

Kamil WENCEL, Artur BABIARZ  
Politechnika Śląska

## PROJEKT ROZPROSZONEGO SYSTEMU STEROWANIA PLATFORMĄ MOBILNĄ

**Streszczenie.** W pracy przedstawiono założenia oraz elementy składowe rozproszonego systemu sterowania platformą mobilną. Komunikacja z robotem mobilnym została oparta na protokole UDP. Ponadto przedstawiono algorytm wyznaczania położenia i orientacji robota. Przedyskutowano również problem dokładności wyznaczania pozycji na podstawie GPS oraz orientacji robota na podstawie pomiarów uzyskanych z magnetometru. W podsumowaniu przedstawiono testy różnych typów sieci wykorzystanych do komunikacji z platformą mobilną.

## PROJECT OF DISTRIBUTED CONTROL SYSTEM OF MOBILE PLATFORM

**Summary.** This article presents the assumptions and elements related to distributed control system for mobile robots. The communication between robot and control station based on UDP protocol. Moreover the algorithm for determination of orientation and position is presented and discussed. The article contains conclusions concerning accuracy position and orientation of mobile platform obtained from GPS, and magnetometer, respectively. Finally the test of various types of network is shown.

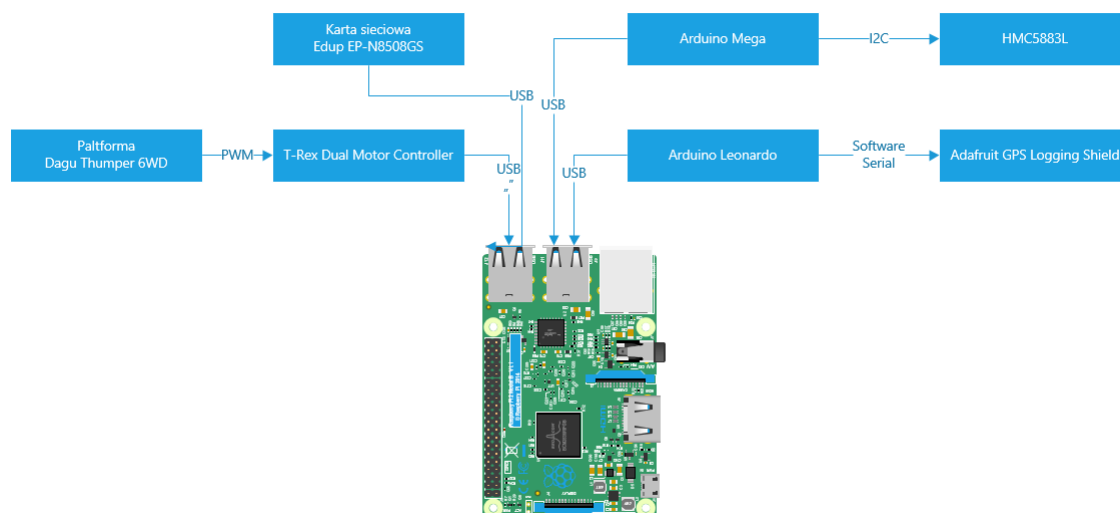
### 1. Wstęp

W ostatnich latach nastąpił gwałtowny postęp technologiczny na gruncie miniaturyzacji urządzeń. Obecnie bardzo wydajne komputery mają rozmiary karty kredytowej. Otwiera to nowe horyzonty w przypadku projektowania wydajnych rozproszonych systemów sterowania, czy też małych wydajnych domowych systemów zarządzania bądź automatyki. Niezaprzeczalnie również rozwija się Internet mobilny. Obecnie standardem jest HSDPA/HSDPA+. HSDPA+ zaczęto rozwijać stosunkowo niedawno, bo w 2008 roku. W teorii HSDPA+ ogranicza prędkość pobierania do 42 Mb/s, a wysyłania do 11 Mb/s. W praktyce parametry te są dużo gorsze i zależą od odległości od nadajnika. Wraz ze wzrostem odległości prędkość gwałtownie spada.

