

Pojazd zdalnie sterowany – projekt i realizacja

Piotr Olbryś, Robert Piotrowski

Wydział Elektrotechniki i Automatyki, Politechnika Gdańska, ul. G. Narutowicza 11/12, 80-233 Gdańsk, Polska

Streszczenie: Pojazdy zdalnie sterowane są stosowane w wielu dziedzinach życia, między innymi w edukacji, ratownictwie oraz wojsku. W artykule przedstawiono proces projektowania i wykonania zdalnie sterowanego pojazdu. Opisano część mechaniczną i elektryczną projektu pojazdu. Przedstawiono etap programowania jednostki sterującej oraz przygotowania aplikacji mobilnej. Przeprowadzono również testy zaimplementowanych algorytmów sterowania.

Słowa kluczowe: pojazd, projektowanie, automatyka, mechatronika

1. Wprowadzenie

Zdalnie sterowane pojazdy znajdują zastosowanie w wielu dziedzinach życia. Umożliwiają poruszanie się po powierzchni lub w przestrzeni i są sterowane na odległość. Zdalne sterowanie pojazdem jest realizowane za pomocą technologii komunikacyjnych, takich jak fale radiowe, podczerwień, sieć bezprzewodowa Wi-Fi oraz Bluetooth.

Tematyka pojazdów zdalnie sterowanych została poruszona w licznych publikacjach naukowych. Opisano m.in. stosowanie pojazdu do badań stanu pionowych konstrukcji betonowych, pozwalając na znalezienie uszkodzeń konstrukcji [1]. Poruszono również tematykę monitorowania lokacji odległej od operatora pojazdu, stosując komunikację przez Internet oraz aparatury pomiarowej zainstalowanej na pojeździe [2]. Opisywane są również projekty zdalnie sterowanych pojazdów z systemami detekcji kolizji [3] oraz umożliwiającymi mapowanie pomieszczeń korzystając z czujników odległości [4].

Prezentowany artykuł przedstawia proces projektowania i budowy zdalnie sterowanego pojazdu. Celem projektu było wykonanie funkcjonalnego zdalnie sterowanego pojazdu oraz implementacja algorytmu PID do sterowania prędkością ruchu. Druga sekcja artykułu opisuje założenia projektowe pojazdu. Trzecia część artykułu dotyczy części mechanicznej projektu. Czwarta sekcja zawiera opis części elektrycznej projektu pojazdu. W piątej części przedstawiono oprogramowanie sterujące pojazdem oraz aplikację mobilną realizującą zdalne sterowanie. Szósta sekcja dotyczy przeprowadzonych badań testowych. Ostatnia część artykułu zawiera podsumowanie.

2. Założenia do projektu

Zaprojektowany oraz wykonany prototyp może poruszać się po płaskiej powierzchni w przód, prawo, lewo, tył. Rama pojazdu składa się z dwóch równolegle połączonych płyt, na których osadzono elementy elektroniczne pojazdu. Założona masa pojazdu wynosiła około 550 g. Pojazd osadzono na czterech kołach sztywno zamocowanych na silnikach prądu stałego. Manewr skrętu pojazdu wykonywany jest przez zadanie różnych prędkości obrotów silników między stronami pojazdu. Pomiar prędkości obrotów silników pojazdu jest realizowany za pomocą enkoderów osadzonych na wałach silników. Aktualny odczyt prędkości silników wykorzystywany jest w algorytmie regulacji PID. Zdalne sterowanie pojazdem jest realizowane telefonem z systemem Android oraz pojazdem wyposażonym w moduł Bluetooth. Aplikacja mobilna umożliwia sterowanie kierunkiem ruchu pojazdu oraz załączeniem lub wyłączeniem regulacji PID, oraz dobranie jednego z trzech zestawu nastaw regulatora.

Pojazd ma wbudowane ograniczenie prędkości wynikające z dużej przekładni osadzonej na wale silników. Ograniczenie wprowadza też rozdzielczość enkoderów zamocowanych na silnikach oraz częstotliwość taktowania mikrokontrolera. Pojemność baterii zasilających pojazd ogranicza czas pracy pojazdu, natomiast ich napięcie ogranicza prędkość maksymalną silników. Odległość zdalnego sterowania pojazdem ograniczona jest zasięgiem działania modułu Bluetooth.

3. Część mechaniczna

Konstrukcja mechaniczna pojazdu bazuje na dwóch połączonych równolegle płytach, które stanowią ramę, mocowania silników wraz z silnikami oraz mocowań koszyka na baterie zasilające. Płyty konstrukcyjne zawierają szereg otworów montażowych umożliwiających bezpośrednie mocowanie elementów sterujących: mikrokontrolera oraz sterowników silników. Przewidziano także otwory technologiczne dla enkoderów zamocowanych na silnikach prądu stałego znajdujących się na dolnej płycie konstrukcyjnej oraz otwór technologiczny na przewody zasilające wyprowadzone z koszyka na baterie, umieszczonego na górnej płycie konstrukcyjnej. Elementy konstrukcyjne zostały połączone za pomocą stalowych śrub i zakrętek. Ramy i mocowania zaprojektowano korzystając ze środowiska Fusion 360

Autor korespondujący:

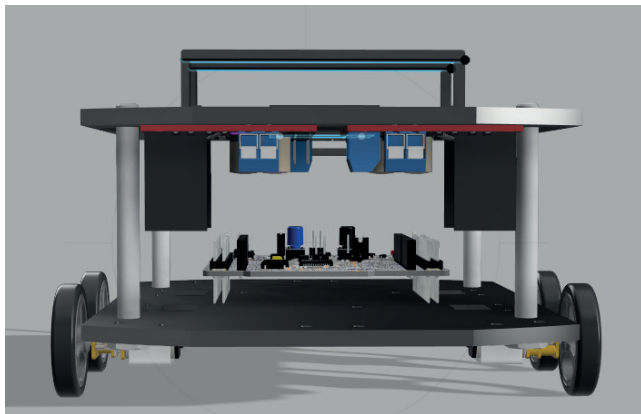
Robert Piotrowski, robert.piotrowski@pg.edu.pl

Artykuł recenzowany

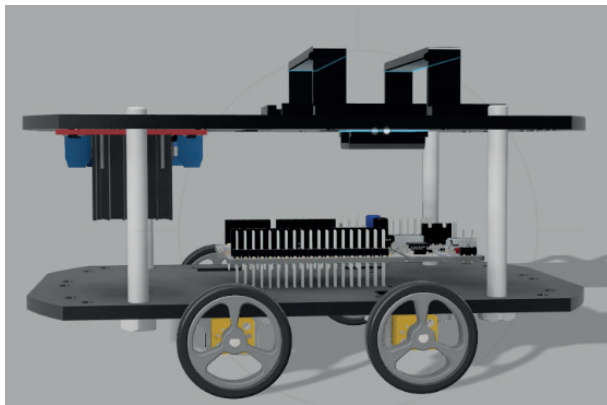
nadesłany 07.02.2025 r., przyjęty do druku 19.05.2025 r.



