ze strony ośrodków naukowo-edukacyjnych i edukacyjnych. Stanowisko z głową EMYS idealnie spełnia funkcję robota edukacyjnego dla dzieci i młodzieży. Programowanie robota przy pomocy prostego interfejsu programistycznego pozwala na uczenie się języka komputerów przez najmłodszych, być może przyszłych inżynierów. Duże możliwości percepcji i ekspresji głowy, jej zabawny i oryginalny kształt, są gwarancją długotrwałego zainteresowania dzieci, gdyż robot ten motywuje do nauki. W przeciwieństwie do stosowanych obecnie rozwiązań edukacyjnych, np. Lego Mindstorms, czy dostępnych na rynku prostych robotów zdalnie sterowanych, głowa EMYS pozwala na tworzenie dużo bardziej rozbudowanych zachowań i wykorzystanie większej liczby sensorów, przez co dłużej utrzymuje zainteresowanie dzieci. To samo stanowisko może być łatwo dostosowane do potrzeb młodzieży o większych umiejętnościach technicznych i programistycznych. Głowa EMYS może być programowana poprzez wydawanie prostych komend w języku skryptowym, zaś tworzenie scenariuszy zachowań jest możliwe dzięki aplikacjom pozwalającym na budowanie automatów stanu. Dla najbardziej zaawansowanych użytkowników przewidziano interfejs programistyczny (API), który dostarcza możliwość pisania własnych bibliotek w językach C i C++. Roboty typu FLASH są również pożądane w ośrodkach naukowych, prowadzących działalność dydaktyczną. Wydziały/instytuty robotyki na uczelniach technicznych wykorzystują tego typu platformy w pracy dydaktycznej ze studentami.

7.3 Aspekty dotyczące komercjalizacji opracowanego stanowiska badawczego

Jeśli chodzi o produkty konkurencyjne wobec robota FLASH na rynku europejskim, jako przykłady można podać takie konstrukcje, jak: francuski robot NAO [182] i włoski iCub [115]. Robot NAO jest niewielkich rozmiarów (58cm wzrostu i 4,3 kg masy). Pomimo swoich bardzo skromnych możliwości ekspresyjnych (brak ruchomych elementów twarzy) i ograniczonych możliwości sensorycznych (dwie kamery RGB, brak możliwości zintegrowania z robotem dodatkowych sensorów), jest obecnie wykorzystywany w ponad 200 instytucjach naukowych na całym świecie. Ten komercyjny robot jest także wykorzystywany do wsparcia edukacji w szkołach średnich, terapii autyzmu, a także w wielu projektach naukowych. Drugą konstrukcją, którą w pewnym zakresie można porównywać do FLASHa jest otwarty projekt rozwijający robota iCub. Jest to humanoidalny robot, przypominający proporcjami ciało 4-letniego chłopca. Pomimo bardzo wysokiej ceny (250 tys. euro), robot został zakupiony przez ponad 20 europejskich instytucji badawczych, które wykorzystują go w swoich działaniach na polu robotyki społecznej. Do zalet robota należy: dopracowana konstrukcja mechaniczna, „skóra” robota posiadająca możliwość czucia dotyku i rozbudowane możliwości planowania ruchu. Z drugiej strony, iCub musi być podłączony do zewnętrznego zasilania, zewnętrznej jednostki obliczeniowej o dużej mocy, a prace nad jego możliwościami lokomocyjnymi są wciąż w toku (w większości aplikacji robot stoi lub siedzi). Jego percepcja otoczenia oparta jest jedynie na kamerach RGB. najslabszą stroną tego robota jest skomplikowany i trudny do modyfikacji system sterowania. iCub posiada małe możliwości ekspresji, a ze względu na podobieństwo do twarzy dziecka, spotyka się czasami z negatywnymi reakcjami (problem doliny osobliwości). Wskazanie konkurencji dla głowy EMYS wydaje się być trudne. Na rynku są dostępne bardzo małe roboty – produkowane w charakterze zabawek dla dzieci. Zazwyczaj mają postać zwierząt. Ich ceny kształtują się od 1 tys. (np. roboty PLEO [201], iCat [114]) do 5 tys. zł. Jednak, że w żaden sposób nie można ich porównywać do EMYSa, ponieważ zabawki te mają zamkniętą architekturę sterowania i ograniczoną funkcjonalność. Ustalenie atrakcyjnej, dla potencjalnych odbiorców ceny sprzedaży robota FLASH, pozwoli w sposób naturalny uzyskać przewagę konkurencyjną producenta robotów. Szacuje się, że cena robota FLASH mogłaby oscylować wokół 250 tys. zł, a samej głowy około 20 tys. zł.

7.4 Słowo końcowe

Jest świetny, super facet, zabieram go do domu! – uczestniczka eksperymentu z robotem FLASH.

Generalnie, robot FLASH wzbudzał pozytywne emocje, a uczestnicy eksperymentów czuli się z nim bezpiecznie. Dostrzegali w nim cechy ludzkie, zarówno na podstawie zachowań jak i wyglądu. Robot zdecydowanie był postrzegany jako emocjonalny, spokojny i opanowany, a większość badanych, zarówno dzieci jak i dorosłych, zadeklarowała chęć ponownego spotkania się z robotem.

Specjalnie wzięłam dwa razy tę samą zabawkę, aby go sprawdzić, bo cały czas mówił, że nie cierpi tej zabawki – uczestnik eksperymentu z robotem FLASH.

Robot podczas wszystkich badań pracował w pełni autonomicznie. W swoich zachowaniach często był schematyczny i nie wykazywał zaawansowanych umiejętności. Pomimo tego, badani mówili o nim, że jest inteligentny i mądry.

Robot mógłby odrabiać za mnie lekcje, bo mi się już nie chce – Bartek, uczestnik eksperymentu z głową EMYS.

Nawet najmłodsi uczestnicy eksperymentów nie mieli żadnych trudności ze wskazaniem przykładowych zastosowań robota. Najczęściej były to czynności, których dzieci nie lubią, jak: odrabianie lekcji, sprząatanie pokoju lub wyprowadzanie psa. Jednakże dorosli równie często wyobrażali sobie robota jako pomoc domową, zwłaszcza kobiety często widziały go w roli kucharza.

Bardzo chciałbym mieć takiego robota. Mógłby bawić się z moją młodszą siostrą – Kacper, uczestnik eksperymentu z głową EMYS.

Nie wielu badanych wykazywało obawy przed nowymi technologiami. Zwłaszcza najmłodsi respondenci w pełni zaufali konstrukcji i głęboko wierzyli, że w przyszłości będą mogli z niej skorzystać, wspólnie spędzać czas i bawić się.

Tak, robot był zaskakujący, bo rozpoznawał swoje zabawki i reagował na to, co mu pokazywałam – uczestniczka eksperymentu z robotem FLASH.

Nierzadko proste układy percepcji i adekwatna reakcja na bodźce z otoczenia znacznie podnoszą ocenę robota. Wyniki eksperymentów z obydwoma robotami zdecydowanie potwierdzają, że badani dostrzegli w nim znacznie więcej umiejętności niż, faktycznie one posiadały.

Jestem zaskoczona, że takie roboty już istnieją, myślałam, że tylko w filmach – uczestniczka eksperymentu z robotem FLASH.

