



AKADEMIA GÓRNICZO-HUTNICZA IM. STANISŁAWA STASZICA W KRAKOWIE

DZIEDZINA NAUK INŻYNIERYJNO-TECHNICZNYCH

**DYSCYPLINA AUTOMATYKA, ELEKTRONIKA, ELEKTROTECHNIKA I
TECHNOLOGIE KOSMICZNE**

ROZPRAWA DOKTORSKA

*Adaptacja modelu matematycznego czujnika radarowego na potrzeby
testów zaawansowanych systemów wspomagania kierowcy
w środowiskach wirtualnej jazdy*

Autor: *Michał Jasiński*

Promotor pracy: *dr hab. inż. Paweł Skruch, prof. AGH*

Promotor pomocniczy: *dr inż. Mateusz Komorkiewicz*

Praca wykonana: *Akademia Górniczo-Hutnicza im. Stanisława Staszica w Krakowie
Wydział Elektrotechniki, Automatyki, Informatyki
i Inżynierii Biomedycznej
Katedra Automatyki i Robotyki*

Kraków, 2024

Chciałbym serdecznie podziękować mojemu promotorowi oraz promotorowi pomocniczemu za poświęcony mi czas oraz ich cenne uwagi. Dziękuje również moim najbliższym za ich wsparcie, w szczególności Ani, Tomkowi i rodzicom. W końcu, dziękuję mojej babci Jadwidze Hoffmann-Grell za pomoc w korekcie tejże rozprawy.

Pracę dedykuje śp. Bohdanowi Grellowi, ukochanemu dziadkowi, bez którego ten tekst nigdy by nie powstał.

Preambuła

Badania te są wynikiem programu „Doktorat wdrożeniowy”, finansowanego przez polskie Ministerstwo Nauki i Szkolnictwa Wyższego (MNiSW), numer projektu 0014/DW/2018/02, i przeprowadzonego we współpracy z Aptiv Services Poland S.A., Centrum Technicznym Kraków i Akademią Górniczo-Hutniczą, Wydział Elektrotechniki, Automatyki, Informatyki i Inżynierii Biomedycznej.

Rezultaty pracy zostały wdrożone do praktyki przemysłowej do weryfikacji elektronicznych systemów sterowania jazdy automatycznej na poziomie L2-L3 przeznaczonych do produkcji seryjnej dla kluczowym producentów samochodów.

Streszczenie

Zaawansowane systemy wspomagania kierowcy (ang. Advanced Driver Assistance Systems, ADAS) pełnią dziś kluczową rolę w kontekście zapewnienia bezpieczeństwa w ruchu drogowym. Systemy te analizują w czasie rzeczywistym najbliższe otoczenie samochodu, bazując na informacjach zwróconych przez czujniki zamocowane na tym pojeździe, by w razie wystąpienia realnego zagrożenia przejąć kontrolę nad maszyną. Jedną z najważniejszych aplikacji jest zaawansowany system hamowania awaryjnego (ang. Advanced Emergency Braking System, AEBS), która ma na celu uniknięcie potencjalnie groźnej w skutkach kolizji, na przykład w momencie utraty koncentracji przez kierowcę lub w warunkach ograniczonej przejrzystości powietrza. Układy takie jak AEBS oparte są w znacznej mierze na pomiarach z radarów samochodowych, które dla pojazdów widocznych z perspektywy sensora obliczają ich odległość od czujnika, prędkość względną oraz kąt w płaszczyźnie horyzontalnej pomiędzy obiektem a celownikiem radaru (ang. boresight).

Weryfikacja urządzeń klasy ADAS to niezwykle kluczowe zagadnienie. W szczególności przed wdrożeniem systemu do produkcji należy w rzetelny sposób ocenić jego działanie w krytycznych sytuacjach. W tym celu przeprowadza się szereg jazd weryfikacyjnych, w trakcie których gromadzi się dane z czujników zamontowanych na samochodzie testowym. Następnie, na podstawie informacji zebranych z sensorów, analizuje się decyzje podjęte przez układ typu ADAS w trakcie danej serii przejazdów. Zrealizowanie tego typu kampanii w całości na drogach publicznych to niesłychanie kosztowna operacja. Dlatego coraz częściej producenci samochodów decydują się wykonać część eksperymentów w symulatorach jazdy. Dzięki takiemu podejściu, określanemu mianem wirtualnej walidacji, możliwe jest nie tylko skrócenie procesu oceny, ale także zweryfikowanie systemu w sytuacjach, które w rzeczywistości nie byłyby możliwe do przetestowania ze względów bezpieczeństwa.

Wirtualna walidacja musi być oczywiście wiarygodna, dlatego konieczne jest, by symulacja precyzyjnie odwzorowywała rzeczywistość. Oznacza to przede wszystkim konieczność uwzględnienia modeli matematycznych sensorów, generujących sztuczne pomiary wiernie naśladujące dane zwracane przez rzeczywiste czujniki. Jak wspomniano, radary samochodowe pełnią kluczową rolę we współczesnych systemach typu ADAS. Niestety, sensory te są silnie niedeterministyczne, głównie ze względu na losowość procesu propagacji fali elektromagnetycznej. Dlatego też opracowanie

modelu matematycznego czujnika radarowego jest zadaniem nietrywialnym. Istotna jest tutaj nie tylko dokładność odwzorowania, ale także niska złożoność obliczeniowa modułu, by umożliwić wirtualny test układu typu ADAS w czasie rzeczywistym.

W literaturze opisano do tej pory wiele skutecznych rozwiązań. Niemniej jednak analizując dostępne prace można zauważyć, że wciąż nie opisano metody charakteryzującej się zarówno wysokim poziomem wiarygodności, jak i prostotą w opracowaniu i w kalibracji. W tej pracy omówiony zostanie innowacyjny model matematyczny czujnika radarowego oparty na niezwykle ciekawej koncepcji centrów rozpraszania, reprezentujących w przybliżony sposób rozproszenie fali elektromagnetycznej w przestrzeni. Model ten zostanie dostosowany do wymagań przemysłu motoryzacyjnego związanych z weryfikacją urządzeń klasy ADAS w symulacji. Po pierwsze, model będzie sformalizowany pod kątem kluczowego w dziedzinie wirtualnej walidacji standardu ASAM OSI. Po drugie, na bazie pomiarów z rzeczywistych czujników radarowych zidentyfikowane i wprowadzone będą do modelu modyfikacje mające na celu zwiększenie rzetelności procesu generacji sztucznych pomiarów w ważnych z perspektywy systemów typu ADAS scenariuszach jazdy. Po trzecie, przeprowadzona zostanie procedura identyfikacji parametrów modelu. W końcu, z wykorzystaniem autorskiej metody weryfikacji, udowodniona będzie zarówno wiarygodność zmodyfikowanego modelu, jak i jego zdolność do pracy w czasie rzeczywistym. Co za tym idzie, uzasadnione zostanie, iż model matematyczny czujnika radarowego bazujący na koncepcji centrów rozpraszania może być użyteczny w kontekście testów systemu klasy ADAS w czasie rzeczywistym w symulatorze jazdy, niezależnie od poziomu skomplikowania wirtualnego scenariusza.

Słowa kluczowe: radar, model matematyczny, symulacja, wirtualna walidacja

Abstract

Advanced Driver Assistance Systems (ADAS) play a key role in keeping road users safe. These applications analyze in real time the car surroundings based on sensor data and, if necessary, take the control of the automobile. One of the key algorithms in ADAS domain is Advanced Emergency Braking System (AEBS), with the goal to avoid potentially lethal accidents, e.g. in case of driver distraction or in bad weather conditions. AEBSs are usually using data provided by radar (Radio Detection and Ranging), including range, relative velocity and angle in the horizontal plane, estimated for all objects visible from the sensor perspective.

ADAS validation process is an extremely important activity. In particular, it is crucial to carefully assess the system behaviour in corner-case scenarios. To achieve this goal usually a set of test drives is conducted, where data from sensors is collected and ADAS key decisions are judged. Performing such campaign entirely on real roads is both time-consuming and expensive. That is why, automotive companies often decide to execute some of the experiments in simulation. Thanks to this approach, called virtual validation, it is possible to shorten verification time and to check the system under extremely dangerous conditions, which would be impossible to test in reality.

Virtual validation must be reliable, thus simulation environment is required to be realistic. Due to this the simulator needs to acquire proper sensor models that faithfully emulate real measurements. As mentioned above, automotive radars are crucial in modern ADAS. Unfortunately, these sensors provide highly nondeterministic data, which is mainly caused by the randomness of the electromagnetic wave propagation process. As a result, development of a radar sensor model is a nontrivial task. The key is not only realism of the emulation itself, but also low computational complexity of the algorithm, in order to comply with the real-time requirement of ADAS tests.

Many cutting-edge solutions in the radar sensor modeling area already exist in the literature. However, none of these models is both versatile and simple to develop and tune. In this thesis a recently published, novel radar simulation based on the scattering-centers concept is presented. The model, however, will be adapted to the requirements related to ADAS virtual validation domain. First, the model will be formalized to be compliant with the ASAM OSI standard. Second, a set of modifications identified using real radar measurements will be introduced to the original model, with the goal to increase

the reliability of the synthetic measurements generation process, in crucial from ADAS perspective scenarios. Third, the fine-tuning of the model parameters will be conducted. Finally, with the help of the state-of-the-art verification method, the sensor model fidelity and its real-time capabilities will be proven. In other words, a clear justification is stated, that a scattering-centers based radar sensor model can be a robust and useful tool for testing ADAS in real-time simulations, regardless of scenario complexity.

Keywords: radar, mathematical model, simulation, virtual validation,

Spis treści

Lista skrótów	17
Notacja	19
1 Wprowadzenie	21
1.1 Sformułowanie problemu	21
1.2 Cel i teza pracy	23
1.3 Struktura pracy	24
2 Czujniki radarowe w przemyśle motoryzacyjnym	27
2.1 Typy czujników radarowych.....	27
2.2 Rozwój technologii radiowej.....	28
2.2.1 Pierwsze eksperymenty.....	28
2.2.2 Zastosowania militarne jako główny czynnik postępu	28
2.2.3 Ewolucja w kontekście przemysłu motoryzacyjnego	30
2.3 Podstawy działania radarów samochodowych	31
2.3.1 Pomiar wartości	31
2.3.2 Równanie radarowe.....	35
2.3.3 Uzyskanie zbioru wykryć	36
2.4 Radary w zaawansowanych systemach wspomagania kierowcy.....	40
2.4.1 Procedura wykrywania obiektów.....	40
2.4.2 Przykładowy przepływ informacji w układach klasy ADAS	40
2.4.3 Wirtualna walidacja	43
3 Kompatybilność modelu matematycznego ze standardami przemysłu motoryzacyjnego	49
3.1 Komitet standaryzacyjny ASAM.....	49
3.2 Standaryzacja w kontekście wirtualnej walidacji.....	51
3.3 Wymiana danych symulacyjnych z wykorzystaniem standardu OSI	51
3.3.1 Zarys problemu	51
3.3.2 Struktura <i>OSI::SensorView</i>	52

3.3.3	Struktura <i>OSI::SensorData</i>	55
3.3.4	Modularyzacja symulacji	59
4	Modele matematyczne czujników radarowych	63
4.1	Modelowanie na poziomie wykryć	63
4.1.1	Modele bazujące na danych	63
4.1.2	Modele geometryczne	64
4.2	Modelowanie na poziomie listy obiektów	67
4.2.1	Modele bazujące na danych	67
4.2.2	Modele geometryczne	68
5	Formalizacja referencyjnego modelu matematycznego czujnika radarowego pod kątem standardu ASAM OSI	69
5.1	Założenia modelu referencyjnego	69
5.2	Wejście do modelu zgodne ze standardem ASAM OSI	70
5.3	Generacja zbioru centrów rozpraszania	73
5.4	Przekształcenie centrów rozpraszania do układu współrzędnych sensora	77
5.5	Definicja zbioru sztucznych wykryć radarowych	81
5.5.1	Wyliczenie wartości dla parametrów pojedynczego wykrycia radarowego	81
5.5.2	Likwidacja pomiarów niezwróconych w kierunku wirtualnego czujnika	82
5.5.3	Odrzucenie pomiarów będących poza polem widzenia czujnika	84
5.5.4	Eliminacja pomiarów przysłoniętych przez pozostałe obiekty obecne w scenie	85
5.5.5	Usunięcie pomiarów na podstawie testu statystycznego i dodanie zakłóceń	88
5.6	Wyjście z modelu zgodne ze standardem ASAM OSI	90
5.7	Wstępna ocena wiarygodności modelu referencyjnego	91
6	Usprawnienie modelu bazowego	99
6.1	Filtr rozdzielczości	99
6.2	Generacja wykryć dla obiektów statycznych	105
6.3	Generacja fałszywych wykryć	110
6.4	Wstępna ocena wiarygodności modelu rozszerzonego	111
7	Dobór nastaw modelu	113
7.1	Parametry modelu sensora	113
7.2	Dane	114
7.3	Wskaźnik jakości	115

7.4	Procedura optymalizacji	118
7.5	Rezultaty	119
8	Eksperymenty weryfikacyjne.....	125
8.1	Ocena wydajności.....	125
8.1.1	Informacja na temat kodu źródłowego.....	125
8.1.2	Procedura oceny wydajności modelu.....	126
8.1.3	Rezultaty	128
8.2	Ocena wiarygodności	128
8.2.1	Dane wejściowe	129
8.2.2	Wskaźniki jakości	133
8.2.3	Procedura oceny wiarygodności	136
8.2.4	Rezultaty	138
9	Podsumowanie	145
	Literatura.....	149
	Wykaz rysunków	163
	Wykaz tabel	169

Lista skrótów

W tabeli zaprezentowanej poniżej wyjaśniono znaczenie najważniejszych skrótów wykorzystywanych w pracy.

Skrót	Znaczenie
ACC	aktywny tempomat (ang. Adaptive Cruise Control)
ADAS	zaawansowany system wspomagania kierowcy (ang. Advanced Driver Assistance System)
AEBS	zaawansowany system hamowania awaryjnego (ang. Advanced Emergency Braking System, AEBS)
RDC	trójwymiarowa, zespolona macierz danych radarowych (ang. radar data cube)
CHIRP	skompresowany impuls radarowy o wysokiej intensywności (ang. Compressed High Intensity Radar Pulse)
FOV	pole widzenia czujnika (ang. field of view)
GSM	uniwersalny model czujnika (ang. generic sensor model)
HIL	symulacja w pętli sprzętowej (ang. hardware in the loop)
RCS	skuteczna powierzchnia odbicia radaru (ang. radar cross section)
radar FMCW	radar z falą ciągłą z modulacją częstotliwości (ang. frequency-modulated continuous-wave radar)
SCS	układ współrzędnych czujnika (ang. sensor coordinate systems)
SIL	symulacja w pętli oprogramowania (ang. software in the loop)
SNR	stosunek sygnału do szumu (ang. signal-to-noise ratio)
VCS	układ współrzędnych samochodu (ang. vehicle coordinate systems)

Notacja

W tabeli zaprezentowanej poniżej wyjaśniono znaczenie najważniejszych oznaczeń matematycznych wykorzystywanych w pracy.

Notacja	Znaczenie
$[a, b]_{\mathbb{N}}$	przedział określony na zbiorze liczb naturalnych: $a, b \in \mathbb{N} : a \leq b$
$[a, b]$	przedział określony na zbiorze liczb rzeczywistych: $a, b \in \mathbb{R} : a \leq b$
$[x, y, z]^T$	transpozycja wektora o współrzędnych: $(x, y, z) \in \mathbb{R}$
p_A, P_A	dolna adnotacja wektora $p_A \in \mathbb{R}^n$ oraz zbioru $P_A \in \mathbb{R}^{n \times m}$
p^B, P^B	górna adnotacja wektora $p^B \in \mathbb{R}^n$ oraz zbioru $P^B \in \mathbb{R}^{n \times m}$
p_A^B, P_A^B	dolna i górna adnotacja wektora $p_A^B \in \mathbb{R}^n$ oraz zbioru $P_A^B \in \mathbb{R}^{n \times m}$
$p_A^{B i, j}$	wektor p_A^B określony dla ustalonych i oraz j
$\{p_1, p_2, p_3\}$	konkatenacja wektorów: $\forall (i, j, k) \in \mathbb{N}, p_1 \in \mathbb{R}^i, p_2 \in \mathbb{R}^j, p_3 \in \mathbb{R}^k :$ $\{p_1, p_2, p_3\} \in \mathbb{R}^{i+j+k}$
$\{p_1, p_2, p_3\}^{A, B}$	podwójna górna adnotacja wektora będącego rezultatem konkatenacji wektorów p_1, p_2 oraz p_3
$\Theta_A^{B^{-1}}$	macierz odwrotna do macierzy $\Theta_A^B \in \mathbb{R}^{3 \times 3}$
$\Theta_A^{B^T}$	transpozycja macierzy $\Theta_A^B \in \mathbb{R}^{3 \times 3}$
$\left\lfloor \frac{x}{y} \right\rfloor$	największa liczba całkowita nie większa od ilorazu x/y

1 Wprowadzenie

1.1 Sformułowanie problemu

W 1999 roku koncern samochodowy Mercedes-Benz zaprezentował przełomowy model W220, czyli pierwszy seryjnie produkowany pojazd, który wyposażono w czujnik radarowy [1]. Radar (ang. Radio Detection and Ranging) zamontowano na potrzeby systemu o nazwie Distronic, realizującego funkcjonalność tzw. aktywnego tempomatu (ang. Adaptive Cruise Control, ACC). Opracowanie automatycznej regulacji prędkości zapoczątkowało gwałtowny rozwój zaawansowanych systemów wspomagania kierowcy (ang. Advance Driver Assistance Systems, ADAS). W szczególności zaczęto pracować nad zaawansowanym systemem hamowania awaryjnego (ang. Advanced Emergency Braking System, AEBS) [2, 3], gdzie również kluczową rolę odgrywa czujnik radarowy [4]. Układ typu AEBS okazał się tak ważny dla zapewnienia bezpieczeństwa w ruchu drogowym, że Unia Europejska rozpoczęła pracę nad sporządzeniem norm homologacyjnych dla pojazdów silnikowych. I tak w 2009 roku wydano rozporządzenie nakładające obowiązek montażu tego typu urządzeń w pojazdach ciężarowych [5]. W roku 2019 normę rozszerzono również o samochody osobowe [6]. Dodatkowo do listy układów typu ADAS, które muszą być uwzględnione w każdym nowo wyprodukowanym pojeździe od 2022 roku, zaliczono także urządzenia oparte na czujnikach wizyjnych, takie jak: inteligentny asystent kontroli prędkości (ang. Intelligent Speed Assistance, ISA) czy system awaryjnego utrzymania pojazdu w pasie ruchu (ang. Emergency Lane-Keeping System, ELKS) [6]. Wynika z tego, że czujniki radarowe (oraz kamery) to obecnie nie tylko domena modeli luksusowych. Sensory te stały się nieodłącznym elementem każdego pojazdu silnikowego.

Regulacje nałożone przez Unię Europejską na przemysł motoryzacyjny są ewidentnym potwierdzeniem trendu we współczesnych układach typu ADAS. Mianowicie, tego typu urządzenia przyjmują coraz większą odpowiedzialność za bezpieczeństwo pasażerów pojazdu oraz wszystkich uczestników ruchu drogowego, przejmując kontrolę nad maszyną w krytycznych sytuacjach. Rosnący wpływ systemów należących do kategorii ADAS na trajektorię jazdy można uzasadnić faktem, iż znaczna część wypadków spowodowana jest przez nieuwagę kierowcy lub niedostosowaniem się do warunków panujących na jezdni [7].

Tak wysoki poziom ingerencji urządzeń klasy ADAS w ruch pojazdu skutkuje tym, że przed wdrożeniem do produkcji każdy tego typu system poddany zostaje rzetelnej weryfikacji. Procedura oceny jakości i niezawodności zaprojektowanego rozwiązania jest niezwykle czasochłonna i kosztowna. Wymaga ona przeprowadzenia szeregu jazd testowych, by sprawdzić zachowanie układu w potencjalnie niebezpiecznych sytuacjach [8]. Dlatego też, chcąc uprościć walidację systemu, producenci samochodów decydują się na przeprowadzenie części eksperymentów w symulatorach jazdy [9, 10]. Dzięki takiej wirtualnej walidacji proces oceny układu klasy ADAS jest zdecydowanie mniej czasochłonny. Dodatkowo można także sprawdzić działanie rozwiązania w skrajnie groźnych sytuacjach, które - ze względów bezpieczeństwa - byłoby niezwykle trudno odtworzyć w rzeczywistości.

Do rzetelnego przetestowania urządzenia na sztucznie wygenerowanych danych konieczne jest, by symulator wiernie odtwarzał rzeczywiste scenariusze jazdy [11]. Oznacza to, że mechanizm renderujący środowiska wirtualnego musi szczegółowo powielać kontury wszystkich elementów sceny oraz odwzorowywać różne warunki atmosferyczne. Ponadto symulator musi uwzględniać dokładne modele fizyki ruchu obiektów. Jednak, co najważniejsze, do uruchomienia układu typu ADAS w wirtualnej rzeczywistości wymagany jest moduł odpowiedzialny za generację danych imitujących informacje zwracane przez rzeczywiste czujniki. Wydaje się, że jest to szczególnie skomplikowane w przypadku radaru, czyli kluczowego czujnika w kontekście systemów ACC i AEBS. Za szczególnie kłopotliwe należy uznać zamodelowanie procesu wiernie naśladowującego propagację sygnału elektromagnetycznego w przestrzeni. Trudność ta wynika przede wszystkim z poziomu skomplikowania procesu rozprzestrzeniania się impulsów radiowych w ośrodkach o gęstym rozmieszczeniu obiektów. W tego typu środowiskach pomiary radarowe są zakłócanie głównie przez zjawisko wielokrotnego załamania fali, czego skutkiem jest pojawienie się fałszywych wykryć w zbiorze danych radarowych. Dodatkowo należy uwzględnić także tzw. efekt mikro Dopplera, czyli błędny odczyt prędkości względnej spowodowany przez obracające się koła pojazdu [12]. Jednak warunkiem, który musi spełnić model matematyczny czujnika radarowego, jest nie tylko wysoki poziom realizmu sztucznie generowanych pomiarów. Kluczowym kryterium jest także niska złożoność obliczeniowa modułu umożliwiająca przetestowanie urządzenia typu ADAS z wykorzystaniem techniki tzw. symulacji w pętli sprzętowej (ang. *hardware in the loop*, HIL), gdzie system weryfikuje się w czasie rzeczywistym.

Modele czujników radarowych dzielą się na dwie kategorie: są to modele bazujące na danych oraz modele geometryczne [13]. Do pierwszej kategorii przypisuje się modele wytwarzające wirtualne dane w sposób niejawni, przez odpowiednio zdefiniowany model statystyczny lub sztuczną sieć neuronową. W ostatnich latach powstało wiele tego typu przełomowych algorytmów w takich dziedzinach, jak np.: klasyfikacja obrazu [14, 15] czy wykrywanie obiektów [16–19]. Osiągnięte rezultaty pokazują, że przy użyciu tej metodologii, określanej często mianem techniki tzw. czarnej skrzynki, można odwzorować złożone, nieliniowe operacje. Niestety, biorąc pod uwagę stopień skomplikowania

samochodowych systemów radarowych, do opracowania rzetelnego i wszechstronnego algorytmu bazującego na metodzie czarnej skrzynki wymagana jest grupa ucząca zawierająca ogromną liczbę należycie zróżnicowanych, oznaczonych informacji. Innymi słowy, zaprojektowanie modelu odpornego na problem tzw. nadmiernego dopasowania (ang. overfitting) to niesłychanie trudny i kosztowny proces.

Do drugiej kategorii należy zaliczyć wszystkie rozwiązania, w których syntetyczne pomiary generowane są na podstawie informacji udostępnionych przez symulator jazdy poprzez z góry ustalone funkcje matematyczne. Takie podejście jest praktycznie niezależne od danych rzeczywistych, gdyż do identyfikacji parametrów modelu jest konieczna tylko niewielka liczba pomiarów. Najdokładniejszy będzie tzw. model fizyczny, gdzie uwzględnia się każdy aspekt funkcjonowania rzeczywistego sensora, w tym szczegółowe odwzorowanie procesu propagacji fali elektromagnetycznej. Wadą tej metody jest natomiast spora złożoność obliczeniowa, wynikająca z konieczności wykonania procedury śledzenia promieni (ang. ray tracing), dzięki któremu model ma dostęp do szczegółowych danych o konturach obiektów wygenerowanych przez środowisko wirtualnej jazdy.

Geometryczny model czujnika radarowego może być jednak zrealizowany przy użyciu metody wymagającej zdecydowanie mniejszej ilości zasobów obliczeniowych, gdzie nie jest konieczna bezpośrednia symulacja propagacji sygnału radiowego w przestrzeni. Metodologia ta opiera się na koncepcji tzw. centrów rozpraszania znanej ze studiów nad skuteczną powierzchnią odbicia radaru (ang. radar cross section, RCS). Idea ta mówi, że odbicie (rozproszenie) fali elektromagnetycznej od dużego obiektu może być przybliżone za pomocą pewnego zbioru punktów zlokalizowanych na tym obiekcie [20]. Innymi słowy, do wygenerowania sztucznych pomiarów radarowych wykorzystuje się jedynie informacje o obiektach zdefiniowanych w danym scenariuszu wirtualnej jazdy; istotne są tutaj następujące parametry: pozycja, wymiary, kinematyka czy też typ konkretnego celu. Oczywiście takie podejście jest sporym uproszczeniem w odniesieniu do rzetelnie opracowanego modelu fizycznego. Okazuje się jednak, że uzupełniając symulację bazującą na centrach rozpraszania o dodatkowy generator zakłóceń podejście to może być wystarczające do uzyskania sztucznego zbioru wykryć radarowych o podobnych właściwościach stochastycznych w stosunku do rzeczywistych pomiarów [21].

1.2 Cel i teza pracy

Zdefiniujmy wpięrcw pojęcia, przy użyciu których można scharakteryzować model matematyczny czujnika:

- (i) Wiarygodność - model matematyczny uznaje się za wiarygodny, kiedy wartości wyliczone dla przyjętych wskaźników jakości mieszczą się w ustalonym przedziale błędu, niezależnie od poziomu skomplikowania i charakterystyki scenariusza wirtualnej jazdy, w którym uruchomiono dany model.

- (ii) Wydajność - model matematyczny uznaje się za wydajny, jeśli złożoność obliczeniowa modelu pozwala na uruchomienie symulacji (układu klasy ADAS) w czasie rzeczywistym dla scenariuszy o wysokim stopniu natężenia ruchu.
- (iii) Użyteczność - model matematyczny jest użyteczny w kontekście wirtualnej walidacji systemów typu ADAS wtedy, gdy jest on wiarygodny i wydajny.

W literaturze opisano do tej pory wiele skutecznych rozwiązań, opartych na wszystkich omówionych w poprzednim podrozdziale metodologiach. Analizując dostępne prace, nietrudno wskazać w każdej z omówionych technik co najmniej jeden istotny problem:

- (i) Problem czarnej skrzynki - wysoka zależność od pomiarów z rzeczywistego czujnika: zaprojektowanie oraz wyuczenie sztucznej sieci neuronowej generującej syntetyczny zbiór wykryć radarowych cechujący się wiarygodnością wymaga zgromadzenia niezwykle obszernego zbioru uczącego, co jest niesłychanie czasochłonnym i kosztownym procesem.
- (ii) Problem modeli fizycznych - duża złożoność obliczeniowa: wiarygodne odwzorowanie procesu propagacji fali elektromagnetycznej w przestrzeni w symulatorze jazdy wymaga dostępu do szczegółowych informacji na temat konturów obiektów zdefiniowanych dla danego momentu scenariusza, co pociąga za sobą konieczność wykonania kosztownej pod kątem zużycia zasobów komputera operacji śledzenia promieni.
- (iii) Problem modeli bazujących na technice centrów rozpraszania - trudność w uzyskaniu wysokiego stopnia wiarygodności: pominięcie procesu propagacji sygnału elektromagnetycznego w przestrzeni powoduje, że model matematyczny nie zawiera odpowiedniej ilości zakłóceń charakterystycznych dla rzeczywistego czujnika radarowego.

Celem pracy jest adaptacja innowacyjnego i skutecznego modelu matematycznego czujnika radarowego, zaproponowanego w [21], pod kątem wymagań przemysłu motoryzacyjnego dotyczących testów systemów typu ADAS w środowiskach wirtualnej jazdy. Innymi słowy, zadanie będzie polegało na zidentyfikowaniu oraz wprowadzeniu do oryginalnego modelu szeregu modyfikacji, przy użyciu nieznacznej ilości danych rzeczywistych, tak, by model charakteryzował się wysokim poziomem wiarygodności i wydajności.

Teza pracy jest z kolei następująca:

Możliwa jest adaptacja sztucznego zbioru danych wygenerowanego przy użyciu modelu matematycznego czujnika radarowego bazującego na koncepcji centrów rozpraszania pod kątem użyteczności w kontekście testów systemu klasy ADAS w czasie rzeczywistym w symulatorze jazdy.

1.3 Struktura pracy

Rozprawę podzielono na kilka logicznie połączonych ze sobą części. W rozdziale 2 omówiono podstawowe zagadnienia dotyczące samochodowych systemów radarowych oraz kluczowe kwestie

związane z weryfikacją układów klasy ADAS w środowiskach wirtualnej jazdy. Rozdział 3 to wprowadzenie do tematyki standardów przemysłu motoryzacyjnego, które każdy komponent systemu typu ADAS musi spełniać. Rozdział 4 jest przeglądem dostępnych w literaturze rozwiązań w dziedzinie modelowania czujników radarowych. W rozdziale 5 zaprezentowano sformalizowany opis modelu matematycznego czujnika radarowego, opublikowanego w [21], na którym oparte są najważniejsze rozważania zawarte w tej pracy; przy czym formalizację definicji przeprowadzono pod kątem standardu omówionego w rozdziale 3. Rozdział 6 ma na celu wprowadzenie do bazowego modelu modyfikacji zidentyfikowanych na podstawie pomiarów z rzeczywistego sensora, zwiększających wiarygodność sztucznych pomiarów radarowych. W rozdziale 7 przeprowadzono proces identyfikacji parametrów modelu. Natomiast w rozdziale 8 udowodniono wiarygodność i wydajność zmodyfikowanego modelu sensora. W końcu, rozdział 9 jest podsumowaniem rozważań zaprezentowanych w tej rozprawie.

2 Czujniki radarowe w przemyśle motoryzacyjnym

Radar jest systemem generującym ciąg impulsów elektromagnetycznych w celu uzyskania informacji o obiektach, które znajdują się w otoczeniu czujnika. Pierwsze tego typu urządzenia zostały zaprojektowane na potrzeby określania odległości do danego celu; stąd też wywodzi się akronim radar - wykrywanie przedmiotów z wykorzystaniem fal radiowych (ang. Radio Detection and Ranging). Historycznie radar można zaliczyć do najważniejszych wynalazków XX wieku. Możliwość namierzania obiektów zarówno przy dobrych warunkach atmosferycznych, jak i w przypadku skrajnej nieprzejrzystości powietrza, okazała się kluczowa w zastosowaniach militarnych oraz cywilnych. Do przykładowych zastosowań można zaliczyć: wojskowe systemy wczesnego ostrzegania, systemy kontroli lotów czy stacje meteorologiczne. Co jednak najważniejsze z perspektywy tej rozprawy, radar stał się nieodłączną częścią urządzeń typu ADAS. Na początku tego rozdziału zostanie wprowadzony podział systemów radarowych na kategorie oraz przedstawiona będzie krótka historia tejże technologii - od zastosowań militarnych do użytku na potrzeby przemysłu motoryzacyjnego. Dodatkowo, w tej części pracy autor wyjaśni kluczowe aspekty dotyczące działania samochodowych czujników radarowych. Ponadto omówione zostaną podstawowe zagadnienia związane z systemami klasy ADAS wraz z ich weryfikacją.

2.1 Typy czujników radarowych

Czujniki radarowe różnią się wielkością oraz mocą i charakterystyką generowanego sygnału. Biorąc pod uwagę rodzaj emitowanej fali radiowej można zdefiniować trzy główne rodzaje aparatur: radar impulsowy, radar z falą ciągłą, jak również radar z falą ciągłą z modulacją częstotliwości (ang. frequency-modulated continuous-wave radar, FMCW radar) [22]. Radary impulsowe są zwykle urządzeniami dalekiego zasięgu, co jest pożądane w kontekście zastosowań militarnych, meteorologicznych, jak i dla zapewnienia bezpieczeństwa w przestrzeni powietrznej (kontrola lotów). Aparatury tego typu służą głównie do pomiaru odległości, która jest obliczana bezpośrednio z różnicy pomiędzy nadanym a odebranym sygnałem. Niestety, ze względu na fakt, iż impulsy generowane są w sposób dyskretny brak tutaj możliwości oszacowania, z jaką szybkością porusza się dany cel, gdyż nie są dostępne informacje o wysokiej rozdzielczości na temat zmienności zarejestrowanego echa. Pomiar prędkości względnej

obiekty umożliwia natomiast radar z falą ciągłą, wykorzystując do tego celu efekt Dopplera. Skutkiem niezmienności i ciągłości generowanego sygnału jest jednak brak zdolności do mierzenia dystansu. Radar FMCW łączy zalety obu wspomnianych typów, umożliwiając jednocześnie pomiar odległości, prędkości oraz kąta. Warto zaznaczyć, że technologie radiowe różnią się znacznie między sobą. Dlatego też kompleksowe wyjaśnienie działania radarów impulsowych i radarów z falą ciągłą wykracza poza zakres tej pracy. Autor skupi się jedynie na omówieniu podstawowych zagadnień dotyczących urządzeń typu FMCW, wykorzystywanych we współczesnych rozwiązaniach projektowanych dla przemysłu motoryzacyjnego.

2.2 Rozwój technologii radiowej

2.2.1 Pierwsze eksperymenty

Pierwsze eksperymenty potwierdzające istnienie fal elektromagnetycznych przeprowadził w roku 1886 Heinrich Hertz. Potwierdził on nie tylko teorię zaproponowaną 22 lata wcześniej przez Maxwella, lecz odkrył także, że fale radiowe mają zdolność do odbijania się od metalowych przedmiotów oraz od dielektryków. Dodatkowo pokazał, iż prędkości rozchodzenia się badanych fal i fal świetlnych są identyczne.

Pierwszy patent opisujący urządzenie służące do wykrywania obiektów przy użyciu fal elektromagnetycznych zgłosił w 1904 roku niemiecki inżynier Christian Hülsmeier [23]. Niestety, w tamtym czasie nie dostrzeżono potencjału tego typu rozwiązań w kontekście namierzania celów w warunkach skrajnej nieprzejrystości powietrza. Prawdziwy przełom pojawił się dopiero w latach 30. XX wieku. Mianowicie, w 1934 roku Albert H. Taylor, Leo C. Young i Lawrence A. Hyland z laboratorium badawczego Marynarki Wojennej Stanów Zjednoczonych (ang. Naval Research Laboratory, NRL) opatentowali i skonstruowali system radarowy zdolny wykryć samolot czy też inny poruszający się obiekt z odległości kilkudziesięciu kilometrów [24].

2.2.2 Zastosowania militarne jako główny czynnik postępu

Pierwszy radar zaproponowany przez NRL można zakwalifikować do kategorii radarów z falą ciągłą. Urządzenie to miało jednak stosunkowo ograniczone możliwości - potrafiło jedynie zarejestrować obiekt bez dokładnego określenia jego pozycji. Głównym powodem była tutaj charakterystyka nadawania fali elektromagnetycznej oraz sposób rejestrowania odebranego sygnału - tj. sygnał radiowy był emitowany w sposób ciągły, natomiast obecność celu określano na podstawie wykrycia interferencji fali odbitej od obiektu z aktualnie generowanym sygnałem.

Członkowie NRL zaczęli więc przeprowadzać eksperymenty z radarem impulsowym. Urządzenie to składało się z części typowych dla współczesnych systemów pulsacyjnych, czyli z nadajnika, odbiornika

oraz pojedynczej anteny. Zadaniem nadajnika jest generowanie sekwencji impulsów o zadany czasie trwania stanu wysokiego (szerokość impulsu - ang. pulse width) i stałym okresie pomiędzy kolejnymi aktywacjami (ang. pulse repetition frequency). Sygnał ten jest przekazany do anteny, która emituje falę radiową w przestrzeni. Z kolei odbiornik ma na celu zabsorbowanie echa nadanego sygnału. Taka konstrukcja, w przeciwieństwie do radarów z falą ciągłą, umożliwia jednoznaczny pomiar odległości poprzez znajomość częstotliwości generowanej fali oraz przerwy w emitowaniu promieniowania. Po kilku latach testów udało się wyprodukować system nazwany CXAM. Marynarka Wojenna Stanów Zjednoczonych z sukcesem wykorzystwała te radary w trakcie II wojny światowej, montując je na 19 najważniejszych okrętach swojej floty [22]. Radar impulsowy CXAM osiągnął maksymalny zasięg na poziomie około 20 kilometrów przy mocy sygnału rzędu 6 kW i częstotliwości pracy równej 200 MHz. Oprócz NRL pracę nad radarem prowadził także Korpus Sygnałowy Armii Stanów Zjednoczonych (ang. United States Army Signal Corps, USASC). USASC rozpoczął eksperymenty w 1936 roku, prawdopodobnie po ówczesnej wizycie w NRL. Opracowano tam system o nazwie SCR-268. Służył on do sterowania baterią przeciwlotniczą we współpracy z reflektorem kierunkowym; konieczność użycia dodatkowego oświetlenia naprowadzającego była spowodowana słabą dokładnością określania kąta [22]. Radar SCR-268 był wykorzystywany w latach 1938-1944, później został zastąpiony przez ulepszony model SCR-584, w którym nie trzeba już było używać dodatkowego oświetlenia. Armia USA zaprojektowała także w roku 1939 radar dalekiego zasięgu SCR-270 na potrzeby systemów wczesnego ostrzegania.

Równolegle ze Stanami Zjednoczonymi badania nad czujnikami radarowymi były prowadzone w Wielkiej Brytanii, choć rozpoczęły się nieco później, bo w 1935 roku [25–27]; pracę nadzorował sir Robert Watson-Watt. Postęp technologiczny następował błyskawicznie. Już w lutym tego samego roku udało się wykryć samolot z odległości rzędu 10 kilometrów przy użyciu sprzętu do komunikacji naziemnej - wykorzystana technika była podobna do techniki stosowanej w trakcie pierwszych eksperymentów z falami radiowymi przeprowadzanymi w Stanach Zjednoczonych (technologia z falą ciągłą). Bardzo szybko jednak Brytyjczykom udało się opracować radar impulsowy. Mianowicie, w czerwcu 1935 roku zademonstrowali oni technikę do pomiaru odległości do lecącego samolotu (dokonali tego przełomu rok wcześniej, niż członkowie NRL). We wrześniu tego samego roku system był zdolny wykryć obiekt latający z odległości kilkudziesięciu kilometrów. W końcu, w marcu 1936 roku, zasięg radaru wzrósł do około 150 kilometrów, przy częstotliwości pracy równej 25 MHz. Co ciekawe, radar ten umożliwiał dodatkowo pomiar wysokości lotu maszyny. Technologia opracowana przez zespół Watsona-Watta szybko została wdrożona do systemów obronnych brytyjskiej armii. W kwietniu 1937 roku zademonstrowano szereg naziemnych stacji radarowych, natomiast rok później radiostacje pracowały już przez całą dobę, aż do końca II wojny światowej. Warto nadmienić, że poza systemami naziemnymi

Brytyjczycy zaprojektowali też radar montowany na samolotach bojowych w celu namierzania wrogich jednostek.

2.2.3 Ewolucja w kontekście przemysłu motoryzacyjnego

Radary stosowane w przemyśle motoryzacyjnym są układami typu FMCW. Pierwsze pomysły dotyczące systemu radarowego z falą radiową emitowaną w sposób ciągły, z dodatkową modulacją częstotliwości sygnału, pojawiły się w latach 40. i 50. XX wieku [28–30]. Później, tj. w latach 60. i 70., technologia FMCW przybierała formę coraz bliższą współczesnym rozwiązaniom [31–33]. W szczególności w [33] zaproponowano modulację z wykorzystaniem sygnału trójkątnego symetrycznego, dzięki której oszacowanie odległości oraz prędkości względnej jest stosunkowo proste do zrealizowania na układzie mikroprocesorowym.

Z kolei, jak podano w obszernym raporcie [34], początek badań nad wykorzystaniem czujników radarowych w przemyśle motoryzacyjnym przypada na lata 60. i 70. XX wieku. Prace prowadzone były głównie w Stanach Zjednoczonych [35–38], Japonii [39, 40] i Niemczech [41–43]; w badania zaangażowały się takie koncerny, jak: General Motors, Mitsubishi i Nissan. Pionierskie urządzenia projektowano na potrzeby wykrywania przeszkód i awaryjnego hamowania. Niektóre z testów zakończyły się sukcesem, aczkolwiek ówczesny poziom zaawansowania prototypów uniemożliwiał ich wdrożenie do masowej produkcji [34].

Następnie, w latach 80. i 90. XX wieku, technikę FMCW dopasowano do zastosowań przemysłu motoryzacyjnego [44–47]. Pierwszą komercyjną aparaturą (typu FMCW) był radar zaprojektowany w 1995 roku przez firmę VORAD [48]. Pracował on na częstotliwości 24 GHz, generując impulsy o mocy mniejszej niż 2 mW. Osiągnięto zasięg rzędu 125 metrów oraz jednoczesny pomiar odległości, prędkości oraz kąta pomiędzy osią wiodącą radaru (oś wzdłuż której zestaw anten generuje największe wzmocnienie) a obiektem [49]. Sensor ten wykorzystano w systemie ostrzegania przed potencjalną kolizją montowanym w pojazdach ciężarowych oraz autobusach - układ wysyłał jedynie komunikaty dźwiękowe oraz świetlne, nie ingerował w żaden sposób w sterowanie pojazdem. Technologia ta nie zyskała jednak dużej popularności wśród kierowców i szybko została wycofana z produkcji. Dopiero w 1999 roku czujnik radarowy zamontowano w seryjnym samochodzie osobowym, w Mercedesie-Benz W220 [1]. Koncern wykorzystał radar o częstotliwości 77 GHz w przełomowym systemie o nazwie Distronic, czyli pierwszy tzw. aktywny tempomat (ang. Adaptive Cruise Control, ACC) wdrożony do produkcji. Rolą układu Distronic było zwiększanie lub zmniejszanie prędkości pojazdu, zmiana przełożeń skrzyni biegów i, w razie potrzeby, hamowanie - zakres prędkości wynosił od 30 do 180 km h⁻¹, zasięg do 150 m, natomiast maksymalną prędkość hamowania ze względów bezpieczeństwa ustawiono na -2 m s^{-2} [50].

Pojawienie się na rynku systemu Distronic zapoczątkowało dynamiczny rozwój systemów klasy ADAS opartych na radarach FMCW. Zaczęto pracować przede wszystkim nad zmniejszeniem kosztów produkcji radarów FMCW, by tego typu urządzenia nie pozostały jedynie domeną marek luksusowych [34]. Kluczowym momentem było wprowadzenie układu scalonego zdolnego do obsługi wielokanałowej transmisji radiowej [51]. Dzięki temu wielowiązkowe systemy radarowe [45] mogły zostać wykonane stosunkowo niewielkim kosztem. Zwiększyło to zdecydowanie potencjał systemów FMCW, gdyż użycie wielu anten nadawczych i odbiorczych z dodatkowym zastosowaniem procedury cyfrowego kształtowania wiązki [52, 53] powoduje, że znacznie powiększa się pole widzenia (ang. field-of-view, FOV) czujnika oraz istotnie poprawia się jego dokładność.

Konsekwencją tych prac są normy homologacyjne dla pojazdów silnikowych wprowadzone przez Unię Europejską odpowiednio w 2009 [5] i 2019 [6] roku. W szczególności norma z 2019 roku nakłada obowiązek montażu w każdym nowo wyprodukowanym pojeździe zaawansowanego systemu hamowania awaryjnego (ang. Advanced Emergency Braking System, AEBS), bazującego na czujnikach radarowych. Pokazuje to, jak istotną rolę pełnią radary we współczesnych urządzeniach typu ADAS.

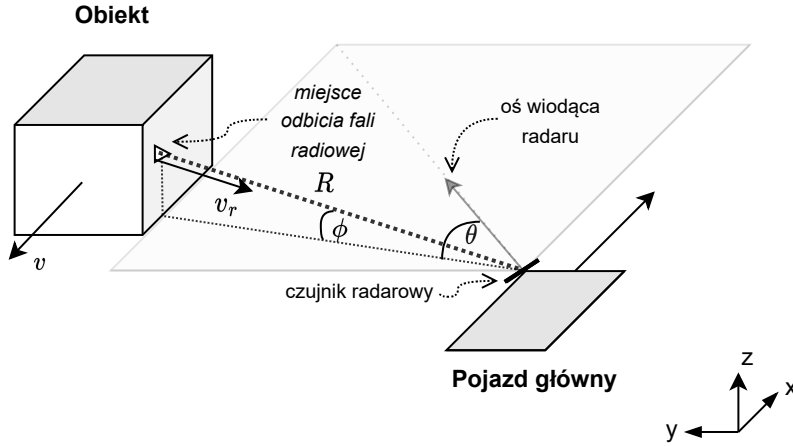
2.3 Podstawy działania radarów samochodowych

2.3.1 Pomiar wartości

Typowy radar FMCW umożliwia jednoczesny pomiar odległości, prędkości względnej oraz kątów w płaszczyźnie horyzontalnej i wertykalnej pomiędzy osią wiodącą radaru (oś wzdłuż której zestaw anten generuje największe wzmocnienie, ang. antenna boresight) a obiektem znajdującym się w jego polu widzenia. Parametry te, powiązane z miejscem na danym obiekcie, od którego odbiła się fala radiowa, jednoznacznie definiują pojedyncze wykrycie radarowe; graficzną reprezentację wspomnianych wielkości pokazano na rysunku 2.1, gdzie:

- R to odległość radialna pomiędzy radarem a miejscem odbicia fali radiowej,
- θ jest kątem w płaszczyźnie horyzontalnej pomiędzy osią wiodącą radaru a obiektem,
- ϕ oznacza kąt w płaszczyźnie wertykalnej pomiędzy osią wiodącą radaru a obiektem,
- v to prędkość bezwzględna obiektu wykrywalnego przez radar,
- v_r reprezentuje prędkość względną danego celu, określaną często mianem prędkości radialnej, rozumianej jako szybkość zmiany odległości R .

Przejdźmy zatem do omówienia szczegółów związanych z uzyskaniem wspomnianych wyżej wartości w urządzeniach FMCW. Radar FMCW generuje falę elektromagnetyczną, której częstotliwość zmienia się w czasie, na podstawie pewnej ustalonej funkcji. W technologii FMCW korzysta się z różnych typów modulacji, na przykład przy użyciu sygnału trójkątnego symetrycznego lub prostokątnego symetrycznego. W ramach tej pracy zakłada się, że system modulowany jest sygnałem piłokształtnym.



Rysunek 2.1. Parametry pojedynczego wykrycia zwracanego przez radar FMCW

W takim przypadku częstotliwość fali elektromagnetycznej rośnie liniowo przez czas T w zadanym paśmie przenoszenia, czyli od wartości minimalnej $f_{\min}$ do wartości maksymalnej $f_{\max}$. W trakcie pojedynczego cyklu emituje się szereg krótkich sygnałów elektromagnetycznych; oznaczmy przez M ilość impulsów w cyklu. Taki pakiet M impulsów określa się mianem skompresowanego impulsu radarowego o wysokiej intensywności (ang. Compressed High Intensity Radar Pulse, CHIRP).

Dla uproszczenia przyjmuje się, że odebrany sygnał (echo) to kopia wyemitowanej fali radiowej [54], opóźniona o czas τ ; τ jest liniowo zależne od odległości R , na której znajduje się dany cel:

$$\tau = \frac{2R}{c}, \quad (2.3.1)$$

gdzie c jest stałą reprezentującą prędkość rozchodzenia się fali elektromagnetycznej w próżni:

$$c = 299792458 \text{ m s}^{-1}. \quad (2.3.2)$$

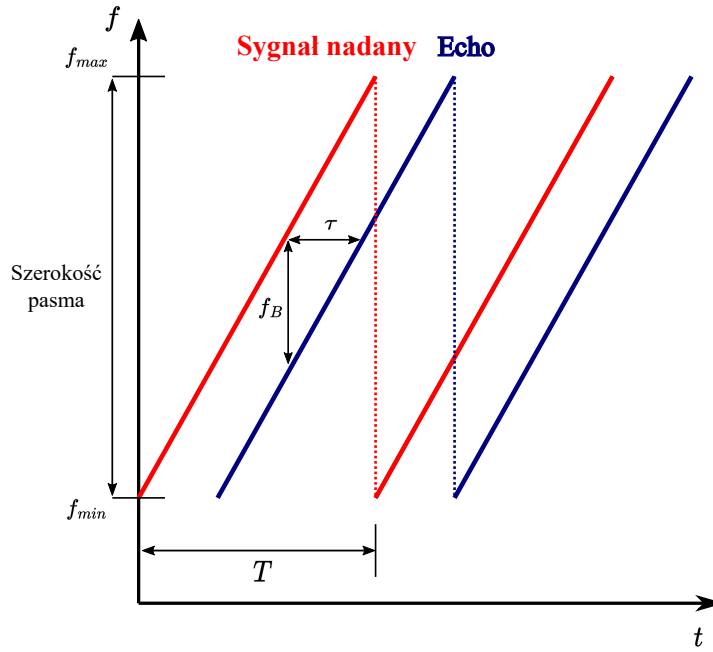
Do wyznaczenia odległości R wykorzystuje się operację miksowania aktualnie nadanego sygnału z odebranym echem, otrzymując w ten sposób różnicowy sygnał sinusoidalny o częstotliwości f_B [22]. Innymi słowy, oszacowanie częstotliwości f_B okazuje się wystarczające do dokładnego określenia dystansu, na którym znajduje się obiekt. Opisane wyżej zależności zaprezentowano na rysunku 2.2.

Jak łatwo zauważyć, f_B jest stała w przedziale $[\tau, T]$ i wynosi:

$$f_B = \alpha \tau, \quad (2.3.3)$$

gdzie α reprezentuje współczynnik kierunkowy powyższej funkcji liniowej:

$$\alpha = \frac{f_{\max} - f_{\min}}{T}. \quad (2.3.4)$$



Rysunek 2.2. Modulacja częstotliwości z wykorzystaniem sygnału piłokształtnego

Ostatecznie, łącząc równania (2.3.1) i (2.3.3), otrzymujemy wzór na odległość R :

$$R = \frac{cf_B}{2\alpha}. \quad (2.3.5)$$

Rozważania zaprezentowane do tej pory zakładały, że obiekt znajdujący się w polu widzenia radaru nie przemieszcza się. Jeśli cel porusza się z prędkością radialną v_r , to zarówno odległość, jak i faza sygnału różnicowego ϕ ulegają ciągłym zmianom [55]. Zmiana ϕ w czasie to pulsacja ω , która z kolei jest bezpośrednio związana z częstotliwością Dopplera f_D [22]. Związek tych trzech wartości z prędkością radialną v_r określa równanie:

$$v_r = \frac{\lambda f_D}{2} = \frac{\lambda \omega}{4\pi} = \frac{\lambda \phi}{4\pi T}, \quad (2.3.6)$$

gdzie:

- ϕ to faza sygnału różnicowego,
- ω jest pulsacją, czyli zmianą ϕ w czasie,
- f_D reprezentuje częstotliwość Dopplera, dodatnią dla obiektu zbliżającego się do czujnika i ujemną w przeciwnym wypadku.
- λ to długość wyemitowanej fali elektromagnetycznej.

Niestety, czas nadawania pojedynczego pakietu (M impulsów) jest za krótki, by analizować zmiany w fazie, gdyż są one zbyt wrażliwe na wahania odległości. Dlatego też pełny okres pomiarowy w układzie FMCW składa się z zestawu zawierającego K pakietów. Okazuje się, że jeśli czas trwania

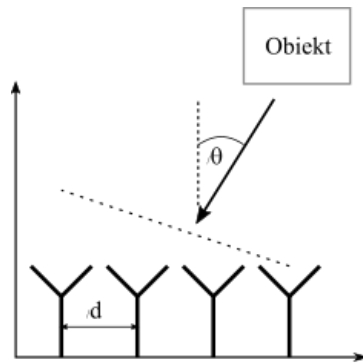
pakietu (T) będzie stosunkowo krótki i horyzont K odpowiednio dobrany, to zmiana częstotliwości f_B w trakcie pełnego okresu pomiarowego będzie niezauważalna z perspektywy systemu radarowego, co jest spowodowane jego skończoną rozdzielczością. Dzięki temu możliwe jest oszacowanie fazy sygnału różnicowego dla całego okresu pomiarowego, co pozwala na dokładne obliczenie prędkości radialnej obiektu.

Korzystając wyłącznie z jednej anteny nadawczej oraz pojedynczej anteny odbiorczej możliwe jest uzyskanie informacji o odległości i prędkości względnej, na podstawie wyemitowania pełnego zestawu pakietów impulsów. Nie jest możliwe, niestety, oszacowanie kąta w płaszczyźnie horyzontalnej pomiędzy danym celem a osią wiodącą radaru, oznaczonego wcześniej grecką literą θ . Dlatego też wykorzystuje się wiele anten odbiorczych, oddzielonych między sobą o stałą odległość d , równą zwykle połowie długości fali λ (ang. phased antenna array) [22]. Przyjmijmy, że w polu widzenia radaru korzystającego z omawianej konfiguracji anten znajduje się pewien obiekt z niezerową wartością kąta θ - sytuację tą pokazano na rysunku 2.3. W takim przypadku sygnał zarejestrowany przez kolejne elementy odbiorcze będzie opóźniony w fazie o kąt ψ , zależny od długości fali λ , odległości d oraz od kąta θ :

$$\psi = 2\pi \frac{d}{\lambda} \sin \theta. \quad (2.3.7)$$

Po odpowiednim przekształceniu równania (2.3.7), uzyskujemy wzór na kąt θ :

$$\theta = \arcsin \frac{\lambda \psi}{2\pi d}. \quad (2.3.8)$$



Rysunek 2.3. Zestaw anten odbiorczych

Teoretycznie do wyznaczenia kąta θ wystarczą dwie anteny odbiorcze. Jednakże taka konfiguracja uniemożliwia rozróżnienie kąta pomiędzy dwoma (lub więcej) celami poruszającymi się z identyczną prędkością oraz na takim samym dystansie. Z tego powodu w zaawansowanych urządzeniach FMCW korzysta się z N elementów odbiorczych ($N > 2$), gdzie dla każdej anteny dostępne są dwuwymiarowe dane - kolejno liczba próbek oraz liczba pakietów - zawierające informacje o odległości i prędkości. Oszacowanie kąta jest analogiczne, jak w przypadku estymacji prędkości. To znaczy, przeprowadza się

analizę zmiany fazy na podstawie N punktów pomiarowych. Warto tutaj zaznaczyć, że jeśli anteny są rozmieszczone w osi poziomej czujnika, to uzyskuje się kąt w płaszczyźnie horyzontalnej (azymut). Natomiast kąt w płaszczyźnie wertykalnej (elewacja) otrzymuje się wówczas, gdy elementy odbiorcze będą rozłożone względem osi pionowej. By umożliwić jednoczesny pomiar zarówno azymutu, jak i elewacji, można albo zamontować dodatkową antenę nadawczą (przesuniętą w pionie względem głównego członu nadawczego), albo rozmieścić elementy odbiorcze w obu płaszczyznach. Jak zostało wspomniane, współczesne radary FMCW obsługują wiele anten nadawczych i odbiorczych poprzez zastosowanie metody cyfrowego kształtowania wiązki [52, 56].

Na tym etapie warto poruszyć zagadnienie, jakim jest rozdzielczość pomiarowa radaru typu FMCW: kolejno ΔR , Δv_r i $\Delta \theta$ dla odległości, prędkości względnej oraz kąta, zdefiniowane w równaniach:

$$\Delta R = \frac{c\Delta f}{2\alpha}, \quad \Delta v_r = \frac{\lambda}{2KT}, \quad \Delta \theta = \frac{2}{N}, \quad (2.3.9a-c)$$

gdzie:

- Δf to jednostkowy przyrost częstotliwości,
- iloczyn K i T wyznacza czas pełnego okresu pomiarowego.

Znaczenie parametrów ΔR , Δv_r oraz $\Delta \theta$ jest takie, że jeśli obiekty znajdują się wewnątrz trójwymiarowej komórki o wymiarze $\Delta R \times \Delta v_r \times \Delta \theta$, to będą traktowane przez radar jako pojedynczy cel. Dodatkowo warto określić maksymalne wartości odległości ($R_{\max}$), prędkości ($v_{\max}$) oraz kąta ($\theta_{\max}$), jakie mogą być zmierzone przez radar:

$$R_{\max} = \frac{Tc}{2}, \quad v_{\max} = \frac{\lambda}{4T}, \quad \theta_{\max} = \frac{\lambda}{2d}. \quad (2.3.10a-c)$$

2.3.2 Równanie radarowe

Ważnym zagadnieniem, o którym należy wspomnieć w kontekście radarów (nie tylko typu FMCW) jest tzw. równanie radarowe [57]. Równanie to określa zależność pomiędzy maksymalnym zasięgiem ($R_{\max}$) i aktualnie ustawionymi parametrami danego systemu radarowego:

$$R^4 = \frac{P_t G_t^2 \lambda^2 \rho G_r}{(4\pi)^3 \chi L_s k T_0 \beta F_n}, \quad (2.3.11)$$

gdzie znaczenie parametrów jest następujące:

- χ : stosunek sygnału do szumu (ang. signal to noise ratio, SNR),
- P_t [W]: moc wyemitowanego impulsu,
- G_t : wzmacnienie anteny nadawczej,
- G_r : wzmacnienie odebranego sygnału,

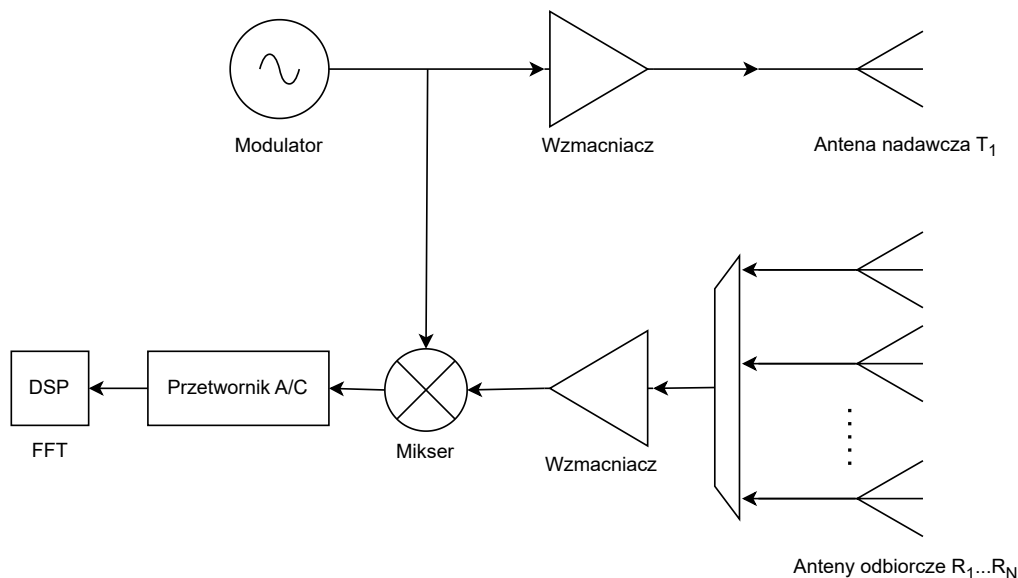
- λ [m]: długość fali,
- ρ [m²]: skuteczna powierzchnia odbicia (ang. radar cross section, RCS),
- L_s : współczynnik strat systemowych,
- $k = 1.38 \times 10^{-23}$ [J K⁻¹]: stała Boltzmanna,
- $T_0 = 290$ [K]: temperatura bezwzględna,
- β [Hz]: pasmo przenoszenia odbiornika (ang. bandwidth),
- F_n [dB]: parametr określający degradację SNR przy przejściu przez odbiornik.

Równanie to można wykorzystać na dwa sposoby:

- do dostrojenia parametrów radaru w celu osiągnięcia pożądanej wartości $R_{\max}$: eksperyment taki należy przeprowadzić w odizolowanym środowisku (w którym poziom SNR jest stały) oraz z obiektem o znanej wartości RCS,
- do oszacowania poziomu SNR dla obiektu o znanej wartości RCS znajdującego się na ustalonym dystansie R : taka procedura może być pomocna w procesie wyznaczenia tzw. adaptacyjnych progów odcięcia [58], które wykorzystuje się do usunięcia ze zbioru wykryć radarowych fałszywych pomiarów.

2.3.3 Uzyskanie zbioru wykryć

Systemy radarowe FMCW zwracają zwykle dla danego okresu pomiarowego zbiór wykryć, gdzie każde jest reprezentowane przez trójkę liczb rzeczywistych - odległość R [m], prędkość radialną v_r [m s⁻¹] oraz kąt θ [deg]. Przykładowy schemat blokowy układu wykorzystującego jedną antenę nadawczą oraz N anten odbiorczych zaprezentowano na rysunku 2.4.