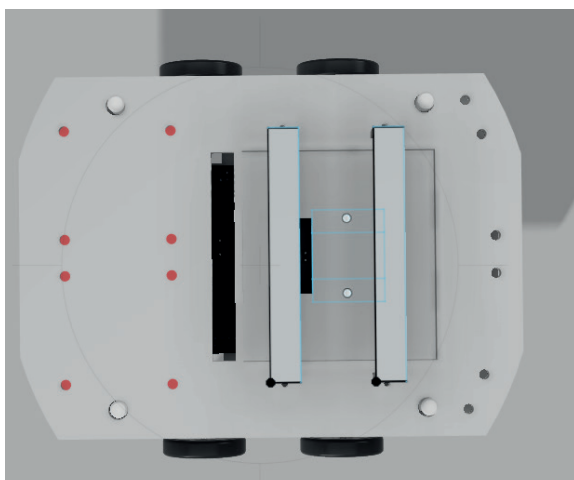
Zezwala się na korzystanie z artykułu na warunkach licencji Creative Commons Uznanie autorstwa 4.0 Int.



Rys. 1. Wizualizacja konstrukcji mechanicznej z przodu
Fig. 1. Visualization of mechanical structure's front



Rys. 2. Wizualizacja konstrukcji mechanicznej z boku
Fig. 2. Visualization of mechanical structure's side

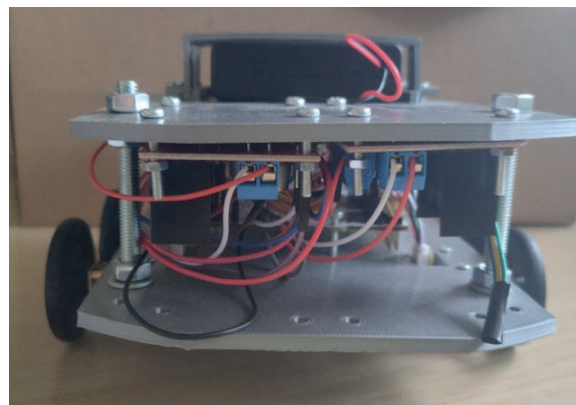


Rys. 3. Wizualizacja konstrukcji mechanicznej z góry
Fig. 3. Visualization of mechanical structure's top

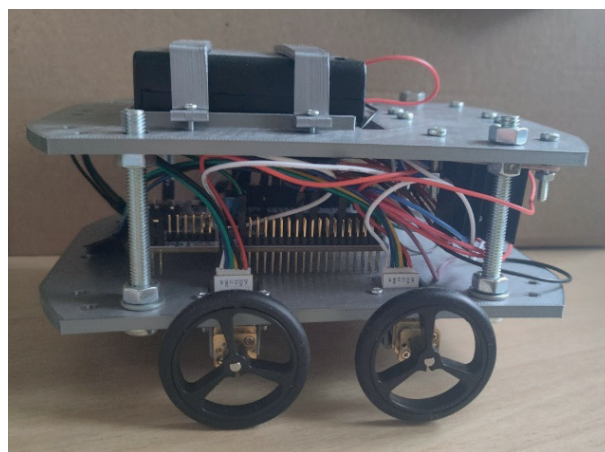
firmy Autodesk [5]. Wizualizację części mechanicznej pojazdu przedstawiono na rys. 1–3. Zbudowaną konstrukcję pojazdu przedstawiono na rys. 4 i 5. Szczegółowe etapy realizacji części mechanicznej opisano w kolejnych podrozdziałach.

3.1. Dobór silników

Po analizie, zdecydowano o zastosowaniu w projekcie silników bezszczotkowych prądu stałego N20 micro zasilanych napięciem 6 V z zamontowanymi enkoderami Halla na wałach. Dobrane silniki charakteryzują się prędkością 100 obr./min z przekładnią zębatą 1:210 i momentem obrotowym o wartości 0,255 Nm [8]. Rozdzielczość zamontowanego na wale enkodera, wykorzy-



Rys. 4. Widok pojazdu z przodu
Fig. 4. Front view of the vehicle



Rys. 5. Widok pojazdu z boku
Fig. 5. Side view of vehicle

stującego efekt Halla do pomiaru prędkości obrotowej silnika, wynosi 7 impulsów na obrót w konfiguracji AB, umożliwiające określenie kierunku obrotów silnika.

3.2. Rama konstrukcyjna

Podstawowym zadaniem ramy konstrukcyjnej jest zapewnienie sztywnej konstrukcji pojazdu oraz możliwość montażu silników, elementów sterujących i zasilania. Ramę podzielono na dwie płyty umieszczone równolegle względem siebie. Płyty wykonano metodą druku 3D. Wykorzystano technologię FDM (ang. *Fused Deposition Modeling*) oraz materiał PLA (ang. *Polylactic Acid*) [6]. Aby zapewnić sztywność i wytrzymałość płyt ustawiono ich grubość na 4 mm oraz wypełnienie wydruku na 60 %. Zdecydowano się na taką wartość wypełnienia ze względu na pomijalny wzrost wytrzymałości przy wyższym wypełnieniu oraz możliwość zmniejszenia zużycia materiału do druku. Pod dolną płytą zamontowano silniki prądu stałego korzystając z mocowania za pomocą zestawu dwóch wkrętów i zakrętek na każdy silnik [9]. Na dolnej płycie przymocowano mikrokontroler sterujący pojazdem. Sterowniki silników umocowano bezpośrednio pod górną płytą konstrukcyjną, natomiast moduł Bluetooth zamontowano korzystając z uprzednio zaprojektowanego mocowania. Koszyk na baterię zamocowano na górnej płycie, zabezpieczając dwoma mocowaniami usztywniającymi. Wyłącznik koszyka pozostaje odsłonięty umożliwiając wyłączenie pojazdu. Płyty zawierają również po sześć dodatkowych otworów montażowych umożliwiających rozbudowanie pojazdu o dodatkową aparaturę pomiarową (np. czujniki odległości lub kamery).