Chociaż robot FLASH ustępuje innym robotom społecznym pod względem umiejętności, jednak jako pierwszy integruje tak wiele różnorodnych technologii robotycznych towarzyszy człowieka. Robot ma duży potencjał rozwoju. Wyniki eksperymentów potwierdzają, że nawet proste ludzkie zachowania, jakie generował robot, były oceniane przez ludzi jako wiarygodne. Interdyscyplinarny charakter dziedziny robotyki społecznej oraz jej emergentny rozwój, może wkrótce zaskoczyć nawet jej twórców.

Miłość, małżeństwo i sex z robotami? Nie za milion lat? Być może znacznie szybciej? – z okładki książki, „Love and Sex with Robots”, D. Levy [153]

Bibliografia

- [1] J. Ackermann. *Sampled-data control systems: analysis and synthesis, robust system design*. Communications and control engineering series. Springer-Verlag, 1985.
- [2] ADEPT. MobileRobots. Autonomous mobile robots, software, and accessories. <http://www.mobilerobots.com>.
- [3] AIBO. Sony, Japonia. <http://support.sony-europe.com/aibo>.
- [4] G. Andersson, K. Höök, D. Mourão, A. Paiva, M. Costa. Using a Wizard of Oz study to inform the design of SenToy. *4th Conference on Designing interactive systems: processes, practices, methods, and techniques*, DIS '02, strony 349–355, New York, NY, USA, 2002. ACM Press.
- [5] K. Arent, K. Tchoń. Roboty społeczne. *Postępy robotyki*, wolumen 2, strony 629–648, Oficyna Wydawnicza Politechniki Warszawskiej, Warszawa, 2012.
- [6] ARIA. ARIA Wiki, MobileRobots Inc. <http://robots.mobilerobots.com/wiki/ARIA>, 2013.
- [7] D. Ariens, B. Houska, H. J. Ferreau. ACADO for Matlab User's Manual. <http://www.acadotoolkit.org>, 2010–2011.
- [8] ARMAR-III. Humanoids and Intelligence Systems Lab - Institute for Anthropomatics, Karlsruhe Institute of Technology. <http://his.anthropomatik.kit.edu/english>, 2013.
- [9] ARNL. ARNL Wiki, MobileRobots Inc. http://robots.mobilerobots.com/wiki/ARNL,_SONARNL_and_MOGS, 2013.
- [10] ASIMO. Honda Motor Co. <http://world.honda.com/ASIMO>, 2013.
- [11] K. J. Aström, B. Wittenmark. *Computer-controlled systems*. Prentice Hall, Upper Saddle River, NJ, USA, 1997.
- [12] R. Aylett, P. Cuba, A. Deshmukh, K. D. Casse, D. Griffiths, K. L. Koay, M. Kriegel, I. Leite, Ł. Małek. Updated integration architecture. *LIREC GROUP Deliverable 9.4*, 2010.
- [13] R. Aylett, K. Dautenhahn, A. Deshmukh, S. Correia, T. Correia, J. Dias, P. Gomes, D. Griffiths, W. C. Ho, M. Keysermann, K.L. Koay, M. Kriegel, I. Leite, C. Martinho, Ł. Małek, F. Melo, A. Paiva, F. Papadopoulos, A. Pereira, T. Ribeiro, D.S. Syrdal, M. Vala, M.L. Walters. Final versions of integrated companions. *LIREC GROUP Deliverable 9.6*, 2012.

- [14] R. Aylett, X. Y. Deng, A. Deshmukh, J. Dias, S. Enz, I. Leite, C. Martinho, A. Paiva, R. Prada, P. A. Vargas, A. Wichert. Theories and mechanisms that support dynamic models of Social Relations. *LIREC GROUP Deliverable 5.1*, 2008.
- [15] J. Bach. The MicroPsi Agent Architecture. *ICCM-5, International Conference on Cognitive Modeling*, strony 15–20, Bamberg, Niemcy, 2003.
- [16] J. Bach, C. Bauer, R. Vuine. Micropsi: contributions to a broad architecture of cognition. *29th annual German conference on Artificial Intelligence*, strony 7–18, Bremen, Niemcy, 2007.
- [17] G. Bailly. Audiovisual Speech Synthesis. *International Journal of Speech Technology*, 6:6–331, 2001.
- [18] S. Baron-Cohen. Precursors to a theory of mind: Understanding attention in others. *Natural Theories of Mind: Evolution, Development and Simulation of Everyday Mindreading*. Basil Blackwell, Oxford, 1991.
- [19] C. Bartneck, J. Forlizzi. A Design Centred Framework for Social Human-Robot Interaction. *13th IEEE International Workshop on Robot and Human Interactive Communication*, strony 591–994, Kurashiki, Okayama, Japonia, 2004.
- [20] C. Becker, S. Kopp, I. Wachsmuth. *Why emotions should be integrated into conversational agents*. University of Bielefeld, Bielefeld, Niemcy, 2007.
- [21] C. Becker-Asano. *WASABI: Affect simulation for agents with believable interactivity*. University of Bielefeld, Bielefeld, Niemcy, 2008.
- [22] K. Berns, J. Hirth. Control of facial expressions of the Humanoid Robot Head ROMAN. *IEEE/RSJ International Conference On Intelligent Robots and Systems (IROS)*, strony 9–15.
- [23] C. L. Bethal, R. R. Murphy. Affective Expression in Appearance-Constrained Robots. *2nd Conference on Human Robot Interaction (HRI '06), ACM SIGCHI/SIGART*, strony 327–328, Salt Lake City, Utah, US, 2006.
- [24] L. Biagiotti, F. Lotti, G. Palli, P. Tiezzi, G. Vassura, C. Melchiorri. Development of UB Hand 3: Early results. *IEEE International Conference on Robotics and Automation, ICRA*, strony 4488 – 4493, 2005.
- [25] T. Bickmore, R. Picard. Establishing and maintaining long-term human-computer relationships. *ACM Transaction on Computer-Human Interaction*, 12(2):293–327, 2005.
- [26] E. Bonabeau, M. Dorigo, G. Theraulaz. *Swarm Intelligence : From Natural to Artificial Systems: From Natural to Artificial Systems*. Oxford University Press, USA, 1999.
- [27] M. Bowler, H. Cramer, S. Correia, K. Dautenhahn, M. Jacobsson, M. Janiak, J. Kędzierski, K. Lee Koay, Ł. Malek, R. Muszyński, A. Oleksy, F. Papadopolos, D. Sverre Syrdal, K. Tchoń, M. Walters, M. Wnuk. Preliminary Studies with Embodied Companions. *LIREC GROUP Deliverable 6.3*, 2011.

