

mgr inż. Jakub Bartoszek, dr inż. Maciej Trojnacki, mgr inż. Piotr Bigaj
Przemysłowy Instytut Automatyki i Pomiarów w Warszawie

ANALIZA WRAŻLIWOŚCI ALGORYTMU SEMIAUTONOMII ROBOTA MOBILNEGO NA AWARIĘ CZUJNIKÓW OTOCZENIA – BADANIA SYMULACYJNE

Praca dotyczy analizy wrażliwości algorytmu semiautonomii mobilnego robota bojowego na awarię czujników otoczenia. Opisano w niej konstrukcję robota i zastosowane na nim czujniki. Przedstawiono algorytm służący do realizacji trybu semiautonomii i uwzględniający awarie czujników otoczenia. Zaprezentowano wyniki badań symulacyjnych algorytmu semiautonomii w środowisku MATLAB/Simulink. Badania te wykonano dla przypadków normalnej pracy czujników otoczenia oraz awarii wybranych z nich. Na tej podstawie dokonano oceny wrażliwości algorytmu na awarię wybranych czujników otoczenia.

SENSITIVITY ANALYSIS OF SEMIAUTONOMY ALGORITHM OF MOBILE ROBOT TO ENVIRONMENTAL SENSORS FAILURES – SIMULATION RESEARCH

This work is concerned on sensitivity analysis of semiautonomy algorithm of mobile combat robot to environmental sensors' damage. The construction of the robot, semiautonomy algorithm and used sensors have been described. This algorithm takes into account environmental sensors' damage. Simulation research results of semiautonomy algorithm using MATLAB/Simulink package was presented. This research was performed for normal environmental sensors' operation and for selected sensors' damage. On that basis, sensitivity of semiautonomy algorithm to selected environmental sensors damage was tested.

1. WSTĘP

Niezwykle duże środki finansowe i praca wielu ośrodków badawczych rozmieszczonych w różnych rejonach globu skupione są na opracowywaniu coraz doskonalszych algorytmów sterowania robotami mobilnymi. Prace te w dużej mierze mają na celu osiągnięcie stanu rozwoju robotyki mobilnej określanego mianem pełnej autonomii. Chwila, w której cel ten zostanie osiągnięty będzie miała niezwykle istotny wpływ na kierunek i szybkość dalszego rozwoju naszej cywilizacji. Osiągnięcie stanu pełnej autonomii pozwoli wielokrotnie zwiększyć funkcjonalność i zarazem liczbę możliwych zastosowań pojazdów autonomicznych. Najważniejszą cechą tych pojazdów będzie brak konieczności sterowania nimi przez człowieka oraz ograniczenie jego nadzoru.

Jednym z najczęściej poruszanych zagadnień w badaniach nad autonomią robotów mobilnych jest problem ich sterowania w dynamicznym, silnie zróznicowanym i nieznanym środowisku [1]. Zagadnienie to w dużym uproszczeniu można sprowadzić do wyznaczenia trajektorii ruchu robota od pozycji aktualnej do wyznaczonego celu w sposób bezkolizyjny i pozwalający osiągnąć zadane położenie w jak najkrótszym czasie [2]. Algorytmy pozwalające wybrać optymalną trasę ze względu na różne wskaźniki jakości są rozwijane przez różne ośrodki naukowe na całym świecie. Każdy z algorytmów korzysta z co najmniej dwóch rodzajów czujników – stanu i otoczenia.

Do czujników stanu należy zaliczyć wszystkie czujniki pozwalające określić aktualny stan robota. W najczęstszym przypadku umożliwiają one określenie jedynie jego położenia w przestrzeni. Metody wyznaczania położenia w zależności od dostępnych danych można podzielić na kilka odrębnych grup. Pierwsza z nich związana z zastosowaniem odometrii wiąże się z nieraz szybko narastającym błędem wraz z wykonywaniem kolejnych pętli programu sterownika. Druga polega na znajomości statycznej mapy terenu i określeniu pozycji na podstawie znaczników rozmieszczonych na trasie robota lub określeniu punktów charakterystycznych za pomocą np. systemu wizyjnego. Trzecia grupa bazuje na GPS lub pochodnych systemach. Każda ze wspomnianych metod obciążona jest pewną niepewnością i związana jest z koniecznością zabezpieczenia programu przed wpływem niepoprawnych danych. Dane te mogą być skutkiem awarii jednego z czujników lub, w przypadku takich systemów, jak GPS, utraty sygnału z satelity.

Czujniki otoczenia to grupa czujników pozwalających stworzyć wirtualny model środowiska na bazie środowiska rzeczywistego, w którym aktualnie znajduje się robot. W zależności od liczby czujników i ich rodzaju model otoczenia może być dwu lub trójwymiarowy. Z kolei ze względu na moc obliczeniową jednostki centralnej robota model ten może przyjmować różne formy. Podejście, w którym budowano reprezentację modelu otoczenia robota zastosowano w publikacjach [3, 4], a pominięto i wyznaczano sterowanie jedynie na podstawie sygnałów z czujników, np. w [5]. Główną cechą charakterystyczną tej grupy czujników jest ich ograniczony zakres. Czujniki te, gdy zastosowane są w robotach mobilnych przeznaczonych do pracy w środowisku zewnętrznym, dodatkowo mają jeszcze jedną istotną cechę wspólną – narażone są na różnorodne awarie. Mogą one być spowodowane czynnikami zewnętrznymi, takimi jak zderzenia z przeszkodami czy wibracje powstałe w skutek ruchu robota lub czynnikami wewnętrznymi, np. przegrzaniem się elementów elektronicznych.

Awarie czujników są w większości przypadków niezwykle trudne do przewidzenia. Ich charakter często można określić dopiero w chwili, gdy już wystąpiły i konieczne jest ich poprawne zdiagnozowanie. Dlatego też sposób diagnozowania i reagowania na awarie czujników powinien zostać uwzględniony już we wstępnej fazie projektowania całego systemu. W literaturze wymieniane są dwa podejścia do tego problemu [6–8]. Jedno z nich opiera się na określeniu modelu pozwalającego wykryć awarie czujnika [9–12]. Niektóre z nich wymagają ingerowania w strukturę systemu ze względu na wykrytą awarię [13].