4. Część elektroniczna

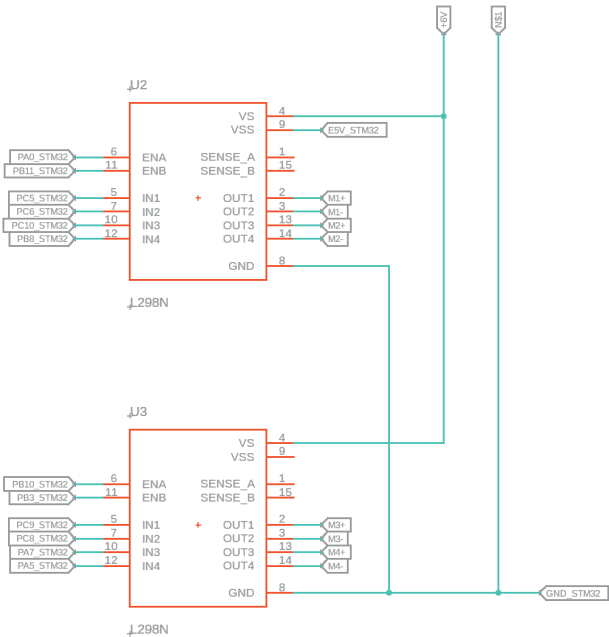
Część elektroniczną podzielono na pięć sekcji – sekcję zasilania, sterowania silnikami, enkoderów, komunikacji Bluetooth oraz płytki z mikrokontrolerem. Do połączeń elektrycznych pojazdu wykorzystano przewody wtykowe o długości 30 cm.

4.1. Zasilanie

Zasilanie pojazdu zrealizowano przez cztery ogniwa typu AA o napięciu 1,5 V umieszczone w koszyku na baterie. Szeregowe połączenie ogniw wewnątrz koszyka dało napięcie zasilające na poziomie 6 V, osiągając napięcie znamionowe dla silników. Wyprowadzenia koszyka podłączono do sterowników silników, umożliwiając pracę silników z napięciem znamionowym. Regulator napięcia wbudowany w sterownik silnikowy zmniejsza napięcie do 5 V, umożliwiając zasilenie mikrokontrolera.

4.2. Sterowanie silnikami

Sterowniki silników podłączono do zasilania, płytki z mikrokontrolerem oraz z silnikami prądu stałego. Złącze 5 V znajdujące się na sterowniku silnika podłączono do pinu zasilania płytki mikrokontrolera E5V. Sterowanie prędkością silników pojazdu zrealizowano przez regulację PWM (ang. *Pulse Width Modulation*). Wyprowadzenia mikrokontrolera skonfigurowano programowo jako wyjścia PWM, doprowadzono na piny ENA i ENB sterowników umożliwiając nastawienie prędkości silników. Piny IN1–IN4 sterowników silników podłączono do wyjść płytki mikrokontrolera skonfigurowanych jako wyjścia cyfrowe, umożliwiając wybór kierunku obrotu silników. O kierunku obrotu silnika decyduje stan wysoki na jednym z pinów IN połączonych z danymi wyprowadzeniami silnika. Schemat połączeń sekcji sterowania silnikami pokazano na rys. 6.



Rys. 6. Schemat sterowania silnikami pojazdu
Fig. 6. Vehicle engine control diagram

4.3. Enkodery

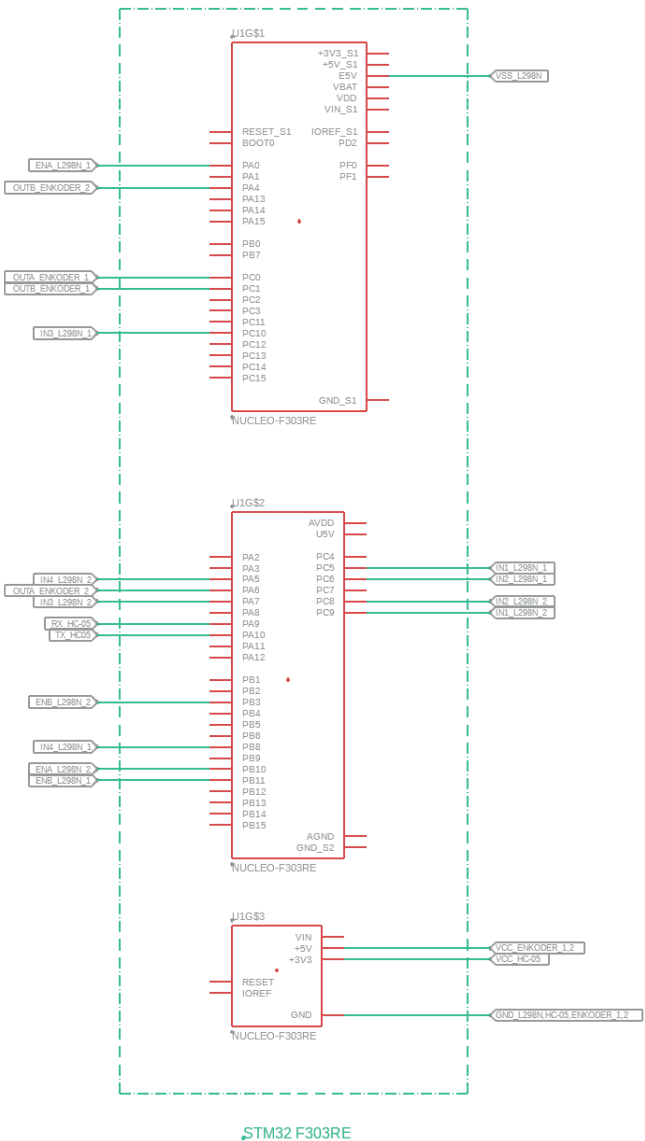
Enkodery osadzone na wałach silników podłączono do płytki mikrokontrolera. Zasilono je napięciem 5 V z wyprowadzeń płytki jednostki sterującej. Złącza OUTA i OUTB podłączono z wejściami skonfigurowanymi jako cyfrowe w programie mikrokontrolera umożliwiając odczyt sygnału impulsowego w czasie odpowiadający prędkości obrotowej silników.

4.4. Komunikacja Bluetooth

Moduł komunikacji Bluetooth HC-05 zasilono napięciem 3,3 V za pomocą wyprowadzenia płytki mikrokontrolera. Komunikacja modułu z mikrokontrolerem odbywa się przez UART (ang. *Universal Asynchronous Receiver-Transmitter*), podłączając wyprowadzenie mikrokontrolera skonfigurowane jako Tx do wyprowadzenia modułu Bluetooth Rx, natomiast pin mikrokontrolera ustawiony na Rx podłączono do pinu modułu Bluetooth Tx. Połączenie to umożliwia komunikację szeregową mikrokontrolera z modulem komunikacyjnym HC-05. Wyprowadzenie Tx odpowiada za wysyłanie danych, natomiast Rx za odczyt danych.