- [28] G. Bradski, A. Kaehler. *Learning OpenCV: Computer Vision with the OpenCV Library*. O'Reilly, Cambridge, MA, 2008.
- [29] Michael Brady. Artificial intelligence and robotics. *Artificial Intelligence*, 26(1):79–121, 1985.
- [30] C. Breazeal. Towards sociable robots. *Robotics and Autonomous Systems*, 42, 2002.
- [31] C. Breazeal. Emotion and sociable humanoid robots. *International Journal of Human-Computer Studies*, 59(1-2):119–155, 2003.
- [32] C. Breazeal. Role of expressive behaviour for robots that learn from people. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 364(1535):3527–3538, 2009.
- [33] C. Breazeal, C. D. Kidd, A. Lockerd Thomaz, G. Hoffman, M. Berlin. Effects of nonverbal communication on efficiency and robustness in human-robot teamwork. *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, strony 383–388, 2005.
- [34] C. L. Breazeal. *Designing Sociable Robots*. Intelligent Robots and Autonomous Agents. A Bradford Book, The MIT Press, 2002.
- [35] R. A. Brooks. A robust layered control system for a mobile robot. *IEEE Journal of Robotics and Automation*, 1(1):14–23, 1986.
- [36] H. Bruyninckx. Open robot control software: the orocos project. *IEEE International Conference on Robotics and Automation, ICRA*, wolumen 3, strony 2523–2528, 2001.
- [37] R. Budziński, J. Kędziński, B. Weselak. Głowa robota społecznego Samuel - konstrukcja. *Postępy robotyki*, wolumen 2, strony 185–194, Oficyna Wydawnicza Politechniki Warszawskiej, 2010.
- [38] E.F. Camacho, C. Bordons. *Model Predictive Control*. Advanced Textbooks in Control and Signal Processing. Springer-Verlag, 2004.
- [39] J. Canemaker. *Tex Avery: The MGM Years*. Turner Press, Atlanta, 1996.
- [40] Care-O-bot. Fraunhofer IPA. <http://www.care-o-bot.de/english/index.php>, 2013.
- [41] C. Chao, J. Lee, M. Begum, A.L. Thomaz. Simon plays Simon says: The Timing of Turn-taking in an Imitation Game. *IEEE International Symposium on Robot and Human Interactive Communication, ROMAN*, 2011.
- [42] O. Chelnokova, B. Laeng. Three-dimensional information in face recognition: An eye-tracking study. *Journal of Vision*, 11(13), 2011.
- [43] C. Chen, O. L. Mangasarian. Smoothing methods for convex inequalities and linear complementarity problems. *Mathematical Programming*, 71(1):51–69, 1995.
- [44] T. H. J. Collett, B. A. MacDonald, B. P. Gerkey. Player 2.0: Toward a practical robot programming framework. *Australasian Conference on Robotics and Automation*, 2005.

- [45] J. Condon. When People talk with People. *Semantics and Communication*, strony 87–107, 1965.
- [46] P. T. Costa, R. R. McCrae. *Revised NEO Personality Inventory (NEO-PI-R) and NEO Five-Factor Inventory (NEO-FFI) manual*. Psychological Assessment Resources, 1992.
- [47] CRONOS. University of Essex and University of Bristol. <http://cronosproject.net>, 2012.
- [48] J. G. Cunningham, R. D. Odom. Differential salience of facial features in children's perception of affective expression. *Child Development*, 57(1):136–142, 1986.
- [49] S. Dalibardand, N. Magnenat-Thalmann, D. Thalmann. Anthropomorphism of artificial agents: a comparative survey of expressive design and motion of virtual characters and social robots. *Workshop on Autonomous Social Robots and Virtual Humans at the 25th Annual Conference on Computer Animation and Social Agents, CASA*, Singapore, 2012.
- [50] K. Dautenhahn. The Art of Designing Socially Intelligent Agents - Science, Fiction and the Human in the Loop. *Applied Artificial Intelligence Journal, Special Issue on Socially Intelligent Agents*, 12:12–7, 1998.
- [51] K. Dautenhahn. Robots we like to live with?!? A developmental perspective on a personalized, life-long robot companion. *3rd IEEE International Workshop on Robot and Human Interactive Communication, ROMAN*, strony 17–22, Kurashiki, Japonia, 2004.
- [52] K. Dautenhahn. Socially intelligent robots: dimensions of human–robot interaction. *Philosophical Transactions of the Royal Society B*, strony 679–704, 2007.
- [53] K. Dautenhahn, A. Billard. Bringing up Robots or – The Psychology of Socially Intelligent Robots: From Theory to Implementation. *3rd International Conference on Autonomous Agents*, strony 366–367, 1999.
- [54] K. Dautenhahn, M. Jacobsson, M. Janiak, J. Kędzierski, Ł. Malek, R. Muszynski, A. Oleksy, K. Tchoń, M. Walters, M. Wnuk. Design of Embodied Companions. *LIREC GROUP Deliverable 6.2*, 2010.
- [55] K. Dautenhahn, M. Jacobsson, M. Janiak, J. Kędzierski, A. Miklosi, S. Pedrosa, A. Pereira, C. Spielhagen, K. Tchoń, M. Walters. Foundations of Embodied Companions. *LIREC GROUP Deliverable 6.1*, 2009.
- [56] K. Dautenhahn, S. Woods, C. Kaouri, M.L. Walters, K. Lee Koay, I. Werry. What is a robot companion - friend, assistant or butler? *IEEE/RSJ International Conference on Intelligent Robots and Systems, (IROS)*, strony 1192–1197, 2005.
- [57] A. De Luca, S. Iannitti, R. Mattone, G. Oriolo. Control problems in underactuated manipulators. *IEEE/ASME International Conference on Advanced Intelligent Mechatronics*, wolumen 2, strony 855–861, 2001.

- [58] A. Deshmukh, G. Castellano, A. Kappas, W. Barendregt, F. Nabais, A. Paiva, T. Ribeiro, I. Leite, R. Aylett. Towards empathic artificial tutors. *8th ACM/IEEE international conference on Human-robot interaction, HRI '13*, strony 113–114. ACM/IEEE Press, 2013.
- [59] G. Diamantopoulos. *Novel eye feature extraction and tracking for non-visual eye-movement applications*. Praca doktorska, The University of Birmingham, Wielka Brytania, 2010.
- [60] J. Dias. FAtiMA Modular: Towards an Agent Architecture with a Generic Appraisal Framework. *Workshop on Standards in Emotion Modeling*, Leiden.
- [61] J. Dias. Fearnot! Creating emotional autonomous synthetic characters for empathic interactions. Praca magisterska, Universidade Técnica de Lisboa, Instituto Superior Técnico, Lisboa, 2005.
- [62] J. Dias, A. Paiva. Feeling and Reasoning: A Computational Model for Emotional Characters. Carlos Bento, Amílcar Cardoso, Gaël Dias, redaktorzy, *Progress in Artificial Intelligence*, wolumen 3808 serii *Lecture Notes in Computer Science*, strony 127–140. Springer Berlin Heidelberg, 2005.
- [63] C. F. DiSalvo, F. Gemperle, J. Forlizzi, S. Kiesler. All robots are not created equal: The design and perception of humanoid robot heads. *4th conference on Designing interactive systems*, strony 321–326. ACM Press, 2002.
- [64] DOMO. Humanoid Robotics Group, MIT Computer Science and Artificial Intelligence Laboratory. <http://people.csail.mit.edu/edsinger>, 2013.
- [65] D. Dörner. The mathematics of emotions. *Fifth International Conference on Cognitive Modeling*, strony 75–79, Niemcy, 2003.
- [66] Dynamixel. Robotis Inc. <http://www.robotis.com>, 2013.
- [67] M. Dziergwa. Interakcje z robotem społecznym. Praca magisterska, Politechnika Wrocławska, Wrocław, 2013.
- [68] P. Ekman, W. Friesen. *Unmasking the face: A guide to recognizing emotions from facial clues*. Prentice-Hall, 1975.
- [69] P. Ekman, W. Friesen. *Facial Action Coding System: A Technique for the Measurement of Facial Movement*. Consulting Psychologists Press, Palo Alto, 1978.
- [70] P. Ekman, W. Friesen, P. Ellsworth. What emotion categories or dimensions can observers judge from facial behavior? *Emotion in the Human Face*, strona 39–55. Cambridge University Press, Wielka Brytania, 1982.
- [71] P. Ekman, W. Friesen, J. C. Hager. *Facial Action Coding System*. Research Nexus Division of Network Information Research Corporation, 2002.
- [72] M. El-Nasr, J. Yen, T. Ioeger. FLAME - A Fuzzy Logic Adaptive Model of Emotions. *Autonomous Agents and Multiagent Systems*, 364(3):219–257, 2000.
- [73] Elu2-RT2. Torso 2, Elumotion Ltd. <http://www.elumotion.com>, 2013.