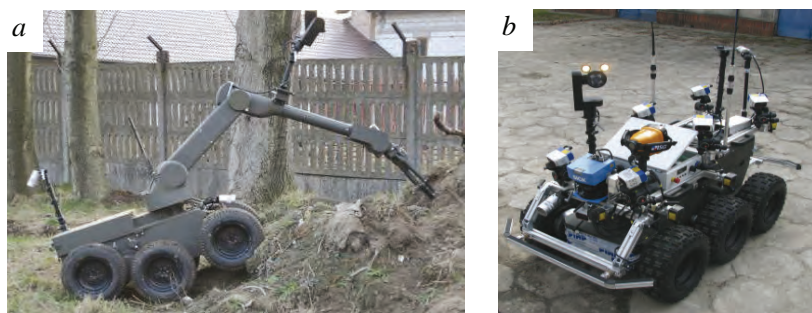
Zupełnie inne podejście do problemu detekcji i reagowania na awarie czujników przedstawił Martin Soika w [6]. Celem jego pracy było opracowanie metody, która będzie umożliwiała wykrycie wcześniej nieuwzględnionej awarii czujnika i prawidłową reakcję na jej wystąpienie. Metoda przez niego opracowana bazowała na określaniu stopnia pewności danych przesyłanych przez czujnik. Healey [14] stworzył system bazujący na modelu awarii czujników jednakże wykorzystujący sztuczną sieć neuronową do wykrywania samych awarii. Takie rozwiązanie pozwala na implementację metody w sterowniku robota w sposób nieograniczający tym samym jego autonomię, niestety znacznie zwiększa złożoność obliczeniową algorytmu.

Ważnym czynnikiem pozwalającym automatycznie reagować i diagnozować uszkodzenia czujników jest ich redundancja. Model otoczenia jest w takim przypadku tworzony na podstawie danych z większej liczby czujników, co pozwala w umożliwić łatwy sposób wykryć potencjalną awarię jednego z nich. Większa liczba czujników, często korzystających z różnych metod pomiaru, wymaga zastosowania bardziej wyszukanych i lepiej zoptymalizowanych algorytmów budowania modelu otoczenia robota. To z kolei pociąga za sobą konieczność zastosowania w sterowniku robota jednostek obliczeniowych o dużej wydajności. Z tego powodu w niektórych przypadkach konieczna staje się rezygnacja z redundancji czujników.

W ramach niniejszej publikacji zaprezentowane zostały metody reagowania na awarie czujników zastosowane w algorytmie sterowania robotem IBIS. Algorytm ten został szerzej opisany w [15].

2. MOBILNY ROBOT BOJOWY

IBIS w wersji komercyjnej (rys. 1a) jest robotem do zastosowań pirotechnicznych i bojowych. Przystosowany został do operacji w trudnym i zróżnicowanym terenie, takim jak piach, śnieg czy podłoże skalne. Dzięki prężności robota umożliwia dynamizację przeprowadzania akcji. Manipulator zapewnia duży zasięg działania, a zastosowane rozwiązanie napędu zapewnia płynność ruchu w pełnym zakresie prędkości.



Rys. 1. Robot IBIS: a – wersja komercyjna z manipulatorem, b – wersja do badań autonomii

Podstawowe dane techniczne robota są następujące: masa 295 kg, wymiary (długość × szerokość × wysokość) 1,3 × 0,85 × 0,95 m, prędkość maksymalna 10 km/h, udźwignię manipulatora 30 kg, maksymalny zasięg manipulatora 3,15 m [16, 17].

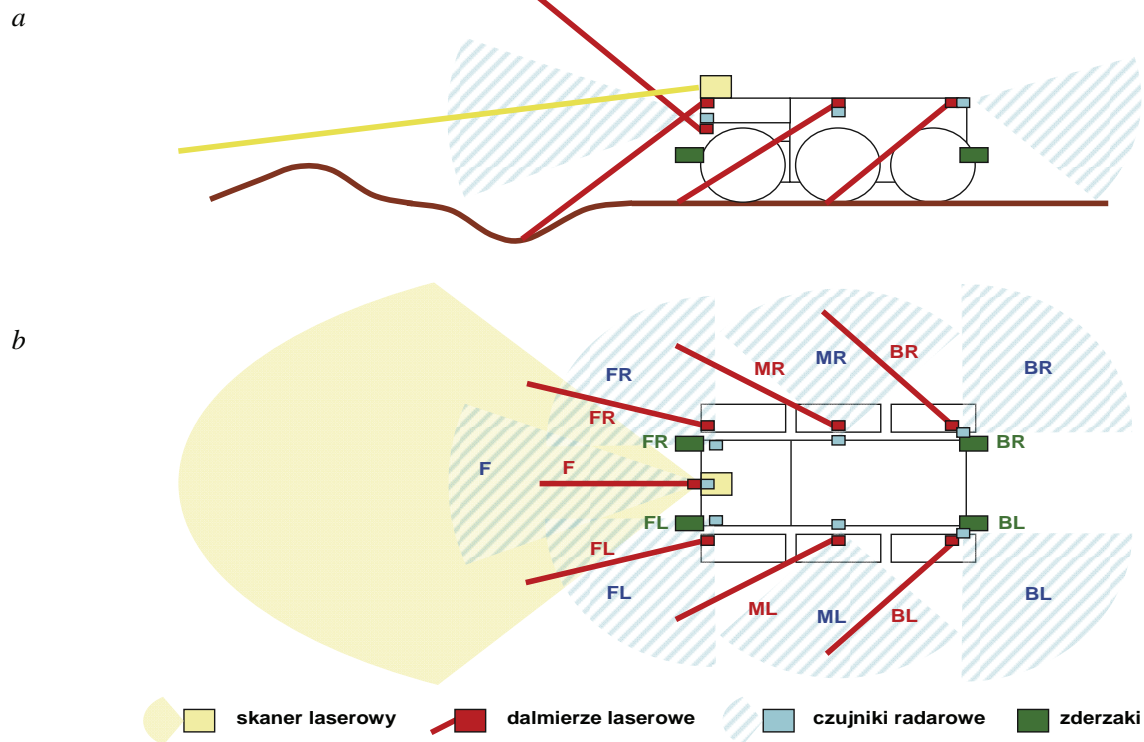
Do celów badawczych została opracowana wersja robota IBIS bez manipulatora (rys. 1b). Robot w tej wersji wyposażony został w dodatkowy nakładany stelaż. Stanowi on modyfikację robota przeznaczony do realizacji trybu semiautonomii. Moduł ten zawiera układy mikroprocesorowe, zestawy czterech kamer oraz czujników lokalizacyjnych i wykrywających przeszkody. Robot może poruszać się w dwóch trybach, tj. teleoperacji – gdy jest sterowany przez operatora, semiautonomii – kiedy samodzielnie realizuje zadane zachowanie pod kontrolą operatora.