4.5. Płytką mikrokontrolera

Centralną jednostkę sterującą zrealizowano stosując moduł programowalny z mikrokontrolerem STM32 Nucleo-F303RE opartym na rdzeniu ARM Cortex M4 z taktowaniem 72 MHz. Płytką ma 50 programowalnych pinów mogących pełnić rolę wejść/wyjść cyfrowych oraz 22 piny z możliwością realizacji wejść analogowych. Duża liczba wyprowadzeń na płycie umożliwia realizację sterowania pojazdem, pozostawiając wiele niepodłączonych wyprowadzeń, które mogą być wykorzystane w celu rozbudowania funkcjonalności pojazdu. Programowanie płytki realizowane jest w języku C/C++ w środowisku programowalnego STM32CubeIDE [10].



Rys. 7. Schemat połączeń płytki z mikrokontrolerem STM32 F303RE
Fig. 7. Connection diagram of the STM32 F303RE microcontroller board

Komunikację z modulem komunikacyjnym Bluetooth zrealizowano przez transmisję szeregową UART, opierającą się na przesyłaniu znaków między modulem, a mikrokontrolerem. Możliwe jest również przesyłanie komunikatów z mikrokontrolera do modułu HC-05, umożliwiając przekazanie informacji o aktualnie wykonywanej komendzie przez pojazd. Schemat połączeń płytki z mikrokontrolerem widoczny jest na rys. 7.

5. Oprogramowanie pojazdu

Proces programowania jednostki centralnej pojazdu był złożony z kilku istotnych etapów. Pierwszym było skonfigurowanie parametrów mikrokontrolera. Następnie wykonano sterowanie silnikami pojazdu. Kolejny etap polegał na zrealizowaniu algorytmu obliczania prędkości i regulacji PID do sterowania prędkością silników. Następnym krokiem była realizacja programowa komunikacji szeregowej mikrokontrolera z modulem Bluetooth. Ostatnim etapem było wykonanie aplikacji Android do zdalnego sterowania pojazdem.

5.1. Konfiguracja płytki mikrokontrolera

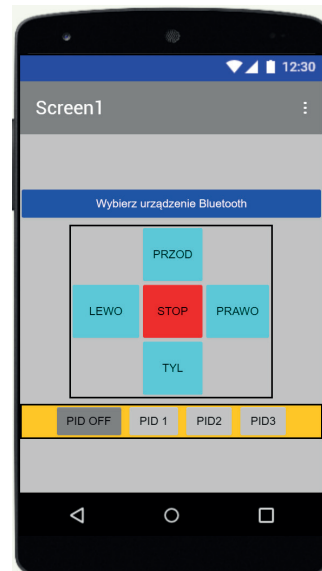
Kluczowym etapem programowania mikrokontrolera było skonfigurowanie jego parametrów. Ustawiono rezonator kryształowy/ceramiczny na HSE (ang. *High-Speed External Oscillator*) oraz ustawienie parametru szybkiego zegara HCLK na 72 MHz. Zapewniło to możliwość wykorzystania pełnej częstotliwości taktowania przez liczniki czasowe mikrokontrolera. Umożliwiło to uzyskanie pomiarów enkoderów o dużej liczbie próbek, zapewniając dokładny pomiar prędkości. Taktowanie szybkiego zegara wpływa również na maksymalną częstotliwość przesyłania sygnału PWM sterującego prędkością silników oraz maksymalną prędkość przesyłania szeregowego z modulem Bluetooth.

Następnie skonfigurowano liczniki czasowe (timery) używane do regulacji PWM silników, zliczania sygnałów enkoderów, obsługi algorytmu obliczania aktualnej prędkości silników oraz regulatorów PID.

Regulację prędkości silników za pomocą PWM zrealizowano licznikiem czasowym wykorzystującym wbudowane w środowisku STM32CubeIDE gotowe funkcje PWM Generation, określając kanały zadające sygnał PWM (użyte piny STM32 – rys. 7). Częstotliwość zadawania sygnału PWM ustawiono na 500 Hz.

Pomiar sygnału impulsowego zadanego przez enkodery zrealizowano przez dwa liczniki czasowe, w których każdy odpowiada za działanie jednego enkodera. Nie ustawiono dzielnika częstotliwości licznika, powodując pracę licznika z częstotliwością maksymalną – 72 MHz. Wykorzystano gotową funkcję zapewnioną przez środowisko programowalne – Encoder Mode. Ustawiono parametr funkcji na Encoder Mode T1 and T2, aby zliczać pomiary z wyprowadzeń A i B enkodera (podłączone piny – rys. 8). Zastosowana konfiguracja AB umożliwia określenie kierunku obrotów silnika, ponieważ A i B są przesunięte czasowo względem siebie. Odczyt sygnału z tylko jednego wyprowadzenia skutkowałby jedynie pomiarem impulsów.

Obsługę algorytmów obliczania aktualnej prędkości silników oraz regulacji PID zrealizowano na liczniku czasowym o częstotliwości wywołania 500 Hz. Wymagało to również wybrania opcji global interrupt (globalne przerwanie) w opcjach licznika czasowego, aby móc zastosować obsługę przerw w kodzie programu. Umożliwia to obsługę kodu algorytmów w jednakowych odstępach czasu – co 2 ms, co jest niezbędne do otrzymania pomiaru z wystarczająco szybkim próbkowaniem oraz poprawnej pracy algorytmu cyfrowego regulatora PID. Wywołanie algorytmów z większą częstotliwością skutkowało zbyt małą ilością impulsów w okresie wywołania funkcji zliczania prędkości, powodując brak zmiany prędkości obliczonej przez



Rys. 8. Interfejs aplikacji mobilnej w środowisku MIT App Inventor
Fig. 8. Mobile application interface in the MIT App Inventor environment

algorytm mimo znacznego zmniejszenia prędkości zadanej. Wywołanie tych funkcji przy mniejszej częstotliwości skutkowało gorszą dynamiką sterowania pojazdu, ponieważ algorytm regulacji prędkości działał w większych odstępach czasu.

Konfiguracja komunikacji mikrokontrolera z modulem Bluetooth została zrealizowana przy pomocy funkcji USART (ang. *Universal Synchronous/Asynchronous Receiver/Transmitter*), ustawiając jej parametry na zgodne z nastawionymi na module HC-05: prędkość przesyłania danych 9600 bit/s, natomiast pozostałe parametry zostawiono domyślne.

Ostatnim krokiem konfiguracji było przypisanie wyprowadzeń mikrokontrolera obsługujących kierunki obrotu silników jako wyjścia GPIO Output – wyjścia cyfrowe (przypisane piny – rys. 6).

Konfiguracja została przygotowana w środowisku STM32CubeIDE, kod w C++ odpowiadający wybranej konfiguracji został wygenerowany automatycznie.