Rysunek 2.4. Uproszczony schemat blokowy radaru FMCW

Do wygenerowania zestawu wykryć na podstawie nadanego oraz odebranego echa radiowego wykorzystuje się zwykle algorytm szybkiej transformacji Fouriera (ang. Fast Fourier Transform, FFT), który realizuje operację dyskretnej transformacji Fouriera (ang. Digital Fourier Transform, DFT). Oznaczmy przez x_i i -tą próbkę sygnału wejściowego, natomiast niech X_m będzie m -tą próbką sygnału stanowiącego wynik operacji FFT ($i, m \in [1, M]$). Elementy X_m są liczbami zespolonymi obliczanymi według następującego wzoru [59]:

$$X_m = \sum_{i=1}^M x_i e^{-j \frac{2\pi i m}{M}}, \quad (2.3.12)$$

gdzie:

- M to ilość impulsów w pojedynczym cyklu modulacji sygnału elektromagnetycznego, dla którego częstotliwość generowanej fali radiowej zmienia się w zakresie od $f_{\min}$ do $f_{\max}$,
- j reprezentuje jednostkę urojoną liczby zespolonej.

Przyjmijmy, że aparatura radarowa wyposażona jest w system mikroprocesorowy składający się z przetwornika analogowo-cyfrowego oraz z procesora sygnałowego (ang. digital signal processor, DSP). Dodatkowo założmy, że układ taktowany jest czasem Δt - w trakcie pojedynczego cyklu mikroprocesora częstotliwość nadawanej fali elektromagnetycznej jest zwiększana o Δf .

Jeden cykl modulacji składa się zatem z M impulsów elektromagnetycznych, z krokowo narastającą częstotliwością (o Δf). Na wyjściu z przetwornika analogowo-cyfrowego uzyskujemy więc M próbek sygnału różnicowego. Następnie, z wykorzystaniem algorytmu FFT, dane przekształcane są do dziedzin częstotliwości. Rezultatem operacji jest zbiór R_{FFT} składający się z M liczb zespolonych, o określonym module oraz argumencie.

Każdy element zbioru R_{FFT} , określane mianem komórki odległości, jest jednoznacznie związany z częstotliwością różnicową f_B , z czego pierwsza komórka odpowiada częstotliwości równej Δf , natomiast ostatnia powiązana jest z wartością równą iloczynowi Δf oraz M . Toteż dla komórki odległości o indeksie $m \in [1, M]$, o niezerowym module liczby zespolonej, parametr wykrycia R może być wyliczony według następującej formuły, będącej modyfikacją wzoru (2.3.5):

$$R = \frac{cf_B}{2\alpha} = \frac{c\Delta f}{2\alpha} m. \quad (2.3.13)$$

Warto zaznaczyć, iż w idealnym przypadku każda komórka odległości o niezerowej wartości modułu będzie jednoznacznie powiązana - poprzez wzór (2.3.13) - z pewnym obiektem znajdującym się w zasięgu czujnika radarowego. W rzeczywistości pomiary radarowe zawierają, niestety, wiele zakłóceń, dlatego często stosuje się tzw. adaptacyjne progi odcięcia [58] w celu odrzucenia fałszywych wartości; komórki zakwalifikowane jako szum nie są brane pod uwagę przy wyliczaniu parametrów danego wykrycia.

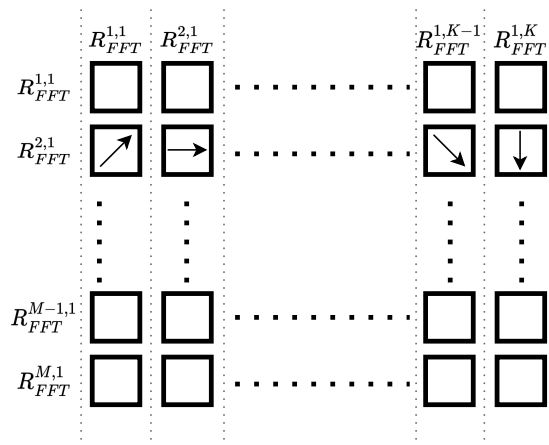
Przypomnijmy, że okres pomiarowy składa się z K pakietów po M próbek. Innymi słowy, komplet danych dla pojedynczego kanału odbiorczego zawiera K zbiorów R_{FFT} , obliczonych dla wszystkich pakietów. Oznaczmy przez $R_{FFT}^{m,k}$ komórkę odległości o indeksie m właściwą dla pakietu (zbioru R_{FFT}) o indeksie k , gdzie: $m \in [1, M]$ oraz $k \in [1, K]$; $R_{FFT}^{m,k}$ uzyskuje się na podstawie takiego oto wzoru:

$$R_{FFT}^{m,k} = \sum_{i=1}^M x_{i,k} e^{-j \frac{2\pi i m}{M}}, \quad (2.3.14)$$

gdzie $x_{i,k}$ reprezentuje i -ty pomiar częstotliwości różnicowej dostępny na wyjściu przetwornika analogowo-cyfrowego w cyklu modulacji (pakiecie) o indeksie k , przy czym: $i \in [1, M]$ oraz $k \in [1, K]$.

Założmy teraz, że w polu widzenia radaru istnieje obiekt o niezerowej prędkości względnej v_r , a co za tym idzie - w każdym wyliczonym zestawie R_{FFT} powinna znajdować się jedna komórka odległości ($R_{FFT}^{m,k}$) z liczbą zespoloną o niezerowej wartości modułu, powiązana jednoznacznie z pewną częstotliwością f_B . Oczywiście skoro obiekt się porusza, to jak wynika z równania (2.3.13) częstotliwość f_B będzie się zmieniać. Niemniej jednak, w trakcie pojedynczego okresu pomiarowego różnice w rejestrowanych wartościach f_B są tak małe, iż dla $k \in [1, K]$ indeks m komórki odległości zwykle nie ulegnie zmianie, co skutkuje brakiem możliwości poprawnego obliczenia prędkości względnej.

Do obliczania prędkości v_r zdecydowanie lepiej wykorzystać zmienność argumentu liczb zespolonych dostępnych w wyznaczonych zbiorach R_{FFT} [55]. Uznajmy, że odbicie od danego obiektu (o niezerowej prędkości względnej v_r) reprezentowane jest przez pojedynczą komórkę $R_{FFT}^{m,k}$, której indeks m nie ulega zmianie dla $k \in [1, K]$. Przykładową tego typu sytuację zobrazowano na rysunku 2.5, gdzie argument oraz moduł liczby zespolonej przypisanej do danej komórki odległości wyrażono w postaci wektora.



Rysunek 2.5. Zmieniająca się na przestrzeni K pakietów faza liczby zespolonej (powiązanej jednoznacznie z częstotliwością różnicową f_B) dla pewnej ustalonej komórki odległości

Przeprowadza się więc tym razem M operacji FFT, analizując argumenty liczb zespolonych powiązanych z komórkami R_{FFT} w czasie (na przestrzeni K pakietów). Jako rezultat uzyskuje się M zbiorów v_{FFT} , których elementy (komórki prędkości) można obliczyć według następującej formuły:

$$v_{FFT}^{m,k} = \sum_{i=1}^K \arg R_{FFT}^{m,i} e^{-j \frac{2\pi i k}{K}}, \quad (2.3.15)$$

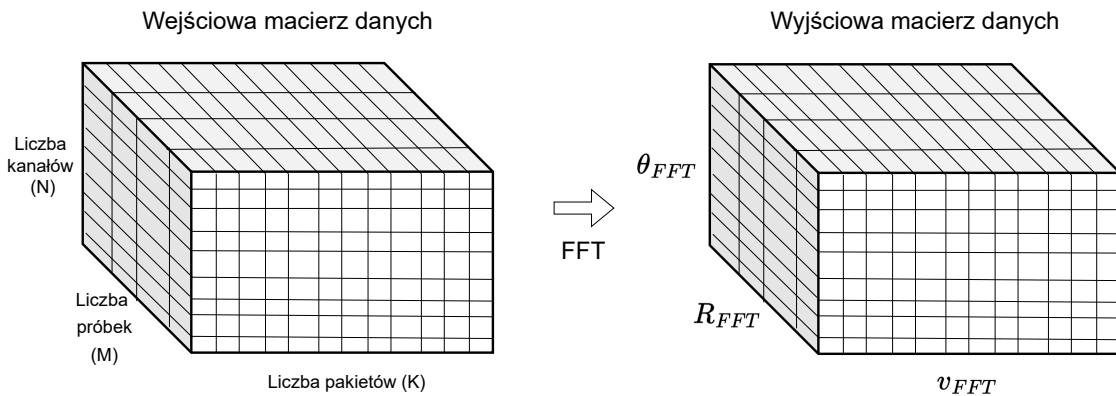
gdzie $v_{FFT}^{m,k}$ to pojedyncza komórka prędkości ($m \in [1, M]$, $k \in [1, K]$) zawierająca pewną liczbę zespoloną o module wskazującym jednoznacznie na przesunięcie fazowe ϕ , na podstawie którego wyznacza się prędkość względną v_r , czyli poprzez wykorzystanie wzoru (2.3.6).

W analogiczny sposób określana jest wartość przesunięcia fazowego ψ zdefiniowanego w równaniu (2.3.7). Mianowicie, należy wykonać N operacji FFT, analizując argumenty liczb zespolonych powiązanych z komórkami R_{FFT} na przestrzeni N kanałów odbiorczych. Produktem tych operacji jest N zbiorów θ_{FFT} , których składowe (komórki kąta) uzyskuje się w taki oto sposób:

$$\theta_{FFT}^{m,n} = \sum_{i=1}^N \arg R_{FFT}^{m,i} e^{-j \frac{2\pi i n}{N}}, \quad (2.3.16)$$

gdzie $\theta_{FFT}^{m,n}$ to pojedyncza komórka kąta ($m \in [1, M]$, $n \in [1, N]$) zawierająca pewną liczbę zespoloną o module wskazującym jednoznacznie na przesunięcie fazowe ψ , na podstawie można obliczyć kąt θ , a więc przy użyciu wzoru (2.3.8).

Podsumowując, rezultatem opisanego procesu przetwarzania sygnałów jest trójwymiarowa, zespolona macierz danych radarowych (ang. Radar Data Cube, RDC) pokazana na rysunku 2.6. Każdy komponent macierzy jest związany z dokładnie jedną komórką R_{FFT} , v_{FFT} oraz θ_{FFT} . Po przeprowadzeniu odpowiedniej filtracji uzyskuje się ostateczny zbiór wykryć radarowych, wyznaczając R , v_r oraz θ dla wszystkich niezerowych komórek.



Rysunek 2.6. Informacje wejściowe do procesu przetwarzania sygnałów w radarach typu FMCW (macierz danych w dziedzinie czasu), wraz z rezultatem tegoż procesu, czyli macierzą danych w dziedzinie częstotliwości (RDC)

2.4 Radary w zaawansowanych systemach wspomagania kierowcy

Jak napisano we wcześniejszych fragmentach rozdziału, kluczowe układy klasy ADAS, takie jak ACC oraz AEBS, są w znacznej mierze oparte na czujnikach radarowych typu FMCW. Niestety, surowy zbiór wykryć wygenerowany przez tego typu sensor nie jest odpowiednim typem danych z perspektywy docelowej funkcji wspomagania kierowcy. W tej części wyjaśniony będzie proces mający na celu wydobycie cennych informacji o pozycji, prędkości i wymiarach obiektów znajdujących się w otoczeniu pojazdu. Dodatkowo zaprezentowany zostanie wysokopoziomowy przepływ informacji w przykładowym urządzeniu z kategorii ADAS. Zdecydowano się także na omówienie różnych metodologii weryfikacji systemów typu ADAS w środowiskach wirtualnej jazdy, czyli kluczowej tematyki w kontekście niniejszej rozprawy.

2.4.1 Procedura wykrywania obiektów

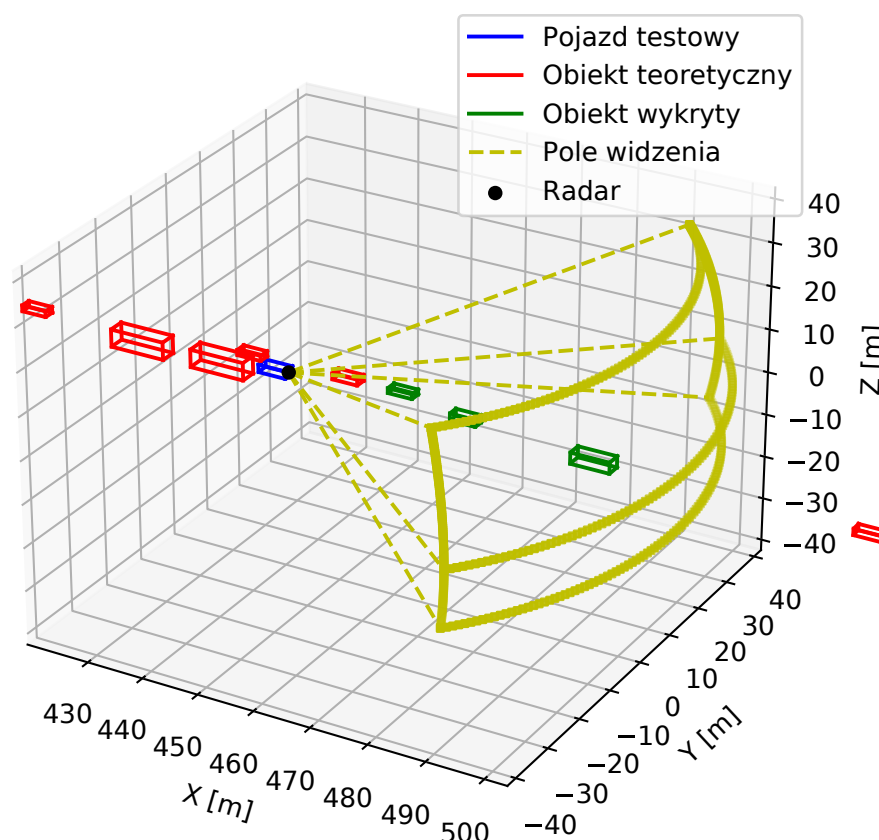
Przed wywołaniem docelowej funkcji typu ADAS konieczne jest przetworzenie surowych pomiarów radarowych do odpowiedniego formatu. W tym celu wykorzystuje się algorytm wykrywania obiektów, którego rolą jest zidentyfikowanie wszystkich pojazdów i obiektów statycznych znajdujących się w polu widzenia czujnika, bazując bezpośrednio na zbiorze wykryć zwróconych przez radar. Innymi słowy, dla każdego obiektu widocznego z perspektywy sensora generuje się odpowiadającą mu tzw. bryłę brzegową. W tym przypadku zakłada się, że jest to najmniejszy, dowolnie zorientowany prostopadłościan w całości otaczający dany cel (ang. Bounding Box). Na rysunku 2.7 pokazano wynik zwrócony przez idealny algorytm wykrywania obiektów [60].

Na rysunku można zaobserwować dwie kategorie obiektów, oprócz pojazdu testowego. Kolorem zielonym oznaczono prostopadłościany wygenerowane przez wspomniany algorytm. Zgodnie z oczekiwaniami zostały one utworzone tylko i wyłącznie dla pojazdów widocznych z perspektywy czujnika radarowego. Z kolei barwą czerwoną wyróżniono bryły brzegowe dla celów nie znajdujących się w polu widzenia sensora. Są to zatem obiekty teoretyczne, których procedura wykrywania obiektów nie powinna wygenerować, gdyż nie ma możliwości skojarzenia z nimi jakichkolwiek wykryć radarowych.

W literaturze funkcjonalność wykrywania obiektów bazująca na danych radarowych jest oparta zarówno na metodach tradycyjnych, takich jak rozszerzony filtr Kalmana oraz modele probabilistyczne [61–66], jak i na algorytmach wykorzystujących sztuczne sieci neuronowe [67, 68].

2.4.2 Przykładowy przepływ informacji w układach klasy ADAS

Funkcjonalność należąca do kategorii ADAS, na przykład ACC czy AEBS, jest zwykle uruchamiana na pewnym urządzeniu mikroprocesorowym. Zadaniem tego typu systemów jest kontrola nad pojazdem

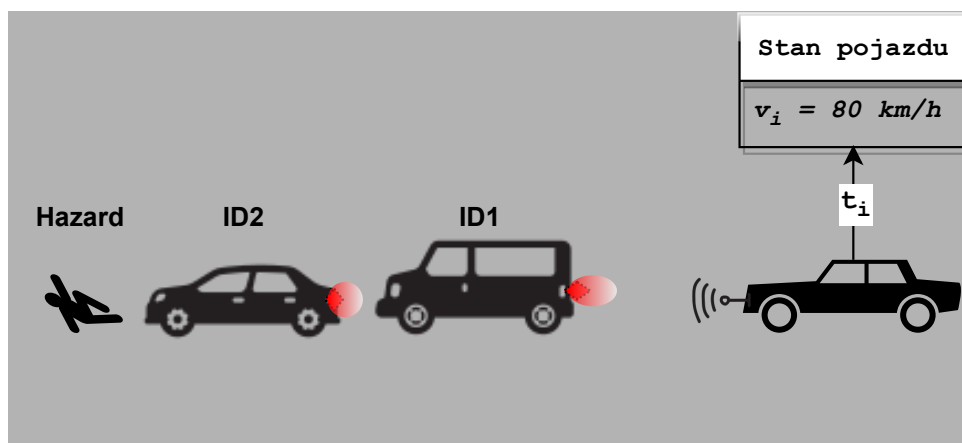


Rysunek 2.7. Rezultat idealnego algorytmu wykrywania obiektów

na podstawie informacji o ruchu i pozycji wszelkich obiektów znajdujących się w jego otoczeniu; kontrola ta może być bezpośrednia (ingerencja w główne elementy wykonawcze samochodu, takie jak np. przepustnica, układ hamulcowy czy drążek kierowniczy) lub pośrednia (wyświetlenie odpowiednich komunikatów czy ostrzeżeń w systemie informacyjno-rozrywkowym pojazdu). W szczególności, urządzenia z kategorii AEBS mają za zadanie aktywowanie układu hamulcowego samochodu oraz ciągłą kontrolę nad siłą hamowania w celu uniknięcia kolizji w przypadku wystąpienia nieoczekiwanego zdarzenia zagrażającego bezpieczeństwu uczestników ruchu drogowego.

Na tym etapie warto w uproszczony sposób zilustrować przepływ informacji w urządzeniach typu ADAS bazujących na czujnikach radarowych. Wspomniana procedura będzie wyjaśniona na przykładzie układu klasy AEBS; oczywiście rzeczywiste tego typu systemy bazują zwykle na fuzji danych z kamery i radaru, jednak na potrzeby zaprezentowania uproszczonego obiegu danych przyjmijmy, że wykorzystuje się tutaj tylko pomiary radarowe. Załóżmy, że zaawansowana funkcja bezpieczeństwa, w którą wyposażony jest samochód, bazuje na pomiarach z czołowego czujnika radarowego. Rozważmy scenariusz, w którym na jezdni przed pojazdem testowym (wiodącym) jadą dwa inne pojazdy o identyfikatorach kolejno *ID1* oraz *ID2*. Przyjmijmy też, iż samochód wiodący przemieszcza się w

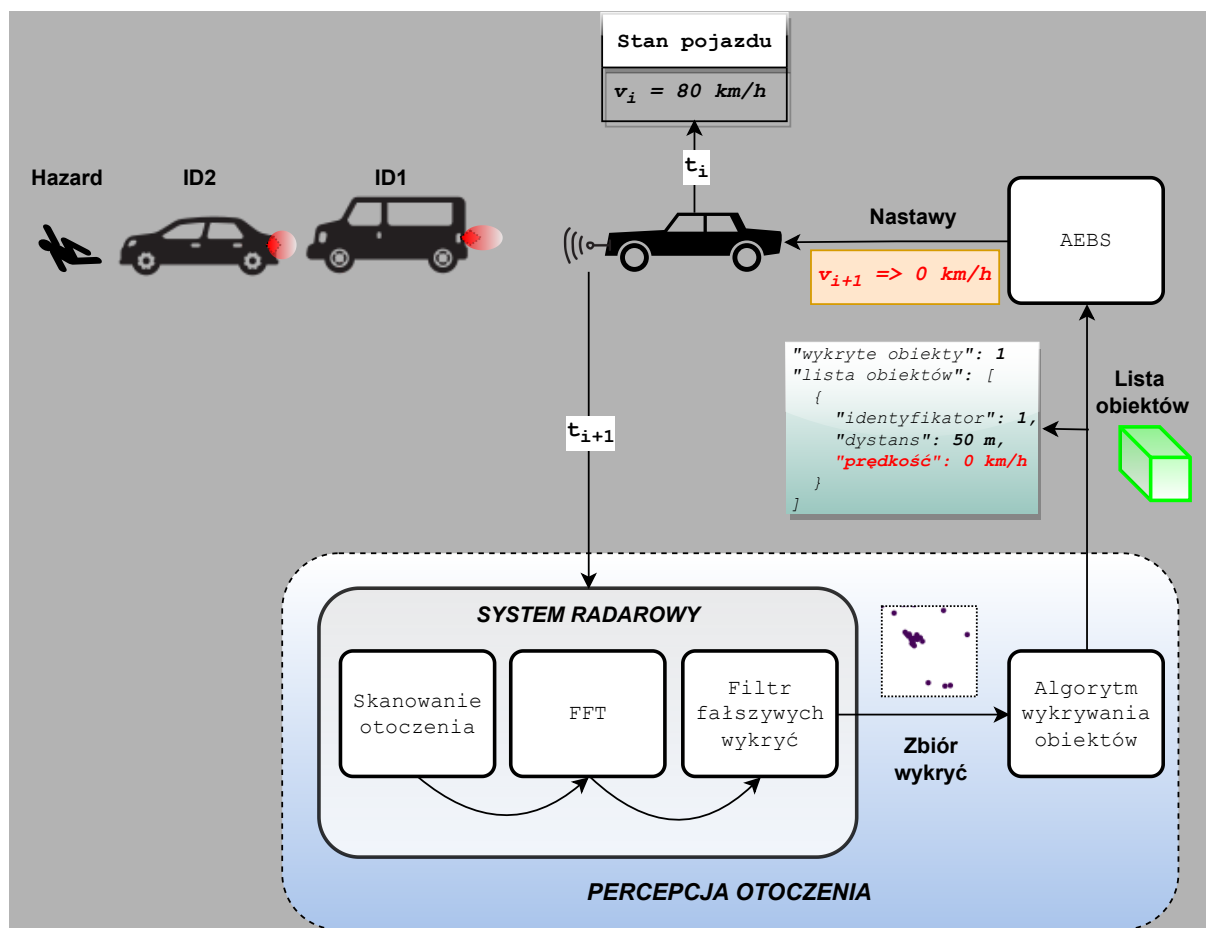
momencie czasu t_i z pewną stałą, niezerową prędkością, przy czym kierowca tego pojazdu ma obniżoną koncentrację i w tej konkretnej chwili nie zwraca uwagi na to, co się dzieje na drodze. Załóżmy także, że na pasie ruchu przed pojazdem testowym dochodzi do niebezpiecznego zdarzenia, zmuszającego pojazdy o identyfikatorach $ID1$ oraz $ID2$ do nagłego zatrzymania. Opisaną wyżej sytuację zobrazowano na rysunku 2.8.



Rysunek 2.8. Przykładowy scenariusz testowy weryfikujący poprawność działania układu typu AEBS

Tymczasem układ mikroprocesorowy przechodzi do cyklu o indeksie $i + 1$. Czujnik radarowy typu FMCW generuje ciąg impulsów elektromagnetycznych z liniowo modulowaną częstotliwością (rysunek 2.2). Następnie anteny odbiorcze rejestrują echo wyemitowanej fali radiowej, zapisując każdą próbkę sygnału różnicowego w pamięci mikrokontrolera przy użyciu przetwornika analogowo-cyfrowego. Później przeprowadza się trójwymiarową operację FFT, której rezultatem jest zespolona macierz danych RDC (rysunek 2.6). Każda komórka zbioru RDC wskazuje jednoznacznie na pojedyncze wykrycie, gdzie poszczególne wymiary danego elementu powiązane są z: odległością R , prędkością względną v_r oraz kątem w płaszczyźnie horyzontalnej θ . Zestaw RDC zawiera jednak sporo zakłóceń. Dlatego przeprowadza się odpowiednią operację filtracji w celu pozbycia się fałszywych pomiarów. Tak przetworzony zbiór wykryć (trójki wartości R , v_r oraz θ) jest na dalszym etapie analizowany przez algorytm wykrywania obiektów. Jak wspomniano wyżej, procedura ta ma za zadanie oszacowanie tzw. brył brzegowych (prostokątów) dla wszystkich nieprzysłoniętych pojazdów, które znajdują się w polu widzenia radaru. Warto też zaznaczyć, że proces trwający od rozpoczęcia skanowania otoczenia przez czujnik radarowy aż do momentu uzyskania listy obiektów określa się terminem percepcji otoczenia. W tym konkretnym przypadku algorytm wykrywania obiektów powinien oszacować na bazie pomiarów z radaru bryłę brzegową dla samochodu o identyfikatorze $ID1$, czyli dostarczyć informację o pozycji oraz wymiarach tejże bryły, jak i oszacowaną prędkość tego celu. Ostatecznie, system typu AEBS ze względu na obecność nieporuszającego się obiektu na kursie kolizyjnym z pojazdem testowym

podejście decyzję o awaryjnym hamowaniu, mającym na celu zmniejszenie prędkości samochodu wiodącego do 0. Omówiony wyżej przykładowy przepływ informacji w urządzeniach klasy ADAS, zaprezentowany na bazie układu z kategorii AEBS, zaprezentowano także w formie graficznej na rysunku 2.9, gdzie zmienne t_i oraz t_{i+1} reprezentują kolejno aktualny i następny cykl mikroprocesora, na którym uruchamia się całą procedurę przetwarzania danych.



Rysunek 2.9. Wysokopoziomowy przepływ informacji w systemach typu ADAS bazujących na czujnikach radarowych, na przykładzie uproszczonego urządzenia klasy AEBS

2.4.3 Wirtualna walidacja

Przed wdrożeniem danego urządzenia z kategorii ADAS do produkcji seryjnej konieczna jest jego rzetelna weryfikacja. W tym celu przeprowadza się szereg jazd testowych, by sprawdzić zachowanie układu w najróżniejszych sytuacjach, które mogą wystąpić na drodze; pojazd testowy wyposaża się oczywiście w docelowy zestaw sensorów oraz w urządzenie mikroprocesorowe z oprogramowaniem typu ADAS podlegającym weryfikacji. Procedura gromadzenia danych to niezwykle kosztowna i

czasochłonna operacja; aby udowodnić, że decyzje systemu nie zagrażają zdrowiu i życiu uczestników ruchu drogowego, należy zgromadzić informacje z przejazdu o długości rzędu miliona kilometrów [8].

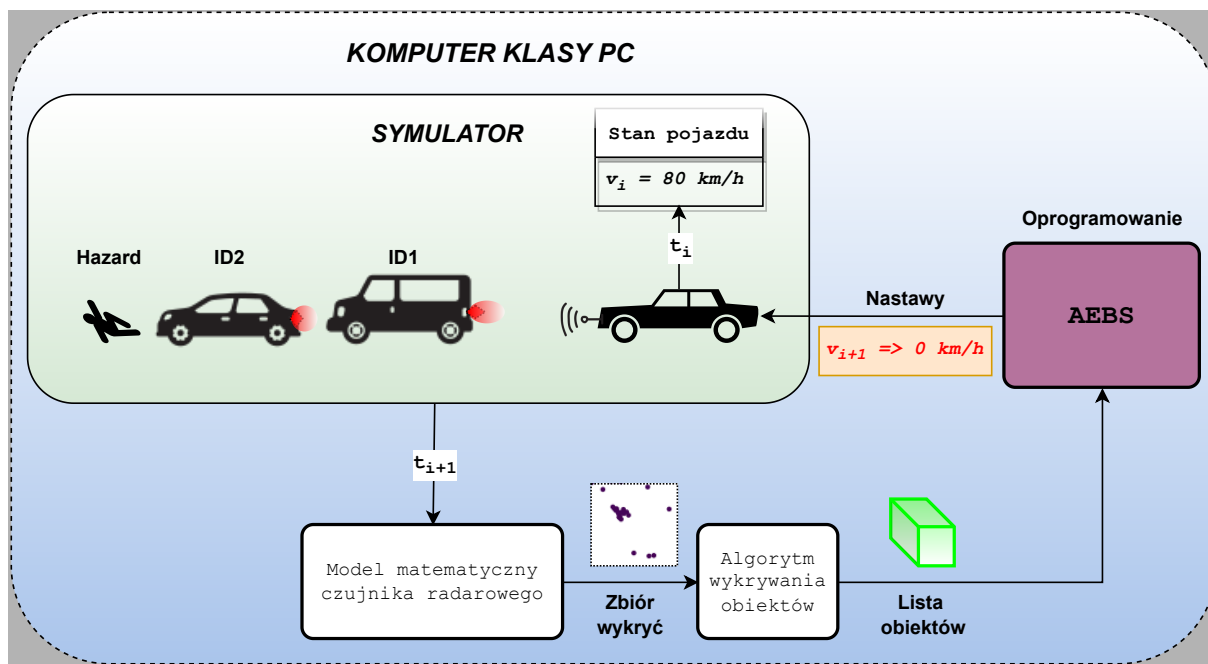
Dlatego też liderzy przemysłu motoryzacyjnego zdecydowali się na wykorzystanie środowisk wirtualnej jazdy do przeprowadzenia części eksperymentów. Proces oceny urządzeń typu ADAS przy użyciu symulacji określa się mianem wirtualnej walidacji. Za największą zaletę tejże metodologii należy uznać fakt, iż w relatywnie krótkim czasie można zbadać działanie układu w szczególnie groźnych sytuacjach, których odtworzenie w rzeczywistości byłoby zbyt niebezpieczne.

Do przeprowadzenia rzetelnego testu w środowisku wirtualnej jazdy wymagany jest symulator dokładnie odwzorowujący prawdziwe scenariusze drogowe oraz modele czujników odpowiedzialne za generację danych syntetycznych będących wierną kopią pomiarów z rzeczywistych sensorów [11].

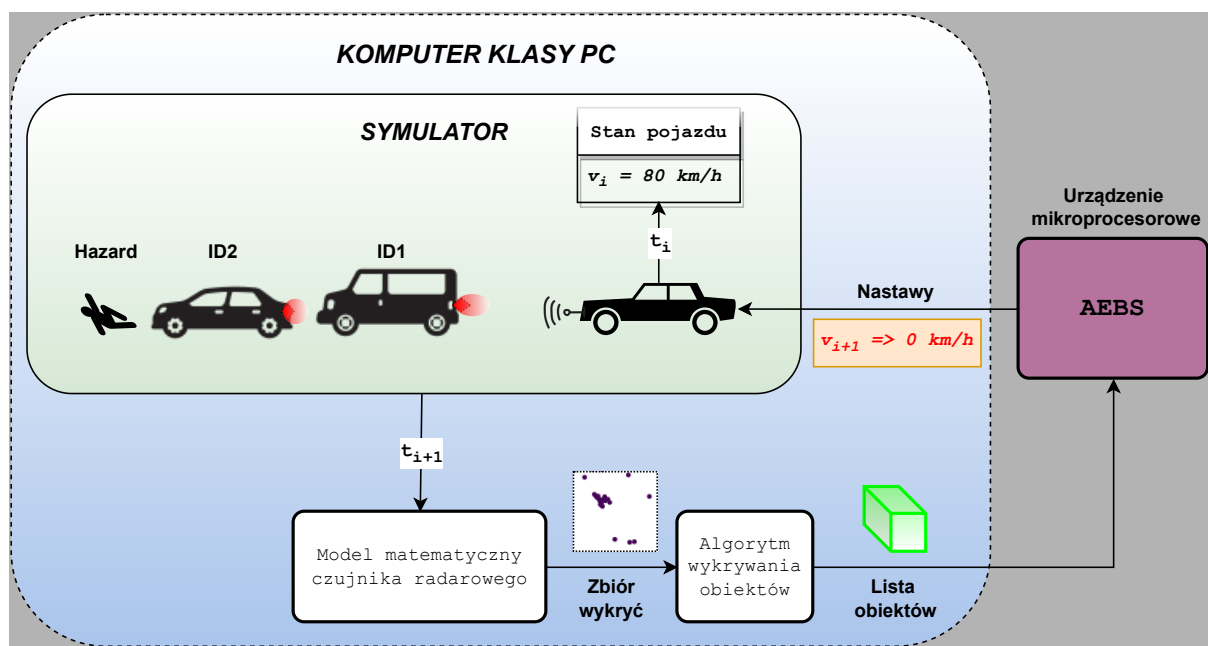
Wirtualną walidację wykonuje się zwykle w dwóch konfiguracjach. Są to: symulacja w pętli sprzętowej (ang. hardware in the loop, HIL) oraz symulacja w pętli oprogramowania (ang. software in the loop, SIL). Test SIL ma na celu zweryfikowanie jakości kodu wykonywalnego, w którym zawarta jest funkcjonalność projektowanego układu klasy ADAS. Z kolei w HIL bada się system docelowy, czyli układ mikroprocesorowy, aplikację i protokół wymiany danych. SIL należy więc traktować jako środowisko do prototypowania, gdzie sprawdza się nowe funkcjonalności. Natomiast HIL to rygorystyczny sprawdzian gotowego rozwiązania, więc oprócz rzetelności opracowanej technologii weryfikuje się tutaj także wydajność układu. Konsekwencją tego jest fakt, iż w teście typu HIL, cała symulacja musi pracować w czasie rzeczywistym. Dlatego też opracowany model sensora powinien charakteryzować się nie tylko wysokim poziomem realizmu, ale też możliwie jak najmniejszą złożonością obliczeniową, by nie wprowadzić zbyt dużego opóźnienia do omawianego procesu.

Warto tutaj zaznaczyć, że docelowy (produkcyjny) system klasy ADAS składa się nie tylko z zaawansowanej funkcji bezpieczeństwa, takiej jak AEBS czy ACC, ale także z układu percepcji, czyli jednego lub kilku czujników samochodowych połączonych z algorytmem wykrywania obiektów. Jednak w ramach tej pracy przez wirtualną walidację urządzenia typu ADAS rozumie się test (w trybie SIL lub HIL) samej funkcjonalności odpowiedzialnej w sposób pośredni albo bezpośredni za ruch pojazdu - w tym przypadku są to wspomniane wielokrotnie funkcje z kategorii AEBS czy ACC.

Wspomniane tryby wirtualnej walidacji zaprezentowano w formie graficznej na rysunkach 2.10 (SIL) oraz 2.11 (HIL), gdzie zmienne t_i oraz t_{i+1} reprezentują kolejno aktualny i następny cykl symulatora, działającego na komputerze klasy PC, w którym uruchamia się pewien scenariusz wirtualnej jazdy, będący realistycznym odwzorowaniem jakiejś konkretnej sytuacji drogowej; symulator ten udostępnia zbiór danych wejściowych dla modelu symulacyjnego czujnika radarowego oraz otrzymuje zestaw nastaw dla pojazdu wiodącego poruszającego się w tymże środowisku wirtualnej jazdy od testowanej funkcjonalności typu AEBS.



Rysunek 2.10. Uproszczony schemat przykładowego środowiska do wirtualnej walidacji funkcjonalności klasy AEBS, należącej do kategorii układów typu ADAS, zrealizowanej w trybie SIL

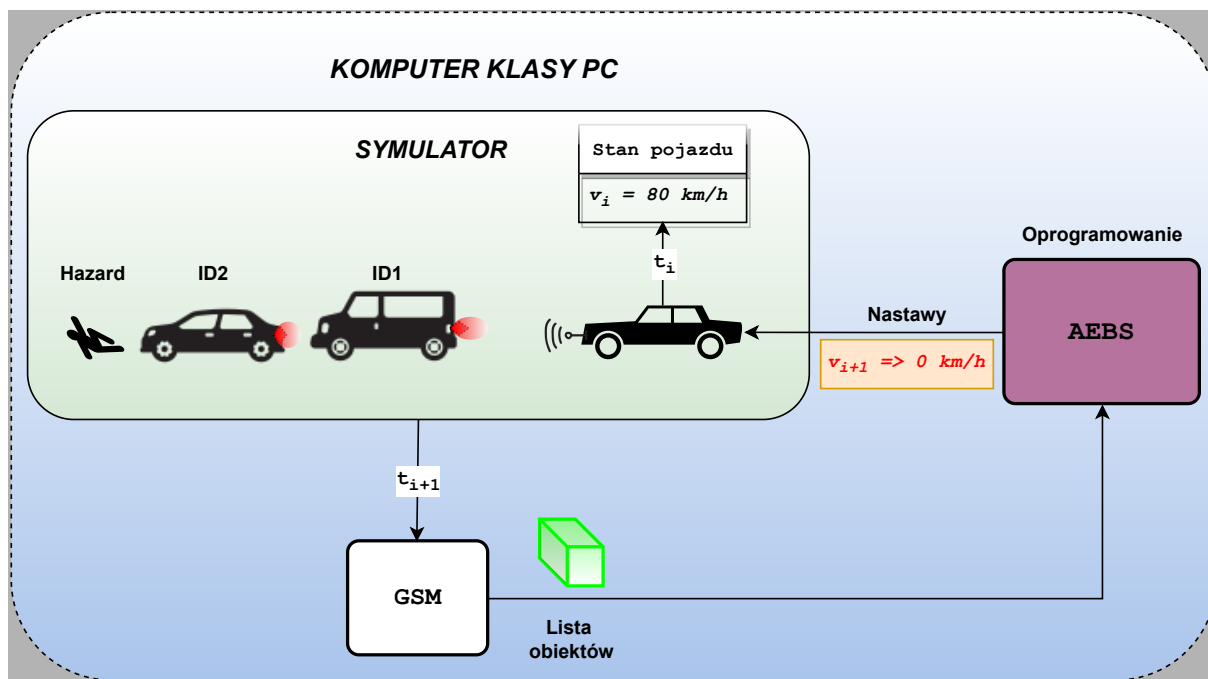


Rysunek 2.11. Uproszczony schemat przykładowego środowiska do wirtualnej walidacji funkcjonalności klasy AEBS, należącej do kategorii układów typu ADAS, zrealizowanej w trybie HIL

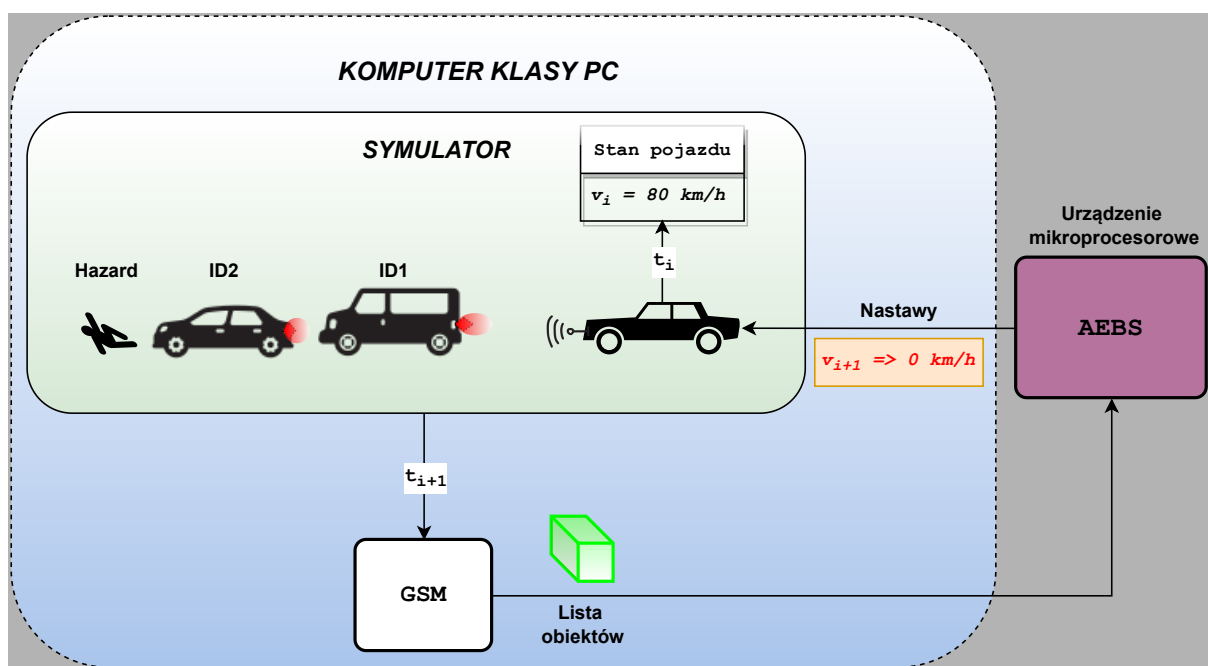
Warto na tym etapie wspomnieć o pewnej istotnej kwestii. Wyniki zwracane przez algorytm wykrywania obiektów są zależne od wykorzystanego czujnika radarowego. Innymi słowy, jeśli zbiór

wykryć oszacowany dla aktualnego otoczenia pojazdu zawiera sporą liczbę zakłóceń, to należy oczekiwać, że parametry każdej bryły brzegowej wygenerowanej przez wykorzystaną procedurę wykrywania również będą obarczone sporym błędem. Można zatem algorytm wykrywania obiektów traktować jako nieodłączną część systemu radarowego. Właściwość ta pozwala na wprowadzenie pewnego uproszczenia do symulacji HIL oraz SIL, czyli zamodelowania systemu radarowego na poziomie brył brzegowych. W takiej konfiguracji pomija się generację sztucznego zbioru wykryć oraz wywołanie rzeczywistego algorytmu wykrywania obiektów i wytwarza się syntetyczny zbiór brył brzegowych bezpośrednio na podstawie informacji dostarczonych ze środowiska wirtualnej jazdy; taką wysokopoziomową symulację czujnika często określa się mianem uniwersalnego modelu czujnika (ang. generic sensor model, GSM). Oczywiście, sztuczna lista obiektów wygenerowana przy użyciu GSM musi być wiernym odwzorowaniem rezultatów zwracanych przez rzeczywistą procedurę wykrywania obiektów. Oznacza to, że GSM powinien oszacować tylko bryły widoczne z perspektywy wirtualnego sensora, z dodatkiem zakłóceń charakterystycznych dla samochodowych systemów radarowych.

Tego typu uproszczona procedura wirtualnej walidacji powinna być (co do zasady) użyta jedynie do wstępnej weryfikacji systemów klasy ADAS, np. na wczesnym etapie projektu. Niestety, symulacja z wykorzystaniem GSM będzie w ogólności charakteryzować się znacznie gorszym poziomem rzetelności w porównaniu do modelu zrealizowanego na poziomie wykryć wraz z rzeczywistym algorytmem wykrywania obiektów. Zmodyfikowany, czyli wykorzystujący koncepcję GSM, proces wirtualnej walidacji zaprezentowano na rysunkach 2.12 (SIL) oraz 2.13 (HIL), gdzie zmienne t_i oraz t_{i+1} reprezentują (podobnie jak w przypadku rysunków 2.11 oraz 2.10) kolejno aktualny i następny cykl symulatora z uruchomionym scenariuszem wirtualnej jazdy, będącym realistycznym odwzorowaniem pewnej sytuacji drogowej; symulator ten tym razem udostępnia zbiór danych wejściowych dla modelu typu GSM oraz otrzymuje zestaw nastaw dla pojazdu wiodącego poruszającego się w tymże środowisku wirtualnej jazdy od testowanej funkcjonalności typu AEBS.



Rysunek 2.12. Uproszczony schemat przykładowego środowiska do wirtualnej walidacji funkcjonalności klasy AEBS, należącej do kategorii układów typu ADAS, zrealizowanej w trybie SIL w oparciu o koncepcję GSM



Rysunek 2.13. Uproszczony schemat przykładowego środowiska do wirtualnej walidacji funkcjonalności klasy AEBS, należącej do kategorii układów typu ADAS, zrealizowanej w trybie HIL w oparciu o koncepcję GSM

3 Kompatybilność modelu matematycznego ze standardami przemysłu motoryzacyjnego

3.1 Komitet standaryzacyjny ASAM

Życie produktu należącego do kategorii zaawansowanych systemów wspomagania kierowcy (ADAS) zaczyna się w momencie zdefiniowania wymagań, jakie docelowy system ma spełniać. Później opracowuje się architekturę układu, zarówno dla części sprzętowej, jak i dla części programowej. Następnie zaczyna się fazę projektową, gdzie implementowane są kolejne funkcjonalności aplikacji. Równolegle z rozwojem produktu przeprowadzane są testy funkcjonalne systemu, których poziom skomplikowania narasta proporcjonalnie do stopnia zaawansowania wytwarzanego układu. Później, kiedy aplikacja jest już odpowiednio dojrzała, przygotowuje się prototyp, czyli urządzenie spełniające postawione na początku wymagania (choć niekoniecznie wszystkie), ale jeszcze niegotowe do masowej produkcji. Taką wstępną wersję systemu typu ADAS weryfikuje się pod kątem najróżniejszych sytuacji drogowych w celu zidentyfikowania błędów w opracowanym rozwiązaniu. Następnie, kiedy działanie układu odpowiada przyjętym założeniom, przygotowuje się oficjalną wersję urządzenia, którą implementuje się na tzw. elektronicznej jednostce sterującej (ang. electronic control unit, ECU), czyli samochodowym systemie wbudowanym czasu rzeczywistego. W końcu, przed wprowadzeniem produktu do masowej produkcji, należy udowodnić, że system nie będzie stwarzać zagrożenia w ruchu drogowym. W tym celu przeprowadza się kampanię jazd testowych, czyli długodystansowego przejazdu po drogach publicznych, w trakcie którego gromadzone są szczegółowe informacje związane z działaniem układu, w szczególności pomiary z czujników zamocowanych na pojeździe, dane o stanie samochodu (prędkość, kąt skręcenia kół itp.) oraz decyzje podejmowane przez dany ADAS.

Widać zatem, że zanim system bezpieczeństwa zostanie wprowadzony do produkcji, wiele zależnych od siebie procesów musi zakończyć się powodzeniem. Co za tym idzie, w trakcie życia produktu przekazuje się ogromną liczbę danych pomiędzy najróżniejszymi komponentami docelowego urządzenia. Dlatego, by skrócić czas opracowywania nowego rozwiązania i w konsekwencji znacząco zredukować koszt układu, firmy z przemysłu motoryzacyjnego zdecydowały się na współpracę, której rezultatem jest powstanie komitetu standaryzacyjnego ASAM (ang. Association for Standardization of

Automation and Measuring Systems). ASAM założono w 1998 roku; obecnie stowarzyszenie to skupia ponad 400 organizacji z całego świata, w tym producentów samochodów, podwykonawców, twórców oprogramowania do projektowania, testowania oraz weryfikacji systemów klasy ADAS, jak również instytucje badawcze.

Przez blisko 25 lat istnienia opracowano wiele standardów kluczowych dla przemysłu samochodowego w takich dziedzinach, jak: pomiary i kalibracja, diagnostyka, rozwój oprogramowania, automatyzacja testów, zarządzanie i analiza danych oraz symulacje; przykładem tego, jak wpływowe okazały się rozwiązania zaproponowane przez wspomniany komitet, jest fakt, iż ponad 90 procent pojazdów poruszających się po drogach skalibrowano z wykorzystaniem wzorców ASAM [69]. Poniżej wymieniono kilka ważniejszych standardów będących częścią ASAM:

- MDF (ang. Measurement Data Format) - format pliku wykorzystywany do zapisu danych nagranych w trakcie przejazdów testowych, służących do ewaluacji danego ADAS,
- CDF (ang. Calibration Data Format) - format opisu danych definiujący sposób przechowywania informacji kluczowych w kontekście kalibracji parametrów ECU,
- ARTI (ang. ASAM Real Time Interfejs) - interfejs do analizy częstotliwości przełączeń procesów uruchamianych na samochodowych systemach wbudowanych czasu rzeczywistego, zaprojektowanych zgodnie z architekturą AUTOSAR (ang. Automotive Open System Architecture) [70],
- MCD-2 NET - standard (określany także mianem Fibex) służący do projektowania, konfiguracji, monitorowania i symulacji komunikacji w sieciach samochodowych; umożliwia on zaprojektowanie topologii sieci składającej się z wielu ECU poprzez zdefiniowanie bram sieciowych, portów sieciowych oraz sposobu wymiany danych pomiędzy modułami (lista sygnałów wejściowych i wyjściowych dla danego ECU),
- OpenCRG (ang. Open Curved Regular Grid) - format pliku służący do dokładnego odwzorowania danej drogi, w szczególności wszelkich nierówności oraz zmian nachylenia występujących na ustalonym odcinku jezdni,
- OpenDRIVE (ang. Open Dynamic Road Information for Vehicle Environment) - struktura danych umożliwiająca zapisanie wszystkich istotnych informacji o sieci dróg, takich jak na przykład: połączenia pomiędzy jezdniami wchodzącymi w skład danej mapy, geometria pasów ruchu, konfiguracja skrzyżowań oraz lokalizacja znaków drogowych i sygnalizacji świetlnej; warto tutaj zaznaczyć, że OpenDRIVE może być wykorzystany zarówno do opisu środowiska wirtualnego, jak i do opisu rzeczywistej mapy drogowej,
- OpenSCENARIO - standard do zapisu dynamiki wirtualnego scenariusza, przy użyciu którego można w przejrzysty sposób opisać skomplikowane, zsynchronizowane manewry pojazdów, pieszych czy też innych uczestników ruchu drogowego; manewr wyraża się za pomocą decyzji

podejmowanych przez dany pojazd (np. zmiana pasa ruchu) lub przy użyciu trajektorii, po której ma poruszać się wirtualny obiekt,

- OSI (ang. Open Simulation Interface) - to interfejs umożliwiający wymianę informacji pomiędzy modelem sensora i urządzeniem klasy ADAS testowanym w środowisku wirtualnej jazdy.

3.2 Standaryzacja w kontekście wirtualnej walidacji

Jak już wspomniano we wcześniejszych fragmentach pracy, przejazd samochodów testowych po drogach publicznych to proces wymagający ogromnych nakładów finansowych, gdyż należy zweryfikować system w najróżniejszych sytuacjach drogowych; raport [8] wspomina, że do udowodnienia niezawodności układu konieczne jest przejechanie dystansu rzędu milionów kilometrów. Dlatego też coraz częściej w procesie weryfikacji urządzeń klasy ADAS wykorzystuje się koncept tzw. wirtualnej walidacji, który polega na zastąpieniu części przejazdów testowych eksperymentami przeprowadzanymi w symulatorze jazdy. Dzięki takiemu podejściu skraca się znacznie czas potrzebny na opracowanie gotowego rozwiązania typu ADAS oraz zmniejsza się końcowa cena produktu.

Ostatnie cztery z wymienionych wyżej standardów (OpenCRG, OpenDrive, OpenSCENARIO, OSI) pełnią kluczową rolę w procesie wirtualnej walidacji układów należących do kategorii ADAS. Mianowicie, dzięki OpenScenario i OpenCRG możliwe jest dokładne odtworzenie w symulatorze rzeczywistej sieci dróg. Z kolei OpenDRIVE pozwala na rzetelną realizację symulacji w pętli zamkniętej (typu HIL lub SIL), gdyż decyzje podejmowane przez dane urządzenie klasy ADAS mogą być precyzyjnie przeliczone na trajektorię pojazdu głównego (kontrolowanego przez zaawansowany system bezpieczeństwa), z uwzględnieniem ruchu pozostałych obiektów będących częścią scenariusza testowego. W końcu przy użyciu OSI można w prosty sposób wyodrębnić kluczowe informacje na temat aktualnego stanu środowiska wirtualnej jazdy, będące wejściem do modelu sensora, oraz równie łatwo przekazać sztuczne pomiary sensoryczne do docelowej funkcji wspomagania kierowcy.

3.3 Wymiana danych symulacyjnych z wykorzystaniem standardu OSI

3.3.1 Zarys problemu

Załóżmy, że system należący do kategorii ADAS (np. AEBS) testowany jest w pętli zamkniętej w symulatorze jazdy, z zachowaniem wymogów czasu rzeczywistego; oczywiście rozważania opisane poniżej dotyczą zarówno konfiguracji SIL (rysunki 2.10 oraz 2.12), jak i HIL (rysunki 2.11 oraz 2.13). Mamy zatem zdefiniowany pewien scenariusz wirtualnej jazdy, z dokładnie określoną siecią dróg oraz z dodatkiem pewnej liczby obiektów statycznych i pojazdów poruszających się wedle z góry ustalonego lub losowego schematu, z których wyodrębniony jest samochód wiodący otrzymujący komendy

od systemu wspomagania kierowcy. Symulacja może być rzecz jasna zaprojektowana przy użyciu standardów OpenDRIVE, OpenSCENARIO i OpenCRG, aczkolwiek do skutecznego wykorzystania OSI nie jest to wymagane. Nie obejmuje się zatem żadnymi ograniczeniami procesu definicji części dynamicznej oraz statycznej sceny wirtualnej. Musi istnieć natomiast możliwość wyodrębnienia w czasie rzeczywistym symulacji pewnych informacji o aktualnym stanie scenariusza.

Za jedną z głównych idei stojących za powstaniem standardu OSI należy niewątpliwie uznać modularność symulacji. Innymi słowy, podział łańcucha procedur wykonywanych w trakcie pojedynczego cyklu testu wirtualnego na kilka ustandaryzowanych poziomów abstrakcji wydaje się być czymś niezwykle pożądanym z perspektywy twórcy produktu z kategorii ADAS. Jako przykład można sobie wyobrazić sytuację, w której dwóch producentów samochodów korzysta z różnych symulatorów jazdy, oczywiście do celów weryfikacji opracowywanych rozwiązań. Przyjmijmy, że obie firmy podchodzą rzetelnie do sprawy ochrony własności intelektualnej, zapewniającej im stabilną pozycję na rynku, w szczególności żadna ze spółek nie ujawnia (bez specjalnych umów o poufności) detali związanych z projektowanym systemem. Zakładając jednak, że w środowisku wirtualnej jazdy każdego z producentów wymiana danych pomiędzy modułami symulacji zrealizowana jest w oparciu o standard OSI, to istnieje możliwość bezproblemowej wymiany danego komponentu na element o innej charakterystyce. Co za tym idzie, możliwa jest szeroko pojęta współpraca firm z przemysłu samochodowego w zakresie rozwoju wirtualnej rzeczywistości. Pozytywnym skutkiem tego podejścia będzie najprawdopodobniej znaczący postęp technologiczny w całym przemyśle motoryzacyjnym, gdyż wysiłek producentów skupia się w większej mierze na ulepszeniu danej funkcjonalności typu ADAS niż na samodzielnym usprawnianiu środowiska testowego.

OSI dzieli zatem symulacje na kilka poziomów abstrakcji, gdzie każdy z nich ma precyzyjnie ustalony typ informacji, jaki ma oczekiwać na wejściu, jak również typ informacji, jaki ma zwracać. Zanim jednak zdefiniowane zostaną główne moduły symulacji, warto wpierw omówić dwie główne struktury danych standardu OSI: *OSI::SensorView* oraz *OSI::SensorData*.

3.3.2 Struktura *OSI::SensorView*

OSI::SensorView stanowi pewien uproszczony zapis aktualnego stanu wirtualnego scenariusza wyodrębniony z symulatora jazdy dla pewnego ustalonego dyskretnego momentu czasu. Do zidentyfikowania, z jaką chwilą symulacji powiązane są dane dostępne w konkretnej instancji *OSI::SensorView*, wykorzystuje się tzw. stempel czasowy (ang. timestamp), czyli liczbę, której wartość (wyrażona w sekundach) odpowiada okresowi, jaki upłynął od rozpoczęcia testu w symulatorze jazdy.

Jak już wspomniano w poprzednim rozdziale, funkcjonalność typu ADAS bazuje na informacjach oszacowanych przez moduł percepcji otoczenia, na który składa się zwykle jeden lub kilka czujników samochodowych, takich jak kamera czy radar, oraz algorytm wykrywania obiektów. Przypomnijmy,

że rolą sensora jest wykonanie skanu otoczenia wyrażonego albo w postaci chmury punktów (radar), albo w formie obrazu składającego się ze zbioru pikseli (kamera). Następnie, bazując na takiego typu surowych danych, algorytm wykrywania obiektów ma za zadanie oszacować zestaw brył brzegowych (prostopadłościanów) widocznych z perspektywy czujnika; przy czym w przypadku systemu wykorzystującego odczyty z wielu sensorów należy w tym procesie wykrywania obiektów uwzględnić także pewnego rodzaju fuzję informacji. Wejściem do funkcji typu ADAS są wobec tego pewne wysokopoziomowe dane będące odpowiednio przetworzoną wersją pomiarów sensorycznych.

Wracając do standardu OSI, strukturę *OSI::SensorView* należy traktować jako wejście do syntetycznego (symulowanego) modułu percepcji, gdyż to w *OSI::SensorView* można zapisać szczegóły dotyczące aktualnego stanu wirtualnego scenariusza. Warto tutaj zaznaczyć, że skoro rzeczywisty proces percepcji otoczenia opiera się na pomiarach zwróconych przez jeden lub wiele czujników samochodowych, to również jego wirtualny odpowiednik powinien wykorzystywać sztucznie wygenerowane odczyty. Co za tym idzie, zbiór *OSI::SensorView* jest tak naprawdę wejściem do modelu sensora, którego rolą jest właśnie odwzorowanie danego czujnika samochodowego.

W ogólności pojedynczą instancję struktury *OSI::SensorView* przypisuje się do pojedynczego modelu sensora; elementem *OSI::SensorView* jednoznacznie wskazującym na to powiązanie jest tzw. identyfikator (*OSI::Identifier*), czyli liczba naturalna o unikalnej wartości. W przypadku emulacji wielosensorowego układu typu ADAS należy utworzyć tyle kopii zbioru *OSI::SensorView*, ile wirtualnych czujników będzie wykorzystywać symulacja - oczywiście wartość identyfikatora musi być różna dla każdego z modeli, natomiast reszta informacji może pozostać niezmienną.

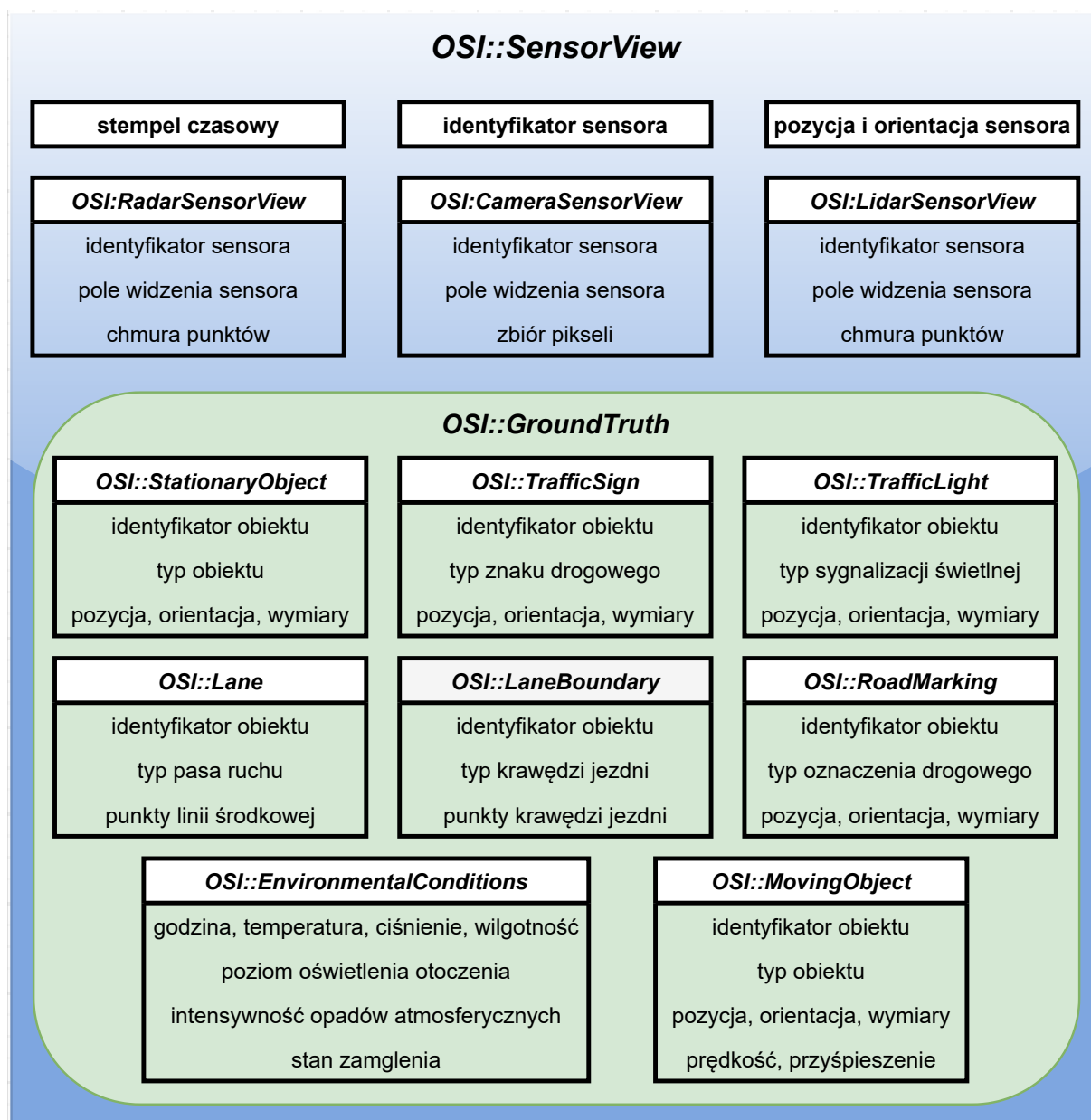
W poprzednim rozdziale wspomniano, że testy SIL oraz HIL będące głównym celem wirtualnej walidacji mogą być przeprowadzone w dwóch wariantach, czyli z wykorzystaniem albo niskopoziomowego modelu sensora (w przypadku próby odwzorowania danych radarowych wyjście stanowi syntetyczny zbiór wykryć), albo przy użyciu wysokopoziomowego modelu sensora określanego mianem GSM (wyjście stanowi syntetyczna lista obiektów). Warto także dodać, że w obu wspomnianych wersjach wymaga się, by sztuczne informacje sensoryczne były generowane w czasie rzeczywistym symulacji. Warunek ten w ogólności stosunkowo nietrudno spełnić, jeśli korzysta się z GSM. Niska złożoność obliczeniowa układów typu GSM wynika głównie z faktu pominięcia procedury generacji surowych pomiarów, niestety uproszczenia wprowadzane przez GSM wpływają negatywnie na jakość odwzorowania danych zwracanych przez rzeczywisty czujnik.

