



AKADEMIA GÓRNICZO-HUTNICZA IM. STANISŁAWA STASZICA W KRAKOWIE

DZIEDZINA NAUK INŻYNIERYJNO-TECHNICZNYCH

DYSCYPLINA AUTOMATYKA, ELEKTRONIKA, ELEKTROTECHNIKA I
TECHNOLOGIE KOSMICZNE

ROZPRAWA DOKTORSKA

Nowe metody rozpoznawania gestów 3D bez użycia
kamer, w aplikacjach w branży motoryzacyjnej

Autor: Piotr Reczek

Promotor rozprawy: dr hab. inż. Andrzej Młyniec, prof. AGH

Promotor pomocniczy: dr hab. inż. Andrzej Wetula

Praca wykonana: Akademia Górniczo-Hutnicza im. Stanisława
Staszica w Krakowie

Wydział Inżynierii Mechanicznej i Robotyki

Katedra Robotyki i Mechatroniki

Kraków, 2023



AGH UNIVERSITY OF SCIENCE AND TECHNOLOGY

FIELD OF SCIENCE: ENGINEERING AND TECHNOLOGY

SCIENTIFIC DISCIPLINE: AUTOMATION, ELECTRONIC, ELECTRICAL
ENGINEERING AND SPACE TECHNOLOGIES

DOCTORAL THESIS

*New methods of 3D gesture recognition in
automotive industry applications*

Author: Piotr Reczek

First supervisor: dr hab. inż. Andrzej Młyniec, prof. AGH

Assisting supervisor: dr hab. inż. Andrzej Wetula

Completed in: AGH University Kraków

The Faculty of Mechanical Engineering and Robotics

Department Robotics and Mechatronics

Krakow, 2023

Streszczenie

Niniejsza praca przedstawia nowe metody rozpoznawania gestów 3D bez użycia kamer, w aplikacjach w branży motoryzacyjnej. W ramach realizacji pracy, dokonano przeglądu literatury oraz istniejących rozwiązań, które mogłyby zostać zastosowane do rozpoznawania gestów w środowisku wnętrza pojazdu. Następnie przeanalizowano istniejące systemy w następujących kategoriach: Systemy wizyjne, Systemy wykorzystujące elektrody pojemnościowe, Systemy oparte o czujniki podczerwieni, Systemy radarowe, Systemy ultradźwiękowe, Systemy zlokalizowane na ręce. Każdą z zaproponowanych technologii poddano krytycznej analizie z perspektywy wymagań dla systemów do rozpoznawania gestów z potencjalnym zastosowaniem w branży samochodowej. Dodatkowo, każdą z nich przeanalizowano pod kątem wykorzystania do utworzenia nowego rozwiązania wykluczając z założenia systemy wizyjne. Na zakończenie przeglądu literatury omówiono fuzję danych oraz klasyfikację metod wykorzystywanych do jej przeprowadzania. W rozdziałach „System do rozpoznawania gestów oparty o czujniki pojemnościowe” oraz „System do rozpoznawania gestów oparty o radar”, przedstawiono autorskie rozwiązania, w tym zastosowaną architekturę sprzętową, opracowane algorytmy oraz sposoby działania obu systemów. Rozwiązania te zostały stworzone w ramach realizacji projektu wdrożeniowego w firmie Merit. W rozdziale traktującym o systemie pojemnościowym zaproponowano również sposoby rozszerzania danych uczących dla przebiegów sygnału pojemnościowego z poszczególnych elektrod. W końcowej części rozdziału opisano proces konwersji modelu sieci neuronowej oraz uruchomienia go na mikrokontrolerze. W kolejnym podrozdziale opisana została implementacja systemu działającego w oparciu o czujniki pojemnościowe, który został zamontowany w samochodzie testowym, co stanowiło wdrożeniowy charakter pracy. Na zakończenie rozdziału, przedstawiono nowatorską metodę automatyzacji akwizycji danych uczących dla potrzeb procesu treningu sieci neuronowej przy użyciu 4 osiowego manipulatora, który zastąpił ludzi wykonujących gesty. W rozdziale dotyczącym systemu używającego radaru fali ciągłej z modulacją częstotliwościową, oprócz opisu konstrukcji radaru oraz jego sprzętowej konfiguracji, przedstawiono zastosowane metody przetwarzania danych otrzymywanych bezpośrednio z radaru do postaci cech, które stanowiły wejście do sieci neuronowej klasyfikującej gesty. W ostatnim rozdziale zostały przedstawione wyniki skuteczności rozpoznawania gestów dla obu opisanych rozwiązań, a także po dokonaniu fuzji danych pochodzących z każdego z nich. Sieć neuronowa wykorzystana do syntezy danych, przyjmowała

rozszerzony wektor wejściowy obejmujący 10 cech w postaci 5 sygnałów z czujnika pojemnościowego, 2 przebiegów określających prędkość obiektu względem anten radaru, 2 przebiegów informujących o odległości wykrywanego obiektu od każdej z anten oraz położenia kąтового wykrywanego obiektu w płaszczyźnie wertykalnej radaru. Przeprowadzone eksperymenty wykazały, że modele sieci neuronowej rozpoznające zarówno 5 oraz 7 gestów, dzięki zastosowaniu fuzji danych, osiągnęły ponad 97 procentową skuteczność ich rozpoznawania, podczas gdy sieci neuronowe wykorzystujące dane tylko z jednego typu sensora (elektrod pojemnościowych i radaru) odpowiednio dla 5 gestów 93% i 94,2% oraz dla 7 gestów 87,8% i 92,5%. Otrzymane rezultaty dowiodły postawionej tezie, że fuzja danych z obu systemów spowoduje wzrost skuteczności wykrywanych gestów. W ostatnim rozdziale przedstawiono obszary, których udoskonalenie powinno zostać rozważone pod kątem dalszego rozwoju przedstawionych systemów, zarówno w kontekście wykorzystania osobno jak i w kontekście ich fuzji.

Streszczenie w języku angielskim

This doctoral thesis presents new methods for 3D gestures recognition without using cameras in automotive industry applications. As part of the work, literature and existing solutions were researched on the areas of potential application in car interior environment. Subsequently, previously existing systems were analysed and divided into following categories: Visual systems, Capacitive electrodes systems, Infrared sensors systems, Radar systems, Ultrasonic systems, Systems mounted on hand. All proposed technologies were subjected to analysis from the perspective of system requirements for gesture recognition and potential application in automotive industry. Additionally, all of them were investigated in terms of using them as a new solution excluding visual systems due to thesis assumption. At the end of the literature review, there is the chapter explaining data fusion term and classification of methods used for its execution. Chapters "System for gesture recognition based on capacitive sensors" and "System for gesture recognition based on radar" present authorial solutions, including applied hardware architecture, developed algorithms and principles of operations for both systems. These solutions were created as part of implementation doctorate program in cooperation with Merit Poland company. In the chapter dealing with Capacitive system, there were proposed techniques for augmentation of data for signals received from each capacitive electrode. The final part of the chapter describes process of the model conversion and its execution on microcontroller. In the next subchapter, implementation of the capacitive system in the test car was presented, as a part of implementational characteristics of the work. At the end of the chapter, the novel method to collect data for neural network model training process was presented. The method assumes using a 4-axis robotic arm manipulator, which replaced people in performing gestures. In the chapter dealing with the system using frequency modulated continuous wave (FMCW) radar, despite of description of the radar structure and its hardware configuration, methods of processing a raw data received directly from the radar to the features was presented. These features constituted input to the neural network, which classified gestures. In the chapter "Data fusion for capacitive and radar systems", the results of gesture recognition accuracy for both solutions were presented. Additionally, accuracy after fusing data from the two systems was exposed. The neural network used for data synthesis, received an extended input vector including ten features. For the first five features' signals were obtained from the capacitive sensor. The next five features were gained from the radar sensor. That was two waveforms representing speed of the object referring to each of

antennas, two waveforms representing range of the object from antennas, and the last waveform reflecting angular position of the object in a vertical plane. Executed experiments demonstrated that, applying data fusion in neural network models recognizing 5 and 7 different gestures, resulted with over 97% accuracy of gestures recognition. Whereas neural networks using data from one type of sensor (capacitive and radar) resulted with the accuracy 93 and 94,2 % for 5 gestures and 87,8 and 92,5% for 7 gestures. The results have proven the assessment, that fusion on the data from both systems will cause increase of the gesture recognition accuracy. In the last chapter, areas for improvement and further investigations were highlighted within the context of data fusion and operation of systems separately.

Podziękowania

Składam serdeczne podziękowania mojemu promotorowi dr hab. inż. Andrzejowi Młyńcowi prof. AGH za wsparcie merytoryczne, zaangażowanie i życzliwość podczas realizacji niniejszego doktoratu. Pragnę wyrazić również głębokie podziękowania dla promotora pomocniczego dr hab. inż. Andrzeja Wetuli za inspirację, nieocenioną pomoc i sugestie, które wpłynęły na ostateczny kształt mojej rozprawy.

Dziękuję również firmie Merit Poland, gdzie miałem możliwość łączenia mojej pracy zawodowej z realizacją doktoratu. W szczególności, moim długoletnim przełożonym: Sylwestrowi Skrzypkowi oraz Łukaszowi Jurankowi za przychylność oraz otwartą postawę wobec moich pomysłów. Serdeczne podziękowania składam również kolegom, którzy współpracowali ze mną w obszarach związanych z moją pracą doktorską – Jakubowi Pańczykowi oraz Dominikowi Stachurze.

Składam szczerze podziękowania moim Rodzicom, za wsparcie na całej mojej drodze edukacji.

Niniejszą rozprawę dedykuję mojej żonie Agnieszce i synowi Stanisławowi.

Spis treści

Streszczenie	3
Streszczenie w języku angielskim	5
Podziękowania.....	7
1. Wstęp	10
2. Przegląd literatury oraz istniejących rozwiązań	12
2.1. Systemy wizyjne	12
2.2. Systemy wykorzystujące elektrody pojemnościowe	15
2.3. Systemy oparte o czujniki podczerwieni	21
2.4. Systemy radarowe	24
2.5. Systemy ultradźwiękowe	29
2.6. Systemy zlokalizowane na ręce	31
2.7. Fuzja danych	34
2.7.1. Zdefiniowanie pojęcia fuzji danych	34
2.7.2. Klasyfikacja metod fuzji danych.....	36
3. Cel i zakres pracy	46
4. System do rozpoznawania gestów oparty o czujniki pojemnościowe	48
4.1. Elektrody oraz układ sterujący MGC	48
4.2. Zabudowa elektrod.....	52
4.2.1. Mikroprocesor główny	53
4.2.2. Algorytm	53
4.3. Model sieci neuronowej – system pojemnościowy	55
4.4. Akwizycja Danych	58
4.5. Rozszerzanie zbioru danych.....	59
4.6. Proces Uczenia.....	63
4.7. Konwersja modelu dla potrzeb uruchomienia na mikrokontrolerze	68
4.8. Testy i implementacja systemu w samochodzie	70
4.9. Wykorzystanie manipulatora do testowania oraz uczenia	71
5. System do rozpoznawania gestów oparty o radar	76
5.1. Moduł BGT60TR13C - Antena in Package.....	77
5.1.1. Układ Anten	77
5.1.2. Układ radaru	80
5.2. Sposób wykrywania odległości i prędkości.....	82
5.3. Dobór i konfiguracja radaru pod kątem wykrywania gestów	86

5.3.1.	Wpływ parametrów na sposób pracy radaru.....	87
5.4.	Radar Fusion GUI Infineon	88
5.5.	Algorytmy wykorzystane do przetwarzania wstępnego	89
5.6.	Model sieci neuronowej – system radarowy	101
6.	Fuzja danych z systemów pojemnościowego oraz radarowego	103
6.1.	Zaproponowane podejście do fuzji danych.....	103
6.2.	Wyniki fuzji danych dla systemów pojemnościowego oraz radarowego.....	108
7.	Podsumowanie oraz możliwe dalsze kierunki rozwoju.....	110
8.	Bibliografia.....	112

1. Wstęp

Komunikacja poprzez gesty stanowi znaczący obszar niewerbalnych sposobów porozumiewania się. Jest uniwersalna, czego można doświadczyć obserwując porozumiewanie się osób rozmawiających w zupełnie różnych językach. Odpowiednia mowa ciała może czasami przekazywać informacje bardziej precyzyjnie niż komunikaty słowne. Fakt ten ma zastosowanie nie tylko u ludzi. Swoje obserwacje dotyczące wyrażania emocji przez ssaki za pomocą gestów oraz mowy ciała opisywał już C. Darwin [1]. Z codziennego doświadczenia wiemy, że przekazanie informacji na temat położenia okazuje się często łatwiejsze przy użyciu gestów niż słownie. Wobec powyższych obserwacji można założyć, że system wykrywający odpowiednie gesty może stanowić intuicyjny interfejs dla człowieka, w szczególności mając na uwadze rosnący stopień skomplikowania systemów teleinformatycznych otaczających ludzi w ich przestrzeni życiowej. Niejednokrotnie użytkownicy wszelakich systemów czują się zagubieni, przez co bez wstępnego instruktarzu nie są w stanie wykorzystać zaawansowanych systemów nawet w podstawowym stopniu. Dobrym przykładem potwierdzającym tę obserwację są nowoczesne samochody, które oferują niezliczoną ilość funkcji. Niestety stopień ich skomplikowania może okazać się przytłaczający, przez co użytkownicy rezygnują z korzystania z części dostępnych funkcji.

Największy odsetek systemów wykrywających gesty stanowią systemy wizyjne. Wydaje się to naturalne, ponieważ jest to sposób najbardziej zgodny z ludzkim sposobem rozpoznawania gestów. Większość systemów opisanych w literaturze z lat 90 oraz przełomu wieków bazuje na przetwarzaniu obrazów [2] [3]. W tym okresie znacząco zwiększa się ilość publikacji traktujących o rozpoznawaniu gestów. Dominującą metodą jest zastosowanie szeroko pojętego uczenia maszynowego. Efektywność tego procesu jest ściśle związana z mocą obliczeniową. Stąd można wywnioskować, że dostępność mocy obliczeniowej umożliwiła przełom w tej dziedzinie.

Pomimo popularności systemów wizyjnych w tego typu aplikacjach jako pierwszy system do detekcji gestów najczęściej wymieniany jest skonstruowany w roku 1977, DataGlove stworzony przez Electronic Visualization Laboratory [4] [5]. Zgodnie z nazwą system stanowił rękawicę zakładaną na rękę użytkownika. Przez rękawicę przeprowadzono elastyczne rurki, które przewodziły światło. Ich zginanie powodowało zmniejszenie ilości przepuszczanego światła. Na podstawie napięcia z fotokomórek umieszczonych na zakończeniach rurek, możliwe było zinterpretowanie który staw został zgięty. Układ pomiarowy

w kontekście topografii naśladuje układ wykonawczy w postaci ściągien umieszczonych w dłoni.

2. Przegląd literatury oraz istniejących rozwiązań

2.1. Systemy wizyjne

Pierwszym przykładem wykorzystania systemów wizyjnych do wykrywania gestów w środowisku samochodowym, jest rozwiązanie zaprezentowane w publikacji z 2000 roku [6]. Współautorami publikacji są dwaj pracownicy BMW. Opisany system wykorzystywał ukryte modele Markowa (HMM), uruchomione na jednostce obliczeniowej w postaci procesora Pentium-II o taktowaniu 333MHz. Zastosowano „lekko zmodyfikowaną” [6] kamerę CCD (charge-coupled device) o rozdzielczości 192x144 pikseli. Według publikacji, autorom udało się uzyskać 90,3 % skuteczności na zestawie 20 gestów oraz 98,1 % na zestawie składającym się z 6 gestów. W podsumowaniu artykułu autorzy jako największy problem opisywanego systemu przytoczyli padanie światła słonecznego w obszar roboczy kamery. Powodowało to brak odpowiedniego poziomu kontrastu analizowanego obrazu. Problem ten został rozwiązany w przedstawionym rozwiązaniu komercyjnym dzięki zastosowaniu własnego podświetlenia, prawdopodobnie wąskopasmowego i z modulacją. Jako potencjalne rozwiązanie autorzy sugerują wykorzystanie „distance images”, czym de facto jest technologia ToF.

Prace nad rozwojem oraz wdrożeniem systemu do produkcji seryjnej trwały przynajmniej kilkanaście lat, czego dowodzi system dostępny od 2015 roku w pojazdach marki BMW [7]. Został on opracowany we współpracy z krakowskim oddziałem firmy Aptiv [8]. Kamera Time of Flight (ToF) zamontowana jest w module dachowym, umieszczonym w przedniej części podsufitki i zawierającym poza kamerą lampy oświetlenia kabiny, przyciski wzywania pomocy czy opcjonalne przełączniki do kontroli szyberdachu. Obszar roboczy obejmuje zakres tunelu środkowego oraz obszaru przed głównym, centralnym ekranem - rysunek 1.



Rysunek 1. Obszar roboczy kamery w systemie do rozpoznawania gestów [8]

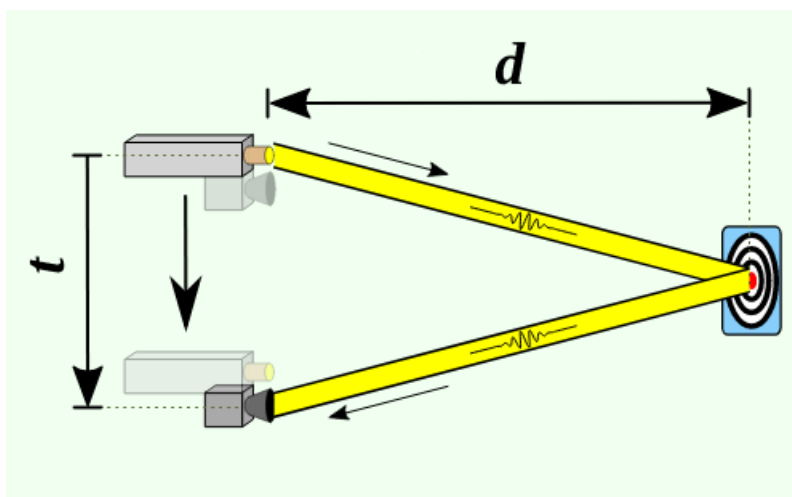
Kamera pracuje w bliskiej podczerwieni, korzystając z umieszczonych wokół niej elementów podświetlających (diod LED). Dzięki temu kamera jest w stanie pracować nawet w słabych warunkach oświetleniowych. Obraz jest przetwarzany w czasie rzeczywistym. Dane w postaci klatek obrazu, przedstawiających chwilowe stany przestrzeni roboczej kamery są przesyłane do jednostki obliczeniowej. Z informacji podanych do publicznej wiadomości wynika, że wykorzystuje ona algorytmy głębokiego uczenia maszynowego.

Analizując podejście do rozwoju tego typu rozwiązań, należy zwrócić uwagę na fakt, że w opisywanym przypadku, od początku starano się testować rozwiązanie w docelowym środowisku pracy. Dzięki temu możliwe było wcześniejsze znalezienie krytycznych problemów, co pozwoliło zmodyfikować koncept w celu ich eliminacji. Jest to istotna wskazówka dla prac badawczych z kontekstem wdrożeniowym. Dodatkowo można zauważyć, że od pierwszych badań nad wspomnianym systemem minęło minimum 15 lat zanim firma BMW zaaplikowała system w samochodzie oferowanym w seryjnej produkcji. Prawdopodobnie wynikało to z konieczności pokonania wykrytych problemów. Ponadto istotnym faktem jest znaczne obniżenie kosztów niezbędnych komponentów elektronicznych w stosunku do ceny samochodu, co jest bardzo istotnym czynnikiem przy masowej produkcji. W tabeli 1 porównano najważniejsze cechy każdego z rozwiązań.

Tabela 1. Porównanie wybranych cech systemów do rozpoznawania gestów w fazie badawczej oraz wdrożonego

Kryterium	System w wersji prototypowej [6]	System w wersji produkcyjnej [8]
Typ Kamery	Kamera CCD 192 x 144	Kamera ToF
Algorytm rozpoznawania gestów	Ukryte modele Markova (Hidden Markov Models)	Głębokie Uczenie Maszynowe (Deep Machine Learning)

Rozszerzając zagadnienie zastosowań systemów wizyjnych do rozpoznawania gestów, poza branżą samochodową, należy zwrócić uwagę na fakt, że większość opisywanych rozwiązań wykorzystuje moduł Microsoft Kinect [9] [10] [11]. System składa się z dwóch kamer - RGB oraz Time of Flight. Pierwsza z nich wykorzystywana jest do rejestracji obrazu - kształtów oraz kolorów. Druga odpowiada za dostarczenie informacji na temat odległości obserwowanych obiektów. Zasada działania kamery Time of Flight polega na pomiarze czasu w jakim wiązka promieniowania elektromagnetycznego (w zakresie podczerwieni 780 - 1000nm) pokonuje odległości kolejno między źródłem promieniowania a obiektem oraz po odbiciu między obiektem a matrycą kamery. Sposób działania kamery przedstawiono na rysunku 2.



Rysunek 2. Zasada działania kamery Time of Flight

Zmierzenie czasu oraz wykorzystanie znajomości wartości prędkości światła pozwala wyznaczyć odległość od odbitego obiektu zgodnie z równaniem (1).

$$d = \frac{c t}{2} \quad (1)$$

d – odległość od obiektu

t – zmierzony czas

c – prędkość światła

Warto zwrócić uwagę na rząd wielkości mierzonych wielkości czasu. Zakładając wykrywanie obiektu oddalonego o 50cm od kamery promień podczerwieni pokonuje drogę tam i z powrotem w zaledwie 1,7ns.

Bezpośrednia adaptacja systemu Kinect nie jest możliwa w zastosowaniu w środowisku samochodu, ze względu na wymagania temperaturowe.

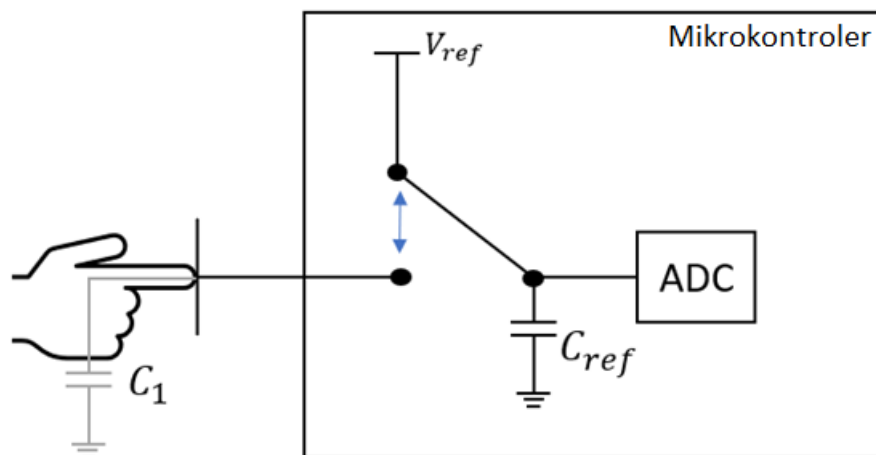
2.2. Systemy wykorzystujące elektrody pojemnościowe

Technologia pojemnościowa ma bardzo szerokie zastosowanie, głównie w ekranach dotykowych, co sprawia, że na świecie istnieją miliardy urządzeń, które ją stosują. Jedną z przyczyn tego stanu rzeczy jest stosunkowo niski koszt implementacji. Zasada działania tego typu systemów polega na dokonywaniu pomiarów zmian pojemności, które są powodowane przez zmiany w polu elektrycznym w przestrzeni wokół elektrody. Zmiany te są zazwyczaj spowodowane pojawianiem się obiektów przewodzących w pobliżu elektrod. Systemy pojemnościowe ze względu na koncepcję wykonywania pomiaru dzielą się na dwie kategorie: pojemności własnej i pojemności wzajemnej (ang. self capacitance and mutual capacitance). Rozważmy pojedynczy przycisk dotykowy.

W przypadku pojemności własnej, wystarczy tylko jedna elektroda do jego realizacji. Pełni ona rolę jednocześnie elektrody nadawczej oraz odbiorczej. Elektroda musi być podłączona do przetwornika analogowo cyfrowego oraz do układu umożliwiającego pojawienie się na niej określonego napięcia. W praktyce najczęściej ścieżka łącząca elektrodę z drugiej strony jest połączona z nóżką mikrokontrolera, który ma możliwość pracy w trybie analogowym i cyfrowym. Dodatkowo konieczne jest zastosowanie kondensatora referencyjnego o znanej stałej wartości. W rozwiązaniach mikroprocesorowych przeznaczonych dla systemów dotykowych, kondensator ten znajduje się wewnątrz układu scalonego. Wykrywanie zmian

pojemności odbywa się przez porównywanie ładunku zgromadzonego w kondensatorze referencyjnym oraz w wirtualnym – stworzonym przez ludzkie ciało. Pomiar pojemności odbywa się w następujący sposób:

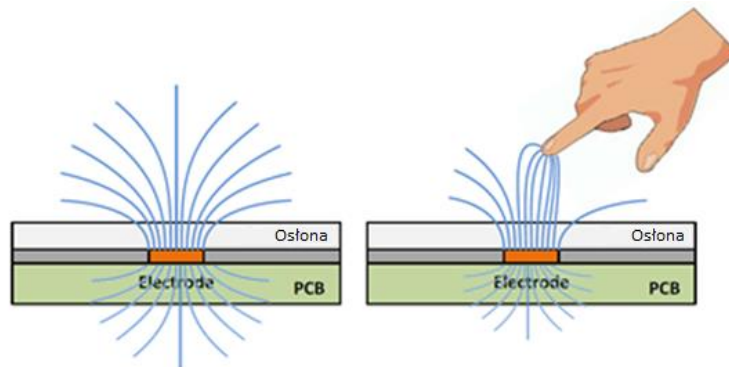
1. Nóżka mikrokontrolera połączona z elektrodą jest połączona z masą, następuje ładowanie kondensatora wewnętrznego.
2. Po naładowaniu kondensatora referencyjnego i osiągnięciu napięcia referencyjnego, nóżka połączona z elektrodą podłączona jest do wejścia przetwornika analogowego cyfrowego oraz równoległe z kondensatorem referencyjnym.
3. Część ładunku przepływa z kondensatora referencyjnego do wirtualnego kondensatora C_1 (rysunek 3).
4. Następnie nóżka połączona z elektrodą jest odłączona i dokonywany jest pomiar napięcia proporcjonalnego do pozostałego ładunku w kondensatorze referencyjnym.



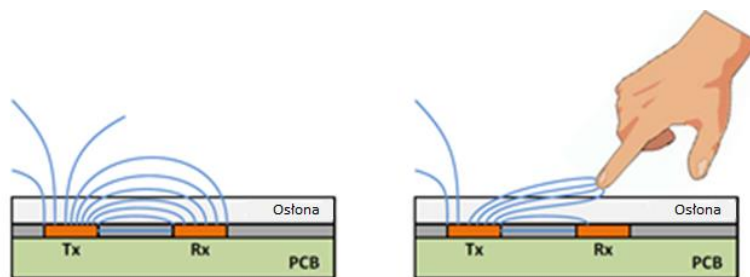
Rysunek 3. Schemat układu pomiarowego dla przycisku dotykowego działającego w trybie pomiaru pojemności własnej

Cykl jest ciągle powtarzany. Po kroku nr 2 między elektrodą a powierzchnią o potencjale neutralnym tworzy się pole elektryczne. Zanim przejdziemy do kroku nr 3 musimy odczekać pewien czas (im większa elektroda, tym większa pojemność, a w efekcie czas potrzebny do jej naładowania). Następnie wykonujemy pomiar napięcia, które jest proporcjonalne do ładunku zgromadzonego na elektrodzie. W przypadku gdy wykrywany obiekt np. palec, pojawia się w polu elektrycznym, następuje zwiększenie pojemności układu pomiarowego. Podczas pomiaru, wartość otrzymywana na przetworniku analogowo cyfrowym jest odwrotnie proporcjonalna do pojemności układu. Porównując zmierzone wartości między

cyklami, jesteśmy w stanie wykryć obecność lub zmianę pozycji obiektu znajdującego się w otoczeniu elektrody. Na rysunku 4 przedstawiono rozkład linii pola elektrycznego dla układu pomiarowego mierzącego pojemność własną.



Rysunek 4. Pole elektryczne dla układu pomiarowego w koncepcji pojemności własnej. Elektroda znajduje się na płycie drukowanej (PCB - Printed Circuit Board), z wierzchu jest zakryta osłoną



Rysunek 5. Pole elektryczne dla układu pomiarowego w koncepcji pojemności wzajemnej. Elektrody nadawcza (Tx) oraz odbiorcza (Rx) znajdują się na płycie drukowanej (PCB - Printed Circuit Board), z wierzchu są zakryte osłoną

Koncepcja pomiaru pojemności wzajemnej wymaga użycia dwóch elektrod – nadawczej i odbiorczej (rysunek 5). W porównaniu do pojemności własnej zamiast na pomiarach pojemności między elektrodą a potencjałem zerowym skupia się na pojemności między elektrodami. Cykl pomiaru jest podobny jak w poprzednim przypadku, z tą różnicą, że elektroda nadawcza na stałe jest podpięta do wyjścia mikrokontrolera, a elektroda odbiorcza jest połączona z wejściem mikrokontrolera. W efekcie cykl pomiaru kształtuje się następująco:

1. Na pinie mikrokontrolera połączonym z elektrodą nadawczą jest ustawiany w stan wysoki.

2. Dokonywany jest pomiar wartości napięcia na określonym pinie połączonym z elektrodą odbiorczą.

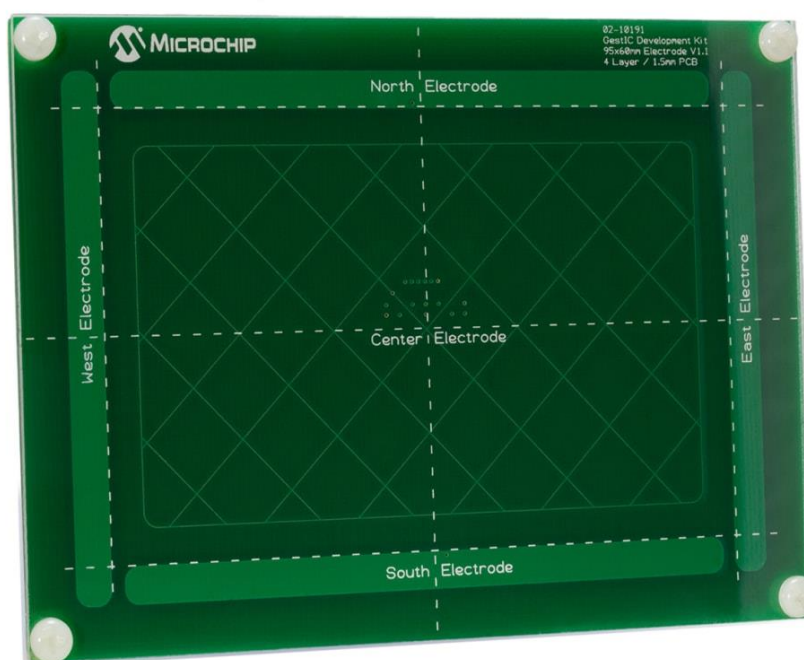
Krok 2 jest wykonywany również w pewnym odstępie czasu zależnym od rozmiaru elektrody.

Rozwiązanie z pojemnością wzajemną jest bardziej skomplikowane od pojemności własnej, jednak lepiej sprawdza się przede wszystkim dla paneli i ekranów dotykowych, gdyż pozwala na łatwą realizację skanowania w układzie wielu elektrod w formie siatki x-y. W takim przypadku np. kolumny są stosowane jako elektrody nadawcze, a wiersze jako odbiorcze. Taki sposób skanowania pozwala na detekcję dotyku w wielu punktach jednocześnie (ang. multi-touch) oraz działa znacznie lepiej od metody pojemności własnej w przypadku urządzeń bateryjnych, przez co jest w praktyce jedynym rozwiązaniem stosowanym w telefonach komórkowych.

Opisałem podstawową zasadę działania systemów pojemnościowych. Bardzo często zamiast ciągłego stanu wysokiego jako sygnał ładujący stosowany jest PWM o różnej częstotliwości. Istnieje również wiele koncepcji za równo po stronie sprzętowej jak i programistycznej. W kontekście tematyki realizowanej pracy, systemy pojemnościowe muszą umożliwić wykrywanie obiektów znajdujących się w pewnej odległości od elektrod. W praktyce sposób działania jest bardzo zbliżony do systemów dotykowych. Kluczową różnicą jest powierzchnia elektrod, a w efekcie znacznie większa pojemność którą operujemy.

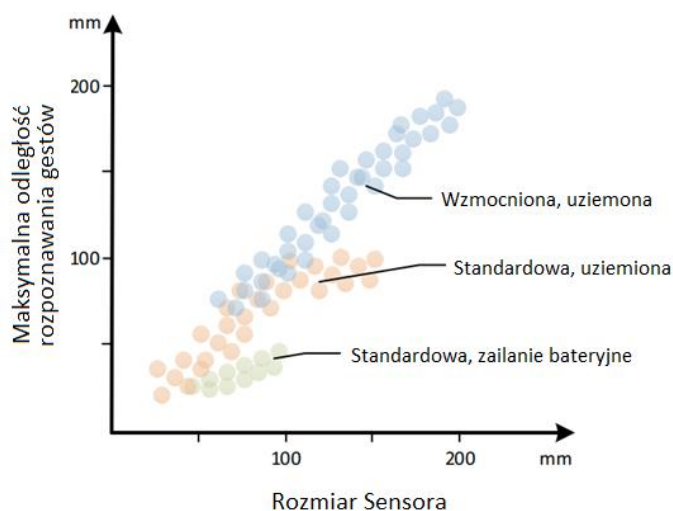
W literaturze można znaleźć przykłady wykorzystania czujników pojemnościowych zarówno w branży samochodowej [12] jak i poza nią [13]. Wspomniane aplikacje reagują na ruchy stopy oraz całej dłoni. Interesującym zagadnieniem jest wykrywanie gestów o większej rozdzielczości, które byłyby wykonywane przy użyciu np. palców. Publikacja [14] przedstawia próbę rozwiązania opartego o elektrody pojemnościowe spełniającego to wymaganie. Poprzez zastosowanie dedykowanej matrycy elektrod oraz wykorzystanie zmiennego napięcia do ładowania elektrod. W efekcie otrzymano system wykrywający 20 gestów z dokładnością 90%. Wspomniane rozwiązania są pracami typowo badawczymi, które dowodzą, że zastosowanie analizowanej technologii w ogóle jest możliwe. Na rynku istnieją rozwiązania, które zostały wdrożone i można spotkać je w samochodach. Przykładem jest moduł otwierania i zamykania dachu w samochodzie Jaguar XF Sportbrake z 2018 roku [15]. Wykorzystuje on 2 gesty w postaci przesunięcia ręki pod obszarem, w którym znajduje się elektroda. System wykorzystuje rozwiązanie firmy Microchip o nazwie GestIC. Opiera się ono o specjalizowany układ scalony [16], który wykonuje operacje skanowania na odpowiednio skonfigurowanych elektrodach (przykładowa elektroda została przedstawiona na rysunku 6).

Dane są następnie dostępne dla głównego procesora przez protokół I2C. Układ umożliwia wykorzystanie wbudowanego algorytmu opartego o Ukryte Modele Markowa, a także odczyt surowych danych bezpośrednio z przetwornika ADC. Pod kątem sygnałów podawanych na elektrody rozwiązanie to jest podobne do opisanego we wspomnianej publikacji [14]. Jeśli chodzi o sposób pomiaru pojemności, układ umożliwia pomiar w trybie pojemności wzajemnej oraz mieszanej. W pierwszym przypadku elektroda centralna jest elektrodą nadawczą oraz skrajne elektrody pracują jako elektrody odbiorcze. W trybie mieszanym elektroda centralna pracuje naprzemiennie jako nadawcza i odbiorcza.



Rysunek 6. Przykładowy rozkład elektrod w systemie GestIC

Jednym z głównych ograniczeń pojemnościowych systemów detekcji gestów jest ich zasięg działania. Czułość układu jest ściśle związana z rozmiarem elektrod. Na rysunku 7 przedstawiono efektywną odległość wykrywania gestów w zależności od rozmiarów elektrod. Wielkość „rozmiar sensora” – określa długość boku obszaru prostokąta zawierającego wszystkie elektrody.



Rysunek 7. Wykres odległości vs rozmiar elektrody [17]

Według danych przedstawionych na rysunku 7, można przyjąć, że dla wymiarów obszaru elektrod rzędu 10 cm otrzymamy zasięg w okolicach 6-8 cm. Jest to zgodne z obserwacjami podczas zastosowania tego rozwiązania, który został zbudowany i zostanie opisany w dalszej części niniejszej rozprawy (rozdział – „System do wykrywania gestów oparty o czujniki pojemnościowe”). Na wykresie widoczne są trzy tryby pracy: bateryjny, standardowy (z podłączeniem do masy instalacji elektrycznej) i wzmocniony (boosted). Tryb wzmocniony polega na zwiększeniu amplitudy napięcia z wartości referencyjnej dla układu (3.3 V) do wartości maksymalnie 18 V. Wykres ma określić w przybliżony sposób czego można się spodziewać po zastosowaniu elektrod określonej wielkości. Na zasięg działania ma wpływ również kształt, proporcje elektrod, a także ilość warstw na płytce PCB, materiał maskujący oraz inne czynniki związane min. z zasilaniem układu. Przykładowo systemy zasilane bateryjnie charakteryzują się znacznie mniejszą czułością, a w efekcie mniejszym zakresem wykrywania obiektów. Wynika to wprost z odizolowania potencjału zerowego układu zasilającego oraz uziemienia (masy instalacji elektrycznej), a tym samym znacznie niższej pojemności między elektrodą a masą. Układ GestIC został stworzony w taki sposób, aby jego zastosowanie było możliwe w bardzo szerokim spektrum aplikacji. Firma Microchip przygotowała obszerną instrukcję jak zaprojektować elektrody oraz schemat elektroniczny dostosowany do docelowego środowiska pracy [17].

Wszystkie rozwiązania oparte o elektrody pojemnościowe bazują na wytworzeniu płytek PCB z odpowiednio uformowanymi obszarami miedzianymi lub folii na których elektrody wykonane są z przezroczystego materiału. Oba rozwiązania są tanie w masowej produkcji, w szczególności płytki PCB. Dodatkowo stosunkowo łatwo jest zbudować elektrody, które

mogą być osłonięte powierzchnią dekoracyjną w taki sposób, że są one niewidoczne dla końcowego użytkownika. Do zamaskowania elektrod, można wykorzystać w zasadzie dowolne materiały dielektryczne. Należy pamiętać tylko o ograniczonej ich grubości.

Podsumowując technologia pojemnościowa nadaje się do zastosowania w branży samochodowej, co pokazują rzeczywiste aplikacje. Największym ograniczeniem systemów jest zasięg ich działania w szczególności w odniesieniu do kamer czy matryc Time of Flight. Jednak w przypadku, gdy ograniczymy i ściśle zdefiniujemy przestrzeń roboczą rozwiązanie staje się bardziej adekwatne. Największe zalety tego typu systemów to niska cena komponentów elektronicznych, duża elastyczność w dopasowaniu do aplikacji oraz łatwość zintegrowania z obecnymi urządzeniami HMI.

2.3. Systemy oparte o czujniki podczerwieni

Sensory podczerwieni wykorzystywane do wykrywania gestów, wykorzystują przynajmniej jedną diodę nadawczą. Długość fali w tego typu systemach mieści się w przedziale od 730 – 950nm, czyli pracują one w bliskiej podczerwieni. Światło jest odbijane od wykrywanego obiektu a następnie pada na kilka elementów elektronicznych takich jak fotodiody lub fototranzystory pracujące w odpowiednim zakresie długości fal. Dzięki wspomnianym komponentom jesteśmy w stanie uzyskać zmiany napięcia, które są proporcjonalne do natężenia odbitych fal. W niniejszym rozdziale rozważamy podejście oparte o zmiany w natężeniu odbitego światła. Innym systemem wykorzystującym promieniowanie podczerwone jest technologia ToF, opisana we wcześniejszych akapitach. W tamtym przypadku mierzony jest czas pomiędzy wysłaniem a odebraniem wiązki światła.