Obecnie do sterowania platformami mobilnymi wykorzystuje się nadajniki radiowe. Sporadycznie wykorzystywane jest sterowanie poprzez sieć Wi-Fi, jednak wyłącznie w sieciach lokalnych.

W pracach [1–5] autorzy przedstawiają różne systemy rozproszone stosowane do sterowania robotów mobilnych. Przegląd różnych algorytmów stosowanych do sterowania rozproszonego robotami mobilnymi można znaleźć w [7].

W proponowanym rozwiązaniu prezentowane jest uniwersalne rozwiązanie do



Rys. 1. Schemat organizacyjny wykorzystanych elementów elektronicznych

sterowania robotem mobilnym. Wspomniane powyżej prace prezentują głównie indywidualne rozwiązania, które mogą być stosowane jedynie do jednego typu platformy mobilnej. Wynika to z zastosowanego typu czujników, opisów kinematyki i dynamiki oraz rodzaju platformy mobilnej.

Motywacją do prac związanych ze sterowaniem rozproszonym jest problem z zasięgiem dowolnych platform mobilnych. Obecnie zasięg platform mobilnych, na przykład quadrokopterów z zewnętrznymi antenami, wynosi maksymalnie około 2 km. Wykorzystanie wspomnianej techniki rozproszonego systemu sterowania z wykorzystaniem VPN rozwiązuje ten problem. Platforma taka przełączając się pomiędzy kolejnymi nadajnikami GSM, będzie mogła być sterowana praktycznie w nieograniczonym zasięgu.

W tej pracy autorzy skupiają się bardziej na problemie samej komunikacji i wyznaczania pozycji oraz orientacji robota mobilnego, a nie nad aspektami modelowania matematycznego dynamiki takiego robota.

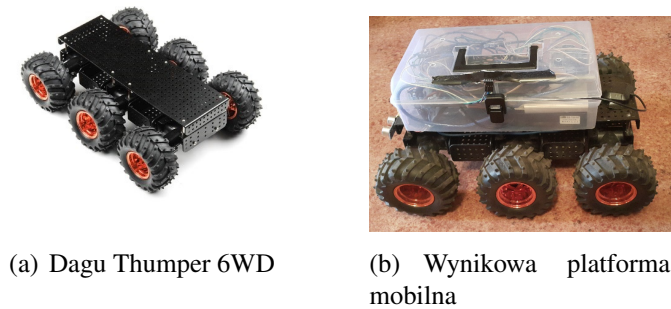
Zatem, celem pracy jest zaprojektowanie oraz stworzenie systemu sterowania platformami mobilnymi z wykorzystaniem dostępnej infrastruktury sieciowej (mobilnej jak i kablowej). Ponadto, korzystając z VPN sprawdzane będą możliwości takiej komunikacji oraz komfort sterowania. Szczegóły projektu są dostępne w [6].

## 2. Elementy składowe projektu

### 2.1. Część elektroniczna

Projekt zakłada stworzenie elastycznej platformy mobilnej, która ma zdolność poruszania się po różnym podłożu. Do wygodnego sterowania platformą wymagany jest zewnętrzny kontroler taki jak klawiatura, joystick czy pad, który miałby możliwość łatwej w sensie programistycznym komunikacji ze stacją sterującą platformą (komputer PC), jednocześnie zapewniając możliwość regulacji siły nacisku. Trzon sterowania stanowi aplikacja na komputerze PC oraz kontroler Xbox360. Po stronie platformy zastosowanie znajduje wiele urządzeń, które współpracując ze sobą tworząc załączek zdalnie sterowanego pojazdu mobilnego (rys. 1).

Głównym elementem robota jest metalowa platforma Dagu Thumper 6WD (rys.



Rys. 2. Platforma mobilna

2) posiadająca 6 silników z przełożeniem 34 : 1. Pojazd przy wymiarach  $42 \times 30 \times 13$  cm waży 2.7 kg. Maksymalny udźwig platformy wynosi 15 kg. Przyglądając się powyższej ilustracji, można odnieść wrażenie nadmiaru wykorzystywanego sprzętu elektronicznego. Można zauważyć, iż magnetometr oraz GPS mogą zostać obsługane wyłącznie wykorzystując mikrokomputer Raspberry Pi.