5.2. Program sterowania silnikami

Sterowanie silnikami zrealizowano programowo wewnątrz instrukcji warunkowych poprzez wybór odpowiedniego dla danego silnika kierunku obrotu odpowiadającego kierunkowi poruszania pojazdu oraz zadanie stopnia wypełnienia sygnału PWM odpowiadającego za prędkość obrotu silnika jako wartość liczbowa z przedziału od 0 do 999. W zależności od wybranego przez użytkownika trybu pracy: PID OFF lub jeden z trzech PID w aplikacji, wartość zadana jest stała lub to wartość wyjściowa z algorytmu regulacji PID.

5.3. Algorytmy regulacji

Pomiar za pomocą enkoderów polega na przekazywaniu informacji jedynie o liczbie impulsów zliczonych przez licznik czasowy, co wymaga zastosowania dodatkowego algorytmu obliczającego aktualną prędkość obrotową silników. W tym celu obliczana jest pochodna od liczby impulsów położenia. Cyfrowa pochodna uzyskiwana jest numerycznie jako różnica wartości aktualnej liczby impulsów od poprzedniej, która zapisywana jest w zmiennej tymczasowej po każdej iteracji algorytmu. W pierwszej iteracji prędkość ustawiana jest na 0.

Dzięki dostępowi do aktualnej wartości prędkości obrotowej zastosowano algorytm regulacji zrealizowany cyfrowym regulatorem PID [7]. Ze względu na przeciwny kierunek obrotu silników po obu stronach pojazdu, algorytm PID wykonywany jest dwa razy. Uchyby regulacji obliczane są w każdym

wywołaniu algorytmu jako różnica między zadanymi wartościami prędkości silników i aktualnymi prędkościami obrotowymi silników. Człon proporcjonalny P regulatora obliczany jest jako iloczyn wzmocnienia K_p i uchybu regulacji. Do realizacji członu całkującego I wymagana była dodatkowa zmienna zawierająca całkę uchybu, czyli sumę poprzednich uchybów, z nałożonym ograniczeniem na wartość maksymalną i minimalną, aby uniknąć zbyt dużych wartości całki, co powodowałoby wystawianie zawsze maksymalnej wartości na wyjście regulatora. Wartość wyjściowa członu I to iloczyn wzmocnienia K_i , całki uchybu oraz czasu próbkowania (2 ms). Człon różniczkujący D zrealizowano dodając zmienną pomocniczą zawierającą uchyb poprzedni. Operacja pochodnej wykonywana jest analogicznie do liczenia uchybu regulacji – jako różnica uchybu aktualnego i poprzedniego. Wartość wyjściowa członu różniczkującego to iloczyn wzmocnienia K_d i pochodnej uchybu regulacji podzielonych przez czas próbkowania 2 ms.

Wartość wyjściowa regulatora PID to suma wyjść części proporcjonalnej, całkującej i różniczkującej ograniczona o wartość maksymalną 999 i minimalną -999, ponieważ to maksymalne wypełnienie sygnału sterującego PWM, odpowiadającego maksymalnej prędkości silników.

5.4. Program obsługi komunikacji Bluetooth

Odbiór wiadomości z modułu Bluetooth odbywa się w pętli programu mikrokontrolera i wiadomość zostaje zapisana jako ciąg znaków w zmiennej o pojemności 50 znaków. Otrzymana wiadomość porównywana jest w instrukcjach warunkowych sterowania silnikami. Po potwierdzeniu otrzymania komendy typu „przód” mikrokontroler wysyła wiadomość zwrotną do modułu HC-05 o wykonywaniu ruchu w przód, natomiast silniki rozpoczynają ruch w kierunku, który spowoduje ruch pojazdu w przód z określoną prędkością zadaną: 100 obr./min. Komenda zatrzymania powoduje wprowadzenie stanu niskiego na wszystkie wyprowadzenia odpowiadające za kierunki obrotów silników oraz wyzerowanie prędkości zadanej. Realizowane są również komendy określające, czy należy wyłączyć/włączyć regulację PID oraz z jakim zestawem nastaw.

5.5. Aplikacja mobilna

Aplikację mobilną sterującą pojazdem wykonano w środowisku wirtualnym MIT App Inventor [11], przeznaczonym do tworzenia aplikacji na systemy Android. Interfejs graficzny środowiska i aplikacji mobilnej pokazano na rys. 8.

Przycisk wyboru urządzenia Bluetooth umożliwia sparowanie telefonu z zainstalowanym interfejsem z pojazdem. Centralna część interfejsu aplikacji mobilnej zawiera przyciski sterujące ruchem pojazdu oraz przycisk STOP umożliwiający zatrzymanie pojazdu w dowolnej chwili. Umieszczono też przyciski umożliwiające włączenie algorytmu sterowania PID z trzema różnymi konfiguracjami parametrów regulatora PID, wraz z możliwością wyłączenia go i funkcjonowania w otwartej pętli sterowania.

Po zaprojektowaniu interfejsu aplikacji mobilnej zaprogramowano logikę działania aplikacji. Po wybraniu przez użytkownika z listy sparowanych urządzeń modułu HC-05, aplikacja umożliwia przesyłanie informacji do modułu Bluetooth. Wybór przycisku w aplikacji spowoduje wysłanie wiadomości odpowiadającej mu funkcji do urządzenia.

6. Badania testowe

Ostatnim etapem realizacji pojazdu były badania testowe. Umożliwiły one dobór parametrów oraz sprawdzenie skuteczności zastosowanego algorytmu regulacji prędkości. Podczas testów stosowano w pełni naładowane ogniwa zasilające. Aby

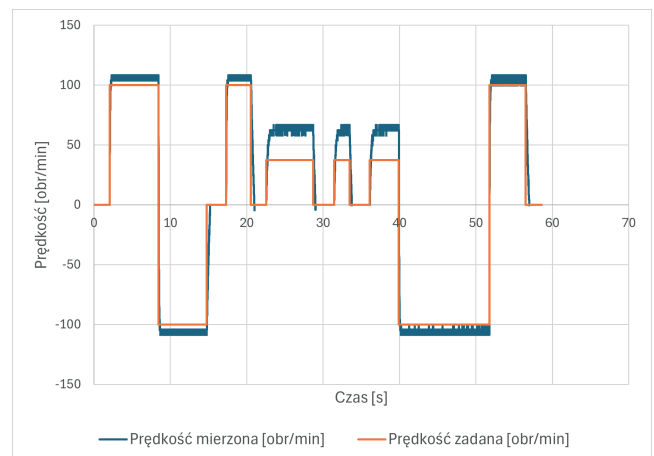
dobrać parametry regulacji wykonano pomiar charakterystyk prędkości zadanej i prędkości zmierzonej otwartego układu sterowania. Uzyskane dane pomiarowe wprowadzono do inżynierskiego środowiska programistycznego MATLAB [12]. Do identyfikacji modelu silników DC użyto bibliotekę System Identification Toolbox. Następnie otrzymane modele wprowadzono do aplikacji PID Tuner środowiska MATLAB. Dokonano tego dwukrotnie, dla pomiarów silnika z obydwu stron pojazdu.