Najprostsze systemy wykorzystujące technologię podczerwieni służą do wykrywania zbliżenia obiektów. W tego typu systemach zazwyczaj wykorzystywany jest jeden element nadawczy i jeden odbiorczy. W przypadku wykrywania gestów, konieczne jest zastosowanie większej ilości czujników. W literaturze opisano wiele rozwiązań, które wykorzystują tą technologię do wykrywania prostych gestów. Ilość odbiorczych elementów waha się od trzech [18] do kilku tysięcy [19], gdzie zastosowano światłoczułą matrycę składającą się z światłoczułych pikseli o wymiarach 80x85. W kontekście tematu realizowanego doktoratu zakładającego brak użycia kamer, pojawia się problem polegający na zdefiniowaniu od jakiej ilości lub rozmiaru sensorów możemy definiować kamery. Opisywane rozwiązanie przypomina kamerę pracującą w zakresie podczerwieni. Porównując do pierwszych kamer o rozdzielczości

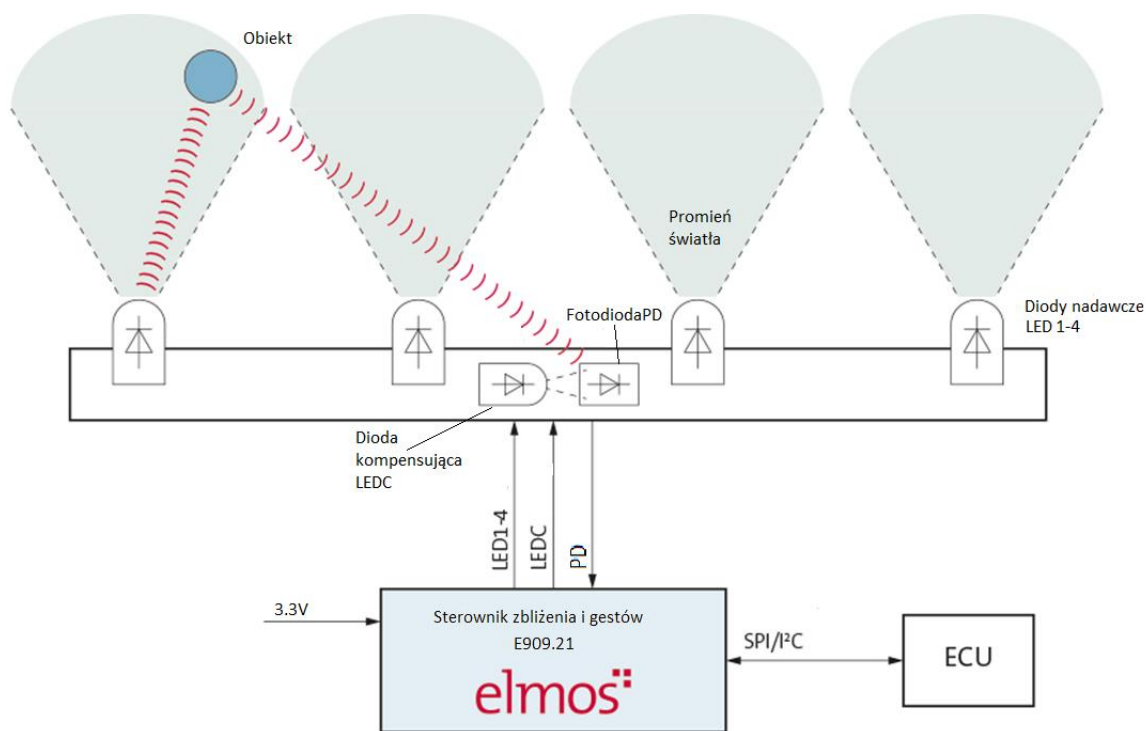
32x32 pikseli, powinniśmy zaklasyfikować rozwiązanie jako kamerę. Z drugiej strony współczesne kamery posiadają matryce o milionach pikseli. Powstaje pytanie o współczesną definicję czym jest kamera. Zagadnienie jest trudne do rozstrzygnięcia. Przedstawione rozwiązanie pozwalające na wykrywanie gestów, również bazowało na pomiarze natężenia światła odbitego, co stanowi argument za tym, żeby klasyfikować to rozwiązanie jako prostą kamerę.

Sama wartość natężenia światła bywa niewystarczająca dla potrzeb systemu rozpoznającego gesty, co pokazano w publikacji [18]. Autorzy wykorzystali inne cechy przebiegu sygnałów natężenia światła takie jak: czas i nachylenie zboczy narastających i opadających, średnie nachylenie zboczy, czas opóźnienia amplitudy. Podejście to pokazuje, że dzięki analizie sygnałów w domenie czasu jesteśmy w stanie uzyskać wiele dodatkowych informacji, które mogą być dodatkowymi cechami wejściowymi do algorytmu klasyfikującego. W opisywanym rozwiązaniu zastosowano klasyfikatory rozmyte.

W przemyśle samochodowym istnieją rozwiązania, które wykrywają proste gesty przy użyciu czujników podczerwieni. W "Cadillac User Experience System" [20] sensory odpowiadały tylko za wykrycie zbliżenia ręki użytkownika. Pierwszym systemem pozwalającym na wykrywanie gestów w trzech wymiarach, który wykorzystywał sensory podczerwieni, jest rozwiązanie zaprezentowane przez Volkswagena [21]. System rozpoznawał proste gesty (ruch w lewo oraz ruch w prawo), wykonywane całą dłonią przed ekranem centralnym na desce rozdzielczej.

Jednym z producentów układów kontrolujących pracę układu diod i fotodiod jest firma Elmos [22]. Od niedawna firma oferuje rozwiązanie oparte o technologię Time of Flight, które jest prezentowane jako dopełnienie systemu istniejącego systemu (rysunek 8). Podczas analizy funkcjonalności systemów wykorzystujących podczerwień, łatwo zauważyć istotne ograniczenia.

W przypadku rozwiązań opartych o klika elementów fotoelektrycznych wyraźnym limitem jest niska rozdzielczość. Co prawda w publikacji [18], pokazano, że jest możliwe wykrywanie większej ilości gestów, jednak nadal są to gesty wykonywane całą dłonią.



Rysunek 8. Schemat działania rozwiązania do rozpoznawania gestów firmy Elmos [23]

Dodatkowym problemem dla tego typu systemów jest wpływ światła słonecznego. Jeśli układ sensoryczny jest narażony bezpośrednio na promienie słoneczne, komponenty fotoelektryczne nasycają się, co w efekcie czyni bardzo trudne rozróżnienie czy sygnał pochodzi od promieni słonecznych czy od wiązki nadanej przez diodę emitującą (nadawczą) systemu. W systemie Halios [23], problem ten został rozwiązany przez odpowiednią modulację częstotliwościową sygnału sterującego diodami. System posiada dodatkowy kanał, który odpowiada za pomiar natężenia promieniowania otoczenia. Dzięki temu możliwe jest „odjęcie” natężenia promieni świetlnych nie pochodzących z diod nadawczych systemu.

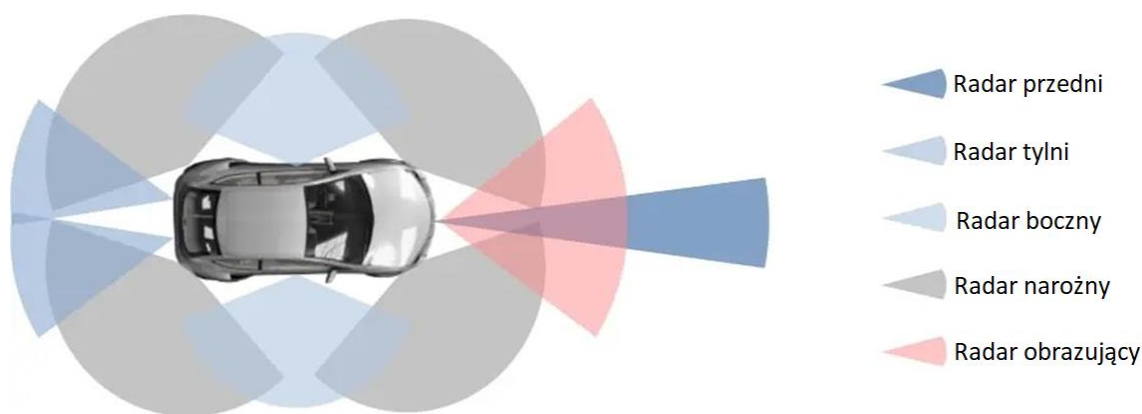
W kontekście adaptacji w samochodzie, istotnym zagadnieniem jest również kwestia doboru materiału, za którym będzie możliwe ukrycie czujników. Powszechnym wymaganiem jest to, żeby system był zupełnie niewidoczny dla końcowego użytkownika. Jednocześnie przenikalność promieniowania fal podczerwonych musi być na odpowiednim poziomie. Przykładowym materiałem, który pozwala spełnić te wymagania jest Optolite™ IR firmy Instrument Plastics [24]. Został on wykorzystany podczas wstępnych testów rozwiązań wykorzystujących promieniowanie w podczerwieni. Jest on stosowany jako filtr przepuszczający promieniowanie o długości fali powyżej 750nm. W tym przypadku, istnieje

ograniczenie natury stylistycznej, gdyż praktycznie wszystkie tego rodzaju materiały mają w świetle widzialnym kolor czarny.

Pomimo wspomnianych wyzwań w implementacji, rozwiązanie może być wykorzystane w branży samochodowej, co pokazały wspomniane przykłady. Podobnie jak w przypadku sensorów pojemnościowych, koszt systemu sensorycznego oraz stopień skomplikowania układu elektronicznego są niewielkie.

2.4. Systemy radarowe

Historia wykorzystania zjawiska odbicia fal elektromagnetycznych od obiektów celem ich wykrywania sięga początków XX wieku. Pierwotne zastosowanie polegało na wykrywaniu statków i samolotów w przestrzeni. Rozwijany w tajemnicy przez Roberta Watsona-Watta, ukończony tuż przed rozpoczęciem wojny system radarów, w dużym stopniu przyczynił się do wygrania przez Aliantów bitwy o Anglię [25]. Wraz z rozwojem techniki, radary stawały się coraz mniejsze co przekładało się na coraz krótsze długości stosowanych fal radiowych oraz wyższe częstotliwości. Obecnie, samochody są wyposażane w kilka do kilkunastu radarów. Są one wykorzystywane głównie przez systemy aktywnego bezpieczeństwa oraz autonomizacji jazdy. Wraz ze zwiększaniem się poziomu autonomiczności pojazdów, liczba radarów najprawdopodobniej będzie wzrastać. Z perspektywy tematyki niniejszej rozprawy, jest to korzystny trend. Im większe będzie zastosowanie radarów w branży samochodowej, tym większa ilość dostępnych rozwiązań. Ponadto przy większej skali naturalnym efektem jest wzrost produkcji oraz spadek ceny komponentów co jest kolejnym pozytywnym czynnikiem zwiększającym prawdopodobieństwo wzrostu liczby aplikacji wykorzystujących radary również do rozpoznawania gestów. Obszary aplikacji radarów w samochodach przedstawiono na rysunku 9.

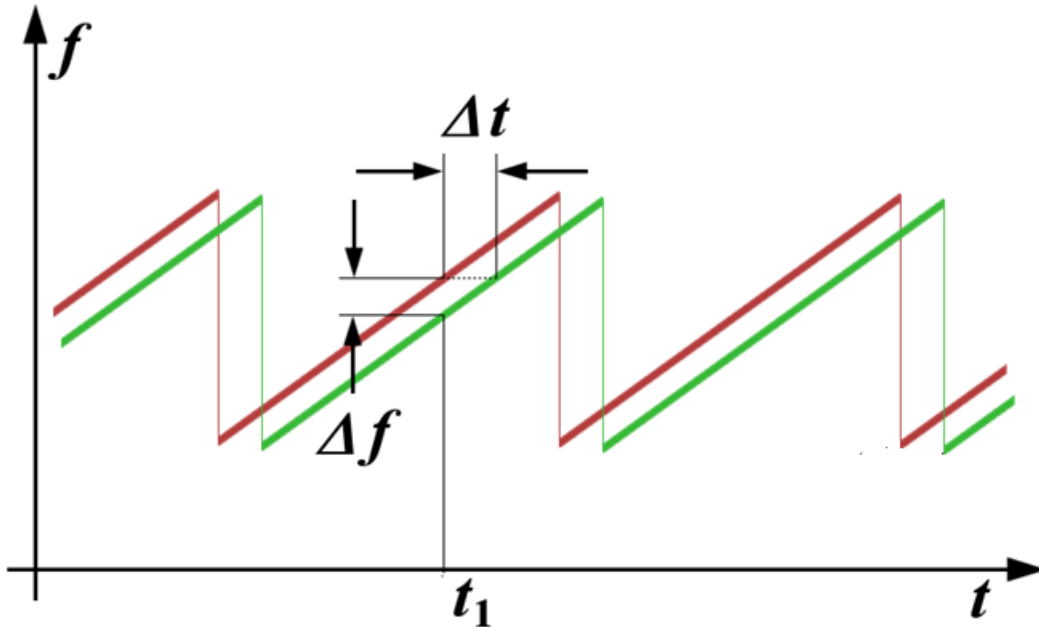


Rysunek 9. Zastosowanie radarów w samochodzie

Podczas przeglądu rozwiązań, znaleziono dwa interesujące zastosowania radarów do rozpoznawania gestów. Pierwszym jest projekt Google Soli [26]. Publikacja “Interacting with soli” [27] pokazuje duże możliwości systemów do rozpoznawania gestów opartych o radary. Rozwiązanie wyróżnia się możliwością rozpoznawania bardzo subtelnych gestów, wykonywanych tylko przy użyciu palców. W prezentowanym w artykule rozwiązaniu, zastosowano kombinację sieci splotowych oraz rekurencyjnych. Projekt Soli wykorzystuje rozwiązania sprzętowe firmy Infineon. Kolejnym przykładem jest publikacja opisująca rozwiązanie oparte o radar wykorzystujący dwie anteny nadawcze oraz cztery odbiorcze [28]. Algorytm pozwalający na rozpoznawanie gestów, który zastosowano to “Las decyzyjny” (Random Forest). Liczba gestów które były możliwe do wykrycia była znacznie mniejsza w porównaniu z pierwszą publikacją. Gesty polegały głównie na dokonywaniu przesunięcia palcami w obszarze radaru. Dzięki wysokiej rozdzielczości możliwe było wykonywanie gestów tylko przy wykorzystaniu palców. Dodatkowym atutem opisanego rozwiązania jest fakt, iż zademonstrowano jego działanie w jeżdżącym samochodzie.

Ogólna zasada działania radaru polega na emitowaniu fal elektromagnetycznych, a następnie ich detekcji po wcześniejszym odbiciu od obiektu. Jedną z mierzonych wielkości jest czas w jakim wyemitowane fale pokonują drogę między radarem a obiektem oraz w drodze powrotnej po odbiciu. Koncepcja wysyłania i detekcji promieniowania jest podobna jak w przypadku czujników podczerwieni oraz systemu Time of Flight. Radary stosowane w branży samochodowej wykrywają obiekty za pomocą Fali Ciągłej (Stałej) z modulacją częstotliwości (ang. Frequency-Modulated Continuous-Wave – FMCW). Metoda ta polega na wysyłaniu fal o zmiennej częstotliwości w pewnym zakresie (np. sygnał świergotowy, ang. chirp od 58 do 63

GHz), a następnie na detekcji częstotliwości odbitych fal. Na poniższym wykresie (rysunek 10) częstotliwości od czasu, kolorem czerwonym zaznaczono przebieg sygnału nadawczego. Zielony odpowiada przebiegowi sygnału odebranego. System rozpoznający gesty wykorzystujący radar został opisany w rozdziale „System do wykrywania gestów oparty o radar”.



Rysunek 10. Wykres przedstawiający przesunięcie częstotliwościowe $f(t)$ między sygnałem wysłanym oraz odebranym [29]

Odległość obiektu od radaru wyznacza się z zależności (2).

$$R = \frac{c |\Delta f|}{2 \left(\frac{d(f)}{d(t)} \right)} \quad (2)$$

c_0 - prędkość światła

Δf - zmierzona różnica częstotliwości

$\frac{d(f)}{d(t)}$ - tempo zmian częstotliwości w czasie sygnału nadawczego

Wartość $\frac{d(f)}{d(t)}$ określająca tempo zmian częstotliwości wysyłanego sygnału oraz różnica częstotliwości między sygnałem nadanym i odebrany pozwala na wyznaczenie odległości od obiektu [29] [30].

Prędkość może zostać wyznaczona za pomocą zmian odległości obserwowanego obiektu zgodnie z równaniem (3).

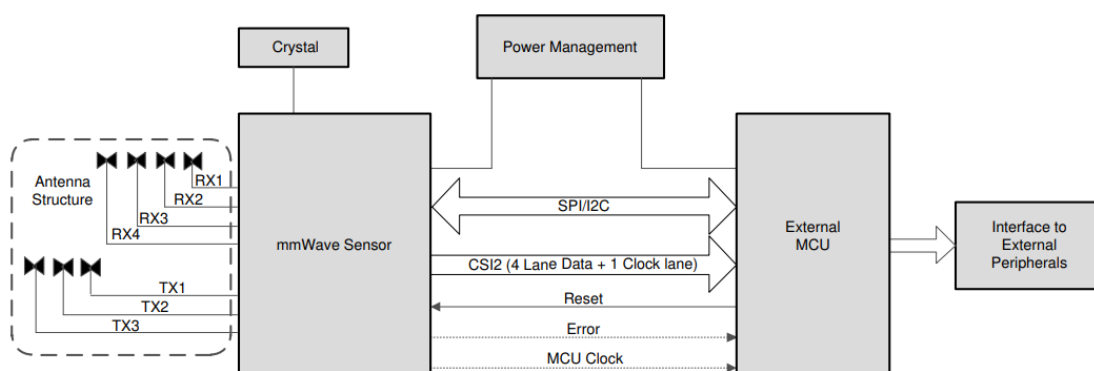
$$v = \frac{\Delta R}{t_c} \quad (3)$$

ΔR – zmiana zakresu

t_c – czas trwania sygnału chirp

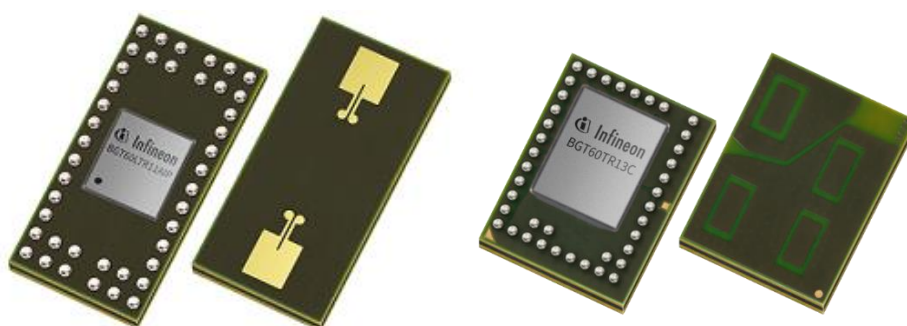
W praktyce ze względu na bardzo niewielkie zmiany odległości podczas trwania sygnału chirp, do wyznaczenia prędkości wykorzystuje się przesunięcia w fazie otrzymanego sygnału. Dokładny sposób został opisany w rozdziale „System do rozpoznawania gestów oparty o radar”, opisującym wykorzystanie radaru do wykrywania gestów.

Ze względu na konieczność bardzo szybkiej realizacji wspomnianych operacji, do ich przeprowadzenia wykorzystuje się rozwiązania sprzętowe oparte o specjalizowane układy scalone. Jest to zrozumiałe, gdy weźmiemy pod uwagę fakt, że czas trwania sygnału chirp w tego typu rozwiązaniach wynosi kilkadziesiąt mikrosekund [31]. Dodatkowo stosowane są systemy które zawierają kilka anten nadawczych i kilka odbiorczych. Większość dostępnych rozwiązań wymaga zaprojektowania anten nadawczych oraz odbiorczych. W efekcie architektura systemu wygląda jak na rysunku 11.



Rysunek 11. Przykładowa aplikacja dla układu AWR1243 (oznaczona jako mmWave Sensor), za dokumentacją [32]

W ostatnim czasie na rynku pojawiły się dwa rozwiązania “Antenna in Package” (Antena na Obudowie), które znacznie ułatwiają zastosowanie tej technologii dzięki eliminacji konieczności realizacji skomplikowanego zagadnienia jakim jest zaprojektowanie dedykowanego układu anten. Układy umożliwiające wrywanie gestów, które posiadają anteny na swojej obudowie, są oferowane przez firmy Infineon oraz Texas Instruments. Przykładowo układy BGT60LTR11AIP oraz BGT60TR13C pierwszej z wymienionych firm, posiadają na swoich obudowach odpowiednio: jedną parę anten, nadawczą i odbiorczą; oraz 3 anteny odbiorcze i jedną nadawczą [29]. Zdjęcia układów są widoczne na rysunku 12. Drugi z producentów oferuje podobne rozwiązania spełniające wymagania temperaturowe dla branży samochodowej. Układ AWR1843 integruje w swojej obudowie 3 anteny nadawcze oraz 4 odbiorcze [33].



Rysunek 12. Przykłady układów Antenna on Package, Infineon - 60LTR11AIP oraz BGT60TR13C

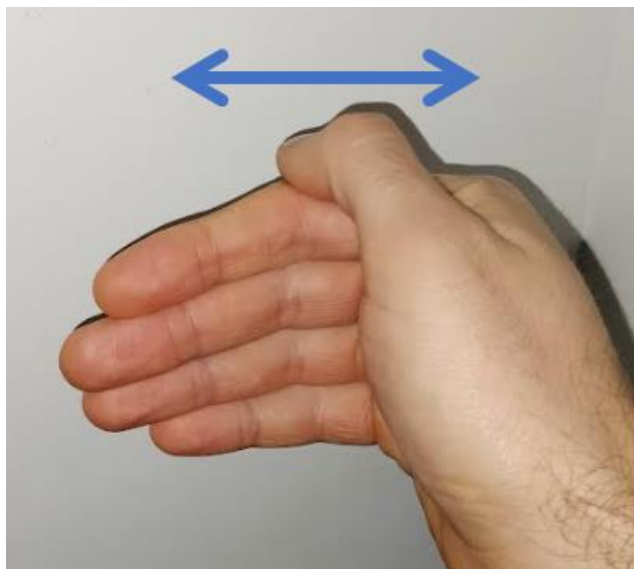
Powszechne zastosowanie radarów w branży samochodowej zwiększa możliwości wyboru rozwiązania do rozpoznawania gestów. Radar może być łatwo schowany za powierzchnią kokpitu, przez co integracja w środowisku samochodu jest stosunkowo prosta. Z drugiej strony, biorąc pod uwagę cenę układów, koszt takiego rozwiązania w porównaniu do systemów pojemnościowych lub podczerwieni jest kilku lub kilkunastokrotnie wyższy. Można jednak oczekiwać spadku cen rozwiązań wraz z ich popularyzacją. Dodatkowo zagadnienia związane z przetwarzaniem sygnałów, a także z projektowaniem odpowiednich anten zwiększają koszty rozwoju podobnych rozwiązań. Rozwiązaniem tego ostatniego problemu mogą być rozwiązania typu “Antenna in Package”. W znaczący sposób upraszczają i skracają czas rozwoju po stronie sprzętowej.

2.5. Systemy ultradźwiękowe

Systemy rozpoznające gesty opisane w literaturze w większości bazują na wykorzystaniu efektu Dopplera. Fala dźwiękowa emitowana jest przez jeden lub kilka nadajników, następnie trafia w poruszający się obiekt, odbija się od niego i ostatecznie trafia do jednego lub wielu odbiorników. Zgodnie z efektem Dopplera, gdy element zbliża się do odbiornika następuje wzrost częstotliwości fali dźwiękowej, analogicznie podczas oddalania się obiektu częstotliwość spada.

Wszystkie systemy pracują w zakresie ultradźwięków, czyli powyżej 20 kHz. Warto zauważyć, że pomimo iż częstotliwości w zakresie 20 do 65 kHz są niesłyszalne dla ludzi, to dla wielu gatunków zwierząt dźwięki będą słyszalne. W szczególności należy zwrócić uwagę na zwierzęta "domowe", które często przebywają z ludźmi. Górny próg słyszalności dla psa i kota to odpowiednio ~45kHz oraz ~65kHz [34]. Biorąc pod uwagę te uwarunkowania zakres częstotliwości powinien przekraczać 70kHz. Co prawda częstotliwości te nadal będą słyszalne przez pewne zwierzęta takie jak np. szczury, myszy, nietoperze, jednak ich przebywanie w okolicy ludzi jest znacznie rzadsze. W kontekście aplikacji systemu w samochodzie należałoby przyjąć zakres powyżej 100 kHz ze względu na powszechność stosowania samochodów w różnorodnym otoczeniu, którego część mogą stanowić rozmaite zwierzęta [34].

Pierwszym przykładem wykorzystania fal ultradźwiękowych do rozpoznawania gestów jest publikacja [35], gdzie przedstawiono system oparty o urządzenia audio zwyczajnego smartfonu. Emitowane przez głośnik fale dźwiękowe o częstotliwości 21kHz po odbiciu od ręki użytkownika były rejestrowane przez mikrofon służący zazwyczaj do prowadzenia rozmów. Autorzy przetestowali kilka algorytmów, z czego najlepsze rezultaty uzyskali wykorzystując sieci neuronowe oraz regresję liniową. Innym rozwiązaniem, które dowodzi działania systemów opartych o fale ultradźwiękowe, jest system dedykowany do rozpoznawania mikro gestów, wykonywanych tylko z wykorzystaniem palców [36]. Rozwiązanie nadaje fale o częstotliwości 300 kHz i wykorzystuje dedykowany nadajnik oraz odbiornik. Kolejnym przykładem jest publikacja [37], przedstawiająca wysoką skuteczność rozpoznawania gestów - 93,9%. Zastosowano układ jednego nadajnika oraz trzech odbiorników. Częstotliwość nadawanych fal to 40kHz. W artykule rozwiązanie zostało porównane z systemem soli, ze względu na fakt, że było możliwe rozpoznawanie niektórych subtelnych gestów wykonywanych tylko przy użyciu palców jak pokazany na rysunku 13. Uzyskana skuteczność rozpoznawania była bardzo zbliżona do rozwiązań wykorzystujących radar Soli.



Rysunek 13. Gest polegający na przesunięciu kciuka po powierzchni palca wskazującego, wykrywalny przez system ultradźwiękowy

Kolejny z systemów, częściowo klasyfikuje się do kolejnej kategorii: "Systemów zlokalizowanych na użytkownika" [38]. Na rękę użytkownika umieszczony jest nadajnik ultradźwiękowy. System bazuje na metodzie "Angle-of-Arrival", co oznacza, że system wyznacza kierunek, z którego nadawany jest sygnał. System zrealizowano przy użyciu matrycy sensorów oraz algorytmu analizującego kąty między kolejnymi zarejestrowanymi położeniami ręki. W efekcie uzyskano system rozpoznający określane gestami cyfry o skuteczności 95%. W praktyce wdrożenie rozwiązania może być kłopotliwe ze względu na konieczność umieszczenia nadajnika na dłoni. Należy oczekiwać ograniczenia stosowalności tego rozwiązania do kontrolowanych środowisk, np. dla operatorów maszyn.

Porównując przedstawione rozwiązania zauważono, że częstotliwość pracy systemu wykracza poza zakres słyszalny dla ssaków tylko w przypadku opisanym w publikacji [36]. Oddziaływanie na organizmy żywe fal o częstotliwościach poniżej 100 kHz musiałyby być zbadane i przeanalizowane. Stosowanie częstotliwości z zakresu 20-100 kHz, może istotnie wpływać na komfort i zachowanie ludzi i zwierząt. Obecnie, sensory ultradźwiękowe są najczęściej używane w systemach wspomagających parkowanie. W tym wypadku zagadnienie oddziaływania na otaczającą faunę nie jest krytyczne ze względu na krótki czas narażenia oraz rzadkość przebywania zwierząt w okolicy pojazdu podczas manewrowania parkingowego. Praca [39] opisuje wpływ systemów odstrasżających gryzonia, jednak częstotliwości wykorzystywane do tego celu mieszczą się w zakresie pasma słyszalnego przez ludzi. We wnioskach nie stwierdzono istotnego wpływu emisji ultradźwięków na zachowanie psów.

Z drugiej strony u ponad 93% badanych psów zauważono zmiany w zachowaniu. W przypadku systemu do rozpoznawania gestów, narażenie zwierzęcia może być znacznie bardziej długotrwałe i intensywne. Jest to obszar konieczny do zbadania w kontekście wdrożenia tej technologii. Powszechnie stosowane są też systemy służące do wykrycia obecności osób we wnętrzu pojazdu, zwykle współpracujące z alarmami. Takie systemy są jednak nieaktywne podczas korzystania z samochodu przez uprawnionych użytkowników, przez co nie mają wpływu na ewentualnie przewożone zwierzęta domowe.

Kolejnym aspektem jest zabudowanie sensorów w sposób akceptowalny przez producentów

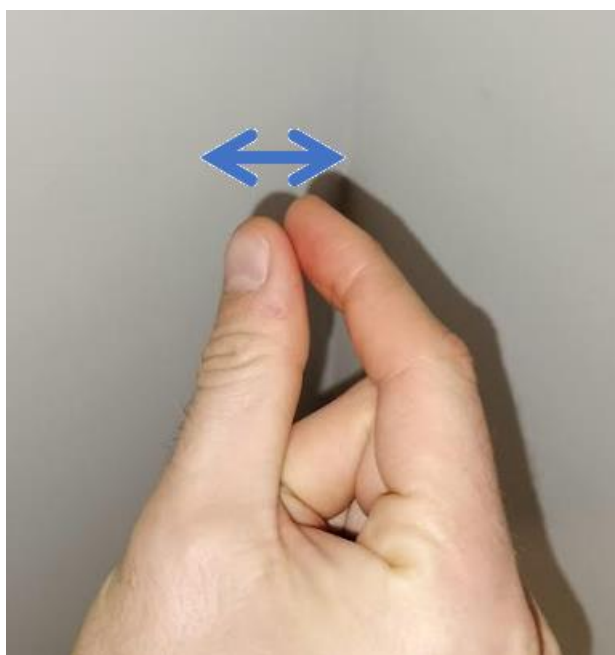
i konsumentów samochodów. Dla prawidłowego działania systemu fala akustyczna musi być poprawnie propagowana w obszarze wykrywania gestów, natomiast każda osłona czy bariera to utrudnia lub uniemożliwia. Umieszczenie przetworników za przepuszczalną osłoną, podnosi ryzyko ich zalania lub uszkodzenia z powodu czynników zewnętrznych. Rozwiązaniem może być zastosowanie wspomnianych przetworników, takich jak w zderzakach samochodów, używanych do wykrywania przeszkód wokół samochodu. Są one bardzo odporne na czynniki środowiskowe panujące na zewnątrz samochodu, będącymi znacznie bardziej wymagającymi niż wewnątrz pojazdu. Zastosowanie to pokazuje jednak, że są one zawsze widoczne, co może być nieakceptowalne pod kątem wizualnym we wnętrzu samochodu. Innym rozwiązaniem problemu może być montaż w miejscu ograniczającym ryzyko wpływu czynników zewnętrznych, tak jak we wspomnianych systemach alarmowych, gdzie sensory zlokalizowane są w obszarze podsufitki pojazdu.

2.6. Systemy zlokalizowane na ręce

Wszystkie typy systemów opisane w poprzednich akapitach mają ograniczony obszar wykrywania gestów spowodowany koniecznością obecności dłoni w odległości dostatecznie bliskiej od sensorów. Rozwiązaniem tego ograniczenia jest koncepcja umiejscowienia sensorów na ręce użytkownika. W praktyce istniejące rozwiązania wykorzystują specjalne rękawice lub opaski. Ze względu na problem zastosowania rękawic dla dłoni o różnym kształcie i rozmiarze oraz założeniu wykorzystania systemu w masowej skali, jako bardziej prawdopodobne do wdrożenia skupiłem się głównie na analizie rozwiązań wykorzystujących opaski.

Znalezione w literaturze systemy, wykorzystują dane wejściowe z dwóch różnych źródeł. Pierwsza grupa używa danych dotyczących dynamiki ruchu dłoni pochodzących

z akcelerometru. Druga część korzysta z sygnałów skorelowanych z aktywnością mięśni oraz ścięgien odpowiedzialnych za ruchy dłoni i palców. Publikacja [40] przedstawia koncepcję bransoletki z wbudowanymi akcelerometrami. Użytkownik porusza ręką w dowolnej przestrzeni w samochodzie lub poza nim. Warunkiem granicznym jest zasięg komunikacji opaski z głównym systemem. Sposób w jaki zrealizowano komunikację nie został opisany. Dostępne rozwiązania oferują wiele możliwości takich jak np. Bluetooth czy inną komunikację radiową. Dane zebrane podczas ruchu trafiają do algorytmu HMM (Hidden Markov Model – Ukryte Modele Markova). We wspomnianej pracy rozpoznawano gesty wykonywane przy pomocy całej dłoni. Kolejnym przykładem zastosowania akcelerometrów jest praca [41], w której zastosowano “smartwatch”. Wykorzystano zainstalowane w zegarku akcelerometr i żyroskop. Komunikacja odbywała się w taki sam sposób jak w przypadku klasycznego użycia smartfonu i telefonu – protokół Bluetooth. Liczba gestów jest niewielka, ale są to gesty najrzadziej rozpoznawane, czyli wykonywane samymi palcami. Przykładem takiego gestu jest przedstawione na rysunku 14 szczyknięcie (“pinch”), który jest wykonywany przy użyciu palca wskazującego i kciuka.



Rysunek 14. Gest szczyknięcia (pinch gesture)

Kolejny system [42] wykorzystał dwa źródła danych, oprócz akcelerometru użyto odbiornika fal podczerwieni. Akcelerometr został wykorzystany podobnie jak w poprzednich przykładach, natomiast zastosowanie odbiornika podczerwieni jest unikalne, ponieważ odbiornik sygnału umiejscowiony był na ręce użytkownika. System składał się z obszernej

bransolety (duży rozmiar wynikał głównie z prototypowego charakteru rozwiązania), na której zamontowano odbiornik fal podczerwieni. W obszarze roboczym rozmieszczano nadajniki podczerwieni, które wysyłały unikalny kod dla każdej z pozycji, w której się znajdowały. System zamontowany w bransoletce odbierał sygnały a następnie przysyłał je za pomocą modułu radiowego do modułu centralnego z jednostką FPGA, na której zaimplementowano sieć neuronową. System rozpoznawał 5 rodzajów gestów wykonywanych przy użyciu całej dłoni. Niestety nie podano skuteczności rozpoznawania gestów.

Ostatnie dwa opisane rozwiązania wykorzystują wielkości odzwierciedlające stan ścięgien oraz mięśni operujących palcami. Pierwsza publikacja [43] przedstawia opaskę z sensorami - rezystorami FSR (ang. Force Sensitive Resistor – Rezystor wrażliwy na działanie siły), które ściśle przylegają do nadgarstka. System wyróżnia się na tle pozostałych dużą ilością rozpoznawanych gestów – 24. Rozpoznawane gesty należą zarówno do kategorii wykonywanych całą dłonią jak i pojedynczymi palcami. Dodatkowo uzyskano bardzo wysoką skuteczność. W przypadku algorytmu kNN (ang. k-Nearest Neighbors – k-najbliżsi sąsiedzi) było to ponad 97%. Niestety czas potrzebny do rozpoznania gestów przez algorytm wynosił prawie 4 sekundy, pomimo zastosowania “desktopowego” komputera. Drugim rodzajem systemów - zdecydowanie powszechniejszym - są rozwiązania oparte o Elektromiografię (EMG). Metoda ta polega na pomiarze potencjałów bioelektrycznych mięśni i nerwów. W pracy [44] wykorzystano dedykowane elektrody. Popularność tej metody pokazuje praca [45], która zawiera zestawienie 65 publikacji, gdzie wykorzystano EMG do rozpoznawania gestów. Kilka z przedstawionych rozwiązań charakteryzowało się bardzo wysoką skutecznością, powyżej 95%.

W kontekście aplikacji w branży samochodowej, głównym ograniczeniem rozwiązań kontaktowych (np. EMG, FSR) jest konieczność zapewnienia bardzo dobrego przylegania do powierzchni skóry. Jest to trudne do osiągnięcia, gdy weźmiemy pod uwagę fakt, że za poprawne dopasowanie musiałby odpowiadać końcowy użytkownik systemu. Główną zaletą sensorów umieszczonych w opasce lub smartwatchu jest większy obszar roboczy, gdzie wykrywane są gesty. Nie musi być on ograniczony tylko do wnętrza samochodu. W przypadku samochodów autonomicznych, można wykorzystać tę właściwość i użyć tego typu systemu do przywołania samochodu za pomocą wybranego gestu, bez konieczności dotarcia do miejsca jego postoju. Następnie po zajęciu pozycji za kierownicą, ten sam system zapewni możliwość sterowania multimediami lub innymi urządzeniami zlokalizowanymi w samochodzie. Dzisiejsze smartwatche mogą pracować w trybie autonomicznym i komunikować się bezpośrednio

z serwerem o dużej mocy obliczeniowej, który ułatwia zastosowanie algorytmów wykorzystujących uczenie maszynowe. Tego typu rozwiązania niosą ze sobą ryzyko ataków hackerskich. Z drugiej strony w przemyśle samochodowym coraz częściej wymagane są rozwiązania z zakresu cyberbezpieczeństwa co bezpośrednio wiąże się z wdrażaniem normy ISO/SAE 21434 "Road vehicles – Cybersecurity Engineering". Podsumowując, działanie systemów umieszczonych na ręku użytkownika posiada główną wadę w postaci konieczności noszenia dodatkowego urządzenia. Niedogodność ta może przynieść korzyści, które nie występują w innych rozwiązaniach, w postaci możliwości wykonywania gestów w dowolnej przestrzeni, co oczywiście niesie za sobą kolejne ryzyka. Jednoznaczna odpowiedź na pytanie: Czy systemy umieszczone na ręku użytkownika są odpowiednie dla branży samochodowej? Będzie mogła zostać udzielona po zaproponowaniu takiego rozwiązania użytkownikom. Z technicznego punktu widzenia wdrożenie takiego systemu jest możliwe w branży samochodowej. Być może bardziej adekwatną aplikacją byłyby rozwiązania np. dla operatorów maszyn, gdzie rozwiązania tego typu mogłyby być zintegrowane z dotychczasowo używaną odzieżą ochronną wymaganą przepisami BHP.

2.7. Fuzja danych

2.7.1. Zdefiniowanie pojęcia fuzji danych

Fuzja danych jest pojęciem obejmującym szeroki zakres zagadnień związanych z wykorzystywaniem wielu źródeł danych w celu uzyskania dodatkowych informacji. Na przestrzeni lat naukowcy oraz inżynierowie zajmujący się tą kwestią zaproponowali kilka poniższych definicji (tłumaczenia własne):

Fuzja danych – „Jest to proces zajmujący się grupowaniem, korelacją i łączeniem danych oraz informacji z jednego lub więcej źródeł danych w celu osiągnięcia przetworzonej pozycji, przybliżonej charakterystyki oraz kompletnych predykcji sytuacji i zagrożeń, a także ich istotności w czasie. Proces charakteryzuje się ciągłym przetwarzaniem predykcji oraz ewaluacją potrzeb dodatkowych zasobów lub modyfikacji dla samego procesu fuzji, w celu osiągnięcia poprawionych rezultatów” [46] – Definicja zaproponowana przez Joint Working Group Directors of Laboratories (JDL).

“Fuzja danych jest formalną strukturą (metodologią), która zawiera środki i narzędzia do łączenia danych pochodzących z różnych źródeł” [47] – L. Wald.

“Fuzja polega na łączeniu lub scalaniu informacji, które pochodzą z kilku źródeł oraz eksploatacji połączonych lub scalonych informacji dla różnych zadań, takich jak odpowiedzi na pytania, podejmowanie decyzji, szacowaniu numerycznemu itp.” [48] – I. Bloch.

“Celem fuzji danych jest maksymalizacja użytecznych informacji pozyskanych z heterogenicznych źródeł, aby wykryć istotne sytuacji i zdarzenia dotyczące obserwowanego środowiska [49]” – Mastrogiovanni.

Przypadek zwielokrotnienia danych pomiarowych polegający na użyciu identycznych redundantnych sensorów, również wpisuje się w temat fuzji danych. Dla osób niezajmujących się na co dzień przetwarzaniem danych, termin fuzji danych może wydawać się abstrakcyjny. Jeśli jednak potraktujemy ludzkie zmysły jako sensory, a nasz mózg jako „jednostkę obliczeniową” dokonującą fuzji danych otrzymanych z sensorów, zagadnienie staje się znacznie bardziej przystępne. Dobrym przykładem jest fakt widzenia stereoskopowego przez ludzi. Jest to możliwe dzięki wykorzystaniu dwóch sensorów w postaci pary oczu. Każde z oczu uzyskuje obraz uzyskany z innej perspektywy wynikającej z oddalania gałek ocznych. Patrząc tylko przy użyciu jednego oka nie jesteśmy w stanie dokładnie określić odległości do obserwowanych obiektów. Po dokonaniu fuzji danych pochodzących z nerwów wzrokowych, mózg jest w stanie złożyć oba obrazy, co pozwala na określenie odległości w jakiej znajdują się obserwowane obiekty. Zgodnie z poniżej zaprezentowaną klasyfikacją jest to przykład kooperacyjnej fuzji danych z sensorów jakimi są ludzkie oczy.