3. CZUJNIKI ROBOTA

Baza mobilna robota nadbudowana została specjalnie zaprojektowaną ramą zapewniającą dokładne pozycjonowanie czujników oraz kontrolerów: nawigacyjnego i pozycyjnego. Architektura fizyczna całości systemu składa się z czterech bloków rozdzielonych funkcjonalnie i fizycznie: czujników, kontrolera pozycyjnego (określającego aktualne położenie robota), kontrolera nawigacyjnego (przetwarzającego dane wyjściowe i wystawiającego sterowanie) oraz sterowników silników. Sterowanie ruchem w trybie teleoperacji jest przekazywane do robota za pośrednictwem modemu ISM, natomiast ruch w trybie semiautonomii jest w całości obliczany przez kontroler nawigacyjny.

Czujniki lokalizacyjne umieszczone są na kontrolerze pozycyjnym i służą do określenia pozycji i pełnej orientacji robota w przestrzeni. Pozycja określana jest w trzech wymiarach: długość, szerokość geograficzna i wysokość nad geoidą Ziemi zgodnie ze specyfikacją NMEA, dzięki czemu może być przedstawiona w dowolnym oprogramowaniu GIS. Do wyznaczenia aktualnej pozycji robota użyto jednofazowego odbiornika GPS wspomaganego przez INS. Określenie współrzędnych robota w systemie WGS-84 uzyskiwane jest za

pomocą filtracji Kalmana pozycji GPS oraz nawigacji i inercyjnej. Azymut robota ustalany jest za pomocą cyfrowego kompasu z kompensacją przechyłu. Do określenia przechylenia i pochylenia robota stosowane są czujniki inklinacji: inklinometry oraz akcelerometry.



Rys. 2. Rozmieszczenie czujników: a – widok z lewej strony robota, b – widok z góry

Na ramie nośnej robota zamontowano cztery rodzaje czujników do detekcji przeszkód: dwuwymiarowy skaner laserowy, dalmierze laserowe, czujnik radarowe o obecności przeszkody i czujniki taktylne (zderzaki). Czujniki te są rozmieszczone w taki sposób (rys. 2), aby pokrywały maksymalną przestrzeń wokół robota i aby większa część informacji z czujników przypadła na przestrzeń przed robotem.

Zadaniem dwuwymiarowego skanera laserowego jest wykrywanie przeszkód znajdujących się przed robotem. Zakres kątowy jego pracy ustawiono na 100° . Jego maksymalny zasięg 80 m wynika z ograniczeń sprzętowych samego czujnika, natomiast aktualny zakres widzenia zależy od ustawienia kąta, pod którym kierowana jest wiązka. Skaner skierowany jest pod niewielkim kątem w dół, co pozwala wykryć przeszkody o wysokości mniejszej niż wysokość na jakiej został on umieszczony.

Dalmierze laserowe służą do wykrywania przeszkód w najbliższej okolicy robota. Są one umieszczone pod różnymi kątami w dół, co pozwala na detekcję zarówno wypukłych, jak i wklęsłych przeszkód. Jeden z nich ma wiązkę skierowaną w górę, co pozwala z kolei na wykrycie zbyt niskich przeszkód. Zastosowane dalmierze umożliwiają pomiar odległości do najbliższej przeszkody w zakresie 0,5–10 m.

Czujniki radarowe są przeznaczone do wykrycia przeszkód w dużej odległości od robota. Zwracają one tylko informację, czy coś znajduje się w zdefiniowanej odległości. Zakres ich działania obejmuje odległości 2–15 m. Umożliwiają one wczesne wykrycie przeszkód znajdujących się w okolicy robota.

Zadaniem czujników taktylnych (zderzaków) jest wykrycie przeszkód, które nie zostały wykryte przez pozostałe czujniki i spowodowanie awaryjnego zatrzymania robota w przypadku uderzenia w przeszkodę.

4. ALGORYTM SEMIAUTONOMII

Żaden z dotychczas zaprojektowanych algorytmów nie zapewnia robotom mobilnym bezbłędnego wykonywania powierzonych im zadań. Wynika to m.in. z faktu, że twórcy algorytmów nie potrafią przewidzieć wszystkich możliwych sytuacji, a czujniki nie są w stanie zawsze wykryć i sklasyfikować wszystkie przeszkody. Z tych powodów w przeprowadzanych badaniach korzysta się z pojęcia semiautonomii. Oznacza to, że robot wykonuje swoje zadanie samodzielnie, ale pod kontrolą operatora, który może w dowolnej chwili przerwać realizację zadania.

Celem badań opisanych w niniejszej pracy było zbadanie wpływu awarii czujników na działanie algorytmu umożliwiającego robotowi IBIS osiągnięcie zadanego celu, z omijaniem przy tym wykrytych ze pomocą czujników przeszkód.

Algorytm sterowania składa się z czterech podzachowań. Pierwsze podzachowanie związane jest z ustawianiem się robota w kierunku celu. Drugie podzachowanie związane jest z ustaleniem prędkości ruchu robota i zależy od odległości robota od celu i najbliższej przeszkody. Jeżeli robot znajduje się w dużej odległości od celu, a najbliższa przeszkoda znajduje się w odległości większej niż zasięg czujników, wówczas robot porusza się z maksymalną prędkością. Trzecie podzachowanie związane jest ze skanerem laserowym, który traktowany jest jak 101 pojedynczych wiązek. Do przetworzenia pozyskiwanych ze skanera informacji o otoczeniu wykorzystuje się zmodyfikowany algorytm VFH [18, 19]. Czwarte podzachowanie wzorowane jest na idei algorytmu Braitenberga, który opiera swoje działanie na bezpośrednim połączeniu czujników z elementami wykonawczymi, przy czym każda połączenie ma osobną wagę. W zależności od wskazań czujników i wartości wagi robot może wykonywać różne zadania. Algorytm semiautonomii z podziałem na jego poszczególne podzachowania został szerzej opisany w [15].

Wagi dla każdego z podzachowań ustalane są na podstawie aktywnego zachowania. Jeżeli aktualne zachowanie jest za mało efektywne wirtualny zbiornik (WaterTank) przepełnia się i następuje przełączenie do innego zachowania. O tym, jakie jest nowe zachowanie decyduje jakościowa ocena wzajemnego położenia względem robota przeszkód i najbliższego celu. Na przykład, jeżeli w momencie przerwania zachowania „podażaj do celu i omijaj przeszkody” przed robotem są przeszkody, a cel znajduje się prosto i na prawo, wówczas robot może przełączyć się na realizację zachowania „śledź przeszkody po lewej stronie”. Przy zmianie zachowania wirtualny zbiornik jest na początek opróżniany.

Jedną z najważniejszych cech charakteryzującą współczesne algorytmy sterowania jest odporność na awarie czujników. Czujniki, które uległy awarii, mogą powodować znaczne zmiany zachowania robota nieraz prowadzące do jego uszkodzenia lub stwarzające zagrożenie dla osób znajdujących się w jego otoczeniu. W celu uniknięcia takich sytuacji prezentowany algorytm został zmodyfikowany pod kątem możliwości wykrycia w sterowniku niepoprawnych danych przesyłanych przez uszkodzone czujniki.