6.1. Badania prawej części pojazdu

Wykonano dwa różne zestawy pomiarów charakterystyk czasowych prędkości zmierzonej i zadanej prawego silnika. W tym celu wykorzystano dwa różne zbiory danych – pierwsze jako dane wejściowe, natomiast drugie jako dane walidacyjne. Takie działanie umożliwiło otrzymanie dokładnego modelu prawego silnika. Otrzymane charakterystyki przedstawiono na rys. 9 i 10.

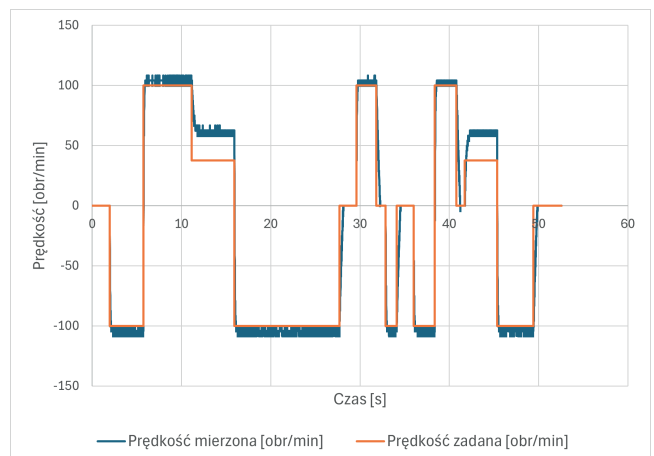
Na otrzymanych charakterystykach prędkości zmierzonej i zadanej dla silnika prawego zaobserwowano duże przeregulowanie. Zaobserwowano również krótkie czasy narastania podczas sterowania w pętli otwartej.

Po wprowadzeniu danych wejściowych i walidacyjnych otrzymano szereg estymacji modeli o zbliżonym stopniu dopasowania do danych walidacyjnych. Najlepsze wyniki otrzymano dla



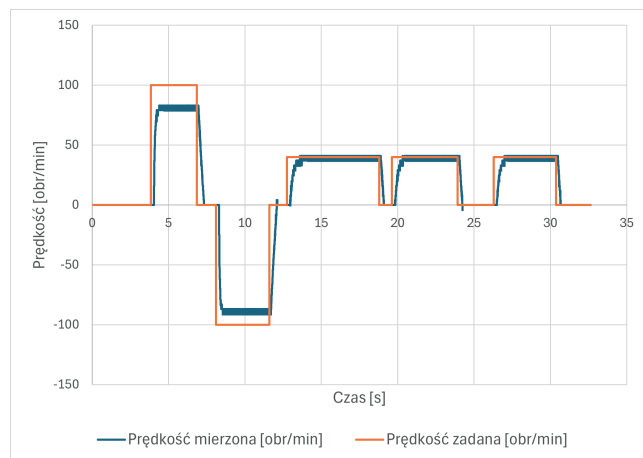
Rys. 9. Charakterystyki prędkości zmierzonej i zadanej silnika prawego – pierwszy pomiar

Fig. 9. Characteristics of measured and set speed of the right motor – first measurement



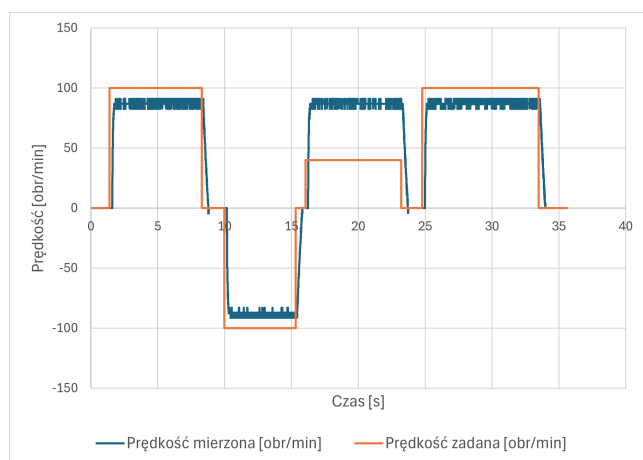
Rys. 10. Charakterystyki prędkości zmierzonej i zadanej silnika prawego – drugi pomiar

Fig. 10. Characteristics of measured and set speed of the right motor – second measurement



Rys. 11. Charakterystyki prędkości mierzonej i zadanej silnika prawego – pierwsze nastawy regulatora PI

Fig. 11. Characteristics of measured and desired speed of the right motor – first PI controller settings



Rys. 12. Charakterystyki prędkości mierzonej i zadanej silnika prawego – drugie nastawy regulatora PI

Fig. 12. Characteristics of measured and set speed of the right motor – second PI controller settings

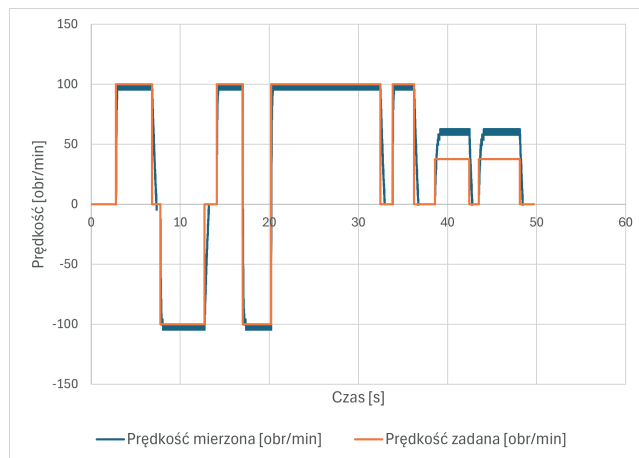
modelu obiektu o dwóch biegunach i jednym zerze – obiekt inercyjny drugiego rzędu. Po wprowadzeniu otrzymanego modelu do aplikacji PID Tuner otrzymano zestaw dwóch nastaw regulatora PI. Wzmocnienie członu D ustawiono na 0, ponieważ szybkozmienne zakłócenia pogarszały znacząco jakość regulacji.

Pierwszy zestaw nastaw to $K_p = 0,8729$ oraz $K_i = 14,87$. Otrzymane charakterystyki po zastosowaniu regulacji PI przedstawiono na rys. 11. Charakterystyki pomiaru prędkości zmierzzonej i zadanej przy zastosowaniu pierwszego zestawu nastaw wykazały wyeliminowanie przeregulowania przy prędkościach zadanych niższych od maksymalnej. Zaobserwowano również niewielkie przeregulowania przy zadanej prędkości maksymalnej.