## 2.2. Część programowa

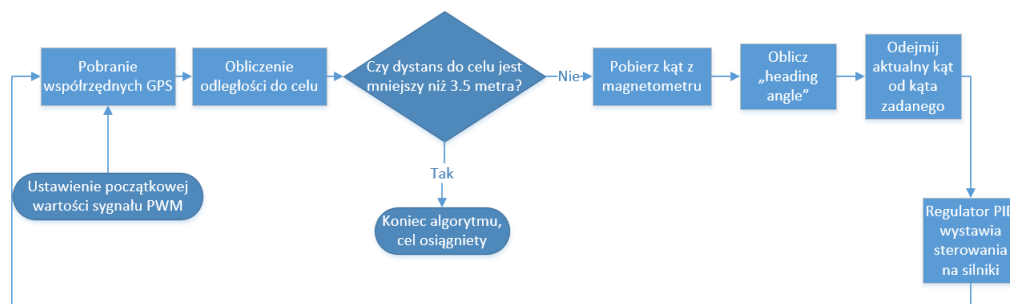
Dla potrzeb projektu wymagana była obsługa kontrolera w technologii WinForms. Najprostszym sposobem (oprócz czasochłonnego niskopoziomowego kodu) jest instalacja SlimDX. SlimDX jest open-source framework, który pozwala deweloperom w łatwy sposób budować aplikację DireX przy użyciu technologii .Net takich jak C# czy VisualBasic. Posiada również domyślnie zaimplementowaną obsługę kontrolera Xbox360.

Komunikacja sieciowa opiera się na modelu interakcji klient-serwer. Klient i serwer komunikują się używając rozumianych przez obie strony poleceń – protokołu warstwy aplikacji. Do przesyłania tych poleceń wykorzystywana jest warstwa transportowa. Klient to program inicjujący połączenie i żądający od serwera usługi. Serwer to program będący dostawcą usługi. Serwer pasywnie oczekuje na żądanie, wykonuje żądanie i wysyła odpowiedź. W przypadku tego projektu naturalnym wyborem wydaje się przyjęcie mikrokomputera Raspberry Pi jako serwera, natomiast stanowiska sterującego jako klienta.

Do implementacji architektury klient-serwer wymagane będzie wykorzystanie dwóch języków programowania - Python po stronie Raspberry Pi oraz C# po stronie komputera PC.

Do połączenia pomiędzy urządzeniami wykorzystano VPN (Virtual Private Network, Wirtualna Sieć Prywatna). Wybrano jedno z darmowych rozwiązań dostępnych na rynku oraz zewnętrzny serwer producenta oprogramowania do zestawienia połączenia, gdyż nie ma możliwości stworzenia własnego. Na podstawie przeprowadzonych rozważań, najlepszym wyborem jest program firmy LogMeIn – Hamachi.

Wykorzystany w transmisji danych protokół warstwy transportowej UDP nie posiada mechanizmów kontroli przepływu, retransmisji pakietów, ani nie numeruje kolejnych ramek danych. Dlatego w rozwiązaniu opartym o protokół UDP istotne jest, aby dokonać w warstwie aplikacji sprawdzenia poprawności dostarczenia danych. Kluczowe dla prawidłowego sterowania jest odebranie całości potrzebnych danych, dlatego też zdefiniowana została ramka danych. Ramka rozpoczyna się od znaku '\$', po której na-



Rys. 3. Schemat algorytmu sterowania autonomicznego - podążanie do celu

stępuje seria danych rozdzielonych znakiem ‘%’. Zakończenie ramki to znak ‘#’.

### 3. Wyniki działania platformy

#### 3.1. Tryb sterowania autonomiczny

Pod pojęciem sterowanie autonomiczne rozumiemy podążanie platformy od aktualnego położenia do położenia zadanego, określonego przez współrzędne GPS. Jest to podstawa do rozwiązania problemu pełnej autonomii pojazdu – omijanie naturalnych przeszkód czy bardziej zaawansowane algorytmy planowania trasy nie są tematem tej pracy.