Struktura *OSI::SensorView* wspiera pracę zarówno ze szczegółowym (niskopoziomowym), jak i z uproszczonym (wysokopoziomowym) modelem sensora. W pierwszym przypadku moduł percepcji (symulowany czujnik wraz z algorytmem wykrywania obiektów) powinien być odwzorowany w najdrobniejszych szczegółach. Wymaga się zatem zapisania w zbiorze *OSI::SensorView* możliwie jak największej liczby detali dotyczących aktualnego stanu scenariusza wirtualnej jazdy. Należy

więc wykorzystać jedną z trzech struktur zdefiniowanych w ramach *OSI::SensorView*, w zależności od rodzaju czujnika użytego w systemie percepcji: *OSI::RadarSensorView*, *OSI::CameraSensorView* lub *OSI::LidarSensorView*. Przy czym trzeba pamiętać o wyodrębnieniu z symulatora wszystkich wymaganych przez dany zbiór informacji. W przypadku radaru i lidar (ang. Light Detection and Ranging) będzie to trójwymiarowa chmura punktów oszacowana w trakcie procesu renderowania wirtualnej rzeczywistości przez tzw. algorytm śledzenia promieni. Z kolei w przypadku kamery jest to obraz widziany z perspektywy ustalonego aktora (finalny efekt renderowania). Natomiast w drugim przypadku moduł percepcji (reprezentowany jednoznacznie przez GSM) powinien bazować na informacjach dostępnych w strukturze *OSI::GroundTruth*, będącej wysokopoziomowym zapisem aktualnego stanu wirtualnego scenariusza. Zbiór *OSI::GroundTruth* można podzielić na następujące kategorie:

- dane o obiektach stacjonarnych: *OSI::StationaryObject*, *OSI::TrafficSign* i *OSI::TrafficLight*,
- informacje o obiektach ruchomych: *OSI::MovingObject*,
- konfiguracja sieci drogowej: *OSI::Lane*, *OSI::LaneBoundary* oraz *OSI::RoadMarking*,
- stan symulowanych warunków atmosferycznych: *OSI::EnvironmentalConditions*.

Informacje zawarte w strukturze *OSI::SensorView* zaprezentowano na schemacie blokowym pokazanym na rysunku 3.1. Schemat ten jest oczywiście uproszczony i nie reprezentuje w pełni poziomu skomplikowania rzeczywistego zbioru *OSI::SensorView*; tego typu rysunek poglądowy wydaje się być jednak wystarczający na potrzeby niniejszej rozprawy. Warto zaznaczyć, że przy użyciu pozycji, orientacji oraz wymiarów można dla danego obiektu jednoznacznie zdefiniować wspomnianą już wiele razy tzw. bryłę brzegową, czyli prostopadłościan opisany na tym obiekcie. Widać zatem, że w wysokopoziomowej wiadomości *OSI::GroundTruth* wszystkie elementy sceny, poza konfiguracją sieci drogowej (*OSI::Lane* i *OSI::LaneBoundary*) oraz symulowanymi warunkami atmosferycznymi (*OSI::EnvironmentalConditions*), reprezentowane są właśnie przy użyciu brył brzegowych. Co równie istotne, dane zapisane w strukturze *OSI::GroundTruth*, takie jak linie drogi czy pozycje i orientacje obiektów, wyrażone są w tzw. globalnym układzie współrzędnych (ang. global coordinate systems, GCS), którego początkiem jest pewien z góry ustalony punkt w trójwymiarowej przestrzeni wirtualnego scenariusza.



Rysunek 3.1. Schemat poglądowy prezentujący w uproszczony sposób zawartość struktury OSI::SensorView

3.3.3 Struktura OSI::SensorData

Struktura *OSI::SensorData* jest nośnikiem danych oszacowanych albo przez model sensora (zarówno niskopoziomowy, jak i wysokopoziomowy), albo przez symulowany moduł percepcji otoczenia; tutaj także wymaga się jednoznacznego zdefiniowania stempla czasowego oraz identyfikatora, pozycji i orientacji wirtualnego sensora.

W przypadku szczegółowego modelu czujnika do zapisu sztucznych pomiarów należy wykorzystać wiadomość *OSI::FeatureData*. Innymi słowy, niskopoziomowe, syntetyczne odczyty sensoryczne trzeba

w odpowiedni sposób wpisać do jednej z trzech niżej wymienionych struktur, w zależności od typu symulowanego czujnika:

- struktura *OSI::RadarDetectionData* (utworzona na podstawie *OSI::RadarSensorView*): tutaj przechowywane są dane o zbiorze wykryć oszacowanym przez model czujnika radarowego, z czego każde wykrycie charakteryzowane jest przez wartości (obliczone względem wirtualnego czujnika), takie jak: odległość radialna R , prędkość względna v_r , kąt θ w płaszczyźnie horyzontalnej pomiędzy obiektem a osią wiodącą radaru, skuteczna powierzchnia odbicia ρ (ang. radar cross section, RCS) oraz stosunek sygnału do szumu χ (ang. signal to noise ratio, SNR),
- struktura *OSI::CameraDetectionData* (utworzona na podstawie *OSI::CameraSensorView*): w tym zbiorze zapisuje się obiekty oszacowane na podstawie surowego obrazu z wirtualnej kamery; dla każdego elementu tego zbioru definiuje się jego identyfikator, typ obiektu oraz typ wygenerowanej bryły wraz z punktami tworzącymi tę bryłę (trójki wartości wyrażone w sferycznym układzie współrzędnych - promień R , azymut θ i elewacja ϕ),
- struktura *OSI::LidarDetectionData* (utworzona na podstawie *OSI::LidarSensorView*): ta struktura służy do zapisania zbioru wykryć wygenerowanych przez sztuczny czujnik lidarowy - podobnie jak w przypadku radaru jest to chmura punktów, choć tym razem wyrażona w sferycznym układzie współrzędnych, a więc każdy element zbioru reprezentowany jest przez promień R , azymut θ i elewację ϕ , wyliczone względem wirtualnego sensora.

Rozważmy teraz przypadek uproszczonego modelu sensora (GSM). Wejście do GSM to zbiór *OSI::GroundTruth*, gdzie w sposób uporządkowany (poprzez podział obiektów na klasy) przechowywane są wysokopoziomowe informacje o aktualnym stanie wirtualnego scenariusza. W ogólności zadaniem GSM jest ocena, czy dany element znajdujący się w strukturze *OSI::GroundTruth* (taki jak pas ruchu, krawędź jezdni, znak drogowy, itp.) może być wykryty przez wirtualny czujnik. Proces ten można uprościć do trzech głównych operacji:

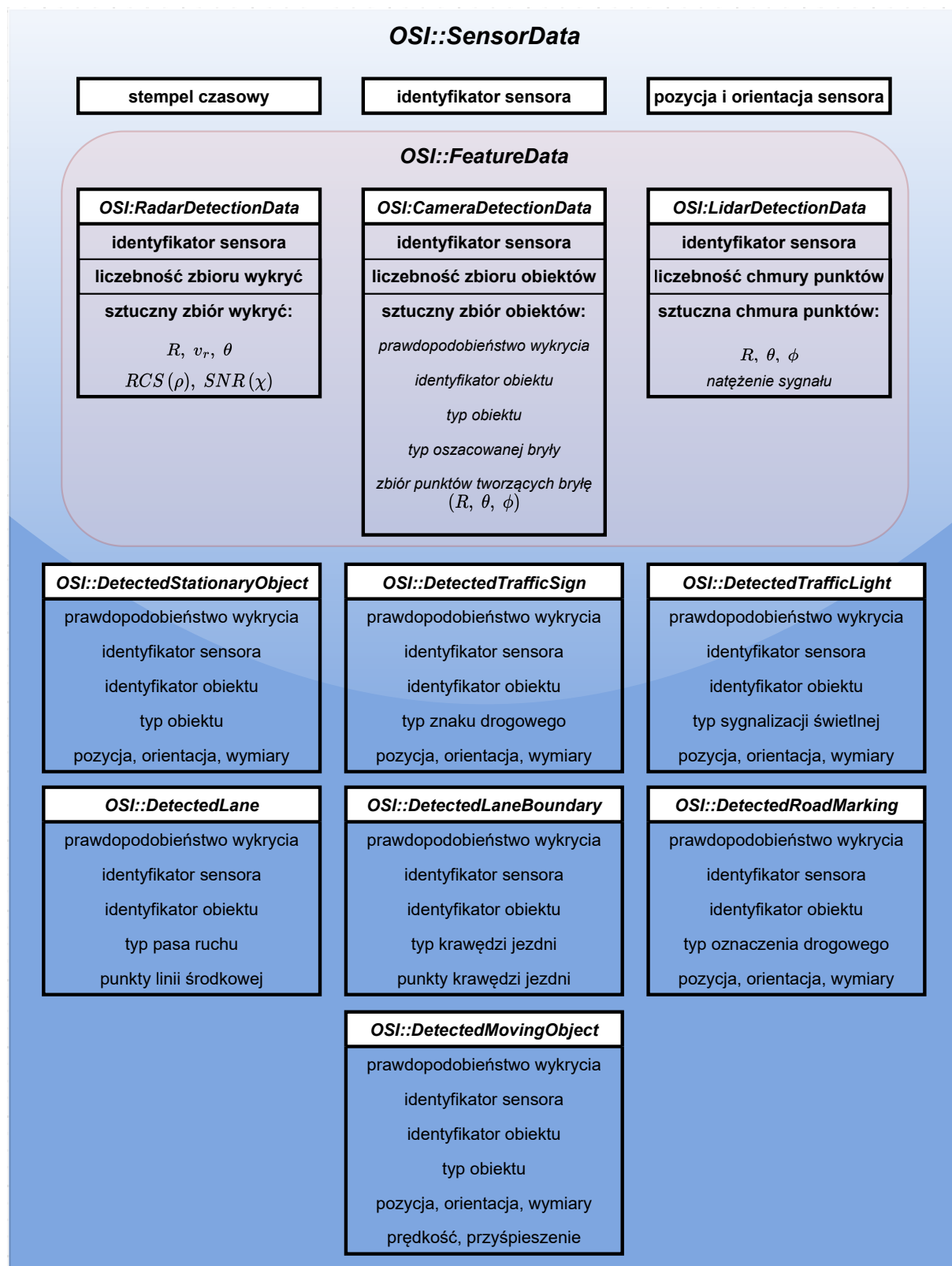
- przekształcenie pozycji i orientacji obiektu z globalnego układu współrzędnych do układu współrzędnych sensora, ustalonego względem położenia wirtualnego czujnika na pojeździe wiodącym,
- sprawdzenie, czy obiekt znajduje się w polu widzenia czujnika,
- zweryfikowanie, czy obiekt nie jest przysłonięty przez pozostałe elementy scenariusza.

Wykrycia wygenerowane przez model typu GSM należy również w odpowiedni sposób zapisać w strukturze *OSI::SensorData*. Mianowicie, jeśli dany komponent zbioru *OSI::GroundTruth* został uznany przez model sensora za widoczny (co oznacza, że w rzeczywistości algorytm wykrywania obiektów oszacowałby poprawnie bryłę brzegową dla tego celu na podstawie surowych pomiarów z czujnika), to rezultat (wykrycie) należy zapisać w odpowiadającej temu komponentowi strukturze. Powiązanie dla każdej klasy zdefiniowanej w zbiorze *OSI::GroundTruth* przedstawiono w tabeli 3.1.

Tabela 3.1. Odpowiedniki obiektów zdefiniowanych w zbiorze *OSI::GroundTruth* będące częścią struktury *OSI::SensorData*

Typ obiektu	Nazwa obiektu w <i>OSI::GroundTruth</i>	Odpowiednik obiektu w <i>OSI::SensorData</i>
Obiekt stacjonarny	<i>OSI::StationaryObject</i>	<i>OSI::DetectedStationaryObject</i>
Obiekt ruchomy	<i>OSI::MovingObject</i>	<i>OSI::DetectedMovingObject</i>
Znak drogowy	<i>OSI::TrafficSign</i>	<i>OSI::DetectedTrafficSign</i>
Sygnalizacja świetlna	<i>OSI::TrafficLight</i>	<i>OSI::DetectedTrafficLight</i>
Pas ruchu	<i>OSI::Lane</i>	<i>OSI::DetectedLane</i>
Krawędź jezdni	<i>OSI::LaneBoundary</i>	<i>OSI::DetectedLaneBoundary</i>
Oznaczenie jezdni	<i>OSI::RoadMarking</i>	<i>OSI::DetectedRoadMarking</i>

Oczywiście elementy wymienione w tabeli 3.1 nie są jedynie kopią informacji wejściowych ze zbioru *OSI::GroundTruth*. Po pierwsze, dane dotyczące pozycji i orientacji obiektu (albo bryły typu *bounding box*, albo punktów linii środkowej czy krawędzi jezdni) są przekształcone do układu współrzędnych sensora (SCS). Po drugie, wymienione wyżej struktury zawierają również parametry dotyczące samego wykrycia, takie jak: prawdopodobieństwo detekcji, wiek obiektu itp. Poglądowy schemat blokowy ukazujący dane dostępne w zbiorze *OSI::SensorData* zaprezentowano na rysunku 3.2.



Rysunek 3.2. Schemat poglądowy prezentujący w uproszczony sposób zawartość struktury OSI::SensorData

3.3.4 Modularyzacja symulacji

Wspomniano już, że funkcja typu ADAS testowana w wirtualnej rzeczywistości bazuje na informacjach oszacowanych przez symulowany moduł percepcji składający się w ogólności z modelu sensora oraz z algorytmu wykrywania obiektów. W rzeczywistości jednak liczba możliwych konfiguracji, przy użyciu których można wygenerować rzetelny zbiór danych wejściowych dla systemu klasy ADAS jest ogromna.

Urządzenie należące do kategorii ADAS może być oparte na informacjach zwróconych przez pojedynczy czujnik lub bazować na pomiarach generowanych przez wielosensorowy układ. Ponadto możliwe jest także wykorzystanie fuzji zarówno na poziomie surowych danych (z pojedynczym algorytmem wykrywania obiektów), jak i na poziomie listy obiektów (wiele algorytmów wykrywania obiektów i jeden algorytm fuzji).

Jak już napisano, *OSI::SensorView* należy traktować jedynie jako wejście do sztucznego modułu percepcji. Z kolei *OSI::SensorData* to struktura wykorzystywana do wymiany danych pomiędzy komponentami symulowanej percepcji otoczenia; *OSI::SensorData* może być również wyjściem z całego układu, czyli wejściem do funkcji typu ADAS testowanej w wirtualnej rzeczywistości. Standard OSI zaprojektowano tak, by możliwe było przeprowadzenie wirtualnej walidacji niezależnie od tego, jak skomplikowana jest wymiana informacji w rzeczywistym systemie percepcji (dostarczającym danych wejściowych dla urządzenia klasy ADAS) oraz na jakim poziomie abstrakcji generowane są syntetyczne pomiary. Wprowadźmy zatem kilka definicji, które posłużą jako narzędzie do omówienia przykładowych konfiguracji wymiany danych w symulatorze jazdy wykorzystywanym w procesie wirtualnej walidacji:

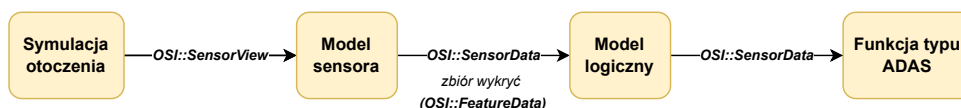
- symulacja otoczenia: to proces renderowania scenariusza jazdy przez środowisko wirtualne, którego efektem jest utworzenie w pełni funkcjonalnej wiadomości *OSI::SensorView*, czyli zgodnie ze schematem poglądowym zaprezentowanym na rysunku 3.1, dla każdego dyskretnego momentu czasu (identyfikowanego jednoznacznie przez pole *OSI::Timestamp*),
- model sensora: do tej kategorii zalicza się każdą procedurę przekształcającą wiadomość *OSI::SensorView* na wiadomość *OSI::SensorData*, w szczególności są to modele czujników opisane wyżej, zarówno niskopoziomowe, jak i wysokopoziomowe,
- model logiczny: do tej kategorii przynależą operacje, dla których wejściem oraz wyjściem jest struktura *OSI::SensorData*; przykładem takiego modelu może być algorytm wykrywania obiektów przekształcający sztuczne pomiary sensoryczne (zapisane w zbiorze *OSI::FeatureData*) na listę obiektów (elementy wymienione w tabeli 3.1) lub algorytm fuzji łączący informacje pochodzące z wielu wirtualnych czujników, czy też wygenerowane przez wiele algorytmów wykrywania obiektów,
- funkcja typu ADAS: to projektowany system, na którym przeprowadza się proces wirtualnej walidacji w trybie HIL albo SIL.

Wykorzystując przyjęte wyżej definicje omówmy pięć różnych wariantów, przy użyciu których można zaprojektować proces wirtualnej walidacji bazujący na standardzie OSI; konfiguracje te zaprezentowano także w formie graficznej na rysunkach od 3.3 do 3.7:

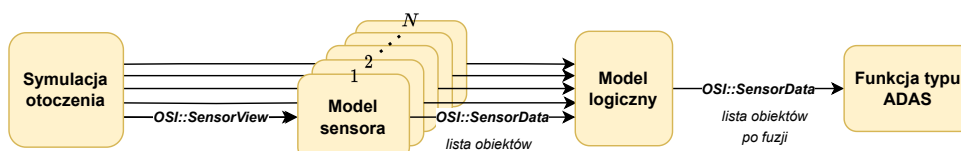
- (i) Wariant *A* (rysunek 3.3) - tutaj jedynym modulem odpowiedzialnym za przekształcenie wiadomości *OSI::SensorView* na wiadomość *OSI::SensorData* jest wysokopoziomowy model sensora, reprezentujący w jednoznaczny sposób sztuczny moduł percepcji, gdzie listę obiektów generuje się bezpośrednio na podstawie informacji zawartych w zbiorze *OSI::GroundTruth*.
- (ii) Wariant *B* (rysunek 3.4) - w tym przypadku symulacja modułu percepcji składa się z dwóch części, czyli z niskopoziomowego modelu czujnika, generującego syntetyczny zbiór pomiarów sensorycznych na podstawie danych dostępnych w strukturze *OSI::SensorView*, oraz z modelu logicznego, realizującego procedurę wykrywania obiektów z wykorzystaniem sztucznych odczytów zawartych wiadomości *OSI::FeatureData*.
- (iii) Wariant *C* (rysunek 3.5): tutaj zakłada się, że rzeczywisty system percepcji korzysta z pomiarów z N czujników samochodowych ($N > 1$) - w tej konfiguracji procedura generacji syntetycznych danych dla symulowanej funkcji typu ADAS wykorzystuje N wysokopoziomowych modeli sensora i pojedynczy model logiczny reprezentujący proces fuzji danych, zamieniający N list obiektów w jedną listę obiektów (jedną wiadomość *OSI::SensorData*) łączącą informacje pochodzące z wielu źródeł pomiarowych.
- (iv) Wariant *D* (rysunek 3.6) - w tej konfiguracji (również odwzorowującej system percepcji mający na wejściu N różnych czujników samochodowych) moduł generacji syntetycznych danych składa się z: N niskopoziomowych modeli sensora, N modeli logicznych realizujących procedurę tworzenia listy obiektów dla każdego z wygenerowanych zbiorów wykryć (zapisanych w N strukturach *OSI::FeatureData*) oraz z jednego modelu logicznego reprezentującego operację fuzji uzyskanych list obiektów.
- (v) Wariant *E* - tutaj, podobnie jak w konfiguracji *D*, mamy do dyspozycji N niskopoziomowych modeli sensora, jednak w tej sytuacji symulowany moduł percepcji realizuje procedurę fuzji na poziomie zbioru wykryć, a zatem w tym schemacie generacji danych syntetycznych korzysta się jeszcze z modelu logicznego łączącego N unikalnych struktur *OSI::FeatureData* w jedną strukturę *OSI::FeatureData* oraz z modelu logicznego reprezentującego procedurę wykrywania obiektów na podstawie informacji z łączonego zestawu *OSI::FeatureData*.



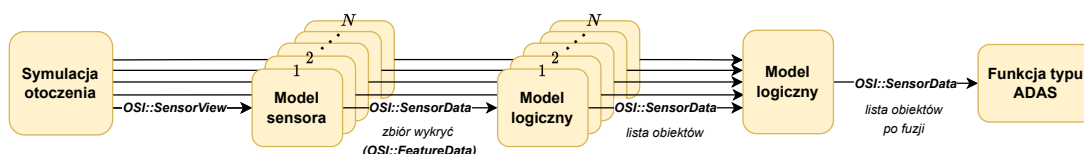
Rysunek 3.3. Wirtualna walidacja przy wsparciu standardu OSI - wariant *A* oparty na wysokopoziomowym modelu sensora



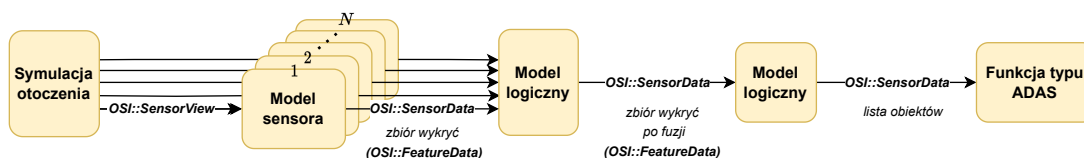
Rysunek 3.4. Wirtualna walidacja przy wsparciu standardu OSI - wariant *B* bazujący na niskopoziomowym modelu sensora oraz na modelu logicznym, realizującym procedurę wykrywania obiektów



Rysunek 3.5. Wirtualna walidacja przy wsparciu standardu OSI - wariant *C* mający na celu odwzorowanie przypadku, w którym system percepcji wykorzystuje dane pochodzące z wielu czujników samochodowych; występuje tutaj N wysokopoziomowych modeli sensora oraz jeden model logiczny dokonujący fuzji informacji z N źródeł w jedną, łączoną listę obiektów



Rysunek 3.6. Wirtualna walidacja przy wsparciu standardu OSI - wariant *D* dla modułu percepcji korzystającego z N czujników samochodowych; strumień przetwarzania danych korzysta z: N niskopoziomowych modeli sensora generujących unikalne instancje struktury *OSI::FeatureData*, N modeli logicznych generujących listę obiektów dla każdego z uzyskanych zbiorów wykryć oraz z jednego modelu logicznego dokonującego fuzji danych, której rezultatem jest łączony zestaw obiektów



Rysunek 3.7. Wirtualna walidacja przy wsparciu standardu OSI - wariant *E* dla symulowanego modułu percepcji odwzorowującego system korzystający z N czujników samochodowych: w tym schemacie wpierv jeden model logiczny realizuje procedurę fuzji zbiorów wykryć wygenerowanych przez N niskopoziomowych modeli sensora (zapisanych w unikalnych strukturach *OSI::FeatureData*), z kolei później drugi model logiczny wykonuje operację wykrywania obiektów na bazie łączonego zestawu danych dostępnego w zestawie *OSI::FeatureData*

Podsumowując, standard ASAM OSI zaprojektowano z myślą o usprawnieniu i uwierzytelnieniu procesów testowania i weryfikacji zaawansowanych systemów wspomagania kierowcy w środowiskach wirtualnej jazdy. OSI dzieli symulację na cztery poziomy abstrakcji: generacja scenariusza testowego, model sensora, model logiczny oraz funkcja typu ADAS. Ponadto, by umożliwić precyzyjną wymianę informacji pomiędzy wspomnianymi modułami symulatora, OSI definiuje także dwie obszerne, spójne i przemyślane struktury danych: *OSI::SensorView* oraz *OSI::SensorData*. Dzięki takiemu podejściu możliwe jest zaprojektowanie niezwykle rzetelnego systemu do generacji syntetycznych pomiarów, wykorzystywanego w procesie wirtualnej walidacji funkcji typu ADAS, zachowującego wysoką jakość odwzorowania prawdziwych odczytów niezależnie od poziomu skomplikowania rzeczywistego systemu percepcji. Co zatem niezwykle ważne w kontekście niniejszej dysertacji, model matematyczny czujnika radarowego projektowany pod kątem wirtualnej walidacji układów klasy ADAS powinien wspierać wymianę danych przy użyciu standardu OSI. Może to być model zarówno niskopoziomowy (generujący syntetyczny zbiór wykryć na podstawie informacji zawartych w strukturze *OSI::RadarSensorView*), jak i wysokopoziomowy (wytwarzający syntetyczne pomiary, bazując na danych zapisanych w zbiorze *OSI::GroundTruth*); istotne jednak, by taki sztuczny czujnik radarowy był zgodny z definicją modelu sensora wprowadzoną przez standard OSI - a więc takiej procedury, dla której wejście to struktura *OSI::SensorView*, natomiast wyjście to struktura *OSI::SensorData*.

4 Modele matematyczne czujników radarowych

Rozdział ten jest przeglądem modeli czujników radarowych opisanych w literaturze. Podzielone one zostaną, jak to już zostało wspomniane, na dwie kategorie: bazujące na danych oraz geometryczne. Najbardziej interesujące z perspektywy niniejszej rozprawy są symulacje sensorów generujące sztuczny zbiór wykryć, czyli moduły wykorzystywane wraz z rzeczywistym algorytmem wykrywania obiektów w podstawowej i zarazem najdokładniejszej konfiguracji HIL oraz SIL (rysunki 2.11 i 2.10). Niemniej jednak, by umożliwić czytelnikowi głębsze zrozumienie omawianej tematyki, zaprezentowane będą także modele zrealizowane na poziomie listy obiektów, użyte w uproszczonej wersji wirtualnej walidacji (rysunki 2.13 oraz 2.12). Warto także zaznaczyć, że do celu porównywania modeli ze sobą wykorzystane będą pojęcia wiarygodności, wydajności i użyteczności modelu matematycznego czujnika radarowego, które wprowadzono w rozdziale 1.

4.1 Modelowanie na poziomie wykryć

4.1.1 Modele bazujące na danych

Przełomowe rozwiązanie należące do tej kategorii ukazało się w 2017 roku [71]. Wtedy to Tim A. Wheeler wraz z zespołem opublikował tzw. stochastyczny model czujnika radarowego bazujący na głębokich sieciach neuronowych [72] oraz na autoenkoderach wariacyjnych [73, 74]. Autorzy modelu zdecydowali się zaprezentować zbiór wykryć radarowych w postaci siatki wyrażonej w układzie polarnym; wartością przypisaną do elementu o ustalonych współrzędnych jest w tym przypadku oszacowana moc odebranego sygnału. Do grupy uczącej zakwalifikowano dwie kategorie informacji: rzeczywiste pomiary radarowe oraz listę brył brzegowych wygenerowaną przy użyciu algorytmu wykrywania obiektów. Model ten jest niewątpliwie wydajny, dzięki czemu może być wykorzystany w symulacjach typu HIL. Niestety, nie udowodniono wiarygodności tego wirtualnego sensora, ponieważ rzetelność procesu generacji sztucznych wykryć radarowych zweryfikowano na zbiorze danych o umiarkowanej różnorodności.

Nieco inaczej, bo w oparciu o tzw. model mieszanin rozkładów Gaussa [75], podeszli do problemu zaprojektowania wirtualnego sensora Alexander Scheel oraz Klaus Dietmayer [67]. Metoda

ta odznacza się znacznie wyższym stopniem wiarygodności, ponieważ rzeczywiste pomiary są wiernie odwzorowywane również w przypadku bardziej rozbudowanych scenariuszy. Jako wadę tego podejścia należy wskazać nieuwzględnienie obiektów statycznych w grupie uczącej. Ponadto, opisany prototyp został zaprojektowany na potrzeby algorytmu wykrywania obiektów, jako moduł przetwarzający surowe dane radarowe. Nie jest więc typowym modelem sensora, gdzie sztuczne wykrycia generowane są podstawie informacji niemających związku z rzeczywistymi pomiarami zwracanymi przez radar. Opisana praca była niewątpliwie inspiracją dla Jasmina Eberta, który opracował podobne rozwiązanie, tym razem bazujące na głębokich sieciach neuronowych [76].

Niezwykłe skuteczny model czujnika radarowego należący do omawianej kategorii zaprezentował także zespół prowadzony przez Filipa Ciepiałę i Mariusza Nowaka [77]. Grupę uczącą (rzeczywiste pomiary radarowe oraz lista brył brzegowych wygenerowana dla pojazdów, dla których istnieje co najmniej jedno wykrycie) zaczerpnięto z darmowej bazy *nuscenes*, opracowanej przez firmę Motional [78]. Warto zaznaczyć, że niezwykle obszerną analizę danych radarowych dostępnych w bazie *nuscenes* opublikowano w [79]. Wejściem do modelu są dane dotyczące prędkości oraz orientacji pojazdu wiodącego, na którym zamocowano radar, oraz wysokopoziomowe informacje o obiektach w scenariuszu, takie jak: pozycja, orientacja, prędkość, wymiary oraz typ danego celu. Z kolei wyjście z modelu to zbiór wykryć radarowych. Autorzy udowodnili rzetelność algorytmu, porównując uzyskane wyniki z rezultatami zwróconymi przez przykładowy model probabilistyczny. Niemniej jednak, do uznania zaprezentowanego rozwiązania za wiarygodne konieczne jest wyuczenie modelu również dla obiektów stacjonarnych.

Interesujący algorytm opracowała też w 2019 roku kilkusobowa grupa na czele z Thomasem Ederem [80]. Do wygenerowania syntetycznych pomiarów autorzy wykorzystali potencjał generatywnych sieci przeciwstawnych (ang. generative adversarial network, GAN) [81]. Niezaprzeczalnym faktem jest zdolność modeli bazujących na architekturze typu GAN do generowania niezwykle realistycznych struktur [82–84]. Analogiczny, bo również bazujący na sieciach neuronowych klasy GAN, wirtualny czujnik radarowy opublikował także, choć rok później, Rob Wenston [85]. Niestety, poprawne wyuczenie tego typu sztucznej sieci neuronowej to niebywale skomplikowany problem wymagający rozbudowanej i zróżnicowanej grupy uczącej.

4.1.2 Modele geometryczne

Syntetyczne dane radarowe można uzyskać poprzez przekształcenie informacji zwróconych przez symulator jazdy przy użyciu z góry ustalonych operacji. Prace należące do tej grupy dzielą się na dwie kategorie: przeprowadzające dokładną symulację procesu propagacji fali elektromagnetycznej oraz wykorzystujące koncepcję centrów rozpraszania do uzyskania sztucznego zbioru wykryć.

Pionierem w dziedzinie efektywnego odwzorowania rozchodzenia się sygnału elektromagnetycznego w przestrzeni jest Hao Ling [86]. W publikacji udostępnionej w 1989 roku opisano metodę tzw. strzelania i odbijania promieni (ang. shooting and bouncing rays, SBR), będącą niewątpliwie bazą teoretyczną dla współczesnych technik śledzenia wiązek. W podejściu tym falę radiową wyrażono w postaci zestawu punktów gęsto rozmieszczonych w trójwymiarowej przestrzeni. Każdy z elementów zbioru przetwarza się oczywiście w niezależny sposób, analizując jego ścieżkę od wirtualnego czujnika do obiektów zdefiniowanych w scenariuszu. Metoda ta jest niezwykle dokładna, gdyż przy wyznaczaniu miejsc przecięcia promieni z celami znajdującymi się w polu widzenia sensora uwzględnia się efekt wielowiązkowości [12], wraz z oszacowaniem siły rozproszonego sygnału. Wada algorytmu natomiast to spora złożoność obliczeniowa.

Ważnym kamieniem milowym w tejże problematyce są rezultaty opublikowane w 1998 roku przez trzysobową grupę prowadzoną przez Chang-Fa Yang [87]. Zespół zaprojektował i wdrożył skuteczną symulację rzeczywistego systemu radarowego w oparciu o zmodyfikowaną wersję techniki SBR.

Nieco później, bo w 2013 roku, Demetrio Gubelli zaprezentował w pełni funkcjonalny model radaru samochodowego bazujący na metodologii śledzenia promieni [88]. Algorytm ten opiera się na informacjach zwracanych przez środowisko wirtualnej jazdy i z wykorzystaniem wspomnianej metodologii śledzenia promieni oblicza odległość oraz prędkość względną dla punktów widocznych z perspektywy sensora.

Za najważniejszą w tej dziedzinie należy jednak wskazać pracę opublikowaną cztery lata później przez zespół Nilsa Hirsenkorna [89]. Kluczowe osiągnięcia tej grupy badawczej to spełnienie warunku czasu rzeczywistego dla wirtualnego sensora zbudowanego w oparciu o technikę SBR oraz zweryfikowanie modelu na podstawie pomiarów z rzeczywistego radaru typu FMCW.

W ostatnich kilku latach do symulacji bazujących na metodologii śledzenia promieni dodano jeszcze kilka istotnych usprawnień, w szczególności: zmniejszenie złożoności obliczeniowej [90–92], uwzględnienie zmieniającej się pozycji i orientacji wirtualnego czujnika [93] oraz różnych warunków pogodowych [94] w wygenerowanym sztucznym zbiorze danych czy też obsługę wielowiązkowych systemów radarowych [95–97].

Inną kategorią modeli geometrycznych są modele czujników radarowych oparte na koncepcji centrów rozpraszania, gdzie pomija się bezpośrednią symulację propagacji fali elektromagnetycznej w przestrzeni. Jak zostało wspomniane w rozdziale 1, centra rozpraszania to punkty zlokalizowane na danym obiekcie, które w przybliżony sposób reprezentują rozproszenie fali elektromagnetycznej odbitej od tego obiektu. Innymi słowy, sztuczny zbiór wykryć radarowych jest w tego typu modelach wyrażony w postaci odpowiednio wygenerowanego zestawu centrów rozpraszania. Takie dane syntetyczne nie odzwierciedlają oczywiście niepewności charakterystycznych dla rzeczywistych systemów radarowych, dlatego często w modelach tej klasy uwzględnia się efekt filtracji spowodowany przez proces przetwarzania sygnałów [98, 99] oraz wpływ zmiennych warunków atmosferycznych [100].

Pionierami w dziedzinie generacji centrów rozpraszania są Rajan Bhalla i Hao Ling, którzy opracowali metodę wyłuskania najważniejszych punktów z gęstej grupy pomiarów uzyskanej przy pomocy techniki SBR [20]. Niezwykle ważne okazały się także rezultaty opublikowane w 2006 roku przez zespół Karin Schuler [101]. Badaczom udało się wykazać, przy użyciu danych z rzeczywistego systemu radarowego, że odpowiednio utworzony zbiór centrów rozpraszania wiarygodnie odwzorowuje odbicia fali elektromagnetycznej. Niezwykle istotną kwestią jest tutaj fakt, iż zestaw centrów rozpraszania uzyskany dla przykładowego samochodu osobowego zawiera zaledwie kilkanaście punktów, zlokalizowanych głównie na narożnikach i kołach pojazdu.

Właściwość tę wykorzystali Markus Bühren oraz Bin Yang w niezwykle innowacyjnym modelu czujnika radarowego [102, 103]. Cel tejże symulacji to wygenerowanie sztucznych pomiarów właśnie dla samochodów osobowych, z czego każdy pojazd określono przy użyciu dwunastu punktów, zlokalizowanych na narożnikach, kołach i karoserii tego obiektu. Algorytm wpierw przeprowadza serię przekształceń geometrycznych w celu stwierdzenia, które z punktów są widoczne z perspektywy wirtualnego sensora. Następnie, dla każdego wykrytego (nieprzysłoniętego) centrum rozpraszania oszacowuje się siłę sygnału odbitego od tego celu i jeśli obliczona wartość jest większa od ustalonego progu, to punkt uznaje się za sztuczne wykrycie radarowe.

Podobną, choć nieco bardziej rozszerzoną symulację opublikowali w 2019 roku trzech badacze: Adam Martowicz, Alberto Gallina i Grzegorz Karpiel [21]. Szkielet rozwiązania jest w zasadzie identyczny, to znaczy wpierw wyznacza się surowy zbiór pomiarów, a za syntetyczne wykrycie uznaje się centrum rozpraszania o mocy sygnału wyższej od progu odcięcia. W tym wirtualnym sensorze informacją wejściową wykorzystywaną do wygenerowania sztucznego zbioru wykryć dla danego obiektu nie jest jednak sztywno ustalony, kilkunastoelementowy zbiór punktów zlokalizowanych na tym obiekcie. Zamiast tego pojazdy podzielono na kilka kategorii i dla każdego typu utworzono zaawansowaną strukturę geometryczną, z której generuje się centra rozpraszania. Dzięki takiemu podejściu zwiększa się znacznie losowość uzyskiwanego sztucznego zbioru danych. Model ten różni się od swojego poprzednika także testem statystycznym, mającym na celu pozbycie się części punktów pomiarowych. Tutaj wnioskowanie jest dynamiczne i bazuje na hipotezie Neymana-Pearsona [104]. Podsumowując, oba zaproponowane modele matematyczne czujników radarowych są niezwykle skuteczne. Za największe zalety dwóch wyżej omówionych rozwiązań należy wskazać zdolność do pracy w czasie rzeczywistym oraz fakt, iż syntetyczne wykrycia generowane dla pojazdów charakteryzują się wysokim poziomem rzetelności. Niestety, brak wsparcia dla sceny statycznej (np. barier, sygnalizacji świetlnej czy znaków drogowych) powoduje, że wspomniane modele nie mogą być uznane za wiarygodne.

4.2 Modelowanie na poziomie listy obiektów

4.2.1 Modele bazujące na danych

Ta kategoria modeli jest niezwykle popularna wśród badaczy. Analizując dostępne teksty widać, że prace oparto zarówno na modelach probabilistycznych, jak i na sztucznych sieciach neuronowych.

Początkowo sztuczną listę obiektów generowano na podstawie prostych, parametryzowanych funkcji [105–107], gdzie nie są wymagane informacje z dokładnego czujnika referencyjnego. Niestety, stosunkowo niewielka wszechstronność tego typu rozwiązań skłoniła naukowców do opracowania nieco bardziej skomplikowanych zależności, konfigurowalnych bezpośrednio na podstawie danych z sensorów. Niezwykle skuteczną symulację na potrzeby testów urządzeń klasy ADAS w czasie rzeczywistym zaprezentowała grupa badaczy na czele z Nilsem Hirsenkornem - wstępną wersję opublikowano w 2015 roku [108], a rok później ją usprawniono [109]. Trzon tego rozwiązania to model statystyczny zbudowany w oparciu o jądrowy estymator gęstości [110, 111]; jako zmienną losową przyjęto tutaj pozycję wykrytego obiektu. Autorzy uwzględnili w wirtualnym sensorze takie czynniki, jak: autokorelacja oszacowanego położenia oraz pole widzenia czujnika. Funkcję gęstości prawdopodobieństwa wyznaczono na podstawie pomiarów z rzeczywistego systemu radarowego. Niestety, mimo że zarejestrowany zbiór danych składa się z niemal stu tysięcy próbek, to model jest nadal podatny na efekt tzw. nadmiernego dopasowania. Pracą opisaną w [109] inspirował się Hexuan Li [112], który wraz z zespołem zaproponował model radaru bazujący na sieciach neuronowych MDN (ang. mixture density network) zdolnych do odwzorowania warunkowych rozkładów prawdopodobieństwa [113].

Nieco inaczej do problemu uzyskania syntetycznej listy obiektów podszedł zespół prowadzony przez Edvina L. Zec [114]. Autorzy skoncentrowali się na odwzorowaniu w symulacji błędów generowanych przez samochodowe czujniki radarowe. Wejście do algorytmu to idealna lista obiektów, w skład której wchodzi tylko bryły widoczne z perspektywy wirtualnego czujnika (patrz rysunek 2.7). Celem jest tutaj generacja szumu specyficznego dla rzeczywistego sensora w pozycji wzdłużnej oraz poprzecznej symulowanego obiektu. Niezwykle istotnym i przełomowym założeniem, które przyjęto w tejże pracy, to potraktowanie błędów pomiarowych jako ciągu zależnych od siebie próbek. Do opracowania tej skomplikowanej zależności wykorzystano ukryty model Markowa z autoregresyjnym modułem wejścia i wyjścia (ang. autoregressive input-output hidden Markov model, AIOHMM). Tę także kluczową symulację sensora usprawniono już rok później, czyli w 2019 roku. Mianowicie, Henrik Arnelid zdecydował się na zastąpienie algorytmu AIOHMM rekurencyjną, warunkowo sterowaną generatywną siecią przeciwną (ang. recurrent conditional generative adversarial network, RCGAN) [115]. Wyniki zaprezentowane w obu wspomnianych pracach należy uznać za imponujące. Niestety, jak już wspomniano, przeprowadzenie efektywnego procesu uczenia zarówno w przypadku AIOHMM, jak i

RCGAN jest niezwykle skomplikowane i wymaga ogromnej liczby danych z czujników rzeczywistych, zarejestrowanych w trakcie jazdy testowej.

4.2.2 Modele geometryczne

Najprostszą, choć niezwykle użyteczną symulacją sensora będzie tzw. uniwersalny model czujnika (ang. generic sensor model, GSM), realizujący funkcjonalność idealnego algorytmu wykrywania obiektów. GSM otrzymuje z symulatora jazdy informację o wszystkich bryłach brzegowych zdefiniowanych dla danego scenariusza (zarówno dla obiektów ruchomych, jak i stacjonarnych) i wybiera tylko te widoczne z perspektywy wirtualnego czujnika. Pomijając stosunkowo niewielki poziom wiarygodności, to niewątpliwie ogromną zaletą tego typu podejścia jest determinizm uzyskiwanych wyników, gdyż korzysta się tutaj tylko i wyłącznie z przekształceń geometrycznych. Tego typu modele są bardzo przydatne, np. w procesie uczenia sieci neuronowych ze wzmocnieniem (ang. reinforcement learning) [116] oraz do analizy krytycznych z punktu widzenia bezpieczeństwa użytkownika ruchu drogowego sytuacji [117]

GSM opisane w literaturze składają się z dwóch głównych operacji: uwzględnienia pola widzenia rzeczywistego sensora oraz obsłużenia efektu wzajemnego przysłaniania się obiektów. Ciekawe rozwiązanie o niesłychanie niskiej złożoności obliczeniowej opublikowali w 2018 roku Michael Stolz i Georg Nestlinger [118]. Autorzy nakreślili użyteczność tego algorytmu w kontekście wirtualnej walidacji systemów typu ADAS o wysokim poziomie automatyzacji. Niestety, wadą zaproponowanego podejścia jest obsługa pola widzenia czujnika oraz przysłonięć tylko w dwuwymiarowej przestrzeni, w stosunkowo ograniczony sposób, czyli poprzez sprawdzenie odpowiednich zakresów kątowych. Dużo dokładniej, bo z wykorzystaniem techniki rzutowania promieni, choć nadal na płaszczyźnie, koncepcję GSM zrealizował zespół prowadzony przez Stefana Muckenhubera [119].

Metodologię GSM rozszerzono do trójwymiarowej przestrzeni (zgodnie ze standardem ASAM OSI) w 2019 w patencie autorstwa Marcina Piątka i Kamila Lelowicza [120]. Do obsługi przysłonięć i pola widzenia wirtualnego czujnika również wykorzystano algorytm rzutowania promieni. Jak udowodniono w publikacji udostępnionej w 2020 roku, model ten umożliwia pracę symulacji w czasie rzeczywistym. Co ciekawe, w 2021 roku opublikowano rozszerzenie do tego modelu, dzięki któremu w procesie likwidacji przysłoniętych obiektów brane są pod uwagę wzniesienia drogi [121], co zdecydowanie zwiększyło wiarygodność i użyteczność omawianego GSM.

5 Formalizacja referencyjnego modelu matematycznego czujnika radarowego pod kątem standardu ASAM OSI

Rozważania zaprezentowane w dysertacji w znacznej mierze opierają się na modelu czujnika radarowego opublikowanego w 2019 roku przez zespół Adama Martowicza [21]. W bieżącym rozdziale zostaną omówione koncepcje i operacje geometryczne, na podstawie których uzyskiwane są sztuczne pomiary radarowe. Teoria zaproponowana przez autorów modelu będzie jednak rozszerzona i sformalizowana pod kątem wymagań standardu ASAM OSI. Po pierwsze, zdefiniowane będzie w sposób jednoznaczny wejście do modelu oraz wyjście z modelu. Po drugie, procedura przetwarzania informacji będzie rozbudowana o przekształcenia pomiędzy różnymi układami współrzędnych określonymi w standardzie ASAM OSI. Po trzecie, zaproponowana zostanie kompaktowa definicja sztucznego zbioru wykryć uwzględniająca w ramach jednej formuły wszystkie operacje właściwe dla omawianego modelu czujnika radarowego.

W kolejnych częściach pracy przedstawione będą modyfikacje wprowadzone do modelu, mające na celu poprawę kluczowych czynników wpływających na użyteczność danych symulacyjnych wykorzystywanych jako źródło wejściowe do testów systemów typu ADAS w środowiskach wirtualnej jazdy, czyli podniesienie poziomu wiarygodności procesu odwzorowania rzeczywistych odczytów radarowych.

5.1 Założenia modelu referencyjnego

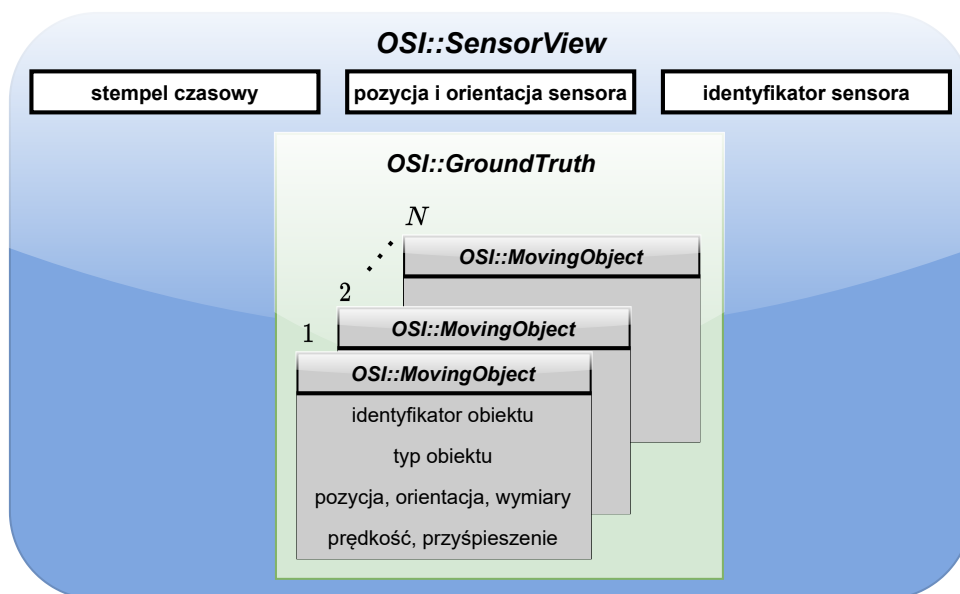
Model symulacyjny czujnika radarowego będący obiektem tych rozważań [21], zwany dalej modelem sensora, generuje sztuczny zbiór wykryć, bazując na koncepcji centrów rozpraszania. Każdy z wygenerowanych ośrodków reprezentuje - w pewien uproszczony sposób - miejsce odbicia fali elektromagnetycznej od danego celu. Pojedyncze centrum rozpraszania jest zatem bezpośrednio wykorzystane do wyliczenia parametrów jednego, syntetycznego wykrycia radarowego, a więc: odległości (R), względnej prędkości radialnej (v_r) oraz kątów w płaszczyźnie horyzontalnej (θ) i wertykalnej (ϕ) pomiędzy osią wiodącą radaru a danym punktem (patrz rysunek 2.1). Dodatkowo w modelu sensora uwzględniono także efekt filtracji charakterystyczny dla systemów radarowych

typu FMCW, gdzie niektóre z uzyskanych pomiarów, zakwalifikowane jako fałszywe na podstawie adaptacyjnego progu odcięcia obliczonego przy użyciu mocy zarejestrowanego sygnału, usuwa się z finalnego zbioru wykryć [58].

Przypomnijmy, że zbiór centrów rozpraszania jest pewnym przybliżeniem rozproszenia fali elektromagnetycznej od obiektów znajdujących się w polu widzenia radaru. Dlatego rozmieszczenie punktów tego zbioru w trójwymiarowej przestrzeni powinno pokrywać się z miejscami, które statystycznie najczęściej są źródłem odbicia o największej sile sygnału. Wynika z tego, iż każdy obiekt wykrywalny przez czujnik radarowy powinien posiadać charakterystyczny dla siebie zestaw centrów rozpraszania. Autorzy modelu sensora skupili się na odwzorowaniu tego typu zbiorów jedynie dla pojazdów, pomijając wszelkie obiekty statyczne, w tym barierki, słupki, sygnalizację świetlną, znaki drogowe itp. Stanowi to oczywiście pewne uproszczenie wpływające negatywnie na wiarygodność modelu w kontekście wirtualnej walidacji układów klasy ADAS.

5.2 Wejście do modelu zgodne ze standardem ASAM OSI

Jak opisano w rozdziale 3, standard ASAM OSI dzieli proces wirtualnej walidacji systemów typu ADAS na kilka poziomów abstrakcji: symulację otoczenia, model sensora, model logiczny oraz funkcję klasy ADAS. Dlatego też, by omawiany model sensora zgadzał się z definicją tegoż właśnie poziomu abstrakcji, zdefiniowanego w ramach wspomnianego standardu, za wejście do modelu należy uznać strukturę *OSI::SensorView*, będącą wysokopoziomowym opisem aktualnego stanu wirtualnego scenariusza, natomiast wyjście z modelu trzeba zamieścić w zbiorze *OSI::SensorData*, w którym zapisywane są sztuczne pomiary sensoryczne. Jak przedstawiono wyżej, aktualna wersja modelu sensora wspiera generację sztucznych wykryć radarowych jedynie dla pojazdów. Co za tym idzie, głównym wejściem do modelu będą obiekty klasy *OSI::MovingObject* o typie *OSI::TYPE_VEHICLE* [122] zawarte w wiadomości *OSI::SensorView*. Schemat poglądowy prezentujący w uproszczony sposób minimalną zawartość wiadomości *OSI::SensorView* wymaganą przez model sensora pokazano na rysunku 5.1 stanowiącym modyfikację rysunku 3.1.

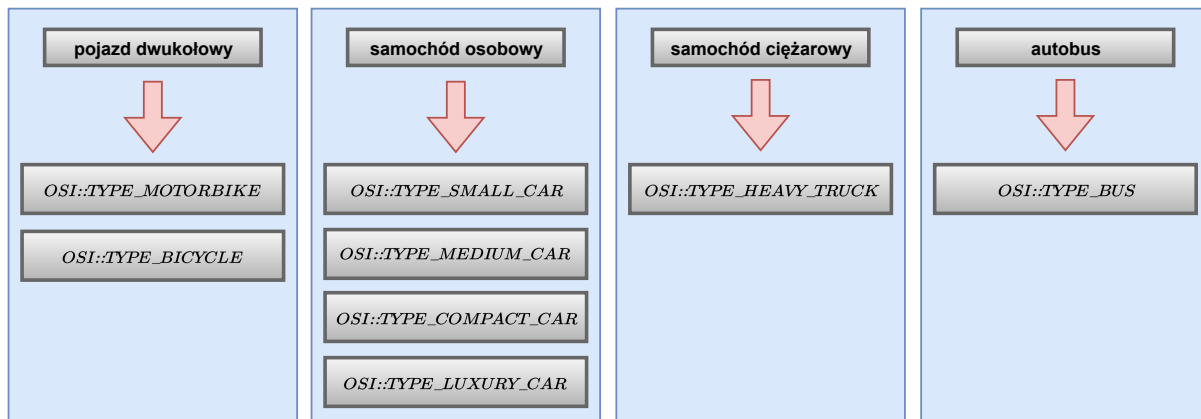


Rysunek 5.1. Schemat poglądowy prezentujący w uproszczony sposób fragment struktury *OSI::SensorView* zawierający N ($N \in \mathbb{N}$) obiektów klasy *OSI::MovingObject*, wymagany do wygenerowania przez model sensora poprawnej wiadomości typu *OSI::SensorData* mieszczącej sztuczny zbiór wykryć radarowych

Drugim wejściem do wirtualnego czujnika jest baza modeli geometrycznych pojazdów, będąca podstawą procesu generacji zbioru centrów rozpraszania, gdyż w bazie tej zdefiniowane są punkty charakterystyczne dla danej klasy pojazdów samochodowych, które z kolei traktuje się jako źródła potencjalnych odbić. Symulatory jazdy wykorzystywane w przemyśle motoryzacyjnym reprezentują zwykle pojazdy samochodowe jako zbiory połączonych ze sobą trójkątów, wyrażonych w trójwymiarowej przestrzeni. Celem realistycznej symulacji jest jak najwierniejsze odwzorowanie rzeczywistych konturów obiektów będących częścią scenariusza wirtualnej jazdy, dlatego tego typu modele geometryczne mogą składać się nawet z kilkudziesięciu tysięcy wielokątów [123]. Jednak, jak pokazują badania z dziedziny centrów rozpraszania, zbiór punktów charakterystycznych - dla których statystycznie najczęściej będzie pojawiać się wykrycie oszacowane przez czujnik radarowy - można w przypadku pojazdów samochodowych ograniczyć do zaledwie kilkudziesięciu elementów, zlokalizowanych głównie na narożnikach i kołach pojazdu [101].

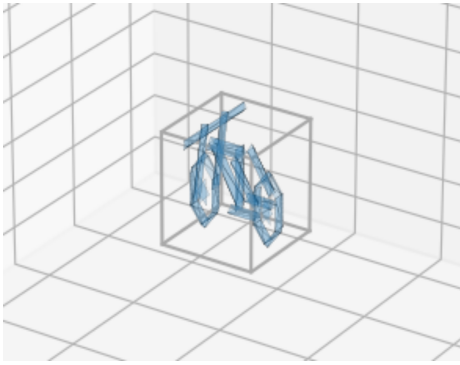
Jak wspomniano w rozdziale 1, do spełnienia warunku użyteczności modelu symulacyjnego czujnika radarowego w kontekście wirtualnej walidacji układów klasy ADAS kluczowe jest osiągnięcie kompromisu pomiędzy ograniczeniem złożoności obliczeniowej symulacji a zachowaniem wysokiego poziomu wiarygodności modelu. Dlatego też twórcy modelu sensora, mając na uwadze wspomniane wyżej właściwości centrów rozpraszania, charakterystyczne dla pojazdów samochodowych, zdecydowali się na utworzenie uproszczonych brył geometrycznych dla kilku

najważniejszych klas pojazdów, uwzględniając krzywizny i kształty specyficzne dla danej kategorii. W szczególności wzięto pod uwagę: pojazd dwukołowy, samochód osobowy, autobus oraz samochód ciężarowy [21]. Standard ASAM OSI udostępnia jednak znacznie szerszą gamę kategorii dla obiektów klasy *OSI::MovingObject* o typie *OSI::TYPE_VEHICLE* określoną w ramach struktury *OSI::VehicleClassification*. W ramach tej pracy proponuje się następujące powiązanie pokazane na rysunku 5.2, które w zależności od charakterystyki testu przeprowadzanego w środowisku wirtualnej jazdy można rozszerzyć o pozostałe typy dostępne w zbiorze *OSI::VehicleClassification*.

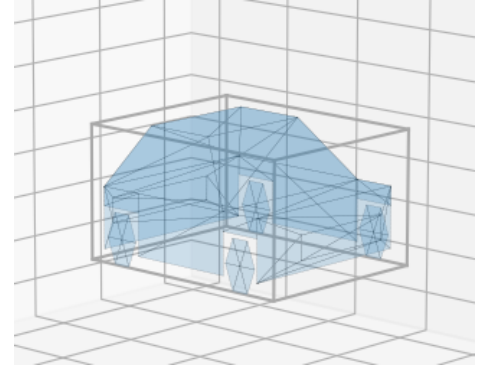


Rysunek 5.2. Powiązanie pomiędzy kategoriami pojazdów wspieranych przez model sensora a typami pojazdów zdefiniowanych w zbiorze *OSI::VehicleClassification*

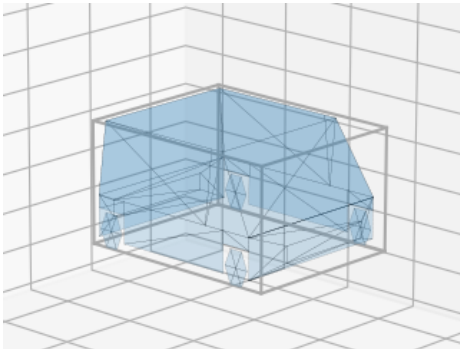
Modele geometryczne dla wymienionych wyżej kategorii pojazdów samochodowych są zaprezentowane na rysunku 5.3.



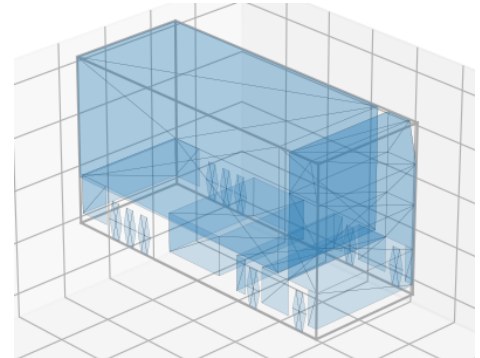
(a) Model geometryczny dla pojazdów dwukołowych właściwy dla typów *OSI::TYPE_MOTORBIKE* i *OSI::TYPE_BICYCLE* ze zbioru *OSI::VehicleClassification*



(b) Model geometryczny dla pojazdów klasy samochód osobowy właściwy dla typów *OSI::TYPE_SMALL_CAR*, *OSI::TYPE_COMPACT_CAR*, *OSI::TYPE_MEDIUM_CAR* i *OSI::TYPE_LUXURY_CAR* ze zbioru *OSI::VehicleClassification*



(c) Model geometryczny dla pojazdów klasy autobus właściwy dla typu *OSI::TYPE_BUS* ze zbioru *OSI::VehicleClassification*



(d) Model geometryczny dla pojazdów klasy samochód ciężarowy właściwy dla typu *OSI::TYPE_HEAVY_TRUCK* ze zbioru *OSI::VehicleClassification*

Rysunek 5.3. Przykłady struktur geometrycznych wygenerowanych dla kilku najważniejszych klas pojazdów samochodowych wspieranych przez model sensora

5.3 Generacja zbioru centrów rozpraszania

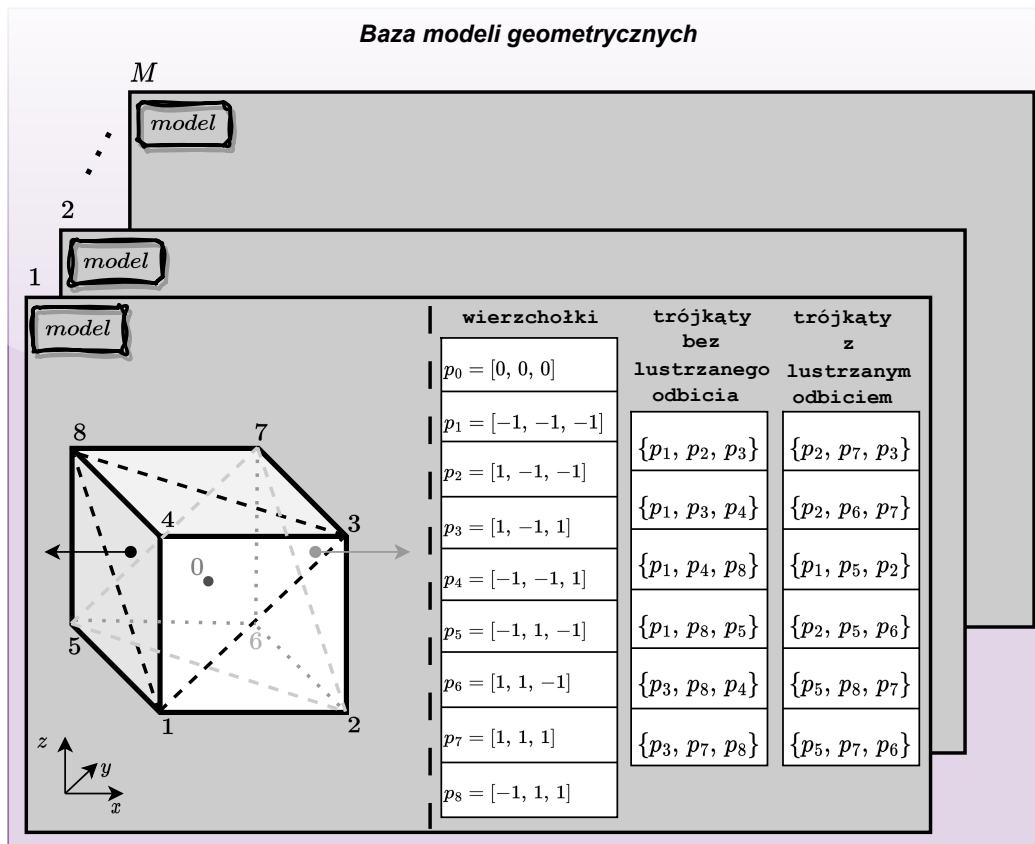
Przyjmijmy na tym etapie pewną notację wykorzystywaną w dalszych częściach pracy. Mianowicie, niech $[a, b]_{\mathbb{N}}$ będzie reprezentacją przedziału określonego na zbiorze liczb naturalnych, gdzie $a, b \in \mathbb{N} : a \leq b$. Wykorzystując tego typu oznaczenie można wprowadzić dwie równoważne definicje zbioru składającego się z wszystkich liczb naturalnych z przedziału $[a, b]_{\mathbb{N}}$:

$$\{x : x \in [a, b]_{\mathbb{N}}\} \iff \{x \in \mathbb{N} : a \leq x \leq b\}. \quad (5.3.1)$$

Jak napisano wyżej, podstawą generacji zbioru centrów rozpraszania jest baza modeli geometrycznych, zawierająca trójwymiarowe struktury zbudowane z połączonych ze sobą w odpowiedni

sposób trójkątów, wygenerowana dla czterech kategorii pojazdów samochodowych, takich jak: motocykl, samochód osobowy, autobus oraz samochód ciężarowy. Do celu wyjaśnienia procedury generacji centrów rozpraszania przyjmijmy jednak, że model sensora korzysta z pewnej bliżej nieokreślonej bazy składającej się z M ($M \in \mathbb{N}$) tego typu trójwymiarowych struktur; przy czym wiadomo, iż jednym z elementów tejże bazy jest model geometryczny pojazdu zbudowany na podstawie sześcianu o boku długości 2 m, zawiera on zatem 12 połączonych ze sobą trójkątów rozpiętych na 8 wierzchołkach tegoż sześcianu. Na rysunku 5.4 zaprezentowano wizualizację tej przykładowej bazy modeli geometrycznych wraz ze wspomnianą pokazową strukturą (opartą na sześcianie), gdzie:

- $p_i \in \mathbb{R}^3$ to współrzędne kartezjańskie wierzchołka o indeksie i ($i \in [1, 8]_{\mathbb{N}}$) sześcianu, na podstawie którego zbudowany jest przykładowy model geometryczny,
- $\{p_a, p_b, p_c\}$ ($a, b, c \in [1, 8]_{\mathbb{N}}$) to reprezentacja pojedynczego trójkąta rozpiętego na wierzchołkach a, b oraz c sześcianu; przy czym kolejność punktów definiujących dany trójkąt jest kluczowa, gdyż ma ona bezpośredni wpływ na kierunek wektora normalnego płaszczyzny rozpiętej na wierzchołkach tego wielokąta - definiuje się zatem dwa zbiory trójkątów (bez lustrzanego odbicia oraz z lustrzanym odbiciem) tak, aby wektor normalny był zawsze skierowany na zewnątrz bryły.