Dwa z opisywanych systemów w rozdziale „Przegląd literatury oraz istniejących rozwiązań” wykorzystywały fuzję danych do rozpoznawania gestów. Pierwszy z nich [9], zatytułowany “3D Hand Gesture Recognition Based on Sensor Fusion of Commodity Hardware” zastosował fuzję danych pochodzących z klasycznej kamery oraz sensora ToF – Kinect. Motywacją zastosowania fuzji tych dwóch sensorów było ograniczenie rozpoznawania gestów wykonanych w odległości dalszej niż jeden metr od sensora Kinect. Drugi z wcześniej opisywanych przykładów, [42] również wykorzystywał fuzję danych, mimo że w artykule ten termin nie pada. Chodzi o system wykorzystujący jednocześnie sensor podczerwieni oraz akcelerometr umieszczony na bransoletce, poruszający się w otoczeniu promieniowania podczerwonego emitowanego w otoczeniu.

W technice, Fuzja Danych jest ściśle związana z rozwojem układów sensorycznych. W publikacji [50] wyszczególniono aspekty rozwoju technologii, które umożliwiły dokonywanie fuzji danych oraz jej rozwój. Systemy sensoryczne charakteryzujące się wymienionymi poniżej cechami istnieją od kilkudziesięciu lat, przez co większość z przytoczonych czynników może

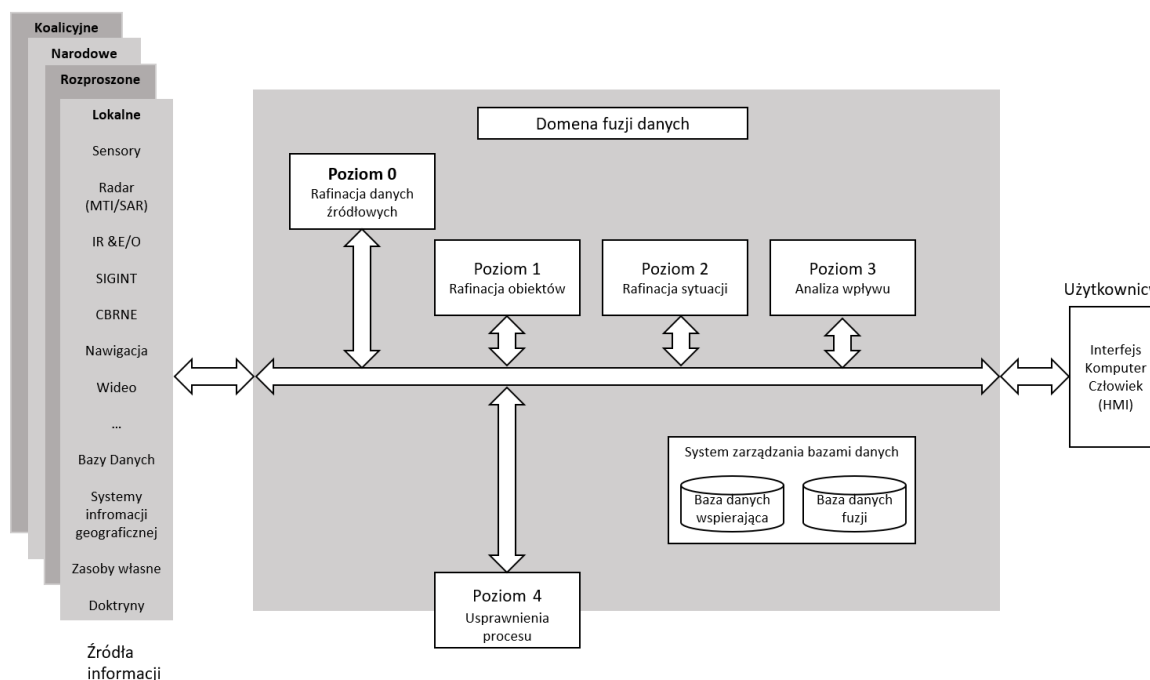
wydawać się z dzisiejszej perspektywy oczywista. Systematyczne ich wymienienie pozwala zrozumieć, dlaczego fuzji danych nie wykorzystywano w technice wcześniej. Poniżej wymieniono sześć czynników które sprawiły, że wykorzystanie fuzji danych stało się uzasadnione:

1. Zaawansowanie i niezawodność systemów sensorycznych oraz zwiększenie czułości w stopniu wykraczającym poza zakres ludzkich zmysłów (np. promieniowanie podczerwone).
2. Komunikacja między systemami sensorów oraz między sensorami i jednostkami obliczeniowymi, stała się wystarczająco szybka. Wyeliminowano opóźnienia, dzięki czemu możemy uzyskać wynik z analizy zanim otrzymamy nowe dane.
3. Technologie informatyczne zapewniające wystarczającą moc obliczeniową. Wzrosła dostępność algorytmów przyspieszających eksplorację danych. Dodatkowo dostępne są rozwiązania sprzętowe które pozwalają przenieść przetwarzanie danych z CPU do peryferii sprzętowych.
4. Dojrzałość systemów nawigacyjnych pozwalających na autonomiczne działanie oraz przyjęcie wspólnych punktów odniesienia poprzez wzajemną korekcję systemów pracujących w oparciu o różne sensory.
5. Interoperacyjność systemów pozwoliła na łatwą wymianę oraz późniejszą interpretację danych pochodzących z różnych sensorów.
6. Zaawansowane i ergonomiczne narzędzia pozwalające na prezentację danych z sensorów dla użytkowników.

2.7.2. Klasyfikacja metod fuzji danych

Wraz ze wzrostem ilości sensorów operujących w różnych domenach, dostarczających coraz większej ilości danych, pojawiły się problemy jak poradzić sobie z ich interpretacją. Trudności wynikały głównie z ograniczonej możliwości percepcji człowieka za równo w kwestii ilości danych jak i prędkości ich przetwarzania. Jak większość nowych zagadnień technicznych, termin fuzji danych pojawił się po raz pierwszy w zakresie zastosowań militarnych oraz bezpieczeństwa. Został on sformułowany w pierwszej połowie lat 80, w ramach rozwiązań rozwijanych dla potrzeb Amerykańskiego Departamentu Obrony (DoD – Department of Defence) jako “Sensor Data and Information Fusion” przez Joint Directors of Laboratories (JDL) będącym ciałem doradczym wspomnianej instytucji [50]. Zaprezentowany wtedy model fuzji informacji oraz danych z sensorów, jest najpopularniejszym sposobem klasyfikacji algorytmów

i jest nadal wykorzystywany we współczesnych publikacjach traktujących o tej tematyce [51] [52]. Został on zaprezentowany na rysunku 15.



Rysunek 15. Przegląd modelu JDL fuzji danych uwzględniający pełny łańcuch funkcjonalny rozproszonych sensorów, bazy danych oraz informacji przekazywanej użytkownikom [53]

Jego celem jest przedstawienie fuzji danych w ujęciu pełnego systemu od źródeł informacji do podejmowanych decyzji i komunikacji ich użytkownikowi. Wyodrębnia on pięć poziomów przetwarzania, a także bazę danych i magistralę komunikacyjną dla danych. Model uwzględnia również źródła informacji takie jak: dane z sensorów, dane z baz danych itp.; interakcję z odbiorcą danych wyjściowych z systemu oraz system zarządzania bazą danych. Pięć poziomów przetwarzania danych zdefiniowano jak poniżej:

1. Poziom 0 - wstępne przetwarzanie danych źródłowych, inaczej preprocesing, głównym celem jest redukcja danych poprzez wybór tylko istotnych.
2. Poziom 1 – ekstrakcja, grupowanie, filtracja danych z poziomu 0, w efekcie których otrzymujemy spójne struktury danych.
3. Poziom 2 – na tym poziomie odbywa się interpretacja otrzymanych danych, przykładowo detekcja rozpoczęcia zdarzeń.
4. Poziom 3 – analiza wpływu wykrytych zdarzeń na system.

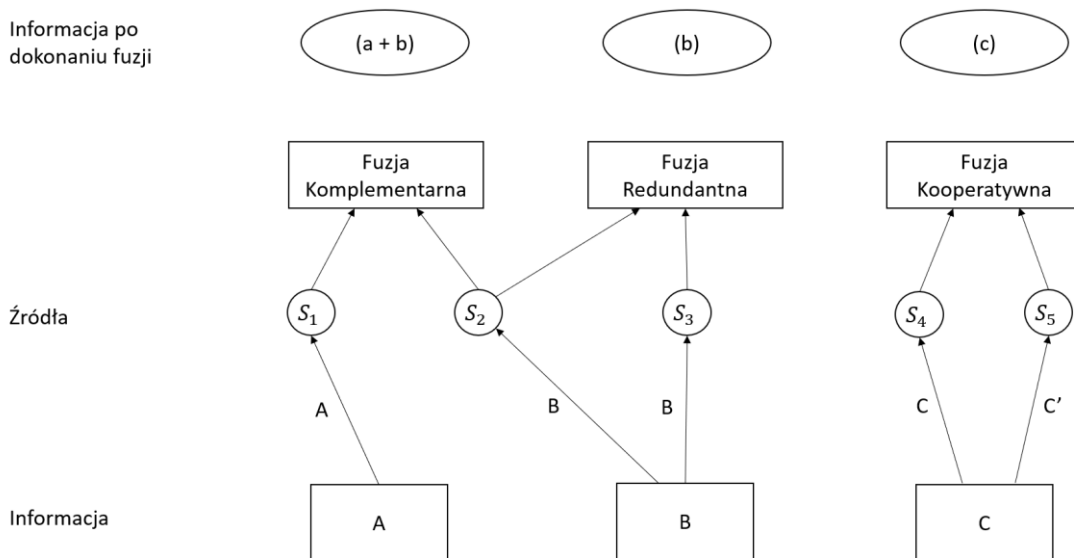
- Poziom 4 – usprawnia procesy na poziomach 0-3 oraz pozwala na zarządzanie sensorami.

Wymienione pięć poziomów zostały podzielone na dwie grupy – fuzji wysoko oraz niskopoziomowej. Wysokopoziomowa fuzja danych rozpoczyna się zazwyczaj od poziomu 2.

Obszerność tematu fuzji danych skłania do podziału i kategoryzacji stosowanych metod oraz modeli. Z uwagi na interdyscyplinarność zagadnienia, a także różnorodność wykorzystywanych systemów sensorycznych odpowiednia klasyfikacja, pozwala na usystematyzowanie istniejących rozwiązań. Dodatkowo umożliwia przegląd istniejących metod oraz wybór najodpowiedniejszego podejścia w przypadku projektowania nowej aplikacji. W kolejnych akapitach przedstawione zostały inne metody klasyfikacji metod fuzji danych.

Kolejną zaproponowaną metodą klasyfikacji, jest przedstawiona na rysunku 16 kategoryzacja ze względu na relację między danymi źródłowymi. Możemy wyszczególnić trzy kategorie fuzji danych:

- Komplementarna
- Redundantna
- Kooperatywna



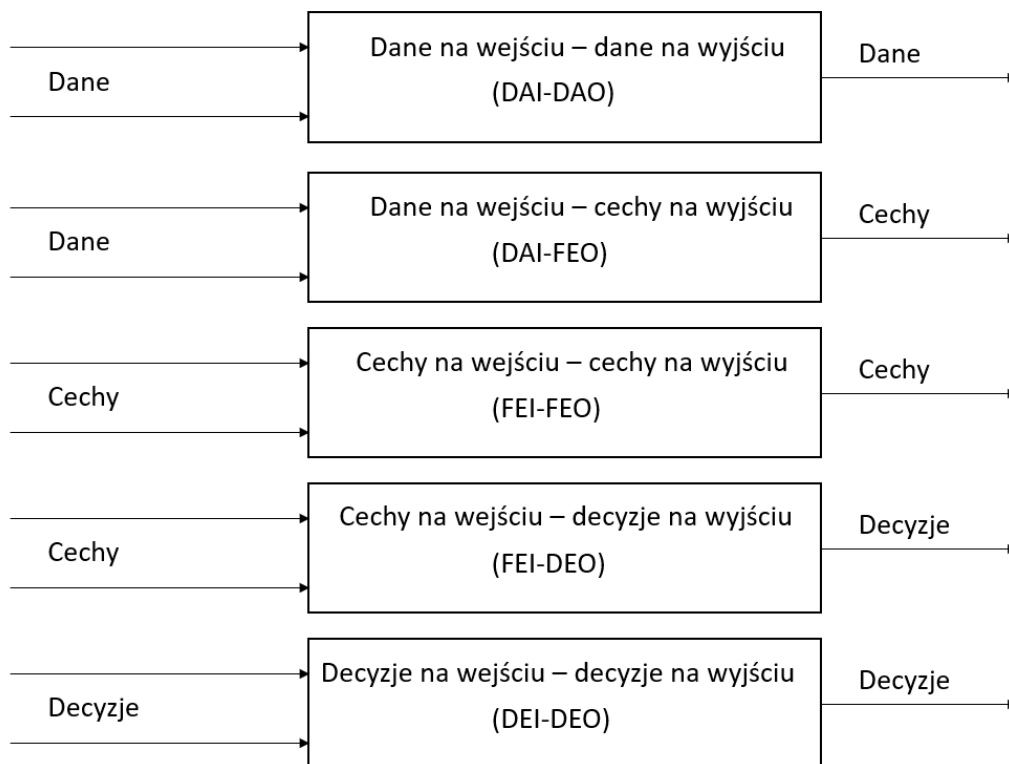
Rysunek 16. Klasyfikacja ze względu na relację między danymi źródłowymi [53]

W przypadku fuzji komplementarnej wykorzystywane są dane z dwóch lub więcej różnych sensorów, dostarczające dane odnoszące się do różnych aspektów lub obszarów

badanej przestrzeni. Dzięki temu możliwe jest uzyskanie pełniejszej informacji. Fuzja redundantna wykorzystuje dane z czujników tego samego rodzaju, badających ten sam obszar lub obszar pokrywający się w jakiejś części. Takie podejście zwiększa pewność otrzymanych danych oraz niezawodność systemu. Dodatkowo w przypadku dysfunkcji jednego z torów pomiarowych pozwala na wykrycie usterki, a także w niektórych przypadkach na dalszą pracę systemu w trybie awaryjnym. Fuzja kooperatywna polega na połączeniu danych dotyczących tego samego obiektu pochodzących z różnych sensorów, dzięki temu możemy uzyskać bardziej złożoną informację. Dla wyjaśnienia różnicy między fuzją redundantną i kooperatywną posłużę się przykładami. Przykładem fuzji redundantnej może być system, który wykorzystuje zdjęcia z dwóch kamer. W przypadku fuzji kooperatywnej zostaną wykorzystane zdjęcia z jednej kamery oraz nagranie dźwięku z mikrofonu zlokalizowanego w pobliżu kamery.

Innym sposobem klasyfikacji metod fuzji danych jest typ danych wejściowych oraz wyjściowych [54]. Systemy mogą wykorzystywać wiele z wymienionych metod na różnych etapach przetwarzania danych, dlatego elementy przedstawione na rysunku 17, mogą stanowić swego rodzaju bloki, z których możliwe jest zbudowanie końcowego systemu. Zaproponowane 5 kategorii to:

1. Dane na wejściu – dane na wyjściu (ang. Data Input – Data Output, DAI-DAO)
2. Dane na wejściu – cechy na wyjściu (ang. Data Input – Feature Output, DAI-FEO)
3. Cechy na wejściu – cechy na wyjściu (ang. Feature Input – Feature Output, FEI-FEO)
4. Cechy na wejściu – decyzje na wyjściu (ang. Feature Input – Decision Output, FEI-DEO)
5. Decyzje na wejściu – decyzje na wyjściu (ang. Decision Input – Decision Output, DEI-DEO)



Rysunek 17. Klasyfikacja ze względu na dane wejściowe oraz wyjściowe zaproponowana przez Dasarathy'ego [54]

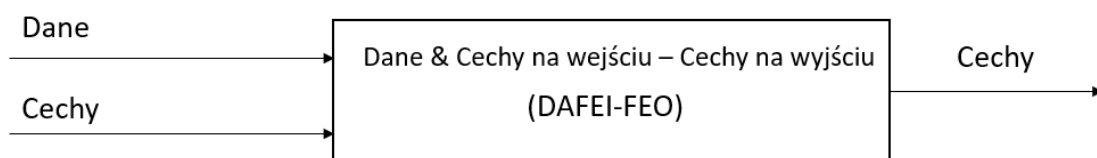
Dwie pierwsze kategorie występują na początku łańcucha przetwarzania danych, kiedy przetwarzane są dane wejściowe. W przypadku (DAI-DAO) zazwyczaj mamy odczynienia również z kategorią danych redundantnych. Dzięki temu możliwe jest zweryfikowanie i uzupełnienie otrzymanych danych na wstępnym etapie ich przetwarzania. W przypadku (DAI-FEO) otrzymujemy cechy opisujące badany układ takie jak np. prędkość lub położenie. Trzecia kategoria (FEI-FEO) określana również jako fuzja cech (Feature fusion), pozwala na uzyskanie nowych informacji, gdy otrzymamy cechy z dwóch systemów. Dzięki fuzji tych dwóch cech możemy otrzymać trzecią. Przykładowo wykorzystując cechy z dwóch układów sensorycznych w postaci kamery oraz radaru – po ekstrakcji cech w postaci obszaru obiektu widzianego przez kamerę oraz zmierzonej radarem odległości poszczególnych punktów obiektu – możliwe jest uzyskanie dodatkowej cechy w postaci kształtu powierzchni obserwowanego obiektu. Kolejną kategorią jest (FEI-DEO), która jest najpowszechniej wykorzystywana. Zazwyczaj cechy są uzyskiwane z danych źródłowych oddzielnie za pomocą odseparowanych algorytmów. Fuzja jest dokonywana na ostatnim etapie przed uzyskaniem ostatecznych danych odpowiadającym funkcji całego systemu. Ostatnia z zaproponowanych kategorii (DEI-DEO) może zostać określona podobnie jak w przypadku fuzji cech – fuzją decyzji (Decision Fusion).

Jest ona stosowana na końcowym etapie przetwarzania danych. Dzięki temu decyzja podjęta przez podsystemy może zostać zmodyfikowana lub system może podjąć całkowicie nową decyzję na podstawie bardziej kompletnej informacji.

Kolejnym typem kategoryzacji jest poziom abstrakcji danych źródłowych. Klasyfikacja jest bardzo podobna do poprzedniej. Istnieją różne podejścia do tej kwestii jednak najczęściej przyjmuje się trzy lub cztery poziomy:

1. Poziom 1 - dane
2. Poziom 2 - dane przetworzone
3. Poziom 3 - cechy
4. Poziom 4 – decyzje

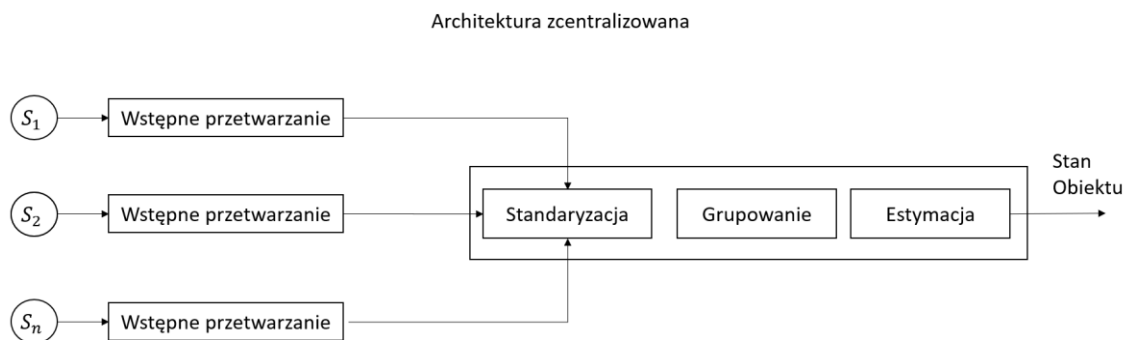
Najniższym poziomem są dane otrzymane bezpośrednio z czujników (poziom 1) oraz wstępnie przetworzone (poziom 2). Przykładowo „surowe” sygnały z matrycy CMOS będą stanowiły poziom 1, natomiast wartości reprezentujące piksele będą odpowiadały poziomowi drugiemu. Wymienione poziomy odnosząc się do wcześniejszej kategorii odpowiadają pojęciu danych. Jak wcześniej wspomniano rozróżnienie na poziomy 1 i 2 dla prostszych układów sensorycznych nie jest konieczne. Następnym poziomem abstrakcji jest poziom cech, odnosząc się do wcześniejszego przykładu będzie to kontur lub kształt obiektu. W najbardziej abstrakcyjnym poziomie 4, fuzja jest dokonywana na danych dotyczących decyzji. Należy zwrócić uwagę na fakt, że istnieją międzypoziomowe algorytmy fuzji danych, które łączą dane bezpośrednio z czujników z cechami. Odnosząc się do poprzedniej kategorii możemy uzyskać podejście zaprezentowane na rysunku 18.



Rysunek 18. Przykładowe podejście do fuzji danych wykorzystujące dane wejściowe na różnym poziomie abstrakcji

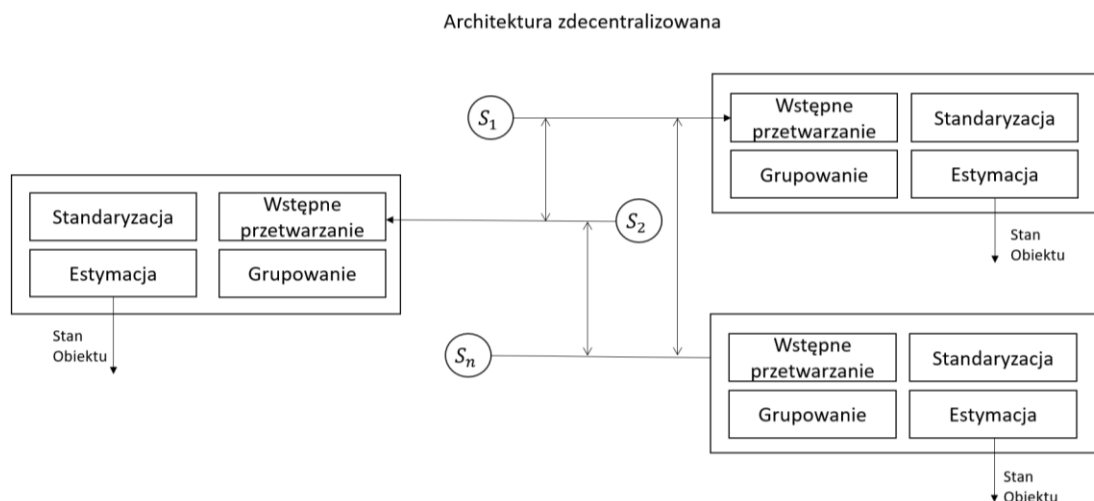
Ostatnim z przedstawionych typów klasyfikacji jest typ architektury systemu fuzji. Możliwość łączenia różnych podejść w fuzji danych sprawia, że ilość specyficznych architektur jest niemal nieograniczona. Architektura fuzji danych odpowiada na pytanie, w którym obszarze systemu zostanie przeprowadzona fuzja danych. Z tego względu zaproponowano trzy

umowne kategorie dotyczące architektury fuzji – zcentralizowaną, zdecentralizowaną, rozproszoną. Poniżej przedstawiono poglądowe schematy wymienionych architektur (rysunki 19, 20, 21).



Rysunek 19. Przykład architektury zcentralizowanej [51]

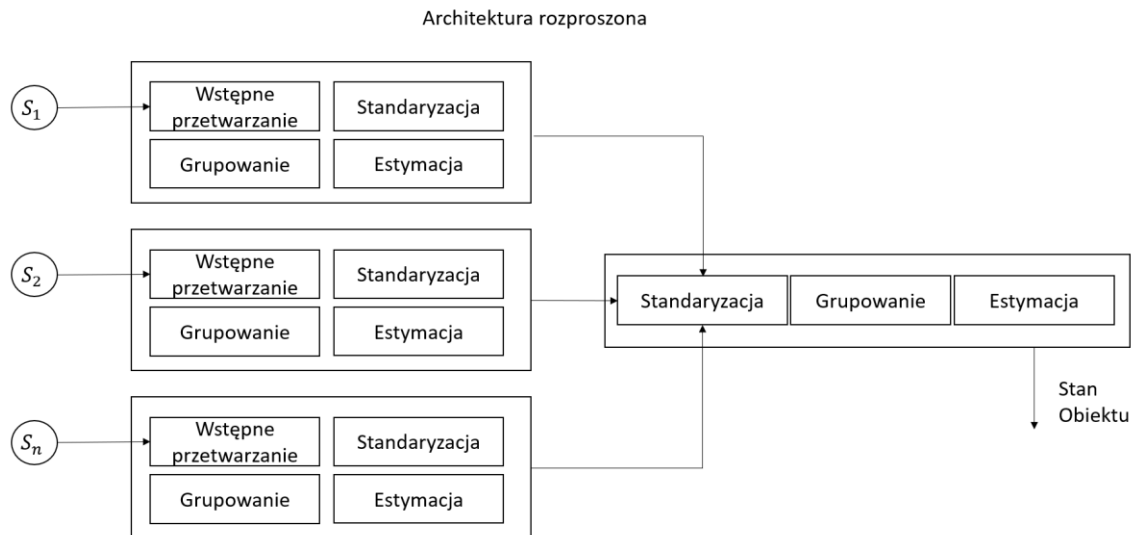
Architektura zcentralizowana zakłada przeprowadzenie wszystkich operacji dotyczących fuzji danych przez jedną jednostkę obliczeniową. Preprocessing może odbywać się poza centralną jednostką obliczeniową tak jak na przedstawionym rysunku 19 lub w ramach centralnej jednostki. Architektura ta jest wrażliwa na opóźnienia w komunikacji, ponieważ występuje wtedy ryzyko utraty synchronizacji pomiarów. Zazwyczaj w systemach wykorzystujących kilka źródeł danych problemy z opóźnieniami nie są wykluczające, co zmienia się proporcjonalnie wraz ze wzrostem liczby kanałów wejściowych [51].



Rysunek 20. Przykład architektury zdecentralizowanej [51]

W systemach ze zdecentralizowaną architekturą (rysunek 20), występuje kilka jednostek obliczeniowych, często tyle samo, ile sensorów. Zazwyczaj każda z jednostek

obliczeniowych dokonujących fuzji otrzymuje dane również z innych sensorów. Największą wadą tej architektury jest konieczność wielokrotnego przesyłania tych samych informacji, co powoduje znaczący nakład zasobów poświęconych na komunikację. Bezpośrednio z tego faktu wynikają trudności ze skalowalnością systemów o tego rodzaju architekturze.



Rysunek 21. Przykład architektury rozproszonej [51]

Architektura rozproszona (rysunek 21) stanowi pewnego rodzaju fuzję architektury zcentralizowanej i zdecentralizowanej, przez co eliminujemy część ich wad. Z powodu rozproszenia procesu fuzji danych potrzebujemy jednak wiele jednostek obliczeniowych, przez co uzyskujemy najwyższy sumaryczny koszt obliczeniowy. Podejście to jednak jest najczęściej spotykane w praktyce, ponieważ pozwala na modułowość poszczególnych systemów sensorycznych co ułatwia współpracę między różnymi zespołami, oraz jednostkami organizacyjnymi.

Wymienione typy stanowią tylko ogólną klasyfikację. Istnieje wiele dedykowanych rozwiązań dla konkretnych aplikacji. Zostały wymienione one w pracy opisującej architektury stosowane w fuzji [55]. Opisane w niniejszej pracy doktorskiej stanowią jedynie ogólną klasyfikację.

W obszarze wykorzystywanych algorytmów fuzja danych również obejmuje bardzo wiele rozwiązań. Wynika to z różnorodności wymienionych wyżej rozwiązań oraz z niemal nieograniczonej skalowalności. Przykładowo dla tylko dwóch sensorów o identycznych

parametrach, wyznaczenie średniej arytmetycznej ze zmierzonych przez nie wielkości, jest prostą formą fuzji danych. Dzięki analizie wartości średniej, możemy wykryć np. awarię jednego z czujników albo uzyskać dokładniejsze informacje na temat obserwowanego zjawiska. Z drugiej strony skali mamy systemy monitorujące przestrzeń powietrzną poddające dane fuzji na wielu poziomach za pomocą różnych algorytmów. W tabeli 2 przedstawiono kilka najpopularniejszych kategorii algorytmów stosowanych dla potrzeb fuzji danych.

Tabela 2. Przykładowe algorytmy fuzji danych

Algorytm	Zastosowanie	Korzyści
Metody statystyczne (np. średnia, odchylenie standardowe)	<p>Wiele sensorów mierzących tą samą wielkość.</p> <p>Odczyty z pojedynczego sensora.</p> <p>Wykorzystywana najczęściej na poziomie najniższym poziomie abstrakcji danych</p>	Przez odnoszenie wartości poszczególnych sensorów do średniej możliwe jest odrzucenie niepoprawnych pomiarów.
Modele Markova	<p>Najczęściej o określania stanu systemu na podstawie stanów poprzednich.</p> <p>Niektóre modele wykorzystywane są do ekstrakcji cech takich jak odległość między pikselami na obrazie w przypadku algorytmu np. Potts Markov Random Field</p>	Szybkie działanie w przypadku ograniczonej liczby stanów
Filtry Kalmana	<p>Estymacja stanów systemów takich jak prędkość, położenie.</p> <p>W fuzji danych stosowany zazwyczaj na poziomie cech</p>	Pozwala uwzględnić wpływ zakłóceń oraz charakterystyki konkretnych sensorów na pomiar, przez co otrzymane estymaty są bliższe rzeczywistym pomiarom
Sieci neuronowe	Przy wielu kanałach wejściowych, niezależnie od poziomu abstrakcji danych (dane z sensorów, cechy, decyzje)	Łatwość łączenia danych z różnych sensorów oraz o różnym poziomie abstrakcji
Sieci Bayesowskie	Zazwyczaj stosowane przy fuzji danych na poziomie abstrakcji decyzyjnym	Pozwala na uzyskanie informacji na podstawie prawdopodobieństwa warunkowego wielu zdarzeń zależnych od siebie

3. Cel i zakres pracy

Niniejsza rozprawa doktorska, która powstała w ramach programu „Doktorat Wdrożeniowy”, jest efektem poszukiwań i analizy nowych metod oraz systemów rozpoznających gesty, które mogłyby być zastosowane w branży motoryzacyjnej. Ze względu na portfolio produktowe zajmującej się komponentami mechatronicznymi interfejsu użytkownika firmy Merit, która była partnerem komercyjnym realizowanego zadania, skupiłem się na poszukiwaniu rozwiązań niewykorzystujących kamery. Wymienione okoliczności narzuciły badaniom konkretny kierunek. System rozpoznający gesty musiał być możliwy do wykonania przy użyciu masowo produkowanych komponentów, które można stosować we wnętrzu samochodu. Dodatkowo system powinien być łatwy do zintegrowania z innymi modułami występującymi w pojazdach.

Znaczna część prac podczas realizacji doktoratu polegała na poszukiwaniach i wstępnej ewaluacji technologii, które mogłyby być zastosowane w pojazdach na masową skalę. Poniżej przedstawiono wymagania, które musiały być możliwe do spełnienia, aby rozwiązanie mogło być brane pod uwagę.

- Poprawna praca systemu w zakresie temperatur od -40 do 85 °C.
- Integracja systemu z istniejącymi elementami wnętrza pojazdu.
- Koszt dodania systemu do rozpoznawania gestów nie powinien zwiększyć kosztu podstawowego modułu o więcej niż 100%.
- System powinien być możliwy do zintegrowania z produktami z portfolio partnera doktoratu – Firmy Merit, czyli przełącznikami i modułami sterującymi zlokalizowanymi wewnątrz samochodu.

Uwzględniając powyższy kontekst rozpoczęto poszukiwania odpowiedniej technologii. Zaowocowały one wytypowaniem 5 kategorii, które mogłyby spełnić postawione oczekiwania:

- Systemy wykorzystujące elektrody pojemnościowe
- Systemy oparte o promieniowanie elektromagnetyczne w zakresie podczerwieni
- Systemy wykorzystujące radar
- Systemy oparte o fale ultradźwiękowe
- Systemy zlokalizowane na użytkowniku

W ramach prowadzonych badań, przeprowadzono krytyczną analizę literatury w tym rozwiązaniach dostępnych komercyjnie co opisano w rozdziale „Przegląd literatury oraz istniejących rozwiązań”. W ramach krytycznego przeglądu, opisano główne cechy oraz ograniczenia poszczególnych rozwiązań wraz z możliwym wykorzystaniem w praktyce przemysłowej. Celem rozdziału było wykazanie luki badawczej możliwej do wypełnienia przy spełnieniu założonych wymagań. Podsumowując dokonany przegląd literatury oraz temat realizowanego doktoratu, wykazano brak rozwiązań dla branży samochodowej, które pozwalałyby na uzyskanie podobnych wyników jak rozwiązanie bazujące na systemach wizyjnych jak te zastosowane w BMW, ale bez wykorzystywania systemów wizyjnych. Na rynku istnieją systemy, które są w stanie rozpoznać mniejszą liczbę gestów w bardzo niewielkiej odległości od sensorów. Po wstępnych testach wyżej wymienionych technologii zdecydowano o budowie dwóch systemów wykorzystujących czujniki pojemnościowe oraz radar.

Rozdział „System do wykrywania gestów oparty o czujniki pojemnościowe” przedstawia autorskie rozwiązanie pozwalające na wykrywanie gestów w trzech wymiarach. Algorytm oparty o sieć neuronową został uruchomiony na systemie wbudowanym co pozwoliło na jego montaż i dalsze testy w samochodzie testowym. System spełnił główne założenia. Największym jego ograniczeniem jest jednak niewielki zakres działania (5-7 cm).

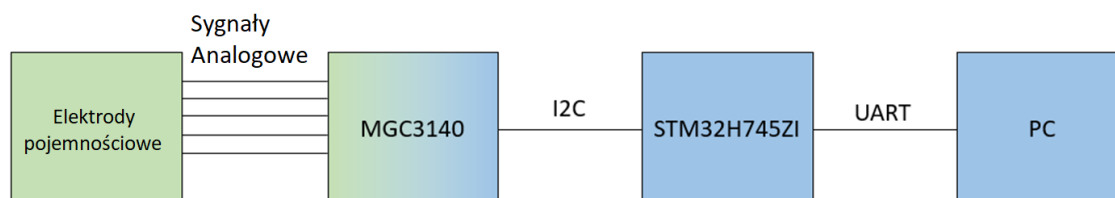
W rozdziale „System do rozpoznawania gestów oparty o radar” opisano rozwiązanie wykorzystujące radar. Prace nad tą technologią rozpoczęły się w końcowej fazie realizowanego doktoratu, dlatego też nie przeprowadzono pełnego wdrożenia tej technologii (testy w samochodzie). Niemniej jednak stworzone rozwiązanie również spełniło wymagane założenia. W tym wypadku największym wyzwaniem było rozpoznawanie gestów wykonywanych wzdłuż jednej z osi (trudność rozpoznania gestów przesunięcie w dół oraz w górę).

Ze względu na ograniczenia każdej z wybranych technologii, postawiono tezę, że fuzja danych z dwóch systemów pozwoli na uzyskanie lepszych rezultatów niż w przypadku wykorzystania każdego z nich osobno. Dotyczy to głównie skuteczności wykrywanych gestów, co pozwoliłoby na większą niezawodność systemu pozwalającą na zastosowanie w pojeździe. Dodatkowo postawiono tezę, że fuzja danych pozwoli na osiągnięcie lepszych rezultatów w przypadku większej liczby gestów. Odpowiedzi na postawione założenia zostały zawarte w rozdziale „Fuzja danych z systemu pojemnościowego oraz radarowego”.

W ostatnim rozdziale pracy znajduje się podsumowanie prowadzonych prac oraz omówione zostały możliwe dalsze kierunki rozwoju.

4. System do rozpoznawania gestów oparty o czujniki pojemnościowe

Pierwszym zrealizowanym w ramach niniejszej pracy systemem do rozpoznawania gestów jest system oparty o technologię pojemnościową. Niniejszy rozdział zawiera opis zarówno pełnego utworzonego systemu jak i jego komponentów składowych.



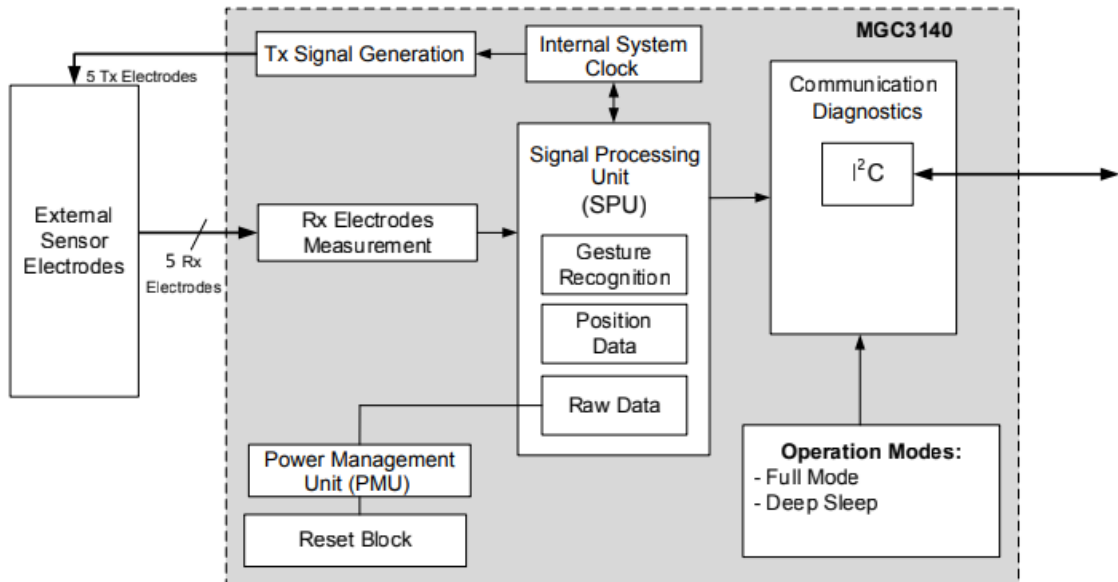
Rysunek 22. Schemat systemu rozpoznającego gesty w oparciu o sygnał z elektrod pojemnościowych. Kolor zielony oznacza część analogową, kolor niebieski cyfrową. Za konwersję sygnałów analogowych do postaci cyfrowej odpowiada układ MGC3140

System składa się z: płytki PCB z elektrodami, układu MGC3140 przetwarzającego sygnały analogowe oraz mikroprocesora STM32H745ZI połączonego z komputerem. Na rysunku 22, kolorem zielonym zaznaczono część analogową, natomiast niebieskim cyfrową.