- [74] EMIEW. Hitachi ltd. <http://www.hitachi.com/rd/research/robotics.html>, 2013.
- [75] K. G. Engelhardt, R. A. Edwards. Human-robot integration for service robotics. Mansour Rahimi, Waldemar Karwowski, redaktorzy, *Human-Robot Interaction*, strony 15–346. Taylor & Francis Ltd, New York, 1992.
- [76] F. W. Fairman. *Linear Control Theory: The State Space Approach*. John Wiley and Sons, Chichester, Wielka Brytania, 1998.
- [77] E. Ferrari, B. Robins, K. Dautenhahn. Therapeutic and educational objectives in robot assisted play for children with autism. *The 18th IEEE International Symposium on Robot and Human Interactive Communication, ROMAN*, strony 108–114, 2009.
- [78] C. G. Fisher. Confusions among visually perceived consonants. *J Speech Hear Res*, 11(4):796–804, 1968.
- [79] FLASH. UObjects - FLASH Control System Documentation. <http://flash.lirec.ict.pwr.wroc.pl/?q=node/73>.
- [80] T. Fong, I. Nourbakhsh, K. Dautenhahn. A Survey of Socially Interactive Robots. 42:143–166, 2003.
- [81] S. Franklin, A. Graesser. Is it an Agent, or Just a Program?: A Taxonomy for Autonomous Agents. *Workshop on Intelligent Agents III, Agent Theories, Architectures, and Languages, ECAI '96*, strony 21–35, London, Wielka Brytania, 1997. Springer-Verlag.
- [82] M. Fridin, A. Barliya, E. Schechtman, B. deGelder, T. Flash. Computational model and the human perception of emotional body language (EBL). *Symposium on Mental States, Emotions and their Embodiment*, strony 16–19, 2009.
- [83] M. Fridin, Y. Yaakobi. Educational Robot for Children with ADHD/ADD. *Architectural Design, International Conference on Computational Vision and Robotics*, Bhubaneswar, Indie, 2011.
- [84] N. H. Frijda. *The Emotions*. Cambridge University Press, Cambridge, 1986.
- [85] D. Frydecka, M. Zagdańska. Postrzeganie robota społecznego FLASH. Instytut Automatyki i Robotyki Politechniki Wrocławskiej, Raport serii SPR nr1/2013, 2013.
- [86] FURO. Future Robot. <http://www.futurerobot.com>, 2013.
- [87] E. Gat. On Three-Layer Architectures. *Artificial Intelligence and Mobile Robots*. MIT Press, 1998.
- [88] P. Gebhard. ALMA - A Layered Model of Affect. *Fourth International Joint Conference on Autonomous Agents and Multiagent Systems*, Utrecht, Holandia, 2005.
- [89] J. Goetz, S. Kiesler, A. Powers. Matching Robot Appearance and Behavior to Tasks to Improve Human-Robot Cooperation. *12th IEEE International Workshop on Robot and Human Interactive Communication*, strony 55–60, Berkeley, CA, USA, 2003.

- [90] E. Goffman. *Zachowanie w miejscach publicznych*. Wydawnictwo Naukowe PWN, Warszawa, 2008.
- [91] G. Granosik, M. Stanusch, K. Wójtowicz. Oprogramowanie robota społecznego Tepson. *Postępy robotyki*, wolumen 1, strony 195–206, Oficyna Wydawnicza Politechniki Warszawskiej, 2010.
- [92] G. Granosik, M. Stanusch, K. Wójtowicz. Robot społeczny Tepson. *Pomiary Automatyka Robotyka*, 14(6), 2010.
- [93] F. Grasser, A. D'Arrigo, S. Colombi, A.C. Rufer. Joe: a mobile, inverted pendulum. *IEEE Transactions on Industrial Electronics*, 49(1):107–114, 2002.
- [94] J. Gratch, S. Marsella. Modeling Coping Behaviour in Virtual Humans: Don't Worry, Be Happy. *Second International Joint Conference on Autonomous Agents and Multiagent Systems*, strony 313–320, 2003.
- [95] J. Gratch, S. Marsella. A domain-independent framework for modeling emotion. *Journal of Cognitive Systems Research*, 5:269–306, 2004.
- [96] H. Górecki, S. Fuksa, A. Korytowski, W. Mitkowski. *Sterowanie optymalne w systemach liniowych*. Warszawa, 1983.
- [97] W. Greve, D. Wentura. *Scientific observation: An introduction*. PVU/Beltz, 1997.
- [98] A.L. Gross, B. Ballif. Children's understanding of emotion from facial expressions and situations: A review. *Developmental Review*, 11(4):368–398, 1991.
- [99] K. Guo. Holistic Gaze Strategy to Categorize Facial Expression of Varying Intensities. (8), 2012.
- [100] E. T. Hall. Adumbration as a Feature of Intercultural Communication. *American Anthropologist*, 66(2), 1964.
- [101] T. Hamada, H. Okubo, K. Inoue, J. Maruyama, H. Onari, Y. Kagawa, T. Hashimoto. Robot therapy as for recreation for elderly people with dementia - Game recreation using a pet-type robot. *The 17th IEEE International Symposium on Robot and Human Interactive Communication, ROMAN*, strony 174–179, 2008.
- [102] D. Hanson. Exploring the Aesthetic Range for Humanoid Robots. *Cognitive Science (CogSci 2006) Workshop on Android Science*, strony 47–52, Vancouver, BC, Kanada, 2006.
- [103] R. Hari, M. V. Kujala. Brain Basis of Human Social Interaction: From Concepts to Brain Imaging. *Physiol Rev*, 89(6):453–479, 2009.
- [104] B. Hartmann, M. Mancini, C. Pelachaud. Towards affective agent action: Modelling expressive ECA gestures. *International conference on Intelligent User Interfaces – Workshop on Affective Interaction*, 2005.
- [105] P. J. Hinds, T. L. Roberts, H. Jones. Whose job is it anyway? a study of human-robot interaction in a collaborative task. *Human Computer Interaction 19*, strony 151–181, 2004.