Czujnik lokalizacyjny w postaci odbiornika GPS w przypadku utracenia sygnałów z nadajników lub wystąpienia dużych różnic w wyznaczanych współrzędnych położenia robo-

ta (powyżej 10 m) powoduje przełączenie się trybu pracy robota z semiautonomii na teleoperację. Czynniki decydującymi o poprawnym działaniu czujnika lokalizacyjnego są różnice w położeniu robota oraz jego odległości od celu w kolejnych iteracjach programu sterownika.

Czujniki otoczenia znajdujące się na robocie to m.in. siedem dalmierzy laserowych i skaner 2D. Awaria tych czujników może przejawiać się w zależności od jej charakteru w różny sposób.

Pierwszy typ awarii występuje gdy niemożliwe jest nawiązanie komunikacji z czujnikiem lub gdy sam czujnik po zainicjowaniu przesyła informacje o błędzie. Mikroprocesor, odpowiedzialny za zbieranie informacji o czujnikach i komunikację z nimi, przesyła w takim przypadku do mikroprocesora głównego informację o awarii czujnika. Informacja ta przesyłana jest w taki sam sposób, jak poprawna wartość zwracana przez czujnik, ale przyjmuje ona wartość 0,01 m. Odpowiada to sytuacji, w której przeszkoda znajduje się w takiej właśnie odległości od czujnika i jest ona w normalnych warunkach pracy niemożliwa do otrzymania.

Drugi typ awarii jest bardziej skomplikowany do wykrycia. Występuje, gdy z jakiegoś powodu czujnik zwraca wartości niemające odzwierciedlenia w jego rzeczywistym otoczeniu. Awaria tego typu może być spowodowana wieloma różnymi czynnikami, takimi jak np. duże wibracje platformy mobilnej występujące podczas szybkiej jazdy po nierównym terenie czy odbicia wiązki czujników laserowych od płaskich powierzchni skierowanych pod małym kątem. Wykrycie i odfiltrowanie błędnych danych w takich przypadkach jest niezwykle trudne. W algorytmie sterowania jedynie wartości wychodzące poza zakres pomiarowy wynikające z konstrukcji robota mobilnego oraz sposobu zamontowania czujników zostają odfiltrowane.

Wykrycie i prawidłowe zdiagnozowanie awarii powoduje zmianę wag dla odpowiedniego podzachowania. W przypadku wykrycia awarii skanera waga dla podzachowania trzeciego zostaje ustawiona na zero, co w rezultacie równoznaczne jest z wyłączeniem jego wpływu na działanie algorytmu i samego robota. W przypadku awarii jednego lub kilku dalmierzy zmieniana jest waga dla podzachowania czwartego. Waga ta jest modyfikowana na podstawie współczynnika wiarygodności, którego sposób wyznaczania przedstawiony został poniżej.

Sterowania dla kół robota obliczane są z zależności:

$$u_i = \sum_{j=1}^4 c_j u_{ij}, \quad (1)$$

gdzie: $i = \{L, R\}$, L i R – odpowiednio lewe i prawe koła robota, j – numer podzachowania, c_j – waga dla podzachowania j definiująca udział sterowania związanego z tym podzachowaniem w sterowaniu całkowitym, u_{ij} – sterowanie dla koła i oraz podzachowania j .

Wagi związane z poszczególnymi podzachowaniami wynoszą odpowiednio: $c_1 = 1$ (nie analizuje się awarii czujnika związanego z określeniem orientacji robota), $c_2 = 1$ (nie analizuje się awarii czujnika związanego z określeniem prędkości robota), $c_3 = 0$ w przypadku awarii skanera laserowego, $c_3 = c_3^* = 0,6$ jeżeli skaner jest sprawny, natomiast waga dla ostatniego podzachowania obliczana jest ze wzoru:

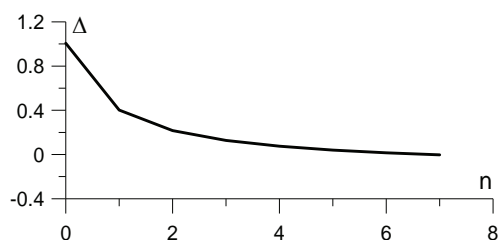
$$c_4 = c_4^* \Delta \quad (2)$$

gdzie: $c_4^* = 0,4$ oznacza wagę dla podzachowania związanego z dalmierzami laserowymi jeżeli wszystkie dalmierze są sprawne, Δ – współczynnik wiarygodności informacji pochodzących z dalmierzy laserowych zależny od liczby uszkodzonych dalmierzy.

Należy zwrócić uwagę, że w przypadku awarii któregoś z czujników zachodzi zależność:

$$c_3 + c_4 < 1. \quad (3)$$

W celu zapewnienia większego bezpieczeństwa pracy robota przyjęto podejście, w którym awaria czujnika otoczenia powoduje ograniczenie maksymalnej prędkości ruchu robota. Przy innym podejściu można by było zwiększać wagę dla tego podzachowania, dla którego czujniki są sprawne, tak, aby suma c_3 i c_4 była równa 1.



Rys. 3. Zmiana współczynnika wiarygodności w zależności od liczby dalmierzy laserowych, które uległy awarii

Współczynnik wiarygodności wyznaczany jest z zależności:

$$\Delta = \frac{1}{n + 0,88} - 0,13, \quad (4)$$

gdzie: n oznacza liczbę dalmierzy laserowych, które uległy awarii.

Zmiana wartości tego współczynnika w zależności od liczby dalmierzy laserowych, które uległy awarii została zilustrowana na rys. 3.

5. BADANIA SYMULACYJNE

W celu przeprowadzenia badań symulacyjnych opracowywanych metod autonomii przygotowano środowisko symulacyjne oparte na pakiecie Matlab/Simulink. Oprogramowanie odpowiedzialne za symulację środowiska i wyznaczanie wskazań wirtualnych czujników zostało oddzielone od oprogramowania realizującego algorytm sterowania robota w trybie teleoperacji i semiautonomii. To drugie oprogramowanie zostało przygotowane w taki sposób, aby była możliwość jego łatwej i szybkiej implementacji na mikrokontrolerze robota. Otoczenie robota do symulacji zostało przygotowane z zastosowaniem programu V-Realm Builder. W oprogramowaniu symulacyjnym poszczególne obiekty sceny zostały przybliżone za pomocą prostopadłościanów i walców.