Rysunek 5.4. Przykładowa reprezentacja graficzna bazy modeli geometrycznych pojazdów samochodowych wraz z pokazową strukturą zbudowaną na bazie sześcianu

Proces generacji zbioru centrów rozpraszania zostanie wyjaśniony na podstawie przykładowej bazy modeli geometrycznych pokazanej na rysunku 5.4 - w przypadku struktur zaprezentowanych na rysunku 5.3, wykorzystywanych przez model sensora, metodologia jest analogiczna. Pojedyncze centrum rozpraszania określa się przy użyciu trzech następujących parametrów zdefiniowanych w równaniach (5.3.2a) - (5.3.2c). Są to: współrzędne kartezjańskie danego centrum rozpraszania ($p_{sc} \in \mathbb{R}^3$), wektor normalny płaszczyzny rozpiętej na wierzchołkach danego trójkąta ($v_{\perp} \in \mathbb{R}^3$) oraz pole powierzchni tego trójkąta ($A \in \mathbb{R}$):

$$p_{sc} = \frac{p_a + p_b + p_c}{3}, \quad v_{\perp} = (p_b - p_a) \times (p_c - p_a), \quad A = \frac{\|v_{\perp}\|}{2}, \quad (5.3.2a-c)$$

gdzie:

- $\{p_a, p_b, p_c\}$ to punkty reprezentujące trójkąt o wierzchołkach a , b i c ,
- $\times$ to operator iloczynu wektorowego,
- $\|\cdot\|$ oznacza normę euklidesową wektora.

Oznaczmy przez $K_m \in \mathbb{N}$ całkowitą liczbę trójkątów, z których składa się model geometryczny o indeksie m należący do bazy struktur geometrycznych pokazanej na rysunku 5.4, przy czym $m \in [1, M]_{\mathbb{N}}$. Procedura generacji zestawu centrów rozpraszania dla modelu o indeksie m , dla pewnego dyskretnego momentu czasu t , polega zatem na utworzeniu zbioru $P_{sc}^{t,m} \in \mathbb{R}^{K_m \times 7}$ zdefiniowanego według następującej formuły:

$$P_{sc}^{t,m} = \left\{ \{p_{sc}, v_{\perp}, A\}^{k_m} \in \mathbb{R}^7 : k_m \in [1, K_m]_{\mathbb{N}} \right\}. \quad (5.3.3)$$

Przypomnijmy, że wejście do modelu sensora to wiadomość $OSI::SensorView$ o unikalnym stemplu czasowym t , która zawiera informacje o N ($N \in \mathbb{N}$) obiektach klasy $OSI::MovingObject$ (patrz rysunek 5.1). Przyjmuje się, że pierwszy element typu $OSI::MovingObject$ (o identyfikatorze równym 1) reprezentuje pojazd wiodący, względem którego określa się pozycję i orientację wirtualnego czujnika [122]. Uzyskanie zbioru centrów rozpraszania dla danej struktury $OSI::SensorView$ jest więc jednoznaczne z określeniem wartości zbioru $P_{sc}^t \in \mathbb{R}^{(N-1) \times f(m_n) \times 7}$ zdefiniowanego na podstawie wzoru:

$$P_{sc}^t = \left\{ \{p_{sc}, v_{\perp}, A\}^{n,k} \in \mathbb{R}^7 : (n, k) \in [2, N]_{\mathbb{N}} \times [1, f(m_n)]_{\mathbb{N}} \right\}, \quad (5.3.4)$$

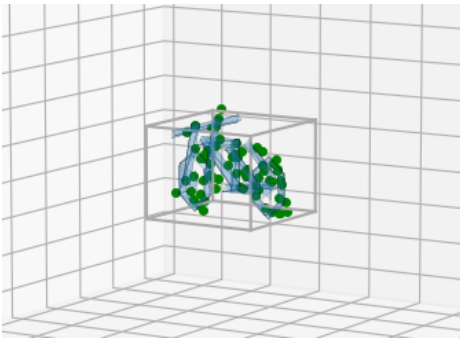
gdzie:

- $\times$ reprezentuje w tym przypadku iloczyn kartezjański zbiorów,
- $m_n \in [1, M]_{\mathbb{N}}$ jest indeksem modelu geometrycznego ze wspomnianej wyżej bazy powiązany jednoznacznie z typem obiektu klasy $OSI::MovingObject$ o indeksie $n \in [2, N]_{\mathbb{N}}$,
- $f : B \rightarrow K$ to funkcja przyporządkowująca każdemu elementowi zbioru B , zawierającego indeksy wszystkich modeli geometrycznych zdefiniowanych dla bazy zaprezentowanej na rysunku

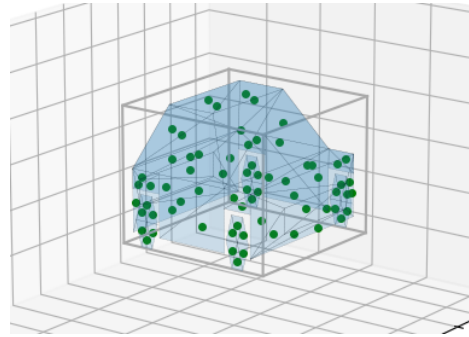
5.4, dokładnie jeden element zbioru K , uwzględniającego całkowite liczby trójkątów dla każdego z modeli geometrycznych; zestawy B , $K \in \mathbb{N}^M$ określono w równaniach:

$$B = \left\{ m : m \in [1, M]_{\mathbb{N}} \right\}, \quad K = \left\{ K_m : m \in [1, M]_{\mathbb{N}} \right\}. \quad (5.3.5a-b)$$

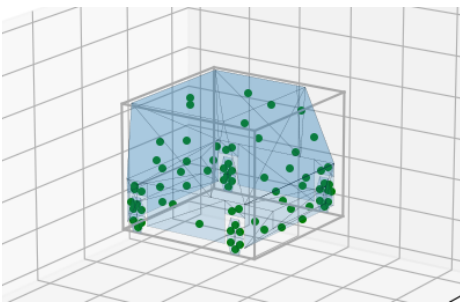
Jak wspomniano w rozdziale 3, obiekty klasy $OSI::MovingObject$ opisane są przy użyciu tak zwanych brył brzegowych o unikalnych wymiarach. Przed określeniem wartości zbioru centrów rozpraszania konieczne jest zatem przeskalowanie każdego z modeli geometrycznych spełniających warunki przynależności do zestawu P_{sc}^t , zdefiniowane w równaniu (5.3.4), do wymiarów danej bryły brzegowej. Zbiory centrów rozpraszania wygenerowane dla modeli geometrycznych wykorzystywanych przez omawiany model sensora zaprezentowano na rysunku 5.5.



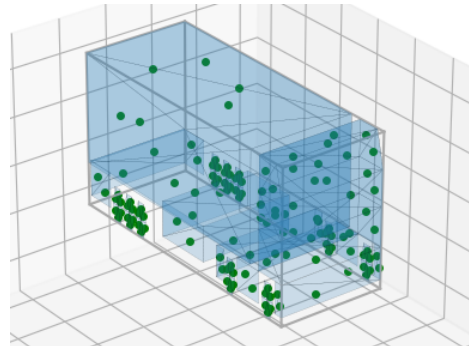
(a) Pozycje p_{sc} centrów rozpraszania uzyskane dla przykładowego pojazdu dwukołowego powiązanego z typami $OSI::TYPE_MOTORBIKE$ i $OSI::TYPE_BICYCLE$ ze zbioru $OSI::VehicleClassification$



(b) Pozycje p_{sc} centrów rozpraszania uzyskane dla przykładowego samochodu osobowego powiązanego z typami $OSI::TYPE_SMALL_CAR$, $OSI::TYPE_COMPACT_CAR$, $OSI::TYPE_MEDIUM_CAR$ i $OSI::TYPE_LUXURY_CAR$ ze zbioru $OSI::VehicleClassification$



(c) Pozycje p_{sc} centrów rozpraszania uzyskane dla przykładowego pojazdu klasy autobus powiązanego z typem $OSI::TYPE_BUS$ ze zbioru $OSI::VehicleClassification$



(d) Pozycje p_{sc} centrów rozpraszania uzyskane dla przykładowego samochodu ciężarowego powiązanego z typem $OSI::TYPE_HEAVY_TRUCK$ ze zbioru $OSI::VehicleClassification$

Rysunek 5.5. Wizualizacja zbiorów centrów rozpraszania wygenerowanych dla modeli geometrycznych pojazdów samochodowych wspieranych przez model sensora

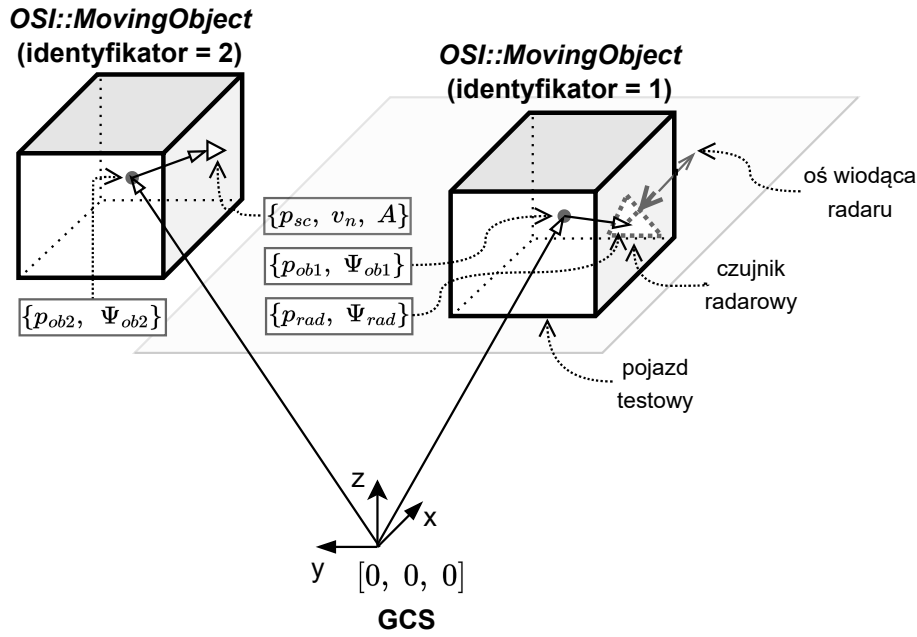
5.4 Przekształcenie centrów rozpraszania do układu współrzędnych sensora

Warto na tym etapie zdefiniować układy współrzędnych wykorzystywane w ramach standardu ASAM OSI. Standard ASAM OSI definiuje trzy następujące układy współrzędnych, wszystkie wyrażone przy użyciu konwencji prawoskrętnej [124]:

- układ współrzędnych obiektu (ang. local coordinate systems, LCS), względem którego generuje się elementy składowe zbioru P_{sc}^t określone w równaniach (5.3.2a) - (5.3.2c),
- globalny układ współrzędnych (ang. global coordinate systems, GCS), na podstawie którego określone są pozycje i orientacje obiektów klasy *OSI::MovingObject* zawarte w strukturze *OSI::SensorView* - jego początek to pewien z góry ustalony punkt w trójwymiarowej przestrzeni wirtualnego scenariusza,
- układ współrzędnych sensora (ang. sensor coordinate systems, SCS), zdefiniowany z wykorzystaniem pozycji i orientacji wirtualnego czujnika, względem którego wyznaczone są sztuczne pomiary sensoryczne zwracane na wyjściu modelu sensora w formie odpowiednio utworzonej wiadomości *OSI::SensorData*.

Przywołajmy raz jeszcze fakt, iż model sensora kompatybilny ze standardem ASAM OSI to procedura przyjmująca na wejściu strukturę *OSI::SensorView* i zwracająca strukturę *OSI::SensorData*. Przed wyliczeniem parametrów syntetycznego wykrycia radarowego należy zatem przekształcić zbiór P_{sc}^t z układów LCS do układu SCS. Reprezentację graficzną wspomnianych wyżej układów współrzędnych przedstawiono na rysunku 5.6, gdzie:

- $p_{ob1}, p_{ob2} \in \mathbb{R}^3$ to współrzędne kartezjańskie pozycji obiektów klasy *OSI::MovingObject* o identyfikatorach wynoszących kolejno 1 i 2 oraz o unikalnych wymiarach, określone względem układu GCS,
- $\Psi_{ob1}, \Psi_{ob2} \in \mathbb{R}^3$ to orientacje obiektów o identyfikatorach 1 i 2 wyrażone względem układu GCS przy użyciu tak zwanych kątów Tait-Bryan [125] - kąt α (odchylenie, ang. yaw), kąt β (pochylenie, ang. pitch), kąt γ (przechylenie, ang. roll) - reprezentujących obroty wokół trzech głównych osi, kolejno: wokół osi *OZ*, wokół osi *OY* i wokół osi *OX* [126],
- $\{p_{sc}, v_{\perp}, A\} \in \mathbb{R}^7$ to zbiór reprezentujący pojedyncze centrum rozpraszania określone względem układu LCS obiektu klasy *OSI::MovingObject* o identyfikatorze równym 2,
- $p_{rad}, \Psi_{rad} \in \mathbb{R}^3$ to kolejno współrzędne kartezjańskie pozycji radaru i orientacja radaru (zdefiniowana z wykorzystaniem kątów Tait-Bryan) wyrażone względem układu LCS obiektu klasy *OSI::MovingObject* o identyfikatorze równym 1.



Rysunek 5.6. Reprezentacja graficzna zależności pomiędzy układami współrzędnych zdefiniowanych w ramach standardu ASAM OSI

Standard ASAM OSI do przekształcenia danego wektora z pewnego układu współrzędnych A do innego układu współrzędnych B wykorzystuje metodę bazującą na macierzy rotacji oraz na wektorze translacji [124]. Przyjmijmy, że układ B wyznaczono względem układu A . Co za tym idzie, znane są składowe wektorów $p^A, \Psi^A \in \mathbb{R}^3$, zdefiniowanych w równaniach (5.4.1a) - (5.4.1b), reprezentujących pozycję i orientację początku układu współrzędnych B wyznaczonych relatywnie do układu A :

$$p^A = [x, y, z]^T, \quad \Psi^A = [\alpha, \beta, \gamma]^T, \quad (5.4.1a-b)$$

gdzie x, y i z to współrzędne kartezjańskie wektora p^A , natomiast α, β i γ to kąty yaw, pitch oraz roll. Niech $\Theta \in \mathbb{R}^{3 \times 3}$ będzie oznaczeniem macierzy rotacji spełniającej warunki ortogonalności zdefiniowane w równaniach [127]:

$$\Theta^{-1} = \Theta^T, \quad \Theta^T \Theta = I, \quad (5.4.2a-b)$$

gdzie $I \in \mathbb{R}^{3 \times 3}$ to macierz jednostkowa.

Każdy wektor określony w układzie B może być w jednoznaczny sposób przekształcony do układu A na podstawie macierzy rotacji $\Theta_B^A \in \mathbb{R}^{3 \times 3}$ i wektora translacji $\Gamma_B^A \in \mathbb{R}^3$ zdefiniowanych w równaniach (5.4.3a) - (5.4.3b); przy czym warto tutaj zaznaczyć, iż w przypadku operatorów rotacji i translacji dolny i górny indeks to oznaczenia odpowiednio referencyjnego oraz docelowego układu współrzędnych:

$$\Theta_B^A = \Theta(\alpha)\Theta(\beta)\Theta(\gamma), \quad \Gamma_B^A = p^A, \quad (5.4.3a-b)$$

gdzie macierze $\Theta(\alpha)$, $\Theta(\beta)$, $\Theta(\gamma) \in \mathbb{R}^{3 \times 3}$ - reprezentujące obroty odpowiednio wokół osi OZ , osi OY oraz osi OX - zdefiniowano w następujący sposób [128]:

$$\Theta(\alpha) = \begin{pmatrix} \cos \alpha & -\sin \alpha & 0 \\ \sin \alpha & \cos \alpha & 0 \\ 0 & 0 & 1 \end{pmatrix}, \quad \Theta(\beta) = \begin{pmatrix} \cos \beta & 0 & \sin \beta \\ 0 & 1 & 0 \\ -\sin \beta & 0 & \cos \beta \end{pmatrix}, \quad \Theta(\gamma) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos \gamma & -\sin \gamma \\ 0 & \sin \gamma & \cos \gamma \end{pmatrix}. \quad (5.4.4a-c)$$

Przeprowadzając mnożenie macierzy zdefiniowane w równaniu (5.4.3a), macierz Θ_B^A można zapisać według takiej oto formuły:

$$\Theta_B^A = \begin{pmatrix} \cos \alpha \cos \beta & \cos \alpha \sin \beta \sin \gamma - \sin \alpha \cos \gamma & \cos \alpha \sin \beta \cos \gamma + \sin \alpha \sin \gamma \\ \sin \alpha \cos \beta & \sin \alpha \sin \beta \sin \gamma + \cos \alpha \cos \gamma & \sin \alpha \sin \beta \cos \gamma - \cos \alpha \sin \gamma \\ -\sin \beta & \cos \beta \sin \gamma & \cos \beta \cos \gamma \end{pmatrix}. \quad (5.4.5)$$

Założmy, że $\nu^B \in \mathbb{R}^3$ jest wektorem określonym w układzie B . Wektor ν^A będący wynikiem przekształcenia wektora ν^B z układu B do układu A uzyskujemy na podstawie równania:

$$\nu^A = \Theta_B^A \nu^B + \Gamma_B^A. \quad (5.4.6)$$

Natomiast przekształcenie odwrotne (z układu A do układu B) realizuje się przy użyciu wzoru:

$$\nu^B = \Theta_B^{A^{-1}} (\nu^A - \Gamma_B^A). \quad (5.4.7)$$

Wykorzystując wprowadzone wyżej definicje, zwłaszcza formułę na transformację danego wektora z referencyjnego układu współrzędnych do docelowego układu współrzędnych oraz formułę na transformację odwrotną - kolejno równania (5.4.6) i (5.4.7) - można zdefiniować procedurę przekształcenia zbioru centrów rozpraszania z układów LCS do układu SCS. Przypomnijmy, że zbiór P_{sc}^t określony w równaniu (5.3.4) zawiera zestawy wartości $\{p_{sc}, v_{\perp}, A\}^{n,k} \in \mathbb{R}^7$: $(n, k) \in [2, N]_{\mathbb{N}} \times [1, f(m_n)]_{\mathbb{N}}$; gdzie n to identyfikator obiektu klasy *OSI::MovingObject* dostępnego w danej strukturze *OSI::SensorView*, natomiast k jest indeksem trójkąta wchodzącego w skład modelu geometrycznego wygenerowanego dla obiektu o indeksie n . Warto zaznaczyć, iż wszystkie elementy typu *OSI::MovingObject* będące częścią wiadomości *OSI::SensorView* są jednoznacznie określone względem układu GCS poprzez swoje wektory pozycji i orientacji - także pojazd wiodący o identyfikatorze równym 1, względem którego określa się pozycję i orientację wirtualnego czujnika. Co za tym idzie, dla każdego obiektu klasy *OSI::MovingObject* zawartego w

strukturze *OSI::SensorView* można zdefiniować unikalny układ współrzędnych LCS_n wyznaczony na podstawie wektorów p_{ob_n} , $\Psi_{ob_n} \in \mathbb{R}^3$ - gdzie n równe 1 odnosi się do pojazdu wiodącego, natomiast $n \in [2, N]_{\mathbb{N}}$ wskazuje na obiekty, dla których oblicza się wartości zbioru P_{sc}^t . Wynika z tego, że zbiór wartości $\{p_{sc}, v_{\perp}, A\}^{n,k}$ uzyskany dla ustalonego identyfikatora $n \in [2, N]_{\mathbb{N}}$ oraz dla $k \in [1, f(m_n)]_{\mathbb{N}}$ określa się względem układu LCS_n , natomiast pozycja i orientacja sensora (kolejno p_{rad} oraz Ψ_{rad}) wyznaczone są względem układu LCS_1 .

Proces przekształcenia zbioru P_{sc}^t do układu SCS wyjaśniony będzie na przykładzie sytuacji zobrazowanej na rysunku 5.6. Struktura *OSI::SensorView* dla scenariusza pokazanego na wspomnianym rysunku zawiera informacje na temat dwóch obiektów klasy *OSI::MovingObject* o identyfikatorach wynoszących 1 i 2, są to odpowiednio: obiekt będący reprezentacją pojazdu wiodącego oraz obiekt, dla którego wyliczone zostaną parametry zbioru P_{sc}^t . Warto zaznaczyć, iż choć w tym przypadku N wynosi 2, to dla $N > 2$ procedura transformacji zestawu P_{sc}^t do układu SCS jest analogiczna.

Definiuje się zatem dwa lokalne układy współrzędnych (powiązane jednoznacznie z elementami wiadomości *OSI::SensorView* o indeksach 1 i 2) określone względem układu GCS - LCS_1 i LCS_2 . Wykorzystując więc wzór (5.4.3a), należy wyliczyć wartości dla takich oto macierzy rotacji: $\Theta_{SCS}^{LCS_1}$, $\Theta_{LCS_1}^{GCS}$, $\Theta_{LCS_2}^{GCS} \in \mathbb{R}^{3 \times 3}$. Dodatkowo, przy użyciu wzoru (5.4.3b), trzeba obliczyć składowe trzech wektorów translacji: $\Gamma_{SCS}^{LCS_1}$, $\Gamma_{LCS_1}^{GCS}$, $\Gamma_{LCS_2}^{GCS} \in \mathbb{R}^3$.

Przypomnijmy, że celem jest tutaj przekształcenie zbioru P_{sc}^t z układów LCS do układu SCS. Przyjmijmy zatem dwa następujące oznaczenia - $P_{sc}^{LCS,t}$, $P_{sc}^{SCS,t} \in \mathbb{R}^{(N-1) \times f(m_n) \times 7}$ - reprezentujące zbiory centrów rozpraszania wyrażone odpowiednio w układach LCS i w układzie SCS, gdzie $P_{sc}^{LCS,t}$ określa się według formuły analogicznej do wzoru (5.3.4):

$$P_{sc}^{LCS,t} = \left\{ \{p_{sc}^{LCS_n}, v_{\perp}^{LCS_n}, A\}^{n,k} \in \mathbb{R}^7 : (n, k) \in [2, N]_{\mathbb{N}} \times [1, f(m_n)]_{\mathbb{N}} \right\}. \quad (5.4.8)$$

Z równania (5.4.8) wynika, iż wyrażenie zbioru $P_{sc}^{LCS,t}$ względem układu SCS polega na odpowiedniemu przekształceniu par wektorów $\{p_{sc}^{LCS_n}, v_{\perp}^{LCS_n} \in \mathbb{R}^3\}^{n,k} \forall n \in [2, N]_{\mathbb{N}} \forall k \in [1, f(m_n)]_{\mathbb{N}}$.

Pojedynczy wektor $p_{sc}^{LCS_n|n,k}$ (określony dla ustalonego n i k) transformuje się według następującego wzoru:

$$p_{sc}^{SCS|n,k} = \Theta_{SCS}^{LCS_1-1} \left(\Theta_{LCS_1}^{GCS-1} \left((\Theta_{LCS_n}^{GCS} p_{sc}^{LCS_n|n,k} + \Gamma_{LCS_n}^{GCS}) - \Gamma_{LCS_1}^{GCS} \right) - \Gamma_{SCS}^{LCS_1} \right). \quad (5.4.9)$$

Z kolei w przypadku wektora $v_{\perp}^{LCS_n|n,k}$, reprezentującego jednoznacznie orientację trójkąta o indeksie k , wchodzącego w skład modelu geometrycznego wygenerowanego dla obiektu o indeksie n , pomija się operację translacji, a zatem procedura przekształcenia wyraża się stosownie do formuły:

$$v_{\perp}^{SCS|n,k} = \Theta_{SCS}^{LCS_1-1} \Theta_{LCS_1}^{GCS-1} \Theta_{LCS_n}^{GCS} v_{\perp}^{LCS_n|n,k}. \quad (5.4.10)$$

Podsumowując, zbiór $P_{sc}^{SCS,t}$ mieszczący centra rozpraszania wygenerowane dla wszystkich obiektów klasy $OSI::MovingObject$, zawartych w wiadomości $OSI::SensorView$ o unikalnym stemplu czasowym t , przekształcone do układu SCS, można zapisać w następujący sposób:

$$P_{sc}^{SCS,t} = \left\{ \{p_{sc}^{SCS}, v_{\perp}^{SCS}, A\}^{n,k} \in \mathbb{R}^7 : (n, k) \in [2, N]_{\mathbb{N}} \times [1, f(m_n)]_{\mathbb{N}} \right\}. \quad (5.4.11)$$

5.5 Definicja zbioru sztucznych wykryć radarowych

Jak wspomniano w rozdziale 2, pojedyncze wykrycie radarowe określa się jednoznacznie przez następującą czwórkę wartości, pokazaną w formie graficznej na rysunku 2.1: $R, v_r, \theta, \phi \in \mathbb{R}$. Oznaczmy jako P_d^t zbiór syntetycznych wykryć radarowych utworzony na podstawie zestawu $P_{sc}^{SCS,t}$, który z kolei wygenerowano bazując na wiadomości $OSI::SensorView$ o unikalnym stemplu czasowym t . Każdy sztuczny pomiar jest powiązany z dokładnie jednym centrum rozpraszania. Zatem zbiór $P_d^t \in \mathbb{R}^{(N-1) \times f(m_n) \times 4}$ definiuje się analogicznie jak zestaw $P_{sc}^{SCS,t}$, a więc na podstawie wzoru (5.5.1); w przypadku P_d^t pomija się górny indeks SCS, gdyż zbiór ten jest z definicji określony względem układu SCS:

$$P_d^t = \left\{ \{R, v_r, \theta, \phi\}^{n,k} \in \mathbb{R}^4 : (n, k) \in [2, N]_{\mathbb{N}} \times [1, f(m_n)]_{\mathbb{N}} \right\}. \quad (5.5.1)$$

Definicja zbioru P_d^t pokazana we wzorze (5.5.1) nie uwzględnia oczywiście procedur wykonywanych w referencyjnym modelu sensora [21]. Dlatego też zaproponowana definicja zostanie rozszerzona o odpowiednie warunki, których zastosowanie jest jednoznaczne z wykonaniem wszystkich operacji właściwych dla omawianego modelu sensora.

5.5.1 Wyliczenie wartości dla parametrów pojedynczego wykrycia radarowego

Zapiszmy wpierw, w jaki sposób obliczane są podstawowe parametry wykrycia. Przyjmijmy, że współrzędne kartezjańskie wektora $p_{sc}^{SCS|n,k}$ reprezentowane są przez trójkę wartości: $x, y, z \in \mathbb{R}$. Procedura wyliczenia wartości dla parametrów $R^{n,k}, \theta^{n,k}$ oraz $\phi^{n,k}$ syntetycznego wykrycia radarowego, zdefiniowana w równaniach (5.5.2a) - (5.5.2c) dla ustalonego n i k , wynika jednoznacznie z przekształcenia pomiędzy układem kartezjańskim a sferycznym:

$$R^{n,k} = \sqrt{x^2 + y^2 + z^2}, \quad \theta^{n,k} = \arccos \frac{z}{r}, \quad \phi^{n,k} = \arctg \frac{y}{x}. \quad (5.5.2a-c)$$

Z kolei obliczenie wartości dla parametru $v_r^{n,k}$, czyli prędkości względnej (prędkości radialnej) danego celu, rozumianej jako szybkość zmiany odległości $R^{n,k}$ (patrz rysunek 2.1), jest nieco bardziej skomplikowane. W rzeczywistym systemie radarowym typu FMCW wartość prędkości radialnej v_r uzyskuje się poprzez analizę przesunięć w fazie sygnału wywołanych przez efekt Dopplera; wielkość ta powinna być ujemna dla obiektu zbliżającego się do radaru i dodatnia w przeciwnym wypadku. Zauważmy, że dla każdego obiektu klasy *OSI::MovingObject* o indeksie $n \in [2, N]_{\mathbb{N}}$ zawartego w strukturze *OSI::SensorView* definiuje się jego prędkość $v^{GCS|n} \in \mathbb{R}^3$ wyrażoną względem układu GCS (patrz rysunek 2.1). Wektor $v^{GCS|n}$ należy oczywiście przekształcić do układu SCS. Procedurę transformacji określa się przy użyciu formuły (5.5.3); wzór ten jest analogiczny do wzoru (5.4.10), jednak w tym przypadku pomija się macierz rotacji $\Theta_{LCS_n}^{GCS}$:

$$v^{SCS|n} = \Theta_{SCS}^{LCS_1-1} \Theta_{LCS_1}^{GCS-1} v^{GCS|n}. \quad (5.5.3)$$

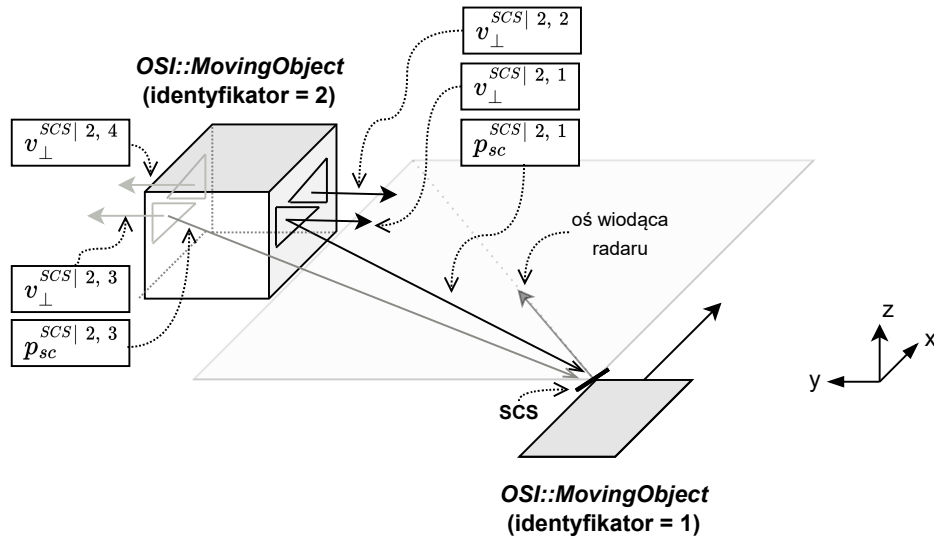
Niech $v_x^{SCS|n}, v_y^{SCS|n} \in \mathbb{R}$ będą rzutami wektora $v^{SCS|n}$ na osie OX i OY układu SCS. Prędkość względną $v_r^{n,k}$ właściwą dla sztucznego wykrycia o ustalonych indeksach n i k wylicza się na podstawie równania (5.5.4), jako kombinację składowych $v_x^{SCS|n}$ i $v_y^{SCS|n}$ [21]:

$$v_r^{n,k} = \cos \theta^{n,k} v_x^{SCS|n} + \sin \theta^{n,k} v_y^{SCS|n}. \quad (5.5.4)$$

5.5.2 Likwidacja pomiarów niezwróconych w kierunku wirtualnego czujnika

Na tym etapie, kiedy zdefiniowane są już wzory na obliczenie elementów składowych zbioru sztucznych wykryć radarowych, można dodać do definicji zbioru P_d^t określonej w równaniu (5.5.1) warunki reprezentujące operacje wykonywane w modelu sensora.

Pierwszą procedurą, jaką przeprowadza się w modelu sensora, jest zweryfikowanie, czy trójkąt o indeksie k wchodzący w skład modelu geometrycznego o indeksie n jest skierowany w kierunku wirtualnego czujnika radarowego [21]. Przyjmijmy, że w danej wiadomości *OSI::SensorView* dostępne są dwa obiekty klasy *OSI::MovingObject*: pojazd wiodący o identyfikatorze równym 1 z przypisanym do niego wirtualnym czujnikiem radarowym oraz obiekt o identyfikatorze równym 2 z modelem geometrycznym składającym się z czterech parami symetrycznych trójkątów. Sytuację tę zobrazowano na rysunku 5.7; gdzie indeksy górne wektorów reprezentujących centra rozpraszania są zgodne z definicją zbioru $P_{sc}^{SCS,t}$ określoną w równaniu (5.4.11), przy czym n przyjmuje wartość równą 2, natomiast $k \in [1, 4]_{\mathbb{N}}$.



Rysunek 5.7. Wizualizacja zbioru centrów rozpraszania wyznaczonych dla obiektu klasy *OSI::MovingObject* o identyfikatorze równym 2 reprezentowanych przez współrzędne kartezjańskie pozycji środka danego trójkąta oraz wektor normalny płaszczyzny rozpiętej na wierzchołkach tego trójkąta. Model geometryczny właściwy dla wspomnianego obiektu składa się z czterech parami symetrycznych wielokątów

Analizując kierunki oraz zwroty wektorów normalnych pokazanych na rysunku 5.7, określonych dla modelu geometrycznego obiektu typu *OSI::MovingObject* o identyfikatorze równym 2, można zauważyć, iż tylko dwa z czterech centrów rozpraszania są widoczne z perspektywy wirtualnego czujnika. W celu stwierdzenia, czy dane centrum rozpraszania może być wykryte przez sensor, oblicza się kąt $\alpha^{n,k} \in \mathbb{R}$ pomiędzy wektorem $-p_{sc}^{SCS|n,k}$ (wektor o przeciwnym kierunku w stosunku do wektora $p_{sc}^{SCS|n,k}$) a wektorem $v_{\perp}^{SCS|n,k}$ na podstawie formuły [21]:

$$\alpha^{n,k} = \arccos \frac{-p_{sc}^{SCS|n,k} \cdot v_{\perp}^{SCS|n,k}}{\| -p_{sc}^{SCS|n,k} \| \| v_{\perp}^{SCS|n,k} \|}, \quad (5.5.5)$$

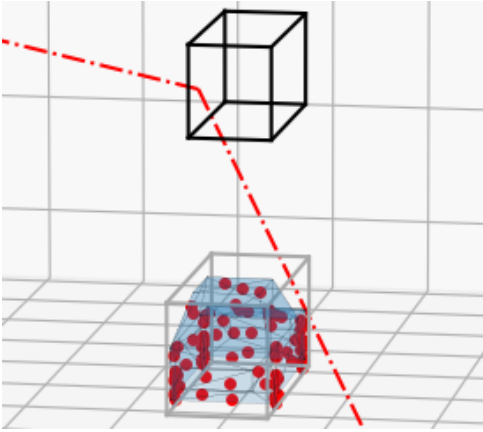
gdzie:

- $\cdot$ to operator iloczynu skalarnego,
- $\| \cdot \|$ oznacza normę euklidesową wektora.

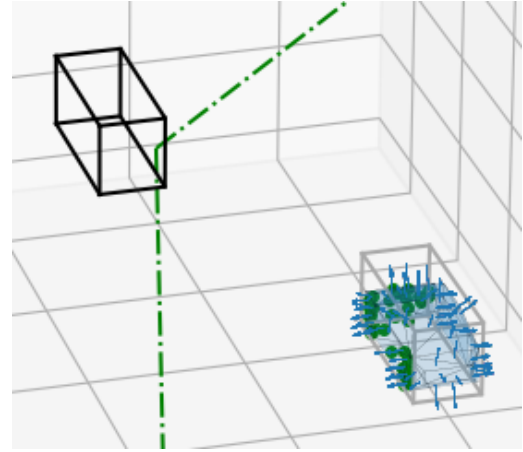
Wykorzystując kąt $\alpha^{n,k}$ można zmodyfikować definicję zbioru P_d^t tak, by sprawdzany był także warunek na orientację trójkąta o ustalonych indeksach n i k względem układu SCS. Dodając zatem do równania (5.5.1) dodatkowe kryterium na kąt $\alpha^{n,k}$ uzyskujemy nową wersję formuły na elementy zbioru P_d^t :

$$P_d^t = \left\{ \{R, v_r, \theta, \phi\}^{n,k} \in \mathbb{R}^4 : (n, k) \in [2, N]_{\mathbb{N}} \times [1, f(m_n)]_{\mathbb{N}} \wedge \alpha^{n,k} \leq \frac{\pi}{2} \right\}. \quad (5.5.6)$$

Na rysunku 5.8 przedstawiono przykładową wizualizację sztucznych zbiorów wykryć zwróconych przez model sensora dla dwóch przypadków. Są to: zestaw P_d^t uzyskany na podstawie równania (5.5.1) (brak warunku na kąt $\alpha^{n,k}$ - rysunek 5.8a) oraz zestaw P_d^t uzyskany na podstawie równania (5.5.6) (uwzględnienie warunku na kąt $\alpha^{n,k}$ - rysunek 5.8b). Jak widać na rysunku 5.8b, sprawdzenie orientacji trójkątów wykorzystywanych w procesie wyliczania wartości zbioru P_d^t względem układu SCS znacznie zwiększa wiarygodność modelu sensora, gdyż syntetyczny zestaw wykryć zawiera jedynie pomiary widoczne z perspektywy wirtualnego czujnika.



(a) Pozycje sztucznych wykryć radarowych uzyskane dla obiektu klasy *OSI::MovingObject* na podstawie wzoru (5.5.1)



(b) Pozycje sztucznych wykryć radarowych uzyskane dla obiektu klasy *OSI::MovingObject* na podstawie wzoru (5.5.6)

Rysunek 5.8. Porównanie sztucznych zbiorów wykryć wyznaczonych kolejno przy użyciu wzorów (5.5.1) oraz (5.5.6). W drugim przypadku sprawdzany jest dodatkowo warunek na kąt $\alpha^{n,k}$ zdefiniowany we wzorze (5.5.5), dzięki czemu zestaw P_d^t zawiera tylko i wyłącznie sztuczne pomiary widoczne z perspektywy wirtualnego czujnika

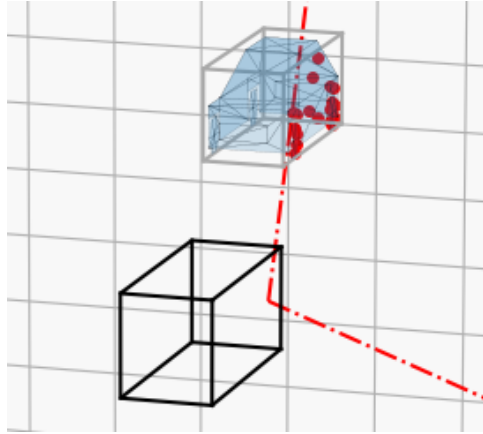
5.5.3 Odrzucenie pomiarów będących poza polem widzenia czujnika

Przypomnijmy, iż w radarach FMCW wykorzystuje się anteny kierunkowe skupiające energię w stosunkowo wąskim przedziale. Można więc zdefiniować pewien przedział kątowy, dla którego moc nadawanego sygnału jest wystarczająco silna, by system mógł odróżnić echo odbitej fali od zakłóceń. Zdefiniujmy zatem tak zwane pole widzenia wirtualnego czujnika, czyli obszar, wewnątrz którego dany obiekt będzie wykrywalny przez sensor. Załóżmy, że region ten reprezentuje się jednoznacznie przy użyciu pewnego wycinka kuli o promieniu $R_{\max} \in \mathbb{R}$. Wycinek ten w płaszczyźnie horyzontalnej ograniczają kąty $\theta_{\min}, \theta_{\max} \in \mathbb{R}$. Z kolei w płaszczyźnie wertykalnej granice fragmentu kuli wyznaczają kąty $\phi_{\min}, \phi_{\max} \in \mathbb{R}$. Przykładowe pole widzenia sensora wyrażone w formie wycinka

kuli zaprezentowano w rozdziale 2 na rysunku 2.7. Drugą procedurą wykonywaną w modelu sensora będzie zatem sprawdzenie, czy parametry sztucznego wykrycia mieszczą się w zadanym przedziale [21]. Zmodyfikowana formuła na elementy zbioru P_d^t uwzględniająca warunki na przynależność do obszaru widoczności wirtualnego czujnika zdefiniowana jest w równaniu:

$$P_d^t = \left\{ \{R, v_r, \theta, \phi\}^{n,k} \in \mathbb{R}^4 : (n, k) \in [2, N]_{\mathbb{N}} \times [1, f(m_n)]_{\mathbb{N}} \wedge \alpha^{n,k} \leq \frac{\pi}{2} \right. \\ \left. \wedge R^{n,k} \in [0, R_{\max}] \wedge \theta^{n,k} \in [\theta_{\min}, \theta_{\max}] \wedge \phi^{n,k} \in [\phi_{\min}, \phi_{\max}] \right\}. \quad (5.5.7)$$

Na rysunku 5.9 zaprezentowano zbiór P_d^t wygenerowany na podstawie wzoru (5.5.7), gdzie uwzględnione są wszystkie omówione do tej pory warunki. Na wspomnianym rysunku można zauważyć, że wykrycia znajdujące się poza polem widzenia sensora nie są uwzględnione (zgodnie z oczekiwaniem) w sztucznym zestawie pomiarów.



Rysunek 5.9. Sztuczny zbiór wykryć wyznaczony przy użyciu wzoru (5.5.7). Jak widać, zestaw P_d^t zawiera tylko i wyłącznie pomiary znajdujące się w polu widzenia wirtualnego czujnika

5.5.4 Eliminacja pomiarów przysłoniętych przez pozostałe obiekty obecne w scenie

W wiadomości *OSI::SensorView* zapisane są wysokopoziomowe informacje dotyczące scenariusza wirtualnej jazdy właściwe dla dyskretnego momentu czasu t . Przyjęto, że w strukturze *OSI::SensorView* znajduje się N obiektów klasy *OSI::MovingObject*, gdzie każdy z elementów typu *OSI::MovingObject* reprezentowany jest przez identyfikator $n \in [1, N]_{\mathbb{N}}$; przy czym identyfikator n równy 1 wskazuje na pojazd wiodący z przypisanym do niego wirtualnym czujnikiem, natomiast dla obiektów o indeksach $n \in [2, N]_{\mathbb{N}}$ wyliczana się wartości zbioru P_d^t . Warto na tym etapie wspomnieć o pewnym istotnym zjawisku, to jest dla $N > 2$ istnieje możliwość wystąpienia efektu przysłonięć. To znaczy, obiekt obserwowany z perspektywy wirtualnego czujnika może być częściowo lub w całości zakryty przez inny element scenariusza. Kolejną procedurą uwzględnioną w modelu sensora jest więc zweryfikowanie, czy wykrycie

o ustalonych indeksach n i k nie jest zasłonięte przez żaden z obiektów klasy $OSI::MovingObject$ o identyfikatorze $n' \in [2, N]_{\mathbb{N}} : n' \neq n$. Autorzy modelu bazują w tej operacji na obliczonej odpowiednio średniej odległości radialnej oraz minimalnym i maksymalnym kącie w płaszczyźnie horyzontalnej pomiędzy wirtualnym czujnikiem a danym obiektem klasy $OSI::MovingObject$ [21]. Rozbudujmy zatem definicję zbioru P_d^t zdefiniowaną w równaniu (5.5.7) o warunki na wystąpienie efektu przysłonięć. Utwórzmy wpierw jednak zbiory pomocnicze $\Lambda, \tilde{\Lambda} \in \mathbb{R}^{(N-1) \times 3}$. Zbiór Λ określa się na podstawie równania:

$$\Lambda = \left\{ \left\{ \bar{R}, \min(\theta), \max(\theta) \right\}^n \in \mathbb{R}^3 : n \in [2, N]_{\mathbb{N}} \right\}, \quad (5.5.8)$$

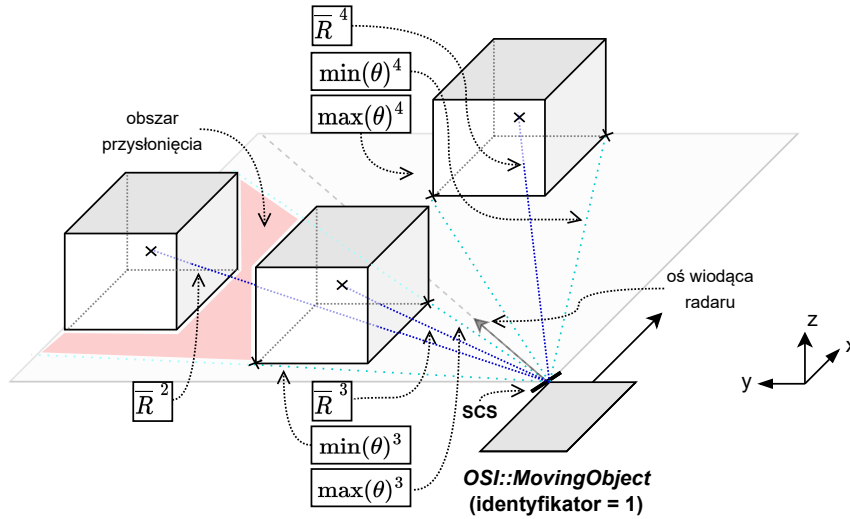
gdzie $\bar{R}^n, \min(\theta)^n, \max(\theta)^n \in \mathbb{R}$ - zdefiniowane kolejno w równaniach (5.5.9a), (5.5.9b) i (5.5.9c) - to odpowiednio średnia odległość radialna oraz minimalny i maksymalny kąt w płaszczyźnie horyzontalnej pomiędzy sensorem a obiektem typu $OSI::MovingObject$ o ustalonym identyfikatorze n ; wartości te wyliczono na podstawie elementów $R^{n,k}$ oraz $\theta^{n,k}$ ($k \in [1, f(m_n)]_{\mathbb{N}}$) zawartych w zbiorze P_d^t wyznaczonym na bazie formuły (5.5.7):

$$\begin{aligned} \bar{R}^n &= \frac{1}{f(m_n)} \sum_{k=1}^{f(m_n)} R^{n,k}, \\ \min(\theta)^n &= \min \left\{ \theta^{n,k} \in \mathbb{R} : k \in [1, f(m_n)]_{\mathbb{N}} \right\}, \\ \max(\theta)^n &= \max \left\{ \theta^{n,k} \in \mathbb{R} : k \in [1, f(m_n)]_{\mathbb{N}} \right\}. \end{aligned} \quad (5.5.9a-c)$$

Z kolei zbiór $\tilde{\Lambda}$ jest równy zbiorowi Λ ($\tilde{\Lambda} \subseteq \Lambda$ oraz $\Lambda \subseteq \tilde{\Lambda}$), przy czym elementy zestawu $\tilde{\Lambda}$ reprezentowane przez trójkę wartości $\left\{ \bar{R}, \min(\theta), \max(\theta) \right\}^n \in \mathbb{R}^3$ ułożone są zgodnie z rosnącą wartością średniej odległości.

Założmy teraz, że istnieje zbiór $\tilde{\Lambda}$ określony dla wszystkich obiektów typu $OSI::MovingObject$ o identyfikatorach od 2 do N zawartych w danej strukturze $OSI::SensorView$, gdzie $N > 2$. Oznaczmy przez n_1 i n_2 identyfikatory dwóch obiektów klasy $OSI::MovingObject$, dla których wyznaczono wartości zestawu $\tilde{\Lambda}$, a więc zgodnie z równaniem (5.5.8). Wykorzystując własność zbioru $\tilde{\Lambda}$, jaką jest uporządkowanie elementów na podstawie średniej odległości radialnej, można sformułować warunek konieczny (choć nie wystarczający) na wystąpienie efektu przysłonięć. Mianowicie, obiekt o numerze n_2 może być zasłonięty przez obiekt o numerze n_1 wtedy i tylko wtedy, gdy $\bar{R}^{n_1} \leq \bar{R}^{n_2}$; kryterium to sprawdza się $\forall n_1, n_2 \in [2, N]_{\mathbb{N}} : n_1 \neq n_2$ [21].

Przykładową tego typu sytuację pokazano na rysunku 5.10. Struktura $OSI::SensorView$ widoczna na wspomnianej grafice składa się z czterech obiektów klasy $OSI::MovingObject$ o identyfikatorach od 1 do 4, przy czym indeks 1 odnosi się do pojazdu wiodącego. Dodatkowo dla parametrów $\bar{R}^2, \bar{R}^3$ oraz $\bar{R}^4$ powiązanych jednoznacznie z elementami wiadomości $OSI::SensorView$ o identyfikatorach 2, 3 i 4 zachodzi następujący warunek: $\bar{R}^3 < \bar{R}^4 < \bar{R}^2$.



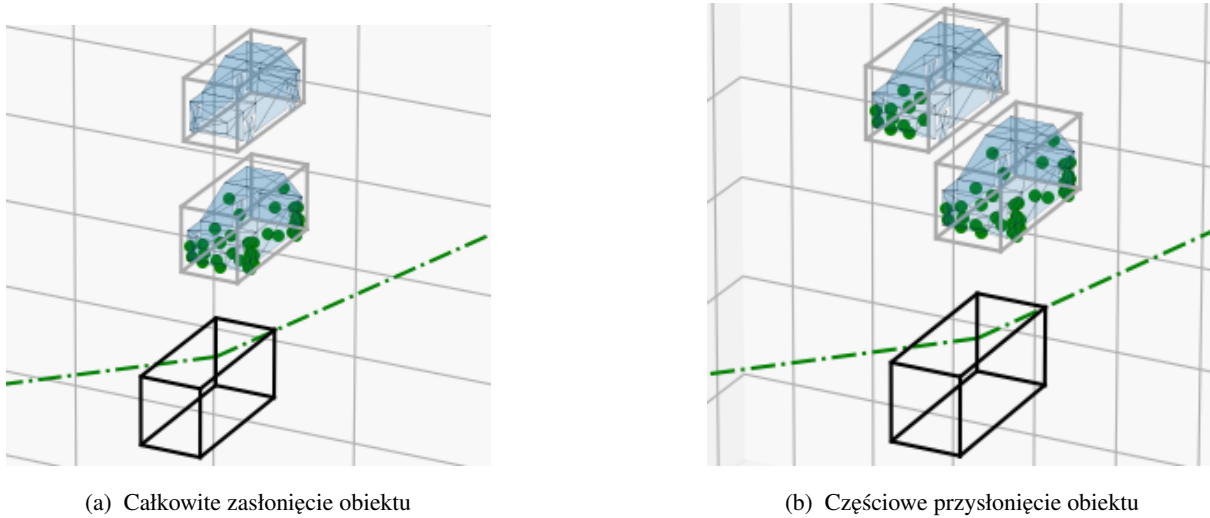
Rysunek 5.10. Przykładowa wizualizacja efektu przysłonięć

Jak widać na rysunku 5.10, obiekt o identyfikatorze 2 jest zakryty przez obiekt o identyfikatorze 3, co zgadza się z przyjętym warunkiem na średnią odległość radialną: $\bar{R}^3 \leq \bar{R}^2$. Natomiast brak efektu przesłonięcia występuje dla obiektu numer 4, pomimo spełnienia wspomnianego kryterium odległości: $\bar{R}^3 \leq \bar{R}^4$. Z tego też względu w modelu sensora w procedurze obsługi przysłonięć uwzględniono także dodatkowy warunek na przynależność do obszaru zakrycia wyznaczonego przez minimalny oraz maksymalny kąt w płaszczyźnie horyzontalnej (patrz rysunek 5.10) [21].

Na tym etapie przeanalizujemy jednak przesłonięcia na poziomie wykryć, zamiast na poziomie obiektów. Przyjmijmy, że ustalono wartości dla indeksów n_2 i $k \in [1, f(m_{n_2})]_{\mathbb{N}}$. Sztuczny pomiar reprezentowany przez identyfikatory n_2 oraz k , uzyskany dla obiektu o numerze 2 i zawarty w zbiorze P_d^t , jest zatem zasłonięty przez obiekt o numerze n_1 wtedy i tylko wtedy, gdy $\exists n_1 \in [2, N]_{\mathbb{N}}: n_1 \neq n_2 \wedge \bar{R}^{n_1} \leq \bar{R}^{n_2} \wedge \theta^{n_2, k} \in [\min(\theta)^{n_1}, \max(\theta)^{n_1}]$. Obsługa procedury przysłonięć sprowadza się więc do uwzględnienia tego kryterium w formule na elementy zbioru P_d^t ; należy oczywiście przeprowadzić negację wspomnianego warunku, gdyż w sztucznym zestawie pomiarów mają znaleźć się jedynie niezasłonięte wykrycia. Innymi słowy, zbiór P_d^t definiuje się według wzoru (5.5.10), w którym uwzględniono trzy omówione wyżej procedury, a zatem - likwidację punktów niewidocznych z perspektywy radaru, odrzucenie punktów będących poza polem widzenia wirtualnego czujnika oraz usunięcie przysłoniętych punktów:

$$P_d^t = \left\{ \{R, v_r, \theta, \phi\}^{n, k} \in \mathbb{R}^4 : (n, k) \in [2, N]_{\mathbb{N}} \times [1, f(m_n)]_{\mathbb{N}} \wedge \right. \\ \left. (\nexists n' \in [2, N]_{\mathbb{N}}: n' \neq n \wedge \bar{R}^{n'} \leq \bar{R}^n \wedge \theta^{n, k} \in [\min(\theta)^{n'}, \max(\theta)^{n'}]) \wedge \right. \\ \left. \alpha^{n, k} \leq \frac{\pi}{2} \wedge R^{n, k} \in [0, R_{\max}] \wedge \theta^{n, k} \in [\theta_{\min}, \theta_{\max}] \wedge \phi^{n, k} \in [\phi_{\min}, \phi_{\max}] \right\}. \quad (5.5.10)$$

Na rysunku 5.11 zaprezentowano wyniki działania opisanego algorytmu. Na rysunkach łatwo zauważyć, że przysłonięte wykrycia nie są uwzględnione w sztucznym zbiorze pomiarów.



Rysunek 5.11. Sztuczny zbiór wykryć wygenerowany przy użyciu wzoru (5.5.10), gdzie uwzględniono procedurę obsługi przysłonięć. Jak widać, zasłonięte pomiary nie są brane pod uwagę

5.5.5 Usunięcie pomiarów na podstawie testu statystycznego i dodanie zakłóceń

Opisane do tej pory operacje wykonywane w modelu sensora są w pełni deterministyczne. Nie odzwierciedla to niestety charakterystyki układów typu FMCW. Przede wszystkim pomiary radarowe oszacowane na podstawie odebranego sygnału elektromagnetycznego będą zawierać sporą liczbę zakłóceń. Dodatkowo jak zostało wspomniane w rozdziale 2, niektóre wykrycia klasyfikowane są jako fałszywe. Dlatego oprogramowanie realizujące proces przetwarzania sygnałów ustala na bieżąco wartość tak zwanego stosunku sygnału do szumu (ang. signal to noise ratio, SNR) i na tej podstawie wyznacza adaptacyjny próg odcięcia [58] wykorzystywany do usunięcia części pomiarów z uzyskanego zbioru danych. Wynika z tego, że rzeczywisty zbiór wykryć radarowych jest nie tylko obciążony znacznymi niepewnościami pomiarowymi, ale charakteryzuje się także tym, iż uzyskane punkty są stosunkowo rzadko rozmieszczone w przestrzeni. Ze względu na ten fakt, w omawianym modelu sensora uwzględniono dwie dodatkowe operacje: dodanie zakłóceń do syntetycznych pomiarów oraz zlikwidowanie niektórych wykryć na podstawie testu statystycznego [21].