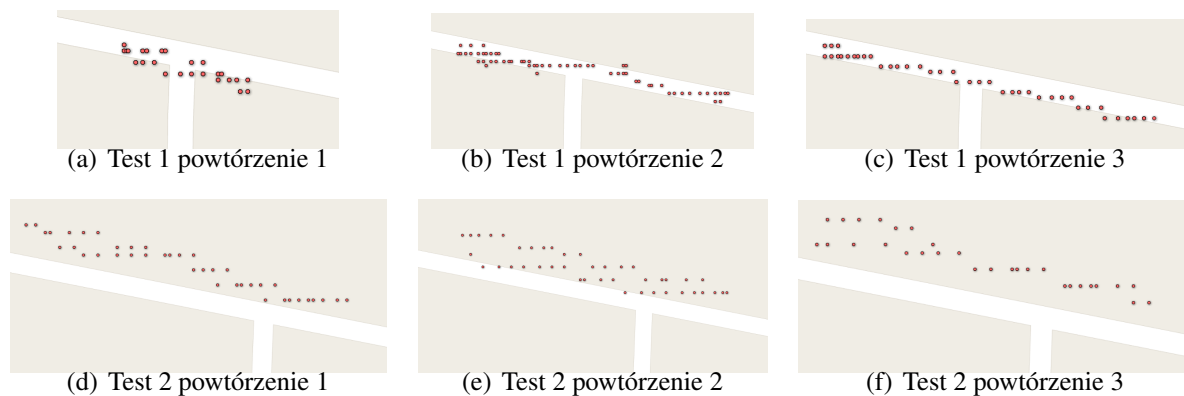
Sterowanie autonomiczne posiada szereg parametrów, które należy określić przed rozpoczęciem ruchu (rys. 3):

- dystans do celu – na tej podstawie pojazd wie kiedy ma się zatrzymać. Moduł posiada dokładność wyznaczania pozycji poniżej 3 m, a więc pojazd „uznaje”, że jest u celu jeżeli jego odległość od celu wynosi mniej niż 3.5 m,
- „Heading angle” – kąt pomiędzy współrzędnymi GPS. Jest to najważniejszy parametr niezbędny do sterowania. To na jego podstawie platforma ma informację, w którym kierunku powinna się poruszać aby osiągnąć cel,
- aktualna orientacja platformy - aby móc wykorzystać parametr „heading angle” ważna jest aktualna orientacja platformy. Parametr ten jest sprzężeniem zwrotnym dla systemu sterowania, który informuje czy platforma porusza się w odpowiednim kierunku aby osiągnąć zadany cel.

Jeżeli wspomniane parametry zostaną wyznaczone rozpoczyna się proces sterowania. Dla celów dalszych rozważań przyjmijmy, że do wyznaczania orientacji używamy magnetometru. Pierwszym etapem procesu sterowania jest taki obrót platformy wokół własnej osi, aby orientacja platformy pokrywała się z wyznaczonym współczynnikiem „heading angle”. Aby tego dokonać należało zaimplementować prosty regulator, który pozwoli uzyskać platformie zamierzony kąt. Do tego celu został zaimplementowany dyskretny regulator PID.

$$U(z) = \left[ K_p + \frac{K_i}{1 - z^{-1}} + K_d(1 - z^{-1}) \right] E(z). \quad (1)$$

Regulator PID to kombinacja trzech elementów: członu proporcjonalnego o wzmacnieniu  $K_p$ , członu całkującego o wzmacnieniu części całkującej  $K_i$  oraz wzmacnienia czę-



Rys. 4. Test GPS

ści różniczkującej  $K_d$ . Parametry regulatora zostały dobrane na podstawie doświadczeń i obserwacji zachowania platformy. Najlepsze rezultaty osiągała platforma dla wzmocnienia proporcjonalnego równego 6.5, wzmocnienia części całkującej równej 0.7 oraz wzmocnienia części różniczkującej wynoszącej 0.1. Po wykonaniu tego etapu następuje kolejna część związana z przemieszczaniem się platformy do celu.