W niniejszej pracy zaprezentowano wyniki badań dla przypadku, w którym na drodze robota do celu występowały cztery przeszkody. Robot realizował złożone zachowanie „podążaj do celu i omijaj przeszkody”. W ramach badań symulacyjnych dokonano oceny wrażliwości algorytmu semiautonomii dla mobilnego robota bojowego na awarię wybranych czujników otoczenia.

W celu uzyskania miarodajnej oceny wyników wprowadzono następujące wskaźniki jakości:

- a) suma kwadratów odległości robota od celu

$$E = \sum_{i=1}^n e_i^2 \Delta t, \quad (5)$$

gdzie e_i – odległość robota od celu w i -tej iteracji, n – liczba iteracji do czasu osiągnięcia przez robota celu lub zakończenia symulacji w przypadku nieosiągnięcia celu,

- b) odchylenie standardowe prędkości ruchu robota

$$S = \sqrt{\frac{\sum_{i=1}^n (v_i - \bar{v})^2}{n-1}}, \quad (6)$$

gdzie: v_i – prędkość robota w i -tej iteracji, \bar{v} – średnia prędkość ruchu,

c) długość drogi od pozycji początkowej do celu

$$s = \sum_{i=1}^n v_i \Delta t, \quad (7)$$

d) czas ruchu robota do celu T , zakładając, że cel jest osiągnięty dla $e \leq 0,5$ m,

e) maksymalna prędkość ruchu robota v_{max} (w przedziale czasu od t_0 do T , gdzie t_0 jest chwilą czasu, w której nastąpiła awaria czujnika),

f) średnia prędkość ruchu robota \bar{v} (w przedziale czasu od 0 do T).

Należy zauważyć, że dla wskaźników a–d należy dążyć do ich minimalizacji, natomiast dla wskaźników e i f do ich maksymalizacji.

W przypadku, gdy robotowi nie udało się dotrzeć do celu w założonym czasie T_{max} , zakłada się, że wskaźniki jakości s i T osiągają wartość $+\infty$, natomiast pozostałe wskaźniki osiągają wyliczone wartości dla czasu T_{max} . W niniejszej pracy przyjmuje się $T_{max} = 100$ s.

Symulacja 1 – wszystkie czujniki otoczenia są sprawne

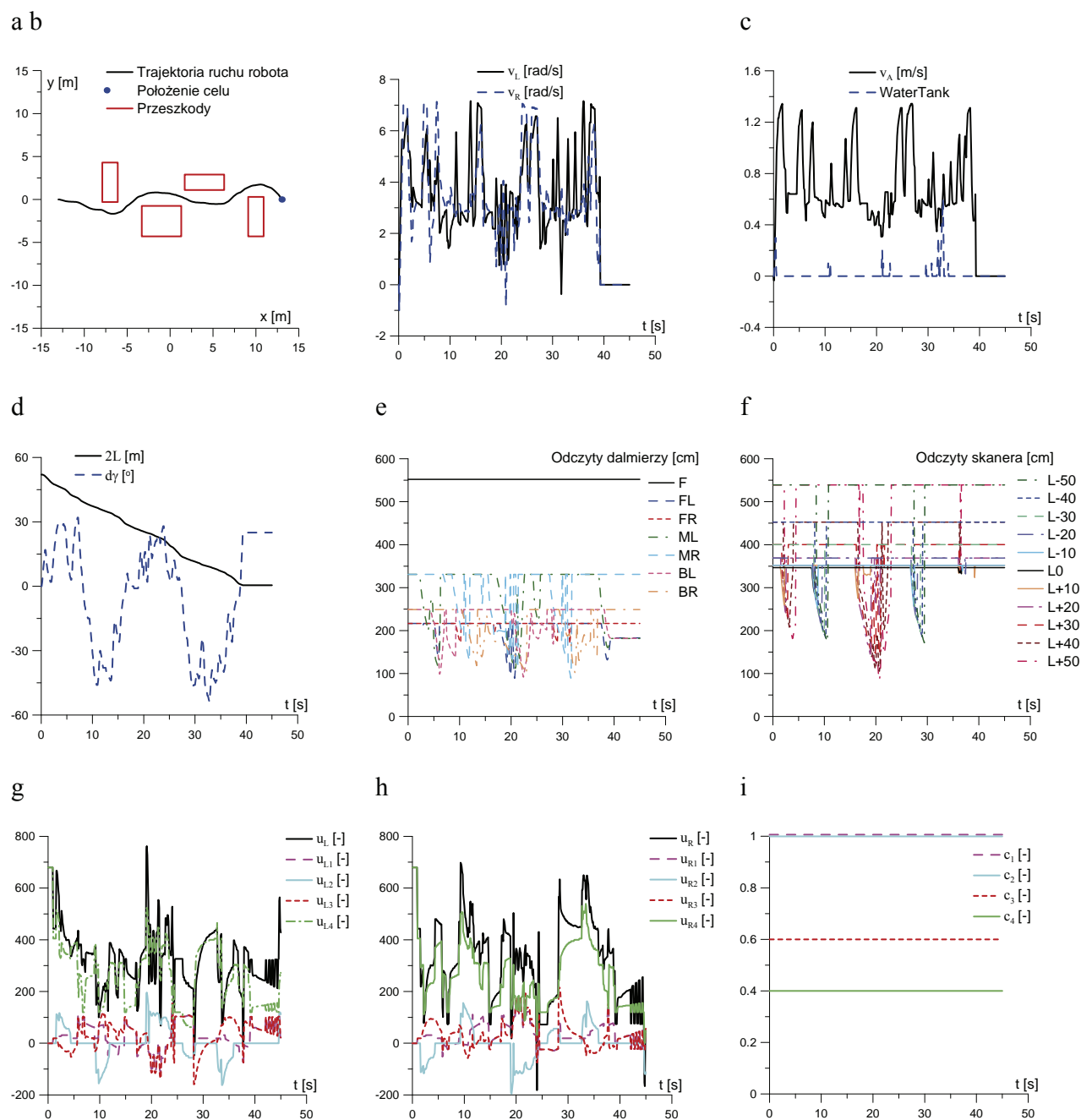
Pierwsza prezentowana symulacja została wykonana dla założenia, że wszystkie czujniki otoczenia umieszczone na robocie są sprawne. Wyniki tej symulacji, przedstawione na rys. 4 i w tab. 1, stanowią punkt odniesienia dla pozostałych symulacji, w których zakłada się awarię wybranych czujników.