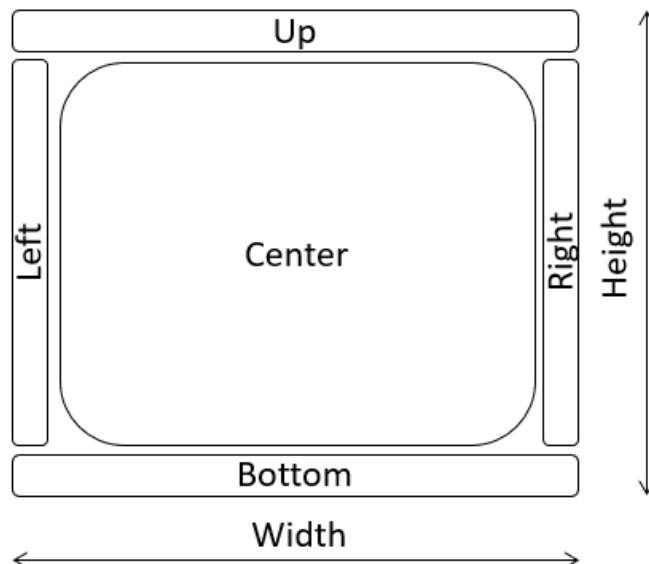
4.1. Elektrody oraz układ sterujący MGC

Układ sterujący MGC3140 kontroluje działanie elektrod oraz dokonuje pomiarów zmian pojemności na każdej z elektrod układu pojemnościowego. Schemat układu zaprezentowano na rysunku 23. Parametry pracy mogą zostać zaprogramowane z zewnątrz przez wgranie konfiguracji za pośrednictwem interfejsu komunikacyjnego. Na rysunku 16 przedstawiono schemat wewnętrzny zastosowanego układu. Układ posiada moduły pozwalające na rozpoznawanie gestów oraz pozycji obiektu znajdującego się w przestrzeni roboczej elektrod pojemnościowych. Wykorzystanie tych modułów nie przyniosło zadowalających rezultatów. Z tego względu, wykorzystano jedynie moduł „Raw data” (surowe dane), zapewniający dostęp do danych bezpośrednio zmierzonych przez przetwornik analogowo cyfrowy na każdym z kanałów. Surowe dane zostały następnie przesłane przez magistralę I2C [56]. Pomiar wartości analogowych był wykonywany co 5 ms – jest to największa możliwa częstotliwość wynikająca

z konstrukcji układu. Procesor główny (STM32H745ZI) co 5 ms wysyłał zapytanie o dane z pomiaru. Urządzenie odsyłało pięć czterobajtowych wartości w 32-bitowym formacie zmiennoprzecinkowym (float). Przesłane wartości odpowiadały pojemności zmierzonej dla każdej z 5 elektrod. Układ MGC po przesłaniu danych, natychmiast rozpoczynał kolejny pomiar.



Rysunek 23. Schemat układu MGC 3140, za dokumentacją [16]

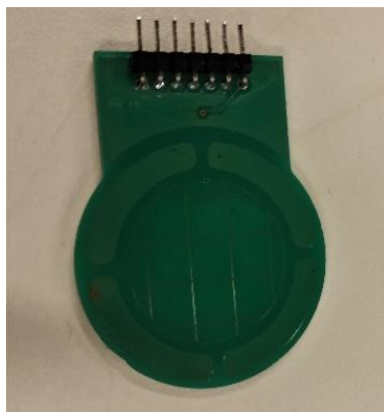


Rysunek 24. Rozmieszczenie elektrod

Zastosowano poniższą konfigurację układu:

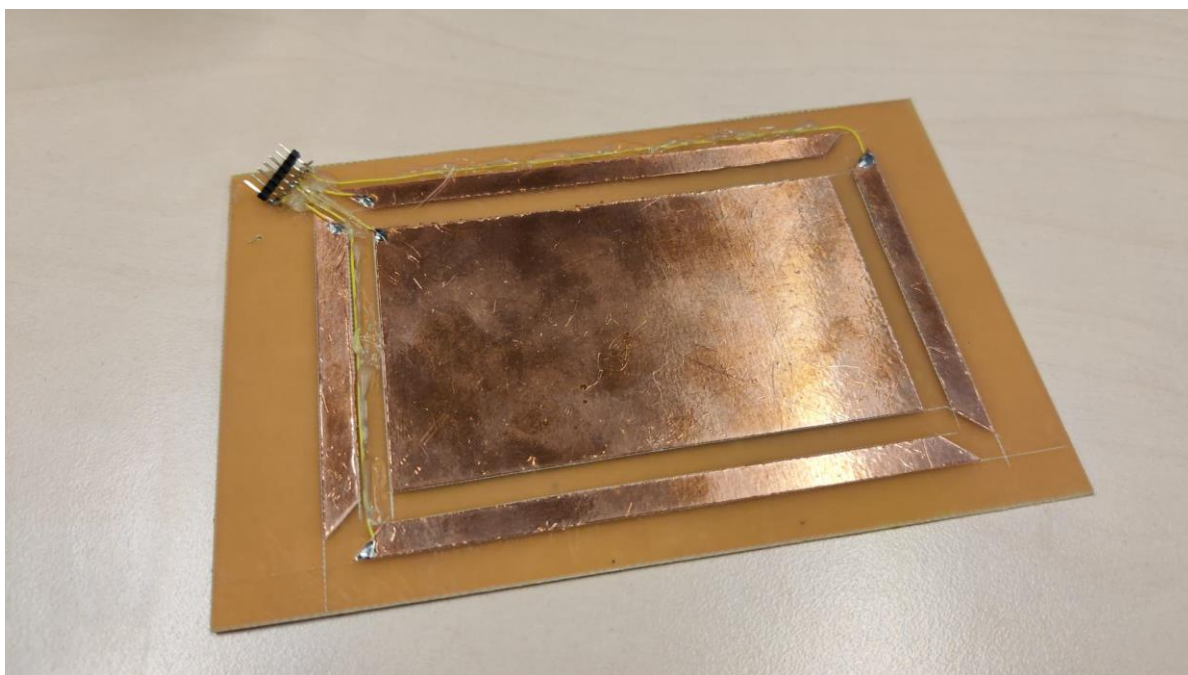
- Elektroda nadawcza – kanał 1
- Elektrody odbiorcze – kanały 1 (powtórzony),2,3,4,5
- Połączenie kanałów z elektrodami (rozmieszczenie elektrod przedstawiono rysunku 24):
 - Kanał 1 – elektroda „Center” (środkowa)
 - Kanał 2 – elektroda „Up” (górna)
 - Kanał 3 – elektroda „Down” (dolna)
 - Kanał 4 – elektroda „Left” (lewa)
 - Kanał 5 – elektroda „Right” (prawa)
- Częstotliwości sygnału podawanego na elektrodę nadawczą: 42, 43, 44, 45, and 100 kHz – Zastosowanie pięciu różnych częstotliwości pozwala na uodpornienie systemu na zakłócenia pochodzące spoza układu pomiarowego. Częstotliwości są zmieniane cyklicznie w narastającej kolejności.
- Typ danych – Signal Deviation (SD) – sygnały otrzymane z elektrod po wykonanym przez układ preprocesingu i kalibracji. Uwzględnia warunki początkowe, gdy w przestrzeni sensora nie znajduje się obiekt – wartość zmierzona w takich warunkach stanowi wartość referencyjną (ang. baseline). Wartości zwracane podczas pomiarów stanowią różnicę wartości bieżących uzyskiwanych z przetwornika analogowo-cyfrowego oraz wartości referencyjnej.

Podczas prac nad systemem przetestowano kilka różnych układów elektrod w postaci płytek PCB. Punktem odniesienia była elektroda dostarczona w zestawie ewaluacyjnym od firmy Microchip, której pole powierzchni obszaru roboczego wynosiło 95 x 60 mm (rysunek 24 - odpowiednio Width oraz Height). Skonstruowano kilka innych układów elektrod pod kątem zastosowania w różnych produktach firmy Merit. Pierwszą z nich była okrągła płytka widoczna na rysunku 25. Celem jej zaprojektowania było potencjalne zastosowanie jej na powierzchni pokrętła, w taki sposób, aby do standardowej funkcjonalności pokrętła dodać gesty. Niestety przeprowadzone testy, pokazały, że efektywny zasięg wykrywanych obiektów w tym przypadku wynosił tylko ok. 2-3 cm nad powierzchnią, co jest dyskwalifikujące dla systemu rozpoznającego gesty 3D. Przeprowadzono również testy w trybie pracy „Boost”, gdzie napięcie na elektrodzie nadawczej było konwertowane z 3.3 na 12-18V. Niestety wraz ze wzrostem wartości sygnałów, zakłócenia sygnałów wzrosły do takiego stopnia, że uniemożliwiło zastosowanie tego typu elektrod.



Rysunek 25. Elektroda pojemnościowa o średnicy 30 mm

Przetestowano również elektrodę o zwiększonym rozmiarze (szerokość 135mm, wysokość 80mm), którą pokazano na rysunku 26. Teoretycznie większa powierzchnia powinna zapewnić większy zasięg działania. Referencyjna elektroda zapewniała zasięg wykrywania obiektów w okolicach 5-7 cm, co nie jest znakomitym wynikiem, ale pozwoliło na skuteczną implementację systemu. W przypadku zwiększonej powierzchni efektywny zakres zwiększył się. Niestety poziom zakłóceń wzrósł także, przez co zaniechano rozwoju prac z wykorzystaniem elektrody o większych rozmiarach.



Rysunek 26. Elektroda o wymiarach 135 x 80 mm

4.2. Zabudowa elektrod

Analizując uwarunkowania konieczne do końcowej aplikacji systemu, istotnym warunkiem jest jego zabudowanie za powierzchnią dekoracyjną, która ukryje część elektroniczną. Generalnie pojemność systemów dotykowych mocno zależy od przenikalności elektrycznej otoczenia elektrod (ϵ_r). Zastosowanie zbyt grubej pokrywy lub z materiału o niskiej przenikalności elektrycznej podobnie jak w klasycznym modelu kondensatora zwiększy pojemność [57]. Spowoduje to spadek czułości systemu, ponieważ aby została zmierzona jakakolwiek zmiana pojemności, zaburzenie pola elektrycznego musi być bardziej znaczące. Mając na uwadze powyższe fakty, przeprowadzono testy osłon elektrod z materiałów o różnej grubości. Testy przeprowadzono używając elektrody referencyjnej osłoniętej różnymi materiałami w konfiguracjach przedstawionych w tabeli 3. Materiały wybrano ze względu na obecne i potencjalne zastosowanie we wnętrzu samochodu.

Tabela 3. Wartości maksymalne sygnałów zmierzonych dla elektrod dla różnych konfiguracji zabudowy elektrod

Typ osłony	Maksymalna wartość sygnału pomiarowego elektrod (wartość z przetwornika ADC w trybie Signal Deviation)	Porównanie wartości maksymalnej otrzymanej dla sygnałów
PA6 (poliamid-6) 1mm	721	100%
PA6 3 mm	686	95,1%
PA6 1mm + pianka PE (Polietylen) 1mm	575	79,8%
PA6 1mm + skóra bydlęca 1mm	609	84,4%

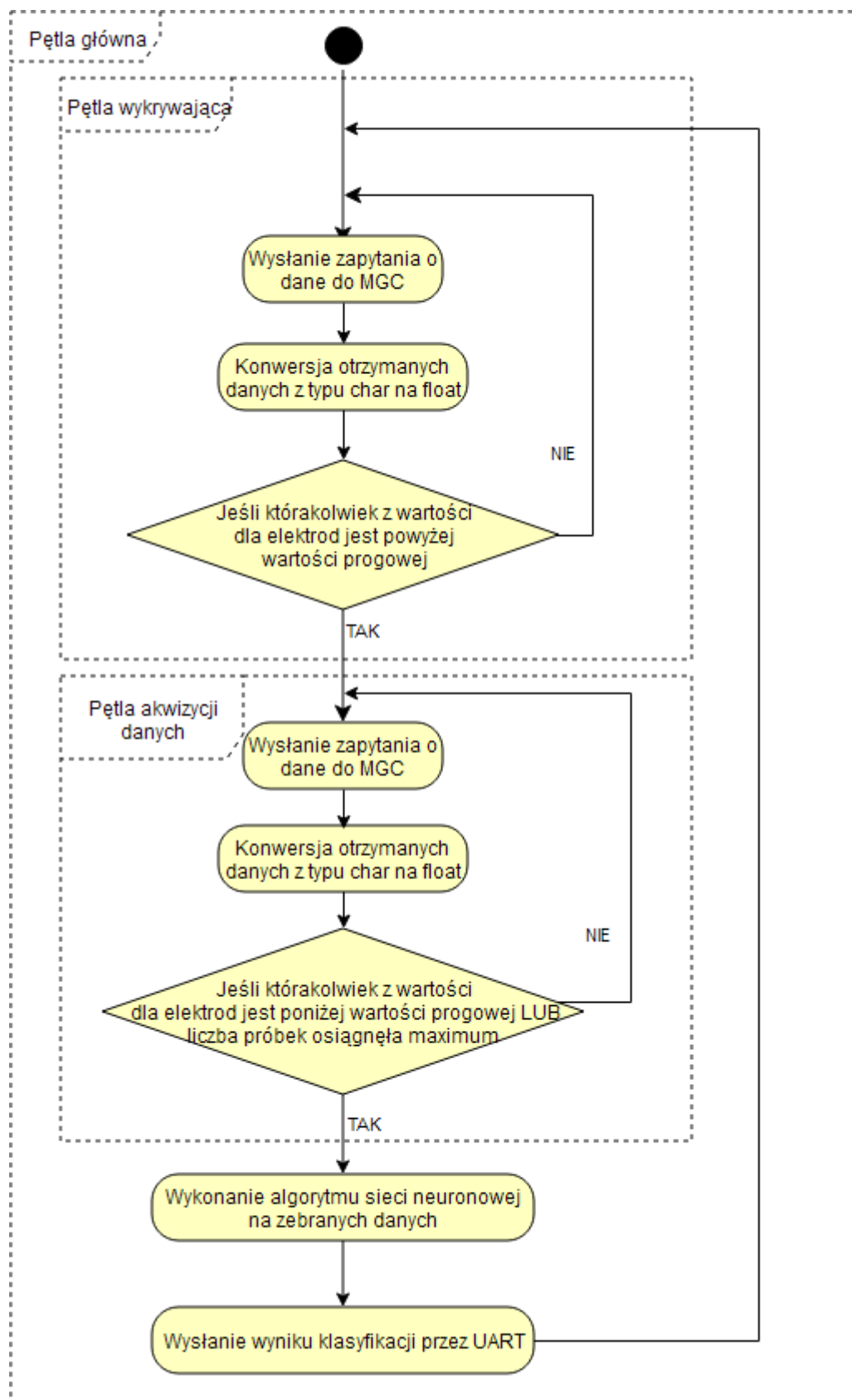
Wyniki z powyższej tabeli pokazują typową zależność dla systemów pojemnościowych polegającą na spadku czułości wraz ze wzrostem grubości materiału izolującego. Ze względu na operowanie na względnie dużych pojemnościach (w porównaniu do systemów dotykowych 2D), nawet zastosowanie osłon o grubości 3mm nie powoduje spadku czułości systemu w stopniu uniemożliwiającym jego działanie. Ostatecznie wybrano wariant z osłoną wykonaną z PA6 o grubości 1 mm, który charakteryzował się najwyższą czułością. Jest to spowodowane niskim zasięgiem wykrywania przez systemy pojemnościowe w odniesieniu do innych technologii.

4.2.1. Mikroprocesor główny

W badaniach wykorzystano mikroprocesor STM32H745ZI ze względu na jego kompatybilność z narzędziem X-CUBE-AI wspierającym konwersję oraz uruchomienie modeli sieci neuronowych na mikrokontrolerach STM. Głównym zadaniem tego procesora było cykliczne odczytywanie danych z układu MGC 3140 a następnie wykonywanie algorytmu sieci neuronowej, która dokonywała klasyfikacji wykonanych gestów na podstawie zebranych sygnałów. Wynik klasyfikacji następnie był przesyłany do komputera, na którym dedykowana aplikacja wyświetlała informację nt. gestu, który uzyskał największą wartość prawdopodobieństwa podczas klasyfikacji.

4.2.2. Algorytm

Na rysunku 27 przedstawiono algorytm, który został był cyklicznie wykonywany na procesorze. Składa się on z 2 pętli, które uruchamiane są w pętli głównej programu. Pierwsza pętla – „Detection loop” (Pętla detekcji) wykonuje się z 5ms interwałem. Jej głównym celem jest wykrycie pojawienia się dłoni w przestrzeni roboczej. Wykrywanie było dokonywane w przypadku przekroczenia wartości progowej przez jeden z pięciu sygnałów odpowiadającym poszczególnym elektrodom. Oznacza to, że pojemność w układzie pomiarowym została zmieniona w takim stopniu, że najprawdopodobniej mamy do czynienia z pojawieniem się obiektu w obszarze roboczym. W efekcie algorytm przechodzi od drugiej pętli – „Gesture collection loop” (Pętla zbierania danych), której zadaniem jest zapisanie sekwencji danych odpowiadającej wykonaniu gestu w obszarze roboczym. Okres jej wykonywania wynosi również 5 ms. Dane sekwencji są zapisywane do buforów po każdej ramce otrzymanej od MGC. Jeśli wartość maksymalna spośród wszystkich kanałów spadnie poniżej wartości progowej, pomiar jest zatrzymywany. Następnie bufor z zebranymi danymi są przekazywane do algorytmu sieci neuronowej, dla której stanowią wektor cech. W przypadku, jeśli sekwencja jest krótsza niż 120 próbek, bufor są dopełniane zerami. Po dokonaniu klasyfikacji algorytm ponownie wraca do pętli wykrywającej.

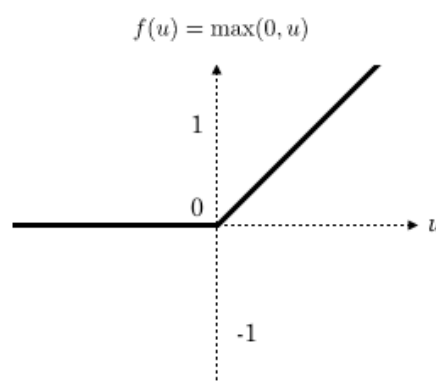


Rysunek 27. Algorytm wykonywany na mikrokontrolerze realizujący wykrywanie rozpoczęcia gestu oraz zbieranie kolejnych próbek, po jego wykryciu

4.3. Model sieci neuronowej – system pojemnościowy

Architektura sieci neuronowej bazuje na koncepcji „Inception Time” [58]. Jest to kategoria splotowych sieci neuronowych (Convolutional Neural Network - CNN) zaprojektowanych pod kątem klasyfikacji przebiegów czasowych (Time Series Classification - TSC). Liczba zastosowań sieci neuronowych wykorzystujących przebiegi czasowe znacząco wzrosła w ostatnich latach. Dla porównania algorytmy przetwarzające obrazy wykorzystują sieci neuronowe od dłuższego czasu. Jest to widoczne w szczególności w obszarach takich jak ochrona zdrowia [59], bezpieczeństwo publiczne [60], monitorowanie i wykrywanie aktywności ludzkiej [61] oraz teledetekcja [62].

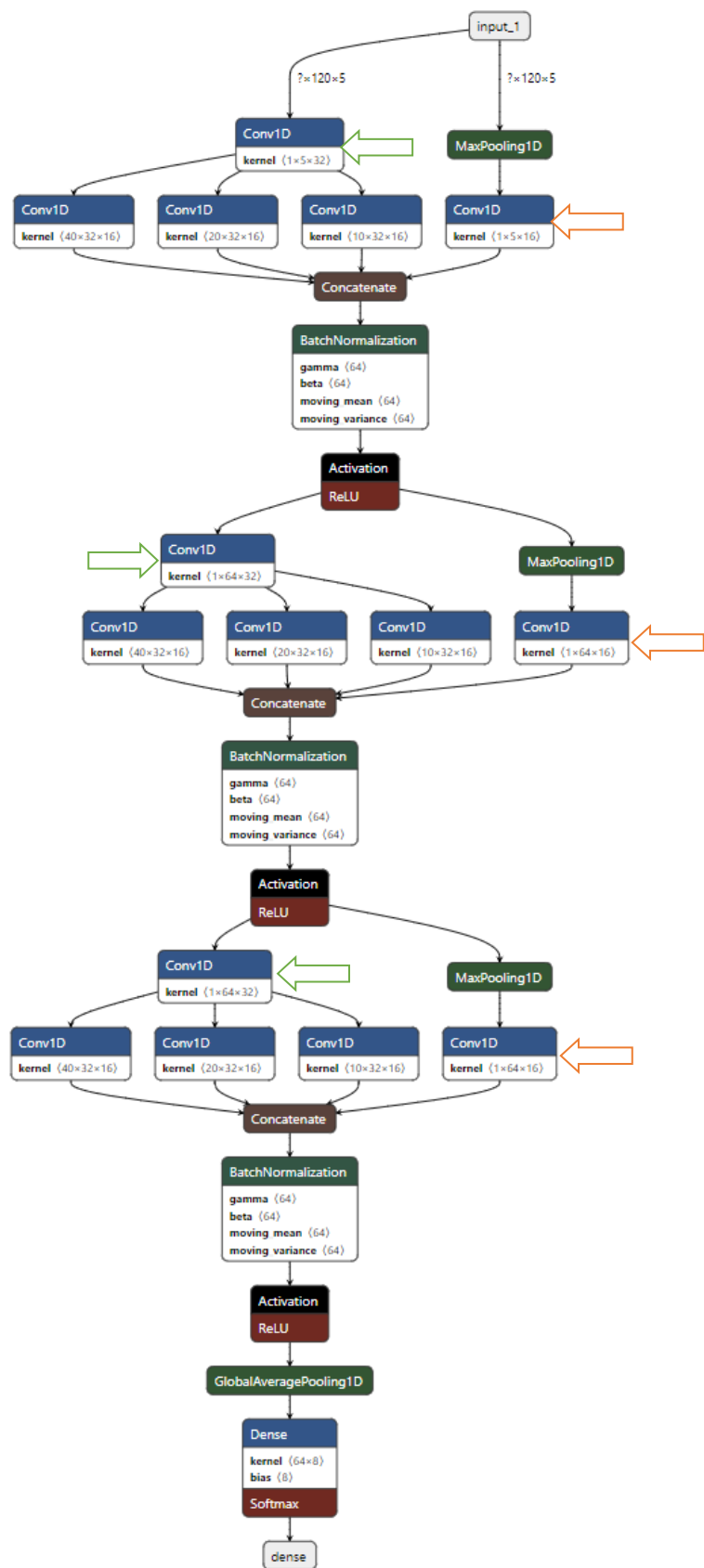
Na rysunku 29 przedstawiono schemat zastosowanej sieci neuronowej wyświetlony w programie Netron. Wektor wejściowy składa się z macierzy 120x5 odpowiadającej pięciu przebiegom czasowym dla każdej z elektrod oraz 120 próbkom na osi czasu. Sieć składa się z trzech warstw. Warstwa druga i trzecia są identyczne, natomiast warstwa pierwsza różni się rozmiarem dwóch komponentów składowych (zaznaczono na schemacie strzałkami w kolorze zielonym i pomarańczowym). Jako funkcje aktywacji dla każdej z warstw wykorzystano funkcję Rektyfikowanej Jednostki Liniowej (Rectified Linear Unit – ReLu) – rysunek 28.



Rysunek 28. Funkcja ReLu oraz jej graficzna reprezentacja

Wykorzystanie jednostki ReLU zwiększa wydajność obliczeniową oraz konwergencję wykorzystanej sieci, ponieważ jest ona oparta na uczeniu pewniejszych cech dzięki rzadszej aktywacji jednostek (neuronów) [63]. Jako funkcję aktywacji na wyjściu zastosowano Global Average Pooling oraz Softmax, jest to powszechne podejście w przypadku ostatnich warstw sieci neuronowych [64]. Głównym zadaniem wykorzystania wspomnianych obiektów jest uzyskanie liczby wyjść z modelu, która odpowiada ilości rozpoznawanych klas. Podejście to

zmniejsza liczbę połączeń między ostatnią warstwą a wektorem wyjściowym przez co podejście całościowo jest bardziej spójne z charakterystyką sieci konwolucyjnych.



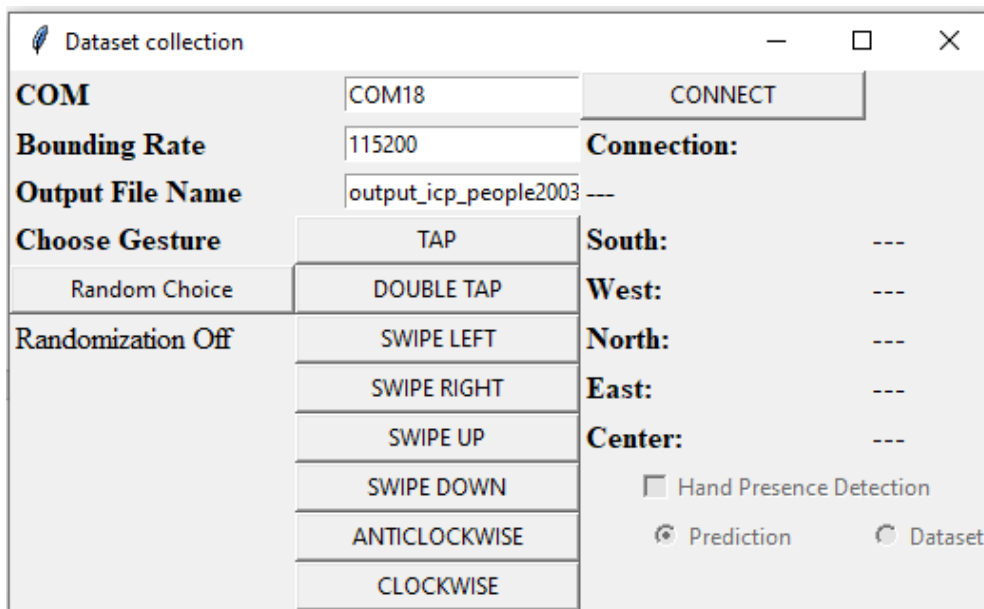
Rysunek 29. Model zastosowanej sieci neuronowej dla rozpoznawania gestów z wykorzystaniem systemu wykorzystującego elektrody pojemnościowe

4.4. Akwizycja Danych

Sposób zbierania danych oraz wynikająca wprost z niego ich jakość, ma kluczowe znaczenie w każdej aplikacji, w której wykorzystujemy algorytmy uczenia maszynowego. Cała idea uczenia maszynowego polega na tym, że sam model jest tylko pewną strukturą połączeń, swego rodzaju szkieletem, który musi być w odpowiedni sposób dostosowany – wyuczony pod konkretne zastosowanie. W tym kontekście odpowiednio przygotowany zestaw danych często ma większe znaczenie w osiągnięciu wysokiej skuteczności klasyfikacji niż świetnie skonstruowany model.

W aplikacji systemu pojemnościowego do zbierania danych wykorzystano pełny system przedstawiony na rysunku 22. W procesie akwizycji danych procesor STM32H745ZI zamiast uruchamiać algorytm sieci neuronowej z danymi wejściowymi z elektrod, przesyłał te zebrane dane przez magistralę UART do aplikacji uruchomionej na komputerze (rysunek 29). Jej głównym celem było przypisanie zebranych sekwencjom danych odpowiednich gestów. Pełna sekwencja czynności wykonywanych przez użytkownika zbierającego dane przedstawia się w następujący sposób:

1. Wybór gestu przez operatora
2. Wykonanie gestu w obszarze roboczym sensora
3. Przesyłanie danych z mikroprocesora do aplikacji
4. Zapis danych do pliku z oznaczeniem uprzednio wybranego gestu



Rysunek 29. Interfejs graficzny aplikacji przeznaczonej do zbierania danych

4.5. Rozszerzanie zbioru danych

Technika rozszerzania zestawu danych uczących oraz testowych (ang. Data Set Augmentation) jest często wykorzystywane przy pracach z sieciami neuronowymi i ma na celu zwiększyć różnorodność danych uczących. Wykorzystywane są różne metody polegające na przykład na zaburzaniu danych wejściowych poprzez celowe ich zniekształcenie [65]. W przypadku rozpoznawania obrazów stosowane są m.in. filtry rozmazujące obraz. Dla sygnałów wprowadza się sztuczne zakłócenia, powodujące większą odporność nauczonego modelu na zakłócenia w danych wejściowych. Innym podejściem jest przekształcenie danych zmieniające ich charakter w sposób całościowy, np. poprzez odbicie lustrzane lub obrót w przypadku aplikacji przetwarzających obrazy [66]. Podczas pracy nad opisywanym systemem, wykorzystano również kilka sposobów na rozszerzenie zestawów danych:

1. Zastąpienie n początkowych lub końcowych próbek sygnału zerami.
2. Zastąpienie każdej k-tej próbki, wartością średniej kwadratowej dla 10 kolejnych próbek włączając k-tą próbkę – wyrażenie (4) (RMS). Dla ostatnich 9 próbek uśrednianie pominięto – ostatnie próbki przebiegu przyjmują wartości około 0.

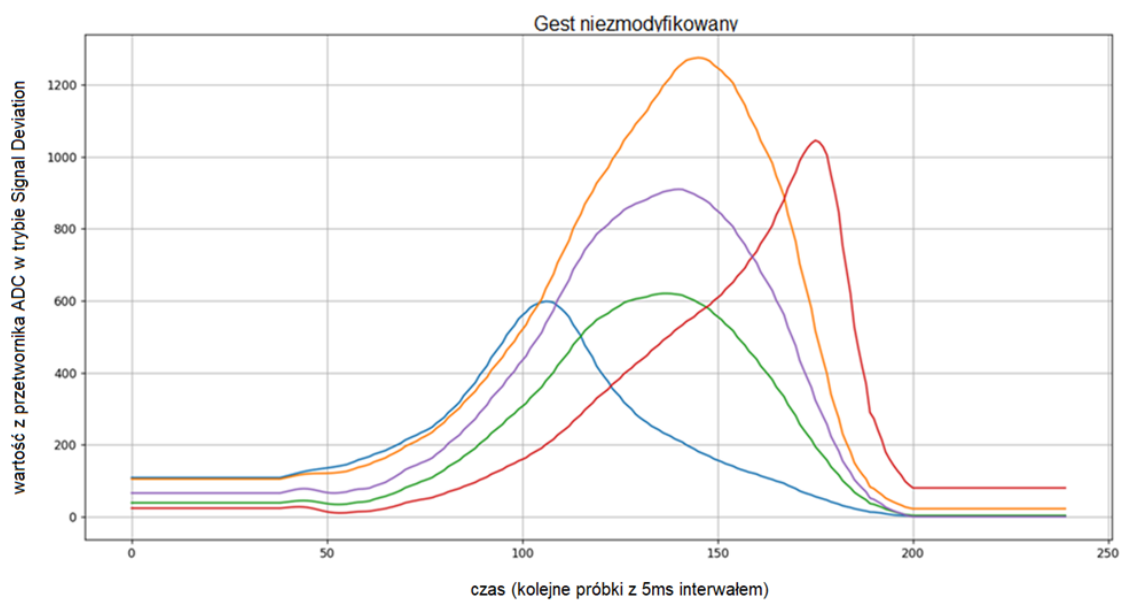
$$p_k = \sqrt{\frac{p_k^2 + p_{k+1}^2 + \dots + p_{k+9}^2}{10}} \quad (4)$$

p_k – stanowi kolejną próbkę

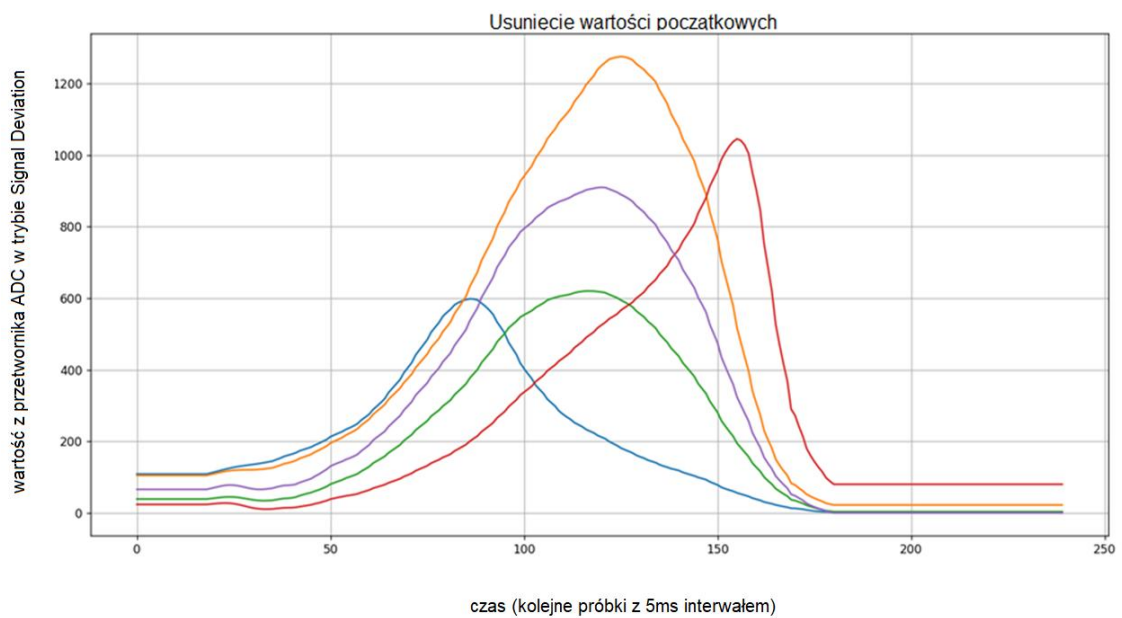
3. Dodanie zakłóceń w postaci addytywnego, szerokopasmowego szumu.
4. Usunięcie n końcowych próbek a następnie przeskalowanie sygnału do rozmiaru początkowego próbki.

Na poniższych rysunkach 30-35 przedstawiono efekty zastosowania wyżej wymienionych sposobów rozszerzenia danych, dla próbki odpowiadającej gestowi „SWIPE LEFT” (przesunięcie dłonią w lewo). Operacje wykonywano na danych przed przeprowadzeniem ich normalizacji. Poszczególne kolory odpowiadają odpowiednim elektrodom z rysunku 24 w następujący sposób:

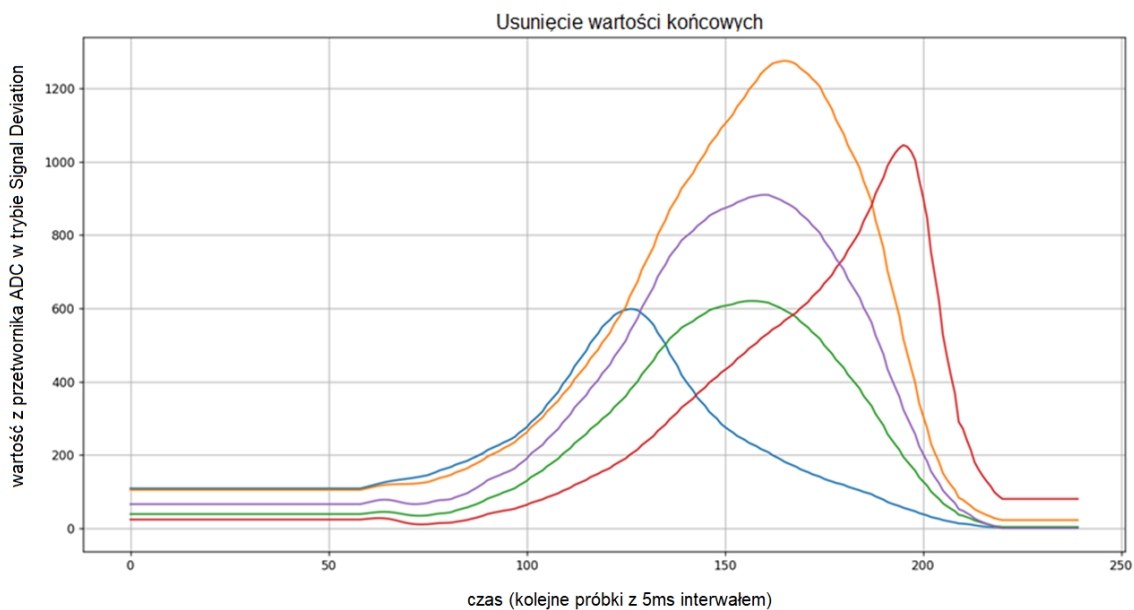
- Kolor niebieski – elektroda prawa
- Kolor czerwony – elektroda lewa
- Kolor zielony – elektroda górna
- Kolor fioletowy – elektroda dolna
- Kolor żółty – elektroda środkowa



Rysunek 30. Przykładowy przebieg sygnałów dla gestu „SWIPE LEFT”

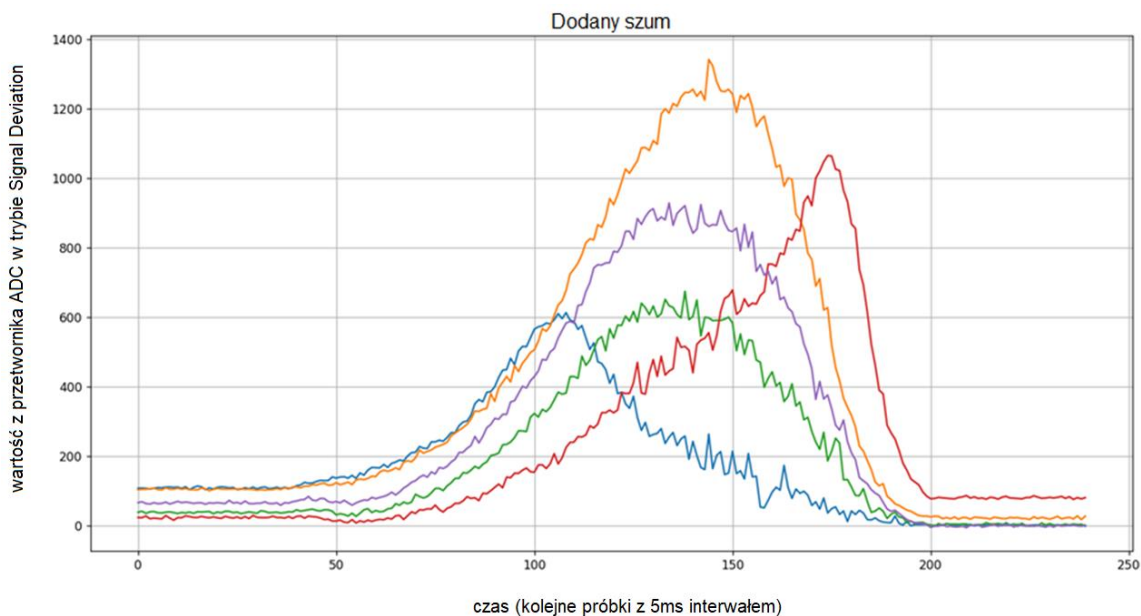


Rysunek 31. Przebieg z usuniętymi 20 próbkami początkowymi oraz dodanymi 20 końcowymi



Rysunek 32. Przebieg z usuniętymi 20 próbkami końcowymi oraz dodanymi 20 początkowymi

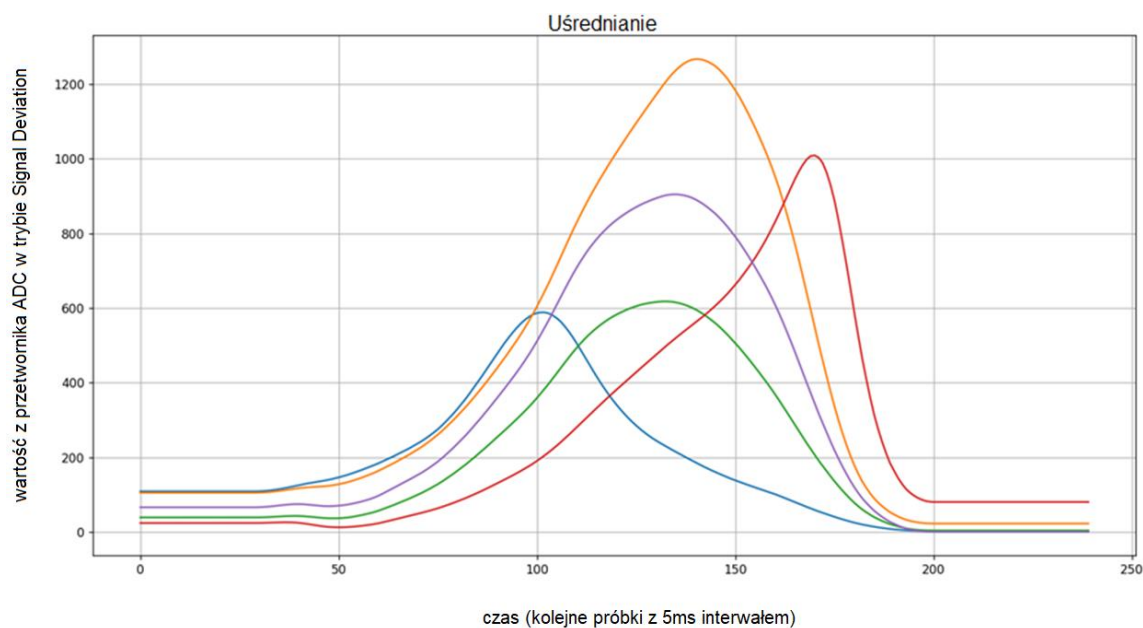
Sztuczne przesunięcie przebiegów ma na celu uodpornić model na trudności z wykrywaniem rozpoczęcia oraz zakończenia gestu.