Wpierw przeanalizujemy procedurę generacji szumu. Proces ten bazuje głównie na wartości SNR wyliczonej dla wykrycia o ustalonych indeksach n i k . SNR oznaczone grecką literą $\chi^{n,k} \in \mathbb{R}$ oblicza się przekształcając tak zwane równanie radarowe - zdefiniowane w rozdziale 2 we wzorze (2.3.11), określające zależność pomiędzy maksymalnym zasięgiem i aktualnie ustawionymi parametrami systemu typu FMCW [57] - a więc na podstawie równania:

$$\chi^{n,k} = \frac{P_t G_t^2 \lambda^2 \rho^{n,k} G_r}{R^{4|n,k|} (4\pi)^3 L_s k T_0 \beta F_n}, \quad (5.5.11)$$

gdzie pierwszy górny indeks reprezentuje wykładnik potęgi, natomiast $R^{4|n,k|}$ i $\rho^{n,k}$ to kolejno: czwarta potęga odległości radialnej oraz skuteczna powierzchnia odbicia (ang. radar cross section, RCS) obliczone dla wykrycia o indeksach n oraz k . Większość wielkości wymienionych w równaniu 5.5.11 jest stała i zależy od specyfikacji danego układu klasy FMCW [21]. Jedyną nieznaną wartość to $\rho^{n,k} \in \mathbb{R}$. Przypomnijmy, że RCS oszacowana dla danego celu określa siłę, z jaką fala elektromagnetyczna może odbić się od tego celu. Na wartość RCS wpływają właściwości obiektu, takie jak: pole powierzchni przekroju widzianego z perspektywy radaru, materiał, z jakiego został wykonany, oraz ukształtowanie jego powierzchni. Przywołajmy także fakt, iż każdy sztuczny pomiar powiązany jest z dokładnie jednym trójkątem (a więc i z dokładnie jednym centrum rozpraszania) wchodzącym w skład modelu geometrycznego danego obiektu. Co za tym idzie, wartość $\rho^{n,k}$ oblicza się na podstawie równania (5.5.12), bazującego na parametrach syntetycznego wykrycia oraz centrum rozpraszania (ze zbioru $P_{sc}^{SCS,t}$, określonego we wzorze (5.4.11)) o ustalonych indeksach n i k :

$$\rho^{n,k} = \kappa_m A^{n,k} \cos \theta^{n,k}, \quad (5.5.12)$$

gdzie $\kappa_m \in \mathbb{R}$ to współczynnik materiału dobrany eksperymentalnie [21]. Jak można zauważyć, $\chi^{n,k}$ zależy od wielkości danego wielokąta (poprzez element $\rho^{n,k}$). Dlatego w procedurze dodania zakłóceń autorzy modelu zdecydowali się wykorzystać znormalizowaną wartość SNR, $\chi^{n,k'}$, odpowiadającą jednostkowej powierzchni:

$$\chi^{n,k'} = \frac{\chi^{n,k}}{A^{n,k}}. \quad (5.5.13)$$

Przyjmuje się, że szum obecny w pomiarach radarowych może być reprezentowany przez zmienną losową X o rozkładzie normalnym, z zerową wartością oczekiwaną μ i odchyleniem standardowym σ , definiowanym na podstawie SNR. Tak więc w modelu sensora zdefiniowane są dwa rozkłady prawdopodobieństwa - osobno dla pomiaru odległości oraz osobno dla pomiaru kąta, kolejno równania:

$$\sigma_R^{n,k} = \frac{1}{3} \zeta_R \left(1 + \frac{\chi_0}{\chi^{n,k'}}\right), \quad \sigma_\theta^{n,k} = \frac{1}{3} \zeta_\theta \left(1 + \frac{\chi_0}{\chi^{n,k'}}\right), \quad (5.5.14a-b)$$

gdzie ζ_R , ζ_θ i χ_0 to odpowiednio dokładność pomiaru odległości oraz kąta i referencyjna wartość SNR (parametry te odczytuje się ze specyfikacji konkretnego systemu FMCW). Procedura generacji zakłóceń polega na dodaniu wartości $\sigma_R^{n,k}$ oraz $\sigma_\theta^{n,k}$ do elementów sztucznego wykrycia, na podstawie równań [21]:

$$R^{n,k} = R^{n,k} + \sigma_R^{n,k}, \quad \theta^{n,k} = \theta^{n,k} + \sigma_\theta^{n,k}, \quad \phi^{n,k} = \phi^{n,k} + \sigma_\phi^{n,k}. \quad (5.5.15a-c)$$

Drugą procedurą jest zmniejszenie gęstości sztucznych pomiarów radarowych w przestrzeni. Innymi słowy, wartości znajdujące się poniżej zdefiniowanej granicy (o niewystarczającej sile sygnału) klasyfikowane są jako szum i zostają usunięte ze zbioru wykryć. Przede wszystkim należy wyznaczyć z wykorzystaniem równania (5.5.16) moc odebranego sygnału $P_{syg}^{n,k}$ dla syntetycznego pomiaru o ustalonych indeksach n i k :

$$P_{syg}^{n,k} = \frac{P_t G_t^2 \lambda^2 \rho^{n,k} G_r}{R^4 |n,k| (4\pi)^3 L_s}, \quad (5.5.16)$$

gdzie pierwszy górny indeks reprezentuje wykładnik potęgi. Następnie zakłada się, że szum obecny na wejściu odbiornika systemu typu FMCW można opisać przy użyciu rozkładu normalnego $N(0, \sigma_{sys}^2)$ o zerowej wartości oczekiwanej i wariancji σ_{sys}^2 obliczanej na podstawie równania:

$$\sigma_{sys}^2 = kT_0 \beta F_n G_r. \quad (5.5.17)$$

Proces usunięcia części pomiarów ze zbioru sztucznych wykryć bazuje na uproszczonej wersji testu Neymana-Pearsona [104]. Przyjmuje on, że jeśli siła sygnału $x_{est}^{n,k} \in \mathbb{R}$ oszacowana dla danego syntetycznego pomiaru jest mniejsza od wartości progowej $x_{th} \in \mathbb{R}$, to sztuczne wykrycie zostaje usunięte z uzyskanego zbioru danych; $x_{est}^{n,k}$ oraz x_{th} zdefiniowano odpowiednio w równaniach [21]:

$$x_{est}^{n,k} = \sqrt{P_{syg}^{n,k}} + N(0, \sigma_{sys}^2), \quad x_{th} = \kappa_{PFA} \sqrt{2\sigma_{sys}^2}, \quad (5.5.18a-b)$$

gdzie $\kappa_{PFA} \in \mathbb{R}$ to współczynnik uzyskany przy użyciu specyfikacji danego układu klasy FMCW bazujący na prawdopodobieństwie wystąpienia fałszywego wykrycia [22]. Ostateczną postać definicji sztucznego zbioru wykryć P_d^t uwzględniającą wszystkie procedury wykonywane przez model sensora zaprezentowano we wzorze (5.5.19); do parametrów pojedynczego syntetycznego pomiaru dodano także wartości $\chi^{n,k}$ oraz $\rho^{n,k}$, gdyż jest to wymagane przez standard ASAM OSI:

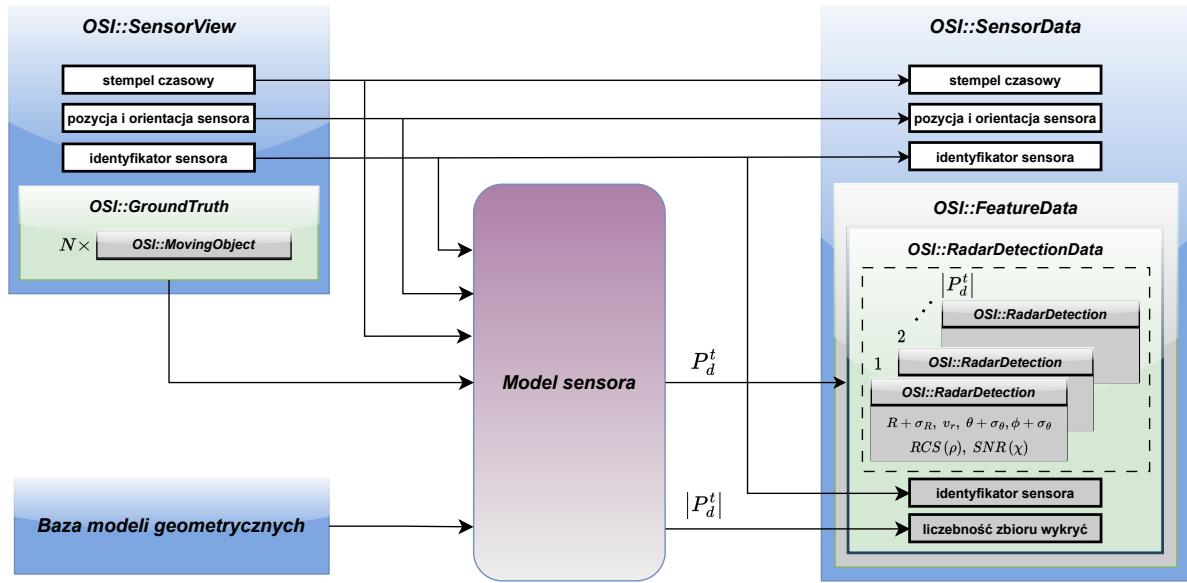
$$P_d^t = \left\{ \left\{ R + \sigma_R, v_r, \theta + \sigma_\theta, \phi + \sigma_\phi, \chi, \rho \right\}^{n,k} \in \mathbb{R}^6 : (n, k) \in [2, N]_{\mathbb{N}} \times [1, f(m_n)]_{\mathbb{N}} \wedge \right. \\ \left. x_{est}^{n,k} \geq x_{th} \wedge (\nexists n' \in [2, N]_{\mathbb{N}} : n' \neq n \wedge \bar{R}^{n'} \leq \bar{R}^n \wedge \theta^{n,k} \in [\min(\theta)^{n'}, \max(\theta)^{n'}]) \wedge \right. \\ \left. \alpha^{n,k} \leq \frac{\pi}{2} \wedge R^{n,k} \in [0, R_{\max}] \wedge \theta^{n,k} \in [\theta_{\min}, \theta_{\max}] \wedge \phi^{n,k} \in [\phi_{\min}, \phi_{\max}] \right\}. \quad (5.5.19)$$

5.6 Wyjście z modelu zgodne ze standardem ASAM OSI

Model sensora kompatybilny ze standardem ASAM OSI musi zapisać w odpowiedni sposób zbiór sztucznych pomiarów w strukturze *OSI::SensorData*. Jak zaprezentowano na rysunku 5.1, pierwszą kategorią informacji, którą przekazuje się na wejście modelu sensora są elementy zbioru

OSI::GroundTruth, czyli N obiektów klasy *OSI::MovingObject* o typie *OSI::TYPE_VEHICLE*. Danych tych nie należy uwzględniać w wiadomości *OSI::SensorData*. Drugim rodzajem informacji są następujące parametry: identyfikator wirtualnego czujnika, jego pozycja i orientacja oraz wartość stempla czasowego. Z kolei elementy składowe struktury *OSI::SensorView* kopiuje się i zapisuje w niezmięnionej formie w wiadomości *OSI::SensorData*, co można zobaczyć na rysunku 3.2. Warto nadmienić, że baza modeli geometrycznych wykorzystywana w procesie generacji zbioru centrów rozpraszania $P_{sc}^{SCS, t}$ zdefiniowanego we wzorze (5.4.11) również nie jest częścią struktury *OSI::SensorData*.

Model sensora zwraca zbiór sztucznych pomiarów P_d^t określony według formuły (5.5.19). Zestaw ten należy wpisać do struktury *OSI::RadarDetectionData* będącej częścią wiadomości *OSI::FeatureData* (patrz rysunek 3.2), gdzie każda szóstka parametrów określona we wzorze (5.5.19) powiązana jest z dokładnie jednym zbiorem *OSI::RadarDetection*. Przykładowy przepływ informacji - od uzyskania wiadomości *OSI::SensorView* o stemple czasowym t , poprzez wywołanie wszystkich procedur zdefiniowanych w modelu sensora, aż po wygenerowanie poprawnej struktury *OSI::SensorData* - zaprezentowano na rysunku 5.12, gdzie $|P_d^t|$ to liczba elementów zbioru P_d^t .



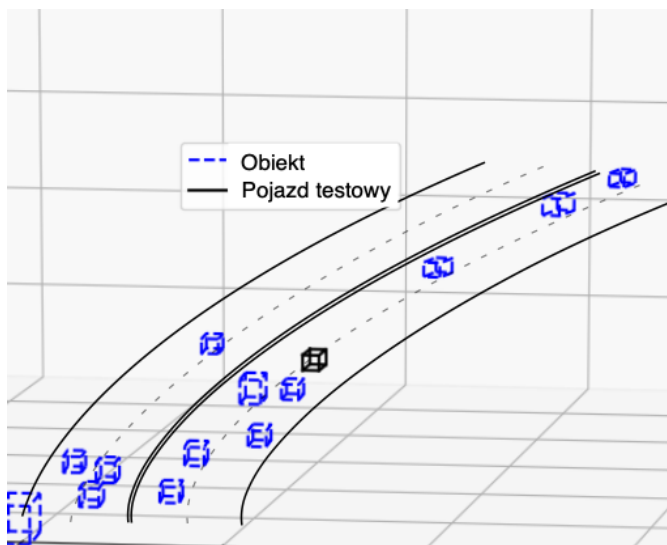
Rysunek 5.12. Przykładowy przepływ informacji w modelu symulacyjnym czujnika radarowego kompatybilnym ze standardem ASAM OSI

5.7 Wstępna ocena wiarygodności modelu referencyjnego

W tym rozdziale wyjaśniono dokładnie wszystkie procedury zawarte w oryginalnym modelu symulacyjnym czujnika radarowego. Dodatkowo, teorię opisaną w [21] rozszerzono o formalne aspekty związane z kompatybilnością modelu ze standardem ASAM OSI. Ponadto, w bieżącym rozdziale zaproponowano także kompaktową definicję zbioru sztucznych wykryć radarowych określoną we

wzorze (5.5.19). Warto nadmienić, że omówiona integracja ze standardem ASAM OSI jest kluczowa w kontekście tezy pracy, która mówi o użyteczności danych syntetycznych wykorzystywanych do testów systemu klasy ADAS w symulatorze jazdy. Przypomnijmy pojęcia wprowadzone w rozdziale 1 - model uznaje się za użyteczny wtedy, gdy jest on wiarygodny i wydajny (zdolny do pracy w czasie rzeczywistym) niezależnie od poziomu skomplikowania scenariusza wirtualnej jazdy.

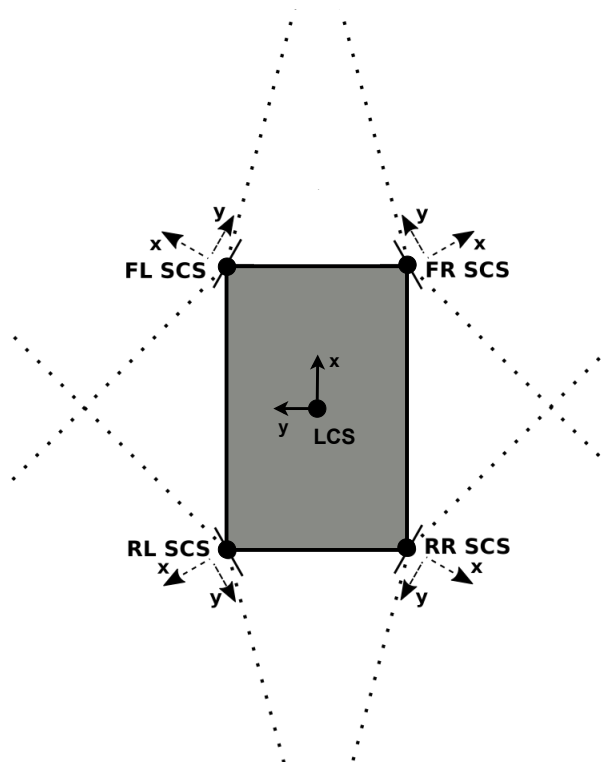
Warto więc na tym etapie przeanalizować wstępnie wiarygodność modelu sensora. W tym celu trzeba zestawiać sztuczne pomiary z odczytami z rzeczywistego czujnika radarowego. Jak wspomniano w rozdziale 1, do kluczowych systemów typu ADAS należy zaliczyć m.in. układy klasy AEBS oraz ACC, które są w znacznej mierze oparte na czujnikach radarowych typu FMCW, ze względu na możliwość bezpośredniego oszacowania prędkości względnej obiektów. Kluczowym środowiskiem testowym z perspektywy weryfikacji urządzeń typu ACC czy AEBS będzie dwupasmowa droga szybkiego ruchu, bez przecinających się jezdni. Przeprowadzono zatem krótki przejazd samochodem testowym w warunkach spełniających wspomniane wymagania. W trakcie scenariusza jazdy rejestrowano dane z czujników radarowych krótkiego zasięgu typu FMCW. Dodatkowo zebrano także informacje z sensora referencyjnego w celu wygenerowania wymaganej przez model sensora wiadomości *OSI::SensorView* - z poprawnie utworzonymi obiektami klasy *OSI::MovingObject*. Obiekty typu *OSI::MovingObject* dostępne w strukturze *OSI::SensorView*, uzyskanej dla przykładowego momentu czasu, z poglądowo naniesionymi pasami drogi, zaprezentowano na rysunku 5.13.



Rysunek 5.13. Obiekty klasy *OSI::MovingObject* zawarte w wiadomości *OSI::SensorView* wygenerowanej dla pewnego ustalonego momentu czasu nagranego scenariusza jazdy

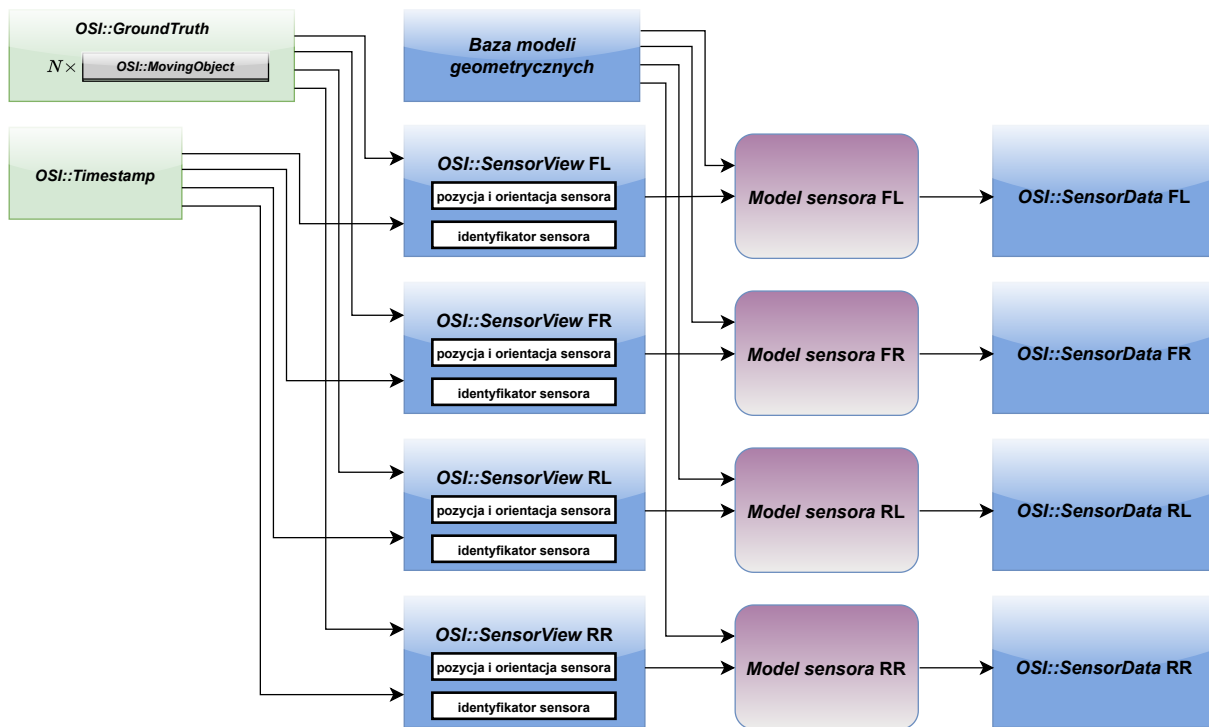
Sposób montażu czujników radarowych na pojeździe testowym zaprezentowano w formie graficznej na rysunku 5.14. Na wspomnianym rysunku widać, że samochód wyposażono w cztery identyczne sensory typu FMCW, które oznaczono dwuliterowymi skrótami, odpowiednio - lewy przedni (FL), prawy

przedni (FR), lewy tylny (RL) oraz prawy tylny (RR). Co za tym idzie, w omawianym przypadku określa się pięć układów współrzędnych: układ współrzędnych obiektu (LCS) oraz cztery układy współrzędnych powiązane z czujnikami (SCS_{FL} , SCS_{FR} , SCS_{RL} i SCS_{RR}).



Rysunek 5.14. Uproszczony schemat mocowania czujników radarowych na samochodzie testowym

Porównanie pomiędzy danymi rzeczywistymi a syntetycznymi będzie bazować na pomiarach ze wszystkich radarów, które różnią się między sobą sposobem zamocowania na pojeździe testowym, a zatem inna jest dla każdego z nich pozycja i orientacja wyrażona względem układu LCS. Należy więc wygenerować cztery różne zestawy sztucznych pomiarów. Dlatego też dla każdego rzeczywistego czujnika trzeba utworzyć jego odpowiednik w postaci modelu sensora o identycznym umiejscowieniu w układzie LCS i o takim samym identyfikatorze. Innymi słowy, w procesie generacji danych syntetycznych wykorzystane będą cztery instancje modelu sensora o dostarczonej na wejściu unikalnej wiadomości *OSI::SensorView*; oczywiście zbiór *OSI::GroundTruth* oraz wartość stempla czasowego są identyczne dla każdej struktury *OSI::SensorView*. Przepływ informacji dla tego przypadku pokazano na rysunku 5.15, bazującym na rysunku 5.12.



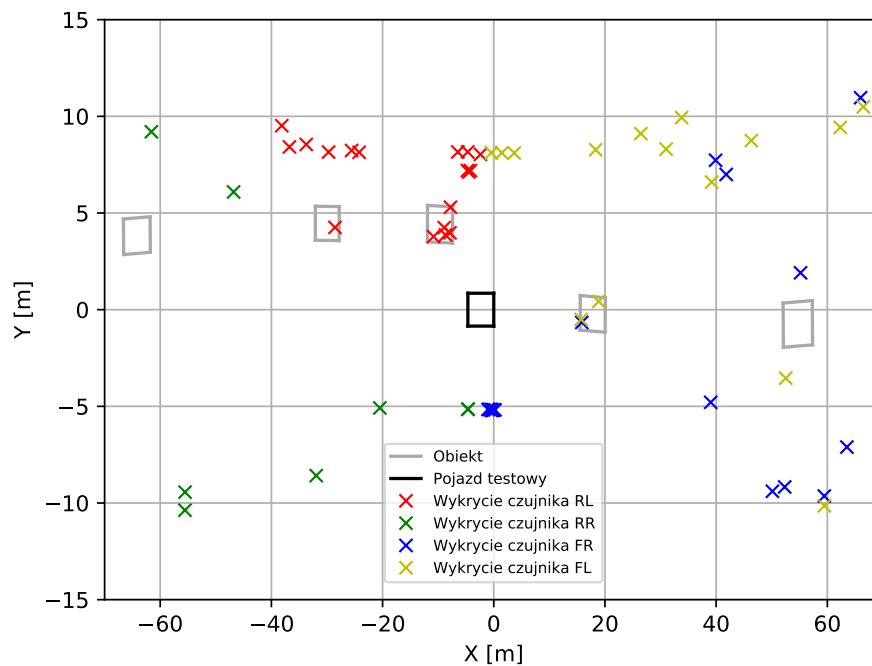
Rysunek 5.15. Przepływ danych przy użyciu czterech instancji modelu sensora

Na rysunku 5.16 zaprezentowano dane z rzeczywistych czujników radarowych. Natomiast na rysunku 5.17 pokazano sztuczne wykrycia radarowe uzyskane przy użyciu czterech instancji oryginalnego modelu sensora. Warto zaznaczyć, że na rysunkach 5.16 oraz 5.17 uwzględniono także obiekty klasy *OSI::MovingObject* zawarte w zbiorze *OSI::GroundTruth* przekazany w pewnym dyskretnym momencie czasu do wszystkich wiadomości *OSI::SensorView*.

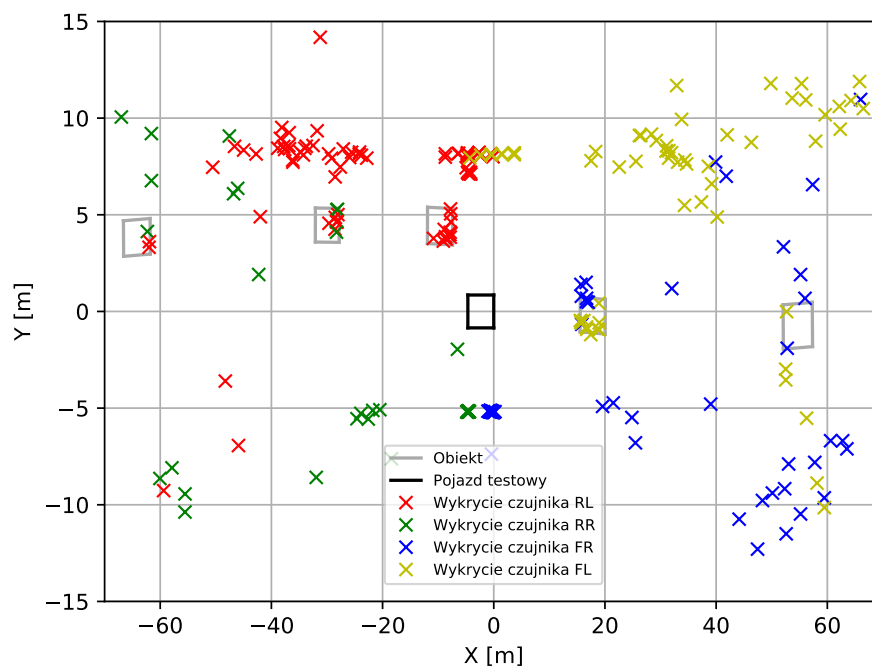
Jak można zaobserwować na rysunku 5.17, wszystkie sztuczne pomiary pokrywają się z odpowiadającymi im obiektami klasy *OSI::MovingObject*. Dodatkowo poprawnie została odwzorowana pozycja, orientacja i pole widzenia każdego wirtualnego czujnika, a więc zgodnie z rysunkiem 5.14. Ponadto dla obiektu znajdującego się po prawej stronie wykresu, w odległości około 55 [m] od pojazdu testowego, można zaobserwować skutki wystąpienia efektu przysłonięć oraz zastosowania testu statystycznego, gdyż dla tego konkretnego celu liczba sztucznych wykryć jest zdecydowanie mniejsza w porównaniu z innymi obiektami. Wynika z tego, że przekształcenia pomiędzy układami współrzędnych oraz wszystkie operacje wykonywane przez model sensora można uznać za poprawne.

Niemniej jednak istnieje parę istotnych różnic pomiędzy syntetycznymi wykryciami i danymi rzeczywistymi. Po pierwsze, w przypadku modelu sensora liczba punktów utworzonych dla pojedynczych brył wydaje się zbyt duża - rzeczywisty czujnik zbliża się do gęstości pomiarowej modelu dopiero po kilku okresach czasu. Po drugie, analizując dane z sensora klasy FMCW z rysunków 5.16a oraz 5.16b wyraźnie widać, że znaczna część wykryć zwróconych przez radar powstaje na skutek odbicia od barierek wyznaczających krawędzie jezdni. Po trzecie, w rzeczywistych pomiarach można również

zidentyfikować fałszywe wykrycia niepowiązane w żaden sposób z elementami scenariusza testowego. Przypomnijmy, iż aktualna wersja modelu sensora zwraca zbiór sztucznych wykryć jedynie dla pojazdów klasy *OSI::MovingObject*. Konieczne jest zatem dodanie do oryginalnego modelu kilku usprawnień zwiększających jego poziom wiarygodności. W szczególności trzeba uwzględnić w wirtualnym czujniku procedury, których rezultatem będzie zmniejszona gęstość pomiarowa oraz obecność w sztucznym zbiorze danych fałszywych wykryć i pomiarów powiązanych z obiektami statycznymi.

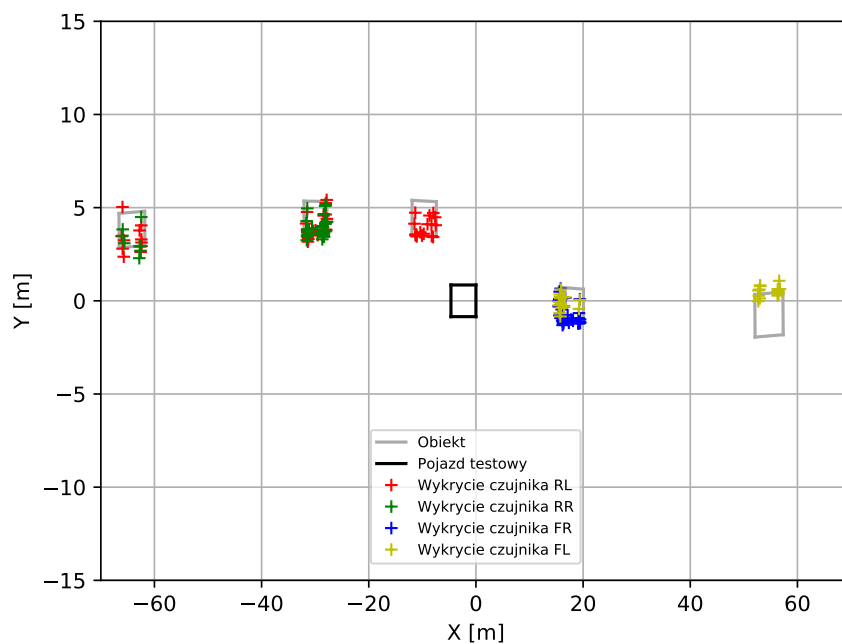


(a) Pojedynczy cykl pomiarowy

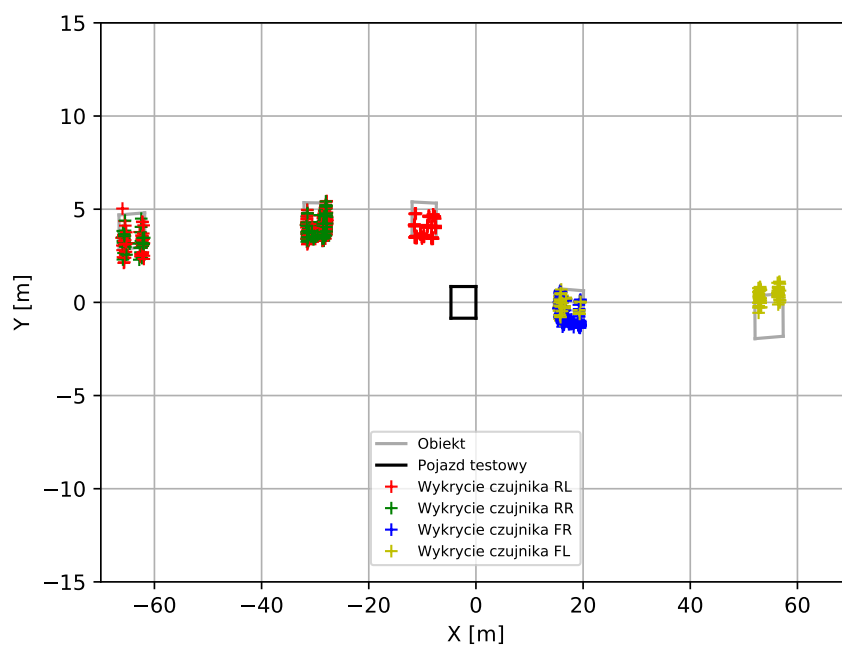


(b) Dane z pięciu kolejnych okresów czasu

Rysunek 5.16. Dane z rzeczywistych czujników radarowych. Na rysunku 5.16a pokazano wykrycia dla pojedynczego cyklu pomiarowego. Z kolei na rysunku 5.16b widać wykrycia zebrane z pięciu kolejnych okresów czasowych



(a) Pojedynczy cykl pomiarowy



(b) Dane z pięciu kolejnych okresów czasu

Rysunek 5.17. Dane uzyskane na podstawie czterech instancji modelu sensora. Na rysunku 5.17a pokazano sztuczne wykrycia dla pojedynczego cyklu pomiarowego. Z kolei na rysunku 5.17b widać syntetyczne pomiary zebrane z pięciu kolejnych okresów czasowych

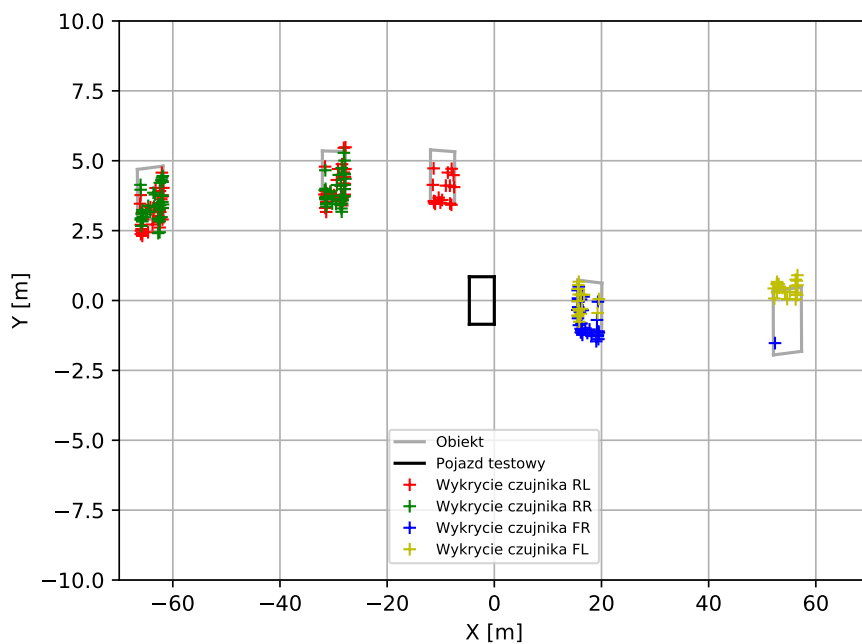
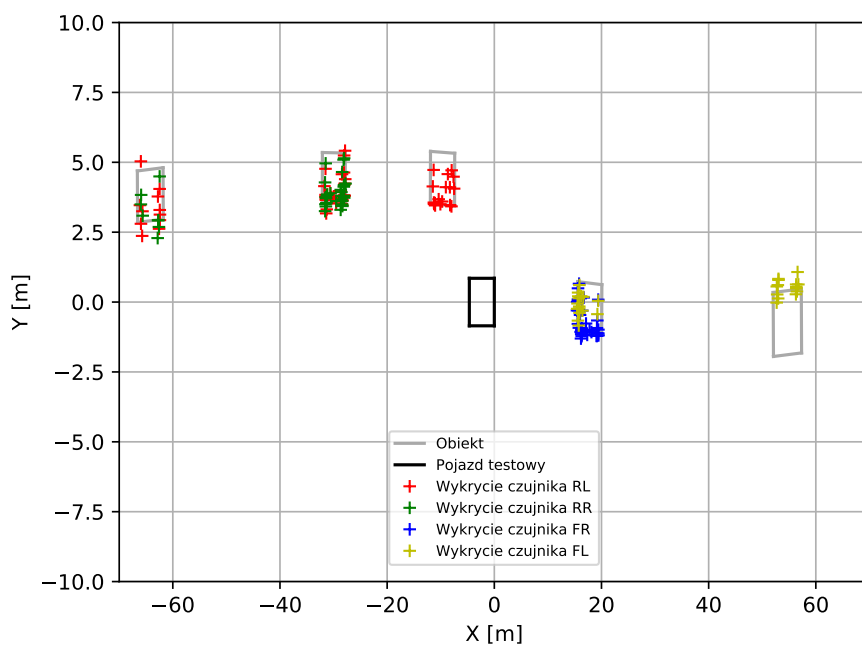
6 Usprawnienie modelu bazowego

W tym rozdziale zaprezentowane będą modyfikacje dodane do oryginalnego modelu sensora, zwiększające wiarygodność zbioru sztucznych wykryć. Usprawnienia te zostały zidentyfikowane w poprzednim rozdziale na podstawie pomiarów z rzeczywistych czujników radarowych zebranych w trakcie jazdy testowej.

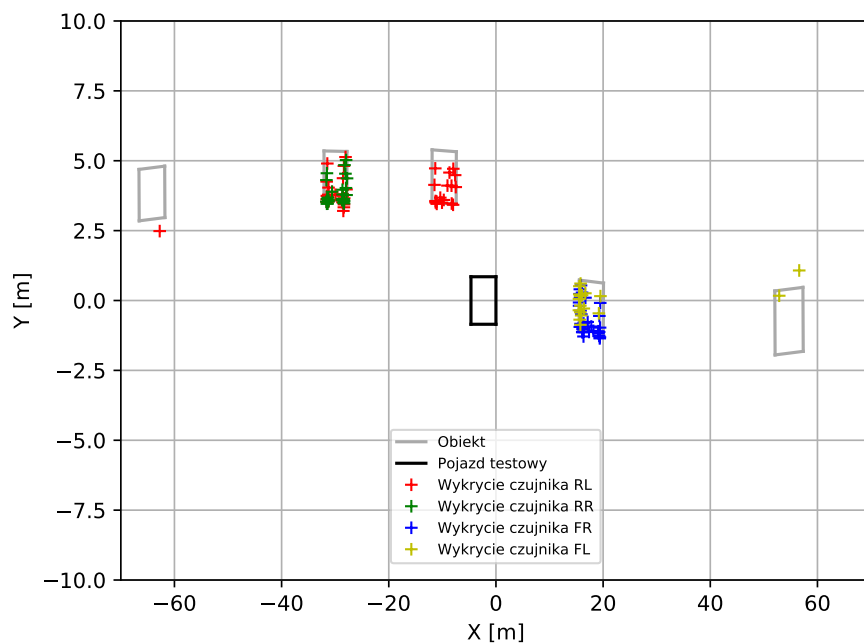
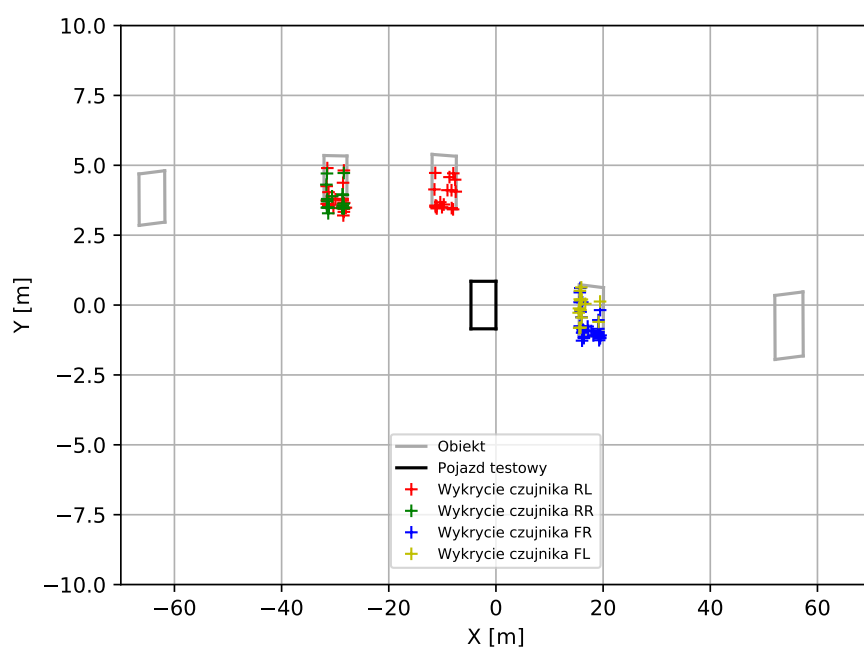
6.1 Filtr rozdzielczości

Pierwszym rozpoznanym problemem w bazowym modelu sensora jest jego zbyt duża gęstość pomiarowa, a więc nadmiernie wysokie nasycenie punktów w trójwymiarowej przestrzeni w ramach pojedynczego zbioru sztucznych wykryć.

Jak wspomniano w poprzednim rozdziale, ostatnia procedura wykonywana w modelu sensora to przeprowadzenie testu statystycznego. Jego rola polega na zdecydowaniu, czy wykrycie o ustalonych indeksach n i k znajdzie się w wyjściowym zbiorze danych zapisanym w strukturze *OSI::SensorData*; syntetyczny pomiar nie jest uwzględniony w wiadomości *OSI::SensorData* wtedy, gdy oszacowana moc odebranego sygnału $x_{est}^{n,k}$ jest mniejsza od obliczonej wartości progowej x_{th} . Gęstość pomiarową można zmienić modyfikując ustawienia w taki sposób, by próg odcięcia był bardziej restrykcyjny. Dobranie idealnych parametrów modelu na podstawie przyjętych wskaźników jakości jest oczywiście zagadnieniem istotnym. Będzie to omówione dokładnie w kolejnym rozdziale. Niestety, okazuje się, że modyfikacja wartości mających wpływ na rezultat eksperymentu Neymana-Pearsona jest niewystarczająca. By zademonstrować istotę problemu, wygenerowano sztuczne pomiary dla kilku różnych ustawień testu statystycznego, bazując na koncepcji symulacji zaprezentowanej w poprzednim rozdziale na rysunku 5.15 wykorzystującej cztery instancje modelu sensora o identycznych wartościach parametrów. Warto nadmienić, iż w omawianym eksperymencie modyfikowano jedynie parametr F_n , który należy traktować jak wzmocnienie - wraz ze zwiększającą się wartością F_n rośnie próg odcięcia, przez co z wyjściowego zbioru danych usuwa się znacznie więcej oszacowanych pomiarów. Rezultaty pokazano na rysunkach 6.1 i 6.2, gdzie wartości F_n to kolejno: -100 [dB], -20 [dB], -16 [dB] oraz -12 [dB] (rysunki 6.1a, 6.1b, 6.2a, jak również 6.2b).

(a) $F_n = -100$ [dB](b) $F_n = -20$ [dB]

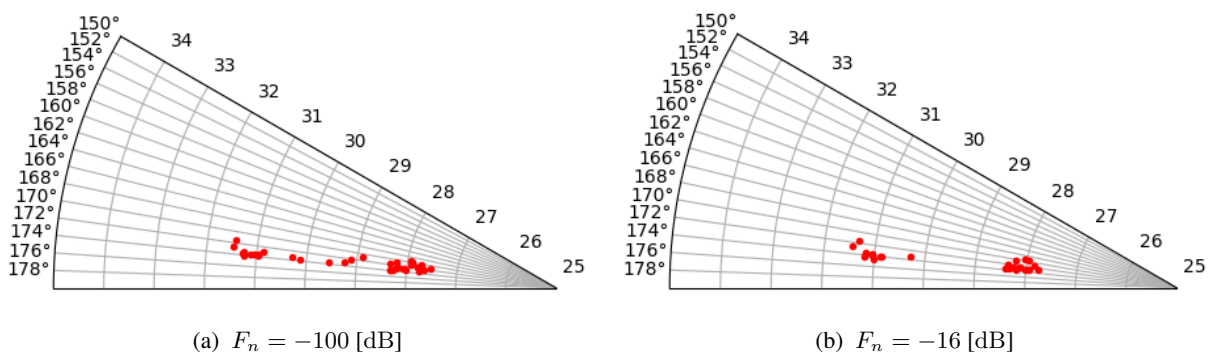
Rysunek 6.1. Zbiory sztucznych wykryć wygenerowane dla pojedynczego cyklu pomiarowego przez cztery instancje modelu sensora dla niskich wartości wzmocnienia testu statystycznego

(a) $F_n = -16$ [dB](b) $F_n = -12$ [dB]

Rysunek 6.2. Zbiory sztucznych wykryć wygenerowane dla pojedynczego cyklu pomiarowego przez cztery instancje modelu sensora dla wysokich wartości wzmocnienia testu statystycznego

Z załączonych rysunków, 6.1 i 6.2, można wyciągnąć kilka istotnych wniosków. Po pierwsze, ustawienie znikomej wartości parametru F_n (rzędu -100 [dB]) jest równoznaczne z pominięciem testu statystycznego, gdyż wszystkie syntetyczne wykrycia znajdują się powyżej progu odcięcia. Po drugie, zgodnie z założeniem, wraz ze zwiększającym się nominałem F_n wzrasta też wpływ procesu odsiewania i rośnie liczebność usuniętych sztucznych pomiarów. Niestety, oszacowana siła sygnału $x_{est}^{n,k}$ w znacznym stopniu zależy od odległości $R^{n,k}$. Efekt tej relacji łatwo zaobserwować na rysunku 6.2b, gdzie ze względu na zbyt wysoką wartość wzmocnienia testu statystycznego zabrakło w zbiorach sztucznych wykryć pomiarów oszacowanych dla obiektu znajdującego się w odległości 60 [m] od pojazdu testowego. Jednakże nasycenie punktów dla celów oddalonych od wirtualnego czujnika o około 20 [m] ulega stosunkowo niewielkiej zmianie i jest wciąż zbyt duże.

Ze względu na to do oryginalnego modelu sensora została dodana dodatkowa operacja, tzw. filtr rozdzielczości, której celem będzie zmniejszenie liczby punktów pomiarowych dla bliżej położonych obiektów bez konieczności ustawienia rygorystycznego progu odcięcia, i co za tym idzie, uzyskanie gęstości charakterystycznej dla rzeczywistego radaru typu FMCW. Przeanalizujemy zatem zbiór sztucznych wykryć wygenerowany przez model sensora dla dwóch trybów pracy: test statystyczny wyłączony (F_n ustawione na -100 [dB]) oraz restrykcyjny próg odcięcia (F_n równe -16 [dB]). Jako przykład posłuży tutaj model sensora powiązany z rzeczywistym czujnikiem radarowym oznaczonym jako lewy tylny. W szczególności interesujące będą syntetyczne pomiary oszacowane dla obiektu znajdującego się w odległości około 30 [m] od pojazdu testowego. Rezultaty uzyskane dla tego celu przy użyciu wspomnianego modelu sensora pokazano na rysunku 6.3. Jest to wykres we współrzędnych biegunowych - rysunki 6.3a oraz 6.3b prezentują wyniki z F_n ustawionym odpowiednio na -100 [dB] i -16 [dB].



Rysunek 6.3. Porównanie zbioru sztucznych wykryć skojarzonych z pojedynczym obiektem utworzonych dla pojedynczego cyklu pomiarowego dla dwóch ustawień parametrów testu statystycznego. Model sensora wygenerował 36 wykryć dla przypadku po lewej stronie. Natomiast przy F_n ustawionym na -16 [dB] liczebność zbioru zmniejszyła się do 25 punktów

Liczebność zbioru wykryć wygenerowanego przez pojedynczy model sensora dla obiektu wspomnianego na rysunku 6.3 zmniejszyła się zatem z 36 do 25 pomiarów. Spadek ten jest niewystarczający, biorąc pod uwagę fakt, iż przy F_n ustawionym na -16 [dB] obiekty znajdujące się w odległości większej niż około 60 [m] niemal przestają być widoczne z perspektywy wirtualnego czujnika.

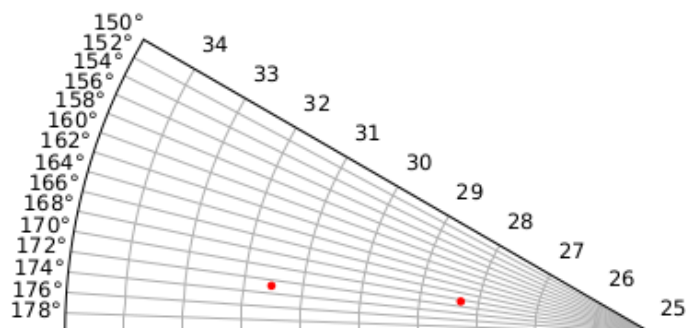
Jak wspomniano w rozdziale 2, dane zwracane przez radar typu FMCW dla pojedynczego cyklu pomiarowego reprezentuje się często przy użyciu trójwymiarowej, zespolonej macierzy danych, czyli tzw. macierzy RDC (patrz rysunek 2.6). Każdy element macierzy RDC można jednoznacznie połączyć z konkretną wartością odległości, prędkości względnej oraz kąta w płaszczyźnie horyzontalnej. Z kolei zmiana parametrów wykrycia pomiędzy sąsiadującymi komórkami zbioru RDC powiązana jest z rozdzielczością pomiarową radaru klasy FMCW, którą określają następujące wartości: ΔR , Δv_r i $\Delta \theta$; parametry te zdefiniowano odpowiednio w równaniach (2.3.9a) - (2.3.9c). Innymi słowy, w zbiorze wykryć oszacowanym przez czujnik radarowy nigdy nie będzie dwóch pomiarów przypisanych do pojedynczej komórki RDC.

Podobny mechanizm należy zatem odwzorować w modelu sensora. W tym celu wygenerowano siatkę w biegunowym układzie współrzędnych. Jako ΔR przyjęto wartość 0.75 [m], natomiast $\Delta \theta$ ustawiono na 1.5 [°]. Procedura filtracji jest następująca: jeśli w pojedynczej komórce utworzonej siatki znajduje się więcej niż jeden punkt pomiarowy, to jako punkt poprawny traktuje się wykrycie o najwyższej sile sygnału - pozostałe elementy syntetycznego zbioru danych są usuwane. Oznaczmy przez $\hat{P}_d^t$ zbiór sztucznych wykryć będący podzbiorem zbioru P_d^t określonego według formuły (5.5.19), a zatem: $\hat{P}_d^t \subset P_d^t$. Definicję zbioru $\hat{P}_d^t$ zawierającą warunki, których uwzględnienie jest jednoznaczne ze zrealizowaniem opisanej procedury określonej mianem filtra rozdzielczości, zaprezentowano we wzorze (6.1.1) bazującym na oznaczeniach zastosowanych w równaniu (5.5.19):

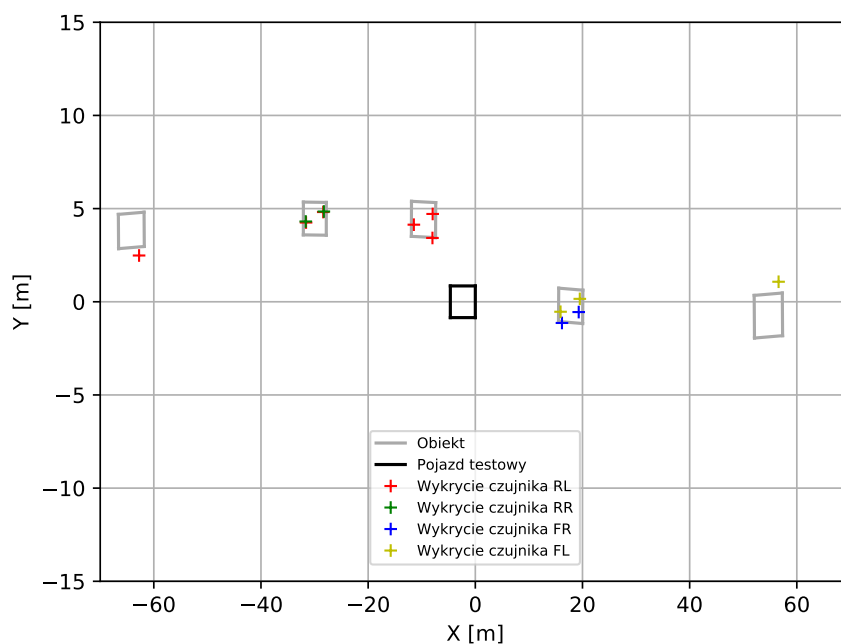
$$\hat{P}_d^t = \left\{ \{R + \sigma_R, v_r, \theta + \sigma_\theta, \phi + \sigma_\phi, \chi, \rho\}^{n,k} \in P_d^t : (n, k) \in [2, N]_{\mathbb{N}} \times [1, f(m_n)]_{\mathbb{N}} \wedge \right. \\ \left. (\nexists n' \in [2, N]_{\mathbb{N}} : n' \neq n \wedge x_{est}^{n',k} \geq x_{est}^{n,k} \wedge |R^{n',k} - R^{n,k}| \leq \Delta R \wedge |\theta^{n',k} - \theta^{n,k}| \leq \Delta \theta) \right\}, \quad (6.1.1)$$

gdzie $|\cdot|$ to wartość bezwzględna danej liczby. Rezultat działania algorytmu zaprezentowano na rysunku 6.4. Po przeprowadzeniu tejże procedury tylko 2 z 36 sztucznych wykryć wygenerowanych dla omawianego obiektu sklasyfikowano jako poprawne. Dodatkowo na rysunku 6.5 pokazano syntetyczne pomiary radarowe uzyskane przy użyciu czterech instancji zmodyfikowanego modelu sensora, a więc bazującego na definicji zbioru sztucznych wykryć zapisanej we wzorze (6.1.1). Rysunek 6.5 odnosi się do tego samego dyskretnego momentu czasu co rysunek 5.16a, gdzie zaprezentowano wykrycia wygenerowane przez rzeczywiste czujniki radarowe. Porównując oba rysunki widać, że gęstość pomiarowa uzyskana dla danych syntetycznych dla obiektów klasy *OSI::MovingObject* zawartych w

zbiorze *OSI::GroundTruth* wymaga jeszcze dostrojenia, jednak już na tym etapie wyniki są dużo bardziej zbliżone do tych zwracanych przez rzeczywiste sensory.



Rysunek 6.4. Zbiór sztucznych wykryć $\hat{P}_d^t$ (uwzględniający efekt filtracji rozdzielczości) utworzony dla pojedynczego cyklu pomiarowego. Dane syntetyczne skojarzone są z tym samym obiektem, który wykorzystano do wygenerowania wykryć pokazanych na rysunku 6.3. W tym przypadku parametr F_n ustawiono na -16 [dB]



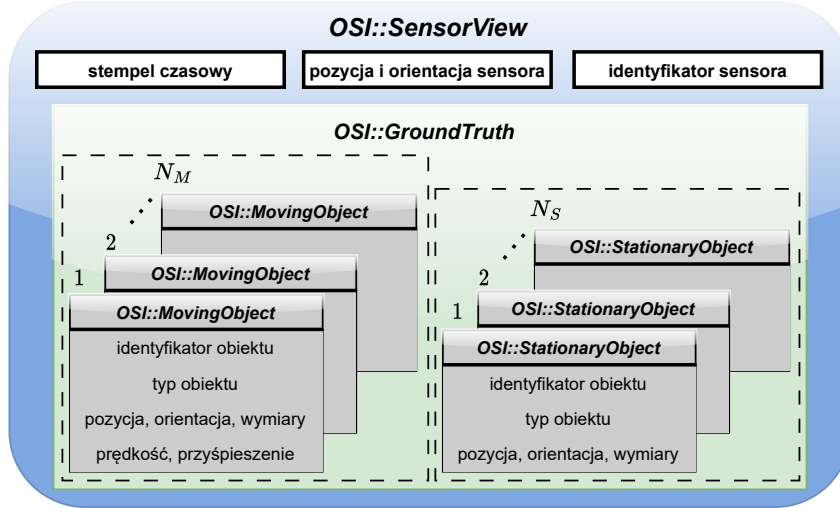
Rysunek 6.5. Zbiory sztucznych wykryć uzyskane na podstawie czterech instancji modelu sensora, bazujących na definicji określonej według formuły (6.1.1). Jest to ten sam moment czasu co na rysunku 5.16a, gdzie pokazano wykrycia z rzeczywistych czujników radarowych

6.2 Generacja wykryć dla obiektów statycznych

Na rysunku 5.16 wyraźnie widać, iż znaczna część pomiarów wytwarzanych przez radar typu FMCW powstaje na skutek odbicia od obiektów statycznych. W tym przypadku, czyli dla jazdy testowej po drodze szybkiego ruchu, źródłem pomiarów są barierki wyznaczające krawędź drogi. Widać to wyraźnie zwłaszcza na rysunku 5.16b, gdzie punkty zebrane z kilku okresów czasu układają się w regularne struktury, na których można bez problemu zidentyfikować granicę jezdni. Ze względu na liczebność zbioru wykryć generowanych przez rzeczywisty czujnik dla takiego typu struktur konieczne jest rozszerzenie modelu sensora o podobną funkcjonalność.

W standardzie ASAM OSI do zapisania informacji na temat danego obiektu stacjonarnego wykorzystuje się strukturę *OSI::StationaryObject*. Podobnie jak dla elementów typu *OSI::MovingObject*, obiekty klasy *OSI::StationaryObject* reprezentowane są przy użyciu brył brzegowych o unikalnych wymiarach, jednak w tym przypadku wektor prędkości jest zerowy. Przede wszystkim należy rozszerzyć bazę modeli geometrycznych, czyli jedno z wejść do modelu sensora, o strukturę wykorzystywaną w procesie generacji zbioru centrów rozpraszania dla nieporuszających się celów. Zakłada się, że wszystkie typy obiektów statycznych wymienione w zbiorze *OSI::StationaryObject::Classification* korzystać będą z modelu geometrycznego zbudowanego na bazie sześcianu, odpowiednio przeskalowanego do wymiarów danej bryły brzegowej. Tego typu strukturę składającą się z 12 połączonych ze sobą trójkątów rozpiętych na 8 wierzchołkach tegoż sześcianu zaprezentowano już w poprzednim rozdziale na rysunku 5.4.

Zmodyfikujmy teraz drugie wejście do modelu sensora, a więc zawartość wiadomości *OSI::SensorView*, by możliwe było wygenerowanie sztucznych wykryć radarowych zarówno dla pojazdów, jak i elementów statycznych scenariusza wirtualnej jazdy. Przypomnijmy, że w oryginalnym modelu sensora korzystano jedynie z obiektów klasy *OSI::MovingObject* o typie *OSI::TYPE_VEHICLE* [122] zapisanych w strukturze *OSI::SensorView*. W tym przypadku wykorzystane będą również informacje na temat elementów *OSI::StationaryObject*. Załóżmy zatem, że w zbiorze *OSI::SensorView* o unikalnym stemplu czasowym t dostępnych jest N_M obiektów klasy *OSI::MovingObject*, gdzie pierwszy element (o identyfikatorze równym 1) odnosi się do pojazdu wiodącego, oraz N_S obiektów klasy *OSI::StationaryObject* ($N_M, N_S \in \mathbb{N}$). Oczywiście identyfikator każdego komponentu wchodzącego w skład zestawu *OSI::SensorView* powinien być unikatowy. Dlatego też przyjmuje się, że do pierwszego elementu typu *OSI::StationaryObject* przypisany jest identyfikator o wartości $N_M + 1$, natomiast ostatni element należy skojarzyć z identyfikatorem o wartości $N_M + N_S + 1$. Schemat poglądowy prezentujący w uproszczony sposób minimalną zawartość wiadomości *OSI::SensorView* wymaganą przez zmodyfikowany model sensora pokazano na rysunku 6.6 stanowiącym modyfikację rysunku 3.1.



Rysunek 6.6. Schemat poglądowy prezentujący w uproszczony sposób fragment struktury *OSI::SensorView* zawierający N_M obiektów klasy *OSI::MovingObject* oraz N_S obiektów klasy *OSI::StationaryObject* ($N_M, N_S \in \mathbb{N}$), wymagany do wygenerowania przez model sensora poprawnej wiadomości typu *OSI::SensorData* mieszczącej sztuczny zbiór wykryć radarowych

Przekształćmy teraz definicję zbioru centrów rozpraszania zaprezentowaną we wzorze (5.4.11) tak, by była zgodna z aktualnie przyjętą postacią struktury *OSI::SensorView* pokazaną na rysunku 6.6. Wartości zbioru $P_{sc}^{SCS, t} \in \mathbb{R}^{(N_M+N_S-1) \times f(m_n) \times 7}$ określa się na podstawie formuły:

$$P_{sc}^{SCS, t} = \left\{ \left\{ p_{sc}^{SCS}, v_{\perp}^{SCS}, A \right\}^{n, k} \in \mathbb{R}^7 : (n, k) \in [2, N_M + N_S + 1]_{\mathbb{N}} \times [1, f(m_n)]_{\mathbb{N}} \right\}. \quad (6.2.1)$$

Zatem surowy zbiór sztucznych wykryć, zdefiniowany oryginalnie we wzorze (5.5.1), określa się według formuły (6.2.2), przy czym zmodyfikowany zbiór $P_d^t \in \mathbb{R}^{(N_M+N_S-1) \times f(m_n) \times 4}$.

$$P_d^t = \left\{ \left\{ R, v_r, \theta, \phi \right\}^{n, k} \in \mathbb{R}^4 : (n, k) \in [2, N_M + N_S + 1]_{\mathbb{N}} \times [1, f(m_n)]_{\mathbb{N}} \right\}. \quad (6.2.2)$$

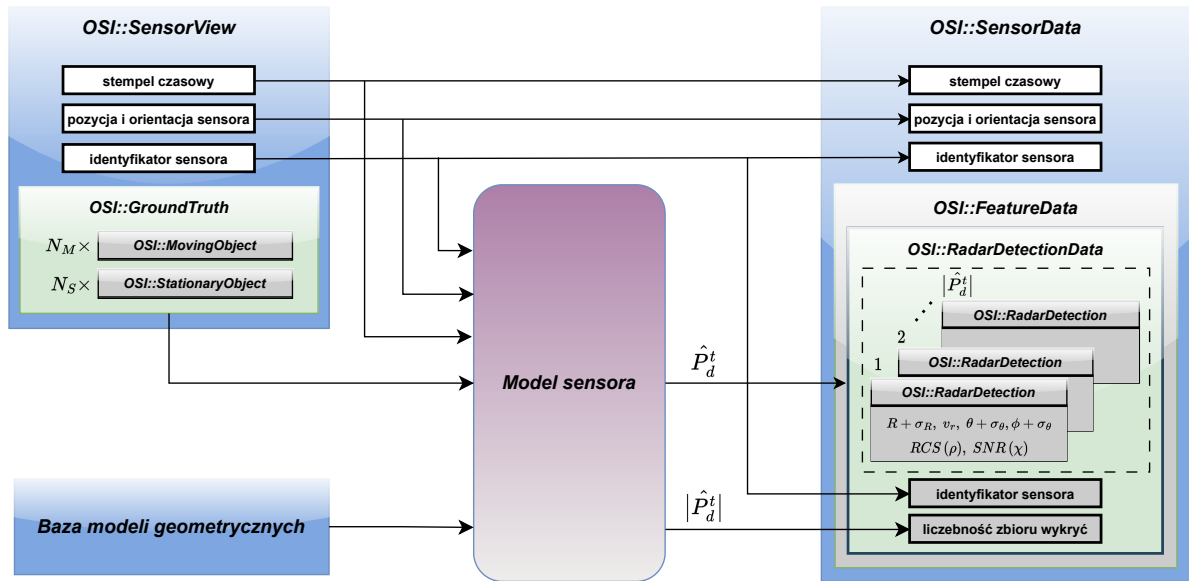
Wszystkie procedury charakterystyczne dla bazowego modelu sensora można zastosować w niezmienionej formie również dla wykryć skojarzonych z obiektami statycznymi. Co za tym idzie, końcowa formuła na zbiór syntetycznych pomiarów, zaprezentowana oryginalnie w równaniu (5.5.19), przyjmuje postać zapisaną we wzorze:

$$\begin{aligned} P_d^t = & \left\{ \left\{ R + \sigma_R, v_r, \theta + \sigma_{\theta}, \phi + \sigma_{\phi}, \chi, \rho \right\}^{n, k} \in \mathbb{R}^6 : (n, k) \in [2, N_M + N_S + 1]_{\mathbb{N}} \times [1, f(m_n)]_{\mathbb{N}} \right. \\ & \wedge x_{est}^{n, k} \geq x_{th} \wedge (\nexists n' \in [2, N]_{\mathbb{N}} : n' \neq n \wedge \bar{R}^{n'} \leq \bar{R}^n \wedge \theta^{n, k} \in [\min(\theta)^{n'}, \max(\theta)^{n'}]) \\ & \left. \wedge \alpha^{n, k} \leq \frac{\pi}{2} \wedge R^{n, k} \in [0, R_{\max}] \wedge \theta^{n, k} \in [\theta_{\min}, \theta_{\max}] \wedge \phi^{n, k} \in [\phi_{\min}, \phi_{\max}] \right\}. \end{aligned} \quad (6.2.3)$$

Co więcej, również operacja wprowadzona w zmodyfikowanym modelu sensora, czyli filtr rozdzielczości, jest aplikowalna bez konieczności uwzględnienia jakichkolwiek zmian także dla elementów typu *OSI::StationaryObject*. Ostatecznie, definicję zbioru $\hat{P}_d^t$ ($\hat{P}_d^t \subset P_d^t$) zaprezentowaną początkowo w równaniu (6.1.1), zawierającą wszystkie warunki określone w oryginalnym modelu sensora oraz wprowadzony w tym rozdziale filtr rozdzielczości, można zapisać według wzoru:

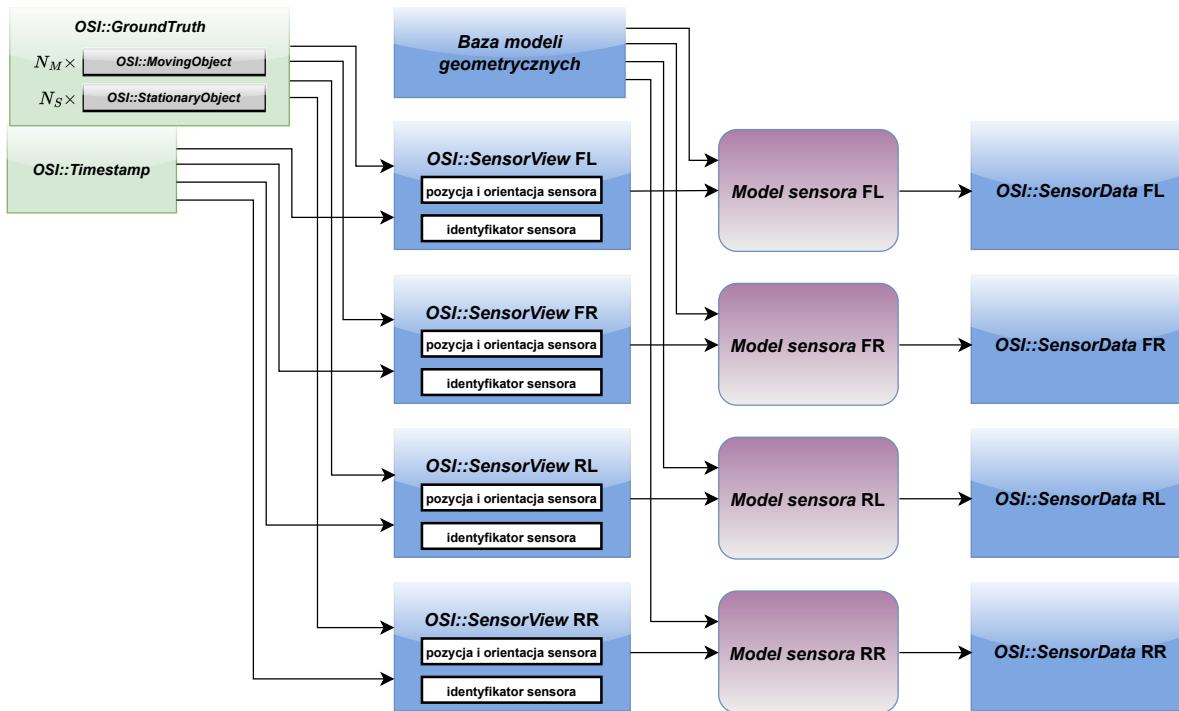
$$\hat{P}_d^t = \left\{ \{R + \sigma_R, v_r, \theta + \sigma_\theta, \phi + \sigma_\phi, \chi, \rho\}^{n,k} \in P_d^t : (n, k) \in [2, N_M + N_S + 1]_{\mathbb{N}} \times [1, f(m_n)]_{\mathbb{N}} \right. \\ \left. \wedge (\nexists n' \in [2, N]_{\mathbb{N}} : n' \neq n \wedge x_{est}^{n',k} \geq x_{est}^{n,k} \wedge |R^{n',k} - R^{n,k}| \leq \Delta R \wedge |\theta^{n',k} - \theta^{n,k}| \leq \Delta\theta) \right\}. \quad (6.2.4)$$

Przykładowy przepływ informacji w zmodyfikowanym modelu sensora - od uzyskania wiadomości *OSI::SensorView* o stemplu czasowym t , uwzględniającej obiekty klasy *OSI::MovingObject* oraz elementy typu *OSI::StationaryObject*, poprzez wywołanie wszystkich omówionych procedur, aż po wygenerowanie poprawnej struktury *OSI::SensorData* - zaprezentowano na rysunku 6.7, gdzie elementy zbioru $\hat{P}_d^t$ określa się na podstawie wzoru (6.2.4), natomiast $|\hat{P}_d^t|$ to liczba elementów zestawu $\hat{P}_d^t$.