Przyjęte na rys. 4e oznaczenia dla mierników laserowych są zgodne z rys. 2. Oznaczając wiązki skanera laserowego na rys. 4f, przyjęto natomiast konwencję, że są one nazywane jako $L\alpha$, gdzie α jest kątem (w stopniach) odchylenia wiązki od osi podłużnej skanera. Kąt dodatni oznacza odchylenie w lewo, a ujemny w prawo. Na rys. 4f pokazano odczyty co dziesiątej wiązki skanera. Na rys. 4g i 4h zamieszczono wartości sterowań dla lewych i prawych kół robota (u_{Li} i u_{Ri} , $i = 1 \dots 4$) dla opisanych wcześniej podzachoń oraz sterowania całkowite (u_L i u_R). W trakcie ruchu robota wartości dla poszczególnych podzachoń robota są stałe (rys. 4i). Wagi te są zmieniane wyłącznie w przypadku wykrycia awarii czujników otoczenia.

Z wykonanej symulacji wynika, że opracowana metoda pozwala na ominięcie przez robota występujących przeszkód i dotarcie do zdefiniowanego celu. Robot porusza się ze zmienną prędkością, co jest związane z tym, że zwalnia tym bardziej im bliżej niego znajdują się przeszkody. Praktycznie cały czas podczas ruchu robota jego odległość od celu ulega zmniejszaniu. W trakcie omijania przeszkód u widocznie się działanie dalmierzy i skanera laserowego, których odczyty ulegają wówczas istotnym zmianom. Efektywność zachowania jest oceniana przy pomocy parametru WatterTank, którego największe wartości występują w chwilach, gdy robot przemieszcza się pomiędzy przeszkodami, co jest widoczne na rys. 4c.

Tab. 1. Uzyskane wartości wskaźników jakości dla Symulacji 1

Wskaźnik jakości	E [m^2/s]	S [m/s]	s [m]	T [s]	v_{max} [m/s]	\bar{v} [m/s]
Wartość	8 533	0,284	28,2	38,8	1,34	0,726



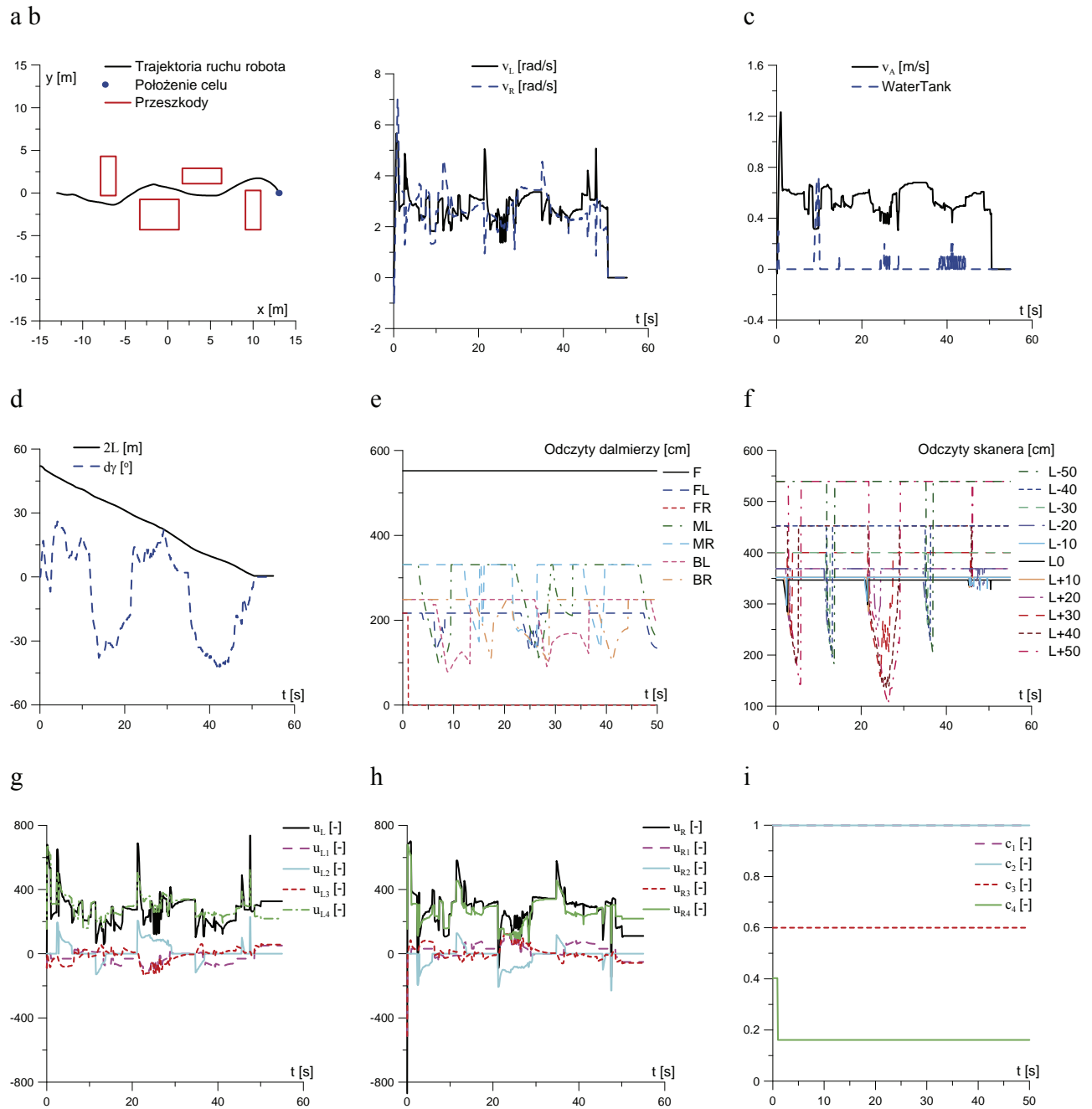
Rys. 4. Wyniki Symulacji 1: a – trajektoria ruchu robota z zaznaczeniem położenia celu i przeszkód, b – zadane prędkości liniowe dla lewych i prawych kół robota, c – prędkość ruchu robota i zmiana parametru WaterTank, d – odległość i odchylenie robota od celu, e – odczyty dalmierzy laserowych, f – odczyty skanera dla wybranych 11. wiązek, g–i – wartości sterowań i wagi dla poszczególnych podzachowań

Symulacja 2 – dalmierz oznaczony jako FR ulega awarii po czasie 1 s

Druga symulacja została przeprowadzona dla przypadku awarii dal mierza laserowego oznaczonego na rys. 2 jako FR. Awaria czujnika nastąpiła po 1 [s] symulacji. Wyniki tej symulacji zilustrowano na rys. 5 i w tab. 2. W efekcie wykrycia przez algorytm semiautonomii awarii dalmierza została zmodyfikowana waga dla pierwszego podzachowania (rys. 5i) związanego ze zmodyfikowanym algorytmem Braitenberga bazującym na wskazaniach dalmierzy laserowych.