Rysunek 33. Przebieg z dodanymi sztucznie wartościami odchylenia standardowego losowego szumu o wartości 0.03

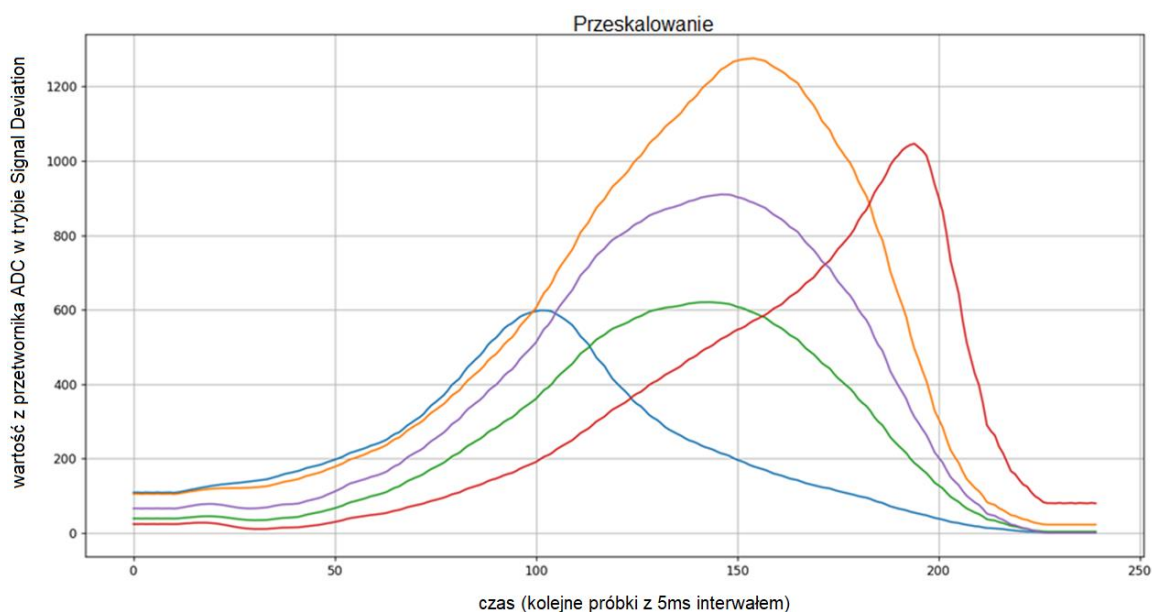
Dodanie szumu losowego miało za zadanie przygotować model do wykrywania przebiegów w warunkach, gdy zawierają one istotny komponent szumu. W opisywanym

systemie może się to zdarzyć np., jeśli w pobliżu znajduje się źródło emitujące fale elektromagnetyczne które mogą być odbierane przez anteny. Zakłócenia pojawiają się również w przypadku obecności obiektów wykonanych z materiałów przewodzących w obszarze roboczym, co musi zostać uwzględnione w końcowej aplikacji systemu, ze względu na powszechność pierścionków oraz innego rodzaju biżuterii. Na oryginalnym przebiegu (rysunek 30), widoczna jest składowa szumu, jednak o dużo mniejszej amplitudzie. Wykorzystanie do uczenia gestów o większej składowej szumu, ma uodpornić model na najgorsze możliwe warunki.



Rysunek 34. Przebieg po przeprowadzeniu operacji uśredniania na każdej z próbek z wykorzystaniem okna obejmującego 10 poprzednich oraz 10 następných próbek

Wygładzenie przebiegów za pomocą RMS, ma na celu przygotowanie modelu na gesty wykonywane z większą prędkością. Z dokonanych obserwacji wynika, że przebiegi mają większą składową zakłóceń, gdy gest jest wykonywany z niewielką prędkością. W przypadku wykonywania gestów bardzo szybko, przebieg jest gładszy.



Rysunek 35. Przebieg po usunięciu 40 próbek początkowych oraz 40 końcowych, a następnie przeskalowaniu sygnałów

Przeskalowanie przebiegów ma na celu zasymulowanie wykonania gestu z różną prędkością. Rysunek 35 przedstawia symulacje spowolnienia tempa wykonanego gestu. Z uwagi na fakt, że zazwyczaj podczas zbierania danych, gesty wykonywane szybko stanowią większość, zastosowano tylko skalowanie symulujące zmniejszenie tempa wykonywania gestu.

4.6. Proces Uczenia

Do przeprowadzenia procesu uczenia oraz zbudowania modelu zastosowano biblioteki języka python - TensorFlow [67] oraz Keras [68]. Jako funkcję kosztu (straty) wykorzystano „Categorical Cross-Entropy” (CCE, Kategoryczna Entropia Krzyżowa) - wzór (5), która wykorzystuje do wyliczenia kosztu logarytm naturalny [69].

$$CCE = - \sum_{i=1}^{i=N} y_i \log(\hat{y}_i) \quad (5)$$

CCE - Categorical Cross-Entropy loss – koszt, lub inaczej błąd rozkładu prawdopodobieństwa predykcji na klasie wyjściowej w stosunku do oczekiwanej wartości

y_i – wartość oczekiwana dla i-tego wyjścia klasyfikatora, zazwyczaj przyjmuje wartość 1 dla klasy zgodnej z oznaczeniem danych oraz 0 dla pozostałych klas

\hat{y}_i - wartość predykcji dla i-tego wyjścia klasyfikatora, suma wszystkich \hat{y}_i parametrów wynosi 1.

N - liczba klas, które może rozpoznawać model

Funkcja CCE najczęściej jest wykorzystywana w sieciach neuronowych o wielu klasach wyjściowych. Podobną funkcją jest „Sparse Categorical Cross-Entropy” (Rzadka Kategoryczna Entropia Krzyżowa). Główną różnicą jest to, że zwraca ona wartość predykcji dla klasy o największej wartości, przez co tracimy informację o wartościach predykcji dla pozostałych klas. W ten sposób część danych zostaje utracona (np. gdy dwie klasy mają bardzo zbliżone wartości predykcji, może okazać się, że lepszym rozwiązaniem jest oznaczenie gestu jako nieznanego, niż klasyfikacja gestu, który osiągnął najwyższą wartość). Biblioteka TensorFlow zawiera implementację wielu funkcji kosztu. W tabeli 4 zestawiono funkcje kosztu stosowane do uczenia modeli sieci neuronowych.

Tabela 4. Porównanie wybranych funkcji kosztu udostępnionych przez bibliotekę TensorFlow, które mogłyby być wykorzystane jako alternatywa do CCE

Nazwa	Opis
Binary Cross Entropy (Binarna Entropia Krzyżowa)	Podobnie jak w przypadku Categorical Cross Entropy bazuje na logarytmie. Przeznaczona dla sieci których wynikiem klasyfikacji jest wartość prawda lub fałsz
Binary Focal Cross Entropy (Ogniskująca Binarna Entropia Krzyżowa)	W odróżnieniu do pozostałych z kategorii „Cross Entropy”, zawiera parametr uwzględniający pewność modelu. Największa różnica jest zauważalna, gdy między wartościami predykcji dla poszczególnych klas występują duże różnice. W takim przypadku algorytm obniża wagi dla najpewniejszych klas, przez co łatwiej wykryć te z które uzyskują mniejsze wartości
Sparse Categorical Crossentropy (Rzadka Kategoryczna Entropia Krzyżowa)	Tak samo jak dla Categorical Cross Entropy, zamiast wektora z wartościami predykcji dla każdej klasy, otrzymujemy najbardziej prawdopodobną klasę
KL-Divergence (Dywergencja Kullbacka-Leiblera)	Metoda polega na określeniu dywergencji między rozkładem prawdopodobieństwa na klasach wyjściowych oraz oczekiwanych. Metoda ta jest bardziej użyteczna, kiedy interesują nas stany pośrednie, czyli np. predykcje będące stanami pośrednimi między klasami.

Biorąc pod uwagę właściwości opisanych funkcji jako najbardziej adekwatną dla rozważanej aplikacji przyjęto CCE.

Zastosowany algorytm optymalizacji modelu – Adam (Adaptive Moment Estimation - Adaptacyjna Estymacja Momentu) jest rozszerzeniem metody najszybszego spadku (Gradient Descent) [70]. Algorytm opisano poniższymi wzorami (6-11). Poniższe wyrażenia odpowiadają za jeden cykl algorytmu. Algorytm jest powtarzany do osiągnięcia odpowiedniego poziomu zbieżności lub do założonej wartości iteracji.

$$g_t = \nabla_{\theta} f_t(\theta_{t-1}) \quad (6)$$

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1) g_t \quad (7)$$

$$v_t = \beta_2 v_{t-1} + (1 - \beta_2) g_t^2 \quad (8)$$

$$\widehat{m}_t = \frac{m_t}{(1 - \beta_1^t)} \quad (9)$$

$$\widehat{v}_t = \frac{v_t}{(1 - \beta_2^t)} \quad (10)$$

$$\theta_t = \theta_{t-1} - \alpha \frac{\widehat{m}_t}{(\sqrt{\widehat{v}_t} + \epsilon)} \quad (11)$$

θ_t, θ_{t-1} – oznaczają parametry modelu obecnej i poprzedniej iteracji

t – obecna iteracja algorytmu

g_t – wartość gradientu dla parametrów modelu z poprzedniej iteracji

m_t – wartość eksponentialnej średniej kroczącej gradientu

v_t – wartość eksponentialnej średniej kroczącej gradientu kwadratowego

\widehat{m}_t – skorygowana wartość eksponentialnej średniej kroczącej gradientu

\widehat{v}_t – skorygowana wartość eksponentialnej średniej kroczącej gradientu kwadratowego

β_1 – parametr kontrolujący wpływ wartości eksponentialnej średniej kroczącej gradientu

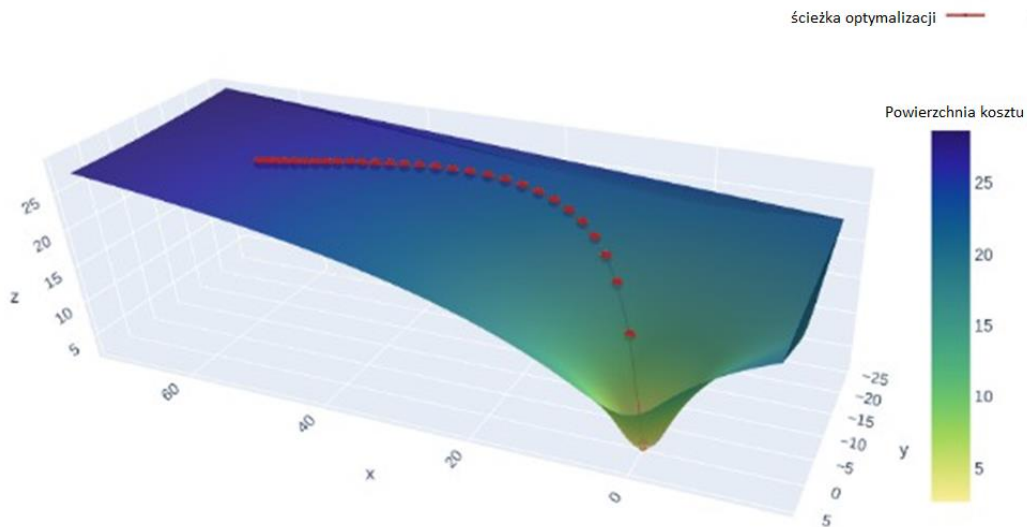
β_2 – parametr kontrolujący wpływ wartości eksponentialnej średniej kroczącej gradientu kwadratowego

α - współczynnik skoku parametrów dla pojedynczej iteracji

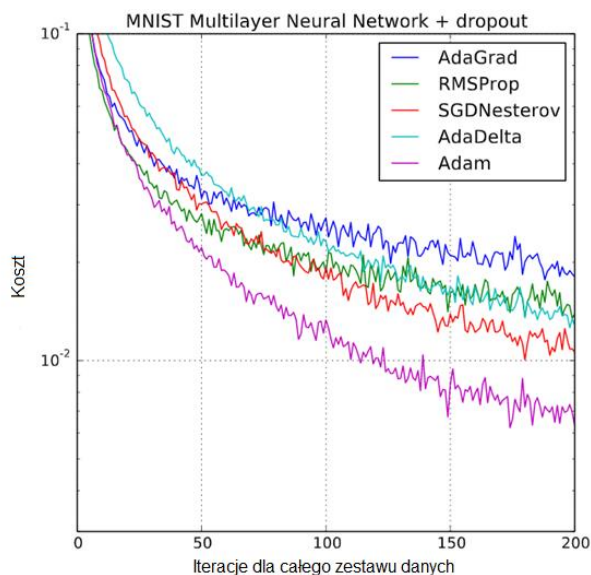
ϵ - współczynnik precyzji

Ostatecznie dla algorytmu optymalizującego Adam, zastosowano następujące parametry: $\alpha = 0.001$, $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon = 10^{-7}$.

Istnieje wiele algorytmów optymalizacji udostępnionych przez bibliotekę Keras [71]. W ogólności celem algorytmu optymalizującego jest znalezienie takich parametrów modelu, aby uzyskać jak najmniejszy błąd. Proces uczenia lub inaczej poszukiwania parametrów o najmniejszym błędzie zobrazowano na rysunku 36. Na rysunku 37 przedstawiono porównanie skuteczności kilku wybranych algorytmów dla wielowarstwowej sieci neuronowej. Widać, że algorytm Adam uzyskuje najlepsze rezultaty dla sieci testowanej w publikacji [54]. Większość wykorzystywanych metod polega na matematycznym wyznaczeniu zależności wartości parametrów od błędu, a następnie poruszaniu się po tej powierzchni poszukując minimalnej wartości za pomocą gradientu.

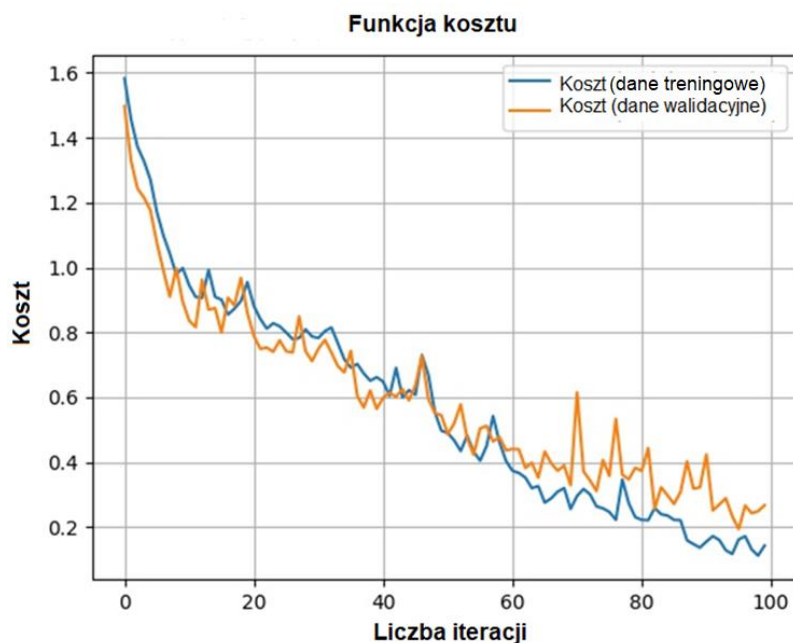


Rysunek 36. Poglądowy sposób działania algorytmów optymalizujących



Rysunek 37. Porównanie uzyskanych kosztów w funkcji liczby iteracji dla wybranych algorytmów optymalizacji [54]

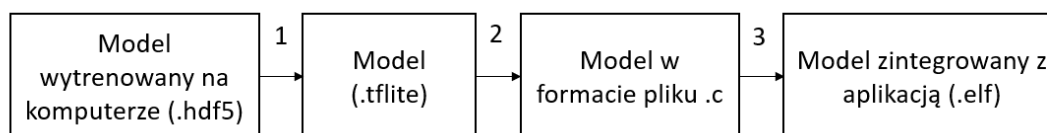
Liczba iteracji procesu uczenia za każdym razem wynosiła 100. W większości przeprowadzonych treningów wartość 50 byłaby wystarczająca, jednak w dwóch przypadkach okazało się, że błąd nie osiągnąłby zadowalających wartości przy przyjęciu takiej liczby operacji. Jako przykład na rysunku 38 przedstawiono wykres wartości błędów dla kolejnych iteracji, podczas uczenia modelu dla danych zebranych w samochodzie.



Rysunek 38. Wykres przedstawiający wartość błędów modelu dla uczenia systemu zainstalowanego w samochodzie

4.7. Konwersja modelu dla potrzeb uruchomienia na mikrokontrolerze

Efektem procesu uczenia sieci neuronowej jest model, którego strukturę przedstawiono na rysunku 22 oraz zbiór parametrów dla połączeń między neuronami. Wszystkie informacje dotyczące modelu zawarte są w pliku wynikowym w formacie .hdf5 [72], który został stworzony do przechowywania i strukturyzacji dużych zestawów danych. W naszym przypadku plik zawiera ponad 130 000 parametrów (warstwa 1: 41552, warstwy 2 i 3: 44288). Wszystkie parametry to czterobajtowe wartości zmiennoprzecinkowe (float), co w efekcie daje ponad 520 000 bajtów danych. Cały plik wraz z danymi dotyczącymi struktury modelu oraz rodzaju wykorzystanych warstw zajął 1.5 MB. Aby uruchomić model na mikrokontrolerze wykorzystano bibliotekę TensorFlow [67] oraz X-CUBE-AI [73] udostępnione wraz ze środowiskiem programistycznym przeznaczonym dla mikrokontrolerów STM32 – STM32CubeMX. Na rysunku 39, przedstawiono schemat postępowania w celu zbudowania pliku wykonywalnego dla mikrokontrolera zawierającego model sieci neuronowej.



Rysunek 39. Proces konwersji modelu do postaci możliwej do wykorzystania przez program mikrokontrolera

W pierwszym kroku należy dokonać konwersji modelu z formatu .hdf5 do .tflite za pomocą biblioteki TensorFlow. Główną różnicą między formatami .hdf5 oraz .tflite jest typ danych parametrów modelu. W pierwszym przypadku wszystkie parametry zapisane są w formacie 32 bitowych zmiennych typu float. W formacie .tflite parametry przyjmują postać wartości 8 bitowych. W drugim kroku, dokonano konwersji modelu do formatu plików c. Wykorzystano do tego wspomniane narzędzie X-CUBE-AI. Na rysunku 40 przedstawiono początkowy fragment pliku c zawierającego tablicę z parametrami modelu, której rozmiar wynosi ponad 460 000 bajtów.

```

main.c network_data.c ×
1 #include "network_data.h"
2
3 ai_handle ai_network_data_weights_get(void)
4 {
5
6     AI_ALIGNED(4)
7     static const ai_u8 s_network_weights[ 460384 ] = {
8         0xbe, 0xe6, 0x95, 0xbd, 0x30, 0xf5, 0x43, 0x3e, 0x03, 0x37,
9         0xea, 0x3d, 0xa0, 0xc5, 0x73, 0xbe, 0x81, 0xad, 0xbe, 0xbd,
10        0x73, 0xc0, 0x00, 0x3e, 0xa1, 0x99, 0xa1, 0x3e, 0xf5, 0x56,
11        0x9e, 0xbd, 0x45, 0x08, 0x4b, 0xbb, 0x05, 0x6a, 0xbd, 0x3e,
12        0x9f, 0xf1, 0xb1, 0x3e, 0x7f, 0x19, 0x89, 0x3e, 0x0f, 0x9e,
13        0xe5, 0xbb, 0xd5, 0x8b, 0xff, 0xbd, 0x72, 0x30, 0xc8, 0xbe,
14        0xf2, 0xd7, 0x28, 0xbe, 0xb8, 0xa8, 0x00, 0x3e, 0xf0, 0x3d,
15        0x93, 0xbe, 0x2c, 0xae, 0x92, 0x3e, 0x99, 0x36, 0x00, 0xbd,
16        0x4f, 0xcd, 0xe8, 0xbd, 0xf8, 0x1e, 0x5f, 0xbe, 0x3e, 0xa6.

```

Rysunek 40. Tablica zawierająca parametry modelu

Jak można zauważyć, w ostatnim kroku przeprowadzona konwersja zmniejszyła rozmiar pamięci konieczny dla przechowywania parametrów o około 10% (z 520 000 B do 460 000 B). Finalne zasoby pamięci wykorzystane przez program dla mikrokontrolera (plik .elf) przedstawiono na rysunku 41.

Region	Start address	End address	Size	Free	Used	Usage (%)
FLASH	0x08000000	0x08100000	1024 KB	513.63 KB	510.37 KB	49.84%
RAM	0x20000000	0x20020000	128 KB	22.58 KB	105.42 KB	82.36%

Rysunek 41. Zestawienie wykorzystanej pamięci mikrokontrolera

Zastosowany mikrokontroler STM32H745ZI posiada dwa sektory pamięci flash o rozmiarze 1MB. Wykorzystanie tej pamięci w tym przypadku to niecałe 50%, co jest wynikiem znacznie lepszym niż oczekiwane wartości, w szczególności biorąc pod uwagę rozmiar pliku wejściowego .hdf5. Zauważalne jest również stosunkowo wysokie wykorzystanie pamięci RAM, co jednak nie jest zaskakujące ze względu na ogromną liczbę operacji wynikających z wykonania pojedynczej klasyfikacji z wykorzystaniem stosunkowo rozległego modelu.

Cały proces uruchomienia systemu na mikrokontrolerze, związany jest z założeniem wdrożenia. Firma patronująca doktoratowi jest producentem mechatronicznych systemów interfejsu człowiek-samochód. Biorąc pod uwagę uwarunkowania bieżących produktów oferowanych przez firmę Merit, takich jak zasoby obliczeniowe wykorzystywanych mikrokontrolerów, a także parametry kosztowe potencjalnego rozwiązania, konieczne było wykorzystanie mikrokontrolera o zasobach umożliwiających spełnienie wspomnianych warunków.

4.8. Testy i implementacja systemu w samochodzie

Z uwagi na wdrożeniowy charakter pracy doktorskiej, następnym krokiem po udanym uruchomieniu systemu w warunkach biurowo-laboratoryjnych, było zamontowanie go w samochodzie firmowym marki – Citroen Berlingo. W tym celu zmodyfikowano obszar podłokietnika samochodu. Z uwagi na ograniczenia systemu infotainment (informacji oraz rozrywki) wykorzystanego samochodu, do połączenia z głównym ekranem sterującym konieczne było zastosowanie smartfonu. Montaż oraz podłączenie z systemem informacyjno-rozrywkowym samochodu przedstawiono na rysunku 42. Jediną modyfikacją systemu przedstawionego na rysunku 15, jest zamiana komputera na smartfon, na który napisano aplikację konwertującą informacje o gestach na konkretne akcje w systemie samochodu.



Rysunek 42. Implementacja systemu w samochodzie. Strzałką oznaczono lokalizację zabudowanej elektrody

Ze względu za sposób działania systemu – elektrody pojemnościowe – przebiegi sygnałów różniły się nieco w stosunku do zebranych w warunkach biurowych. Z tego względu skuteczność wykrywanych gestów spadła o około 10%. Dlatego konieczne było ponowne

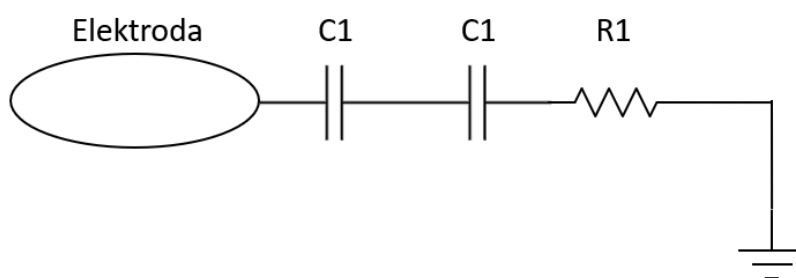
zebranie gestów w warunkach docelowych samochodu. Jak już wcześniej wspomniano, bardzo ważnym aspektem był fakt zapewnienia odpowiedniego potencjału masy. Dla przykładu, jeśli system był zasilany z baterii laptopa, czyli de facto w odizolowaniu elektrycznym od samochodu, otrzymywane przebiegi w ogóle nie przypominały tych które uzyskiwano w biurze. Po zasileniu systemu z układu zasilania samochodu, sytuacja znacznie poprawiła się, jednak obserwowano więcej szumów, a sygnał był bardziej niejednorodny. Powyższe obserwacje stanowiły kolejny czynnik, który wpłynął na decyzję o wykorzystaniu augmentacji danych.

4.9. Wykorzystanie manipulatora do testowania oraz uczenia

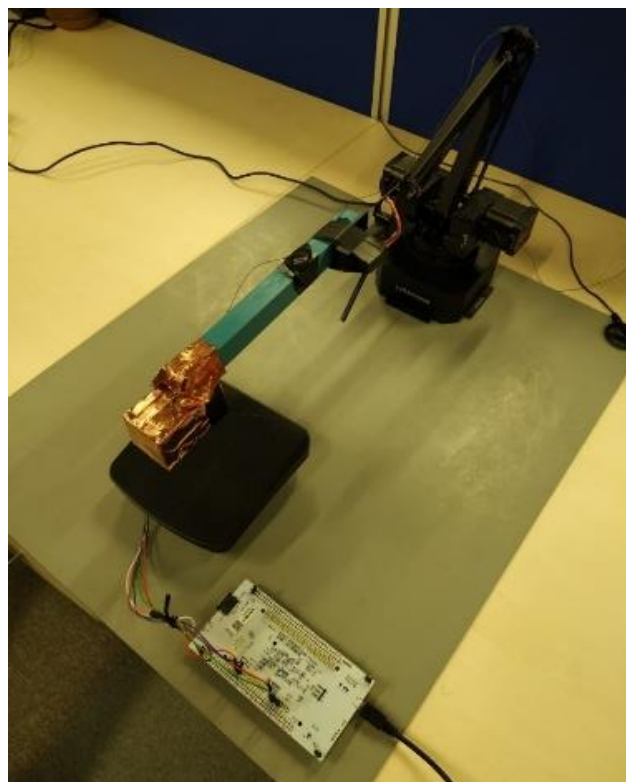
Zagadnienie zbierania danych jest wąskim gardłem w procesach uczenia maszynowego w niemal wszystkich przypadkach [74] [75]. Problem ten jest w szczególności widoczny w obszarze interfejsów człowiek-maszyna (ang. Human Machine Interface – HMI). Proces wymaga poświęcenia dużej ilości zasobów. W tym przypadku nie jest zazwyczaj możliwe wykorzystanie gotowych zestawów danych, tak jak to ma miejsce w przypadku systemów wizyjnych, gdzie istnieje bardzo wiele baz danych z oznaczonymi obrazami, które można wykorzystać w tego typu aplikacjach. Z uwagi na unikalność rozwiązań, dane są zbierane od wielu użytkowników. Naturalną konsekwencją takiego stanu rzeczy jest długi czas potrzebny na zebranie zestawu danych o wystarczającej objętości. Rozwiązaniem tego problemu jest rozszerzanie (augmentacja) danych, poprzez modyfikację pewnych parametrów danych i wykorzystanie do uczenia danych oryginalnych i zmodyfikowanych. Nadal jednak konieczne jest zebranie pokaźnej ilości rekordów w sposób manualny.

Podczas prac z opisywanym systemem pojawił się pomysł zastąpienia ruchów wykonywanych przez użytkowników, ruchami manipulatora. Do realizacji pomysłu wykorzystano czteroosiowy manipulator uArm [76]. Ze względu na fakt, że miałem do czynienia z systemem opartym o elektrody pojemnościowe konieczne było zmodyfikowanie efektora. Na rysunku 43, przedstawiono schemat elektryczny zamontowany na efektorze manipulatora. Układ ma na celu zasymulować impedancję elektryczną między ludzką ręką a potencjałem uziemienia. Zastosowano rezystor R1 o oporności 1500Ω oraz dwa kondensatory C1 o pojemności 20nF . Rysunek 44 przedstawia stanowisko, które wykorzystano do zbierania gestów z użyciem manipulatora. Obszar dłoni jest zasymulowany blokiem oklejonym miedzianą taśmą, który następnie jest połączony przez układ CR z uziemieniem. W praktyce oznacza to, że układ impedancyjny był podłączony do przewodu PEN najbliższego gniazdka, podobnie jak

widoczna na zdjęciu mata ESD. Jest to istotne ze względu na fakt, że fizyczną wartością, która jest mierzona, a także jest źródłem wszystkich informacji jest pojemność między elektrodą a obszarem o potencjale odniesienia – potencjale zerowym. O tym jak bardzo jest to istotne przekonałem się podczas opisanych wcześniej testów systemu w samochodzie. W przypadku braku podłączenia systemu z masą samochodu (zasilanie z laptopa pracującego na baterii), sygnały z elektrod zawierały dużo zakłóceń. Sytuacja diametralnie poprawiała się po zasileniu systemu z gniazda 12V samochodu (z wykorzystaniem przetwornicy napięcia). Wtedy system działał z praktycznie taką samą skutecznością jak podczas testów wykonywanych w warunkach laboratoryjnych.



Rysunek 43. Układ elektroniczny symulujący impedancję ludzkiej ręki



Rysunek 44. Zdjęcie poglądowe stanowiska do zbierania gestów z wykorzystaniem manipulatora

Pierwsze testy z manipulatorem potwierdziły założenie, że ruch przystosowanego efektora nad obszarem elektrody pozwala na uzyskanie zbliżonych sygnałów wejściowych tak jak w przypadku wykonywania gestów przez człowieka. Zaproponowane podejście pozwalało uzyskać zadowalające rezultaty na poziomie pojedynczego gestu, jednak zastosowanie uczenia maszynowego implikuje konieczność zastosowania większej ilości danych uczących. Wysoka powtarzalność manipulatora jest w tym wypadku niepożądana, ponieważ uzyskanie zestawu identycznych danych uczących spowoduje, że model sieci neuronowej będzie przeuczony (overfitting) – zjawisko to ma miejsce zazwyczaj, gdy zestaw danych uczących jest zbyt mały oraz gdy zestaw danych niewystarczająco odzwierciedla warunki rzeczywiste. W celu uniknięcia tego problemu zarówno trajektorie oraz prędkości poszczególnych ruchów manipulatora, zostały zmodyfikowane z wykorzystaniem generatora liczb losowych. Przed wykonaniem ruchu trajektoria bazowa została poddana randomizacji. Polegało to na zmodyfikowaniu współrzędnych w zakresie +/-5% od współrzędnych referencyjnych. Podobnie postąpiono z zadaną prędkością. Do generacji liczb losowych wykorzystano generator „Mersenne Twister” o okresie 2^{19937} co zagwarantowało niepowtarzalność wariacji współrzędnych [77].

W opisany powyżej sposób zebrano 8780 gestów, które stanowiły dane wykorzystane do uczenia modelu sieci neuronowej. Dane zostały podzielone w losowo na dwa podzbiory treningowy oraz walidacyjny, w stosunku 80 -20. Część testowa została pominięta ze względu na fakt, że do testów zaproponowanej koncepcji wykorzystano gesty wykonywane przez użytkowników. Ostatecznie uzyskano skuteczność 80%. Rezultat ten pokazuje, iż akwizycja danych z wykorzystaniem manipulatora jest efektywna i może być wykorzystywana w praktyce. Otwartym pytaniem jest czy może stanowić wyłączone źródło danych, czy jest konieczne zebranie dodatkowych danych w inny sposób.

W poniższej tabeli 5, przedstawiono wyniki testów algorytmu nauczonego z wykorzystaniem danych zebranych wyłącznie przy użyciu manipulatora.

Tabela 5. Wyniki klasyfikacji gestów użytkowników dla modelu nauczonego z wykorzystaniem danych z manipulatora

	← Przesunięcie w Lewo	→ Przesunięcie w Prawo	↑ Przesunięcie w górę	↓ Przesunięcie w dół	Zbliżenie i oddalenie	Suma
Użytkownik 1	100	100	80	100	100	96
Użytkownik 2	90	80	90	90	90	88
Użytkownik 3	90	100	50	100	100	88
Użytkownik 4	100	90	70	70	80	82
Użytkownik 5	100	100	40	50	100	78
Użytkownik 6	100	80	100	60	50	78
Użytkownik 7	100	90	60	40	70	72
Użytkownik 8	90	90	0	80	100	72
Użytkownik 9	90	80	50	100	20	68
Użytkownik 10	100	100	60	70	70	80
Suma	96	91	60	76	78	80.2

Ponadto zbieranie gestów w opisany sposób jest około 3-4 krotnie szybsze niż podczas manualnego podejścia. Zebranie 1173 gestów z wykorzystaniem manipulatora zabrało 2 godziny i 45 minut, podczas gdy zbieranie 1173 gestów metoda tradycyjną – 6 godziny i 45 minut. W efekcie w metodzie automatycznej uzyskano 427 gestów/h, a w manualnej 123 gestów/h. W przedstawionych danych uwzględniono tylko czas spędzony przez użytkowników przy stanowisku testowym. W przypadku większej ilości zbieranych gestów różnica byłaby jeszcze większa ze względu na konieczność przerw wynikających z kilkudniowego zbierania danych oraz na fakt, że manipulator może pracować 24h na dobę [78].

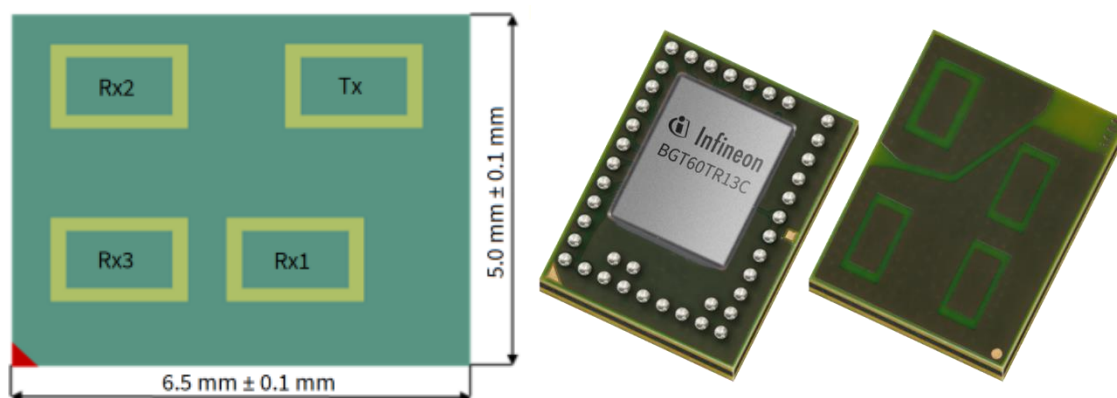
Dodatkowo, zaletą podejścia automatycznego jest to, że nie występują błędy oznaczenia próbek, wynikające z błędów ludzkich spowodowanych powtarzalnością zadania polegającego na ręcznym oznaczaniu danych. Zaprezentowane rozwiązanie może stanowić

również propozycję powtarzalnego systemu testującego systemy do rozpoznawania gestów. Opieranie się na danych zebranych od ludzi zawsze wprowadza pewne odchylenie (ang. bias) parametrów modelu. Przykładowo ten sam system może uzyskać różne wyniki, gdyby był testowany np. na różnych kontynentach. Podobny problem występuje w systemach dotykowych 2D gdzie manipulatory są wykorzystywane do testów od dawna.

Przytoczone wyniki oraz przedstawione wnioski zostały przedstawione w publikacji [78], która powstała podczas pracy nad zagadnieniami związanymi z realizacją doktoratu.

5. System do rozpoznawania gestów oparty o radar

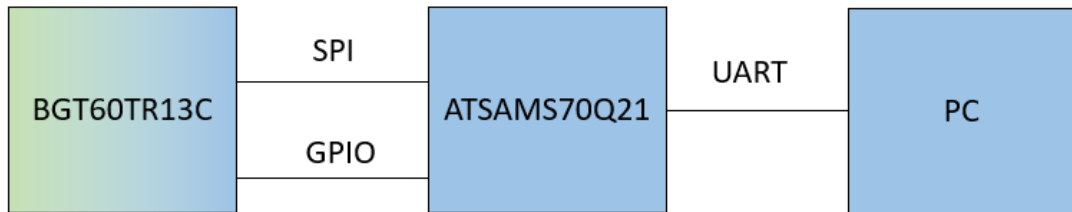
Kolejnym urządzeniem, nad którym podjęto prace w ramach niniejszego doktoratu był system wykorzystujący radar. Zastosowano wcześniej wspomniane rozwiązanie w postaci układu „Antenna in Package” (Antena zintegrowana w obudowie) firmy Infineon w postaci układu BGT60TR13C [29] zawartego na płycie ewaluacyjnej. Koncepcja umieszczenia układu anten na obudowie układu znacząco skróciła czas pracy nad uruchomieniem systemu, ponieważ zupełnie wyeliminowało to konieczność ich zaprojektowania oraz wykonania. Na rysunku 45 przedstawiono układ anten znajdujący się na obudowie układu. Kolejną zaletą wynikającą z umieszczenia anten na obudowie, jest wyraźne odseparowanie obwodu wysokiej częstotliwości radaru od pozostałej części obwodu. Pozwala to na stosowanie popularnych i budżetowych rozwiązań do tworzenia płytek PCB na których umieszczony jest układ. Dodatkowo niewielkie rozmiary (6.5mm x 5.0mm x 0.85 mm) znacząco ułatwiają zastosowanie ich w niemal dowolnej lokalizacji. Ma to duże znaczenie szczególnie w przypadku konieczności rozszerzenia funkcjonalności istniejącego produktu o rozwiązanie wykorzystujące radar. Moduł jest w stanie wykrywać ludzi nawet w odległości do 10m, co powoduje, że może zostać on wykorzystany również jako czujnik obecności osób w pojeździe lub poza nim. W rozważanej aplikacji konfiguracji parametrów działania radaru dobrano w taki sposób, aby wykrywał gesty w odległości do 50 cm.



Rysunek 45. Wymiary układu oraz zdjęcie poglądowe układu BGT60TR13C

Schemat zastosowanego systemu przedstawiono na rysunku 46. Układ składał się z płytki zawierającej układ sterujący radarem BGT60TR13C oraz procesor ATSAM570Q21. Procesor odpowiada za bezpośrednią komunikację z sensorem oraz przesyłanie danych

z kolejnych pomiarów do komputera z wykorzystaniem protokołu UART/USB. W tym przypadku część analogowa obejmowała tylko układ anten oraz część analogową układu BGT60TR13C.