Drugi zestaw nastaw to $K_p = 6,736$ oraz $K_i = 211,5$. Wyniki regulacji przedstawiono na rys. 12. Na otrzymanych charakterystykach zaobserwowano, iż zastosowanie tych parametrów regulatora PI spowodowało wystąpienie jednakowej prędkości na prawym silniku bez względu na prędkość zadaną. Czasy narastania również uległy zwiększeniu w porównaniu ze sterowaniem w pętli otwartej.

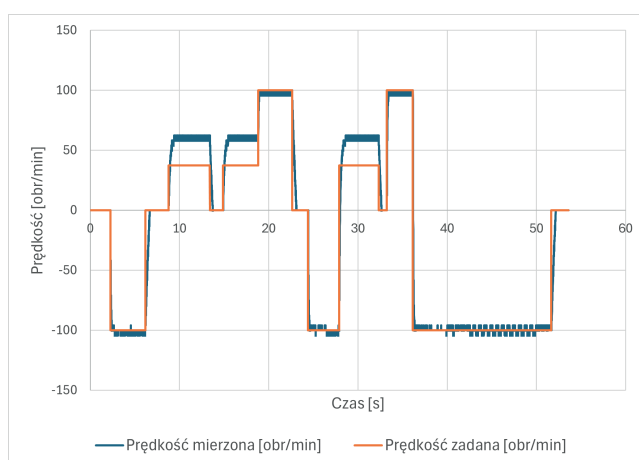
6.2. Badania lewej części pojazdu

Badania silnika lewej części pojazdu wykonano analogicznie do badań prawej części. Pomiar wejściowy i walidacyjne charakterystyki prędkości zadanej i zmierzzonej silnika lewego przedstawiono na rys. 13 i 14.



Rys. 13. Charakterystyki prędkości mierzonej i zadanej silnika lewego – pierwszy pomiar

Fig. 13. Characteristics of measured and set speed of the left motor – first measurement



Rys. 14. Charakterystyki prędkości mierzonej i zadanej silnika lewego – drugi pomiar

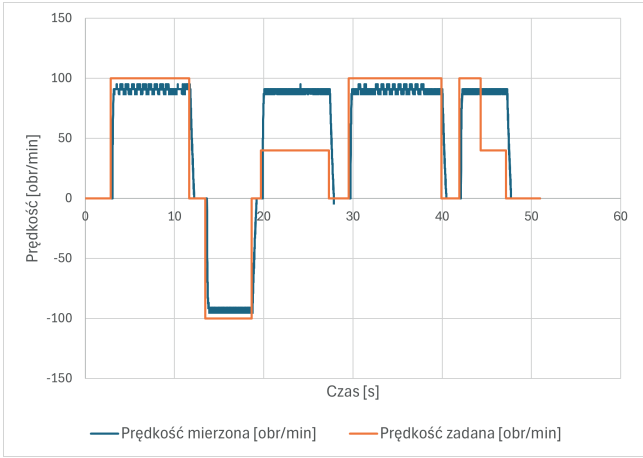
Fig. 14. Characteristics of measured and set speed of the left motor – second measurement

Na otrzymanych charakterystykach prędkości zmierzzonej i zadanej dla lewego silnika także zaobserwowano duże przeregulowanie. Analogicznie jak w prawym silniku, zaobserwowano krótkie czasy narastania mimo sterowania w pętli otwartej.

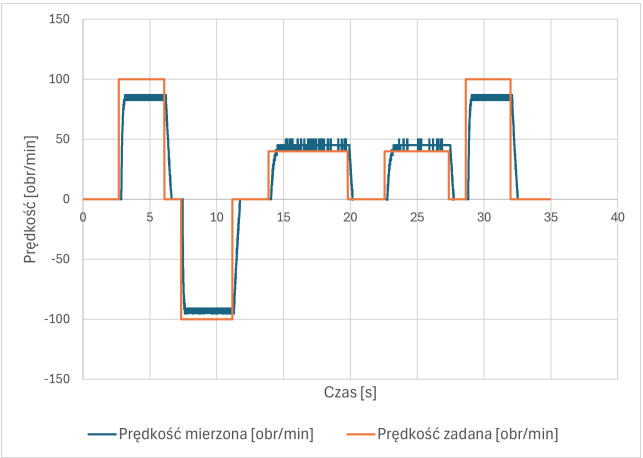
Po estymacji modelu silnika na podstawie powyższych danych otrzymano najlepsze dopasowanie dla modelu z trzema biegunami i jednym zerem – obiekt inercyjny trzeciego rzędu. Otrzymany model wprowadzono do aplikacji PID Tuner i otrzymano następujące wzmocnienia regulatora PI: $K_p = 3$ oraz $K_i = 39,85$. Charakterystyki uzyskane po zastosowaniu regulacji PI pokazano na rys. 15. Otrzymane charakterystyki prędkości lewego silnika po zastosowaniu pierwszych nastaw regulatora PI wykazały jednakową prędkość silnika bez względu na wartość prędkości zadanej.

Jako drugi zestaw nastaw przyjęto nastawy dobrane dla prawego silnika o wzmocnieniach $K_p = 0,8729$ oraz $K_i = 14,87$. Wynik zastosowania tych nastaw przedstawiono na rys. 16.

Zastosowanie drugiego zestawu nastaw regulatora PI na lewym silniku umożliwiło wyeliminowanie dużego przeregulowania przy wartościach prędkości zadanych niższych od prędkości maksymalnej. Zaobserwowano również wydłużenie czasu narastania.