Rysunek 6.7. Przykładowy przepływ informacji w zmodyfikowanym modelu symulacyjnym czujnika radarowego kompatybilnym ze standardem ASAM OSI, wspierającym generację zbioru sztucznych wykryć zarówno dla obiektów klasy *OSI::MovingObject*, jak i dla obiektów typu *OSI::StationaryObject*

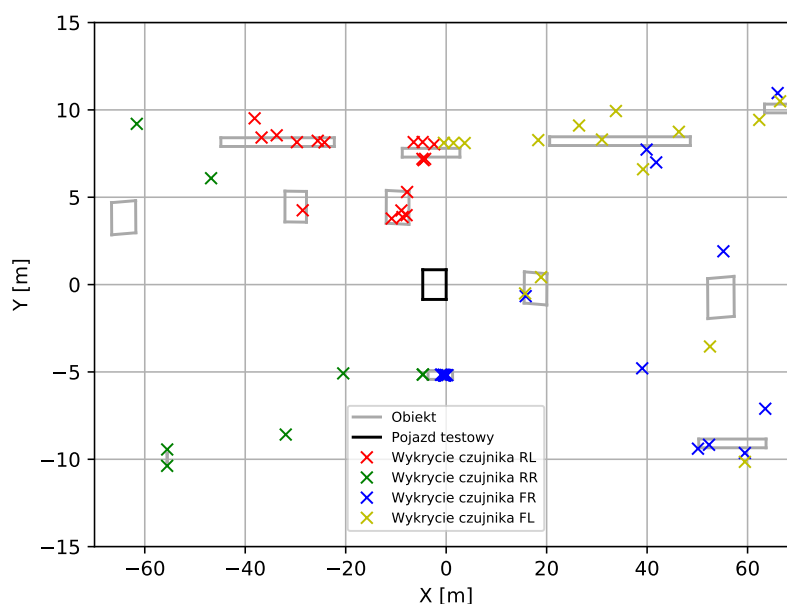
Z kolei przepływ danych dla konfiguracji wykorzystującej cztery instancje nowej wersji modelu sensora, wspierającej generację sztucznych pomiarów także dla statycznych elementów wirtualnego scenariusza, pokazano na rysunku 6.8, będącym modyfikacją rysunku 5.15.



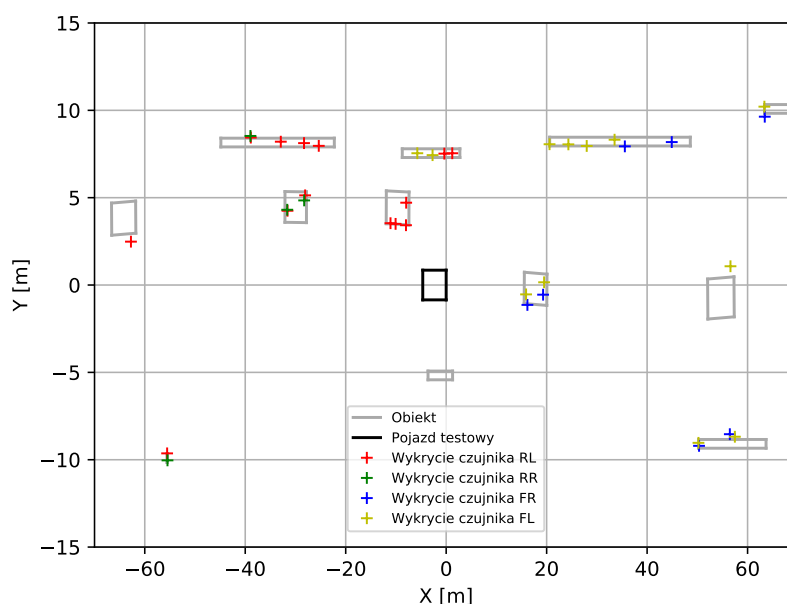
Rysunek 6.8. Przepływ danych przy użyciu czterech instancji zmodyfikowanego modelu sensora, wytwarzającego syntetyczne wykrycia również dla obiektów typu *OSI::StationaryObject*

Na rysunku 6.9 zaprezentowano dane z rzeczywistych czujników radarowych. Z kolei na rysunku 6.10 pokazano sztuczne pomiary radarowe wygenerowane przy użyciu czterech instancji zmodyfikowanego modelu sensora, a więc wyznaczone na podstawie formuły (6.2.4). Ponadto na obu rysunkach uwzględniono także elementy zbioru *OSI::GroundTruth*, przekazanego w pewnym dyskretnym momencie czasu do wszystkich wiadomości *OSI::SensorView*, zawierającego zarówno obiekty typu *OSI::MovingObject*, jak i obiekty typu *OSI::StationaryObject*. Warto zaznaczyć, że wszystkie informacje widoczne na dwóch wspomnianych rysunkach uzyskano dla tego samego dyskretnego momentu czasu co na rysunku 5.16a.

Porównując rysunki 6.9 i 6.10 można zauważyć, iż po wprowadzeniu opisanych wyżej modyfikacji zwiększył się poziom wiarygodności modelu sensora. Po pierwsze, dzięki zastosowaniu filtra rozdzielczości gęstość sztucznego zbioru danych zbliżyła się zdecydowanie do gęstości charakterystycznej dla rzeczywistych pomiarów radarowych. Po drugie, na rysunku 6.10 pojawiły się sztuczne wykrycia skojarzone z obiektami klasy *OSI::StationaryObject*, co jest niezwykle istotne, gdyż jak wyjaśniono w komentarzu do rysunku 5.16, pomiary powstałe na skutek odbicia od statycznych elementów scenariusza jazdy stanowią znaczną część zbioru rzeczywistych wykryć radarowych.



Rysunek 6.9. Rzeczywiste pomiary radarowe wraz ze zbiorem *OSI::GroundTruth* rozszerzonym o obiekty typu *OSI::StationaryObject*. Dane odnoszą się do pojedynczego cyklu pomiarowego pokazanego na rysunku 5.16a



Rysunek 6.10. Zbiory sztucznych wykryć radarowych uzyskane przy użyciu formuły (6.2.4) wraz ze zbiorem *OSI::GroundTruth* rozszerzonym o obiekty typu *OSI::StationaryObject*. Dane odnoszą się do pojedynczego cyklu pomiarowego pokazanego na rysunku 5.16a

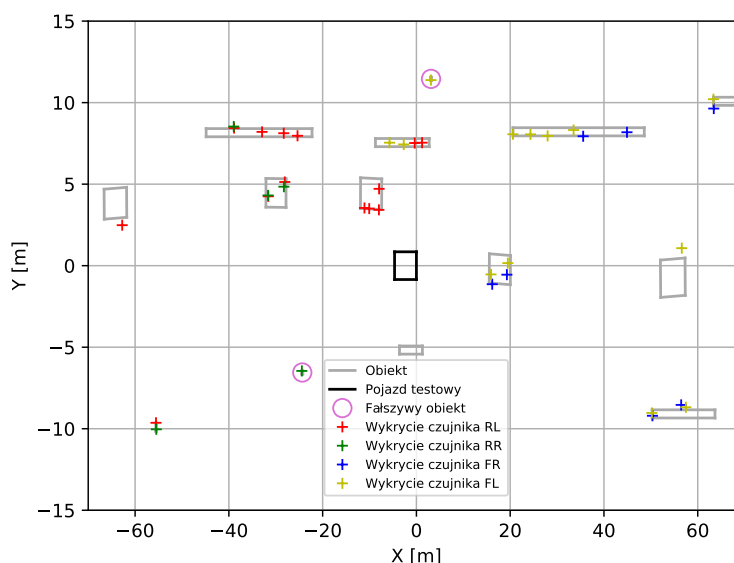
6.3 Generacja fałszywych wykryć

W rozdziale 4 wspomniano, że czujnik radarowy może zwracać tzw. fałszywe wykrycia spowodowane głównie przez wielokrotne załamanie się fali elektromagnetycznej w przestrzeni [12]. Oczywiście, w rzeczywistych systemach FMCW przeprowadza się adaptacyjny proces filtracji [58], którego celem jest pozbycie się błędnych odczytów. Niestety, tego typu procedura będzie zawsze pewnym kompromisem, gdyż zbyt rygorystyczny proces odsiewania może doprowadzić do usunięcia również poprawnych informacji. Podobną sytuację można zaobserwować w rzeczywistym zbiorze danych ukazanym na rysunku 6.9. Otóż gęstość zestawów wykryć zwróconych przez czujniki radarowe typu FMCW jest stosunkowo niewielka, co świadczy o obecności pewnej procedury filtracji. Jednak można także jednoznacznie zidentyfikować pewną niewielką liczbę pomiarów w żaden sposób nie powiązanych z obiektami obecnymi w scenariuszu. Są to właśnie fałszywe wykrycia, których aparatura klasy FMCW nie zdołała usunąć. Tego typu efekt także należy odwzorować w modelu sensora.

Postępowanie będzie podobne jak w przypadku generacji sztucznych pomiarów dla nieruchomych elementów scenariusza jazdy. Przyjmijmy, że każde fałszywe wykrycie można powiązać z pewnym nieistniejącym obiektem statycznym o stosunkowo niewielkich rozmiarach. Za taki fikcyjny cel uznajmy zatem obiekt typu *OSI::StationaryObject* reprezentowany przez sześciąt o boku długości 0.5 [m] (patrz rysunek 5.4), przy czym zakłada się, iż współrzędne x oraz y tych dodatkowych obiektów klasy *OSI::StationaryObject* są dwiema niezależnymi zmiennymi losowymi o rozkładzie normalnym, odpowiednio: $X \sim N(\mu_x, \sigma_x)$ oraz $Y \sim N(\mu_y, \sigma_y)$, gdzie:

- $\mu_x = \mu_y = 0$: wartość oczekiwana rozkładu normalnego,
- $\sigma_x = 50$ [m] : odchylenie standardowe dla pozycji wzdłużnej,
- $\sigma_y = 50$ [m]: odchylenie standardowe dla pozycji poprzecznej.

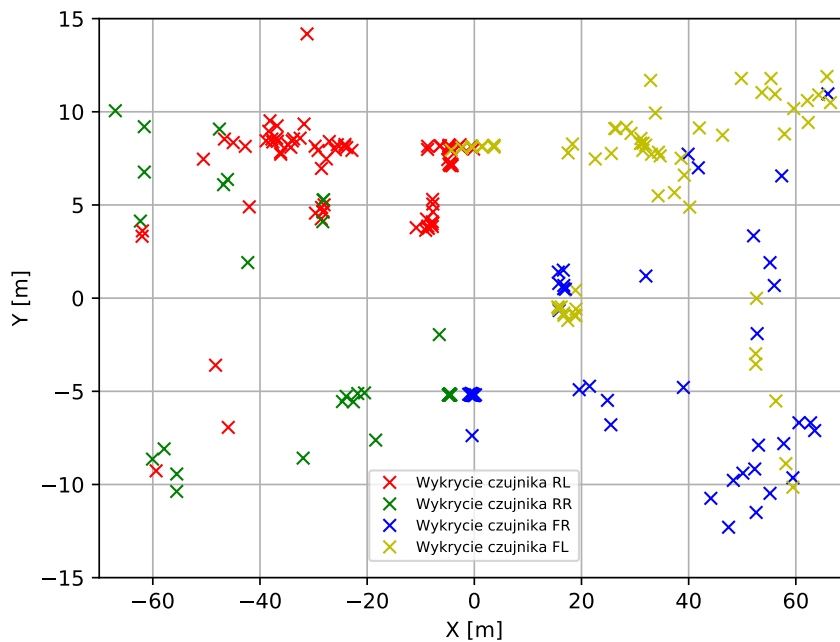
Wartości obu odchyłeń standardowych należy oczywiście traktować jako tymczasowe, gdyż mogą one ulec zmianie w trakcie przeprowadzenia procesu identyfikacji parametrów modelu. By uzyskać dane syntetyczne zawierające pewną liczbę fałszywych odczytów, wystarczy więc tylko rozszerzyć zbiór *OSI::GroundTruth*, zawarty w danej strukturze *OSI::SensorView*, o pewną liczbę tego typu wielościanów. Innymi słowy, zawartość wiadomości *OSI::SensorView* wymagana przez model sensora i pokazana na rysunku 6.6 powinna także uwzględniać elementy klasy *OSI::StationaryObject* wygenerowane pod kątem fałszywych wykryć. Co za tym idzie, przepływ informacji dla zmodyfikowanego modelu sensora (rysunek 6.7) oraz przepływ danych dla czterech instancji tegoż modelu (rysunek 6.8) również muszą zawierać fikcyjne obiekty typu *OSI::StationaryObject*. Rezultaty działania metody zaprezentowano na rysunku 6.11. Widać na nim, że do zbioru sztucznych wykryć zostało poprawnie dodane kilka fałszywych wykryć.



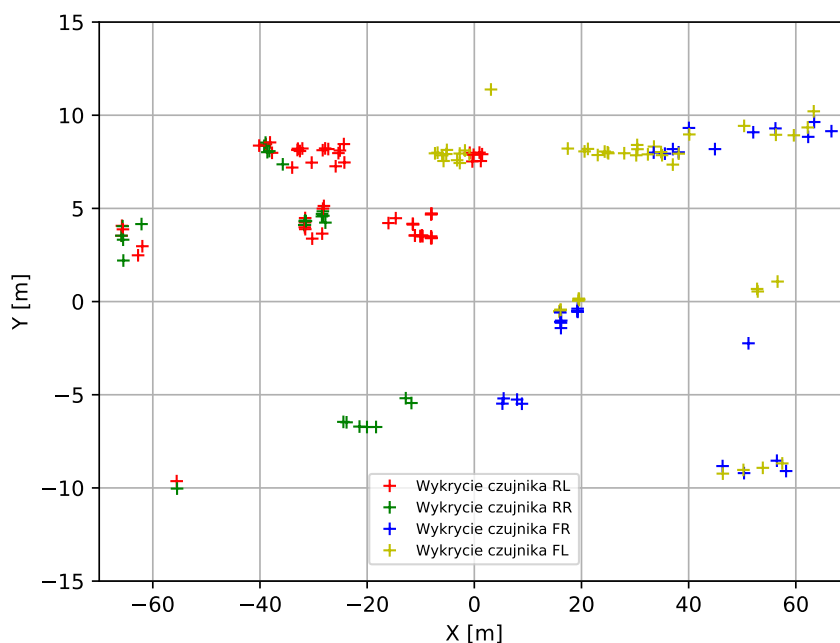
Rysunek 6.11. Zbiory sztucznych wykryć radarowych uzyskane przy użyciu formuły (6.2.4) wraz ze zbiorem *OSI::GroundTruth* rozszerzonym o obiekty typu *OSI::StationaryObject*, powiązane zarówno z rzeczywistymi elementami scenariusza jazdy, jak i z fikcyjnymi celami. Dane odnoszą się do pojedynczego cyklu pomiarowego pokazanego na rysunku 5.16a

6.4 Wstępna ocena wiarygodności modelu rozszerzonego

Ostatecznie, na rysunkach 6.12 i 6.13 zestawiono ze sobą rzeczywiste dane radarowe ze sztucznymi pomiarami wygenerowanymi przez cztery instancje modelu sensora uwzględniającego wszystkie opisane w tym rozdziale procedury, a więc: filtr rozdzielczości oraz generację syntetycznych wykryć zarówno dla obiektów stacjonarnych, jak również dla celów fikcyjnych. Warto zaznaczyć, że oba rysunki odnoszą się do tych samych pięciu kolejnych momentów czasu scenariusza jazdy, które zaprezentowano na rysunkach 5.17b i 5.16b, przy czym w tym przypadku, by ułatwić ocenę wizualną, zrezygnowano z prezentacji obiektów ze zbioru *OSI::GroundTruth* i pokazano jedynie wykrycia radarowe. Jak można zauważyć, po wprowadzeniu poprawek, gęstość pomiarowa modelu sensora jest porównywalna z gęstością charakterystyczną dla danych rzeczywistych. Ponadto, w zbiorach sztucznych wykryć pojawiły się także pomiary pochodzące od statycznych obiektów obecnych w scenariuszu jazdy (barierki wyznaczające krawędzie jezdni). Dodatkowo, w danych syntetycznych obecne są też fałszywe wykrycia niepowiązane w żaden sposób z rzeczywistymi elementami sceny. Podsumowując, po wprowadzeniu do modelu sensora modyfikacji opisanych w tym rozdziale, zwiększyła się znacznie wiarygodność sztucznego zbioru danych. Niemniej jednak, konieczne jest przeprowadzenie identyfikacji parametrów modelu oraz rzetelna ocena jego wiarygodności, co zostanie opisane w kolejnych rozdziałach pracy.



Rysunek 6.12. Dane z rzeczywistych czujników radarowych zebrane z pięciu kolejnych okresów czasu



Rysunek 6.13. Dane uzyskane na podstawie czterech instancji rozszerzonego modelu sensora, a więc uzyskane na podstawie równania (6.2.4), z uwzględnieniem wykryć dla obiektów fikcyjnych, zebrane z pięciu kolejnych okresów czasu

7 Dobór nastaw modelu

Aby model sensora był użyteczny w kontekście testów systemów typu ADAS w środowisku wirtualnej jazdy, należy rzetelnie ocenić wiarygodność procesu generacji zbioru sztucznych wykryć. W poprzednim rozdziale wprowadzono do oryginalnego modelu kilka modyfikacji zidentyfikowanych na bazie pomiarów z rzeczywistych czujników radarowych. Wstępna ocena wizualna pokazała wzrost wiarygodności modelu. Niemniej zanim przeprowadzona będzie rzetelna ocena działania wirtualnego czujnika, konieczne jest wykonanie procedury identyfikacji parametrów modelu. Model sensora posiada pewien zestaw modyfikowalnych parametrów, które wpływają na postać syntetycznych wykryć. Należy zatem dobrać jak najlepsze wartości dla tychże nastaw na podstawie przyjętego wskaźnika jakości.

7.1 Parametry modelu sensora

Nastawy modelu można podzielić na stałe, czyli związane z właściwościami danego systemu FMCW, oraz konfigurowalne, dzięki którym można do pewnego stopnia zmienić postać generowanych danych. Wartości, które nie ulegną zmianie w trakcie przeprowadzenia procedury identyfikacji parametrów modelu, zaczerpnięto ze specyfikacji rzeczywistego systemu radarowego. Do tej kategorii zalicza się parametry określające pole widzenia czujnika - $R_{\max}$, $\theta_{\min}$, $\theta_{\max}$, $\phi_{\min}$ i $\phi_{\max}$ - oraz parametry wymienione w tabeli 7.1, zdefiniowane w równaniu (2.3.11) (rozdział 2) oraz w równaniach (5.5.14a) i (5.5.14b) (rozdział 5):

Natomiast w tabeli 7.2 zaprezentowano wielkości, które będą modyfikowane w trakcie procesu optymalizacji. Są to: wzmocnienie testu statystycznego (F_n [dB]) i parametry filtra rozdzielczości oraz generatora fałszywych wykryć, czyli procedur zdefiniowanych w rozdziale 6; przy czym w operacji uzupełniającej zbiór sztucznych pomiarów o fikcyjne odczyty zdecydowano się także na modyfikację liczby fałszywych obiektów typu *OSI::StationaryObject*, powiązanych jednoznacznie z fikcyjnymi wykryciami, dodawanych w danym momencie czasu do zbioru *OSI::GroundTruth*. Warto zaznaczyć, że nie należy mylić zmiennych ΔR i $\Delta \theta$ z teoretyczną rozdzielczością danego radaru typu FMCW zdefiniowaną w rozdziale 2. Gęstość pomiarów zwróconych przez samochodowy czujnik radarowy w środowisku pełnym zakłóceń będzie stosunkowo ograniczona, co wynika z zastosowania progu odcięcia, którego celem jest pozbycie się fałszywych wykryć. Z tego względu rolą parametrów ΔR

Tabela 7.1. Niezmienne parametry modelu sensora

Parametr	Opis
P_t [W]	moc wyemitowanego impulsu elektromagnetycznego
G_t	wzmocnienie anteny nadawczej
G_r	wzmocnienie odebranego sygnału
λ [m]	długość fali radiowej
L_s	współczynnik strat systemowych
$k = 1.38 \times 10^{-23}$	stała Boltzmanna
$T_0 = 290$ [K]	temperatura bezwzględna
β [Hz]	pasmo przenoszenia odbiornika
ζ_R	dokładność pomiaru odległości
ζ_θ	dokładność pomiaru kąta
χ_0	referencyjna wartość SNR

oraz $\Delta\theta$ zdefiniowanych w modelu sensora w ramach filtra rozdzielczości jest odwzorowanie gęstości rzeczywistego zbioru danych.

Tabela 7.2. Konfigurowalne parametry modelu sensora

Parametr	Definicja
F_n [dB]	wzmocnienie testu statystycznego
ΔR [m]	rozdzielczość radialna
$\Delta\theta$ [°]	rozdzielczość kątowna
σ_x [m]	odchylenie standardowe generatora fałszywych wykryć dla pozycji wzdłużnej
σ_y [m]	odchylenie standardowe generatora zakłóceń dla pozycji poprzecznej
M	liczba fałszywych wykryć dla pojedynczego cyklu pomiarowego

7.2 Dane

Do wyliczenia wskaźnika jakości, koniecznego do przeprowadzenia procesu optymalizacji, wykorzystane zostaną zarówno rzeczywiste, jak i syntetyczne pomiary radarowe. Będą to te same dane, jakie opisano w rozdziale 5. Dostępne są zatem odczyty z czterech czujników radarowych krótkiego zasięgu typu FMCW, zebrane z krótkiej jazdy testowej po dwupasmowej drodze szybkiego ruchu, oraz sztuczne wykrycia wygenerowane przez cztery instancje modelu sensora; przy czym każdy model sensora jest jednoznacznie powiązany z rzeczywistym czujnikiem radarowym poprzez

identyczną konfigurację pozycji i orientacji względem układu LCS (patrz rysunek 5.14). Warto zaznaczyć, że syntetyczne pomiary oraz wykrycia radarowe zarejestrowane w trakcie jazdy testowej zostały odpowiednio zsynchronizowane. W tym celu dla każdego dyskretnego momentu czasu, dla którego uzyskano rzeczywisty zbiór wykryć, utworzona została unikalna wiadomość *OSI::SensorView*, zawierająca obiekty klasy *OSI::MovingObject* oraz obiekty typu *OSI::StationaryObject*, będąca wejściem dla wszystkich instancji modelu sensora (patrz rysunek 6.8). Innymi słowy, każdy wygenerowany zbiór *OSI::SensorView* może być jednoznacznie przypisany do rzeczywistego zestawu pomiarów poprzez wartość stempla czasowego.

Należy także wspomnieć o jedynym istotnym założeniu, jakie przyjęto w kontekście omawianej procedury identyfikacji parametrów modelu sensora. Otóż pomiary utworzone w danym, dyskretnym momencie czasu przez cztery czujniki radarowe, czy to rzeczywiste, czy to wirtualne - kolejno lewy przedni, prawy przedni, lewy tylny oraz prawy tylny - traktowane są jako pojedynczy zbiór danych (rzeczywisty lub syntetyczny). Przyjmując więc, że horyzont czasowy wskaźnika jakości wynosi pięć okresów czasu, to zestawy wykryć zaprezentowane na rysunku 6.12 analizuje się jako jeden pakiet danych rzeczywistych, natomiast zestawy wykryć pokazane na rysunku 6.13 rozpatrywane są jako jeden pakiet danych syntetycznych.

7.3 Wskaźnik jakości

Zadanie polega na porównaniu dwóch zbiorów wykryć - rzeczywistego z syntetycznym - gdzie każdy zestaw danych radarowych wykorzystywanych do wyliczenia wartości wskaźnika jakości to zlepek informacji pochodzących z czterech czujników radarowych (lub czterech modeli sensora) zarejestrowanych na przestrzeni kilku kolejnych okresów pomiarowych. Jak pokazano w obszernym raporcie [129], tematyka oceny dokładności modeli symulacyjnych czujników jest w dalszym ciągu intensywnie analizowana przez badaczy. Jako wskaźnik jakości, podobnie jak w [130], zostanie wykorzystana odległość Wassersteina [131]. Metryka ta rozwiązuje problem zagadnienia transportowego, gdyż określa, jak duża ilość pracy musi być włożona, by zbliżyć rozkład prawdopodobieństwa μ do innego rozkładu prawdopodobieństwa ν . Jest ona niezwykle użyteczna - została już z sukcesem zaaplikowana w wielu pracach naukowych, w szczególności w procesie uczenia generatywnych sieci przeciwnych (ang. Generative Adversarial Networks, GAN) [82] oraz do efektywnego szufladkowania danych [132, 133].

Zauważmy, że zbiory wykryć prezentowane były do tej pory na płaszczyźnie $\mathbb{R}^2$ (patrz rysunki 6.12 i 6.13). Oczywiście pominięcie wartości elewacji dostępnej w każdym pomiarze (czy to sztucznym, czy to rzeczywistym) jest pewnym uproszczeniem. Niemniej jednak, jak można zaobserwować w tabeli 7.2, kąt w płaszczyźnie wertykalnej nie jest brany pod uwagę w zbiorze konfigurowalnych parametrów modelu sensora. Dlatego też procedura doboru nastaw modelu także będzie rozważona na płaszczyźnie

$\mathbb{R}^2$. Przejdźmy teraz do definicji wskaźnika jakości. Dla każdych dwóch rozkładów prawdopodobieństwa μ i ν określonych na płaszczyźnie $\mathbb{R}^2$ odległość Wassersteina $W(\mu, \nu)$ stopnia p jest zdefiniowana następująco [131]:

$$W(\mu, \nu) = \left(\inf_{\gamma \in \Gamma(\mu, \nu)} \int_{\mathbb{R}^2 \times \mathbb{R}^2} d(x, y)^p d\gamma(x, y) \right)^{\frac{1}{p}} \quad (7.3.1)$$

gdzie d to metryka Euklidesowa, natomiast $\Gamma(\mu, \nu)$ jest zbiorem wszystkich miar prawdopodobieństwa zdefiniowanych na zbiorze $\mathbb{R}^2 \times \mathbb{R}^2$, dla których μ i ν to rozkłady brzegowe; innymi słowy dla każdego $A \subset \mathbb{R}^2$ zachodzi [131]:

$$\gamma(A \times \mathbb{R}^2) = \mu(A), \quad \gamma(\mathbb{R}^2 \times A) = \nu(A) \quad (7.3.2a-b)$$

Zauważmy, że wzór (7.3.1) odnosi się do rozkładów ciągłych. Zbiór wykryć radarowych można jednak traktować jak przestrzeń dyskretną, gdyż czujnik klasy FMCW charakteryzuje się pewną rozdzielczością pomiarową, co zdefiniowano w rozdziale 2 w równaniach (2.3.9a) - (2.3.9c). Zmodyfikujmy więc definicję zaprezentowaną we wzorze (7.3.1) tak, by była odpowiednia dla rozkładów dyskretnych. Teorię dla problemu analogicznego do zagadnienia aktualnie rozważanego, czyli do próby zestawienia ze sobą dwóch zbiorów wykryć, opisano rzetelnie w [133]. W publikacji tej przedstawiono rezultaty pracy nad porównywaniem złożonych scenariuszy jazdy zarejestrowanych pod kątem weryfikacji systemów klasy ADAS. Mianowicie, autorzy wspomnianego artykułu zdefiniowali dwuwymiarową, dyskretną siatkę, na którą nałożyli trajektorie wszystkich obiektów dostępnych w danym scenariuszu jazdy, przy czym jako trajektorie ustalonego celu należy rozumieć zbiór jego pozycji zapisanych w trakcie jazdy testowej. Następnie, dla takiej dyskretniej przestrzeni zdefiniowano metrykę Wassersteina. Uzyskane wyniki są jednoznacznym dowodem na skuteczność przyjętej metodologii.

Wykorzystajmy zatem podejście przyjęte przez twórców przywołanej publikacji. Wprowadźmy w pierw formułę (7.3.3) na sumaryczny zbiór wykryć $\mathbb{P}^t \subset \mathbb{R}^2$ wyrażony w kartezjańskim układzie współrzędnych. Zawierać on będzie wszystkie punkty na płaszczyźnie $\mathbb{R}^2$, dla których w momencie czasu t istnieje co najmniej jedno wykrycie zwrócone przez czujniki radarowe wykorzystane w omawianej procedurze - $P_d^{FL, t}$, $P_d^{FR, t}$, $P_d^{RL, t}$ oraz $P_d^{RR, t}$:

$$\mathbb{P}^t = \left\{ \{x, y\} \in \mathbb{R}^2 : x = R \cos \theta, y = R \sin \theta : \{R, \theta\} \in P_d^{FL, t} \cup P_d^{FR, t} \cup P_d^{RL, t} \cup P_d^{RR, t} \right\}. \quad (7.3.3)$$

Warto zaznaczyć, że dane zawarte w zestawie $\mathbb{P}^t$ mogą być powiązane albo z rzeczywistymi sensorami, albo z ich wirtualnymi odpowiednikami w postaci czterech instancji modelu sensora, przy czym w

przypadku informacji syntetycznych każdy z czterech zbiorów wykryć określony jest według równania (6.2.4).

Zdefiniujmy teraz dyskretną siatkę G_h określoną na płaszczyźnie $\mathbb{R}^2$. Komórkę siatki G_h o indeksie i oraz o rozmiarze h określa się według formuły [133]:

$$G_h(i) = \left\{ \{x, y\} \in \mathbb{R}^2 : \left\lfloor \frac{x}{h_x} \right\rfloor = i_1, \left\lfloor \frac{y}{h_y} \right\rfloor = i_2 \right\}, \quad (7.3.4)$$

gdzie $h = \{h_x, h_y\} \in \mathbb{R}^2$, $h_x > 0$, $h_y > 0$, $i = \{i_1, i_2\} \in \mathbb{Z}^2$, $\left\lfloor \frac{x}{h_x} \right\rfloor$ to największa liczba całkowita nie większa od ilorazu x/h_x , $\left\lfloor \frac{y}{h_y} \right\rfloor$ to największa liczba całkowita nie większa od ilorazu y/h_y , natomiast $\mathbb{Z}$ reprezentuje zbiór liczb całkowitych. Wykonajmy zatem rzut zbioru $\mathbb{P}^t$ na siatkę zdefiniowaną we wzorze (7.3.4). Przekształcony zestaw $\mathbb{P}_h^t$ uzyskuje się według takiej oto formuły:

$$\mathbb{P}_h^t = \left\{ i \in \mathbb{Z}^2 : \exists \{x, y\} \in \mathbb{P}^t : \{x, y\} \in G_h(i) \right\}. \quad (7.3.5)$$

Zauważmy, iż ilość elementów zbioru $\mathbb{P}_h^t$ jest zależna od wartości parametru h , który należy dobrać pod kątem konkretnej aplikacji. Niech $H^t : \mathbb{P}_h^t \rightarrow \mathbb{N}$ będzie tzw. histogramem zajętości określonym dla dyskretnego momentu czasu t , którego wartość dla elementu $i \in \mathbb{Z}^2$ wyznacza się na podstawie równania [133]:

$$H^t(i) = \sum_{P_d^t \in \mathfrak{P}^t} \left| \left\{ \{x, y\} \in \mathbb{R}^2 : x = R \cos \theta, y = R \sin \theta : \{R, \theta\} \in P_d^t \wedge \{x, y\} \in G_h(i) \right\} \right|, \quad (7.3.6)$$

gdzie $\mathfrak{P}^t = \{P_d^{FL,t}, P_d^{FR,t}, P_d^{RL,t}, P_d^{RR,t}\}$, natomiast $|\cdot|$ oznacza ilość elementów zbioru. Dla przestrzeni $\mathbb{P}_h^t$ sformułowanej w równaniu (7.3.5) można zdefiniować dyskretny rozkład prawdopodobieństwa $\mu : \mathbb{P}_h^t \rightarrow [0, 1]$ w taki oto sposób:

$$\mu(i) = \frac{H^t(i)}{\sum_{j \in \mathbb{P}_h^t} H^t(j)}, \quad i \in \mathbb{P}_h^t. \quad (7.3.7)$$

Wykorzystując wprowadzone wyżej definicje, można zmodyfikować wzór na odległość Wassersteina stopnia p tak, by był odpowiedni dla rozważanego dyskretnego problemu. Mianowicie, zadanie obliczenia metryki Wassersteina w przypadku dwóch rozkładów prawdopodobieństwa μ i ν określonych na podstawie równania (7.3.7) traktuje się jako zagadnienie z kategorii optymalnego transportu, które znajduje rozwiązanie poprzez sprowadzenie go do postaci problemu programowania liniowego. Dla każdego zbioru $\mathbb{P}_h^t$ można zdefiniować funkcję wzajemnie jednoznaczną $\eta : \mathbb{M} \rightarrow \mathbb{P}_h^t$, gdzie $\mathbb{M} = \{1, \dots, |\mathbb{P}_h^t|\}$. Oznaczmy $|\mathbb{P}_h^t|$ jako $\mathcal{M}$. Żeby obliczyć odległość Wassersteina $W(\mu, \nu)$, należy znaleźć funkcję $f^* : \mathbb{M} \times \mathbb{M} \rightarrow [0, 1]$, która minimalizuje sumę [133]:

$$\sum_{k=1}^{\mathcal{M}} \sum_{j=1}^{\mathcal{M}} f(k, j) d(k, j), \quad (7.3.8)$$

gdzie $d(k, j)$ to odległość Euklidesowa pomiędzy wektorami $\eta(k)$ oraz $\eta(j)$. Ograniczenia dla funkcji f^* są następujące [133]:

$$\begin{aligned} \forall k, j \in \mathbb{M} : 0 \leq f(k, j) \leq 1, \\ \forall k \in \mathbb{M} : \sum_{j=1}^{\mathcal{M}} f(k, j) &= \mu(\eta(k)), \\ \forall j \in \mathbb{M} : \sum_{k=1}^{\mathcal{M}} f(k, j) &= \nu(\eta(j)), \\ \sum_{k=1}^{\mathcal{M}} \sum_{j=1}^{\mathcal{M}} f(k, j) &= 1. \end{aligned} \quad (7.3.9a-d)$$

Do rozwiązania tak sformułowanego zadania programowania liniowego można wykorzystać algorytm opisany w [134] lub [135]. Ostatecznie, odległość Wassersteina $W(\mu, \nu)$ stopnia p oblicza się na podstawie równania [133]:

$$W(\mu, \nu) = \left(\sum_{k=1}^{\mathcal{M}} \sum_{j=1}^{\mathcal{M}} f^*(k, j) d(k, j) \right)^{\frac{1}{p}}. \quad (7.3.10)$$

7.4 Procedura optymalizacji

Przypomnijmy, że proces identyfikacji parametrów modelu sensora ma na celu znalezienie takich wartości dla nastaw wymienionych w tabeli 7.2, dla jakich model jest najbardziej wiarygodny w kontekście przyjętego wskaźnika jakości zaprezentowanego we wzorze (7.3.10). Warto zaznaczyć, że równanie (7.3.10) służy do wyznaczenia odległości Wassersteina dla pojedynczego cyklu pomiarowego, gdzie jako jeden pakiet danych rozumie się informacje zwrócone w pewnym momencie czasu przez cztery czujniki radarowe oraz przez ich wirtualne odpowiedniki w postaci czterech instancji modelu sensora skonfigurowanych przy użyciu ustalonych wartości parametrów określonych w tabeli 7.2.

Niemniej ocena jakości syntetycznego zbioru wykryć wygenerowanego przez model sensora z wykorzystaniem jednej ramki pomiarów rzeczywistych nie byłaby zbyt wiarygodna ze względu na stochastyczną naturę systemów typu FMCW. Dlatego też przyjęto, iż ostateczna wartość wskaźnika jakości (dla danych wartości nastaw) będzie średnią arytmetyczną obliczoną na podstawie 100 kolejnych okresów odczytu. Innymi słowy, wartość wskaźnika jakości $\mathbb{W}$ wykorzystywaną przez omawianą procedurę optymalizacji oblicza się według następującej formuły:

$$\mathbb{W} = \frac{1}{100} \sum_{i=1}^{100} W^i(\mu, \nu), \quad (7.4.1)$$

gdzie $W^i(\mu, \nu)$ jest odległością Wassersteina drugiego stopnia uzyskaną dla pakietu danych o indeksie i na podstawie równania (7.3.10). Jako algorytm optymalizacji wybrano metodę Powella [136], dzięki której można skutecznie wyznaczyć minimum nieliniowej funkcji wielu zmiennych, bez konieczności obliczania pochodnych [137]. Technika ta idealnie sprawdzi się zatem w tym przypadku, gdyż wskaźnik jakości wylicza się dla pewnego procesu stochastycznego, kontrolowanego przez parametry wymienione w tabeli 7.3.

7.5 Rezultaty

Algorytm Powella zwraca lokalne minimum dla zadanej funkcji. Rezultaty będą się zatem różnić w zależności od wyboru punktu startowego. Wyniki dla kilku różnych nastaw początkowych wraz z wartością wskaźnika jakości $\mathbb{W}$ zaprezentowano w tabeli 7.3. Warto zaznaczyć, że w każdym przypadku ukazanym we wspomnianej tabeli metoda Powella potrzebowała zaledwie kilkuset iteracji do obliczenia lokalnego minimum.

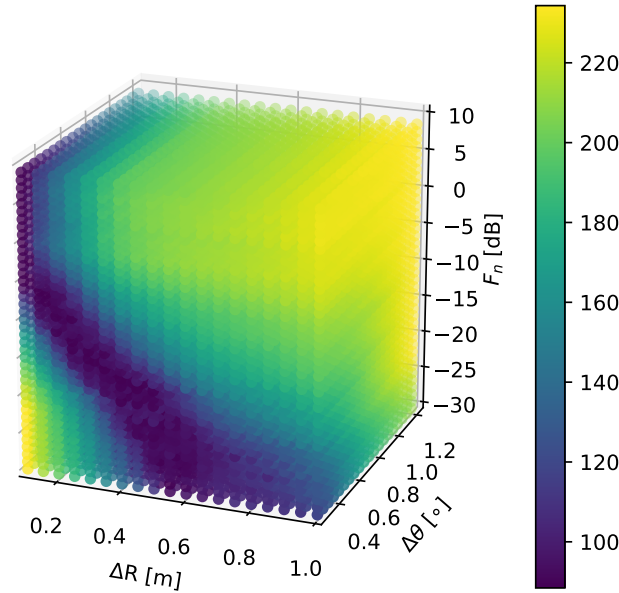
Tabela 7.3. Optymalne wartości nastaw modelu sensora zwrócone przez algorytm Powella dla różnych punktów startowych wraz z wartością wskaźnika jakości $\mathbb{W}$

$[\Delta R, \Delta\theta, F_n, \sigma_x, \sigma_y, M, \mathbb{W}]$		
	Wartości początkowe	Wartości końcowe
1.	[0.8, 1.0, -12.0, 50.0, 50.0, 6, 229.07]	[0.23, 0.46, -20.79, 53.81, 53.82, 8, 89.56]
2.	[0.8, 1.0, 5.0, 60.0, 60.0, 10, 292.01]	[0.18, 0.37, -10.02, 56.65, 56.26, 8, 102.63]
3.	[0.1, 0.25, 5.0, 60.0, 60.0, 10, 189.67]	[0.31, 0.33, -15.84, 51.25, 57.55, 9, 85.86]
4.	[0.1, 0.25, -30.0, 60.0, 60.0, 10, 229.11]	[0.58, 0.34, -26.11, 56.89, 55.42, 7, 94.58]

Jak widać w tabeli 7.3, najniższy poziom wskaźnika $\mathbb{W}$ uzyskano w eksperymencie numer trzy. Warto zadać tutaj pytanie, jak kształtuje się wartość przyjętej metryki dla zmieniających się parametrów modelu sensora. Z tabeli 7.3 z kolumny z nastawami początkowymi można wywnioskować, że zarówno zbyt wysokie, jak i zbyt niskie wartości przypisane do parametrów ΔR , $\Delta\theta$ oraz F_n negatywnie wpływają na liczbę $\mathbb{W}$. Parametry modelu sensora podzielono zatem na dwie kategorie. Do pierwszej kategorii zaliczono ustawienia generatora fałszywych wykryć: σ_x , σ_y oraz M . Wartości tych atrybutów odpowiadają rozwiązaniu uzyskanemu dla eksperymentu numer trzy, co pokazano w tabeli 7.3. Do drugiej kategorii zaliczono ustawienia filtra rozdzielczości oraz wzmocnienie testu statystycznego: ΔR , $\Delta\theta$ i F_n . Następnie, dla parametrów ΔR , $\Delta\theta$ oraz F_n wygenerowano trójwymiarową, dyskretną siatkę wartości - $\mathbb{I} \times \mathbb{J} \times \mathbb{K} \subset \mathbb{R}^3$, gdzie:

$$\begin{aligned}
\mathbb{I} &= \left\{ \Delta R \in \mathbb{R} : 0.1 \leq \Delta R \leq 1.0 \wedge \frac{\Delta R}{0.05} \in \mathbb{Z} \right\}, \\
\mathbb{J} &= \left\{ \Delta \theta \in \mathbb{R} : 0.25 \leq \Delta \theta \leq 1.25 \wedge \frac{\Delta \theta}{0.05} \in \mathbb{Z} \right\}, \\
\mathbb{K} &= \left\{ F_n \in \mathbb{Z} : -30 \leq F_n \leq 10 \right\}.
\end{aligned} \tag{7.5.1a-c}$$

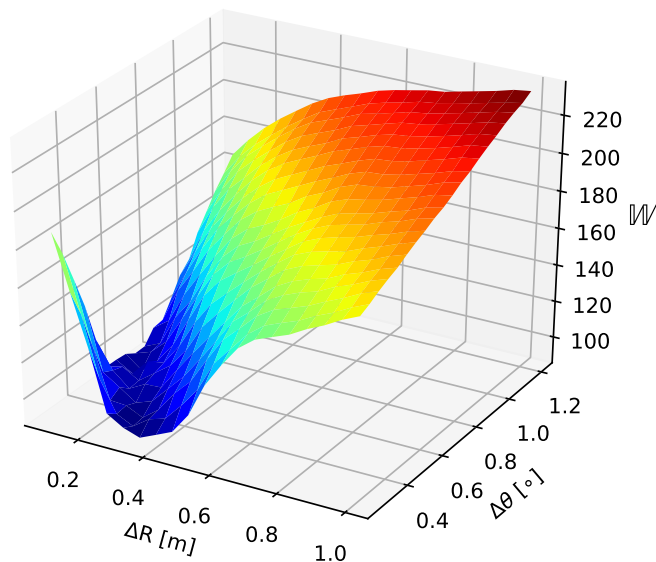
W końcu, powtórzono wspomniany wyżej proces optymalizacji, czyli uzyskanie liczby $\mathbb{W}$ na podstawie 100 kolejnych okresów odczytu, dla każdego $\{\Delta R, \Delta \theta, F_n\} \in \mathbb{I} \times \mathbb{J} \times \mathbb{K}$. Na rysunku 7.1 zaprezentowano, jak kształtuje się wartość przyjętego wskaźnika jakości dla zmieniających się nastaw ΔR , $\Delta \theta$ i F_n .



Rysunek 7.1. Zmienność wskaźnika jakości dla siatki wartości ΔR , $\Delta \theta$ i F_n zdefiniowanej w równaniach (7.5.1a) - (7.5.1c). Pasek koloru widoczny na wykresie reprezentuje wszystkie wartości liczby $\mathbb{W}$, jakie uzyskano dla zbioru $\mathbb{I} \times \mathbb{J} \times \mathbb{K}$

Zgodnie z oczekiwaniami odległość Wassersteina rośnie wraz ze wzrostem parametru F_n . Dzieje się tak dlatego, że im wyższa wartość wzmocnienia testu statystycznego, tym więcej odrzuconych pomiarów, przez co liczebność sztucznego zbioru wykryć może być zbyt mała w stosunku do danych zwróconych przez rzeczywisty czujnik. Dodatkowo widać, że wskaźnik jakości narasta także dla ΔR i $\Delta \theta$ dążących do zera, gdzie punkty pomiarowe oszacowane przez model sensora są zbyt gęsto rozmieszczone w przestrzeni. Można także zauważyć, że zbiór $\mathbb{I} \times \mathbb{J} \times \mathbb{K}$ zawiera sporą liczbę kombinacji nastaw ΔR , $\Delta \theta$ oraz F_n , dla których wskaźnik jakości osiąga wartość mniejszą od największej końcowej wartości

liczby $\mathbb{W}$ zaprezentowanej w tabeli 7.3. Potwierdza to wspomniany wcześniej fakt, że rozwiązanie zaproponowane przez algorytm Powella jest jedynie minimum lokalnym zadanej funkcji i będzie się ono różnić w zależności od wyboru punktu początkowego. Dodatkowo na rysunku 7.2 pokazano przekrój poprzeczny sześcianu uwidocznionego na rysunku 7.1. W szczególności parametr F_n ustawiono zgodnie z optymalną wartością zaprezentowaną w tabeli 7.3 dla eksperymentu numer trzy, natomiast zakres dla nastaw ΔR i $\Delta\theta$ pozostał bez zmian; zatem oś pionowa to wartość wskaźnika jakości wyliczonego zgodnie ze wzorem (7.4.1). Widać więc wyraźnie, że metoda Powella poradziła sobie skutecznie ze znalezieniem minimum lokalnego dla zadanej funkcji.



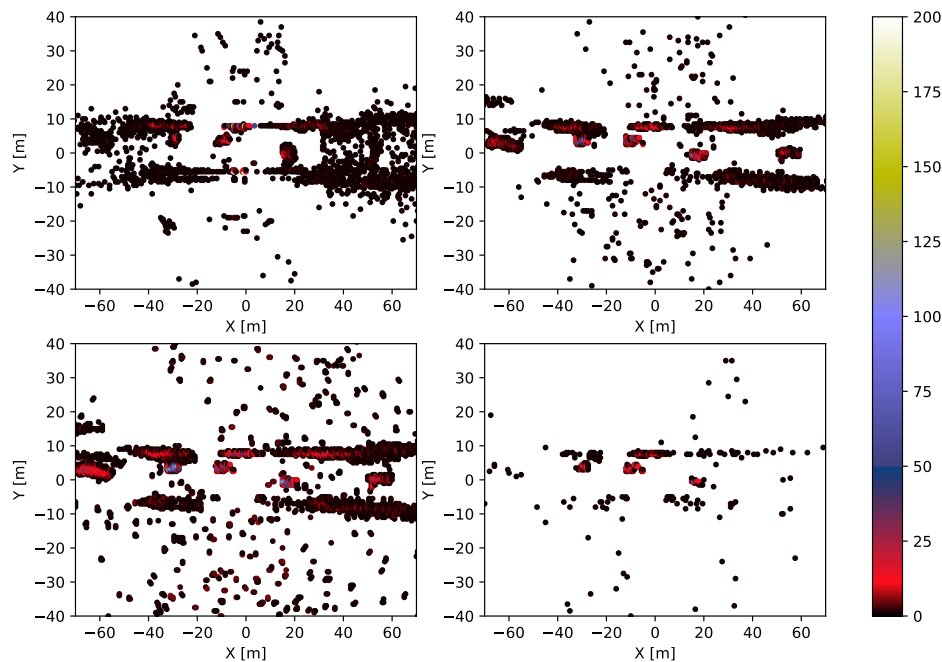
Rysunek 7.2. Zmienność wskaźnika jakości dla optymalnej wartości F_n oraz dla przyjętej siatki ΔR i $\Delta\theta$. Jest to zatem przekrój poprzeczny sześcianu pokazanego na rysunku 7.1. Oś pionowa to w tym przypadku odległość Wassersteina wyliczona dla pojedynczego cyklu optymalizacji, czyli zgodnie ze wzorem (7.4.1)

Dodatkowo na rysunku 7.3 zestawiono ze sobą cztery różne zbiory wykryć, z których każdy zawiera dane uzyskane z pełnego cyklu optymalizacji, czyli 100 kolejnych okresów odczytu. Na rysunku 7.3 mamy zatem takie oto typy informacji:

- rzeczywisty zbiór wykryć (lewy górny róg),
- zbiór wykryć wygenerowany przez model sensora skonfigurowany przy użyciu nastaw optymalnych (minimum lokalne zwrócone przez algorytm Powella) zapisanych w tabeli 7.3 dla eksperymentu numer trzy (prawy górny róg),

- zbiór wykryć wygenerowany przez model sensora dla nastaw odpowiadających odpowiednio: dolnemu zakresowi ograniczeń zbioru $\mathbb{I} \times \mathbb{J} \times \mathbb{K}$, czyli wysokiej rozdzielczości połączonej z bardzo niskim wzmocnieniem testu statystycznego (lewy dolny róg), oraz górnemu zakresowi ograniczeń zbioru $\mathbb{I} \times \mathbb{J} \times \mathbb{K}$, a więc niskiej rozdzielczości zestawionej ze sporym wzmocnieniem testu statystycznego (prawy dolny róg).

Warto zaznaczyć, że rysunek 7.3 wygenerowano w formie tak zwanej mapy ciepła. Mianowicie, we wszystkich czterech wymienionych wyżej przypadkach pomiary rzutowano na dwuwymiarową siatkę o komórkach długości 0.5 [m] i szerokości 0.5 [m]. Następnie każdemu elementowi siatki przypisany został kolor, na podstawie liczby wykryć, jaką zakwalifikowano do tego elementu; przy czym zależność barwy od liczby pomiarów ukazano na pasku kolorów widocznym po prawej stronie rysunku. Oczywiście należy tutaj wspomnieć, iż obecność punktu na omawianym rysunku oznacza, że w danej komórce znalazło się co najmniej jedno wykrycie.



Rysunek 7.3. Mapy ciepła wygenerowane dla czterech typów danych zebranych ze 100 kolejnych okresów odczytu - rzeczywisty zbiór wykryć (lewy górny róg), sztuczny zbiór wykryć uzyskany dla nastaw zwróconych przez algorytm Powella (prawy górny róg) oraz syntetyczny zestaw pomiarów uzyskany dla nastaw odpowiadających kolejno dolnemu i górnemu zakresowi ograniczeń zbioru $\mathbb{I} \times \mathbb{J} \times \mathbb{K}$ (odpowiednio lewy dolny róg oraz prawy dolny róg)

Na rysunku 7.3 widać wyraźnie, że wyniki są zadowalające jedynie dla przypadku pokazanego w prawym górnym rogu, gdzie model sensora skonfigurowano przy użyciu nastaw zwróconych przez algorytm Powella dla eksperymentu numer trzy (patrz tabela 7.3). Po pierwsze, gęstość rozmieszczenia punktów jest podobna w stosunku do danych rzeczywistych. Po drugie, można także zaobserwować, że model sensora oraz prawdziwy czujnik radarowy wykrywają te same obiekty - kontury pojazdów zarysowane po przeprowadzeniu pełnego cyklu optymalizacji są odwzorowane w porównywalny sposób. Inaczej ma się sytuacja w pozostałych dwóch przypadkach. Ustawienie wysokiej rozdzielczości pomiarowej oraz bardzo niskiego wzmocnienia testu statystycznego (dolny zakres ograniczeń zbioru $\mathbb{I} \times \mathbb{J} \times \mathbb{K}$) powoduje, że model sensora zwraca zbyt dużą ilość sztucznych pomiarów. Z kolei dla niskiej rozdzielczości ze sporym wzmocnieniem testu statystycznego (górny zakres ograniczeń zbioru $\mathbb{I} \times \mathbb{J} \times \mathbb{K}$) model sensora usuwa zdecydowanie zbyt wiele syntetycznych wykryć z wygenerowanego zbioru danych.

Podsumowując, zaprezentowane wyżej rezultaty jednoznacznie świadczą o tym, iż omówiona w tym rozdziale procedura identyfikacji parametrów modelu znacząco przyczyniła się do poprawy jakości (rzetelności) sztucznego zbioru wykryć radarowych. W kolejnym rozdziale opisana zostanie metoda oceny wiarygodności procesu generacji zestawu syntetycznych pomiarów. Wykonanie tejże procedury z wykorzystaniem zidentyfikowanego w tym rozdziale modelu sensora będzie konieczne do zweryfikowania użyteczności modelu w kontekście testów systemów klasy ADAS w środowisku wirtualnej jazdy.

8 Eksperymenty weryfikacyjne

Teza pracy wspomina, że sztuczne wykrycia wygenerowane na podstawie modelu matematycznego czujnika radarowego bazującego na koncepcji centrów rozpraszania mogą być użyteczne w kontekście testów systemów klasy ADAS w środowiskach wirtualnej jazdy. Innymi słowy, model sensora powinien być wiarygodny oraz na tyle wydajny, by syntetyczne dane były generowane w czasie rzeczywistym, niezależnie od poziomu skomplikowania wirtualnego scenariusza. W poprzednim rozdziale przeprowadzono procedurę identyfikacji parametrów modelu z zamiarem dobrania jak najlepszych wartości dla nastaw modelu w kontekście przyjętego wskaźnika jakości. Analiza wizualna zaprezentowanych wyników pozwala stwierdzić, iż ustawienia wygenerowane przez algorytm Powella znacząco poprawiły rzetelność sztucznych pomiarów. W tym rozdziale przedstawione zostaną wyniki eksperymentów, których celem jest udowodnienie tezy postawionej na początku pracy. Mianowicie, model sensora będzie oceniony pod kątem wiarygodności i wydajności.

8.1 Ocena wydajności

8.1.1 Informacja na temat kodu źródłowego

Jak wspomniano w rozdziale 6 i zaprezentowano na rysunku 6.8, wejściem do pojedynczej instancji zmodyfikowanego modelu sensora jest struktura *OSI::GroundTruth*, zawarta w wiadomości *OSI::SensorView* o stemplu czasowym t , oraz baza modeli geometrycznych, natomiast wyjście z modelu to zestaw sztucznych wykryć zapisany w zbiorze *OSI::SensorData*. Standard ASAM OSI można w prosty sposób zintegrować z takimi językami, jak C++ czy Python. W tym przypadku zdecydowano, że kod źródłowy modelu sensora będzie oparty na składni języka C++.

Przywołajmy tutaj fakt, iż prawie wszystkie procedury opisane w rozdziałach 5 oraz 6, począwszy od generacji zbioru centrów rozpraszania $P_{sc}^{SCS, t}$, aż po uzyskanie zestawu syntetycznych pomiarów $\hat{P}_d^t$ - patrz równania (6.2.1) i (6.2.4) - zdefiniowano tak, iż możliwe jest przeprowadzenie wszelkich operacji algebraicznych w sposób równoległy (tak zwana wektoryzacja obliczeń). Wyjątkiem są tutaj procedury, w których przekształcenia na sztucznym zbiorze wykryć nie są wykonywane niezależnie dla każdego pomiaru. Są to: algorytm obsługi przysłonięć oraz filtr rozdzielczości.

Niemniej, jak napisano wyżej, znaczną część operacji wykonuje się w sposób równoległy. Biorąc natomiast pod uwagę to, że wydajność modelu jest kluczowa w kontekście osiągnięcia właściwości czasu rzeczywistego niezależnie od poziomu skomplikowania wirtualnego scenariusza, należy stwierdzić, iż zastosowanie wektoryzacji obliczeń będzie tutaj konieczne. Dlatego też wszystkie najważniejsze procedury opisane w rozdziałach 5 oraz 6 oparto na funkcjonalnościach niezwykle popularnej i rozbudowanej biblioteki Eigen [138], gdzie temat wektoryzacji zrealizowano w bardzo skuteczny sposób.

Warto także wspomnieć o tym, jak przetwarzane są elementy bazy modeli geometrycznych. Postanowiono, iż obiekty wchodzące w skład tejże bazy zostaną zapisane w formacie *OBJ* zaprojektowanym przez firmę Wavefront [139], który służy właśnie do opisu trójwymiarowych obiektów reprezentowanych przez zbiór połączonych ze sobą trójkątów. Jako bibliotekę języka C++ realizującą funkcjonalność wczytania danego zestawu obiektów wybrano pakiet *tinyobjloader*, czyli bardzo wydajne oprogramowanie umożliwiające przetworzenie ogromnych modeli geometrycznych składających się nawet z kilku milionów trójkątów [140].