## 4. Testy robota mobilnego

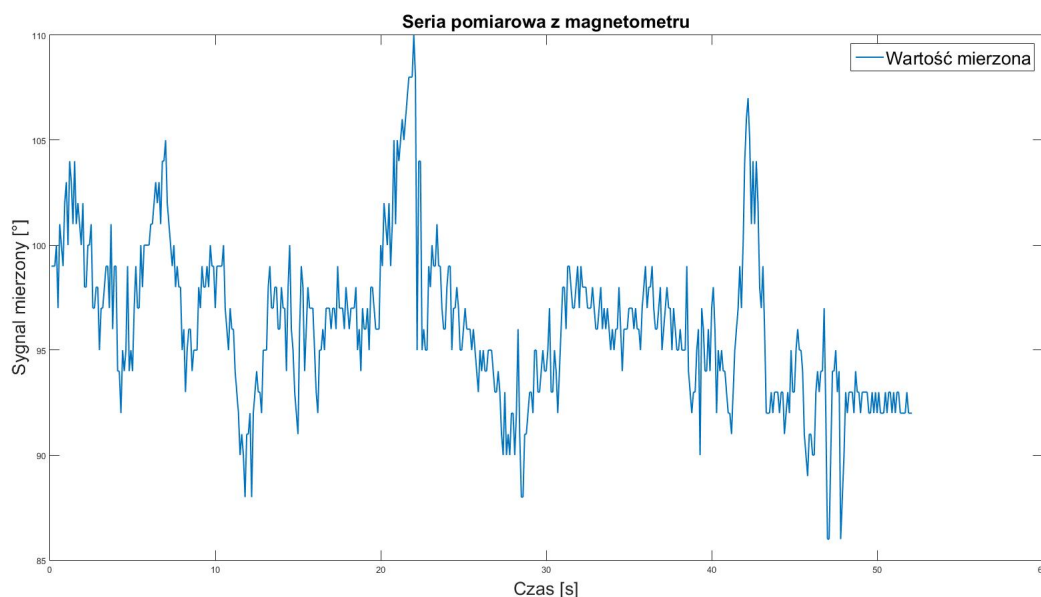
### 4.1. Dokładność odbiornika GPS

Sterowanie manualne jest stosunkowo prostą czynnością, gdyż sterując wychyleniem galek na padzie możemy poruszać się platformą. Mamy tutaj kilka zmiennych, które wspierają operatora w procesie sterowania platformą. Kluczowe z punktu operatora platformy jest aktualne położenie platformy na mapie. W tym celu przeprowadzono badanie, celem którego jest sprawdzenie, jak położenie platformy wskazywane przez odbiornik GPS oddaje rzeczywistą pozycję platformy. W tym celu przeprowadzono dwukrotnie test. Każdy test zawierał trzy powtórzenia. Testy były wykonywane w różne dni aby sprawdzić również wpływ zmieniających się warunków atmosferycznych i zmian rozmieszczenia satelitów na niebie. Pojazd poruszał się wzdłuż drogi, którą przedstawia rysunek 4. Kropki oznaczają kolejne położenia pojazdu. Z przeprowadzonych testów wynika, że nie ma możliwości wykorzystania zmieniających się współrzędnych GPS do określania orientacji platformy ze względu na otrzymaną dokładność.

Przykładem rozwiązania problemu zwiększenia dokładności może być DGPS. Jest to metoda pozwalająca uzyskać większą dokładność, niż zastosowanie pojedynczego odbiornika. W tej metodzie wykorzystuje się stację bazową, czyli odbiornik ustawiony dokładnie w wyznaczonym punkcie (takie miejsce może być wyznaczone na przykład przez pomiar geodezyjny), który wyznacza na bieżąco poprawki różnicowe dla każdego satelity, co pozwala wyeliminować większą część błędów. Drugi odbiornik musi mieć możliwość odebrania tych poprawek i ich wykorzystania.