Tab. 2. Uzyskane wartości wskaźników jakości dla Symulacji 2

Wskaźnik jakości	E [m ² s]	S [m/s]	s [m]	T [s]	v_{max} [m/s]	\bar{v} [m/s]
Wartość 11	016	0,116	27,7	50,0	0,68 ¹	0,555



Rys. 5. Wyniki Symulacji 2: a – trajektoria ruchu robota z zaznaczeniem położenia celu i przeszkód, b – zadane prędkości liniowe dla lewych i prawych kół robota, c – prędkość ruchu robota i zmiana parametru WaterTank, d – odległość i odchylenie robota od celu, e – odczyty dalmierzy laserowych, f – odczyty skanera dla wybranych 11. wiązek, g-i – wartości sterowań i wagi dla poszczególnych podzachowań

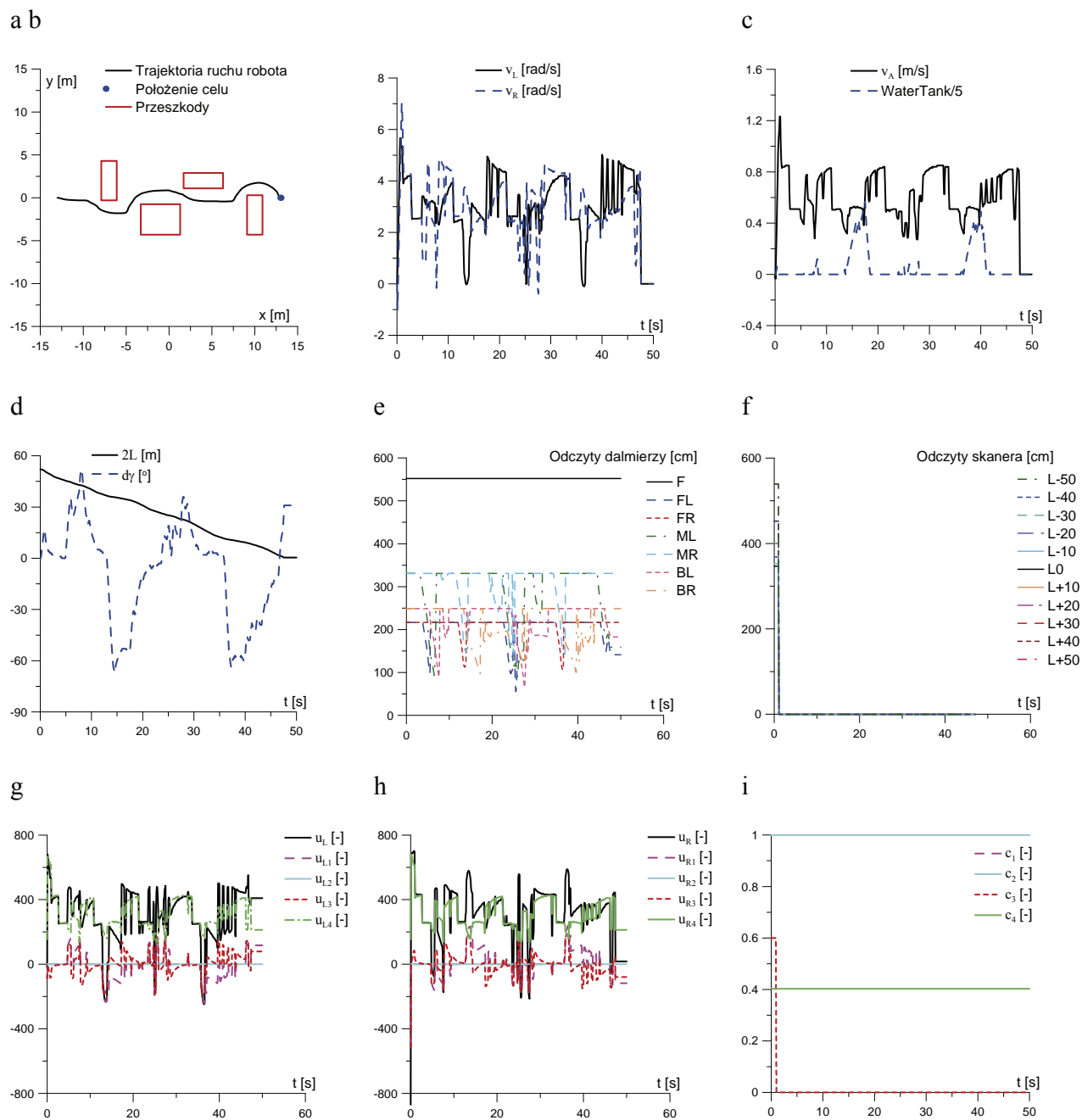
Porównując wyniki Symulacji 1 i 2 można stwierdzić, że w przypadku awarii dalmierza laserowego FR robot osiągnął cel po dłuższym czasie. Odnotowano gorsze wartości wskaźników ja-

¹ Pominięto początkowe maksymalne wartości, które są efektem działania algorytmu przed awarią czujnika.

kości E , T , v_{max} i \bar{v} . Z kolei wskaźniki jakości S i s osiągnęły lepsze wartości. Można to wytłumaczyć tym, że w związku z awarią czujnika robot jechał wolniej, ale z bardziej równomierną prędkością i poruszał się bliżej przeszkód przez co droga przez niego wykonana była krótsza.

Symulacja 3 – skaner laserowy ulega awarii – po czasie 1 [s]

Ostatnią symulację wykonano przy założeniu, że w następującej awaria skanera laserowego po 1 s. Wyniki symulacji pokazane są na rys. 6 i w tab. 3.



Rys. 6. Wyniki Symulacji 3: a – trajektoria ruchu robota z zaznaczeniem położenia celu i przeszkód, b – zadane prędkości liniowe dla lewych i prawych kół robota, c – prędkość ruchu robota i zmiana parametru WaterTank, d – odległość i odchylenie robota od celu, e – odczyty dalmierzy laserowych, f – odczyty skanera dla wybranych 11. wiązek, g–i – wartości sterowań i wagi dla poszczególnych podzachowań

Tab. 3. Uzyskane wartości wskaźników jakości dla Symulacji 3

Wskaźnik jakości	E [m ² s]	S [m/s]	s [m]	T [s]	v_{max} [m/s]	\bar{v} [m/s]
Wartość 10	564	0,177	28,9	47,2	0,85 ² 0,61	33

W tym przypadku uzyskano gorsze wartości wszystkich wskaźników jakości w porównaniu z Symulacją 1 poza wskaźnikiem S , co było związane z tym, że robot poruszał się z mniejszą i bardziej równomierną prędkością. W porównaniu z Symulacją 2 robot osiągał cel szybciej poruszając się z większą średnią prędkością, co było związane z tym, że robot miał uboższą informację o przeszkodach, przez co poruszał się bardziej swobodnie.