Rysunek 46. Schemat systemu do wykrywania gestów za pomocą radaru. Kolor zielony oznacza część analogową, kolor niebieski – cyfrową. Moduł radaru odpowiada za połączenie części analogowej i cyfrowej

5.1. Moduł BGT60TR13C - Antena in Package

5.1.1. Układ Anten

Anteny odbiorcze zostały rozmieszczone zgodnie z koncepcją ULA (ang. Uniform Linear Array – Jednolitego Układu Linowego) [79] [80], której założeniem jest umożliwienie określenia kierunku, z którego nadchodzą fale radiowe. Radar typu FCMW (ang. Frequency Modulated Continuous Wave Radar – Radar Fali Ciągłej z Modulacją Częstotliwościową) operuje w zakresie częstotliwości od 58.0 do 63.5 GHz [81], zatem zgodnie z założeniami koncepcji ULA (rysunek 47) odległość między antenami powinna maksymalnie stanowić połowę długości fali. Długość fali określa wzór (12). Warunek graniczny dla maksymalnej odległości między antenami zapisano poniżej (13).

$$\lambda = \frac{c}{f} \quad (12)$$

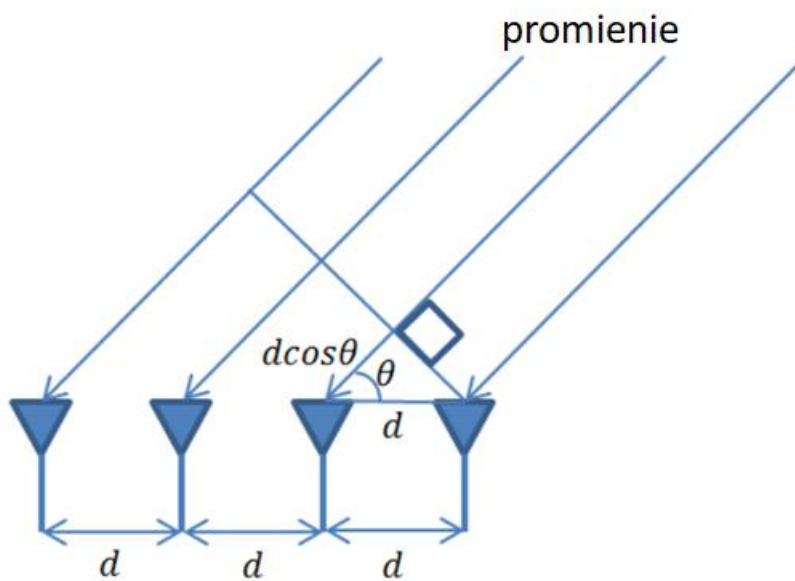
$$d \leq \frac{\lambda}{2} \quad (13)$$

c – prędkość światła równa w przybliżeniu $3 * 10^8 \frac{m}{s}$

f – częstotliwość fali radaru $60 * 10^9 \text{ Hz}$

d – odległość między antenami (rysunek 47)

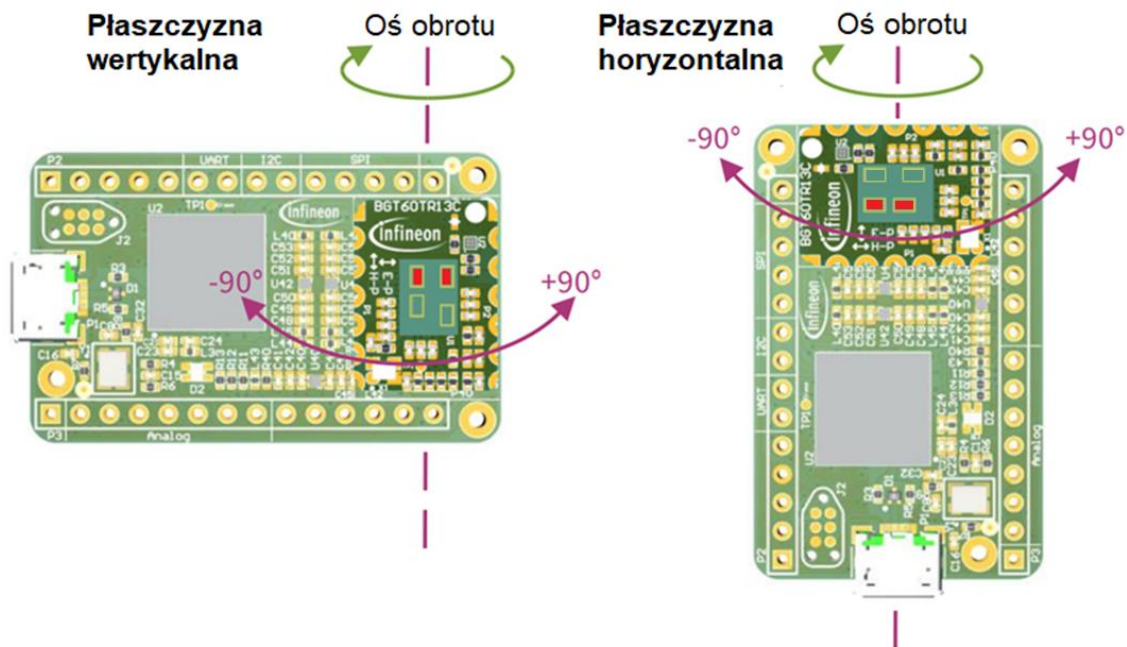
W efekcie otrzymujemy, że dla radaru o częstotliwości pracy 60 GHz maksymalna odległość anten wynosi 2.5 mm (jest to przybliżenie wynikające z przybliżenia wartości prędkości światła). Zmierzone odległości (między osiami anten) to 2.47mm między Rx2 i Rx3 oraz 2.48 mm między Rx3 i Rx1. Należy zwrócić uwagę, że odległość graniczna między antenami dla częstotliwości skrajnych 58 GHz oraz 63.5 GHz wynoszą odpowiednio 2.58 mm oraz 2.36 mm. Podczas przeprowadzonych testów algorytm wykrywania kąta działał poprawnie nawet przy zastosowaniu zakresu częstotliwości 62-63 GHz. W dalszej części rozdziału opisano wykorzystanie właściwości pomiaru kąta jako jeden z parametrów wejściowych dla algorytmu sieci neuronowej.



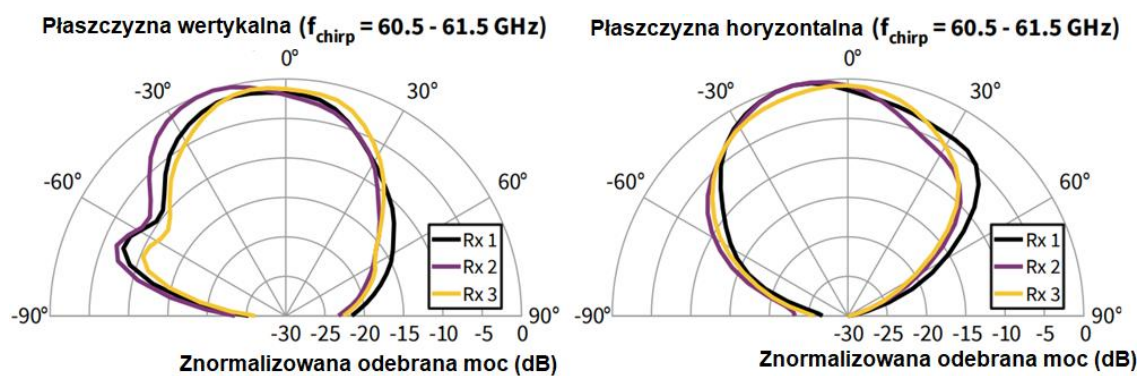
Rysunek 47. Rozmieszczenie anten pozwalające na wyznaczenie kierunku

Zastosowany układ daje możliwość pomiaru kąta w dwóch płaszczyznach. Zależność zakresu wykrywanego kąta w przestrzeni oraz wykorzystywanych w tym celu anten pokazano na rysunku 48. W opisywanej aplikacji zastosowano orientację wertykalną wykorzystującą anteny Rx3 oraz Rx2. Na rysunku 49 pokazano moc otrzymanego sygnału na każdej z anten, podczas testów anteny wykonanych przez producenta [82]. Test przeprowadzono w bezechowej komorze RF, polegał on na nadawaniu sygnału chirp w zakresie częstotliwości od 60.5 do 61.5 GHz w kierunku reflektora oddalonego o 40cm od powierzchni układu, a następnie pomiarowi natężenia odbitego sygnału przez anteny odbiorcze. Przetestowano obie orientacje anteny. Przeprowadzone podczas prac nad doktoratem doświadczenia pokazały, że efektywny obszar wykrywania kąta położenia mocno zależy również od

pozostałych parametrów pracy radaru. Można jednak przyjąć, że maksymalnym możliwym zakresem jest -60° do 60° . Podczas zastosowania konfiguracji dedykowanej rozpoznawaniu gestów (opisano w dalszej części rozdziału) realny zakres wykrywanego kąta to -45° do 45° .



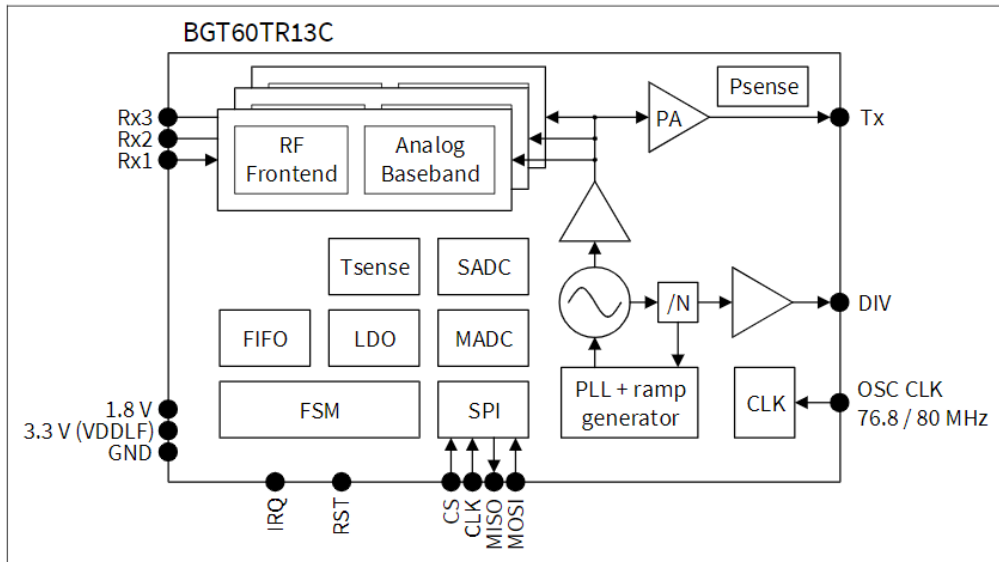
Rysunek 48. Wykorzystanie anten w zależności od płaszczyzny, dla której wyznaczany jest kąt, rysunek z dokumentacji radaru Infineon [82]



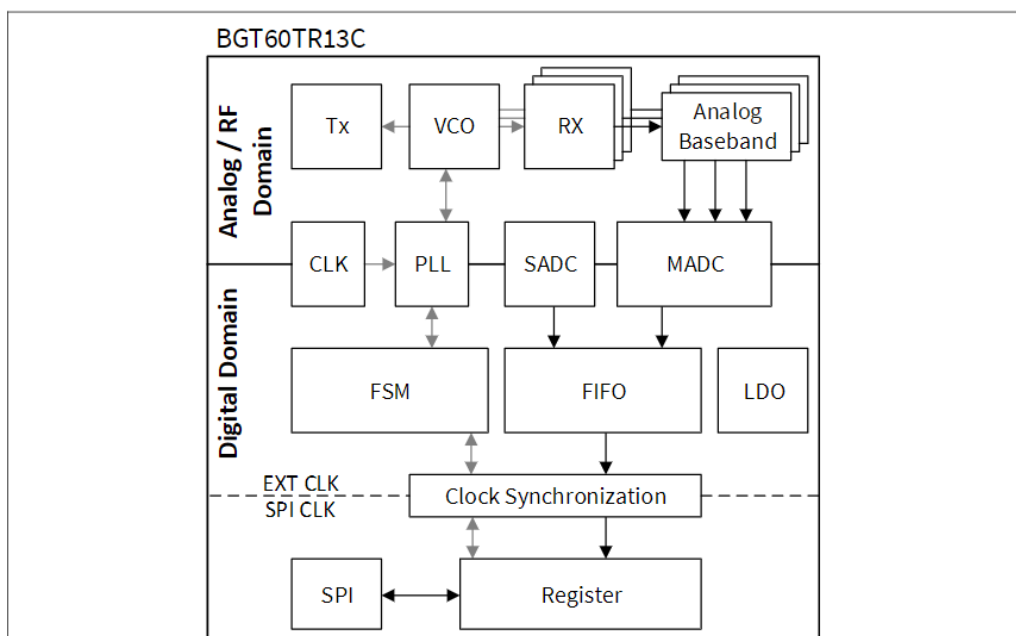
Rysunek 49. Zakres kątowy wykrywania obiektów przez pary anten oraz natężenie odebranego sygnału dla wszystkich trzech anten odbiorczych w obu orientacjach, rysunek z dokumentacji radaru Infineon [82]

5.1.2. Układ radaru

Poza antenami umieszczonymi na obudowie układ zawiera wbudowany układ elektroniczny składający się z modułów realizujących funkcje nadawania, odbierania sygnału a także komunikacji z procesorem. Na rysunkach 50 oraz 51 przedstawiono odpowiednio schemat blokowy układu, oraz schemat funkcjonalny.

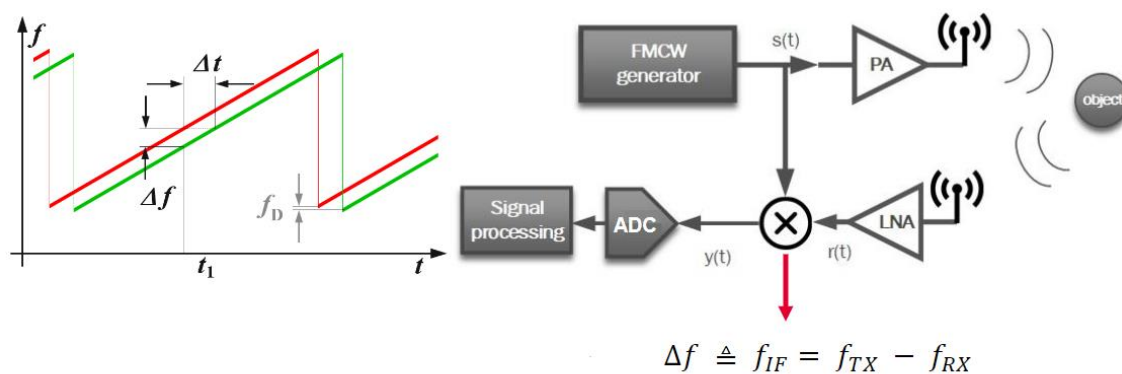


Rysunek 50. Schemat Blokowy Układu BGT60TR13C, za dokumentacją układu [81]



Rysunek 51. Schemat Funkcjonalny Układu BGT60TR13C, za dokumentacją układu [81]

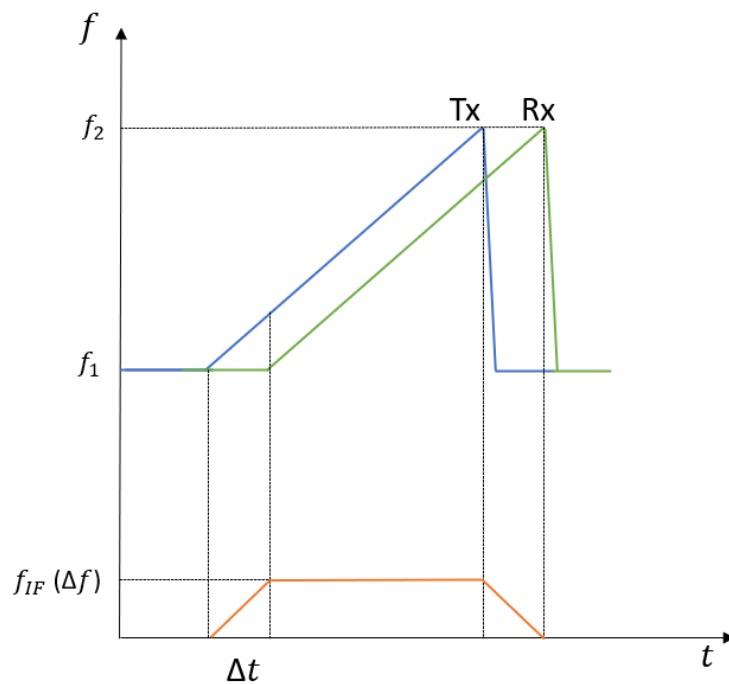
Układ zawiera generator sygnału chirp, z liniową modulacją częstotliwości w zakresie konfigurowalnym w obszarze od 58 GHz do 63,5 GHz. Możliwe jest również określenie liczby sygnałów nadawanych w jednym cyklu skanowania. Nadawany sygnał jest przesyłany do anteny nadawczej, a także do obwodu odbiornika. Obwód odbiornika realizuje demodulację homodynową, która w tym zastosowaniu sprawdza się dobrze dzięki pełnej kontroli nad częstotliwością nośną oraz niewielkiemu wpływowi zakłóceń przy niewielkiej drodze przebywanej przez fale radiowe. Demodulacja ma charakter częstotliwościowy i jej efektem jest sygnał o częstotliwości różnicowej nadajnika i odbiornika. Przebiegi czas-częstotliwość sygnału nadawanego i odbieranego oraz schemat blokowy demodulatora przedstawiono na rysunku 52. Należy zaznaczyć, że w dalszej części używana jest zgodnie z dokumentacją układu termin „częstotliwość pośrednia”, choć w praktyce nie mamy tu do czynienia z klasyczną demodulacją heterodynową. Termin ten w opisywanym kontekście oznacza częstotliwość różnicową, a nazwa wynika z faktu, że demodulacja nie jest ostatnim krokiem analizy.



Rysunek 52. Demodulacja sygnału odbieranego z wykorzystaniem sygnału nadawczego

5.2. Sposób wykrywania odległości i prędkości

Na rysunku 53, pokazano różnicowy charakter przebiegów czasowych otrzymywanych z urządzenia. W wyniku odjęcia sygnału z anteny nadawczej oraz odbiorczej częstotliwość pośrednia (f_{IF}) osiągnie stałą wartość w przypadku odbicia się fal elektromagnetycznych od wykrywanego obiektu. Operację różnicy pokazującą ciągłą wartość sygnału częstotliwości pośredniej w czasie przedstawiono na poniższym wykresie.



Rysunek 53. Operacja różnicy sygnału z anteny nadawczej oraz odbiorczej

Uwzględniając czas pokonany przez falę elektromagnetyczną w przestrzeni oraz czas jej powrotu, odległość wykrytego obiektu może zostać wyznaczona z równania (14).

$$r = \frac{f_{IF} c t_c}{f_2 - f_1} \quad (14)$$

r – odległość między antenami a obiektem

f_{IF} – częstotliwość pośrednia

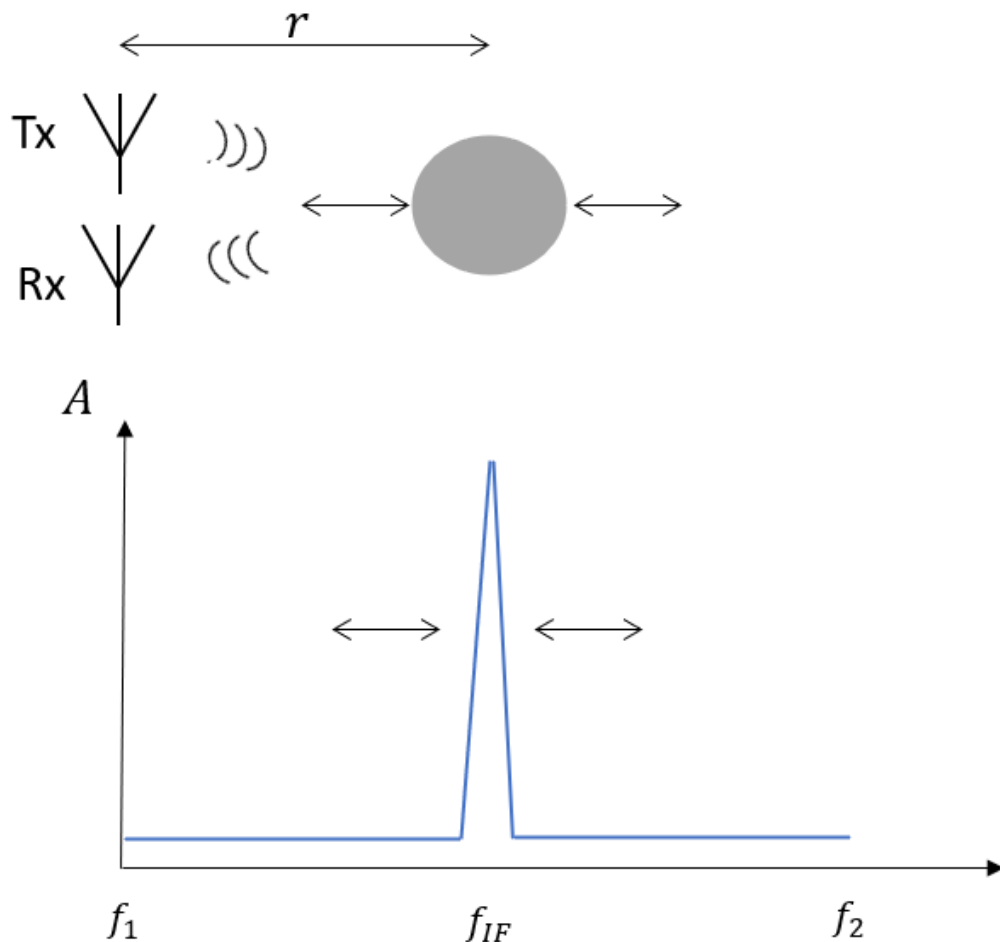
f_1 – częstotliwość początkowa sygnału chirp

f_2 – częstotliwość końcowa sygnału chirp

t_c – czas trwania jednego cyklu sygnału chirp

c – prędkość światła równa w przybliżeniu $3 * 10^8 \frac{m}{s}$

Parametry f_2 , f_1 , t_c mają wartości stałe i wynikają z konfiguracji radaru, zatem odległość jest proporcjonalna do uzyskanej częstotliwości pośredniej. Dzięki temu po wykonaniu szybkiej transformaty Fouriera dla otrzymanego sygnału częstotliwości pośredniej otrzymamy znacznie wyższą amplitudę dla częstotliwości odpowiadającej odległości obiektu od anteny. Wyższa częstotliwość odpowiada większej odległości. W efekcie wraz z oddalaniem się obiektu pik na wykresie amplitudowo-częstotliwościowym również będzie się przemieszczał. Rysunek 54 przedstawia zależność odległości wykrywanego obiektu oraz częstotliwości dominującej na wykresie amplitudowo-częstotliwościowym.



Rysunek 54. Korelacja między przemieszczaniem się obiektu a uzyskiwaną częstotliwością przenoszenia na wykresie amplitudowo-częstotliwościowym

Drugą wielkością, którą można uzyskać na podstawie przebiegów radaru jest prędkość przemieszczania się obiektu. Teoretycznie byłoby możliwe porównywanie częstotliwości otrzymywanych z kolejnych przebiegów chirp analogicznie jak w przypadku pomiaru odległości, jednak zmiana odległości skutkuje stosunkowo niewielką zmianą częstotliwości. Zamiast tego

wykorzystano przesunięcie fazowe między sygnałami otrzymanym dla dwóch kolejnych cykli sygnału chirp. Prędkość obiektu jest wyznaczana na podstawie równania (15).

$$v = \frac{\Delta\varphi \lambda}{4\pi t_c} \quad (15)$$

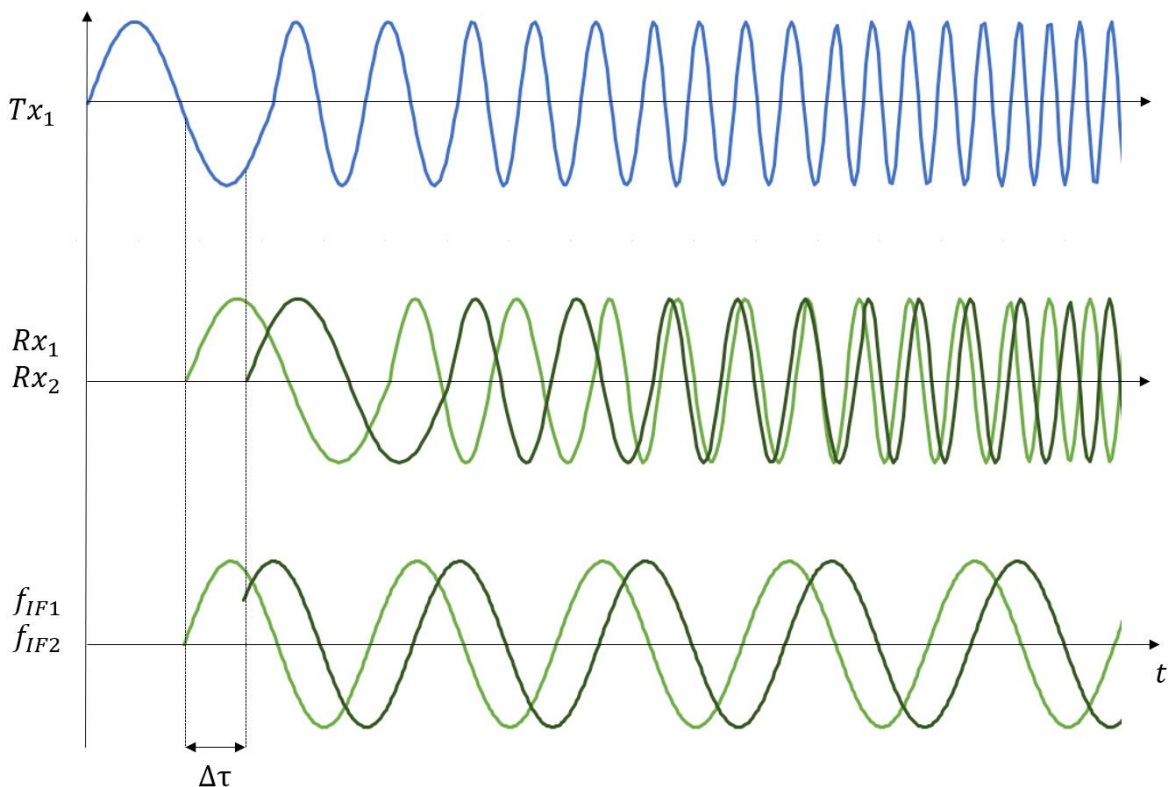
v – prędkość obiektu

$\Delta\varphi$ – przesunięcie fazowe między dwoma kolejnymi odpowiedziami na sygnał chirp

λ – długość fali

t_c – czas trwania jednego cyklu sygnału chirp

Ponownie można zauważyć, że parametry λ oraz t_c wynikają ze stałych parametrów konfiguracyjnych radaru, stąd wynika, że prędkość obiektu jest proporcjonalna do przesunięcia fazowego sygnału z kolejnych chirpów. Na rysunku 55 przedstawiono odpowiednio sygnał nadawczy, sygnały po odbiciu od ruchomego obiektu dla dwóch kolejnych przebiegów chirp oraz uzyskaną dla tych sygnałów częstotliwość przenoszenia.



Rysunek 55. Zmiana fazy sygnału dla dwóch kolejnych sygnałów chirp, w przypadku poruszającego się obiektu

W celu wyznaczenia przesunięcia fazowego, należy skorzystać z mierzonej wartości $\Delta\tau$ będącej różnicą otrzymanych sygnałów dla kolejnych przebiegów, wynikającą z przemieszczenia obiektu. Różnica może być wyznaczona zgodnie z równaniem (16).

$$\Delta\tau = \frac{2 \Delta r}{c} \quad (16)$$

$$\Delta\varphi = 2\pi f_c \Delta\tau = \frac{4\pi \Delta r}{\lambda} \quad (17)$$

$\Delta\tau$ – różnica w czasie między rozpoczęciem odbioru dwóch kolejnych cykli sygnału chirp

Δr – różnica odległości zmierzonej dla dwóch kolejnych cykli sygnału chirp

c – prędkość światła równa w przybliżeniu $3 * 10^8 \frac{m}{s}$

$\Delta\varphi$ – przesunięcie fazowe między dwoma kolejnymi odpowiedziami na sygnał chirp

f_c – częstotliwość przenoszenia

λ – długość fali

Przykładowo dla przemieszczenia 0.5 mm uzyskujemy ze wzoru (17) wartość 72° która jest łatwo mierzalna i jest wartością kilkudziesięciokrotnie przekraczającą dokładność pomiarową. W przypadku pomiaru zmiany częstotliwości można zastosować wzór (18). Ponownie dla wartości przemieszczenia 0.5 mm uzyskamy zmianę częstotliwości około 365 Hz, która stanowi zaledwie poniżej 0.5% wartości częstotliwości f_{IF} . Powyższe fakty w jasny sposób uzasadniają wykorzystanie przesunięcia fazowego do wyznaczania prędkości obiektu.

$$\Delta f_{IF} = \frac{(f_2 - f_1) 2 \Delta r}{t_c c} \quad (18)$$

Δf_{IF} – różnica częstotliwości pośredniej dla dwóch kolejnych cykli sygnału chirp

f_1 – częstotliwość początkowa sygnału chirp

f_2 – częstotliwość końcowa sygnału chirp

Δr – różnica odległości zmierzonej dla dwóch kolejnych cykli sygnału chirp

t_c – czas trwania jednego cyklu sygnału chirp

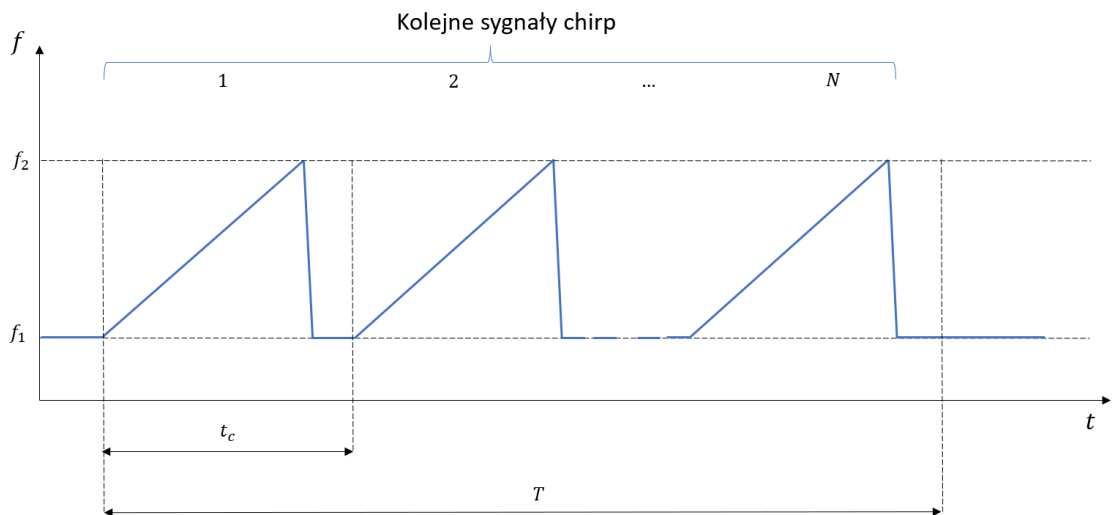
c – prędkość światła równa w przybliżeniu $3 * 10^8 \frac{m}{s}$

5.3. Dobór i konfiguracja radaru pod kątem wykrywania gestów

Układ posiada wiele parametrów kontrolujących pracę układu nadawczo-odbiorczego. Dzięki temu urządzenie może być wykorzystywane zarówno do wykrywania ruchu osób w odległości 10 metrów jak i do rozpoznawania gestów wykonanych dłońią w niewielkiej odległości rzędu kilkudziesięciu centymetrów. Zastosowaną konfigurację przedstawia tabela 6, a wpływ parametrów na działanie radaru i sposób ich doboru zostanie przedstawiony w następnym podrozdziale.

Tabela 6. Zastosowane parametry konfiguracyjne radaru

F_{ADC}	Częstotliwość próbkowania sygnału	2 MHz
G	Wzmocnienie sygnału częstotliwości pośredniej IF	30 dB
P_{out}	Moc Sygnału wyjściowego	5.206 dBm (parametr o wart. 31)
f_1	Częstotliwość startowa sygnału chirp	58 GHz
f_2	Częstotliwość końcowa sygnału chirp	63 GHz
N	Liczba nadanych sygnałów chirp na jeden cykl	64
n	Liczba próbek dla każdego cyklu sygnału chirp	32
t_c	Czas między początkiem kolejnych sygnałów chirp	390.4 us
T	Czas trwania cyklu	50 ms
R_{xn}	Liczba anten odbiorczych	2



Rysunek 56. Przebieg jednego z N cykli pomiarowych

Na rysunku 56 przedstawiono uproszczony przebieg jednego cyklu z naniesionymi parametrami odpowiadającymi opisom w tabeli. W ramach jednego skanu sygnał chirp jest powtarzany N razy, a następnie macierz danych o wymiarach $N * n * R_{xn}$ jest przesyłana do procesora głównego za pomocą protokołu SPI. Cykl powtarza się co zadany czas T , czyli nie występuje odstęp czasowy pomiędzy kolejnymi cyklami.

5.3.1. Wpływ parametrów na sposób pracy radaru.

Proces dobrania parametrów konfiguracyjnych radaru, należy rozpocząć od określenia wymagań dla systemu. W rozważanym przypadku założono wykrywanie gestów w odległości do 30 cm. Jako maksymalną prędkość gestów, które mogą zostać wykonane przez użytkownika na bazie pomiarów własnych przyjęto 1 m/s. Z uwagi na fakt, że rozważano wykrywanie pięciu gestów wykonywanych całą dłonią w różnych odległościach od radaru, rozdzielczość wykrywania odległości nie była kluczowym parametrem. W związku z tym założono maksymalną dopuszczalną rozdzielczość jako 5 cm. W przypadku rozdzielczości prędkości przyjęto wartość maksymalną 0.1 m/s. Rozdzielczość jest bezpośrednio związana z liczbą sygnałów chirp w cyklu pracy. Wzrost rozdzielczości wymaga większej długości cyklu, co znacząco zwiększa rozmiar ramki danych. W dalszej części rozdziału opisano związek parametrów systemowych z parametrami konfiguracyjnymi radaru. Poniżej przedstawiono zależności (19)-(22) opisujące wpływ parametrów konfiguracyjnych układu radarowego na parametry pracy systemu.

$$r_{res} = \frac{c}{2(f_2 - f_1)} \quad (19)$$

$$r_{max} = \frac{r_{res} n}{2} \quad (20)$$

$$v_{max} = \frac{\lambda}{4t_c} \quad (21)$$

$$v_{res} = \frac{2v_{max}}{N} \quad (22)$$

r_{res} - rozdzielczość mierzonego zakresu odległości między antenami a wykrytym obiektem

r_{max} - maksymalny zasięg wykrywania obiektu

v_{max} - maksymalna prędkość z jaką porusza się obiekt która może zostać zmierzona

v_{res} - rozdzielczość pomiaru prędkości poruszającego się obiektu

Po podstawieniu wartości z tabeli 4 otrzymano:

$$r_{res} = 0.03m = 30mm$$

$$r_{max} = 0.48m = 48cm$$

$$v_{max} = 3.18 \frac{m}{s}$$

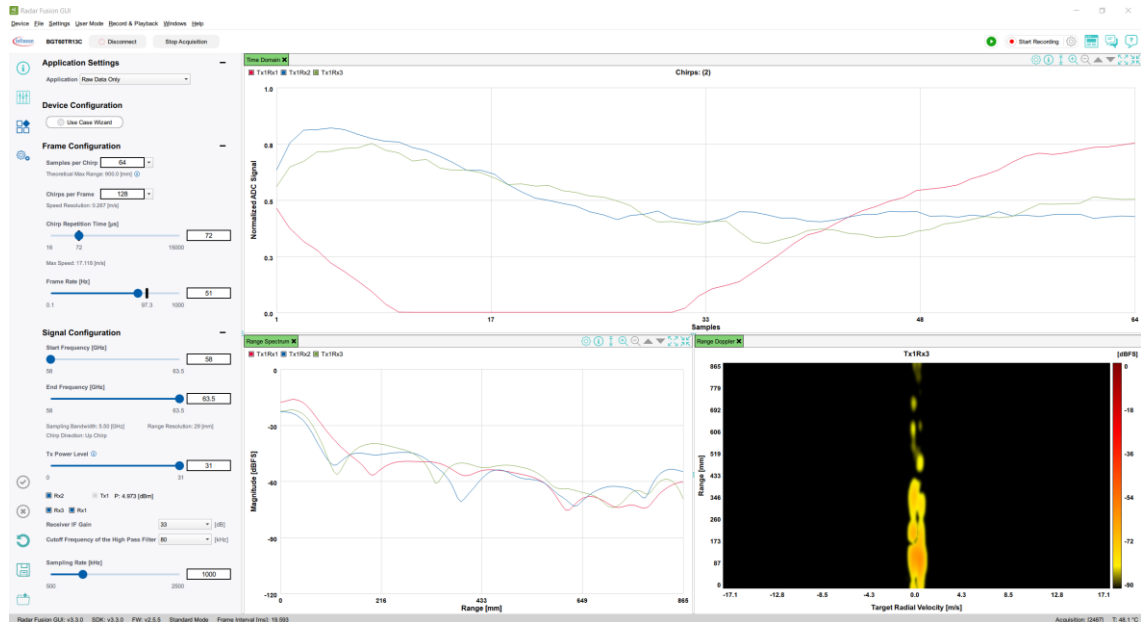
$$v_{res} = 0.099 \frac{m}{s}$$

Otrzymane wyniki spełniły wyżej opisane założenia.

5.4. Radar Fusion GUI Infineon

Firma Infineon udostępniła pakiet narzędzi, które znacznie ułatwiają rozpoczęcie pracy z omawianym modułem. Są one dostępne do pobrania ze strony producenta, po uprzednim założeniu konta [83]. Zawierają one przykłady algorytmów w kilku językach programowania, a także dedykowaną aplikację przeznaczoną do analizy wpływu parametrów pracy radaru na otrzymywane sygnały. Dzięki wykorzystaniu tego narzędzia możliwe jest natychmiastowe zobaczenie wpływu opisywanych wcześniej parametrów na pracę systemu oraz otrzymywane

przebiegi. Na rysunku 57 pokazano przebieg sygnałów analogowych, wykres amplitudy i częstotliwości oraz wykres Dopplera (we współrzędnych prędkość – odległość).



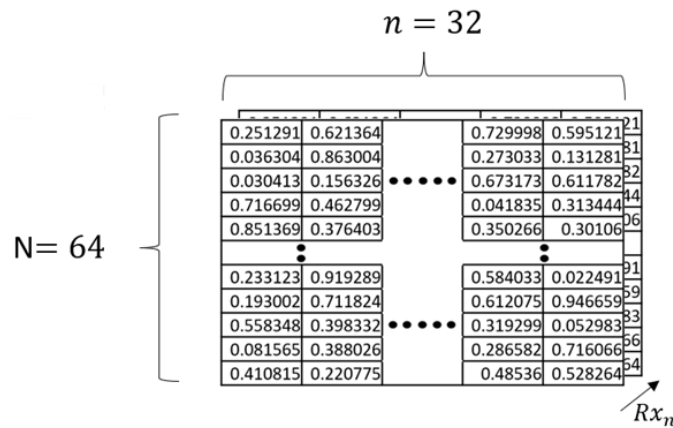
Rysunek 57. Widok narzędzia Radar Fusion GUI

5.5. Algorytmy wykorzystane do przetwarzania wstępnego

Dane otrzymane bezpośrednio z przetwornika analogowo-cyfrowego każdego z kanałów radaru są trudne w interpretacji. Zawierają zakłócenia oraz składowe pochodzące od fal odbitych od obiektów znajdujących się poza obszarem wykrywania. Należy zwrócić uwagę, że skonfigurowanie radaru do działania w bliskiej odległości od jego powierzchni, nie zatrzymuje propagacji fal poza tą przestrzeń, a także nie jest w stanie zablokować fal przybywających spoza obszaru roboczego. Powyższe uwarunkowania wymuszają zastosowanie stosunkowo wielu operacji przetwarzających dane, w celu otrzymania wielkości użytecznych do rozpoznawania gestów. Opisane w literaturze rozwiązania z wykorzystaniem radaru w podobnych aplikacjach, potwierdzają konieczność wykorzystania kilku do kilkunastu operacji na danych, zanim ostatecznie uzyskane cechy zostaną podane jako dane wejściowe dla sieci neuronowej [84], [85], [86].

Zgodnie z danymi konfiguracyjnymi, radar BGT60TR13C przesyła kolejne ramki z danymi odpowiadającymi kolejnym cyklom pomiarowym uruchamianym co czas T . W opisywanym przypadku wartość ta została ustalona na 50ms co odpowiada otrzymywaniu pakietu danych z częstotliwością 20 Hz. Każda ramka składa się z trójwymiarowej tablicy o wymiarach

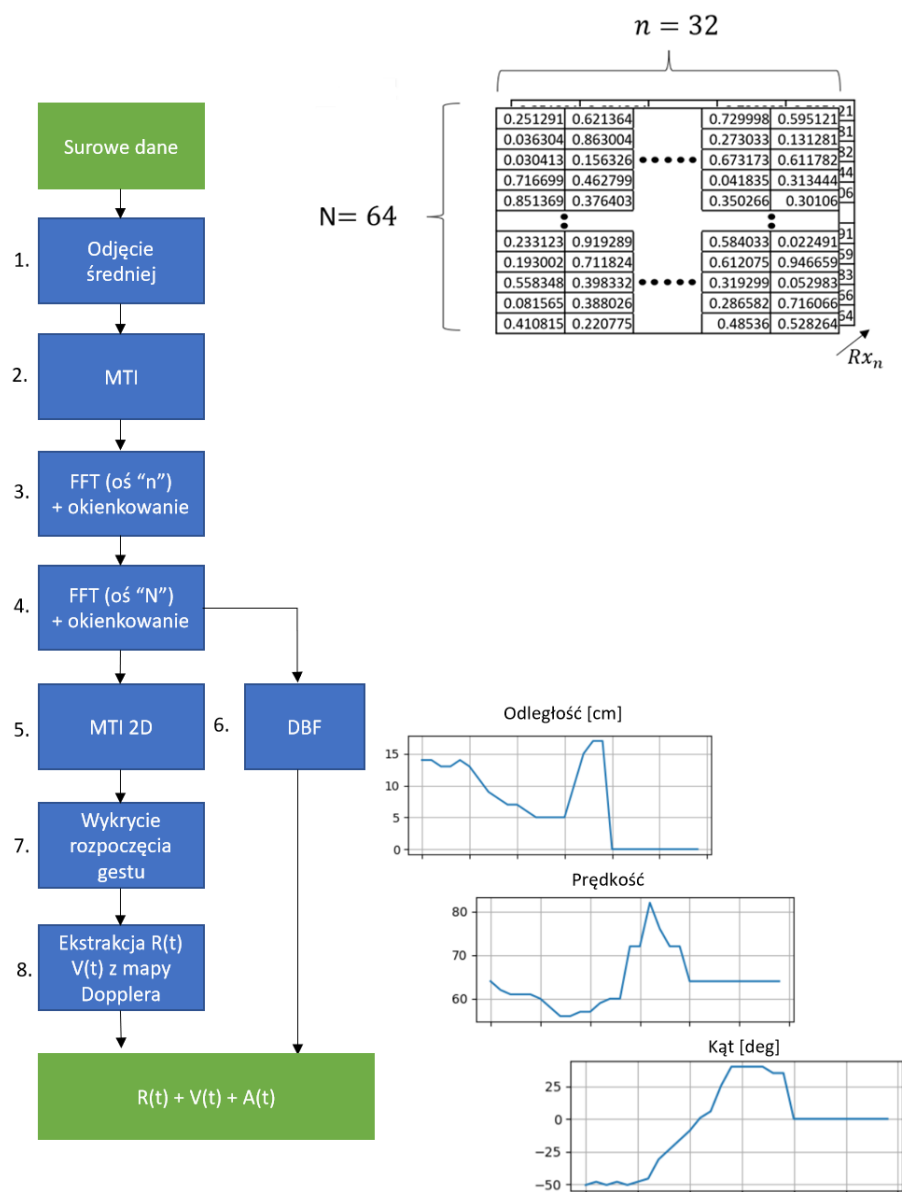
$N * n * R_{xn}$, gdzie N jest liczbą sygnałów chirp dla jednego cyklu pomiarowego, n stanowi liczbę próbek dla każdego sygnału, a R_{xn} to liczba wykorzystanych anten. W opisywanym systemie w każdym cyklu nadawano 64 sygnały chirp, które próbkowano 32 razy. Finalnie wykorzystano dwie anteny. Ostatecznie ramka danych przyjęła wymiary 64x32x2 komórki. Struktura ramki danych została przedstawiona na rysunku 58.