### 4.2. Orientacja platformy

Orientacja platformy może być wyznaczona w dwojaki sposób. Pierwszy sposób to wykorzystanie magnetometru, drugi wykorzystanie współrzędnych GPS. Drugie rozwiązanie zostało odrzucone ze względu na wyniki powyższych testów. Rysunek 5



Rys. 5. Wykres orientacji platformy w czasie

przedstawia pomiary pozyskane z magnetometru. Jak można zauważyć sygnał jest mocno zaszumiony, co skutecznie utrudniałoby sterowanie autonomiczne platformą.

Aby ułatwić regulatorowi sterowanie platformą należy przefiltrować sygnał pochodzący z magnetometrów. Pierwszym sposobem przefiltrowania tego sygnału jest średnia ruchoma. Po przeprowadzonych testach okazuje się, że średnia ruchoma, jako filtr dla danych pochodzących z magnetometru nie jest najlepszym rozwiązaniem.

Do filtrowania tego typu sygnałów wybrano filtr Kalmana, jako drugą metodę. Algorytm filtru Kalmana składa się z dwóch faz. Pierwsza polega na wyliczeniu jednokrokowej predykcji stanu, czyli estymaty a priori  $\hat{x}(t+1|t)$  i jej macierzy kowariancji  $P(t+1|t)$ .

$$\begin{aligned}\hat{x}(t+1|t) &= A\hat{x}(t|t) + Bu(t), \\ P(t+1|t) &= AP(t|t)A^T + V.\end{aligned}\quad (2)$$

Po tej operacji następuje zwiększenie czasu i algorytm przechodzi do fazy drugiej czyli aktualizacji pomiarów:

$$\begin{aligned}\varepsilon(t) &= y(t) - C\hat{x}(t|t-1), \\ S(t) &= CP(t|t-1)C^T + W, \\ K(t) &= P(t|t-1)C^T S^{-1}(t), \\ \hat{x}(t|t) &= \hat{x}(t|t-1) + K(t)\varepsilon(t), \\ P(t|t) &= P(t|t-1) - K(t)S(t)K(t)^T.\end{aligned}\quad (3)$$

W tym przypadku mamy do czynienia z układem o jednej zmiennej stanu, co znacznie ułatwia cały proces projektowania. Wówczas macierze  $A$ ,  $C$ ,  $V$  oraz  $W$ , jak i macierze  $P$ ,  $K$  oraz  $S$  są skalarami. Wartości macierzy  $A$ ,  $C$ ,  $V$  są równe odpowiednio: 1, 1 i 5. Wartości początkowe  $\hat{x}(0|0)$  i  $P(0|0)$  wynoszą 0 oraz 1.

### 4.3. Komunikacja z platformą na różnym poziomie rozproszenia

Najważniejszym wnioskiem płynącym z pracy powinna być odpowiedź na pytanie czy sterowanie w rozproszonej sieci Internetowej jest możliwe, a jeżeli tak czy zapewnia komfort przy sterowaniu. Na podstawie przeprowadzonych eksperymentów i doświadczeń wywnioskowano, iż aby komfortowo sterować platformą pakiet danych powinien być dostarczany co około 50 – 60 ms. Wykonano kilka eksperymentów przy różnych typach sieci, aby sprawdzić czy występują różnice pomiędzy nimi. Wykorzystany protokół to UDP.

Pierwszy eksperyment jest doświadczeniem, do którego będą porównywane pozostałe. Komunikacja odbywa się w sieci lokalnej w obrębie jednego routera. Kolejnym eksperymentem jest test tego samego połączenia, jednak dane zamiast być transmitowane bezpośrednio zostaną przekazane przez tunel VPN zestawiony pomiędzy urządzeniami. Zestawione połączenie VPN to połączenie P2P. Zagłębiając się w dalszą analizę dokonano eksperymentu pełnego rozproszenia. Aplikacja na komputerze PC poprzez łącznie światłowodowe komunikowała się z platformą podłączoną poprzez sieć mobilną w technologii LTE. Tutaj również został zestawiony tunel VPN, aby urządzenia mogły się ze sobą skomunikować. Ostatnim eksperymentem było połączenie obu urządzeń z siecią mobilną. Każde z nich korzystało z innej sieci. Więc komunikacja odbywała się prawdopodobnie poprzez różne nadajniki oraz różnych dostawców. Nadal wykorzystano połączenie tunelowe VPN.