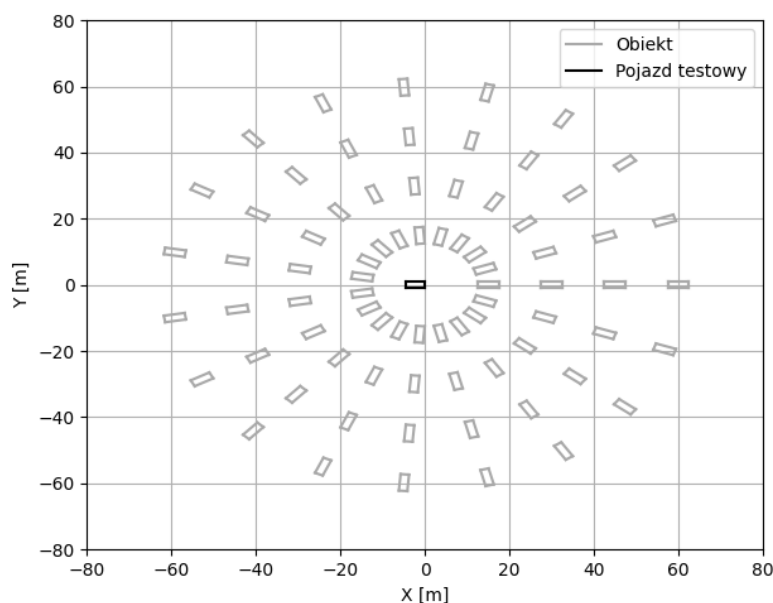
8.1.2 Procedura oceny wydajności modelu

Powiedzmy, że układ typu ADAS testowany w symulatorze jazdy taktowany jest z częstotliwością wynoszącą 40 [Hz]. Przyjmijmy też, że urządzeniem, na którym przeprowadza się procedurę wirtualnej walidacji, jest komputer klasy PC. Dodatkowo założmy, iż wszystkie operacje przeprowadzone są jedynie przy użyciu dostępnych jednostek procesora (ang. central processing unit, CPU). Oczywiście w razie potrzeby, gdyby złożoność obliczeniowa modelu uniemożliwiała wykonanie wszystkich obliczeń w trakcie trwania pojedynczego cyklu symulacji, można przeprowadzić część (lub większość) operacji za pomocą procesora graficznego (ang. graphical processing unit, GPU). Jednak biorąc pod uwagę wysoką wydajność pakietu Eigen [138], ograniczenie się jedynie do jednostek CPU powinno być wystarczające do spełnienia wspomnianego warunku czasu rzeczywistego. W pojedynczym cyklu symulacji ADAS (trwającym 25 [ms]) należy w sposób sekwencyjny wykonać trzy następujące procesy:

- (i) Wygenerowanie czterech zbiorów sztucznych wykryć radarowych przy użyciu czterech instancji modelu sensora z dostarczoną na wejściu unikalną wiadomością *OSI::SensorView*, czyli zgodnie z procedurą zaprezentowaną w graficzny sposób na rysunku 6.8.
- (ii) Oszacowanie tak zwanych brył brzegowych (prostopadłościanów) dla pojazdów obecnych w scenie z wykorzystaniem algorytmu wykrywania obiektów (patrz rysunek 2.7), bezpośrednio na podstawie pomiarów zwróconych przez wspomniane wirtualne czujniki radarowe.
- (iii) Wywołanie docelowej funkcji typu ACC lub AEBS przy użyciu uzyskanych danych syntetycznych według metodologii opisanej w rozdziale 2 i ukazanej na rysunku 2.10.

Wynika z tego, że czas generacji wszystkich czterech zbiorów sztucznych wykryć powinien być wyraźnie krótszy od zadanego okresu 25 [ms], gdyż należy pozostawić pewien bufor potrzebny

na wykonanie pozostałych dwóch procesów. By ocenić wydajność modelu sensora, zdecydowano się zaprojektować następujący scenariusz wirtualnej jazdy. Składa się on z pewnej liczby sztucznie wygenerowanych, identycznych obiektów klasy *OSI::MovingObject* zapisanych w wiadomości *OSI::GroundTruth*, będącej częścią struktury *OSI::SensorView*, zgodnie ze schematem poglądowym pokazanym na rysunku 5.1. Bryły podzielono na dwudziestoelementowe grupy, gdzie elementy pojedynczego zestawu rozmieszczono równomiernie na okręgu o zadanej średnicy. Postanowiono przeprowadzić eksperyment wydajności dla czterech poziomów złożoności scenariusza: 20, 40, 60 oraz 80 pojazdów. Spełnienie postawionego warunku czasu rzeczywistego dla scenariusza składającego się z 80 pojazdów będzie dowodem pierwszej części tezy, gdyż dla stosunkowo niewielkiego otoczenia dookoła pojazdu testowego trudno byłoby przekroczyć taką liczbę obiektów nawet dla bardzo zatłoczonej drogi szybkiego ruchu. Na rysunku 8.1 zaprezentowano zbiór *OSI::GroundTruth* właśnie dla najbardziej wymagającej konfiguracji.



Rysunek 8.1. Wirtualny scenariusz zaprojektowany na potrzeby oceny wydajności modelu sensora

Oczywiście, wskaźnikiem wydajności jest tutaj czas potrzebny na wykonanie pojedynczego zbioru operacji, czyli wygenerowania czterech zestawów sztucznych wykryć. Dodatkowo, by zwiększyć wiarygodność uzyskanych rezultatów opisywane doświadczenie wykonano dla dwóch różnych konfiguracji sprzętowych: komputer klasy PC wyposażony w procesor Intel Core i7-9750H oraz komputer klasy PC bazujący na CPU Intel Core i5-7200U.

8.1.3 Rezultaty

W tabelach 8.1 (procesor Intel Core i7-9750H) oraz 8.2 (procesor Intel Core i5-7200U) zaprezentowano kolejno minimalny, maksymalny oraz średni czas cyklu; wartości czasów dla każdej konfiguracji scenariusza wyliczono na podstawie 1000 powtórzeń wspomnianej wyżej procedury generacji danych. Warto zaznaczyć, że w przypadku 80 obiektów każda instancja modelu sensora musi przetworzyć 11040 centrów rozpraszania, z których zbudowane są modele geometryczne obiektów, co w sumie daje zbiór składający się z 44160 elementów. Pomimo tak licznej chmury punktów do przetworzenia czas generacji syntetycznych danych radarowych nie przekroczył 6 [ms] dla CPU Intel Core i7-9750H oraz 14 [ms] dla CPU Intel Core i5-7200U. Pozostawiony został zatem spory bufor dla pozostałych dwóch procesów. Można więc przyjąć, że warunek czasu rzeczywistego został spełniony, co dowodzi pierwszej części postawionej tezy. Warto tutaj także podkreślić niezwykle stabilność oprogramowania modelu sensora. Przede wszystkim uzyskane czasy charakteryzują się stosunkowo niewielkim odchyleniem standardowym. Ponadto opóźnienie wzrasta niemal liniowo dla zwiększającej się liczby obiektów w scenariuszu.

Tabela 8.1. Ocena wydajności modelu sensora na procesorze Intel Core i7-9750H

Liczba obiektów	Czas minimalny	Czas maksymalny	Czas średni
20	2.3 [ms]	2.4 [ms]	2.3 [ms]
40	3.4 [ms]	3.6 [ms]	3.5 [ms]
60	4.5 [ms]	4.8 [ms]	4.6 [ms]
80	5.6 [ms]	6.0 [ms]	5.7 [ms]

Tabela 8.2. Ocena wydajności modelu sensora na procesorze Intel Core i5-7200U

Liczba obiektów	Czas minimalny	Czas maksymalny	Czas średni
20	4.6 [ms]	4.7 [ms]	4.6 [ms]
40	7.6 [ms]	7.8 [ms]	7.7 [ms]
60	10.3 [ms]	10.6 [ms]	10.4 [ms]
80	13.2 [ms]	14.0 [ms]	13.5 [ms]

8.2 Ocena wiarygodności

Do zweryfikowania wiarygodności modelu sensora będzie wykorzystana metoda zaproponowana przez autora tejże pracy, gdzie sformułowanie problemu zostało zaprezentowane w [13], natomiast dokładny opis wraz z eksperymentami opublikowano w [130]. Stwierdza się zatem, że dokładność

modelu będzie akceptowalna wtedy, gdy lista brył brzegowych wygenerowana przez algorytm wykrywania obiektów na podstawie sztucznych wykryć jest porównywalna z rzeczywistym zbiorem brył brzegowych uzyskanym dla danego momentu czasu scenariusza wirtualnej jazdy. Na potrzeby tego zestawienia należy oczywiście zdefiniować odpowiedni wskaźnik jakości. Przypomnijmy, iż do spełnienia warunku wiarygodności konieczne jest, by wskaźnik jakości mieścił się w zadanym przedziale tolerancji niezależnie od poziomu skomplikowania wirtualnego scenariusza. Zauważmy, że wymóg ten jest spełniony przez to, iż model sensora został zintegrowany ze standardem ASAM OSI. Innymi słowy, rzetelny zbiór syntetycznych pomiarów będzie wygenerowany niezależnie od poziomu rozbudowania scenariusza jazdy, pod warunkiem dostarczenia na wejściu modelu poprawnej wiadomości *OSI::SensorView*, czyli zgodnie ze schematem poglądowym zaprezentowanym na rysunku 6.6.

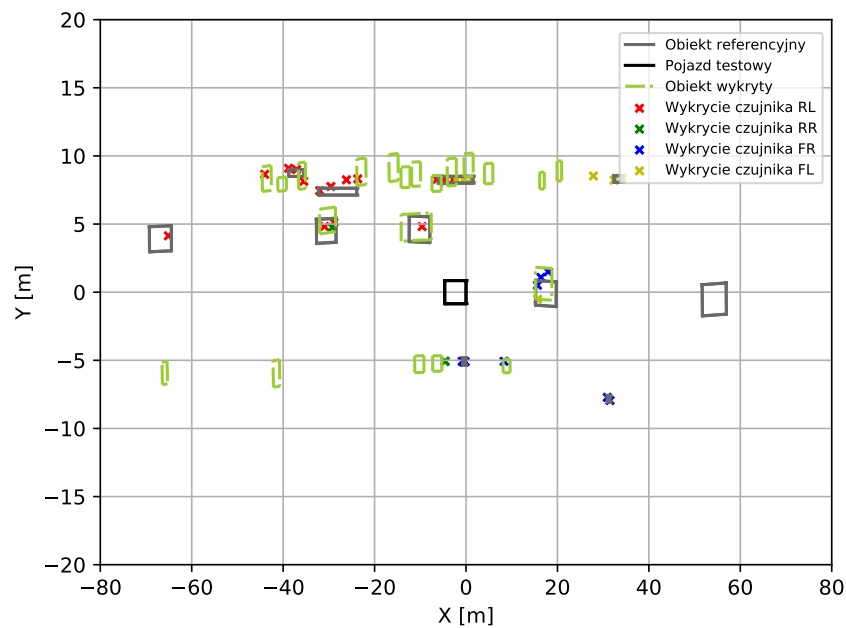
8.2.1 Dane wejściowe

By wyliczyć wskaźnik jakości, konieczne będzie zestawienie ze sobą danych syntetycznych z odpowiadającymi im informacjami rzeczywistymi. Pomocne okażą się tutaj pomiary z czujników radarowych oraz z sensora referencyjnego zarejestrowane w trakcie krótkiej jazdy testowej po dwupasmowej drodze szybkiego ruchu; pomiary te omówiono w rozdziale 5. Tym razem, podobnie jak w [130], zdecydowano się skorzystać z 500 ramek danych. Przypomnijmy, że dla każdego dyskretnego momentu czasu dostępne są zbiory wykryć z czterech radarów krótkiego zasięgu typu FMCW, których sposób montażu na samochodzie testowym zaprezentowano na rysunku 5.14 z wykorzystaniem dwuliterowych skrótów, odpowiednio - lewy przedni (FL), prawy przedni (FR), lewy tylny (RL) oraz prawy tylny (RR). Dodatkowo, dla wszystkich dyskretnych chwil czasu wygenerowana została wiadomość *OSI::GroundTruth*, na podstawie danych pochodzących z sensora referencyjnego. Warto zaznaczyć, że struktura *OSI::GroundTruth* jest zgodna ze schematem poglądowym ukazanym na rysunku 6.6, a więc stanowi część wiadomości *OSI::SensorView* oraz zawiera zarówno obiekty dynamiczne (*OSI::MovingObject*), jak i statyczne (*OSI::StationaryObject*).

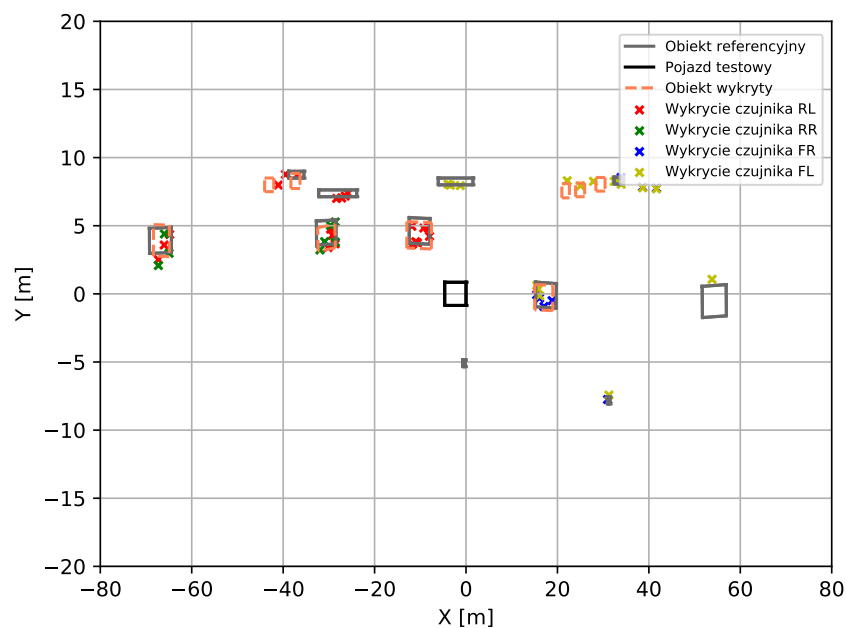
Załóżmy, że procedura wykrywania obiektów bazuje na informacjach z wszystkich czujników radarowych, generując jeden zestaw brył brzegowych dla całego otoczenia pojazdu testowego. Poniżej zaprezentowano przykładowe wyjście zwrócone przez algorytm wykrywania obiektów, uruchomiony w dwóch konfiguracjach - tj. osobno dla danych rzeczywistych (rysunek 8.2) oraz osobno dla danych syntetycznych (rysunek 8.3). Wykresy są oznaczone według następującego schematu:

- wykrycia radarowe umiejscowione są na wykresie w postaci punktów, przy czym do każdego czujnika przypisany jest jeden kolor: czerwony - RR, zielony - RL, niebieski - FR oraz żółty - FL,
- bryły brzegowe oszacowane na podstawie algorytmu wykrywania obiektów to kolorowe prostokąty o przerywanych liniach (rzeczywiste - kolor zielony; sztuczne - kolor pomarańczowy),
- obiekty referencyjne zawarte w wiadomości *OSI::GroundTruth* są bryłami zabarwionymi na szaro,

— pojazd testowy oznaczono kolorem czarnym.



Rysunek 8.2. Rzeczywista lista brył brzegowych zwrócona przez algorytm wykrywania obiektów na podstawie pomiarów z czujników radarowych typu FMCW



Rysunek 8.3. Sztuczna lista brył brzegowych oszacowana przez algorytm wykrywania obiektów na podstawie danych z czterech instancji modelu sensora

By umożliwić zestawienie rezultatów zwróconych przez algorytm wykrywania obiektów uruchomiony w dwóch konfiguracjach, czyli kolejno na podstawie symulowanych wykryć radarowych oraz wykryć zarejestrowanych w trakcie jazdy testowej, należy do każdego elementu ze zbioru $OSI::GroundTruth$ przypisać dokładnie jeden obiekt rzeczywisty oraz jeden obiekt syntetyczny. Zakłada się tutaj, że algorytm wykrywania obiektów zapisuje bryły brzegowe albo w formacie $OSI::MovingObject$, gdy dany element scenariusza zostanie zakwalifikowany jako cel poruszający się, albo w formacie $OSI::StationaryObject$, jeśli wykryty obiekt jest stacjonarny. Co za tym idzie, lista brył brzegowych uzyskana w danym momencie czasu przez procedurę wykrywania obiektów może być również zapisana w formacie $OSI::GroundTruth$, zgodnie ze schematem poglądowym zaprezentowanym na rysunku 6.6.

Przyjmijmy teraz, że każdy obiekt klasy $OSI::MovingObject$ oraz typu $OSI::StationaryObject$ może być reprezentowany przez taki oto zbiór wartości: $\{p, \Psi, d, \zeta\}$; gdzie $p, \Psi, d \in \mathbb{R}^3$ to odpowiednio współrzędne kartezjańskie pozycji obiektu i jego orientacja wyrażone względem układu GCS (patrz rozdział 5, rysunek 5.6) oraz wymiary tego obiektu, natomiast $\zeta \in \{0, 1\}$ to zmienna logiczna wskazująca na typ danego obiektu - wartość logiczna 0 powiązana jest z klasą $OSI::MovingObject$, z kolei wartość logiczną 1 wiąże się z typem $OSI::StationaryObject$. Warto tutaj skorzystać ponownie z oznaczenia, które wprowadzono w rozdziale 5 we wzorze (5.3.1). Mianowicie, niech $a, b \in \mathbb{N} : a \leq b$. Zbiór składający się z kolejnych liczb naturalnych spełniających warunek $a \leq x \leq b$ ($x \in \mathbb{N}$) można zapisać w taki oto sposób: $\{x : x \in [a, b]_{\mathbb{N}}\}$. Wykorzystując informacje dostępne dla całego wspomnianego wyżej zbioru pomiarów składającego się z 500 ramek danych, możemy zdefiniować trzy zbiory obiektów: $\mathbb{G}_{ref}$, $\mathbb{G}_{rzecz}$ i $\mathbb{G}_{synt}$, które określono w równaniach:

$$\begin{aligned}\mathbb{G}_{ref} &= \left\{ \{p, \Psi, d, \zeta\}^{i,n} \in G_{ref}^i : (i, n) \in [1, 500]_{\mathbb{N}} \times [1, N_{ref}^i]_{\mathbb{N}} \right\}, \\ \mathbb{G}_{rzecz} &= \left\{ \{p, \Psi, d, \zeta\}^{i,n} \in G_{rzecz}^i : (i, n) \in [1, 500]_{\mathbb{N}} \times [1, N_{rzecz}^i]_{\mathbb{N}} \right\}, \\ \mathbb{G}_{synt} &= \left\{ \{p, \Psi, d, \zeta\}^{i,n} \in G_{synt}^i : (i, n) \in [1, 500]_{\mathbb{N}} \times [1, N_{synt}^i]_{\mathbb{N}} \right\},\end{aligned}\tag{8.2.1a-c}$$

gdzie:

- i reprezentuje indeks ramki danych,
- G_{ref}^i, G_{rzecz}^i oraz G_{synt}^i to zbiór $OSI::GroundTruth$ wygenerowany dla dyskretnego momentu czasu i , odpowiednio na podstawie: danych referencyjnych oraz informacji zwróconych przez algorytm wykrywania obiektów przy użyciu kolejno pomiarów rzeczywistych i syntetycznych; przy czym zestaw G_{ref}^i nie zawiera informacji o pojeździe testowym wykorzystanym do przeprowadzenia jazdy testowej,
- n jest indeksem obiektu w danym zbiorze $OSI::GroundTruth$,
- N_{ref}^i, N_{rzecz}^i oraz N_{synt}^i to całkowita liczba obiektów w pojedynczym zbiorze $OSI::GroundTruth$, a więc kolejno: w G_{ref}^i, G_{rzecz}^i i G_{synt}^i .

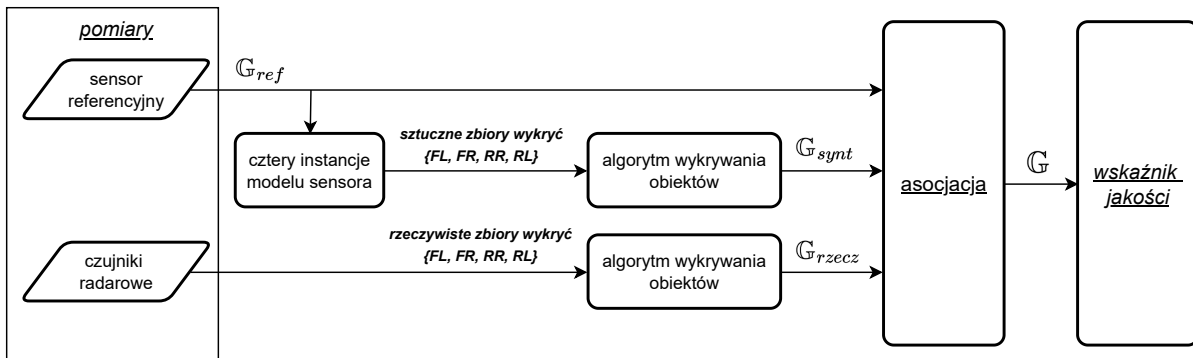
Oznaczmy zestaw wartości reprezentujących dany obiekt - $\{p, \Psi, d, \zeta\}$ - jako $\mathfrak{g}$, a zatem spełniona jest zależność: $p, \Psi, d, \zeta \in \mathfrak{g}$. Teraz wprowadźmy warunek określony we wzorze (8.2.2) na podobieństwo dwóch obiektów reprezentowanych kolejno przez $\mathfrak{g}_A$ i $\mathfrak{g}_B$:

$$\mathfrak{g}_A \sim \mathfrak{g}_B \Leftrightarrow \left(\exists (p_A, \zeta_A) \in \mathfrak{g}_A, \exists (p_B, \zeta_B) \in \mathfrak{g}_B : \|p_A - p_B\| < \epsilon \wedge \zeta_A = \zeta_B \right), \quad (8.2.2)$$

gdzie $\|\cdot\|$ to norma Euklidesowa wektora, natomiast $\epsilon \in \mathbb{R}$ jest stałą wartością reprezentującą tolerancję asocjacji. Zbiór skojarzonych ze sobą danych, konieczny do wyliczenia wskaźnika jakości, definiuje się na podstawie formuły:

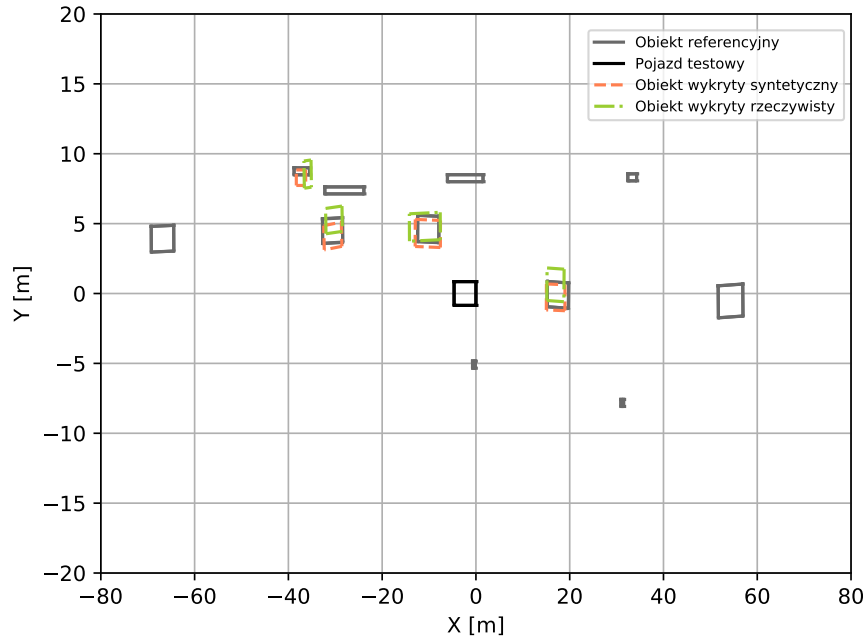
$$\mathbb{G} = \left\{ \{\mathfrak{g}_A, \mathfrak{g}_B, \mathfrak{g}_C\} : \mathfrak{g}_A \in \mathbb{G}_{synt}, \mathfrak{g}_B \in \mathbb{G}_{rzecz}, \mathfrak{g}_C \in \mathbb{G}_{ref} : \mathfrak{g}_A \sim \mathfrak{g}_B \wedge \mathfrak{g}_A \sim \mathfrak{g}_C \right\}. \quad (8.2.3)$$

Graficzną reprezentację opisaną wyżej procedury pokazano na schemacie blokowym widocznym na rysunku 8.4, ukazującym uproszczony przepływ informacji w procesie weryfikacji wiarygodności modelu sensora.



Rysunek 8.4. Schemat blokowy ukazujący uproszczony przepływ danych w procedurze weryfikacji wiarygodności modelu sensora

Z kolei przykładowy podzbiór skojarzonych ze sobą danych pokazano na rysunku 8.5. Znaczenie kolorów jest analogiczne jak na rysunkach 8.2 i 8.3.



Rysunek 8.5. Przykładowy podzbiór skojarzonych ze sobą danych

8.2.2 Wskaźniki jakości

Do oceny wiarygodności zostaną wykorzystane dwa wskaźniki jakości: $\mathbb{W}_A$ i $\mathbb{W}_B$. Oba wskaźniki będą bazować na tak zwanym współczynniku podobieństwa Jaccarda (ang. Intersection over Union, IoU) [141]. Miara ta zwraca wartość 1 w przypadku pełnego pokrycia dwóch brył oraz 0, gdy nie istnieje pomiędzy nimi żadna część wspólna. Przyjmijmy, że mamy dwie trójwymiarowe bryły brzegowe (prostopadłościany) o dowolnej orientacji reprezentowane kolejno przez $\mathbf{g}_A$ i $\mathbf{g}_B$. Współczynnik Jaccarda dla obiektów $\mathbf{g}_A$ oraz $\mathbf{g}_B$ wylicza się na podstawie następującego wzoru:

$$\mathbb{J}(\mathbf{g}_A, \mathbf{g}_B) = \frac{V_{\mathbf{g}_A \cap \mathbf{g}_B}}{V_{\mathbf{g}_A} + V_{\mathbf{g}_B} - V_{\mathbf{g}_A \cap \mathbf{g}_B}}, \quad (8.2.4)$$

gdzie $V_{\mathbf{g}_A}$ i $V_{\mathbf{g}_B}$ to kolejno objętość prostopadłościanu $\mathbf{g}_A$ i objętość prostopadłościanu $\mathbf{g}_B$, natomiast $V_{\mathbf{g}_A \cap \mathbf{g}_B}$ to objętość ich części wspólnej. Obliczenie objętości $V_{\mathbf{g}_A \cap \mathbf{g}_B}$ w przypadku trójwymiarowych obiektów o dowolnej orientacji jest zadaniem nietrywialnym. Niemniej jednak, w literaturze opisano do tej pory kilka skutecznych algorytmów rozwiązujących wspomniany problem [142–146].

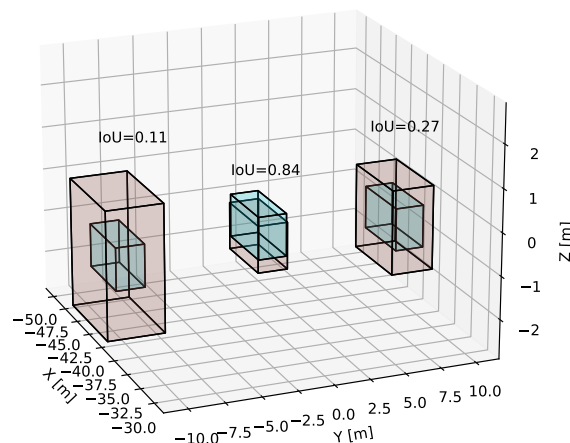
Niezwykle dokładną i jednocześnie bardzo wydajną metodę zaproponowali współtwórcy pakietu *PyTorch3D*, czyli biblioteki zaprojektowanej pod kątem sieci neuronowych operujących na trójwymiarowych danych [145]. Mianowicie, autorzy zauważyli, że część wspólna dwóch dowolnie zorientowanych brył, to w ogólności wypukły wielościan zbudowany z połączonych ze sobą w odpowiedni sposób trójkątów (podobnie jak modele geometryczne obiektów, patrz rysunek 5.4). Do

wyznaczenia wierzchołków wchodzących w skład docelowego wielościanu (punktów przecięcia brył A i B) wykorzystano usprawnioną wersję algorytmu Sutherlanda–Hodgmana [147]. Dzięki wprowadzonym modyfikacjom możliwe jest pominięcie wywołania metody *Quickhull* [148], w celu wyznaczenia otoczki wypukłej z uzyskanego zbioru punktów przecięcia, jak w przypadku metodologii zaproponowanej w [146]. Wspomniany wypukły wielościan, czyli część wspólna brył g_A oraz g_B , składa się z czworościanów, z których każdy rozpięty jest na wierzchołkach pojedynczego trójkąta oraz na środku wielościanu - kolejno punkty $a, b, c, d \in \mathbb{R}^3$. Objętość czworościanu o wymienionych wyżej wierzchołkach (o środku w punkcie d) oblicza się na podstawie wzoru:

$$V = \frac{|(a - d) \cdot ((b - d) \times (c - d))|}{6}, \quad (8.2.5)$$

gdzie $|x|$ to wartość bezwzględna z liczby x , natomiast $(\cdot)$ oraz $(\times)$ to operatory iloczynu skalarnego i wektorowego. Podsumowując, objętość $V_{g_A \cap g_B}$ to objętość omawianego wielościanu, która jest sumą objętości wszystkich czworościanów wchodzących w jego skład.

Na rysunku 8.6 zaprezentowano trzy pary obiektów wraz z wyliczonym współczynnikiem $\mathbb{J}$. Jak można zauważyć na wspomnianym rysunku, dla dwóch środkowych obiektów (o wysokim stopniu pokrycia) metryka $\mathbb{J}$ osiągnęła wartość 0.84, a więc bliską 1. Natomiast dla par obiektów odpowiednio po prawej i lewej stronie, gdzie stopień pokrycia się brył spada, wartość metryki $\mathbb{J}$ zmniejsza się do poziomu bliskiego zeru - kolejno 0.27 oraz 0.11.



Rysunek 8.6. Wartość współczynnika Jaccarda dla trzech par obiektów

Na tym etapie warto przypomnieć, że wykrycia radarowe, zarówno rzeczywiste, jak i symulowane, są obarczone sporymi niepewnościami. Należy zatem oczekiwać, że lista brył brzegowych wygenerowana przez algorytm wykrywania obiektów będzie również poddana pewnym zaburzeniom. Innymi słowy, obiekty radarowe nie będą idealnie dopasowane do ich odpowiedników ze zbioru $OSI::GroundTruth$ - ich pozycje, orientacje oraz wymiary będą oscylować z pewną amplitudą. Tego typu fluktuacje widać wyraźnie na załączonych rysunkach 8.2 oraz 8.3. Załóżmy, że zakłócenia zawarte w danym zbiorze brył brzegowych zwróconych przez algorytm wykrywania obiektów mogą być reprezentowane przez pomiar $\mathbb{J}$. Należy zatem wyznaczyć dwa zbiory określone w równaniach (8.2.6a) oraz (8.2.3b), kolejno: $\mathbb{J}_R(\mathbb{G}), \mathbb{J}_S(\mathbb{G}) \in \mathbb{R}^{|\mathbb{G}|}$, gdzie $|\mathbb{G}|$ to liczba elementów zestawu $\mathbb{G}$. Wspomniane zbiory zawierają więc wartości wskaźnika Jaccarda wyliczone dla: rzeczywistych obiektów względem danych referencyjnych oraz syntetycznych obiektów względem danych referencyjnych:

$$\begin{aligned}\mathbb{J}_R(\mathbb{G}) &= \left\{ \mathbb{J}(\mathbf{g}_B, \mathbf{g}_C) : \{\mathbf{g}_A, \mathbf{g}_B, \mathbf{g}_C\} \in \mathbb{G} \right\}, \\ \mathbb{J}_S(\mathbb{G}) &= \left\{ \mathbb{J}(\mathbf{g}_A, \mathbf{g}_C) : \{\mathbf{g}_A, \mathbf{g}_B, \mathbf{g}_C\} \in \mathbb{G} \right\}.\end{aligned}\tag{8.2.6a-b}$$

Przejdźmy teraz do definicji wskaźnika $\mathbb{W}_A$. W tym przypadku do zestawienia ze sobą danych rzeczywistych oraz syntetycznych zostanie wykorzystana metryka Wassersteina. Postępowanie będzie analogiczne, jak w rozdziale 7. Przypomnijmy, że $\mathbb{J}_R, \mathbb{J}_S \subset [0, 1]$. Oznaczmy przez $\mathbb{J} \subset [0, 1]$ zestaw wartości wskaźnika Jaccarda powiązany albo ze zbiorem $\mathbb{J}_R$, albo ze zbiorem $\mathbb{J}_S$. Zdefiniujmy teraz, podobnie jak w poprzednim rozdziale, dyskretną siatkę G_h ; z tym, że tym razem będzie ona określona na przedziale $[0, 1]$. Komórkę siatki G_h o indeksie i oraz o rozmiarze h określa się na podstawie następującego wzoru [133]:

$$G_h(i) = \left\{ x \in [0, 1] : \left\lfloor \frac{x}{h} \right\rfloor = i \right\},\tag{8.2.7}$$

gdzie $i \in \mathbb{N}$, $h = 10^{-n}$, $n \in \mathbb{N}^+$, $\left\lfloor \frac{x}{h} \right\rfloor$ to największa liczba naturalna nie większa od ilorazu x/h , natomiast $\mathbb{N}$ oraz $\mathbb{N}^+$ oznaczają kolejno zbiór liczb naturalnych i zbiór liczb naturalnych bez zera. Następnie wykonajmy rzut zbioru $\mathbb{J}$ na siatkę zdefiniowaną we wzorze (8.2.7). Przekształcony zestaw $\mathbb{J}_h$ zdefiniowano w równaniu:

$$\mathbb{J}_h = \left\{ i \in \mathbb{N} : \exists x \in \mathbb{J} : x \in G_h(i) \right\}.\tag{8.2.8}$$

Z kolei histogram zajętości $H : \mathbb{J}_h \rightarrow \mathbb{N}$ wyznacza się na podstawie wzoru:

$$H(i) = \left| \left\{ x \in \mathbb{J} : x \in G_h(i) \right\} \right|,\tag{8.2.9}$$

gdzie $|\cdot|$ oznacza liczbę elementów zbioru.

Natomiast dyskretny rozkład prawdopodobieństwa $\mu : \mathbb{J}_h \rightarrow [0, 1]$, właściwy dla przestrzeni $\mathbb{J}_h$ określonej w równaniu (8.2.8), można wyznaczyć przy użyciu takiej oto formuły:

$$\mu(i) = \frac{H(i)}{\sum_{j \in \mathbb{J}_h} H(j)}, \quad i \in \mathbb{J}_h. \quad (8.2.10)$$

W końcu, analogicznie jak w rozdziale 7, zdefiniujemy funkcję wzajemnie jednoznaczłą $\eta : \mathbb{M} \rightarrow \mathbb{J}_h$, gdzie $\mathbb{M} = \{1, \dots, \mathcal{M}\}$, $\mathcal{M} = |\mathbb{J}_h|$.

Na tym etapie wprowadzone zostały wszystkie elementy potrzebne do obliczenia metryki Wassersteina. Innymi słowy, wskaźnik $\mathbb{W}_A(\mathbb{G})$ (właściwy dla zbioru $\mathbb{G}$) to odległość Wassersteina $W(\mu, \nu)$ pomiędzy dwoma rozkładami prawdopodobieństwa μ oraz ν wyznaczonymi na podstawie równania (8.2.10), reprezentującymi pomiar wskaźnika Jaccarda odpowiednio dla danych rzeczywistych i danych syntetycznych; wskaźnik ten uzyskuje się na podstawie wzoru (7.3.10). Warto zaznaczyć, że funkcję $f^* : \mathbb{M} \times \mathbb{M} \rightarrow [0, 1]$ definiuje się identycznie jak w poprzednim rozdziale, a więc musi ona minimalizować sumę zapisaną we wzorze (7.3.8) oraz konieczne jest spełnienie ograniczeń zapisanych w równaniach (7.3.9a) - (7.3.9d).

Z kolei wskaźnik $\mathbb{W}_B(\mathbb{G})$ określa się na podstawie formuły:

$$\mathbb{W}_B(\mathbb{G}) = \left| \sum_{i \in \mathbb{J}_S} \frac{i}{|\mathbb{J}_S|} - \sum_{i \in \mathbb{J}_R} \frac{i}{|\mathbb{J}_R|} \right|, \quad (8.2.11)$$

gdzie $|\cdot|$ to wartość bezwzględna w przypadku operacji na wyniku zwróconym przez różnicę sum lub liczba elementów danego zbioru w pozostałych sytuacjach, natomiast $\mathbb{J}_S$ i $\mathbb{J}_R$ są zbiorami określonymi we wzorach odpowiednio (8.2.6a) oraz (8.2.3b).

8.2.3 Procedura oceny wiarygodności

Weryfikacja wiarygodności będzie przeprowadzona przy użyciu czterech różnych konfiguracji modelu sensora. Wartości parametrów są powiązane z przeprowadzonym w rozdziale 7 procesem optymalizacji, mającym na celu dobór optymalnych nastaw modelu, czyli uzyskania minimum lokalnego przy użyciu algorytmu Powella, dla przestrzeni określonej przez dobrany wskaźnik jakości. Innymi słowy, wybrane ustawienia modelu sensora odpowiadają następującym przypadkom:

- model sensora skonfigurowany przy użyciu nastaw optymalnych (minimum lokalne zwrócone przez algorytm Powella) zapisanych w tabeli 7.3 dla eksperymentu numer trzy (zestaw 1),
- model sensora skonfigurowany z wykorzystaniem nastaw odpowiadających kolejno: dolnemu zakresowi ograniczeń zbioru $\mathbb{I} \times \mathbb{J} \times \mathbb{K}$, czyli wysokiej rozdzielczości połączonej z bardzo niskim wzmocnieniem testu statystycznego (zestaw 2), oraz górnemu zakresowi ograniczeń zbioru $\mathbb{I} \times \mathbb{J} \times \mathbb{K}$, a więc niskiej rozdzielczości zestawionej ze sporym wzmocnieniem testu statystycznego (zestaw 3); gdzie zbiór $\mathbb{I} \times \mathbb{J} \times \mathbb{K}$ zdefiniowano w równaniach (7.5.1a) - (7.5.1c),

- oryginalny model sensora z ustawioną optymalną wartością wzmocnienia testu statystycznego (zestaw 4).

W tabeli 8.3 zaprezentowano wartości parametrów modelu sensora dla wymienionych wyżej konfiguracji. Warto wspomnieć, iż oryginalna wersja modelu wspiera generację wykryć jedynie dla obiektów klasy *OSI::MovingObject*, brak tutaj: sztucznych wykryć skojarzonych z elementami typu *OSI::StationaryObject*, filtracji rozdzielczości oraz generatora fałszywych pomiarów.

Tabela 8.3. Cztery zestawy nastaw wykorzystane na potrzeby oceny wiarygodności modelu sensora

Parametr	Zestaw 1	Zestaw 2	Zestaw 3	Zestaw 4
F_n	−15.84 [dB]	−30.0 [dB]	10 [dB]	−15.84 [dB]
ΔR	0.31 [m]	0.1 [m]	1.0 [m]	—
$\Delta\theta$	0.33 [°]	0.25 [°]	1.25 [°]	—
σ_x	51.25 [m]	50 [m]	50 [m]	—
σ_y	57.55 [m]	50 [m]	50 [m]	—
M	9	6	6	—

Jak wspomniano wyżej, do oceny wiarygodności potrzebne są dwa wskaźniki jakości. Rolą wskaźnika $\mathbb{W}_A$ jest zweryfikowanie, czy modyfikacje wprowadzone do oryginalnego modelu sensora przyczyniły się do faktycznego wzrostu wiarygodności sztucznego zbioru pomiarów oraz ocena skuteczności przeprowadzonej procedury identyfikacji parametrów modelu. Natomiast wskaźnik $\mathbb{W}_B$, czyli błąd bezwzględny pomiędzy sztucznym zbiorem danych a rzeczywistym zbiorem danych, wyznaczony na podstawie średnich odczytów pomiaru Jaccarda, służy do odpowiedzi na pytanie o wiarygodność modelu sensora.

Stwierdza się, że model sensora wytwarza wiarygodny zestaw wykryć (chmurę punktów), jeśli zbiór brył brzegowych uzyskany w oparciu o dane syntetyczne charakteryzuje się szumem o podobnej charakterystyce jak lista obiektów wygenerowana na podstawie rzeczywistych pomiarów radarowych. Wykorzystując wprowadzone wyżej definicje, warunek ten można sformułować w następujący sposób - model sensora jest wiarygodny, gdy:

- Wskaźnik $\mathbb{W}_A$ osiąga najniższą wartość dla trybu z modelem sensora skonfigurowanym przy użyciu nastaw zwróconych przez opisaną w poprzednim rozdziale procedurę optymalizacji, czyli dla zestawu 1.
- Wskaźnik $\mathbb{W}_B$ wyznaczony dla zestawu 1 jest mniejszy od ustalonej tolerancji.

Innymi słowy, procedurę oceny wiarygodności modelu sensora można sprowadzić do zweryfikowania, czy spełnione są warunki zdefiniowane we wzorach:

$$\min \left(\left\{ \mathbb{W}_A(\mathbb{G}_1), \mathbb{W}_A(\mathbb{G}_2), \mathbb{W}_A(\mathbb{G}_3), \mathbb{W}_A(\mathbb{G}_4) \right\} \right) = \mathbb{W}_A(\mathbb{G}_1), \quad \mathbb{W}_B(\mathbb{G}_1) < \epsilon, \quad (8.2.12a-b)$$

gdzie $\epsilon > 0$ to wspomniana tolerancja procesu weryfikacji wiarygodności.

8.2.4 Rezultaty

Przypomnijmy, że w pełni przetworzony zestaw danych wejściowych do wskaźników $\mathbb{W}_A$ i $\mathbb{W}_B$ to zbiory $\mathbb{J}_R(\mathbb{G})$ oraz $\mathbb{J}_S(\mathbb{G})$ określone w równaniach (8.2.6a) i (8.2.3b), które wyznaczono na podstawie 500 następujących po sobie dyskretnych momentów czasu nagranej kampanii testowej. Zauważmy, iż części składowe obu wspomnianych zbiorów, czyli wartości wskaźnika Jaccarda, obliczane są tylko dla elementów z zestawu $\mathbb{G}$, a więc dla każdej skojarzonej ze sobą trójki obiektów: $\{\mathbf{g}_A, \mathbf{g}_B, \mathbf{g}_C\} \in \mathbb{G}$. Co za tym idzie, całkowita liczba skojarzeń (liczba $|\mathbb{G}|$) różnić się będzie w zależności od wybranego wariantu. Dzieje się tak dla tego, iż charakterystyka oraz liczba brył generowanych przez algorytm wykrywania obiektów zależy w znacznym stopniu od właściwości sztucznego zbioru wykryć, które z kolei są uwarunkowane aktualnie dobranymi nastawami modelu. W tabeli 8.4 zaprezentowano wartość wskaźnika jakości $\mathbb{W}_A$ wyliczoną dla zestawów parametrów wymienionych w tabeli 8.3, wraz z liczbą skutecznych asocjacji.

Tabela 8.4. Wartość wskaźnika $\mathbb{W}_A$ wraz z całkowitą liczbą skutecznych asocjacji dla czterech różnych konfiguracji modelu sensora

Konfiguracja	Wskaźnik $\mathbb{W}_A$	Liczba skojarzeń
Zestaw 1	0.0901	2404
Zestaw 2	0.1461	2516
Zestaw 3	0.1044	1799
Zestaw 4	0.1305	1962

Zgodnie z oczekiwaniami liczba skojarzeń różni się w zależności od konfiguracji modelu. Dla wysokiej rozdzielczości oraz niewielkiego wzmocnienia testu statystycznego liczba elementów zbioru $\mathbb{G}$ jest największa. Efekt ten należy tłumaczyć znaczną gęstością rozmieszczenia sztucznych wykryć radarowych w przestrzeni. Z kolei dla niskiej rozdzielczości oraz dużego wzmocnienia testu statystycznego liczba skojarzeń drastycznie spadła, co również wynika z gęstości syntetycznego zbioru pomiarów. Oba opisane przypadki można zaobserwować na rysunku 7.3. Znacznie mniejszą liczbę asocjacji uzyskano także dla konfiguracji odpowiadającej oryginalnemu modelowi sensora, co jest skutkiem braku wykryć wygenerowanych dla obiektów stacjonarnych.

Najważniejszą kwestią w kontekście oceny wiarygodności pozostaje jednak odpowiedź na pytanie, czy model sensora generuje realistyczną listę obiektów dla nastaw dobranych w procesie optymalizacji.

Analizując dane dostępne w tabeli 8.4 można zauważyć, że najniższa wartość wskaźnika $\mathbb{W}_A$ przypada dla optymalnych nastaw modelu sensora. Natomiast wyniki uzyskane dla pozostałych trybów są wyraźnie gorsze. Warto jednak skomentować nieco dokładniej otrzymane rezultaty. Skoncentrujmy się w pierwszej kolejności na konfiguracjach numer 2 oraz numer 3. Jak już wspomniano, zestaw numer 2 to tryb, w którym zarówno filtr rozdzielczości, jak i wzmocnienie testu statystycznego są ustawione w taki sposób, że wykrycia zwrócone przez model sensora będą stosunkowo gęsto rozmieszczone w przestrzeni. Sytuacja ta nie pokrywa się, niestety, z rzeczywistością, gdyż radary typu FMCW charakteryzują się niewielkim zagęszczeniem pomiarów dla poszczególnych obiektów (patrz rysunek 5.16a). Rozbieżność tę odzwierciedla wyższa wartość wskaźnika $\mathbb{W}_A$. Dla konfiguracji numer 3 jest wprost przeciwnie. Niska rozdzielczość oraz wysokie wzmocnienie testu statystycznego redukują zbytnio liczebność zestawu sztucznych wykryć generowanych przez model sensora. Można to zaobserwować nie tylko w wyniku zwróconym dla tego zestawu przez wskaźnik $\mathbb{W}_A$, który jest gorszy w porównaniu do zestawu 1. Drastycznie niższa w stosunku do pozostałych trybów jest także liczba skojarzeń. Przyczyny należy szukać w tym, że generacja brył brzegowych z wykorzystaniem algorytmu wykrywania obiektów, na podstawie zbioru sztucznych wykryć o tego typu charakterystyce, była znacznie utrudniona. Natomiast jeśli chodzi o konfigurację odpowiadającą oryginalnemu modelowi sensora, to jest ona podobna do zestawu 2. Pominięcie filtracji rozdzielczości skutkuje w zbyt dużej gęstości sztucznego zbioru pomiarów, co z kolei negatywnie wpływa na wartość wskaźnika $\mathbb{W}_A$. Natomiast, jak wspomniano wcześniej, brak wykryć generowanych dla obiektów klasy *OSI::StationaryObject* powoduje to, iż algorytm wykrywania obiektów nie zwraca brył brzegowych dla stacjonarnych elementów scenariusza (w przeciwieństwie do danych rzeczywistych), dlatego też liczba skojarzeń jest dużo niższa, niż dla zestawu 1. Podsumowując, dla optymalnych ustawień modelu sensora wskaźnik $\mathbb{W}_A$ osiągnął najniższą wartość, przy zachowaniu wysokiej liczby skojarzeń, a zatem spełniony jest warunek zapisany we wzorze (8.2.12a).

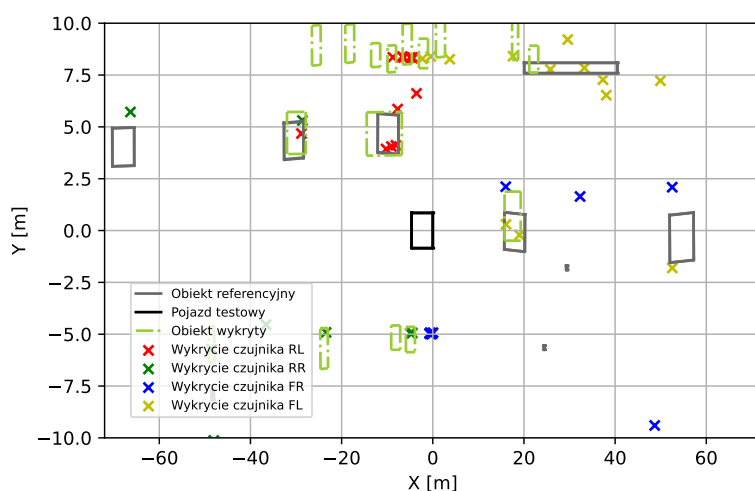
W tabeli 8.5 zaprezentowano natomiast wyniki zwrócone przez wskaźnik $\mathbb{W}_B$ dla ustawień wymienionych w tabeli 8.3. Tutaj również najniższą wartość uzyskano dla zestawu 1. Warunek określony we wzorze (8.2.12b) będzie zatem spełniony dla $\epsilon > 0.08266$. Jest to akceptowalna wartość tolerancji, a więc, zgodnie z wprowadzonymi definicjami, model sensora uznaje się za wiarygodny.

Tabela 8.5. Wartość wskaźnika $\mathbb{W}_B$ dla czterech różnych konfiguracji modelu sensora

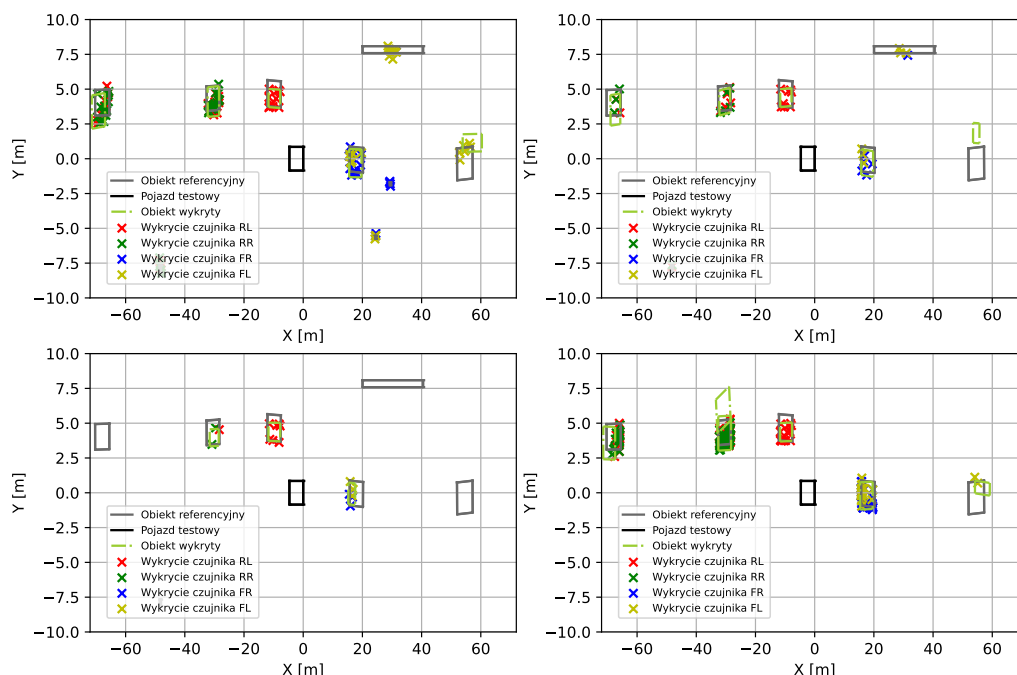
Konfiguracja	Wskaźnik $\mathbb{W}_B$
Zestaw 1	0.08266
Zestaw 2	0.14139
Zestaw 3	0.09190
Zestaw 4	0.12469

Na rysunkach 8.7 - 8.12 zaprezentowano listy brył brzegowych wygenerowane przez algorytm wykrywania obiektów dla trzech dyskretnych momentów czasu wykorzystanego scenariusza testowego (A, B oraz C), dla różnych typów pomiarów radarowych. W szczególności, na rysunkach 8.7, 8.9 i 8.11 ukazano listy obiektów uzyskane na podstawie rzeczywistych wykryć radarowych, kolejno dla czasu A, B oraz C. Natomiast na rysunkach 8.8, 8.10 i 8.12 - odpowiednio czas A, B, C - pokazano listy obiektów wygenerowane przy użyciu sztucznych pomiarów radarowych zwróconych przez model sensora skonfigurowany z wykorzystaniem czterech typów nastaw wymienionych w tabeli 8.3.

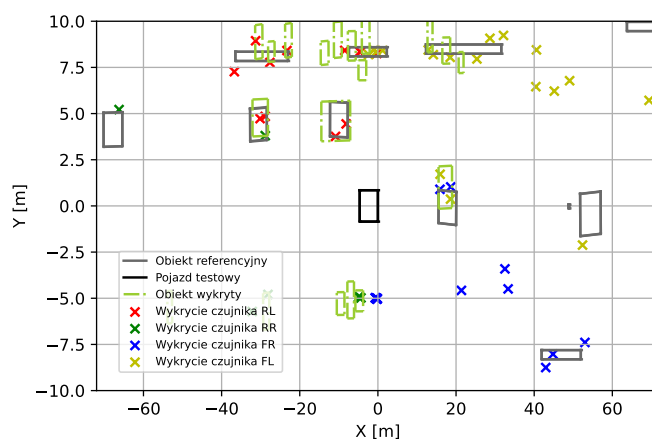
Podsumowując, w tym rozdziale opisano procedury oceny wydajności i wiarygodności modelu, czyli kluczowych czynników wykorzystywanych do weryfikacji, czy model sensora jest użyteczny w kontekście testów systemów typu ADAS w środowisku wirtualnej jazdy. Rezultaty zaprezentowane w tabelach 8.1 i 8.2 potwierdzają, że zmodyfikowany model sensora będzie zdolny do testów uruchamianych w czasie rzeczywistym, niezależnie od poziomu skomplikowania wirtualnego scenariusza. Z kolei wyniki widoczne na rysunkach 8.7 - 8.12 uwierzytelniają dane zamieszczone w tabelach 8.4 oraz 8.5. Innymi słowy, liczba oraz charakterystyka brył brzegowych wygenerowanych przez algorytm wykrywania obiektów jest najbardziej zbliżona do rzeczywistych danych dla modelu sensora działającego z wykorzystaniem nastaw dobranych w przeprowadzonym w poprzednim rozdziale procesie optymalizacji. Spełnione są więc warunki zapisane we wzorach (8.2.12a) i (8.2.12b), co zaświadcza o wiarygodności modelu.



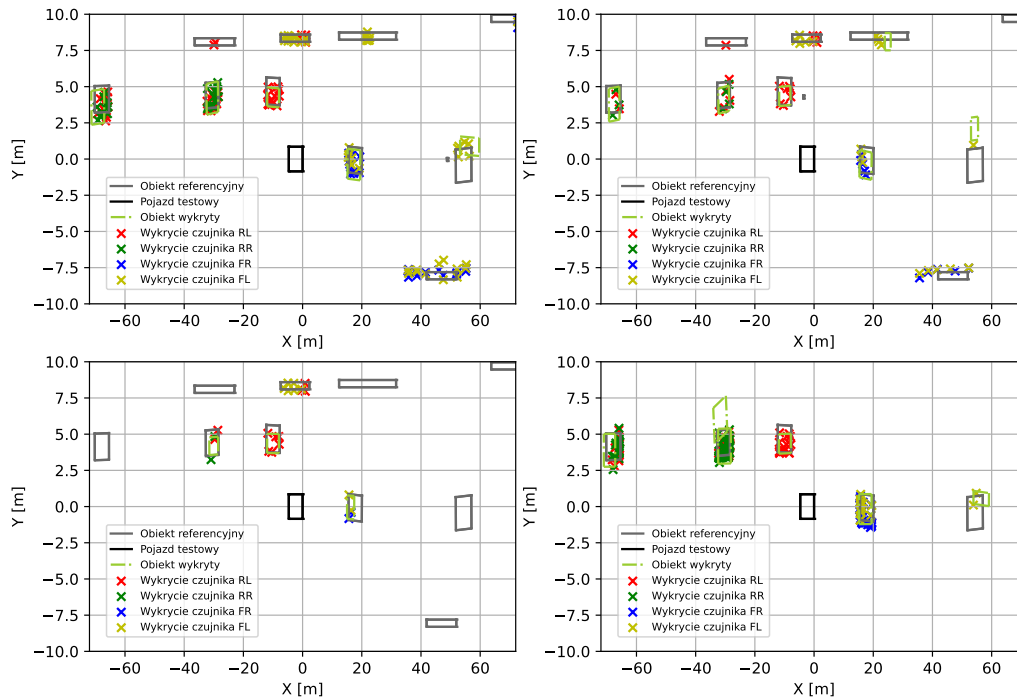
Rysunek 8.7. Lista brył brzegowych oszacowana przez algorytm wykrywania obiektów dla dyskretnego momentu czasu A, na podstawie rzeczywistych pomiarów radarowych



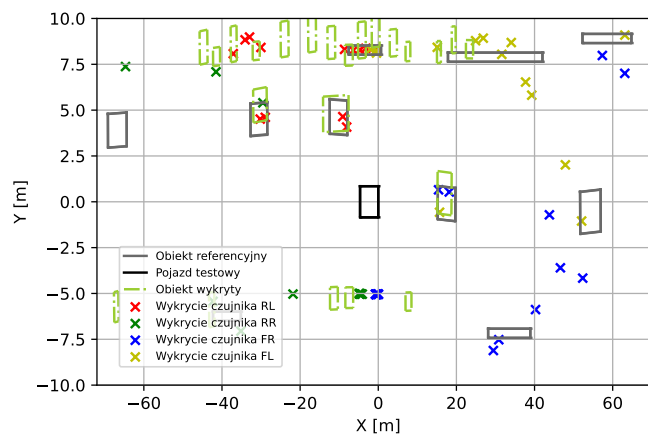
Rysunek 8.8. Lista brył brzegowych oszacowana przez algorytm wykrywania obiektów dla dyskretnego momentu czasu $\mathbb{A}$, na podstawie sztucznych pomiarów wygenerowanych przez model sensora skonfigurowany przy użyciu nastaw zawartych odpowiednio w: zestawie 2 (lewy górny róg), zestawie 1 (ustawienia optymalne, prawy górny róg), zestawie 3 (lewy dolny róg), zestawie 4 (oryginalny model sensora, prawy dolny róg)



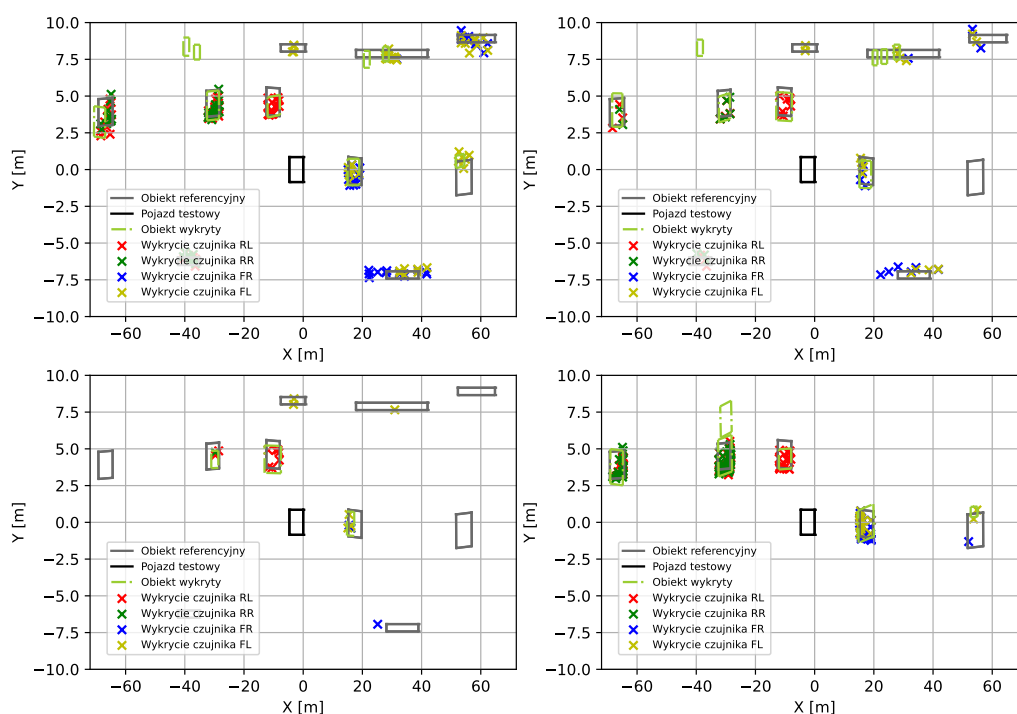
Rysunek 8.9. Lista brył brzegowych oszacowana przez algorytm wykrywania obiektów dla dyskretnego momentu czasu $\mathbb{B}$, na podstawie rzeczywistych pomiarów radarowych



Rysunek 8.10. Lista brył brzegowych oszacowana przez algorytm wykrywania obiektów dla dyskretnego momentu czasu $\mathbb{B}$, na podstawie sztucznych pomiarów wygenerowanych przez model sensora skonfigurowany przy użyciu nastaw zawartych odpowiednio w: zestawie 2 (lewy górny róg), zestawie 1 (ustawienia optymalne, prawy górny róg), zestawie 3 (lewy dolny róg), zestawie 4 (oryginalny model sensora, prawy dolny róg)



Rysunek 8.11. Lista brył brzegowych oszacowana przez algorytm wykrywania obiektów dla dyskretnego momentu czasu $\mathbb{C}$, na podstawie rzeczywistych pomiarów radarowych



Rysunek 8.12. Lista brył brzegowych oszacowana przez algorytm wykrywania obiektów dla dyskretnego momentu czasu $\mathbb{C}$, na podstawie sztucznych pomiarów wygenerowanych przez model sensora skonfigurowany przy użyciu nastaw zawartych odpowiednio w: zestawie 2 (lewy górny róg), zestawie 1 (ustawienia optymalne, prawy górny róg), zestawie 3 (lewy dolny róg), zestawie 4 (oryginalny model sensora, prawy dolny róg)

9 Podsumowanie

W ostatnich latach w przemyśle motoryzacyjnym obserwujemy gwałtowny rozwój zaawansowanych systemów wspomagania kierowcy (ang. Advanced Driver Assistance Systems, ADAS). Jedną z przyczyn takiego stanu rzeczy są coraz bardziej rygorystyczne normy homologacyjne nałożone przez Unię Europejską na producentów samochodów. W szczególności norma z 2019 roku [6] wspomina, że każdy nowo wyprodukowany pojazd, począwszy od 2022 roku, musi być obowiązkowo wyposażony w takie systemy, jak: zaawansowany system hamowania awaryjnego (ang. Advanced Emergency Braking System, AEBS), inteligentny asystent kontroli prędkości (ang. Intelligent Speed Assistance, ISA) czy system awaryjnego utrzymywania pojazdu w pasie ruchu (ang. Emergency Lane-Keeping System, ELKS). Aplikacje te bazują zwykle na informacjach przekazywanych przez czujniki wizyjne oraz czujniki radarowe. Radar samochodowy jest szczególnie istotny w przypadku układu typu AEBS, czyli systemu pozwalającego na uniknięcie groźnych w skutkach kolizji czołowych, gdyż dla każdego pojazdu widocznego z perspektywy sensora mierzy jednocześnie jego odległość od aparatury, prędkość względną oraz kąt w płaszczyźnie horyzontalnej.