Rysunek 58. Struktura danych otrzymywana dla każdego cyklu pomiarowego

Każda komórka danych odpowiada 12 bitowej wartości ADC. W opisywanym systemie do komunikacji z radarem wykorzystano procesor ATSAM570Q21 z wbudowanym oprogramowaniem dostarczonym przez firmę Infineon. W aplikacji uruchomionej na komputerze wykorzystano biblioteki napisane w języku Python przeznaczone do komunikacji z mikrokontrolerem. W efekcie dla każdej komórki danych otrzymano wartość zmiennoprzecinkową w zakresie od 0 do 1 odpowiadającą 12-bitowym wartościom przesłanym bezpośrednio z radaru do procesora.

Na rysunku 59 przedstawiono kolejne operacje wykonywane na danych otrzymanych z radaru.



Rysunek 59. Algorytm przetwarzający dane otrzymane z radaru do przebiegów czasowych odległości, prędkości oraz kąta

Po wykonaniu wszystkich operacji z pobranych danych otrzymujemy wartości odległości, prędkości oraz kąta dla jednego cyklu pomiarowego. Kumulując otrzymane wartości z kolejnych cykli pomiarowych otrzymujemy przebiegi czasowe wymienionych wartości. Otrzymane przebiegi w następnym kroku są wykorzystywane jako dane wejściowe do sieci neuronowej.

1. Pierwszą z wykonywanych operacji jest odjęcie wartości średniej od wszystkich wartości w ramce. Równanie (23) przedstawia macierz S_0 , która reprezentuje dane otrzymane dla jednego cyklu pomiarowego i jednej anteny. Następnie wyrażenie (24) odpowiadające

odjęciu wartości średniej od wszystkich elementów macierzy S_0 , dające w rezultacie macierz S_1 .

$$S_0 = \begin{bmatrix} S_{1,1} & \cdots & S_{1,n} \\ \vdots & \ddots & \vdots \\ S_{N,1} & \cdots & S_{N,n} \end{bmatrix} \quad (23)$$

$$S_1 = S_0 - \overline{S_0} \quad (24)$$

S_0 – macierz zawierająca dane z pełnej serii N cykli pomiarowych o długości n każdy

$\overline{S_0}$ – macierz zawierająca wartości średniej macierzy S_0

S_1 – macierz po odjęciu wartości średniej

2. W drugim kroku wykonywany jest algorytm MTI (Moving Target Indicator – Wskaźnik Ruchomego Celu). Jest on wykorzystywany w przetwarzaniu sygnałów radarowych w celu uwidocznienia ruchomych obiektów, a także odfiltrowania szumu oraz sygnałów odbitych od obiektów statycznych. Jest to de facto filtr o nieskończonej odpowiedzi impulsowej (IIR) pierwszego rzędu. Operację przedstawiono na wyrażeniach (25) i (26). Otrzymujemy ostatecznie macierz S_2 , która powstaje poprzez odjęcie od macierzy S_1 danych historycznych, dotyczących poprzedniej ramki oznaczonych jako macierz SH_{k-1} . Parametr k jest liczbą całkowitą i określa kolejne ramki. Macierz SH powstaje w wyniku operacji z wykorzystaniem współczynnika α , który w przypadku opisywanego systemu przyjęto jako wartość 0.9.

$$S_2 = S_1 - SH_{k-1} \quad (25)$$

$$SH_k = \alpha S_1 + (1 - \alpha) SH_{k-1} \quad (26)$$

S_1 – macierz po odjęciu wartości średniej

SH_{k-1} – macierz z wartościami po wykonaniu algorytmu MTI dla poprzedniej serii pomiarowej

S_2 – macierz po przefiltrowaniu algorytmem MTI

SH_k - macierz z wartościami po wykonaniu algorytmu MTI dla obecnej serii pomiarowej

α – współczynnik filtrowania dla filtru IIR

3. Kolejnym krokiem było pobranie próbek sygnału za pomocą czterookresowego okna Blackmanna-Harrisa [87], którego charakterystykę przedstawiono na rysunku 60.

Okna stosuje się przede wszystkim w celu zminimalizowania wpływu nieciągłości sygnału na jego krańcach. Zastosowanie okna BH pozwala uzyskać większą rozdzielczość amplitudową, gwarantuje również brak zafałszowania widma [88]. Wadą jest niższa rozdzielczość częstotliwościowa. W opisywanej aplikacji różnice częstotliwości są na tyle duże, że mniejsza rozdzielczość jest akceptowalna. W opisywanej sekwencji obliczeń, każdy wiersz macierzy S_2 odpowiadający sygnałowi dla pojedynczego przebiegu chirp został pomnożony (28) przez funkcję okienkującą (27). Następnie wykonano operację FFT (Fast Fourier Transform – Szybka Transformata Fouriera) [89], [90] dla wszystkich przebiegów zawartych w macierzy S_W – wyrażenie (29). Transformata pozwala uzyskać informacje dotyczące przebiegu w dziedzinie częstotliwościowej i jest powszechnie wykorzystywana w przypadku operacji na sygnałach. Operacja FFT jest wykonywana na „szybkiej osi czasu” co oznacza, że jako oś czasu przyjęto oś odpowiadającą liczbie próbek dla jednego sygnału chirp - n . Dzięki temu uzyskujemy informację dotyczącą odległości obiektu od radaru. W efekcie otrzymujemy macierz liczb zespolonych o wymiarach 64x64. W tym wypadku wybieramy tylko dodatnią część spektrum (odległość od radaru będzie zawsze dodatnia), przez co ostatecznie otrzymujemy macierz S_3 o wymiarach 64 x 32. Pomnożenie wartości macierzy przez 2 wynika z konieczności zachowania energii sygnałów po odcięciu ujemnej części macierzy (30).

$$w(j) = 0.35875 - 0.48829 \cos\left(\frac{2\pi j}{n-1}\right) + 0.14128 \cos\left(\frac{4\pi j}{n-1}\right) - 0.01168 \cos\left(\frac{6\pi j}{n-1}\right) \quad (27)$$

$$S_W(i, j) = S_2(i, j)w(j) \quad (28)$$

$$S_F = FFT(S_W) \quad (29)$$

$$S_3 = 2 S_F[\gamma, \delta], \quad \gamma = \{0, N\}, \delta = \{n, 2n\} \quad (30)$$

$w(j)$ – funkcja okienkująca Blackmana-Harrisa

j – indeks dla kolejnych próbek odpowiedzi na pojedynczy sygnał chirp, $j = \{0, n\}$

i – indeks dla wektorów z próbkami odpowiedzi dla kolejnych serii sygnału chirp $i = \{0, N\}$

$S_W(i, j)$ – element i, j z macierzy po operacji okienkowania

$S_2(i, j)$ – element i, j z macierzy po filtrowaniu MTI

$FFT(S_W)$ – funkcja szybkiej transformaty Fouriera wykonana na macierzy z przebiegami po okienkowaniu

S_F – macierz uzyskana po przeprowadzeniu szybkiej transformaty Fouriera

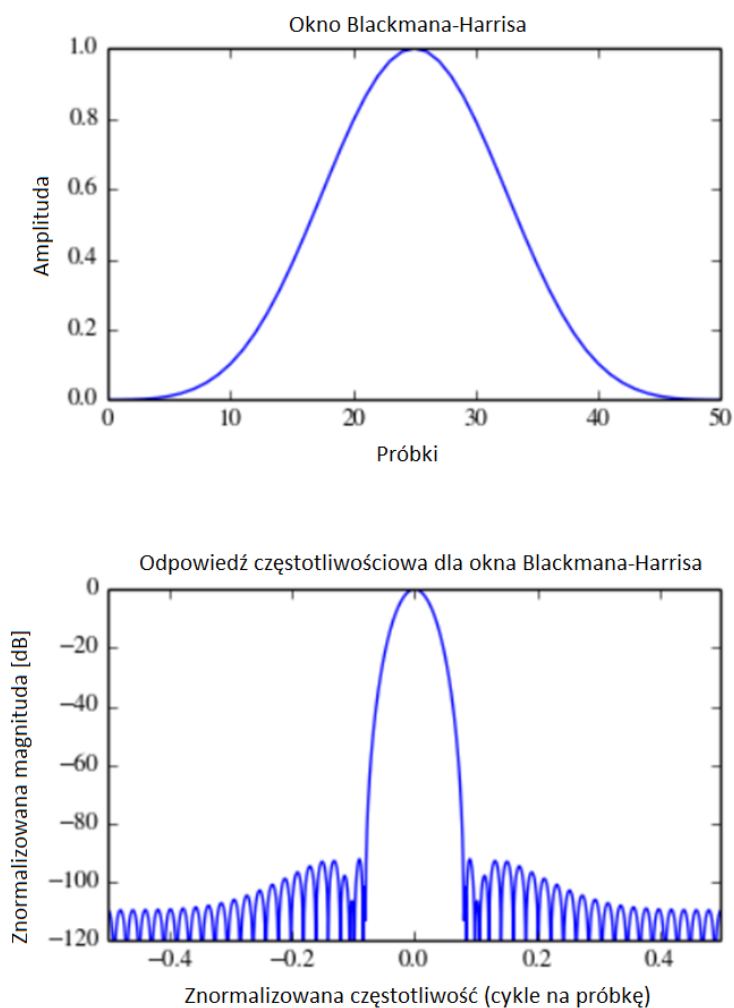
S_3 – macierz wynikowa z kroku nr 3

γ – zakres wierszy wyciąganych z macierzy S_F do uzyskania macierzy S_3

δ – zakres kolumn wyciąganych z macierzy S_F do uzyskania macierzy S_3

N - ilość nadanych sygnałów chirp na jeden cykl/serię pomiarową

n - ilość próbek dla każdego cyklu sygnału chirp



Rysunek 60. Okno czasowe Blackmana-Harrisa w odniesieniu do kolejnych próbek oraz w dziedzinie częstotliwości

4. Przed kolejnym krokiem dokonano transponowania macierzy S_3 . Jest to konieczne, ponieważ w tym kroku zostanie przeprowadzona transformata Fouriera sygnału wzdłuż „wolnej osi czasu”, czyli jako oś czasu zostanie przyjęty wymiar macierzy – N . Przed wyznaczeniem transformaty Fouriera ponownie wykonano operacje przemnożenia sygnału przez okno Blackmanna-Harisa, podobnie jak w poprzednim kroku (31). W efekcie wykonania transformacji Fouriera (32) otrzymano macierz o wymiarach 32x128. W tym wypadku nie było konieczne dzielenie otrzymanej macierzy S_4 ponieważ prędkość, która jest reprezentowana przez wymiar 128 macierzy może przyjmować zarówno wartości dodatnie jak i ujemne.

$$S_{W2}(j) = S_3^T(j)(i) w(i) \quad (31)$$

$$S'_4 = FFT(S_{W2}) \quad (32)$$

$w(i)$ – funkcja okienkująca Blackmana-Harrisa

j – indeks dla kolejnych próbek odpowiedzi na pojedynczy sygnał chirp, $j = \{0, n\}$

i – indeks dla wektorów z próbkami odpowiedzi dla kolejnych serii sygnału chirp $i = \{0, N\}$

$S_{W2}(i)$ – wiersz macierzy uzyskanej po okienkowaniu

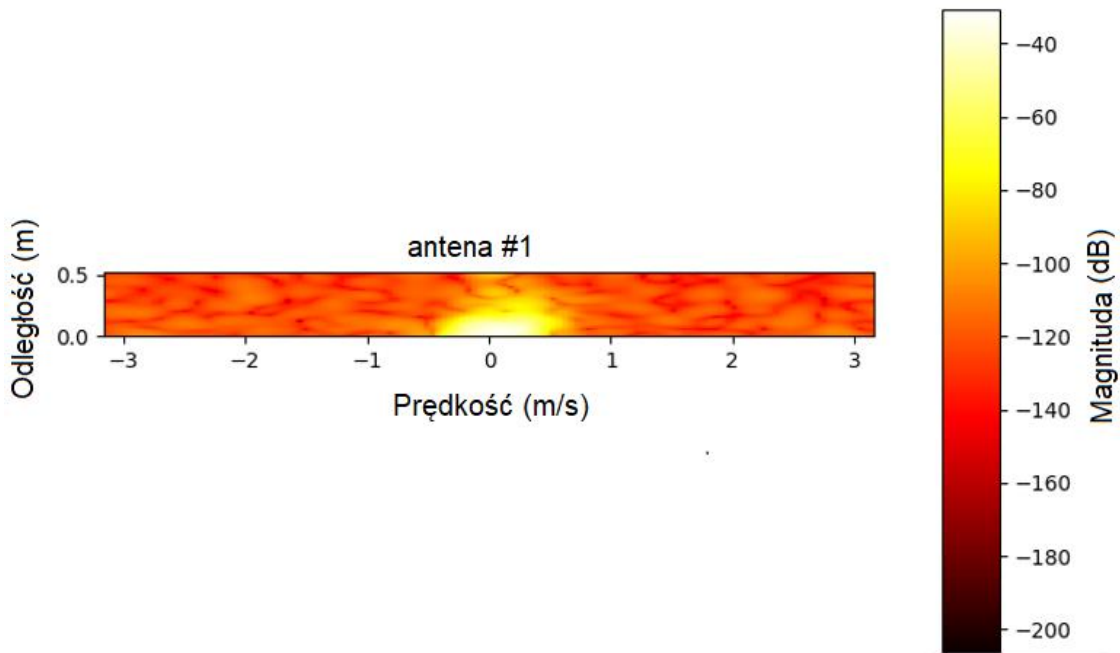
$FFT(S_{W2})$ – funkcja szybkiej transformaty Fouriera dokonanej na macierzy po okienkowaniu

S'_4 – macierz uzyskana po wykonaniu transformaty Fouriera

Po otrzymaniu macierzy S_4 w postaci liczb zespolonych, przeliczono wartości zespolone na decybele – wyrażenie (33). Na rysunku 61 przedstawiono graficzną reprezentację otrzymanej macierzy. Normalizacja nie była konieczna, ponieważ dane wejściowe na samym początku sekwencji zostały dostarczone w postaci znormalizowanej.

$$S_4 = 20 \log_{10}|abs(S'_4)| \quad (33)$$

S'_4 – macierz odpowiadająca wykresowi Dopplera wyrażona w decybelach



Rysunek 61. Wykres przedstawiający otrzymaną macierz, która odpowiada wykresowi Dopplera zakresu od prędkości

5. Ze względu na wyraźne rozmycie obiektu na powyższym rysunku 61, a także z powodu widocznych licznych szumów, zastosowano ponownie algorytm MTI, tym razem w wersji 2D (34). Macierz S_{avg} zawiera średnie wartości z pięciu poprzednich próbek czasowych. Dzięki temu uzyskano wykres, który w bardziej precyzyjny sposób wskazuje parametry wykrytego obiektu – rysunek 62.

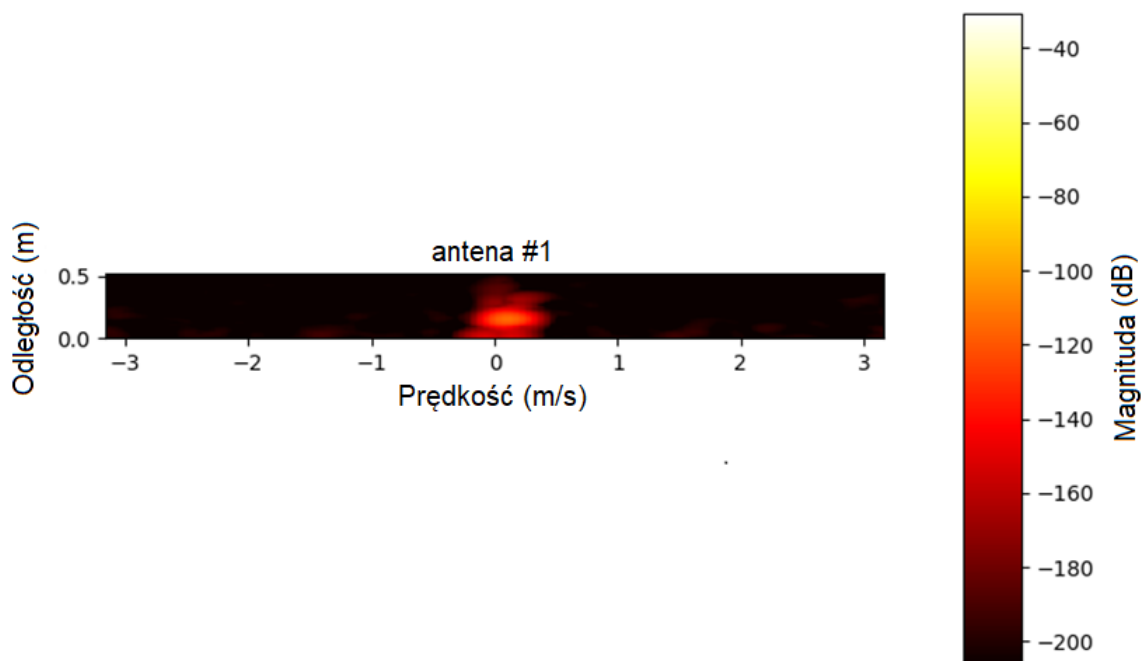
$$S_5 = S'_4 - \alpha_{2D} S_{avg} \quad (34)$$

S'_4 – macierz odpowiadająca wykresowi Dopplera wyrażona w decybelach

S_{avg} – macierz zawierająca wartości średnie dla elementów macierzy S'_4 , dla pięciu poprzednich przebiegów

α_{2D} – współczynnik określający sopieńodejmowania tła. Dla opisywanej aplikacji przyjęto 0.95.

S_5 - macierz odpowiadająca wykresowi Dopplera po odfiltrowaniu



Rysunek 62. Wykres Dopplera po zastosowaniu algorytmu MTI 2D

6. Równoległe do Algorytmu 2D MTI, dla potrzeb wyznaczenia położenia kąowego obiektu względem radaru wykorzystano algorytm DBF (Digital Beam Former). Wymaga on wykorzystania sygnału z minimum dwóch anten. Tak jak wspomniano w sekcji dotyczącej fizycznych właściwości radaru warunkiem koniecznym jego działania jest spełnienie zależności (11). Ogólna zasada działania algorytmu polega na wyznaczeniu wirtualnych promieni odpowiadających kierunkom, w których będzie analizowana energia sygnału dla otrzymanych diagramów Dopplera. W praktyce wyznaczono wektor promieni B . Do jego wyznaczenia należy określić minimalny i maksymalny kąt oraz liczbę promieni. W opisywanym systemie zakres kątowy przyjęto od -45° do 45° ($-\theta, \theta$) oraz liczbę promieni b wynoszącą 45, co w efekcie pozwala uzyskać rozdzielczość kątową 2° ($\Delta\theta$). Jest to odzwierciedlone poprzez wektor A (35), który posłuży do wygenerowania wektora B (36). Interpretację geometryczną algorytmu przedstawiono na rysunku 63. Wyrażenie (37) przedstawia operację mnożenia macierzy S_5 reprezentującej wykres Dopplera przez współczynniki każdego z promieni. Następnie dla każdej macierzy wynikowej odpowiadającej k -temu promieniowi obliczono iloraz z jej normy Frobeniusa oraz pierwiastka liczby promieni (38). W efekcie otrzymano wektor S_{BE} , zawierający wartości energii odpowiadające każdemu z promieni. Kąt wyznaczamy znajdując indeks odpowiadający maximum wektora S_{BE} , a następnie pobierając za jego pomocą, ostateczną wartość kąta z wektora A (k).

$$A = \{-\theta, -\theta + \Delta\theta, \dots, +\theta\} \quad (35)$$

$$B = e^{2\pi i \frac{d}{\lambda} \sin A} \quad (36)$$

$$S_B(k) = S_5 B(k) \quad (37)$$

$$S_{BE}[k] = \frac{\|S_B[k]\|_F}{\sqrt{b}} \quad (38)$$

$-\theta, \theta$ – zakres wykrywanych kątów

b – liczba promieni

$\Delta\theta$ – rozdzielczość wyznaczania kąta

A – wektor zawierający wszystkie możliwe do wyznaczenia wartości kąta

B – wektor promieni

d – odległość między antenami

λ – długość fali

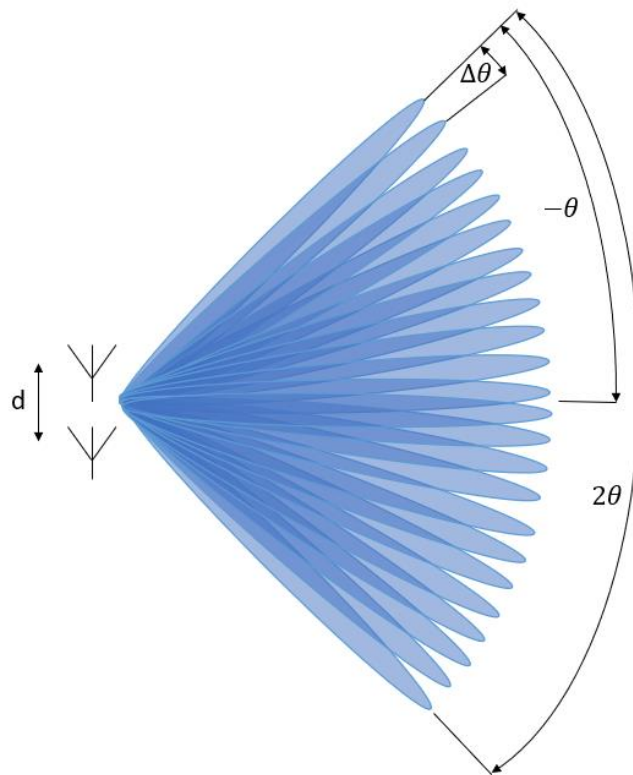
i – jednostka urojona

k – indeks kolejnych elementów wektora B , w zakresie $\{0, b\}$

S_B – macierz wynikająca z pomnożenia mapy Dopplera z wektorem promieni

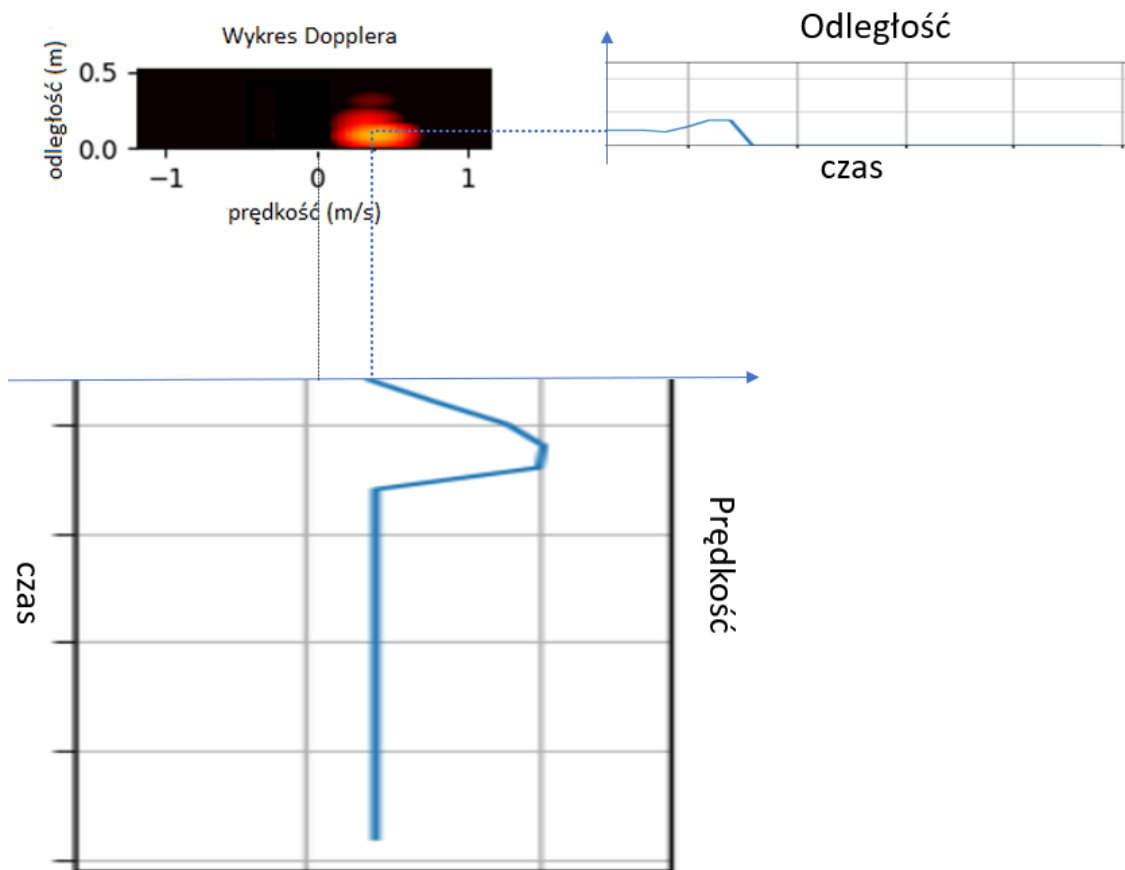
$\|S_B[k]\|_F$ – norma Frobeniusa dla k -tego elementu macierzy S_B

S_{BE} – macierz uzyskana z ilorazu normy Frobeniusa z macierzy S_B oraz pierwiastka liczby promieni



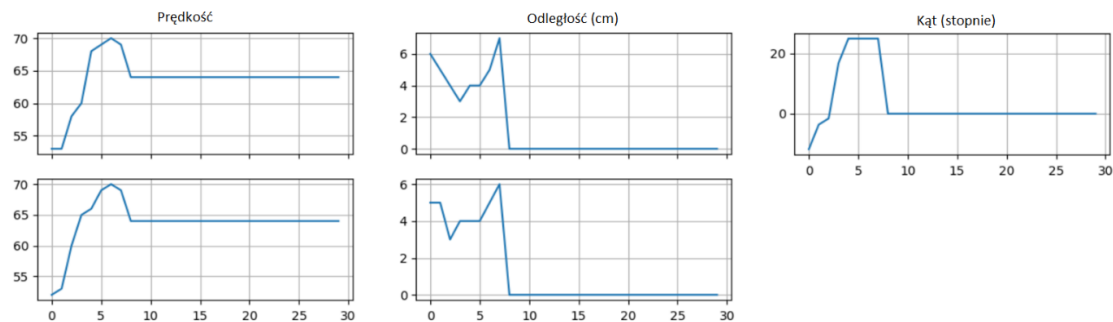
Rysunek 63. Geometryczna ilustracja algorytmu Digital Beam Forming

7. Po zakończeniu kroków 5 i 6 uzyskano mapy dopplerowskie oraz wartość kąta dla pojedynczej ramki danych. W kolejnym kroku należało dokonać detekcji pojawienia się obiektu w przestrzeni roboczej radaru. Do tego celu po każdym uruchomieniu radaru zbierane są dane z dwudziestu pierwszych cykli pomiarowych. Następnie z każdego cyklu wybierana jest wartość maksymalnej energii sygnału w decybelach. W kolejnym kroku dodawana jest dodatkowa wartość korygująca. W efekcie otrzymywana jest wartość progowa, po której przekroczeniu wykrywane jest pojawienie się obiektu w przestrzeni roboczej, co ostatecznie oznacza rozpoczęcie wykonywania gestu. Następnie dane z trzydziestu kolejnych próbek są przypisywane do dedykowanego bufora. Wynika to z założonego okna czasowego o długości 1.5 s.
8. Zbieranie i dalsze przetwarzanie wszystkich trzydziestu map dla dwóch anten byłoby bardzo obciążające w kontekście mocy obliczeniowej. Dlatego zdecydowano o zredukowaniu pełnego obrazu do pojedynczych wartości reprezentujących prędkość oraz odległość punktu na mapie Dopplera, dla którego energia sygnału była maksymalna. Wykorzystano do tego centroid (środek masy) wykresu. Rzutowanie danych przedstawiono na rysunku 64.



Rysunek 64. Ekstrakcja odległości oraz prędkości z Mapy Dopplera

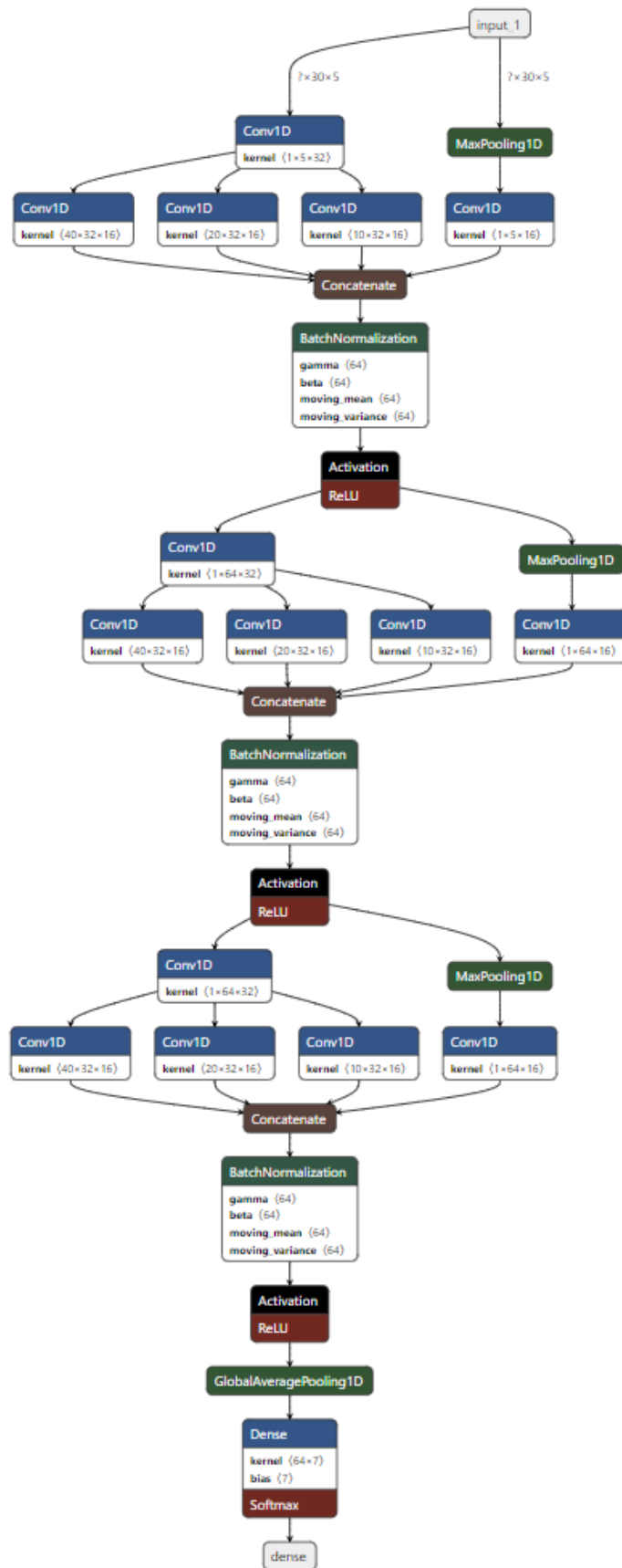
Po zebraniu dla wszystkich trzydziestu próbek wartości prędkości, odległości oraz kąta, dane są zapisywane do buforów. W ten sposób otrzymujemy przebiegi czasowe pokazane na rysunku 65. Kompletne przebiegi są następnie normalizowane oraz przekazywane jako cechy do modelu sieci neuronowej.



Rysunek 65. Odległość, prędkość oraz kąt w czasie okna pomiarowego

5.6. Model sieci neuronowej – system radarowy

Podobnie jak w przypadku systemu wykorzystującego jako źródło sygnału elektrody pojemnościowe, zastosowano sieć neuronową o architekturze „Inception Time” [58]. W opisywanym systemie również jako cechy wejściowe dla modelu wykorzystywane są przebiegi czasowe. Wybór tej samej architektury został podyktowany planem zastosowania fuzji danych opisanej w kolejnych rozdziałach. Wektor wejściowy stanowiło pięć przebiegów czasowych: prędkość dla anteny 1 i 2, Odległość dla anteny 1 i 2 oraz kąt. Na poniższym rysunku 66 przedstawiono strukturę zastosowanego modelu. Jest ona taka sama jak opisana w rozdziale „Model sieci neuronowe – system pojemnościowy”. Główną różnicą jest format wektora wejściowego, co jest spowodowane inną długością przebiegów czasowych stanowiących dane wejściowe dla modelu.

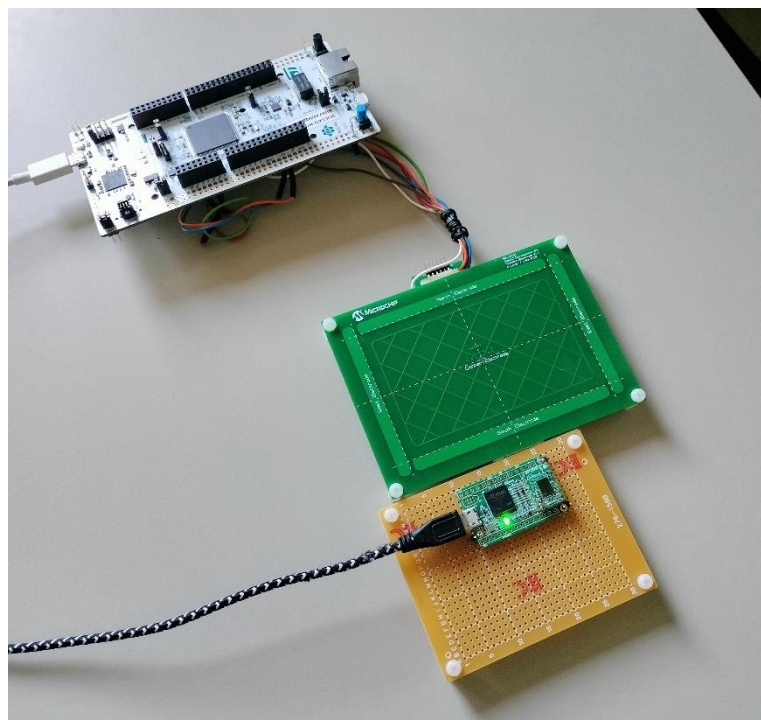


Rysunek 66. Struktura modelu sieci neuronowej dla systemu radarowego

6. Fuzja danych z systemów pojemnościowego oraz radarowego

6.1. Zaproponowane podejście do fuzji danych

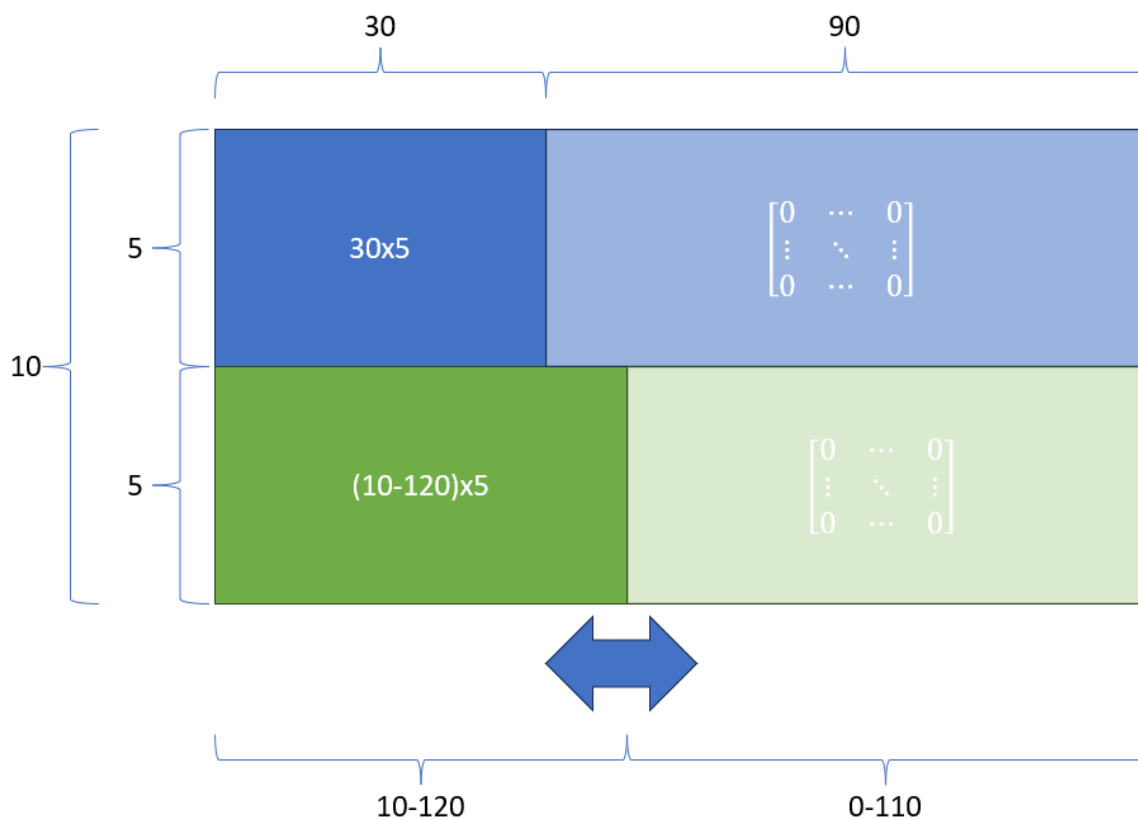
W niniejszym rozdziale zostaną przedstawione wyniki testów przeprowadzonych z wykorzystaniem systemów pojemnościowego oraz radarowego. Każde z rozwiązań zostało szerzej opisane w rozdziałach: „System do wykrywania gestów oparty o czujniki pojemnościowe” oraz „System do wykrywania gestów oparty o radar”. Na rysunku 67 przedstawiono wzajemne umiejscowienie systemów.