- [106] Hitec. HS-82 MG, Hitec RCD USA Inc. <http://www.hitecrd.com>, 2013.
- [107] B. Houska, H. J. Ferreau. ACADO Toolkit User's Manual. <http://www.acadotoolkit.org>, 2009–2011.
- [108] B. Houska, H.J. Ferreau, M. Diehl. ACADO Toolkit – An Open Source Framework for Automatic Control and Dynamic Optimization. *Optimal Control Applications and Methods*, 32(3):298–312, 2011.
- [109] J. Höysniemi, P. Hämäläinen, L. Turkki. Wizard of Oz prototyping of computer vision based action games for children. *Conference on Interaction design and children: building a community*, IDC '04, strony 27–34, New York, NY, USA, 2004. ACM Press.
- [110] HRP-4. Kawada Industries, Inc. <http://global.kawada.jp/mechatronics>, 2013.
- [111] HUBO. HUBO Lab, Humanoid Robot Research Center. <http://hubolab.kaist.ac.kr>, 2013.
- [112] E. Hudlicka. Affective computing for game design. *4th Intl. North American Conference on Intelligent Games and Simulation (GAMEON-NA)*, strony 5–12, Kanada, 2008. McGill University.
- [113] I. Iacono, H. Lehmann, P. Marti, B. Robins, K. Dautenhahn. Robots as social mediators for children with Autism - A preliminary analysis comparing two different robotic platforms. *Development and Learning (ICDL), 2011 IEEE International Conference on*, wolumen 2, strony 1–6, 2011.
- [114] iCat. Research robot iCat, Philips, Holland. www.research.philips.com/technologies/syst_softw/robotics/index.html.
- [115] iCub. An open source cognitive humanoid robotic platform. <http://www.icub.org>, 2013.
- [116] IROMECC. IROMECC - Interactive RObotic social MEdiators as Companions. <http://www.iromec.org>.
- [117] A. Isidori. *Nonlinear Control Systems*. Springer-Verlag, London, 1995.
- [118] V. Jones J. Han, M. Jo, J. H. Jo. Comparative Study on the Educational Use of Home Robots for Children. *Journal of Information Processing Systems*, 4(4):159, 2008.
- [119] M. Janiak. *Jakobianowe algorytmy kinematyki odwrotnej manipulatorów mobilnych z ograniczeniami na sterowanie, stan i zachowanie*. Praca doktorska, Politechnika Wrocławska, Wrocław, 2009.
- [120] M. Janiak, K. Tchoń. Constrained robot motion planning: Imbalanced jacobian algorithm vs. optimal control approach. *15th International Conference on Methods and Models in Automation and Robotics (MMAR)*, strony 25–30, 2010.
- [121] J. B. Janssen, C. C. Wal, M. A. Neerincx, R. Looije. Motivating Children to Learn Arithmetic with an Adaptive Robot Game. Bilge Mutlu, Christoph Bartneck, Jaap Ham, Vanessa Evers, Takayuki Kanda, redaktorzy, *Social Robotics*, wolumen 7072 serii *Lecture Notes in Computer Science*, strony 153–162. Springer Berlin, 2011.

- [122] E. Jezierski. *Dynamika robotów*. Wydawnictwa Naukowo-Techniczne, Warszawa, 2006.
- [123] JSK. Assistant Robot, Johou Systems Kougaku Laboratory, University of Tokyo. <http://www.jsk.t.u-tokyo.ac.jp>, 2012.
- [124] S. J. Julier, J. K. Uhlmann. A new extension of the Kalman filter to nonlinear systems. strony 182–193, 1997.
- [125] M. Kabała K. Tchoń, M. Wnuk. Trajectory tracking algorithm for MK mobile robot. *XIV National Conference on Automation*, wolumen 2, strony 663–668. Zielona Góra, Poland, 2004.
- [126] M. Kabała, R. Muszyński, M. Wnuk. Singularity robust, dynamic linearization control algorithm for MK mobile robot. J. Z. Sasiadek, I. Duleba, redaktorzy, *Robot Control 2003*, wolumen 2, strony 467–472. Elsevier, Oxford, 2003.
- [127] T. Kaczorek. *Teoria sterowania i systemów*. Wydawnictwo Naukowe PWN, Warszawa, 1999.
- [128] R. Kandepe, L. Imsland, Bjarne A. Foss. Constrained state estimation using the Unscented Kalman Filter. *Control and Automation, 2008 16th Mediterranean Conference on*, strony 1453–1458, 2008.
- [129] Karto. Software for robots on the move, SRI International. <http://www.kartorobotics.com/>, 2013.
- [130] Z. Kasap, N. Magnenat-Thalmann. Intelligent virtual humans with autonomy and personality: State-of-the-art. *New Advances in Virtual Humans*, strony 43–84, 2008.
- [131] KASPAR. KASPAR the robot. <http://kaspar.herts.ac.uk>.
- [132] J. Kędzierski. Filtr Kalmana – zastosowania w prostych układach sensorycznych, Raport Koła Naukowego Robotyków „KoNaR”, 2008.
- [133] J. Kędzierski. Robot balansujący - model i sterowanie. Praca magisterska, Politechnika Wroclawska, Wrocław, 2008.
- [134] J. Kędzierski, M. Janiak. *Budowa robota społecznego FLASH*. Prace Naukowe - Politechnika Warszawska, 2012.
- [135] J. Kędzierski, Ł. Małek, A. Oleksy. *Zastosowanie otwartego oprogramowania w systemie sterowania robotem społecznym*. Prace Naukowe - Politechnika Warszawska, 2012.
- [136] J. Kędzierski, R. Muszynski, C. Zoll, A. Oleksy, M. Frontkiewicz. Emys – Emotive Head of a Social Robot. *International Journal of Social Robotics*, 5(2):237–249, 2013.
- [137] J. Kędzierski, K. Tchoń. Feedback control of a balancing robot. *14th International Conference on Methods and Models in Automation and Robotics*, wolumen 6, strony 495–500, Oxford : Elsevier, 2010.
- [138] Keepon. Keepons and MyKeepon – the dancing robot. <http://www.keepons.co.uk>.

- [139] J. F. Kelley. An iterative design methodology for user-friendly natural language office information applications. *ACM Transactions on Information Systems*, 2(1):26–41, 1984.
- [140] A. Kendon. Movement coordination in social interaction: Some examples described. *Acta Psychologica*, 32:101–125, 1970.
- [141] G. Keren, A. Ben-David, M. Fridin. Kindergarten assistive robotics (KAR) as a tool for spatial cognition development in pre-school education. *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, strony 1084–1089, 2012.
- [142] S. Kiesler, J. Goetz. *Machine Trait Scales for Evaluating Mechanistic Mental Models of Robots and Computer-Based Machines*, 2002.
- [143] D. Kim, S. Lim, G. Han. Single-Chip Eye Tracker Using Smart CMOS Image Sensor Pixels. *Analog Integrated Circuits and Signal Processing*, 45(2):131–141, 2005.
- [144] M. Kinde. *Nonverbal Communication*, rozdział Playing with Power in Movies, Television and Video Games: From Muppet Babies to Teenage Mutant Ninja Turtles. University of California Press, Berkeley, 1991.
- [145] K. L. Koay, D. S. Syrdal, M. L. Walters, K. Dautenhahn. Living with Robots: Investigating the Habituation Effect in Participants' Preferences during a Longitudinal Human-Robot Interaction Study. *International Symposium on Robot and Human Interactive Communication*, strony 564–569. IEEE, 2007.
- [146] P. Kotler, G. Armstrong, V. Wong, J. A. Saunders. *Principles of Marketing*. Person Education, New Jersey, 2008.
- [147] Z. Kowalczyk, M. Czubenko. xEmotion - obliczeniowy model emocji dedykowany dla inteligentnych systemów decyzyjnych. *Pomiary, Automatyka, Robotyka*, 17(2):60–65, 2013.
- [148] H. Kozima, M. P. Michalowski, C. Nakagawa. Keepon: A playful robot for research, therapy, and entertainment. *International Journal of Social Robotics*, 1(1):3–18, 2009.
- [149] H. Kozima, C. Nakagawa, Y. Yasuda. Interactive robots for communication-care: a case-study in autism therapy. *IEEE International Workshop on Robot and Human Interactive Communication, ROMAN*, strony 341–346, 2005.
- [150] S. R. Kuindersma, E. Hannigan, D. Ruiken, R. A. Grupen. Dexterous mobility with the uBot-5 mobile manipulator. *International Conference on Advanced Robotics, ICRA*, strony 1–7, 2009.
- [151] R. S. Lazarus. On the primacy of cognition. *American Psychologist*, 39(2):124–129, 1984.
- [152] J. Levine. *Analysis and Control of Nonlinear Systems. A Flatness-based Approach*. Springer-Verlag, Berlin Heidelberg, 2009.
- [153] D. Levy. *Love and Sex with Robots*. NY: HarperCollins, New York, 2007.