W ramach realizacji niniejszej pracy wykonano także badania symulacyjne zakładające awarię pozostałych czujników dla przypadku awarii pojedynczego czujnika.

Na podstawie wszystkich wykonanych symulacji można stwierdzić, że w większości przypadków opracowany algorytm semiautonomii dobrze radzi sobie z awarią pojedynczego czujnika.

Animacje ruchu robota zrealizowane z zastosowaniem przybornika Simulink 3D Animation dla opisanych w pracy przypadków symulacji można znaleźć pod adresem [17].

6. PODSUMOWANIE, WNIOSKI I KIERUNKI DALSZYCH BADAŃ

W niniejszej pracy opisano wyniki prac badawczych dotyczących analizy wrażliwości algorytmu semiautonomii mobilnego robota bojowego na awarię czujników otoczenia. Opracowany algorytm semiautonomii uwzględniał możliwość wystąpienia awarii czujników, korygując w odpowiedni sposób podzachowanie robota. Przedstawiono wyniki badań symulacyjnych zrealizowanych z zastosowaniem pakietu Matlab/Simulink dla przypadków normalnej pracy czujników otoczenia oraz awarii wybranych z nich. Dokonano porównania uzyskanych wyników symulacji za pomocą przyjętych sześciu wskaźników jakości. Uzyskane wyniki świadczą o tym, że opracowany algorytm semiautonomii wykazuje dużą odporność na awarię czujników. Co więcej, w przypadku awarii niektórych czujników uzyskuje się lepsze wartości wskaźników jakości. Takie zachowanie algorytmu można wyjaśnić tym, że robot wskutek awarii jednego z czujników przejeżdża bliżej niektórych przeszkód, przez co jego droga do celu staje się krótsza. Nie świadczy to jednak o tym, że w tym przypadku algorytm lepiej funkcjonuje i że należy usunąć z robota czujnik, który uległ awarii.

Kierunki dalszych badań będą dotyczyły symulacji awarii w większej liczbie czujników oraz implementacji zmodyfikowanego algorytmu semiautonomii na mobilnym robocie bojowym oraz realizacji badań eksperymentalnych związanych z analizą wrażliwości tego algorytmu na imitowaną awarię czujników otoczenia.

² Pominięto początkowe maksymalne wartości, które są efektem działania algorytmu przed awarią czujnika.

BIBLIOGRAFIA

- [1] A. Wołoszczuk, M. Andrzejczak i P. Szykarczyk, "Architecture of mobile robotics platform planned for intelligent robotic port er system – IRPS project," *Journal of Automation, Mobile Robotics & Intelligent Systems*, vol. 1, 59–63, 2007.
- [2] M. Trojnecki i P. Szykarczyk, "Tendencje rozwoju mobilnych robotów lądowych (3). Autonomia robotów mobilnych – stan obecny i perspektywy rozwoju" , *Pomiary Automatyka Robotyka*, Nr. 9/2008, 5–9.
- [3] A. Ghorbani, S. Shiry and A. Nodehi, "Using Genetic Algorithm for a Mobile Robot Path Planning," in *International Conference on Future Computer and Communication*, 164–166, 2009.
- [4] M. Wang i J. N. K. Liu, "Fuzzy logic-based real-time robot navigation in unknown environment with dead ends," *Robotics and Autonomous Systems* vol. 56, 625–643, 2008.
- [5] X.-J. Jing, "Behaviour dynamics based motion planning of mobile robots in uncertain dynamic environments," *Robotics and Autonomous Systems* vol. 53, 99–123, 2005.
- [6] M. Soika, "A sensor failure detection framework for autonomous mobile robots", in *Intelligent Robots and Systems, 1997. IROS '97., Proceedings of the 1997 IEEE/RSJ International Conference on*, 1997, 1735–1740 vol.3.
- [7] N. Ranganathan, M. I. Patel and R. Sathyamurthy, "An intelligent system for failure detection and control in an autonomous underwater vehicle," *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on*, vol. 31, 762–767, 2001.
- [8] K. Bouibed, A. Aitouche and M. Bayart, "Sensor fault detection by sliding mode observer applied to an autonomous vehicle," in *Advances in Computational Tools for Engineering Applications, 2009. ACTEA '09. International Conference on*, 621–626, 2009.
- [9] I. J. Rudas, I. Ori and A. Toth, "Design methodology and environment for robot diagnosis," in *Industrial Electronics, 1993. Conference Proceedings, ISIE'93 - Budapest., IEEE International Symposium on*, 367–372, 1993.
- [10] K. Kroschel and A. Wernz, "Sensor Fault Detection and Localisation Using Decorrelation Methods," *Sensors and Actuators*, vol. 25–27, pp. 43–50, 1991.
- [11] G. J. S. Ra e and S. E. Dunn, "On-Line detection for AUV," in *IEEE Symp. Autonomous Underwater Vehicle Technology*, 383–392, 1994.
- [12] T. J. Farrel and B. Appl eby, "Using learning techniques to accommodate unanticipated faults," in *IEEE Trans. Control Syst. Technol.*, 1993, 40–49.
- [13] M. L. Visinsky, J. R. Cavallaro and J. D. Walker, "Expert System Framework for Fault Detection and Fault Tolerance in Robotics," *Computers & Electrical Engineering*, vol. 20(5), 421–435, 1994.
- [14] "Toward an automatic health monitor for autonomous underwater vehicles using parameter identification," in *Amer. Control Conf.*, 585–589, 1993.
- [15] P. Bigaj, M. Trojnecki i J. Bartoszek, "Robot IBIS – realizacja ruchu w trybie teleoperacji i semiautonomii," *Prace naukowe Politechniki Warszawskiej, Seria: Elektronika, Zeszyt 175, Tom 1*, 135–148, 2010.
- [16] *Oferta robota IBIS produkcji PIAP* (stan na dzień 24.11.2010) dostępna na stronie internetowej: http://www.antyterrorizm.com/robot_bojowy.php
- [17] *Filmy prezentujące możliwości robota IBIS produkcji PIAP* dostępne pod adresem (stan na dzień 24.11.2010): <http://www.youtube.com/user/osmpiap>
- [18] J. Borenstein and Y. Koren, "High-speed obstacle avoidance for mobile robots", in *Intelligent Control, 1988. Proceedings., IEEE International Symposium on*, 382–384, 1988.
- [19] I. Ulrich and J. Borenstein, "VFH+: reliable obstacle avoidance for fast mobile robots", in *Robotics and Automation, 1998. Proceedings. 1998 IEEE International Conference on*, 1572–1577, 1998.