Rysunek 67. Umiejscowienie systemu pojemnościowego oraz radarowego podczas zbierania danych dla potrzeb przeprowadzenia fuzji danych

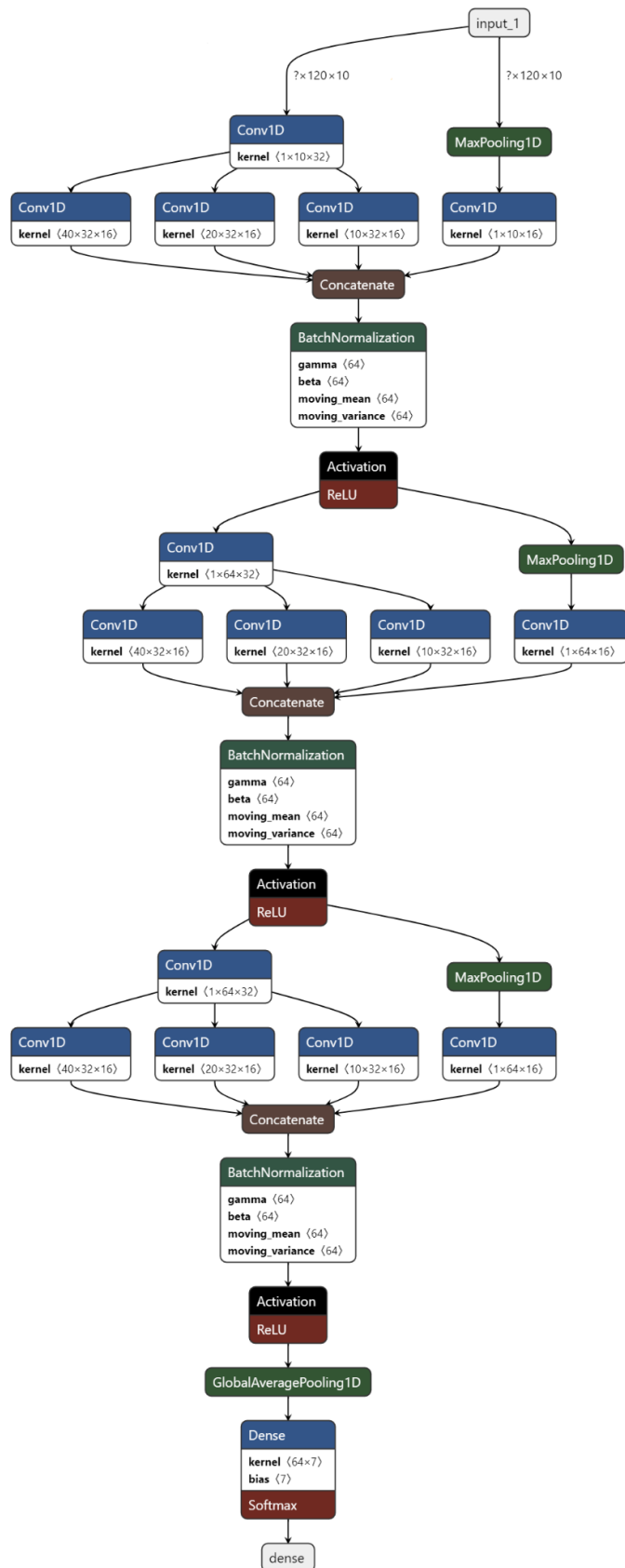
Systemy zostały połączone przez UART/USB z komputerem, na którym równoległe uruchomiono dwie aplikacje służące do akwizycji oraz oznaczenia próbek zebranych dla każdego z systemów. Synchronizacja czasowa zebranych próbek odbywała się za pomocą zapisania czasu rozpoczęcia akwizycji danych dla każdej zebranej próbki na obu systemach. Ze względu na różnice w interwale pomiarów, a także długości okna pomiarowego obu systemów, zebrane struktury danych posiadały inny rozmiar. W przypadku systemu pojemnościowego zostały zaimplementowane kryteria wykrywania rozpoczęcia oraz zakończenia gestu.

W związku z tym okno pomiarowe dla gestu miało zmienną długość. Ze względu na potrzebę ujednolicenia danych między systemami, były one rozszerzane przez dodanie zer w celu osiągnięcia takiej samej długości dla każdego z przebiegów. Rozwiązanie z radarem miało okno pomiarowe o stałej długości – trzydzieści próbek co 50 ms. W tym przypadku dla przetwarzania danych nie zachodziła potrzeba rozszerzania danych. Stało się to konieczne w kontekście fuzji danych z obu systemów. Zastosowano podobne podejście jak w przypadku systemu pojemnościowego. Sposób wyrównywania długości wszystkich elementów dla wektora wejściowego dla sieci neuronowej przedstawiono na poniższym rysunku 68.



Rysunek 68. Wyrównywanie rozmiaru danych wejściowych dla sieci neuronowej dokonującej ich fuzji

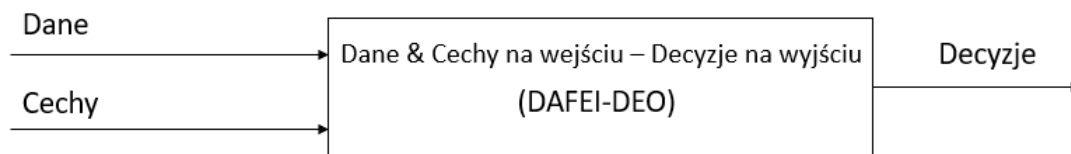
Fuzja danych została przeprowadzona poprzez wykorzystanie modelu sieci neuronowej Inception Time (tak jak w poprzednich przypadkach) z odpowiednio rozszerzonym wektorem wejściowym – rysunek 68. Podsumowując, fuzja de facto odbyła się w modelu sieci neuronowej, której strukturę przedstawiono na rysunku 69.



Rysunek 69. Model sieci neuronowej zastosowany dla fuzji danych

Zaproponowane rozwiązanie można zakwalifikować w następujący sposób:

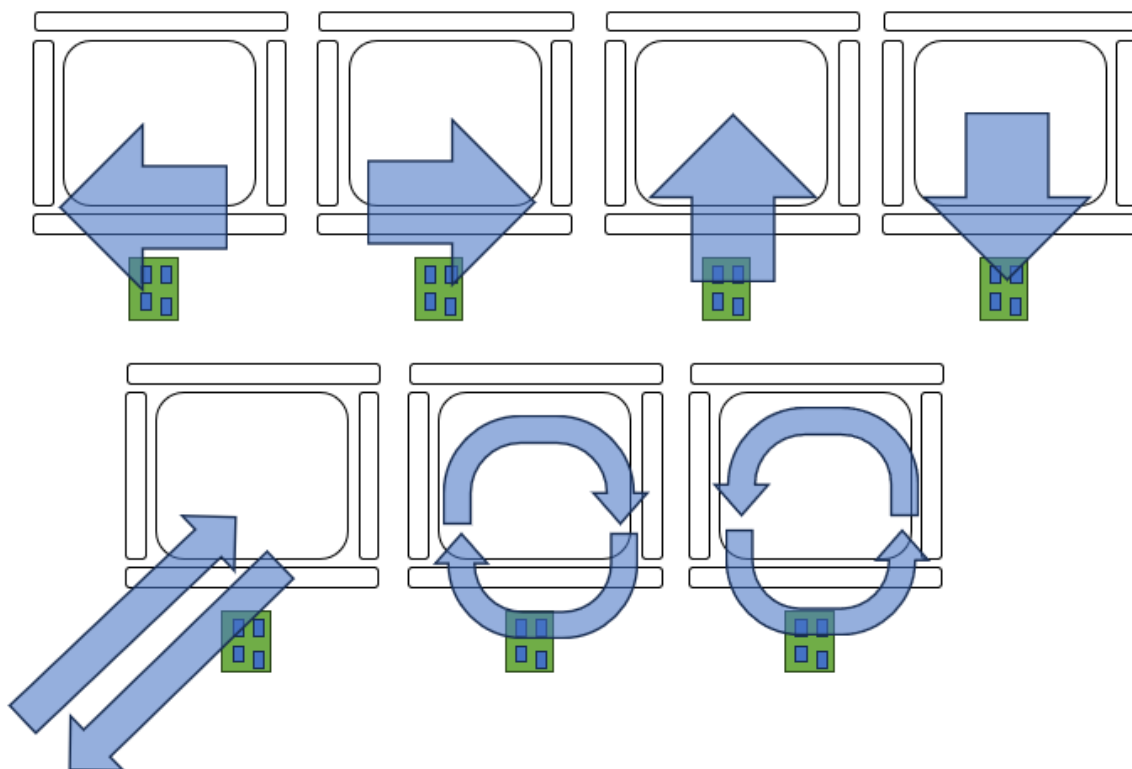
- Ze względu na architekturę: architektura zcentralizowana – przetwarzanie i ekstrakcja danych następują osobno, następnie fuzja jest dokonywana przez jeden model w postaci rozszerzonej sieci neuronowej.
- Ze względu na poziom abstrakcji danych wejściowych: jest to podejście łączące metody wymienione w rozdziale „Klasyfikacja metod fuzji danych”: w postaci DAI-FEO (Dane na wejściu – Cechy na wyjściu) oraz FEI-DEO (Cechy na wejściu – Decyzje na wyjściu).
W efekcie można zaproponować klasyfikację przedstawioną na rysunku 70 (DAFEI-DEO - Dane oraz Cechy na wejściu – Decyzje na wyjściu). Kanał wejściowy z systemu pojemnościowego stanowi kanał z danymi, natomiast dane z radaru zostały poddane ekstrakcji cech (prędkość, odległość, kąt), i w takiej formie trafiają do modelu sieci neuronowej.
- Ze względu na relacje między danymi źródłowymi: Komplementarna – Dane pochodzą z dwóch niezależnych sensorów pracujących w różnych domenach.
- Ze względu na poziom abstrakcji: W zależności od interpretacji jest to połączenie poziomu 2 i 3 lub poziom 3.



Rysunek 70. Zaproponowany sposób fuzji danych – Dane i Cechy na wejściu – Decyzje na wyjściu

W celu przeprowadzenia procesu uczenia oraz testowania skuteczności fuzji danych zebrano 4000 gestów. Ze względu na znacząco niższy zakres działania systemu pojemnościowego, zarejestrował on 3284 gesty. Gesty zostały zebrane od 15 osób w wieku od 25-65 lat obu płci. Zestaw gestów został podzielony na dane uczące, walidacyjne oraz testowe w proporcji 80:10:10. W pierwszym podejściu przeprowadzono uczenie modelu za pomocą 5 gestów (przesunięcie w lewo (swipe left), przesunięcie w prawo (swipe right), przesunięcie w górę (swipe up), przesunięcie w dół (swipe down), zbliżenie (tap)). W drugim podejściu rozszerzono zestaw gestów o kołowy ruch zgodnie ze wskazówkami zegara oraz przeciwnie do

wskazówek zegara (rotation clockwise, rotation anticlockwise). Wszystkie gesty przedstawiono na rysunku 71.



Rysunek 71. Rysunek przedstawiający wykonywane gesty w obszarze roboczym systemów. Kolejno: przesunięcie w lewo (swipe left), przesunięcie w prawo (swipe right), przesunięcie w górę (swipe up), przesunięcie w dół (swipe down), zbliżenie (tap), kołowy ruch zgodnie ze wskazówkami zegara (rotation clockwise), Kołowy przeciwnie do ruchu wskazówek zegara (rotation anticlockwise)

6.2. Wyniki fuzji danych dla systemów pojemnościowego oraz radarowego

W tabeli 7 przedstawiono skuteczność rozpoznawania gestów dla systemu pojemnościowego, radarowego oraz po dokonaniu fuzji na podstawie danych z obu systemów.

Tabela 7. Skuteczność rozpoznawania gestów dla poszczególnych modeli sieci neuronowej

5 Gestów			7 Gestów		
System radarowy	System pojemnościowy	System radarowy i pojemnościowy	System radarowy	System pojemnościowy	System radarowy i pojemnościowy
93.0%	94.2%	97.7%	87.8%	92.5%	97.1%

Niższa skuteczność dla systemu radarowego wynika z większego zakresu pracy, przez co w sygnałach zakłócenia występowały częściej niż w przypadku systemu pojemnościowego. Dodatkowo konstrukcja sensora pojemnościowego znacząco ułatwia rozpoznawanie przesunięć ze względu na osobne elektrody dla każdego kierunku. W przypadku radaru, wykorzystano tylko dwie anteny odbiorcze co utrudnia rozpoznawanie kierunku. Rozważając systemy pracujące osobno widać spadek skuteczności przy wzroście liczby rozpoznawanych gestów. W przypadku systemu pojemnościowego jest on niewielki – poniżej 2 %. Dla systemu wykorzystującego radar jest to ponad 5 %. Zarówno w przypadku mniejszego oraz większego zestawu gestów połączenie danych za pomocą modelu sieci neuronowej o rozszerzonym wektorze wejściowym przyniosło pozytywne rezultaty. Wzrost skuteczności rozpoznawania gestów wyniósł odpowiednio 3.5% oraz 4.6% w stosunku do skuteczniejszego systemu pojemnościowego. **Przedstawione wyniki dowodzą słuszności tezy o zwiększeniu skuteczności rozpoznawania gestów dzięki, zastosowaniu fuzji danych zawartej w rozdziale „Cel i zakres pracy” i pokazują, iż w rozważanym przypadku najlepszym rozwiązaniem jest zaproponowane rozwiązanie z wykorzystaniem danych z obu systemów.**

Porównując metodę rozpoznawania gestów dla wszystkich trzech przypadków, koszt obliczeniowy dla samego wykonania algorytmu sieci neuronowej jest minimalnie większy (zwiększona ilość danych wejściowych). Dodatkowo, zazwyczaj będzie istnieć konieczność wyrównywania lub synchronizacji danych. Z punktu widzenia ilości obliczeń, wzrost w przypadku zaproponowanej metody fuzji jest bardzo niewielki. Ograniczenia metody są

tożsame dla wszystkich systemów wykorzystujących sieci neuronowe. Przede wszystkim jest to ilość zasobów pamięciowych oraz obliczeniowych. Mikrokontrolery o większych zasobach stanowią większy koszt, przez co w niektórych aplikacjach dodanie funkcjonalności sterowania gestami może okazać się nieopłacalne. Poza wspomnianymi limitami, elastyczność sieci neuronowych znacząco ułatwia ich wykorzystanie do łączenia danych z różnych sensorów.

7. Podsumowanie oraz możliwe dalsze kierunki rozwoju

Podczas realizacji programu doktorat wdrożeniowy przeprowadzono szereg analiz oraz testów dotyczących potencjalnych technologii, które mogłyby być wykorzystane do rozpoznawania gestów. Dokonano przeglądu rozwiązań dostępnych na rynku oraz oceniono ich przydatność pod kątem zastosowania do systemu rozpoznającego gesty. Przetestowano wiele sensorów następujących typów: elektrody pojemnościowe, podczerwieni, ultradźwiękowe oraz radar. Następnie zbudowano dwa prototypy systemów – pojemnościowy oraz radarowy. W obu systemach algorytmem służącym do rozpoznawania gestów była sieć neuronowa oparta na modelu dedykowanym dla przebiegów czasowych. System pojemnościowy został zamontowany w samochodzie, dzięki czemu możliwe jest testowanie tej technologii w docelowym środowisku, a także prezentacja rozwiązania potencjalnym klientom firmy Merit. Ograniczenia związane z niewielkim zakresem pracy systemu pojemnościowego doprowadziły do zbudowania systemu opartego o radar. Oba systemy spełniają postawione wymagania oraz charakteryzują się zadawalającą skutecznością wykrywania gestów. W kontekście zwiększenia ilości rozpoznawania gestów oraz wzrostu skuteczności, podjęto prace zakończone przeprowadzeniem eksperymentów z fuzją danych, które potwierdziły jej opłacalność.

Analizując otrzymane wyniki przez pryzmat fuzji danych, pierwszym aspektem, którego poprawa może przynieść jeszcze lepsze rezultaty, jest większa integracja obu systemów. Można to osiągnąć poprzez lepszą synchronizację akwizycji danych oraz ustalenie identycznego okna pomiarowego. W zaprezentowanym rozwiązaniu próbki danych tworzące przebiegi pochodziły z różnych momentów czasowych. Dodatkowo okno pomiarowe systemu radarowego było znacznie dłuższe niż pojemnościowego. Skutkowało to zwiększoną wrażliwością systemu radarowego na niejednoznaczności sygnału podczas zakończenia wykonywania gestu – wynikające przykładowo z powolnego wycofywania ręki z obszaru roboczego. Problem ten mógłby zostać rozwiązany poprzez zaimplementowanie kryterium rozpoznającego zakończenie gestu dla systemu radarowego, podobnie jak ma to miejsce dla systemu pojemnościowego. Kolejnym aspektem, który mógłby poprawić uzyskane wyniki, może być zastosowanie jednego okna pomiarowego na podstawie wykrycia rozpoczęcia oraz zakończenia gestu przez jeden z systemów. W takim przypadku drugi z nich nie wykrywałby rozpoczęcia gestu, a tylko zajmował się akwizycją kolejnych danych.

Następnym rozwiązaniem ułatwiającym synchronizację oraz fuzję danych dla rozważanych systemów jest stworzenie jednej aplikacji na komputerze, a docelowo systemu wbudowanego

integrującego oba systemy. W przypadku systemu radarowego istotnym czynnikiem, może okazać się wykorzystanie dodatkowej anteny co pozwoli wykrywać kąt w drugiej płaszczyźnie. W opisanym systemie, konfiguracja radaru (krótki zakres pracy w stosunku do możliwości sprzętowych) nie pozwalał na wykorzystanie wszystkich anten, ze względu na brak wyraźnego sygnału na trzeciej z nich. Na rynku istnieją rozwiązania z większą ilością anten co może stanowić obiecująca alternatywę.

Projekt został zakończony pomyślnie, ponieważ zaprojektowano i przetestowano rozwiązanie niewymagające użycia systemów wizyjnych, w postaci systemu pojemnościowego. Dodatkowo jest ono niskokosztowe i łatwo integrowalne w produktach z portfolio firmy Merit. System pojemnościowy wymaga wykorzystania układu MGC3140 oraz wykonania elektrod w postaci dodatkowych ścieżek na płycie PCB, co stanowi koszt około 10% wszystkich materiałów wykorzystanych do budowy standardowego przełącznika. Mikrokontrolery są wykorzystywane w większości produktów firmy Merit. W związku z tym w przypadku rozszerzenia modułu o funkcjonalność gestów, konieczne jest wybranie wariantu mikrokontrolera z większą ilością pamięci RAM i flash. Integracja za równo systemu radarowego oraz obu systemów w kontekście fuzji danych, jest również wykonalna, ponieważ oba systemy mogą zostać z łatwością zabudowane. Ze względu na dużą liczbę operacji koniecznych do wykonania na danych otrzymywanych z radaru, wzrost wymagań dla mikrokontrolera byłby znaczący. Dodatkowo modułowość rozwiązań oraz ich niewielkie rozmiary, ułatwiają potencjalną integrację w produktach. Przedstawione w niniejszej rozprawie systemy są prezentowane klientom firmy Merit jako kompatybilne rozwiązania z obecnie oferowanymi produktami. W efekcie stały się częścią portfolio produktowego.

8. Bibliografia

- [1] C. Darwin, *The Expression of the Emotions in Man and Animals*, Cambridge: Cambridge University Press, 2013.
- [2] A. Tsukamoto i C. W. Lee, „A Methodological approach on real-time gesture recognition using multiple silhouette models” w *Proceedings 4th IEEE International Workshop on Robot and Human Communication*, Tokyo, 1995.
- [3] H. Yabe, T. Nishimura, R. Oka i T. Mukai, „Recognition of gestures using morphological features of networks made of gesture motion images and word sequences” w *Proceedings - International Workshop on Recognition, Analysis, and Tracking of Faces and Gestures in Real-Time Systems, RATFG-RTS 1999*, 1999.
- [4] D. J. Sturman i D. Zeltzer, „A survey of glove-based input” *IEEE Computer Graphics and Applications*, tom 14, nr 1, pp. 30-39, Styczeń 1994.
- [5] D. L. Quam, „Gesture recognition with a DataGlove” *IEEE Conference on Aerospace and Electronics*, nr 2, pp. 755-760 vol.2, 1990.
- [6] S. Akyol, U. Canzler, K. Bengler i W. Hahn, „Gesture Control for use in Automobiles” w *Proceedings of the IAPR Conference on Machine Vision Applications (IAPR MVA 2000)*, Tokyo, 2000.
- [7] W. Cunningham, „BMW finally embraces touchscreens, adds gesture control.” CNET, 6 Jan 2015. [Online]. Dostęp: <https://www.cnet.com/roadshow/news/bmw-finally-embraces-touchscreens-adds-gesture-control/>. [Data uzyskania dostępu: 18 06 2023].
- [8] Aptiv, „What Is Gesture Recognition?” Aptiv, 30 April 2021. [Online]. Dostęp: <https://www.aptiv.com/en/insights/article/what-is-gesture-recognition>. [Data uzyskania dostępu: 24 08 2023].
- [9] M. Caputo, K. Denker, B. Dums i G. Umlauf, „3D Hand Gesture Recognition Based on Sensor Fusion of Commodity Hardware” *Mensch & Computer 2012: interaktiv informiert – allgegenwärtig und allumfassend*, pp. 293-302, 2012.
- [10] Y. Liu, M. Dong, S. Bi, D. Gao, Y. Jing i L. Li, „Gesture recognition based on Kinect” w *6th Annual IEEE International Conference on Cyber Technology in Automation, Control and Intelligent Systems, IEEE-CYBER 2016*, 2016.
- [11] T. Q. Vinh i N. T. Tri, „Hand gesture recognition based on depth image using kinect sensor” *Proceedings of 2015 2nd National Foundation for Science and Technology Development Conference on Information and Computer Science, NICS 2015*, pp. 34-39, 2015.
- [12] S. Frank i A. Kuijper, „Enabling Driver Feet Gestures Using Capacitive Proximity Sensing” w *Proceedings - 2018 International Conference on Intelligent Environments, IE 2018*, 2018.
- [13] S. Zeiß, A. Marinc, A. Braun, T. Grosse-Puppendahl i S. Beck, „A Gesture-Based Door Control Using Capacitive Sensors BT” w *International Conference on Distributed, Ambient, and Pervasive Interactions, DAPI 2014*, 2014.
- [14] H. Wei, P. Li, K. Tang, W. Wang i X. Chen, „Alternating electric field-based static gesture-recognition technology” *Sensors (Switzerland)*, tom 19, nr 10, 2019.
- [15] Motor1, „2018 Jaguar XF Sportbrake gesture blind - YouTube” [Online]. Dostęp: <https://www.youtube.com/watch?v=vPYzoB6viqU>. [Data uzyskania dostępu: 24 06 2023].

- [16] Microchip, „MGC3140 3D Tracking and Gesture Controller Data Sheet” 2022. [Online]. Dostęp: <https://ww1.microchip.com/downloads/aemDocuments/documents/MCU08/ProductDocuments/DataSheets/MGC3140-3D-Tracking-and-Gesture-Controller-Data-Sheet-40002037C.pdf>. [Data uzyskania dostępu: 30 05 2023].
- [17] Microchip, „GestIC[®] Design Guide” 2013. [Online]. Dostęp: <https://ww1.microchip.com/downloads/en/DeviceDoc/40001716C.pdf>. [Data uzyskania dostępu: 24 06 2023].
- [18] G. Batchuluun, B. Odgerel i C. H. Lee, „Hand Gesture Recognition Using an Infrared Proximity Sensor Array” *The International Journal of Fuzzy Logic and Intelligent Systems*, tom 15, nr 3, pp. 186-191, 2015.
- [19] H. Miura, H. Ishiwata, Y. Matunaga, S. Numazaki, A. Morisita, M. Doi i A. Semiconductor, „A 100 frame/s CMOS active pixel sensor for 3D-gesture recognition system” w *1999 IEEE International Solid-State Circuits Conference. Digest of Technical Papers. ISSCC. First Edition (Cat. No.99CH36278)*, San Francisco, 1999.
- [20] Avnet, „Gesture Recognition, Proximity Sensors Drive Advances in Automotive Infotainment.” 17 3 2017. [Online]. Dostęp: <https://www.avnet.com/wps/portal/apac/resources/article/gesture-recognition-proximity-sensors-drive-advances-auto-infotainment/>. [Data uzyskania dostępu: 24 06 2023].
- [21] Car and Driver, „We Swipe Right on Volkswagen's Production-Bound Gesture-Control Interface – News – Car and Driver” 7 01 2016. [Online]. Dostęp: <https://www.caranddriver.com/news/a15349506/we-swipe-right-on-volkswagens-production-bound-gesture-control-interface/>. [Data uzyskania dostępu: 24 06 2023].
- [22] Elmos, „Gesture and Object Recognition. Elmos HALIOS and ToF Imager” [Online]. Dostęp: <https://www.elmos.com/produkte/sensor-ics/tof-imager.html>. [Data uzyskania dostępu: 24 06 2023].
- [23] Elmos, „HALIOS[®] Controller for Proximity and Gesture Recognition” [Online]. Dostęp: <https://www.elmos.com/produkte/sensor-ics/optical-ir-sensor-ic/automotive/e90921.html>. [Data uzyskania dostępu: 24 06 2023].
- [24] Instrument Plastics, „Optolite IR - Nota Katalogowa” [Online]. Dostęp: <https://www.instrumentplastics.co.uk/perch/resources/data-sheets/optolite-ir.-infra-red-filters-1.pdf>. [Data uzyskania dostępu: 27 04 2023].
- [25] Advancing Physics, „April, 1935: British Patent for Radar System for Air Defense Granted to Robert Watson-Watt” *APS News*, tom 15, nr 4, Kwiecień 2006.
- [26] Google, „Soli” [Online]. Dostęp: <https://atap.google.com/soli/>. [Data uzyskania dostępu: 24 06 2023].
- [27] S. Wang, J. Song, J. Lien, I. Poupyrev i O. Hilliges, „Interacting with soli: Exploring fine-grained dynamic gesture recognition in the radio-frequency spectrum” w *UIST 2016 - Proceedings of the 29th Annual Symposium on User Interface Software and Technology*, 2016.
- [28] K. A. Smith, C. Csech, D. Murdoch i G. Shaker, „Gesture Recognition Using mm-Wave Sensor for Human-Car Interface” *IEEE Sensors Letters*, tom 2, nr 2, pp. 1-4, 2018.
- [29] Infineon, „BGT60TR13C Operating Principle Introduction Radar workshop during Evertiq” 2022.

- [30] RadarTutorial.eu, „Frequency-Modulated Continuous-Wave Radar (FMCW Radar)” [Online]. Dostęp: <https://www.radartutorial.eu/02.basics/Frequency%20Modulated%20Continuous%20Wave%20Radar.en.html>. [Data uzyskania dostępu: 24 06 2023].
- [31] C. Iovescu, „The fundamentals of millimeter wave radar sensors” Maj 2017. [Online]. Dostęp: <http://www.ee.fju.edu.tw/images/file/data3169/9a41b2657fe717233110268d676d38ef.pdf>. [Data uzyskania dostępu: 24 06 2023].
- [32] Texas Instruments, „AWR1243 Single-Chip 77- and 79-GHz FMCW Transceiver” Grudzień 2021. [Online]. Dostęp: https://www.ti.com/lit/ds/symlink/awr1243.pdf?ts=1685517172412&ref_url=https%253A%252F%252Fwww.ti.com%252Fproduct%252FAWR1243. [Data uzyskania dostępu: 30 05 2023].
- [33] Texas Instruments, „AWR1843. Single-chip 76-GHz to 81-GHz automotive radar sensor integrating DSP, MCU and radar accelerator” [Online]. Dostęp: <https://www.ti.com/product/AWR1843>. [Data uzyskania dostępu: 24 06 2023].
- [34] R. R. Fay i A. N. Popper, *Comparative Hearing: Mammals*, Nowy Jork: Springer-Verlag, 1994.
- [35] Y. Qifan, T. Hao, Z. Xuebing, L. Yin i Z. Sanfeng, „Dolphin: Ultrasonic-based gesture recognition on smartphone platform” w *Proceedings - 17th IEEE International Conference on Computational Science and Engineering, CSE 2014, Jointly with 13th IEEE International Conference on Ubiquitous Computing and Communications, IUCC 2014, 13th International Symposium on Pervasive Systems,*, 2015.
- [36] Y. Sang, L. Shi i Y. Liu, „Micro hand gesture recognition system using ultrasonic active sensing” *IEEE Access*, tom 6, pp. 49339-49347, 2018.
- [37] F. Zhou, X. Li i Z. Wang, „Efficient High Cross-User Recognition Rate Ultrasonic Hand Gesture Recognition System” *IEEE Sensors Journal*, tom 20, nr 22, pp. 13501-13510, 11 2020.
- [38] H. Chen, T. Ballal, M. Saad i T. Y. Al-Naffouri, „Angle-of-arrival-based gesture recognition using ultrasonic multi-frequency signals” w *2017 25th European Signal Processing Conference (EUSIPCO)*, Kos, Grecja, 2017.
- [39] R. Moxon, A. Allison i G. C. England, „Effect of Ultrasonic Devices on the Distraction Behaviour of Guide Dogs” *International Journal of Orientation & Mobility*, tom 3, nr 1, pp. 27-37, 2010.
- [40] M. Haid, B. Budaker, M. Geiger, D. Husfeldt, M. Hartmann i N. Berezowski, „Inertial-Based Gesture Recognition for Artificial Intelligent Cockpit Control using Hidden Markov Models” w *2019 IEEE International Conference on Consumer Electronics, ICCE 2019*, 2019.
- [41] H. Wen, J. Ramos Rojas i A. K. Dey, „Serendipity: Finger Gesture Recognition Using an Off-the-Shelf Smartwatch” w *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, Nowy Jork, 2016.
- [42] S. Oniga, J. Vegh i I. Orha, „Intelligent human-machine interface using hand gestures recognition” w *2012 IEEE International Conference on Automation, Quality and Testing, Robotics, AQTR 2012 - Proceedings*, 2012.
- [43] Y. Zhang, B. Liu, Z. Liu, J. Huang i R. Sun, „WristPress: Hand gesture classification with two-array wrist-mounted pressure sensors” w *2019 IEEE 16th International Conference on Wearable and Implantable Body Sensor Networks, BSN 2019 - Proceedings*, 2019.

- [44] T. S. Saponas, D. S. Tan, D. Morris, J. Turner i J. A. Landay, „Making Muscle-Computer Interfaces More Practical” w *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '10)*, Nowy Jork, 2010.
- [45] A. Jaramillo-Yáñez, M. E. Benalcázar i E. Mena-Maldonado, Real-time hand gesture recognition using surface electromyography and machine learning: A systematic literature review, tom 20, MDPI AG, 2020.
- [46] F. E. White, „Data_Fusion_Lexicon” Defense Technical Information Center, 1991.
- [47] L. Wald, „Definitions and terms of reference in data fusion” w *International Archives of Photogrammetry and Remote Sensing*, Valladolid, 1999.
- [48] E. I. by Bloch, A. Hunter, A. Appriou, A. Ayoun, S. Benferhat, P. Besnard, L. Cholvy, R. Cooke, F. Cuppens, D. Dubois, H. Fargier, M. Grabisch, R. Kruse, J. Lang, S. Moral, H. Prade, A. Saffiotti, P. Smets i C. Sossai, „Fusion: General Concepts and Characteristics” *Article in International Journal of Intelligent Systems*, tom 16, pp. 1107-1134, 2001.
- [49] F. Mastrogiovanni, A. Sgorbissa i R. Zaccaria, „A Distributed Architecture for Symbolic Data Fusion.” w *IJCAI 2007, Proceedings of the 20th International Joint Conference on Artificial Intelligence*, Hyderabad, 2007.
- [50] W. Koch, M. Framework i S. Applications, Tracking and Sensor Data Fusion, Berlin: Springer Verlag, 2013.
- [51] F. Castanedo, „A review of data fusion techniques” *The Scientific World Journal*, tom 2013, 2013.
- [52] M. Huang, Z. Liu i T. Yang, „Mechanical fault diagnosis and prediction in IoT based on multi-source sensing data fusion” *Simulation Modelling Practice and Theory*, tom 102, July 2020.
- [53] D. Durrant-Whyte, „Sensor Models and Multisensor Integration” *The International Journal of Robotics Research*, tom 7, nr 6, pp. 97-113, 1988.
- [54] B. V. Dasarathy, „Sensor Fusion Potential Exploitation-Innovative Architectures and Illustrative Applications” *Proceedings of the IEEE*, tom 85, nr 1, pp. 24-38, January 1997.
- [55] S. Ben Ayed, H. Trichili i A. M. Alimi, „Data fusion architectures: A survey and comparison” w *International Conference on Intelligent Systems Design and Applications, ISDA*, 2016.
- [56] Microchip, „MGC3140 GestIC® Library Interface Description User's Guide” 2019. [Online]. Dostęp: <http://ww1.microchip.com/downloads/en/devicedoc/mgc3140-gestid-library-interface-description-40001875a.pdf>. [Data uzyskania dostępu: 24 06 2023].
- [57] Microchip, „AN2934 Capacitive Touch Sensor Design Guide” 2020. [Online]. Dostęp: <https://ww1.microchip.com/downloads/en/Appnotes/Capacitive-Touch-Sensor-Design-Guide-DS00002934-B.pdf>. [Data uzyskania dostępu: 2023 06 24].
- [58] H. Ismail Fawaz, B. Lucas, G. Forestier, C. Pelletier, D. F. Schmidt, J. Weber, G. I. Webb, L. Idoumghar, P. A. Muller i F. Petitjean, „InceptionTime: Finding AlexNet for time series classification” *Data Mining and Knowledge Discovery*, tom 34, nr 6, pp. 1936-1962, 11 2020.
- [59] H. I. Fawaz, G. Forestier, J. Weber, F. Petitjean, L. Idoumghar i P.-A. Muller, „Automatic alignment of surgical videos using kinematic data” w *Lecture Notes in Computer Science*, Springer, 2019.
- [60] F. Yi, Z. Yu, F. Zhuang, X. Zhang i H. Xiong, „An Integrated Model for Crime Prediction Using Temporal and Spatial Factors” w *Proceedings - IEEE International Conference on Data Mining, ICDM*, Singapur, 2018.

- [61] Y. Yuan, G. Xun, F. Ma, Y. Wang, N. Du, K. Jia, L. Su i A. Zhang, „MuVAN: A Multi-view Attention Network for Multivariate Temporal Data” w *Proceedings - IEEE International Conference on Data Mining, ICDM*, Singapur, 2018.
- [62] C. Pelletier, G. I. Webb i F. Petitjean, „Temporal convolutional neural network for the classification of satellite image time series” *Remote Sensing*, tom 11, nr 5, Marzec 2019.
- [63] A. Horzyk, „Sztuczna Inteligencja, Uczenie głębokie i głębokie sieci neuronowe - Deep Learning” [Online]. Dostęp: <https://home.agh.edu.pl/~horzyk/lectures/ai/SztucznaInteligencja-UczenieG%C5%82%C4%99bokichSieciNeuronowych.pdf>. [Data uzyskania dostępu: 22 06 24].
- [64] K. Banerjee, V. P. C, R. R. Gupta, K. Vyas, A. H i B. Mishra, „Exploring Alternatives to Softmax Function” 11 2020. [Online]. Dostęp: <http://arxiv.org/abs/2011.11538>. [Data uzyskania dostępu: 2023 06 24].
- [65] P. Y. Simard, D. Steinkraus i J. C. Platt, „Best practices for convolutional neural networks applied to visual document analysis” w *Proceedings of the International Conference on Document Analysis and Recognition, ICDAR*, 2003.
- [66] A. Krizhevsky, I. Sutskever i G. E. Hinton, „ImageNet Classification with Deep Convolutional Neural Networks” *Communications of the ACM*, tom 60, nr 6, pp. 84-90, Czerwiec 2017.
- [67] Google, „TensorFlow” [Online]. Dostęp: <https://www.tensorflow.org/>. [Data uzyskania dostępu: 2023 06 24].
- [68] Google, „Keras” [Online]. Dostęp: <https://keras.io/>. [Data uzyskania dostępu: 24 06 2023].
- [69] N. Gowdra, R. Sinha, S. MacDonell i W. Yan, „Maximum Categorical Cross Entropy (MCCE): A noise-robust alternative loss function to mitigate racial bias in Convolutional Neural Networks (CNNs) by reducing overfitting” w *International Conference on Learning Representations*, Wiedeń.
- [70] D. P. Kingma i J. Ba, „Adam: A Method for Stochastic Optimization” 12 2014. [Online]. Dostęp: <http://arxiv.org/abs/1412.6980>. [Data uzyskania dostępu: 24 06 2023].
- [71] Google, „Optimizers - Keras” [Online]. Dostęp: <https://keras.io/api/optimizers/>. [Data uzyskania dostępu: 24 06 2023].
- [72] HDF5 group, „HDF5 File Format Specification Version 3.0” [Online]. Dostęp: https://docs.hdfgroup.org/hdf5/develop/_f_m_t3.html. [Data uzyskania dostępu: 24 06 2023].
- [73] ST, „AI expansion pack for STM32CubeMX” [Online]. Dostęp: <https://www.st.com/en/embedded-software/x-cube-ai.html#overview>. [Data uzyskania dostępu: 24 06 2023].
- [74] Y. Roh, G. Heo i S. E. Whang, „A Survey on Data Collection for Machine Learning: a Big Data -- AI Integration Perspective” 11 2018. [Online]. Dostęp: <http://arxiv.org/abs/1811.03402>. [Data uzyskania dostępu: 24 06 2023].
- [75] C. Baru, Institute of Electrical and Electronics Engineers i IEEE Computer Society, „Eliminating Data Collection Bottleneck for Wake Word Engine Training Using Found and Synthetic Data” w *2019 IEEE International Conference on Big Data (Big Data)*, 2019.
- [76] UFactory, „UFACTORY uArm Test Kit” [Online]. Dostęp: <https://www.ufactory.cc/product-page/ufactory-uarm-test-kit>. [Data uzyskania dostępu: 2023 06 23].
- [77] M. Matsumoto i T. Nishimura, „Mersenne Twister: A 623-Dimensionally Equidistributed Uniform Pseudo-Random Number Generator” *ACM Trans. Model. Comput. Simul.*, tom 8, nr 1, pp. 3-30, 1 1998.

- [78] P. Reczek, J. Panczyk, A. Wetula i Młyniec Andrzej, „Data Collection Automation in Machine Learning Process Using Robotic Manipulator” w *IFIP Advances in Information and Communication Technology*, Leon, 2023.
- [79] Y. Ahmed, „Fundamentals of a Uniform Linear Array (ULA)” 21 January 2018. [Online]. Dostęp: <https://www.raymaps.com/index.php/fundamentals-of-a-uniform-linear-array-ula/>. [Data uzyskania dostępu: 2023 06 23].
- [80] A. Chepala, V. Fusco, U. Naeem i A. McKernan, „Uniform Linear Antenna Array Beamsteering Based on Phase-Locked Loops” *Electronics (Switzerland)*, tom 12, nr 4, 2 2023.
- [81] Infineon, „BGT60TR13C 60 GHz Radar Sensor - Datasheet V2.4.6,” [Online]. Dostęp: https://www.infineon.com/dgdl/Infineon-BGT60TR13CDataSheet-DataSheet-v01_00-EN.pdf?fileId=8ac78c8c7d718a49017d94bac88e5d43. [Data uzyskania dostępu: 24 06 2023].
- [82] Infineon, „BGT60TR13C shield 60 GHz radar system platform Board version 2.2,” 2021. [Online]. Dostęp: https://www.infineon.com/dgdl/Infineon-BGT60TR13C_Shield-ApplicationNotes-v02_04-EN.pdf?fileId=8ac78c8c7d0d8da4017d3318e31c2ad6. [Data uzyskania dostępu: 24 06 2023].
- [83] Infineon, „Radar Development Kit,” [Online]. Dostęp: <https://softwaretools.infineon.com/tools/com.ifx.tb.tool.ifxradarsdk>. [Data uzyskania dostępu: 24 06 2023].
- [84] M. Chmurski, G. Mauro, A. Santra, M. Zubert i G. Daganan, „Highly-optimized radar-based gesture recognition system with depthwise expansion module” *Sensors*, tom 21, nr 21, 11 2021.
- [85] G. Mauro, M. Chmurski, L. Servadei, M. Pegalajar-Cuellar i D. P. Morales-Santos, „Few-Shot User-Definable Radar-Based Hand Gesture Recognition at the Edge” *IEEE Access*, tom 10, pp. 29741-29759, 2022.
- [86] P. Nallabolu, L. Zhang, H. Hong i C. Li, „Human Presence Sensing and Gesture Recognition for Smart Home Applications with Moving and Stationary Clutter Suppression Using a 60-GHz Digital Beamforming FMCW Radar” *IEEE Access*, tom 9, pp. 72857-72866, 2021.
- [87] F. J. Harris, „On the Use of Windows for Harmonic Analysis with the Discrete Fourier Transform” *Proceedings of the IEEE*, tom 66, nr 1, pp. 51-83, 1978.
- [88] K. Dąbrowski, *Cyfrowa Obróbka Sygnałów OE1KDA*, Wiedeń, 2018.
- [89] E. O. Brigham i R. E. Morrow, „The fast Fourier transform” *IEEE Spectrum*, tom 4, nr 12, pp. 63-70, 1967.
- [90] C. M. RADER i G. C. MALING, „What Is the Fast Fourier Transform?” *Proceedings of the IEEE*, tom 55, nr 10, pp. 1664-1674, 1967.