Jako że urządzenia klasy ADAS mają bezpośredni wpływ na trajektorię pojazdu, to przed wdrożeniem danego systemu do produkcji konieczna jest jego rzetelna weryfikacja w celu ocenienia zachowania projektowanego algorytmu w najróżniejszych sytuacjach drogowych. Proces ten wymaga przeprowadzenia szeregu jazd testowych, w trakcie których zarejestrowane będą zarówno pomiary z czujników zamocowanych na pojeździe, jak i decyzje podjęte przez system wspomagania kierowcy na podstawie danych zwróconych przez sensory. Zrealizowanie weryfikacji systemów typu ADAS w całości na drogach publicznych byłoby przedsięwzięciem zbyt kosztownym i trudnym, ponieważ wymagałoby zgromadzenia wspomnianych wcześniej informacji z możliwie jak najbardziej zróżnicowanych scenariuszy; należałoby tutaj uwzględnić w szczególności takie czynniki, jak: warunki pogodowe, pora dnia, natężenie ruchu itp. Z tego też względu producenci samochodów coraz częściej decydują się na wdrożenie koncepcji tzw. wirtualnej walidacji polegającej na wykonaniu części eksperymentów weryfikacyjnych w symulatorach jazdy. Takie podejście skraca nie tylko czas i koszt całej operacji. Metodologia ta umożliwia także sprawdzenie w kontrolowany i powtarzalny sposób, jak dany układ klasy ADAS radzi sobie w skrajnie niebezpiecznych sytuacjach. Wirtualna walidacja musi być oczywiście

wiarygodna. Co za tym idzie, wykorzystana w tym procesie symulacja musi spełniać dwa kluczowe warunki. Po pierwsze, konieczne jest dokładne odwzorowanie prawdziwych scenariuszy jazdy. Po drugie, należy wygenerować w czasie rzeczywistym sztuczne dane wiernie naśladujące pomiary z czujników. Ta ostatnia operacja, określana mianem modelu sensora, jest nadzwyczaj trudna w przypadku radaru, głównie ze względu na silny niedeterminizm procesu propagacji sygnału elektromagnetycznego.

W tej pracy wykorzystano niezwykle innowacyjny model matematyczny czujnika radarowego oparty na koncepcji tzw. centrów rozpraszania, opublikowany w [21]. Wydaje się, że idea ta ma ogromny potencjał w kontekście zaprojektowania wirtualnego czujnika spełniającego wymagania wirtualnej walidacji. Zaletą tej metody to przede wszystkim brak bezpośredniej symulacji procesu rozchodzenia się fali radiowej w przestrzeni, czyli cechy charakterystycznej tzw. modeli fizycznych, dzięki czemu można utworzyć syntetyczny zbiór wykryć radarowych przy użyciu stosunkowo niewielkich zasobów obliczeniowych. Ponadto do zaprojektowania wspomnianego modelu nie są wymagane pokaźne zasoby danych z rzeczywistego systemu radarowego, tak jak to jest w przypadku metodologii bazującej na sztucznych sieciach neuronowych; do przeprowadzenia procedury identyfikacji parametrów jest konieczna tylko niewielka liczba pomiarów.

Niestety, po przeprowadzeniu dokładnego porównania prawdziwych wykryć radarowych z odpowiadającymi im sztucznymi pomiarami wygenerowanymi przy użyciu omawianego modelu radaru okazuje się, że wirtualny czujnik zgodny z wersją zaprezentowaną w przytoczonej publikacji nie jest wystarczająco rzetelny do wiarygodnego przetestowania układu typu ADAS w symulatorze jazdy. W trakcie badań zidentyfikowano w wykorzystanym modelu matematycznym sensora następujące braki. Po pierwsze, syntetyczne wykrycia generowane są tylko dla pojazdów; w sztucznym zbiorze danych brak pomiarów powiązanych z obiektami statycznymi, takimi jak na przykład barierki charakterystyczne dla wielopasmowych dróg szybkiego ruchu. Po drugie, model czujnika nie wytwarza fałszywych wykryć, czyli efektu, który można zaobserwować w danych zwracanych przez radary samochodowe. Po trzecie, gęstość rozmieszczenia punktów w zbiorze syntetycznych wykryć wydaje się zbyt duża w odniesieniu do informacji rzeczywistych.

Cel tejże rozprawy to adaptacja wspomnianego modelu matematycznego czujnika radarowego pod kątem wymagań przemysłu motoryzacyjnego dotyczących testów systemów klasy ADAS w środowiskach wirtualnej jazdy. Należało więc wprowadzić do oryginalnego modelu szereg modyfikacji w każdym z trzech rozpoznanych wyżej obszarów, przy użyciu nieznacznej ilości danych rzeczywistych, tak, by zapewnić odpowiedni poziom rzetelności wirtualnego czujnika. Przede wszystkim zintegrowano model z kluczowym w dziedzinie wirtualnej walidacji standardem ASAM OSI. Proces integracji wymagał następujących kroków: zmodyfikowanie wejścia do modelu i wyjścia z modelu, by były one zgodne ze standardem ASAM OSI, oraz zdefiniowanie transformacji pomiędzy typami obiektów obsługiwanych przez oryginalny model sensora a typami obiektów zawartych w strukturze

OSI::VehicleClassification. Dzięki wsparciu standardu ASAM OSI zapewniona została wszechstronność modelu matematycznego, czyli fakt, iż jest on wiarygodny niezależnie od poziomu rozbudowania scenariusza jazdy. Następnie rozszerzono symulację o wsparcie dla obiektów statycznych, co zaprezentowano na przykładzie barierek wyznaczających krawędzie jezdni. Ponadto dodano także do modelu moduł odpowiedzialny za generację fałszywych wykryć, a także specjalny filtr rozdzielczości mający na celu zredukowanie zagęszczenia syntetycznych pomiarów.

Zauważmy, że stwierdzenie wiarygodności i wydajności modelu jest wystarczające do udowodnienia postawionej na początku tezy, iż model matematyczny czujnika radarowego bazujący na koncepcji centrów rozpraszania może być użyteczny w kontekście testów systemu urządzeń typu ADAS w czasie rzeczywistym w symulatorze jazdy, niezależnie od stopnia skomplikowania wirtualnego scenariusza. By udowodnić postawioną tezę należy więc ocenić, czy sztuczne dane wygenerowane przy użyciu modelu sensora wiernie odwzorowują pomiary z rzeczywistego systemu radarowego. Żeby to osiągnąć, przeprowadzono dwie kluczowe operacje. Wpierw wykonano procedurę identyfikacji parametrów modelu z oryginalnie dobranym wskaźnikiem jakości bazującym na metryce Wassersteina porównującym w bezpośredni sposób rzeczywisty i sztuczny zbiór wykryć; umożliwiło to dobranie odpowiednich ustawień modelu, czyli takich, dla których funkcja kosztu osiąga najniższą wartość. W końcu przeprowadzono ocenę wiarygodności sztucznie generowanych pomiarów z użyciem autorskiej metody walidacji modelu matematycznego czujnika radarowego, polegającej na wykorzystaniu listy brył brzegowych utworzonej przez algorytm wykrywania obiektów. Innymi słowy, wygenerowano dwa zestawy brył brzegowych, a więc dla danych rzeczywistych oraz dla danych syntetycznych, które następnie porównano ze sobą, stosując kolejny raz metrykę Wassersteina, choć w tym wypadku z inną funkcją odległości.

Rezultaty osiągnięte w procesie weryfikacji modelu sensora jednoznacznie pokazują, że wirtualny radar zbudowany na bazie koncepcji centrów rozpraszania po wprowadzeniu opisanych wyżej modyfikacji w sposób wiarygodny odwzorowuje dane zwracane przez rzeczywisty system radarowy. Co najważniejsze, rozmieszczenie poszczególnych wykryć jest zbliżone do pomiarów generowanych przez wykorzystany radar samochodowy. Ponadto algorytm wykrywania obiektów wytwarza zbieżne listy brył brzegowych zarówno dla informacji rzeczywistych, jak i syntetycznych. W końcu zweryfikowano także zdolność modelu matematycznego do pracy w czasie rzeczywistym. Potwierdza to zatem tezę, że zmodyfikowany model sensora jest użyteczny w procesie weryfikacji urządzeń klasy ADAS w środowiskach wirtualnej jazdy.

Podsumowując, za główny wkład pracy w odniesieniu do celu i tezy rozprawy należy uznać:

- (i) Formalizację modelu pod kątem standardu ASAM OSI, w szczególności: zdefiniowanie w jednoznaczny sposób wejścia do modelu oraz wyjścia z modelu zgodnego ze standardem ASAM OSI, określenie powiązania pomiędzy typami obiektów wspieranymi przez oryginalny model

radaru a typami obiektów dostępnych w strukturze *OSI::VehicleClassification*, wprowadzenie szeregu wzorów umożliwiających formalny i kompaktowy zapis operacji wykonywanych w ramach modelu matematycznego, co przyczyniło się do określenia jasnej i czytelnej definicji wskaźnika jakości.

- (ii) Zidentyfikowanie opisanych wyżej obszarów, w których oryginalny model matematyczny może być udoskonalony, na podstawie pomiarów z rzeczywistego czujnika radarowego.
- (iii) Dodanie nowatorskich usprawnień do wspomnianego modelu matematycznego, takich jak: rozszerzenie modelu o wsparcie dla obiektów statycznych, dodanie modułu odpowiedzialnego za generację fałszywych wykryć oraz wprowadzenie specjalnego filtra rozdzielczości, mającego na celu zredukowanie zagęszczenia zbioru sztucznych wykryć.
- (iv) Opracowanie i wykonanie procedury identyfikacji parametrów modelu z oryginalnie dobranym wskaźnikiem jakości.
- (v) Zrealizowanie autorskiej metody oceny wiarygodności i wydajności modelu, dzięki czemu potwierdzono użyteczność modelu sensora w kontekście wirtualnej walidacji systemów typu ADAS.

Warto na koniec wspomnieć, że rezultaty pracy zostały wdrożone do praktyki przemysłowej do weryfikacji elektronicznych systemów sterowania jazdy automatycznej na poziomie L2-L3 przeznaczonych do produkcji seryjnej dla kluczowym producentów samochodów.

Literatura

- [1] I. Gresham, N. Jain, T. Budka, A. Alexanian, N. Kinayman, B. Ziegner, S. Brown, and P. Staecker. A compact manufacturable 76-77-ghz radar module for commercial acc applications. In *IEEE Transactions on Microwave Theory and Techniques*, volume 49, pages 44–58, 2001. doi: 10.1109/22.899961.
- [2] K. Kodaka, M. Otabe, Y. Urai, and H. Koike. Rear-end collision velocity reduction system. In *SAE Transactions*, volume 112, pages 502–510. SAE International, 2003. doi: 10.4271/2003-01-0503.
- [3] M. Maurer. Forward collision warning and avoidance. In *Handbook of Intelligent Vehicles*, pages 657–687. Springer London, London, 2012. doi: 10.1007/978-0-85729-085-4_25.
- [4] A. Ziebinski, R. Cupek, H. Erdogan, and S. Waechter. A survey of adas technologies for the future perspective of sensor fusion. In *Computational Collective Intelligence*, volume 9876, pages 135–146, Cham, 2016. Springer International Publishing. doi: 10.1007/978-3-319-45246-3_13.
- [5] Rozporządzenie parlamentu europejskiego i rady (we) nr 661/2009 z dnia 13 lipca 2009 r. w sprawie wymagań technicznych w zakresie homologacji typu pojazdów silnikowych dotyczących ich bezpieczeństwa ogólnego, ich przyczep oraz przeznaczonych dla nich układów, części i oddzielnych zespołów technicznych. In *Dziennik Urzędowy Unii Europejskiej*, volume L200, pages 1–25. 2009-07-31.
- [6] Rozporządzenie parlamentu europejskiego i rady (ue) 2019/2144 z dnia 27 listopada 2019 r. w sprawie wymogów dotyczących homologacji typu pojazdów silnikowych i ich przyczep oraz układów, komponentów i oddzielnych zespołów technicznych przeznaczonych do tych pojazdów, w odniesieniu do ich ogólnego bezpieczeństwa oraz ochrony osób znajdujących się w pojeździe i niechronionych uczestników ruchu drogowego. In *Dziennik Urzędowy Unii Europejskiej*, volume L325, pages 1–41. 2019-12-16.
- [7] T. L. Overton, T. E. Rives, C. Hecht, S. Shafi, and R. R. Gandhi. Distracted driving: prevalence, problems, and prevention. *International Journal of Injury Control and Safety Promotion*, 22(3): 187–192, 2015. doi: 10.1080/17457300.2013.879482.

- [8] N. Kalra and S. M. Paddock. Driving to safety: How many miles of driving would it take to demonstrate autonomous vehicle reliability? *Transportation Research Part A: Policy and Practice*, 94:182 – 193, 2016. doi: 10.1016/j.tra.2016.09.010.
- [9] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun. Carla: An open urban driving simulator. In *Proceedings of the 1st Annual Conference on Robot Learning*, volume 78, pages 1–16. PMLR, 2017.
- [10] S. Shah, D. Dey, C. Lovett, and A. Kapoor. Airsim: High-fidelity visual and physical simulation for autonomous vehicles. In *Field and Service Robotics*, pages 621–635, Cham, 2018. Springer International Publishing. doi: 10.1007/978-3-319-67361-5_40.
- [11] T. Sulkowski, P. Bugiel, and J. Izydorczyk. In search of the ultimate autonomous driving simulator. In *2018 International Conference on Signals and Electronic Systems (ICSES)*, pages 252–256, Sep. 2018. doi: 10.1109/ICSES.2018.8507288.
- [12] M. Holder, P. Rosenberger, H. Winner, T. D’hondt, V. P. Makkapati, M. Maier, H. Schreiber, Z. Magosi, Z. Slavik, O. Bringmann, and W. Rosenstiel. Measurements revealing challenges in radar sensor modeling for virtual validation of autonomous driving. In *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, pages 2616–2622, 2018. doi: 10.1109/ITSC.2018.8569423.
- [13] M. Jasiński. A generic validation scheme for real-time capable automotive radar sensor models integrated into an autonomous driving simulator. In *2019 24th International Conference on Methods and Models in Automation and Robotics (MMAR)*, pages 612–617, 2019. doi: 10.1109/MMAR.2019.8864669.
- [14] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016. doi: 10.1109/CVPR.2016.90.
- [15] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger. Densely connected convolutional networks. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2261–2269, 2017. doi: 10.1109/CVPR.2017.243.
- [16] X. Chen, H. Ma, J. Wan, B. Li, and T. Xia. Multi-view 3d object detection network for autonomous driving. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6526–6534, 2017. doi: 10.1109/CVPR.2017.691.
- [17] M. Simon, S. Milz, K. Amende, and H.-M. Gross. Complex-yolo: An euler-region-proposal for real-time 3d object detection on point clouds. In *Computer Vision – ECCV 2018 Workshops*:

- Munich, Germany, September 8-14, 2018, Proceedings, Part I*, page 197–209, Berlin, 2019. Springer-Verlag. doi: 10.1007/978-3-030-11009-3_11.
- [18] A. H. Lang, S. Vora, H. Caesar, L. Zhou, J. Yang, and O. Beijbom. Pointpillars: Fast encoders for object detection from point clouds. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12689–12697, 2019. doi: 10.1109/CVPR.2019.01298.
- [19] J. Ku, M. Mozifian, J. Lee, A. Harakeh, and S. Waslander. Joint 3d proposal generation and object detection from view aggregation. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1–8, 10 2018. doi: 10.1109/IROS.2018.8594049.
- [20] R. Bhalla and Hao Ling. Three-dimensional scattering center extraction using the shooting and bouncing ray technique. *IEEE Transactions on Antennas and Propagation*, 44(11):1445–1453, 1996. doi: 10.1109/8.542068.
- [21] A. Martowicz, A. Gallina, and G. Karpziel. Uncertainty propagation for vehicle detections in experimentally validated radar model for automotive application. In *2019 24th International Conference on Methods and Models in Automation and Robotics (MMAR)*, pages 606–611, 2019. doi: 10.1109/MMAR.2019.8864641.
- [22] M. I. Skolnik. *Introduction to radar systems 2nd edition*. New York, Auckland McGraw-Hill, 1980.
- [23] Ch. Hülsmeyer. Hertzian-wave projecting and receiving apparatus adapted to indicate or give warning of the presence of a metallic body, such as a ship or a train, in the line of projection of such waves, 1904. British Patent 13170.
- [24] A. H. Taylor, L. C. Young, and L. A. Hyland. System for detecting objects by radio, 1934. U.S. Patent 1981884.
- [25] A. P. Rowe. *One Story of Radar*. Cambridge University Press, New York, 1948.
- [26] Sir R. Watson-Watt. *Three Steps to Victory*. Odhams Press, Ltd., London, 1957.
- [27] Sir R. Watson-Watt. *The Pulse of Radar*. The Dial Press, Inc., New York, 1959.
- [28] I. Wolff and R. C. Sanders Jr. Frequency-modulated radio altimeter, 1939. U.S. Patent 2257830.
- [29] V. D. Landon and W. L. Carlson. Frequency modulation altimeter and range finder, 1947. U.S. Patent 2537593.
- [30] H. Goldberg. Continuous wave frequency modulation distance measuring apparatus, 1957. U.S. Patent 3112481.

- [31] S. James. Fm cw distance measuring system, 1963. U.S. Patent 3229286.
- [32] S. H. Peperone. Frequency modulated ranging device, 1969. U.S. Patent 3605094.
- [33] J. C. Fletcher, H. Brey, and P. E. Geise Jr. Fm/cw radar system, 1976. U.S. Patent 4118701.
- [34] Ch. Waldschmidt, J. Hasch, and W. Menzel. Automotive radar — from first efforts to future systems. *IEEE Journal of Microwaves*, 1(1):135–148, 2021. doi: 10.1109/JMW.2020.3033616.
- [35] A.L. Merlo. Automotive radar for the prevention of collisions. In *IEEE Transactions on Industrial Electronics and Control Instrumentation*, volume IECI-11, pages 1–6, 1964. doi: 10.1109/TIECI.1964.234462.
- [36] E.J. Baghdady. Automobile radar. In *1971 IEEE GMTT International Microwave Symposium Digest*, pages 170–171, 1971. doi: 10.1109/GMTT.1971.1122951.
- [37] J.E. Stevens and L.L. Nagy. A duplex doppler radar automotive obstacle detection system. In *23rd IEEE Vehicular Technology Conference*, volume 23, pages 70–81, 1972. doi: 10.1109/VTC.1972.1622126.
- [38] L.L. Nagy and J.A.M. Lyon. An ultrashort pulse radar sensor for vehicular precollision obstacle detection. In *IEEE Transactions on Vehicular Technology*, volume 24, pages 41–45, 1975. doi: 10.1109/T-VT.1975.23603.
- [39] M. Kiyoto, T. Kondoh, Kazuhiro B., and K. Shirahata. Radar sensor for automobiles. In *1974 IEEE International Solid-State Circuits Conference. Digest of Technical Papers*, volume 17, pages 74–75, 1974. doi: 10.1109/ISSCC.1974.1155268.
- [40] T. Tamama, A. Iwabe, Kazuhiro B., M. Tsudo, S. Mitsui, K. Baba, M. Kiyoto, Hiroshi E., and N. Fujiki. Radar sensor for automotive collision prevention. In *1978 IEEE-MTT-S International Microwave Symposium Digest*, pages 168–170, 1978. doi: 10.1109/MWSYM.1978.1123825.
- [41] D. H. Zur and H. Oehlen. Radar anticollision warning system for road vehicles. In *Electrical Communication*, volume 52, pages 141–145. International Telephone and Telegraph Corporation, 1977.
- [42] G. Neininger. Fm/cw radar with high resolution in range and doppler; application for anti-collision radar for vehicles. In *1977 International Radar Conference*, pages 526–534, 1977.
- [43] E. H. Düll and H. J. Peters. Collision avoidance system for automobiles. In *SAE Transactions*, volume 87, pages 1250–1260, 1978. doi: 10.4271/780263.

- [44] T. Senzaki and E. Murao. Fm-cw radar system for use in an automotive vehicle, 1980. U.S. Patent 4348675.
- [45] H. Ono and J. Kojima. Fm-cw multibeam radar apparatus, 1994. U.S. Patent 5652589.
- [46] E. Lissel, R. Mende, and H. Rohling. Radar process for the measurement of distances and relative speeds between a vehicle and one or more obstructions, 1995. U.S. Patent 5757308.
- [47] Y. Yamada. Fm-cw radar, 1997. U.S. Patent 6121917.
- [48] J.D. Woll. Vorad collision warning radar. In *Proceedings International Radar Conference*, pages 369–372, 1995. doi: 10.1109/RADAR.1995.522574.
- [49] J.D. Woll. Monopulse doppler radar for vehicle applications. In *Proceedings of the Intelligent Vehicles '95. Symposium*, pages 42–47, 1995. doi: 10.1109/IVS.1995.528255.
- [50] H. H. Meinel. Evolving automotive radar — from the very beginnings into the future. In *The 8th European Conference on Antennas and Propagation (EuCAP 2014)*, pages 3107–3114, 2014. doi: 10.1109/EuCAP.2014.6902486.
- [51] H. P. Forstner, H. Knapp, H. Jager, E. Kolmhofer, J. Platz, F. Starzer, M. Treml, A. Schinko, G. Birschkus, J. Bock, K. Aufinger, R. Lachner, T. Meister, H. Schafer, D. Lukashevich, S. Boguth, A. Fischer, F. Reininger, L. Maurer, J. Minichshofer, and D. Steinbuch. A 77ghz 4-channel automotive radar transceiver in sige. In *2008 IEEE Radio Frequency Integrated Circuits Symposium*, pages 233–236, 2008. doi: 10.1109/RFIC.2008.4561425.
- [52] S. W. Alland. Digital beamforming for an electronically scanned radar system, 2005. U.S. Patent 7474262.
- [53] I. Slomian, P. Kaminski, J. Sorocki, I. Piekarz, K. Wincza, and S. Gruszczynski. Multi-beam and multi-range antenna array for 24 ghz radar applications. In *2014 20th International Conference on Microwaves, Radar and Wireless Communications (MIKON)*, pages 1–4, 2014. doi: 10.1109/MIKON.2014.6899867.
- [54] J.A. Scheer. Coherent radar system performance estimation. In *IEEE International Conference on Radar*, pages 125–128, 1990. doi: 10.1109/RADAR.1990.201149.
- [55] A. Wojtkiewicz, J. Misiurewicz, M. Nałęcz, K. Jedrzejewski, and K. Kulpa. Two-dimensional signal processing in fmcw radars. 1997.
- [56] S. W. Alland and J. F. Searcy. Radar system and method of digital beamforming, 2008. U.S. Patent 7639171.

- [57] M. A. Richards. *Fundamentals of Radar Signal Processing 2nd edition*. New York, McGraw-Hill Education, 2014.
- [58] H. Rohling. Radar cfar thresholding in clutter and multiple target situations. In *IEEE Transactions on Aerospace and Electronic Systems*, volume AES-19, pages 608–621, 1983. doi: 10.1109/TAES.1983.309350.
- [59] T. P. Zieliński. *Cyfrowe przetwarzanie sygnałów: od teorii do zastosowań*. Wydawnictwa Komunikacji Łączności Warsaw, Poland, 2005.
- [60] K. Lelowicz, M. Jasiński, and M. Piątek. Generic sensor model for object detection algorithms validation. In *Advanced, Contemporary Control*, pages 1249–1260, Cham, 2020. Springer International Publishing. doi: 10.1007/978-3-030-50936-1_104.
- [61] K. Granström, M. Baum, and S. Reuter. Extended object tracking: Introduction, overview, and applications. *Journal of Advances in Information Fusion*, 12, 2017.
- [62] J. Gunnarsson, L. Svensson, L. Danielsson, and F. Bengtsson. Tracking vehicles using radar detections. In *2007 IEEE Intelligent Vehicles Symposium*, pages 296–302, 2007. doi: 10.1109/IVS.2007.4290130.
- [63] A. Scheel, Ch. Knill, S. Reuter, and K. Dietmayer. Multi-sensor multi-object tracking of vehicles using high-resolution radars. In *2016 IEEE Intelligent Vehicles Symposium (IV)*, pages 558–565, 2016. doi: 10.1109/IVS.2016.7535442.
- [64] K. Thormann, M. Baum, and J. Honer. Extended target tracking using gaussian processes with high-resolution automotive radar. In *2018 21st International Conference on Information Fusion (FUSION)*, pages 1764–1770, 2018. doi: 10.23919/ICIF.2018.8455630.
- [65] L. Zhang and J. Lan. Tracking of extended object using random matrix with non-uniformly distributed measurements. *IEEE Transactions on Signal Processing*, 69:3812–3825, 2021. doi: 10.1109/TSP.2021.3090946.
- [66] M. Kumru and E. Özkan. Three-dimensional extended object tracking and shape learning using gaussian processes. *IEEE Transactions on Aerospace and Electronic Systems*, 57(5):2795–2814, 2021. doi: 10.1109/TAES.2021.3067668.
- [67] A. Scheel and K. Dietmayer. Tracking multiple vehicles using a variational radar model. *IEEE Transactions on Intelligent Transportation Systems*, 20(10):3721–3736, 2019. doi: 10.1109/TITS.2018.2879041.

- [68] Y. Xia, P. Wang, K. Berntorp, L. Svensson, K. Granström, H. Mansour, P. Boufounos, and P. V. Orlik. Learning-based extended object tracking using hierarchical truncation measurement model with automotive radar. *IEEE Journal of Selected Topics in Signal Processing*, 15(4):1013–1029, 2021. doi: 10.1109/JSTSP.2021.3058062.
- [69] ASAM e. V. Association for standarization of automation and measuring systems - standards, . URL <https://asam.net/standards/>. Data dostępu: 20-03-2023.
- [70] AUTOSAR. Automotive open system architecture - standards. URL <https://www.autosar.org/>. Data dostępu: 22-03-2023.
- [71] T. Wheeler, M. Holder, H. Winner, and M. Kochenderfer. Deep stochastic radar models. pages 47–53, 2017. doi: 10.1109/IVS.2017.7995697.
- [72] A. Krizhevsky, I. Sutskever, and G. Hinton. Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6):84–90, 2017. doi: 10.1145/3065386.
- [73] J. Walker, C. Doersch, A. Gupta, and M. Hebert. An uncertain future: Forecasting from static images using variational autoencoders. In *Computer Vision – ECCV 2016*, volume 9911, pages 835–851, Cham, 2016. Springer International Publishing. doi: https://doi.org/10.1007/978-3-319-46478-7_51.
- [74] K. Sohn, H. Lee, and X. Yan. Learning structured output representation using deep conditional generative models. In *Proceedings of the 28th International Conference on Neural Information Processing Systems*, volume 2, page 3483–3491, Cambridge, 2015. MIT Press. doi: 10.5555/2969442.2969628.
- [75] H. Attias. Inferring parameters and structure of latent variable models by variational bayes. In *Proceedings of the Fifteenth Conference on Uncertainty in Artificial Intelligence*, page 21–30, San Francisco, 1999. Morgan Kaufmann Publishers Inc. doi: 10.5555/2073796.2073799.
- [76] J. Ebert, T. Gump, S. Münzner, A. Matskevych, Alexandru P. Condurache, and C. Gläser. Deep radar sensor models for accurate and robust object tracking. In *2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC)*, pages 1–6, 2020. doi: 10.1109/ITSC45102.2020.9294735.
- [77] F. Ciepiela, M. K. Nowak, D. Dworak, and M. Komorkiewicz. Automotive radar detection level modeling with neural networks. In *Advanced, Contemporary Control*, volume 709, pages 254–265, Cham, 2023. Springer Nature Switzerland. doi: 10.1007/978-3-031-35173-0_25.

- [78] H. Caesar, V. Bankiti, A. H. Lang, S. Vora, V. Liong, Q. Xu, A. Krishnan, Y. Pan, G. Baldan, and O. Beijbom. nuscenes: A multimodal dataset for autonomous driving. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11618–11628, Los Alamitos, 2020. IEEE Computer Society. doi: 10.1109/CVPR42600.2020.01164.
- [79] S. Muckenhuber, E. Museljic, and G. Stettinger. Performance evaluation of a state-of-the-art automotive radar and corresponding modeling approaches based on a large labeled dataset. *Journal of Intelligent Transportation Systems*, 26(6):655–674, 2022. doi: 10.1080/15472450.2021.1959328.
- [80] T. Eder, R. Hachicha, H. Sellami, C. van Driesten, and E. Biebl. Data driven radar detection models: A comparison of artificial neural networks and non parametric density estimators on synthetically generated radar data. In *2019 Kleinheubach Conference*, pages 1–4, 2019.
- [81] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial networks. *Communications of the ACM*, 63(11):139–144, 2020. doi: 10.1145/3422622.
- [82] M. Arjovsky, S. Chintala, and L. Bottou. Wasserstein generative adversarial networks. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70, page 214–223. Journal of Machine Learning Research, 2017. doi: 10.5555/3305381.3305404.
- [83] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros. Image-to-image translation with conditional adversarial networks. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5967–5976, Los Alamitos, 2017. IEEE Computer Society. doi: 10.1109/CVPR.2017.632.
- [84] T. Karras, S. Laine, and T. Aila. A style-based generator architecture for generative adversarial networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(12):4217–4228, 2021. doi: 10.1109/TPAMI.2020.2970919.
- [85] R. Weston, O. P. Jones, and I. Posner. There and back again: Learning to simulate radar data for real-world applications. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 12809–12816, 2021. doi: 10.1109/ICRA48506.2021.9562111.
- [86] H. Ling, R. . Chou, and S. . Lee. Shooting and bouncing rays: calculating the rcs of an arbitrarily shaped cavity. *IEEE Transactions on Antennas and Propagation*, 37(2):194–205, 1989.
- [87] Ch.-F. Yang, B.-Ch. Wu, and Ch.-J. Ko. A ray-tracing method for modeling indoor wave propagation and penetration. *IEEE Transactions on Antennas and Propagation*, 46(6):907–919, 1998. doi: 10.1109/8.686780.

- [88] D. Gubelli, O. A. Krasnov, and O. Yarovy. Ray-tracing simulator for radar signals propagation in radar networks. In *2013 European Radar Conference*, pages 73–76, 2013.
- [89] N. Hirsenkorn, P. Subkowski, T. Hanke, A. Schaermann, A. Rauch, R. Rasshofer, and E. Biebl. A ray launching approach for modeling an fmcw radar system. In *2017 18th International Radar Symposium (IRS)*, pages 1–10, 2017. doi: 10.23919/IRS.2017.8008120.
- [90] M. M. Taygur and T. F. Eibert. A ray-tracing algorithm based on the computation of (exact) ray paths with bidirectional ray-tracing. *IEEE Transactions on Antennas and Propagation*, 68(8): 6277–6286, 2020. doi: 10.1109/TAP.2020.2983775.
- [91] M. Holder, C. Linnhoff, P. Rosenberger, and H. Winner. The fourier tracing approach for modeling automotive radar sensors. In *2019 20th International Radar Symposium (IRS)*, pages 1–8, 2019. doi: 10.23919/IRS.2019.8768113.
- [92] Ch. Stetco, B. Ubezio, S. Mühlbacher-Karrer, and H. Zangl. Radar sensors in collaborative robotics: Fast simulation and experimental validation. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 10452–10458, 2020. doi: 10.1109/ICRA40945.2020.9197180.
- [93] K. Sabet, A. Sabet, J. Kral, and Ch. Woischwill. Hybrid computer simulation of automotive radar systems in high multipath environments. In *2020 IEEE International Symposium on Antennas and Propagation and North American Radio Science Meeting*, pages 1561–1562, 2020. doi: 10.1109/IEEECONF35879.2020.9330417.
- [94] J. Thieling, S. Frese, and J. Roßmann. Scalable and physical radar sensor simulation for interacting digital twins. *IEEE Sensors Journal*, 21(3):3184–3192, 2021. doi: 10.1109/JSEN.2020.3026416.
- [95] Ch. Schüßler, M. Hoffmann, J. Bräunig, I. Ullmann, R. Ebel, and M. Vossiek. A realistic radar ray tracing simulator for large mimo-arrays in automotive environments. *IEEE Journal of Microwaves*, 1(4):962–974, 2021. doi: 10.1109/JMW.2021.3104722.
- [96] Ch. Schüler, M. Hoffmann, V. Wirth, B. Eskofier, T. Weyrich, M. Stamminger, and M. Vossiek. Achieving efficient and realistic full-radar simulations and automatic data annotation by exploiting ray meta data from a radar ray tracing simulator. In *2023 IEEE Radar Conference (RadarConf23)*, pages 1–6, 2023. doi: 10.1109/RadarConf2351548.2023.10149641.
- [97] Ch. Schöffmann, B. Ubezio, Ch. Böhm, S. Mühlbacher-Karrer, and H. Zangl. Virtual radar: Real-time millimeter-wave radar sensor simulation for perception-driven robotics. *IEEE Robotics and Automation Letters*, 6(3):4704–4711, 2021. doi: 10.1109/LRA.2021.3068916.

- [98] J. Guo, W. Deng, S. Zhang, S. Qi, X. Li, Ch. Wang, and J. Wang. A novel method of radar modeling for vehicle intelligence. *SAE International Journal of Passenger Cars - Electronic and Electrical Systems*, 10:50–56, 2016. doi: 10.4271/2016-01-1892.
- [99] Z. F. Magosi and A. Eichberger. A novel approach for simulation of automotive radar sensors designed for systematic support of vehicle development. *Sensors*, 23(6), 2023. doi: 10.3390/s23063227.
- [100] Z. Slavik and K. V. Mishra. Phenomenological modeling of millimeter-wave automotive radar. In *2019 URSI Asia-Pacific Radio Science Conference (AP-RASC)*, pages 1–4, 2019. doi: 10.23919/URSIAP-RASC.2019.8738137.
- [101] K. Schuler, D. Becker, and W. Wiesbeck. Extraction of virtual scattering centers of vehicles by ray-tracing simulations. *Antennas and Propagation, IEEE Transactions on*, 56:3543 – 3551, 12 2008. doi: 10.1109/TAP.2008.2005436.
- [102] M. Bühren and B. Yang. Simulation of automotive radar target lists using a novel approach of object representation. In *2006 IEEE Intelligent Vehicles Symposium*, pages 314–319, 2006. doi: 10.1109/IVS.2006.1689647.
- [103] M. Bühren and B. Yang. Simulation of automotive radar target lists considering clutter and limited resolution. 2011.
- [104] S. M. Key. *Fundamentals of Statistical Signal Processing, Volume II: Detection Theory*. Pearson Education, London, 1998.
- [105] R.H. Rasshofer, J. Rank, and G. Zhang. Generalized modeling of radar sensors for next-generation virtual driver assistance function prototyping. *Intelligent Transportation Society of America - 12th World Congress on Intelligent Transport Systems 2005*, 4:2511–2522, 01 2009.
- [106] S. Bernsteiner, Z. F. Magosi, D. Lindvai-Soos, and A. Eichberger. Radar sensor model for the virtual development process. *ATZelektronik worldwide*, 10:46–52, 04 2015. doi: 10.1007/s38314-015-0521-1.
- [107] T. Hanke, N. Hirsenkorn, B. Dehlink, A. Rauch, R. Rasshofer, and E. Biebl. Generic architecture for simulation of adas sensors. In *2015 16th International Radar Symposium (IRS)*, pages 125–130, 2015. doi: 10.1109/IRS.2015.7226306.
- [108] N. Hirsenkorn, T. Hanke, A. Rauch, B. Dehlink, R. Rasshofer, and E. Biebl. A non-parametric approach for modeling sensor behavior. In *2015 16th International Radar Symposium (IRS)*, pages 131–136, 2015. doi: 10.1109/IRS.2015.7226346.

- [109] N. Hirsenkorn, T. Hanke, A. Rauch, B. Dehlink, R. Rasshofer, and E. Biebl. Virtual sensor models for real-time applications. *Advances in Radio Science*, 14:31–37, 2016. doi: 10.5194/ars-14-31-2016.
- [110] E. Parzen. On Estimation of a Probability Density Function and Mode. *The Annals of Mathematical Statistics*, 33(3):1065–1076, 1962. doi: 10.1214/aoms/1177704472.
- [111] M. Rosenblatt. Remarks on Some Nonparametric Estimates of a Density Function. *The Annals of Mathematical Statistics*, 27(3):832–837, 1956. doi: 10.1214/aoms/1177728190.
- [112] H. Li, T. Kanuric, and A. Eichberger. Automotive radar modeling for virtual simulation based on mixture density network. *IEEE Sensors Journal*, 23(11):11117–11124, 2023. doi: 10.1109/JSEN.2022.3223765.
- [113] Ch. M. Bishop. Mixture density networks. Workingpaper, Aston University, 1994.
- [114] E. Zec, N. Mohammadiha, and A. Schliep. Statistical sensor modelling for autonomous driving using autoregressive input-output hmms. In *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, pages 1331–1336, 2018. doi: 10.1109/ITSC.2018.8569592.
- [115] H. Arnelid, E. Zec, and N. Mohammadiha. Recurrent conditional generative adversarial networks for autonomous driving sensor modelling. In *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, pages 1613–1618, 10 2019. doi: 10.1109/ITSC.2019.8916999.
- [116] W. Turlej. High-level sensor models for the reinforcement learning driving policy training. *Electronics*, 12(1), 2023. doi: 10.3390/electronics12010071.
- [117] K. Lelowicz and A. K. Piłat. Generic sensor model usecase exemplified by pedestrian crossing. *IEEE Sensors Journal*, 22(22):21986–21995, 2022. doi: 10.1109/JSEN.2022.3211092.
- [118] M. Stolz and G. Nestlinger. Fast generic sensor models for testing highly automated vehicles in simulation. *e & i - Elektrotechnik und Informationstechnik*, 135, 2018. doi: 10.1007/s00502-018-0629-0.
- [119] S. Muckenhuber, H. Holzer, J. Rübsam, and G. Stettinger. Object-based sensor model for virtual testing of adas/ad functions. In *2019 IEEE International Conference on Connected Vehicles and Expo (ICCVE)*, pages 1–6, 2019. doi: 10.1109/ICCVE45908.2019.8965071.
- [120] M. Piątek and K. Lelowicz. Method for estimating visibility of objects, 2019. U.S. Patent 11199392.

- [121] M. Jasiński. Method for road occlusions handling in generic sensor models. In *2021 25th International Conference on Methods and Models in Automation and Robotics (MMAR)*, pages 179–184, 2021. doi: 10.1109/MMAR49549.2021.9528443.
- [122] T Hanke, N Hirsenkorn, C van Driesten, P Garcia-Ramos, M Schiementz, S Schneider, and E Biebl. Open simulation interface - a generic interface for the environment perception of automated driving functions in virtual scenarios, 2017. URL <https://www.hot.ei.tum.de/forschung/automotive-veroeffentlichungen/>. Data dostępu: 10-09-2021.
- [123] CARLA Simulator. Carla documentation, 2023. URL <https://carla.readthedocs.io/>. Data dostępu: 20-04-2023.
- [124] ASAM e. V. Asam osi user guide, . URL <https://www.asam.net/standards/detail/osi/>. Data dostępu: 12-04-2023.
- [125] E. W. Weisstein. Euler angles, 2009. URL <https://mathworld.wolfram.com/EulerAngles.html>. Data dostępu: 10-08-2023.
- [126] D.A. Varshalovich, A.N. Moskalev, and V.K. Khersonskii. Description of rotation in terms of the euler angles. *Quantum theory of angular momentum*, pages 21–23, 1988. doi: 10.1142/0270.
- [127] E. W. Weisstein. Rotation matrix, 2003. URL <https://mathworld.wolfram.com/RotationMatrix.html>. Data dostępu: 10-08-2023.
- [128] H. Goldstein. *Classical Mechanics 2nd edition*. Addison-Wesley Publishing Company, Boston, 1980.
- [129] S. Riedmaier, B. Danquah, B. Schick, and F. Diermeyer. Unified framework and survey for model verification, validation and uncertainty quantification. *Archives of Computational Methods in Engineering*, 28, 2020. doi: 10.1007/s11831-020-09473-7.
- [130] M. Jasinski, P. Skruch, and M. Komorkiewicz. Validation framework for generic radar sensor models. *IEEE Access*, 10:18257–18267, 2022. doi: 10.1109/ACCESS.2022.3150284.
- [131] A. Ramdas, N. Garcia, and M. Cuturi. On wasserstein two-sample testing and related families of nonparametric tests. *Entropy*, 19(2), 2017. doi: 10.3390/e19020047.
- [132] P. Kowalczyk, P. Bugiel, J. Izydorczyk, and M. Szelest. Clustering of test scenes by use of Wasserstein metric and analysis of tracking quality in optically degraded videos. *Elektronika, Telekomunikacja, Mobilność*, 2021. doi: 10.5281/zenodo.4628038.

- [133] P. Kowalczyk, M. Komorkiewicz, P. Skruch, and M. Szelest. Efficient characterization method for big automotive datasets used for perception system development and verification. *IEEE Access*, 10:12629–12643, 2022. doi: 10.1109/ACCESS.2022.3145192.
- [134] W. H. Cunningham. A network simplex method. *Mathematical Programming*, 11:105–116, 1976. doi: 10.1007/BF01580379.
- [135] N. Bonneel, M. Van De Panne, S. Paris, and W. Heidrich. Displacement interpolation using lagrangian mass transport. *ACM Transactions on Graphics*, 30(6):1–12, 2011. doi: 10.1145/2070781.2024192.
- [136] M. J. D. Powell. An efficient method for finding the minimum of a function of several variables without calculating derivatives. *The Computer Journal*, 7(2):155–162, 1964. doi: 10.1093/comjnl/7.2.155.
- [137] A. Korytowski and M. Ziółko. *Metody optymalizacji z ćwiczeniami laboratoryjnymi*. Skrypty Uczelniane - Akademia Górniczo-Hutnicza im. Stanisława Staszica. Wydawnictwo AGH, Kraków, 1992.
- [138] G. Guennebaud, B. Jacob, et al. Eigen v3, 2010. URL <http://eigen.tuxfamily.org>. Data dostępu: 20-07-2023.
- [139] J. D. Murray and VanRyper W. *Encyclopedia of Graphics File Formats 2nd edition*. O'Reilly Media, Sebastopol, USA, 1996.
- [140] Syoyo Fujita et al. Tinyobjloader. URL <https://github.com/tinyobjloader/tinyobjloader/>. Data dostępu: 20-09-2023.
- [141] A. Rosebrock. Intersection over union (iou) for object detection, 2016. URL <https://www.pyimagesearch.com/2016/11/07/intersection-over-union-iou-for-object-detection/>. Data dostępu: 18-08-2023.
- [142] Y. Zheng, D. Zhang, S. Xie, J. Lu, and J. Zhou. Rotation-robust intersection over union for 3d object detection. In *Computer Vision – ECCV 2020*, pages 464–480, Cham, 2020. Springer International Publishing. doi: 10.1007/978-3-030-58565-5_28.
- [143] J. Xu, Y. Ma, S. He, and J. Zhu. 3d-giou: 3d generalized intersection over union for object detection in point cloud. *Sensors*, 19(19), 2019. doi: 10.3390/s19194093.
- [144] J. Li, H. Dai, L. Shao, and Y. Ding. From voxel to point: Iou-guided 3d object detection for point cloud with voxel-to-point decoder. In *Proceedings of the 29th ACM International Conference*

- on Multimedia*, page 4622–4631, New York, 2021. Association for Computing Machinery. doi: 10.1145/3474085.3475314.
- [145] J. Johnson, N. Ravi, J. Reizenstein, D. Novotny, S. Tulsiani, Ch. Lassner, and B. S. Accelerating 3d deep learning with pytorch3d. In *SIGGRAPH Asia 2020 Courses*, New York, 2020. Association for Computing Machinery. doi: 10.1145/3415263.3419160.
- [146] A. Ahmadyan, L. Zhang, J. Wei, A. Ablavatski, and M. Grundmann. Objectron: A large scale dataset of object-centric videos in the wild with pose annotations. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7818–7827, 2021. doi: 10.1109/CVPR46437.2021.00773.
- [147] I. E. Sutherland and G. W. Hodgman. Reentrant polygon clipping. *Communications of the ACM*, 17(1):32–42, 1974. doi: 10.1145/360767.360802.
- [148] C. B. Barber, D. P. Dobkin, and H. Huhdanpaa. The quickhull algorithm for convex hulls. *ACM Transactions on Mathematical Software*, 22(4):469–483, 1996. doi: 10.1145/235815.235821.

Wykaz rysunków

2.1	Parametry pojedynczego wykrycia zwracanego przez radar FMCW	32
2.2	Modulacja częstotliwości z wykorzystaniem sygnału piłokształtnego	33
2.3	Zestaw anten odbiorczych	34
2.4	Uproszczony schemat blokowy radaru FMCW	36
2.5	Zmieniająca się na przestrzeni K pakietów faza liczby zespolonej (powiązanej jednoznacznie z częstotliwością różnicową f_B) dla pewnej ustalonej komórki odległości	38
2.6	Informacje wejściowe do procesu przetwarzania sygnałów w radarach typu FMCW (macierz danych w dziedzinie czasu), wraz z rezultatem tegoż procesu, czyli macierzą danych w dziedzinie częstotliwości (RDC)	39
2.7	Rezultat idealnego algorytmu wykrywania obiektów	41
2.8	Przykładowy scenariusz testowy weryfikujący poprawność działania układu typu AEBS	42
2.9	Wysokopoziomowy przepływ informacji w systemach typu ADAS bazujących na czujnikach radarowych, na przykładzie uproszczonego urządzenia klasy AEBS	43
2.10	Uproszczony schemat przykładowego środowiska do wirtualnej walidacji funkcjonalności klasy AEBS, należącej do kategorii układów typu ADAS, zrealizowanej w trybie SIL	45
2.11	Uproszczony schemat przykładowego środowiska do wirtualnej walidacji funkcjonalności klasy AEBS, należącej do kategorii układów typu ADAS, zrealizowanej w trybie HIL	45
2.12	Uproszczony schemat przykładowego środowiska do wirtualnej walidacji funkcjonalności klasy AEBS, należącej do kategorii układów typu ADAS, zrealizowanej w trybie SIL w oparciu o koncepcję GSM	47
2.13	Uproszczony schemat przykładowego środowiska do wirtualnej walidacji funkcjonalności klasy AEBS, należącej do kategorii układów typu ADAS, zrealizowanej w trybie HIL w oparciu o koncepcję GSM	47
3.1	Schemat poglądowy prezentujący w uproszczony sposób zawartość struktury OSI::SensorView	55

3.2	Schemat poglądowy prezentujący w uproszczony sposób zawartość struktury <i>OSI::SensorData</i>	58
3.3	Wirtualna walidacja przy wsparciu standardu OSI - wariant <i>A</i> oparty na wysokopoziomowym modelu sensora	60
3.4	Wirtualna walidacja przy wsparciu standardu OSI - wariant <i>B</i> bazujący na niskopoziomowym modelu sensora oraz na modelu logicznym, realizującym procedurę wykrywania obiektów	61
3.5	Wirtualna walidacja przy wsparciu standardu OSI - wariant <i>C</i> mający na celu odwzorowanie przypadku, w którym system percepcji wykorzystuje dane pochodzące z wielu czujników samochodowych; występuje tutaj N wysokopoziomowych modeli sensora oraz jeden model logiczny dokonujący fuzji informacji z N źródeł w jedną, łączoną listę obiektów	61
3.6	Wirtualna walidacja przy wsparciu standardu OSI - wariant <i>D</i> dla modułu percepcji korzystającego z N czujników samochodowych; strumień przetwarzania danych korzysta z: N niskopoziomowych modeli sensora generujących unikalne instancje struktury <i>OSI::FeatureData</i> , N modeli logicznych generujących listę obiektów dla każdego z uzyskanych zbiorów wykryć oraz z jednego modelu logicznego dokonującego fuzji danych, której rezultatem jest łączony zestaw obiektów	61
3.7	Wirtualna walidacja przy wsparciu standardu OSI - wariant <i>E</i> dla symulowanego modułu percepcji odwzorowującego system korzystający z N czujników samochodowych: w tym schemacie wpierw jeden model logiczny realizuje procedurę fuzji zbiorów wykryć wygenerowanych przez N niskopoziomowych modeli sensora (zapisanych w unikalnych strukturach <i>OSI::FeatureData</i>), z kolei później drugi model logiczny wykonuje operację wykrywania obiektów na bazie łączonego zestawu danych dostępnego w zestawie <i>OSI::FeatureData</i>	61
5.1	Schemat poglądowy prezentujący w uproszczony sposób fragment struktury <i>OSI::SensorView</i> zawierający N ($N \in \mathbb{N}$) obiektów klasy <i>OSI::MovingObject</i> , wymagany do wygenerowania przez model sensora poprawnej wiadomości typu <i>OSI::SensorData</i> mieszczącej sztuczny zbiór wykryć radarowych	71
5.2	Powiązanie pomiędzy kategoriami pojazdów wspieranych przez model sensora a typami pojazdów zdefiniowanych w zbiorze <i>OSI::VehicleClassification</i>	72
5.3	Przykłady struktur geometrycznych wygenerowanych dla kilku najważniejszych klas pojazdów samochodowych wspieranych przez model sensora	73

5.4	Przykładowa reprezentacja graficzna bazy modeli geometrycznych pojazdów samochodowych wraz z pokazową strukturą zbudowaną na bazie sześcianu	74
5.5	Wizualizacja zbiorów centrów rozpraszania wygenerowanych dla modeli geometrycznych pojazdów samochodowych wspieranych przez model sensora	76
5.6	Reprezentacja graficzna zależności pomiędzy układami współrzędnych zdefiniowanych w ramach standardu ASAM OSI	78
5.7	Wizualizacja zbioru centrów rozpraszania wyznaczonych dla obiektu klasy <i>OSI::MovingObject</i> o identyfikatorze równym 2 reprezentowanych przez współrzędne kartezjańskie pozycji środka danego trójkąta oraz wektor normalny płaszczyzny rozpiętej na wierzchołkach tego trójkąta. Model geometryczny właściwy dla wspomnianego obiektu składa się z czterech parami symetrycznych wielokątów	83
5.8	Porównanie sztucznych zbiorów wykryć wyznaczonych kolejno przy użyciu wzorów (5.5.1) oraz (5.5.6). W drugim przypadku sprawdzany jest dodatkowo warunek na kąt $\alpha^{n,k}$ zdefiniowany we wzorze (5.5.5), dzięki czemu zestaw P_d^t zawiera tylko i wyłącznie sztuczne pomiary widoczne z perspektywy wirtualnego czujnika	84
5.9	Sztuczny zbiór wykryć wyznaczony przy użyciu wzoru (5.5.7). Jak widać, zestaw P_d^t zawiera tylko i wyłącznie pomiary znajdujące się w polu widzenia wirtualnego czujnika	85
5.10	Przykładowa wizualizacja efektu przysłoneń	87
5.11	Sztuczny zbiór wykryć wygenerowany przy użyciu wzoru (5.5.10), gdzie uwzględniono procedurę obsługi przysłoneń. Jak widać, zasłonięte pomiary nie są brane pod uwagę	88
5.12	Przykładowy przepływ informacji w modelu symulacyjnym czujnika radarowego kompatybilnym ze standardem ASAM OSI	91
5.13	Obiekty klasy <i>OSI::MovingObject</i> zawarte w wiadomości <i>OSI::SensorView</i> wygenerowanej dla pewnego ustalonego momentu czasu nagranych scenariusza jazdy	92
5.14	Uproszczony schemat mocowania czujników radarowych na samochodzie testowym	93
5.15	Przepływ danych przy użyciu czterech instancji modelu sensora	94
5.16	Dane z rzeczywistych czujników radarowych. Na rysunku 5.16a pokazano wykrycia dla pojedynczego cyklu pomiarowego. Z kolei na rysunku 5.16b widać wykrycia zebrane z pięciu kolejnych okresów czasowych	96
5.17	Dane uzyskane na podstawie czterech instancji modelu sensora. Na rysunku 5.17a pokazano sztuczne wykrycia dla pojedynczego cyklu pomiarowego. Z kolei na rysunku 5.17b widać syntetyczne pomiary zebrane z pięciu kolejnych okresów czasowych	97

- 6.1 Zbiory sztucznych wykryć wygenerowane dla pojedynczego cyklu pomiarowego przez cztery instancje modelu sensora dla niskich wartości wzmocnienia testu statystycznego . 100
- 6.2 Zbiory sztucznych wykryć wygenerowane dla pojedynczego cyklu pomiarowego przez cztery instancje modelu sensora dla wysokich wartości wzmocnienia testu statystycznego 101
- 6.3 Porównanie zbioru sztucznych wykryć skojarzonych z pojedynczym obiektem utworzonych dla pojedynczego cyklu pomiarowego dla dwóch ustawień parametrów testu statystycznego. Model sensora wygenerował 36 wykryć dla przypadku po lewej stronie. Natomiast przy F_n ustawionym na -16 [dB] liczebność zbioru zmniejszyła się do 25 punktów 102
- 6.4 Zbiór sztucznych wykryć $\hat{P}_d^t$ (uwzględniający efekt filtracji rozdzielczości) utworzony dla pojedynczego cyklu pomiarowego. Dane syntetyczne skojarzone są z tym samym obiektem, który wykorzystano do wygenerowania wykryć pokazanych na rysunku 6.3. W tym przypadku parametr F_n ustawiono na -16 [dB] 104
- 6.5 Zbiory sztucznych wykryć uzyskane na podstawie czterech instancji modelu sensora, bazujących na definicji określonej według formuły (6.1.1). Jest to ten sam moment czasu co na rysunku 5.16a, gdzie pokazano wykrycia z rzeczywistych czujników radarowych . 104
- 6.6 Schemat poglądowy prezentujący w uproszczony sposób fragment struktury *OSI::SensorView* zawierający N_M obiektów klasy *OSI::MovingObject* oraz N_S obiektów klasy *OSI::StationaryObject* ($N_M, N_S \in \mathbb{N}$), wymagany do wygenerowania przez model sensora poprawnej wiadomości typu *OSI::SensorData* mieszczącej sztuczny zbiór wykryć radarowych 106
- 6.7 Przykładowy przepływ informacji w zmodyfikowanym modelu symulacyjnym czujnika radarowego kompatybilnym ze standardem ASAM OSI, wspierającym generację zbioru sztucznych wykryć zarówno dla obiektów klasy *OSI::MovingObject*, jak i dla obiektów typu *OSI::StationaryObject* 107
- 6.8 Przepływ danych przy użyciu czterech instancji zmodyfikowanego modelu sensora, wytwarzającego syntetyczne wykrycia również dla obiektów typu *OSI::StationaryObject* 108
- 6.9 Rzeczywiste pomiary radarowe wraz ze zbiorem *OSI::GroundTruth* rozszerzonym o obiekty typu *OSI::StationaryObject*. Dane odnoszą się do pojedynczego cyklu pomiarowego pokazanego na rysunku 5.16a 109
- 6.10 Zbiory sztucznych wykryć radarowych uzyskane przy użyciu formuły (6.2.4) wraz ze zbiorem *OSI::GroundTruth* rozszerzonym o obiekty typu *OSI::StationaryObject*. Dane odnoszą się do pojedynczego cyklu pomiarowego pokazanego na rysunku 5.16a 109

6.11	Zbiory sztucznych wykryć radarowych uzyskane przy użyciu formuły (6.2.4) wraz ze zbiorem $OSI::GroundTruth$ rozszerzonym o obiekty typu $OSI::StationaryObject$, powiązane zarówno z rzeczywistymi elementami scenariusza jazdy, jak i z fikcyjnymi celami. Dane odnoszą się do pojedynczego cyklu pomiarowego pokazanego na rysunku 5.16a	111
6.12	Dane z rzeczywistych czujników radarowych zebrane z pięciu kolejnych okresów czasu .	112
6.13	Dane uzyskane na podstawie czterech instancji rozszerzonego modelu sensora, a więc uzyskane na podstawie równania (6.2.4), z uwzględnieniem wykryć dla obiektów fikcyjnych, zebrane z pięciu kolejnych okresów czasu	112
7.1	Zmienność wskaźnika jakości dla siatki wartości ΔR , $\Delta\theta$ i F_n zdefiniowanej w równaniach (7.5.1a) - (7.5.1c). Pasek koloru widoczny na wykresie reprezentuje wszystkie wartości liczby $\mathbb{W}$, jakie uzyskano dla zbioru $\mathbb{I} \times \mathbb{J} \times \mathbb{K}$	120
7.2	Zmienność wskaźnika jakości dla optymalnej wartości F_n oraz dla przyjętej siatki ΔR i $\Delta\theta$. Jest to zatem przekrój poprzeczny sześcianu pokazanego na rysunku 7.1. Oś pionowa to w tym przypadku odległość Wassersteina wyliczona dla pojedynczego cyklu optymalizacji, czyli zgodnie ze wzorem (7.4.1)	121
7.3	Mapy ciepła wygenerowane dla czterech typów danych zebranych ze 100 kolejnych okresów odczytu - rzeczywisty zbiór wykryć (lewy górny róg), sztuczny zbiór wykryć uzyskany dla nastaw zwróconych przez algorytm Powella (prawy górny róg) oraz syntetyczny zestaw pomiarów uzyskany dla nastaw odpowiadających kolejno dolnemu i górnemu zakresowi ograniczeń zbioru $\mathbb{I} \times \mathbb{J} \times \mathbb{K}$ (odpowiednio lewy dolny róg oraz prawy dolny róg)	122
8.1	Wirtualny scenariusz zaprojektowany na potrzeby oceny wydajności modelu sensora . .	127
8.2	Rzeczywista lista brył brzegowych zwrócona przez algorytm wykrywania obiektów na podstawie pomiarów z czujników radarowych typu FMCW	130
8.3	Sztuczna lista brył brzegowych oszacowana przez algorytm wykrywania obiektów na podstawie danych z czterech instancji modelu sensora	130
8.4	Schemat blokowy ukazujący uproszczony przepływ danych w procedurze weryfikacji wiarygodności modelu sensora	132
8.5	Przykładowy podzbiór skojarzonych ze sobą danych	133
8.6	Wartość współczynnika Jaccarda dla trzech par obiektów	134
8.7	Lista brył brzegowych oszacowana przez algorytm wykrywania obiektów dla dyskretnego momentu czasu $\mathbb{A}$, na podstawie rzeczywistych pomiarów radarowych . . .	140

- 8.8 Lista brył brzegowych oszacowana przez algorytm wykrywania obiektów dla dyskretnego momentu czasu $\mathbb{A}$, na podstawie sztucznych pomiarów wygenerowanych przez model sensora skonfigurowany przy użyciu nastaw zawartych odpowiednio w: zestawie 2 (lewy górny róg), zestawie 1 (ustawienia optymalne, prawy górny róg), zestawie 3 (lewy dolny róg), zestawie 4 (oryginalny model sensora, prawy dolny róg) . . 141
- 8.9 Lista brył brzegowych oszacowana przez algorytm wykrywania obiektów dla dyskretnego momentu czasu $\mathbb{B}$, na podstawie rzeczywistych pomiarów radarowych . . . 141
- 8.10 Lista brył brzegowych oszacowana przez algorytm wykrywania obiektów dla dyskretnego momentu czasu $\mathbb{B}$, na podstawie sztucznych pomiarów wygenerowanych przez model sensora skonfigurowany przy użyciu nastaw zawartych odpowiednio w: zestawie 2 (lewy górny róg), zestawie 1 (ustawienia optymalne, prawy górny róg), zestawie 3 (lewy dolny róg), zestawie 4 (oryginalny model sensora, prawy dolny róg) . . 142
- 8.11 Lista brył brzegowych oszacowana przez algorytm wykrywania obiektów dla dyskretnego momentu czasu $\mathbb{C}$, na podstawie rzeczywistych pomiarów radarowych . . . 142
- 8.12 Lista brył brzegowych oszacowana przez algorytm wykrywania obiektów dla dyskretnego momentu czasu $\mathbb{C}$, na podstawie sztucznych pomiarów wygenerowanych przez model sensora skonfigurowany przy użyciu nastaw zawartych odpowiednio w: zestawie 2 (lewy górny róg), zestawie 1 (ustawienia optymalne, prawy górny róg), zestawie 3 (lewy dolny róg), zestawie 4 (oryginalny model sensora, prawy dolny róg) . . 143

Wykaz tabel

3.1	Odpowiedniki obiektów zdefiniowanych w zbiorze <i>OSI::GroundTruth</i> będące częścią struktury <i>OSI::SensorData</i>	57
7.1	Niezmiennie parametry modelu sensora	114
7.2	Konfigurowalne parametry modelu sensora	114
7.3	Optymalne wartości nastaw modelu sensora zwrócone przez algorytm Powella dla różnych punktów startowych wraz z wartością wskaźnika jakości $\mathbb{W}$	119
8.1	Ocena wydajności modelu sensora na procesorze Intel Core i7-9750H	128
8.2	Ocena wydajności modelu sensora na procesorze Intel Core i5-7200U	128
8.3	Cztery zestawy nastaw wykorzystane na potrzeby oceny wiarygodności modelu sensora .	137
8.4	Wartość wskaźnika $\mathbb{W}_A$ wraz z całkowitą liczbą skutecznych asocjacji dla czterech różnych konfiguracji modelu sensora	138
8.5	Wartość wskaźnika $\mathbb{W}_B$ dla czterech różnych konfiguracji modelu sensora	139