- [154] D. Li. Low-cost eye-tracking for human computer interaction. Praca magisterska, Iowa State University, USA, 2006.
- [155] J. Lin, M. Spraragen, M. Zyda. Computational Models of Emotion and Cognition. *Advances in Cognitive Systems 2*, strony 59–76. 2012.
- [156] LIREC. Living with Robots and Interactive Companions. Projekt zrealizowany w ramach 7 Programu Ramowego, nr. 215554. <http://www.lirec.eu>, 2008-2012.
- [157] Loquendo. Loquendo - We Speak. We Listen. We Understand. <http://www.loquendo.com/en>.
- [158] D. G. Luenberger. *Introduction to dynamic systems*. John Wiley and Sons, New York, 1979.
- [159] F. K. MacDorman, N. Kageki (translated by). The Uncanny Valley By Masahiro Mori. *Robotics & Automation Magazine*, 19(2), 2012.
- [160] J. M. Maciejowski. *Predictive Control with Constraints*. Prentice-Hall, 2002.
- [161] M. Mackiewicz, J. Ciecuch. The picture-based personality survey for children – a new instrument to measure the big five in childhood. *11th European Conference on Psychological Assessment*, Poster, 2011.
- [162] Ł. Małek., M. Frontkiewicz, P. Kaczmarek, B. Komarnicki, W. Matlakiewicz, T. Mazurkiewicz, D. Piłat, M. Puchalska, A. Pyka, B. Weselak. Autorytet jako narzędzie perswazji robota społecznego. *Postępy robotyki*, wolumen 2, strony 715–724, Oficyna Wydawnicza Politechniki Warszawskiej, 2012.
- [163] N. Magnenat-Thalmann, D. Thalmann. Virtual humans: thirty years of research, what next? *The Visual Computer*, 21(12):997–1015, 2005.
- [164] MAHRU/AHRA. Samsung Electronics Co. Ltd. <http://humanoid.kist.re.kr/eng/mahruahra>, 2013.
- [165] S. Marsella, J. Gratch. Ema: A computational model of appraisal dynamics. *Cybernetics and Systems*, strony 601–606, 2006.
- [166] S. Marsella, J. Gratch, P. Petta. *A Blueprint for Affective Computing: A sourcebook and manual*. Oxford University Press, 2012.
- [167] MATHEMATICA. *version 7.0.1.0 (R2010a)*. Wolfram Research Inc., 2009.
- [168] MATLAB. *version 7.9.0 (R2009b)*. The MathWorks Inc., Natick, Massachusetts, 2009.
- [169] D. Maulsby, S. Greenberg, R. Mander. Prototyping an intelligent agent through Wizard of Oz. *Conference on Human Factors in Computing Systems, INTERACT, CHI '93*, strony 277–284, New York, NY, USA, 1993. ACM Press.
- [170] P. S. Maybeck. *Stochastic models, estimation, and control*, wolumen 141 serii *Mathematics in Science and Engineering*. 1979.
- [171] A. Mazur. Model dynamiki i kinematyki manipulatora mobilnego typu RTR, raport serii spr.

- [172] D. McNeill. *What Gestures Reveal about Thought*. The University of Chicago Press, Chicago, 1992.
- [173] A. Mehrabian, J. A. Russell. *An Approach to Environmental Psychology*. The MIT Press, Cambridge, 1974.
- [174] MEKA. Meka Robotics. <http://mekabot.com>, 2013.
- [175] G. Metta, P. Fitzpatrick, L. Natale. YARP: Yet Another Robot Platform. *International Journal on Advanced Robotics Systems*, 3(1):43–48, 2006.
- [176] S. Mithen. *The Prehistory of the Mind*, Thames and Hudson Ltd. London, 1996.
- [177] H. Miwa, T. Okuchi, K. Itoh, H. Takanobu, A. Takanishi. A new mental model for humanoid robots for human friendly communication introduction of learning system, mood vector and second order equations of emotion. *IEEE International Conference on Robotics and Automation, ICRA*, 2003.
- [178] A. A. Montgomery, P. L. Jackson. Physical characteristics of the lips underlying vowel lipreading performance. *The Journal of the Acoustical Society of America*, 73(6):2134–2144, 1983.
- [179] M. Mori. The uncanny valley. *Energy*, 7(4):33–35, 1970.
- [180] MPC555. MPC555/MPC556 User's Manual, Revised 15, Freescale Inc., 2000.
- [181] M. Namysł. Koncepcja systemu wizyjnego do rozpoznawania emocji na obrazie twarzy. *Prace Naukowe Politechniki Warszawskiej. Elektronika*, 166(1):91–100, 2008.
- [182] NAO. Aldebaran. <http://www.aldebaran-robotics.com/en>.
- [183] F. C. Nather, J. L. Oliveira Bueno, E. Bigand. Time estimation and eye-tracking movements in human body static images. *Fechner Day*, wolumen 25, 2009.
- [184] R. Negenborn. Robot localisation and Kalman filters: On finding your position in a noisy world. Praca magisterska, Utrecht University, 2003.
- [185] U. Nilsson, J. Maluszynski. *Logic, Programming, and PROLOG*. John Wiley and Sons, Inc., New York, NY, USA, wydanie 2nd, 1995.
- [186] L. Odetti, G. Anerdi, M.P. Barbieri, D. Mazzei, E. Rizza, P. Dario, G. Rodriguez, S. Micera. Preliminary experiments on the acceptability of animaloid companion robots by older people with early dementia. *29th Annual International Conference of the IEEE Engineering in Medicine and Biology Society, EMBS.*, strony 1816–1819, 2007.
- [187] S. Y. Okita, V. Ng-Thow-Hing, R. Sarvadevabhatla. Learning together: ASIMO developing an interactive learning partnership with children. *18th IEEE International Symposium on Robot and Human Interactive Communication, ROMAN*, strony 1125–1130, 2009.
- [188] A. Oleksy, R. Muszyński. Integration of Gesture Planning System with Social Robot Hand. *Prace Naukowe Politechniki Warszawskiej. Elektronika*, z. 175, t. 2:703–712, 2010.