Wyniki powyższych eksperymentów zestawiono w tabeli 1.

Tabela 1

Tabela podsumowująca czasy odpowiedzi serwera

Typ połączenia klienta	Typ połączenia serwera	Tunel VPN	Średni czas odpowiedzi [ms]
Sieć lokalna	Sieć lokalna	Nie	1.11
Sieć lokalna	Sieć lokalna	Tak	2.52
Sieć mobilna	Sieć kablowa	Tak	2.93
Sieć kablowa	Sieć mobilna	Tak	2.97
Sieć mobilna	Sieć mobilna	Tak	3.25

## 5. Wnioski

Zdalne sterowanie platformą z wykorzystaniem elementów algorytmu autonomicznego sterowania nie jest zagadnieniem prostym. Złożoność algorytmów, fuzji czujników i przewidywania wielu przeszkód na drodze sprawia że temat jest ciekawy, ale jednocześnie bardzo złożony. Zaczynając od położenia platformy, czyli współrzędnych, GPS problemem jest dokładność. Pierwsza kwestia to celowe pogorszenie sygnału dla zastosowań cywilnych. Inną kwestią są warunki atmosferyczne czy aktualna lokalizacja satelitów na niebie. Niestety wykorzystując jeden odbiornik GPS trzeba liczyć się z tym, iż otrzymane pomiary nie będą zbyt dokładne. Coraz większą popularnością cieszą się rozwiązania wykorzystujące różne rodzaje nawigacji satelitarnej, czy też kilka odbiorników w różnej konfiguracji.

Kolejnym problemem jest magnetometr, który jest bardzo czuły na wiele czynników. Pozostaje jeszcze jeden problem dotyczący wykorzystywanego czujnika. Kompas jest wrażliwy na wszelkie przechylenia, fałszując kalkulację kierunku południka magnetycznego. Musi znajdować na płaskiej powierzchni, aby wyniki były poprawne. Niepożądany efekt wpływu przechyleń można wyeliminować za pomocą akcelerometru.

Ważnym czynnikiem jest również dobór odpowiednich parametrów regulatora PID. W tym przypadku parametry zostały dobrane na podstawie doświadczeń i obserwacji platformy. Aby jednak wykorzystać prawdziwy potencjał regulatora należy dokonać identyfikacji modelu platformy, wyeliminować strefę nieczułości i do takiego modelu dopiero dobierać parametry regulatora.

## 6. Podziękowania

Praca finansowana z funduszy Ministerstwa Nauki i Szkolnictwa Wyższego, BK-213/RAu1/2016.

## LITERATURA

1. Antonelli G., Arrichiello F., Caccavale F., Marino A.: Decentralized Time-Varying Formation Control for Multi-Robot Systems. *The International Journal of Robotics Research*, 33, 2014, pp. 1029–1043.
2. Cheng T.M., Savkin A.V.: Distributed control of mobile robotic sensor networks for multi-level barrier coverage. *11th International Conference on Control Automation Robotics & Vision (ICARCV)*, Singapore, 2010, pp. 2460–2465.
3. Ge B., Zou X., G. Yasuda G.: Distributed control for mobile robots using PSD sensors. *10th World Congress on Intelligent Control and Automation (WCICA)*, Beijing, 2012, pp. 3943–3947.
4. Loukianov A., Sugisaka M., Kimura H.: Implementing Distributed Control System for Intelligent Mobile Robot. *Artificial Life and Robotics*, 8(2), 2004, pp. 159–162.
5. Poza-Lujan J.L., Posadas-Yague J.L., Simo-Ten J.E., Simarro R., Benet G.: Distributed Sensor Architecture for Intelligent Control that Supports Quality of Control and Quality of Service. *Sensors (Basel)*, 15(3), 2015, pp. 4700–4733.
6. Wencel K.: Rozproszony system sterowania platformą mobilną. Praca magisterska, Politechnika Śląska, Gliwice, 2016.
7. Zhou Y.: Distributed Control of Multi-Robot Deployment Motion, *Motion Control, Federico Casolo (Ed.)*, InTech, 2010.