Rys. 15. Charakterystyki prędkości mierzonej i zadanej silnika lewego – pierwsze nastawy PI
Fig. 15. Characteristics of measured and desired speed of the left motor – first PI controller settings



Rys. 16. Charakterystyki prędkości mierzonej i zadanej silnika lewego – drugie nastawy PI
Fig. 16. Characteristics of measured and desired speed of the left motor – second PI controller settings

6.3. Wnioski

W celu porównania wyników regulacji otrzymanych przez estymację modeli silników sporządzono tabelę 1. Zawarto w niej czasy narastania oraz uchyby regulacji w zależności od prędkości zadanej dla obu silników i obu dobranych zestawów nastaw. Przeprowadzone badania pozwoliły stwierdzić, że najlepsze wyniki dla prawej strony pojazdu, ze względu na krótszy

Tab. 1. Porównanie otrzymanych charakterystyk prędkości
Tab. 1. Comparison of the speed characteristics obtained

Badanie	Czas narastania [s]	Uchyb przy prędkości zadanej 40 obr./min	Uchyb przy prędkości zadanej 100 obr./min
Badanie 1. prawego silnika	0,381	–2	–19
Badanie 2. prawego silnika	0,415	–47	–13
Badanie 1. lewego silnika	0,304	51	9
Badanie 2. lewego silnika	0,347	5	7

czas narastania i znacząco mniejszy uchyb w stanie ustalonym dla niższej prędkości zadanej, otrzymano dla układu regulacji z pierwszym zestawem nastaw: $K_p = 0,8729$ oraz $K_i = 14,87$. Badania dla lewej strony, po podstawieniu tych samych wzmocnień, również okazały się korzystniejsze z powodu znacznego zmniejszenia uchybu regulacji przy niższych prędkościach, skutkując jedynie stosunkowo niewielkim wzrostem czasu narastania. Zmniejszenie uchybu regulacji przy niskich prędkościach zadanych jest bardzo korzystne przy sterowaniu pojazdem, ponieważ umożliwia uzyskanie większego kąta skrętu pojazdu. Skręcanie odbywa się przy zmniejszonej prędkości na stronę, w którą pojazd ma skręcić, jednocześnie utrzymując maksymalną prędkość obrotów na drugiej stronie pojazdu.

7. Podsumowanie i kierunki dalszych prac

Główny cel, czyli zaprojektowanie oraz wykonanie zdalnie sterowanego pojazdu, został w pełni zrealizowany. Spełniono wszystkie założenia projektowe. Zaprojektowany pojazd można rozbudować zapewniając skalowalność. Skalowalność pojazdu może obejmować następujące modyfikacje: większa rama pojazdu umożliwiającą dołączenie więcej peryferii, np. duży akumulator, dodatkowe silniki, chwytak pozwalający na podnoszenie przedmiotów. Można rozbudować go o dodatkowe urządzenia pomiarowe, np. czujniki odległości, które przewidziano podczas projektowania ram pojazdu, umieszczając na nich dodatkowe otwory montażowe.

Pojazd można również rozbudować stosując silniki o większej mocy i obrotach oraz dokładniejsze enkodery do pomiaru ich prędkości. Istnieje również możliwość rozbudowy oprogramowania mikrokontrolera sterującego pojazdem przez zastosowanie zaawansowanych algorytmów regulacji, takich jak sterowanie adaptacyjne lub poszerzenie funkcjonalności pojazdu o systemy detekcji kolizji, samodzielne parkowanie oraz podążanie za linią lub obiektem.

Aplikacja sterująca pojazdem również może być rozszerzona o nowe funkcje, np. sterowanie prędkością zadaną pojazdu, monitorowanie parametrów pojazdu na ekranie lub wyświetlanie obrazu na żywo z kamery po zamontowaniu jej na pojeździe. Badania testowe wykazały możliwość doboru zadowalających nastaw regulatorów przez estymację modeli silników pojazdu. Badania te mogą być rozszerzone o dobór lepszych modeli matematycznych przez zastosowanie większej liczby danych pomiarowych w procesie estymacji.

Bibliografia

- Guimaraes M., Lindberg J., *Remote controlled vehicle for inspection of vertical concrete structures*, Proceedings of the 2014 3rd International Conference on Applied Robotics for the Power Industry, Brazylia, 2014, DOI: 10.1109/CARPI.2014.7030048.
- Olakanmi O.O., Benyeogor M.S., *Internet based tele-autonomous vehicle system with beyond line-of-sight capability for remote sensing and monitoring*, "Internet of Things", Vol. 5, 2019, 97–115, DOI: 10.1016/j.iot.2018.12.003.
- Huan L., Dahu W., Tong Z., Keming H., *Research on a remote control vehicle*, 2010 2nd International Conference on Education Technology and Computer, Shanghai, 2010, V2-263–V2-266, DOI: 10.1109/ICETC.2010.5529390.
- Nguyen T., Gutierrez A., Gutierrez P., Jimenez J., Akhetuamen N.A.R., Wang Y., *RC Auto Mapper: Remote Controlled Robotic Vehicle for Autonomous Area Mapping*, 2022 IEEE International Conference on Imaging Systems and Techniques (IST), Taiwan, 2022, DOI: 10.1109/IST55454.2022.9827666.

5. Jaskulski B., *Autodesk Inventor 2022 PL/2022+ Fusion 360. Podstawy metodyki projektowania*, Helion, Gliwice, 2022.
6. Průša J., Bach M., *Basics of 3D printing with Josef Prusa*, [www.prusa3d.com].
7. Brzózka J., *Regulatory i układy automatyki*, Wydawnictwo MIKOM, Warszawa 2004.

Inne źródła

8. GA12-N20 DC Geared Motor, Speed Measurement with Hall Encoder, [www.arduiner.com].
9. Model mocowania Pololu do silników N20 micro, [www.pololu.com/file/0J1723/micro-metal-gearmotor-bracket.step].
10. STM32CubeIDE, *Integrated Development Environment for STM32*, [www.st.com/en/development-tools/stm32cubeide.html].
11. MIT App Inventor, [https://appinventor.mit.edu].
12. MATLAB, [www.mathworks.com/products/matlab.html].

Remote Controlled Mobile Vehicle – Design and Realization

Abstract: Remote controlled vehicles find applications in various areas of life, including education, rescue operations, and the military. This article outlines the process of designing and implementation of remotely controlled vehicle. The mechanical and electrical aspects of the vehicle project are described, along with programming of the control unit and mobile application. Tests of implemented control algorithms are also conducted.

Keywords: vehicle, designing, automatics, mechatronics

inż. Piotr Olbryś

s185789@student.pg.edu.pl
ORCID: 0009-0008-8298-220X

Absolwent studiów inżynierskich pierwszego stopnia na Wydziale Elektrotechniki i Automatyki Politechniki Gdańskiej (kierunek Automatyka i Robotyka, specjalność Automatyka i Systemy Sterowania). Zainteresowania: automatyka, robotyka, programowanie.



dr hab. inż. Robert Piotrowski

robert.piotrowski@pg.edu.pl
ORCID: 0000-0002-8660-300X

Absolwent i pracownik Wydziału Elektrotechniki i Automatyki Politechniki Gdańskiej. W 2005 r. uzyskał stopień doktora nauk technicznych w dyscyplinie naukowej automatyka i robotyka. W 2018 r. uzyskał stopień doktora habilitowanego nauk technicznych w dyscyplinie naukowej automatyka i robotyka. Zawodowo związany z Katedrą Inteligentnych Systemów Sterowania i Wspomagania Decyzji. Główne obszary zainteresowań dotyczą modelowania i zaawansowanych metod sterowania procesami przemysłowymi.