- [189] OpenCV. OpenCV Wiki. <http://code.opencv.org/projects/opencv/wiki>, 2013.
- [190] ORIENTUS. Orientus, Advanced Navigation Pty Ltd. <http://www.xsens.com>, 2013.
- [191] A. Ortony, G. Clore, A. Collins. *The Cognitive Structure of Emotions*. Cambridge University Press, Cambridge, 1988.
- [192] E. Owens, B. Blazek. Visemes observed by hearing-impaired and normal-hearing adult viewers. *J Speech Hear Res*, 28(3):381–393, 1985.
- [193] D. W. Park, J. P. MacInnis. *Brand attachment: constructs, consequences and causes*. Publishers Inc, Hanover, 2008.
- [194] D. W. Park, J. P. MacInnis, A. B. Eisingerich, D. Iacobucci. Brand Attachment and Brand Attitude Strength: Conceptual and Empirical Differentiation of Two Critical Brand Equity Drivers. *Journal of Marketing*, 74(6), 2001.
- [195] E. Park, K. J. Kim, A. P. del Pobil. Facial Recognition Patterns of Children and Adults Looking at Robotic Faces. *International Journal of Advanced Robotic Systems*, 2012.
- [196] PARO. Paro Therapeutic Robot. <http://www.parorobots.com>, 2013.
- [197] C. Pelachaud. Modelling multimodal expression of emotion in a virtual agent. *Philosophical Transactions B*, 364(12), 2009.
- [198] J. Perchonok, P. E. Green. *Facilitating Driver Interaction with a Robotic Driving Assistant: Some Insights from the Literature*. University of Michigan Transportation Research Institute, 2009.
- [199] P. Petta, R. Trappl, redaktorzy. *Creating Personalities for Synthetic Actors, Towards Autonomous Personality Agents*, wolumen 1195 serii *Lecture Notes in Computer Science*. Springer, 1997.
- [200] PLAYER. The Player Project. <http://playerstage.sourceforge.net>, 2013.
- [201] PLEO. Innvo Labs. <http://www.pleoworld.com>, 2013.
- [202] R. Plutchik. *Emotion. A psychoevolutionary synthesis*. New York: Harper & Row, 1980.
- [203] PRG. Personal Robots Group, mit media lab. <http://robotic.media.mit.edu>, 2013.
- [204] S Prillwitz, R Leven, H Zienert, T Hanke, J Henning. *HamNoSys. Version 2.0. Hamburg Notation System for Sign Language. An Introductory Guide*. Signum, Hamburg, 1992.
- [205] PrimeSense. Primesense 3D sensing technology. <http://www.primesense.com>, 2013.
- [206] M. Quigley, B. Gerkey, K. Conley, J. Faust, T. Foote, J. Leibs, E. Berger, R. Wheeler, A. Y. Ng. Ros: an open-source Robot Operating System. *Open-Source Software workshop at the International Conference on Robotics and Automation, ICRA*, 2009.

- [207] A. Ratajczak. *Planowanie ruchu układów robotycznych z deficytem napędów*. Praca doktorska, Politechnika Wrocławska, Wrocław, 2011.
- [208] A. Ratajczak, K. Tchon. Motion planning of a balancing robot with threefold sub-tasks: An endogenous configuration space approach. *IEEE International Conference on Robotics and Automation, ICRA*, strony 6096–6101, Shanghai, Chiny, 2011.
- [209] J.B. Rawlings, D.Q. Mayne. *Model Predictive Control: Theory and Design*. Nob Hill Publishing, 2009.
- [210] RE-MAX. DC Maxon Motor. <http://www.maxonmotor.com>, 2013.
- [211] REEM-C. PAL Robotics. <http://www.pal-robotics.com>, 2012.
- [212] T. Ribeiro, I. Leite, J. Kędzierski, A. Oleksy, A. Paiva. Expressing Emotions on Robotic Companions with Limited Facial Expression Capabilities. *11th International Conference on Intelligent Virtual Agents*, wolumen 6895/2011 serii *Lecture Notes in Computer Science*, strony 466–467, Eindhoven, The Netherlands, 2011. Springer Berlin, Heidelberg.
- [213] T. Ribeiro, A. Paiva. The illusion of robotic life: principles and practices of animation for robots. *Seventh annual ACM/IEEE international conference on Human-Robot Interaction, HRI '12*, strony 383–390, New York, NY, USA, 2012. ACM Press.
- [214] T. Ribeiro, M. Vala, A. Paiva. Thalamus: Closing the Mind-Body Loop in Interactive Embodied Characters. *Intelligent Virtual Agents IVA '2012*, wolumen 7502 serii *Lecture Notes in Computer Science*, strony 189–195, Eindhoven, The Netherlands, 2012. Springer Berlin / Heidelberg.
- [215] B. Robins, K. Dautenhahn. Developing Play Scenarios for Tactile Interaction with a Humanoid Robot: A Case Study Exploration with Children with Autism. ShuzhiSam Ge, Haizhou Li, John-John Cabibihan, YeowKee Tan, redaktorzy, *Social Robotics*, wolumen 6414 serii *Lecture Notes in Computer Science*, strony 243–252. Springer Berlin Heidelberg, 2010.
- [216] B. Robins, K. Dautenhahn, R. Boekhorst, A. Billard, S. Keates, J. Clarkson, P. Langdon, P. Robinson. Effects of repeated exposure of a humanoid robot on children with autism. *Designing a More Inclusive World*, strony 225–236. Springer-Verlag, 2004. Sponsor: Swiss National Science Foundation.
- [217] Robolink. IGUS Inc. <http://www.igus.eu>, 2013.
- [218] ROBONAUT. NASA GM. <http://www.nasa.gov/topics/technology/features/robonaut.html>, 2013.
- [219] ROBOTHESPIAN. Engineered Arts Limited. <http://www.robothespian.co.uk>, 2013.
- [220] I. J. Roseman, C. A. Smith. *Appraisal Theory: Overview, Assumptions, Varieties, Controversies*. Oxford University Press, NY, 2001.
- [221] I. J. Roseman, M. S. Spindel, P. E. Jose. By Appraisals of emotion-eliciting events: Testing a theory of discrete emotions. *Journal of Personality and Social Psychology*, 59(5):899–915, 1990.

- [222] M. Saerbeck, T. Schut, C. Bartneck, M. D. Janse. Expressive robots in education: varying the degree of social supportive behavior of a robotic tutor. *SIGCHI Conference on Human Factors in Computing Systems, CHI '10*, strony 1613–1622, New York, NY, USA, 2010. ACM Press.
- [223] SAPI. Speech SDK 5.3, Microsoft. <http://www.microsoft.com>, 2013.
- [224] K. R. Scherer, A. Schorr, T. Johnstone. *Appraisal Processes in Emotion: Theory, Methods, Researchs*. Oxford University Press, 2001.
- [225] S. Schomaker, S. Muench, K. Hartung. A Taxonomy of Multimodal Interaction in the Human Information Processing System: Report of the Esprit Project 8579 MIAMI. Raport instytutowy <http://hwr.nici.kun.nl/%7Emiami/>, Esprit/BRA, NICI, Nijmegen, 1995.
- [226] SDL. Simple DirectMedia Layer. <http://www.libsdl.org>.
- [227] SFML. Simple and Fast Multimedia Library, Laurent Gomila. <http://www.sfml-dev.org/license.php>, 2013.
- [228] S. Shamsuddin, H. Yussof, L. Ismail, F.A. Hanapiah, S. Mohamed, H.A. Piah, N. Ismarrubie Zahari. Initial response of autistic children in human-robot interaction therapy with humanoid robot NAO. *IEEE 8th International Colloquium on Signal Processing and its Applications (CSPA)*, strony 188–193, 2012.
- [229] SMI. Mobile eye-tracker, SensoMotoric Instruments Inc. <http://www.smivision.com>, 2013.
- [230] C. A. Smith, R. Lazarus. Emotion and Adaptation. *Handbook of Personality: theory and research*, strony 609–637. Guilford Press, NY, 1990.
- [231] E. D. Sontag. *Mathematical Control Theory: Deterministic Finite Dimensional Systems, Second Edition*. Springer, New York, 1998.
- [232] M. W. Spong. Underactuated mechanical systems. Bruno Siciliano, KimonP. Valavanis, redaktorzy, *Control Problems in Robotics and Automation*, wolumen 230 serii *Lecture Notes in Control and Information Sciences*, strony 135–150. Springer Berlin Heidelberg, 1998.
- [233] A. Steinfeld, O. Chadwicke Jenkins, B. Scassellati. The oz of wizard: simulating the human for interaction research. *4th ACM/IEEE international conference on Human robot interaction, HRI '09*, strony 101–108, New York, NY, USA, 2009. ACM Press.
- [234] Stellaris. Texas Instruments Inc. <http://www.ti.com>, 2013.
- [235] M. Stilman, J. Olson, W. Gloss. Golem Krang: Dynamically stable humanoid robot for mobile manipulation. *IEEE International Conference on Robotics and Automation, ICRA*, strony 3304–3309, 2010.
- [236] SURENA-II. Advanced Dynamic and Control Systems Laboratory, Tehran University. <http://www.adcslab.com>, 2012.

- [237] F. Tanaka, S. Matsuzoe. Children Teach a Care-Receiving Robot to Promote Their Learning: Field Experiments in a Classroom for Vocabulary Learning. *Journal of Human-Robot Interaction*, 1(1), 2012.
- [238] P. Tatjewski. *Sterowanie zaawansowane obiektów przemysłowych: struktury i algorytmy*. Monografie - Komitet Automatyki i Robotyki Polskiej Akademii Nauk. Akademicka Oficyna Wydawnicza Exit, 2002.
- [239] K. Tchoń, J. Jakubiak. Endogenous configuration space approach to mobile manipulators: A derivation and performance assessment of jacobian inverse kinematics algorithms. *International Journal of Control*, 76(14):1387–1419, 2003.
- [240] K. Tchoń, A. Mazur, I. Duleba, R. Hossa, R. Muszyński. *Manipulatory i roboty mobilne: Modele, planowanie ruchu i sterowanie*. Akademicka Oficyna Wydawnicza, Warszawa, 2000.
- [241] R. Tedrake. *Underactuated Robotics: Learning, Planning, and Control for Efficient and Agile Machines*. Course Notes, MIT, 2009.
- [242] F. Thomas, O. Johnston. *Disney Animation: The Illusion of Life*. Abbeville Press, New York, 1981.
- [243] Bernard Tiddeman, David Perrett. Prototyping and transforming visemes for animated speech. *Computer Animation Conference Proceedings*, strony 248–251, 2002.
- [244] TIRO. Hanoor Robotics Corp. <http://www.tiro.co.kr>, 2013.
- [245] N. A. Torres, N. Clark, I. Ranatunga, D. Popa. Implementation of interactive arm playback behaviors of social robot Zeno for autism spectrum disorder therapy. *5th International Conference on Pervasive Technologies Related to Assistive Environments*, PETRA '12, strony 1–7, New York, NY, USA, 2012. ACM Press.
- [246] TOYOTA. Toyota Motor Corporation, partner Robot Technology. http://www.toyota-global.com/innovation/partner_robot, 2013.
- [247] R. Trappl, P. Petta, S. Payr. *Emotions in Humans and Artifacts*. MIT Press, Cambridge, 2003.
- [248] TUBO. Department of Robot System Engineering, Tongmyong University. <http://robot.tu.ac.kr>, 2013.
- [249] A. M. Turing. Computing Machinery and Intelligence. *Mind* 49, 59(236):433–460, 1950.
- [250] Twendy-One. Sugano Laboratory, WASEDA University. <http://twendyone.com>, 2013.
- [251] J. Uhlmann. *Dynamic Map Building and Localization: New Theoretical Foundations*. Praca doktorska, University of Oxford, 1995.
- [252] URBI. The Urbi Software Development Kit. GOSTAI. <http://www.urbiforge.org>, 2013.
- [253] URBI. The Urbi Software Development Kit – documentation, Revision 2.7.5, GOSTAI. <http://www.urbiforge.org/index.php/Main/Docs>, styczeń, 2012.

- [254] VCS. Measuring PARO's impact, with an emphasis on residents with alzheimer's disease or dementia. *Vincentian Collaborative System*, 2010.
- [255] L. S. Vygotsky. *Mind and society: The development of higher mental processes*. MA: Harvard University Press, Cambridge, 1978.
- [256] K. Wada, T. Shibata, T. Saito, K. Tanie. Effects of robot-assisted activity for elderly people and nurses at a day service center. *IEEE*, 92(11):1780–1788, 2004.
- [257] K. Wada, T. Shibata, T. Saito, K. Tanie. Psychological and social effects in long-term experiment of robot assisted activity to elderly people at a health service facility for the aged. *IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS*, wolumen 3, strony 3068–3073, 2004.
- [258] J. Wainer, K. Dautenhahn, B. Robins, F. Amirabdollahian. Collaborating with Kaspar: Using an autonomous humanoid robot to foster cooperative dyadic play among children with autism. *10th IEEE-RAS International Conference on Humanoid Robots (Humanoids)*, strony 631–638, 2010.
- [259] WAKAMARU. Mitsubishi Heavy Industries Ltd. <http://www.mhi.co.jp/en/products/detail/wakamaru.html>, 2013.
- [260] M. L. Walters. *The Design Space for Robot Appearance and Behaviour for Social Robot Companions*. Praca doktorska, University of Hertfordshire, Hatfield, 2008.
- [261] M. L. Walters, D. S. Syrdal, K. Dautenhahn, I. René J. A. te Boekhorst, K. L. Koay. Avoiding the uncanny valley: robot appearance, personality and consistency of behavior in an attention-seeking home scenario for a robot companion. *Autonomous Robots*, 24(2):159–178, 2008.
- [262] E. A. Wan, R. van der Merwe. *The Unscented Kalman Filter*. John Wiley and Sons, Inc., 2002.
- [263] WASEDA. Takanishi Laboratory, Waseda University. <http://www.takanishi.mech.waseda.ac.jp/top/research/index.htm>, 2012.
- [264] G. Welch, G. Bishop. An introduction to the Kalman Filter, Technical Report, University of North Carolina at Chapel Hill Chapel Hill, NC, USA, 1995.
- [265] W. M. Wonham. *Linear multivariable control: a geometric approach*. Applications of mathematics. Springer-Verlag, 1985.
- [266] S. N. Woods, K. Dautenhahn, J. Schulz. The Design Space of Robots: Investigating Children's Views. *13th IEEE International Workshop on Robot and Human Interactive Communication, ROMAN*, strony 47–52, Kurashiki, Japonia, 2004.
- [267] XSENS. MTi-30 AHRS: Attitude and Heading Reference System, Xsens Technologies. <http://www.xsens.com>, 2013.
- [268] L. A. Zebrowitz, J. A. Hall, N. A. Murphy, G. Rhodes. Looking Smart and Looking Good: Facial Cues to Intelligence and Their Origins. *Personality and Social Psychology Bulletin*, 28(2):238–249, 2004.
- [269] ZENO. Hanson Robokind. <http://hansonrobokind.com>.

- [270] C. Zieliński, T. Winiarski. Motion Generation in the MRROC++ Robot Programming Framework. *The International journal of robotics research*, 29(4):386–413, 2010.
- [271] C. Zieliński, T. Winiarski, W. Szykiewicz, T. Kornuta, P. Trojanek. MRROC++ - programowa struktura ramowa do tworzenia sterowników systemów wielorobotowych. S. Ambroszkiewicz, A. Borkowski, K. Cetnarowicz, C. Zieliński, redaktorzy, *Inteligencja wśród nas. Współdziałanie agentów softwareowych, robotów, inteligentnych urządzeń*, strony 317–384. AOF EXIT, Warszawa, 2010.