Sterowanie w pojeździe z adaptacyjnym zawieszeniem

Jakub Wojciechowski[1], Tymoteusz Walczak[2], Hubert Janiec[3], Jakub Mikulski[4]
[1,3]*AGH Akademia Górniczo-Hutnicza im. Stanisława Staszica*
Wydział Inżynierii Mechanicznej i Robotyki,
Katedra Robotyki i Mechatroniki,
Al. Mickiewicza 30, 30-059 Kraków
[1]*jakwojc@student.agh.edu.pl*, [3]*hubjancu@student.agh.edu.pl*

[2,4]*AGH Akademia Górniczo-Hutnicza im. Stanisława Staszica*
Wydział Inżynierii Mechanicznej i Robotyki,
Katedra Robotyki i Mechatroniki,
Al. Mickiewicza 30, 30-059 Kraków
[2]*twalczak@student.agh.edu.pl*, [4]*mikulski@student.agh.edu.pl*

SŁOWA KLUCZOWE: Sterowanie, samochód, mechatronika, aplikacja, Bluetooth, Arduino.

STRESZCZENIE

Sterowanie to niezbędny element każdego współczesnego pojazdu. W tym artykule rozważono różne implementacje systemu sterującego w samochodzie terenowym z zawieszeniem adaptacyjnym typu Tatra.

1. Wstęp

Pierwsze samochody posiadały sterowanie mechaniczne, czyli koła przekręcały się dzięki sile kierowcy wspomaganej przez układ hydrauliczny[1]. Aktualnie powszechnie stosowane jest sterowanie z elektronicznym wspomaganiem i powoli zbliżamy się do upowszechnienia nowego typu sterowania – autonomicznego[2], gdzie samochód steruje sam sobą, bez konieczności ingerencji kierowcy.

W tym artykule skupiono się na dwóch ostatnich sposobach oraz innych rozwiązaniach, które są od nich pochodne.

2. Cel i zakres pracy

Celem pracy jest wyposażenie pojazdu w mechatroniczny system sterowania, tak aby widoczne było działanie adaptacyjnego zawieszenia.

W rozdziale 1 przybliżono tematykę artykułu.

W rozdziale 2 przedstawiono cel i zakres pracy.

W rozdziale 3 dokonano przeglądu istniejących rozwiązań.

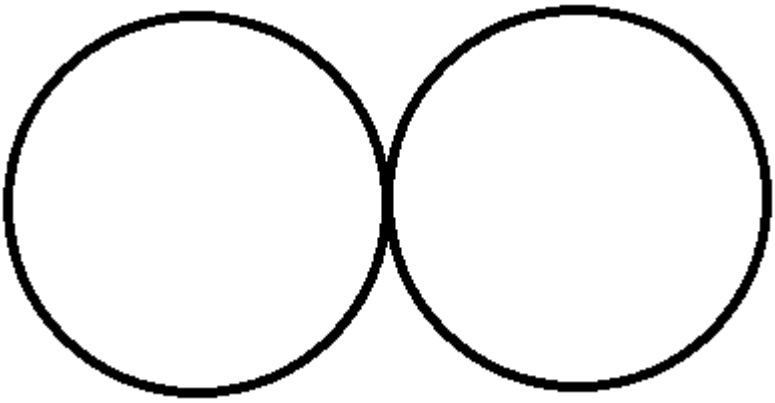
W rozdziale 4 przedstawiono wyniki pracy i zasadę działania stworzonego systemu sterowania.

W rozdziale 5 podsumowano pracę i przedstawiono wnioski z niej wynikające

3. Przegląd istniejących rozwiązań

3.1. Sterowanie poprzez zaprogramowanie trasy

Jeżeli trasa, na której będzie przeprowadzony test jest znana, można zrezygnować ze sterowania, które wymagałoby kierowcy. Można zaprogramować pojazd w taki sposób, aby poruszał się po wyznaczonej wcześniej trasie. Podobne rozwiązanie wykorzystuje się w robotach magazynowych[3], które poruszają się po wcześniej ustalonych trasach. W naszym przypadku rozważone było poruszanie się pojazdu po tzw. ‘ósemce’ (rys. 1.).



Rys. 1. Trasa w kształcie ósemki.

3.2. Sterowanie autonomiczne.

Sterowanie autonomiczne również nie wymaga ingerencji kierowcy, ale w odróżnieniu od poprzedniego rozwiązania, nie wymaga ono również znajomości toru. Jest jednak bardzo trudne do zaimplementowania w swojej właściwej formie, gdzie głównym czujnikiem jest kamera. Jednak możliwe było zaimplementowanie uproszczonej wersji, wykorzystującej głównie czujniki ultradźwiękowe. Podobne rozwiązanie jest wykorzystywane w samochodach, ale tylko jako wspomaganie dla kierowcy[4], jednak w małym pojeździe, którego celem nie jest dotarcie do specyficznego miejsca, może być wykorzystane jako ‘autonomiczny’ układ sterowania.

3.3 Sterowanie zdalne.

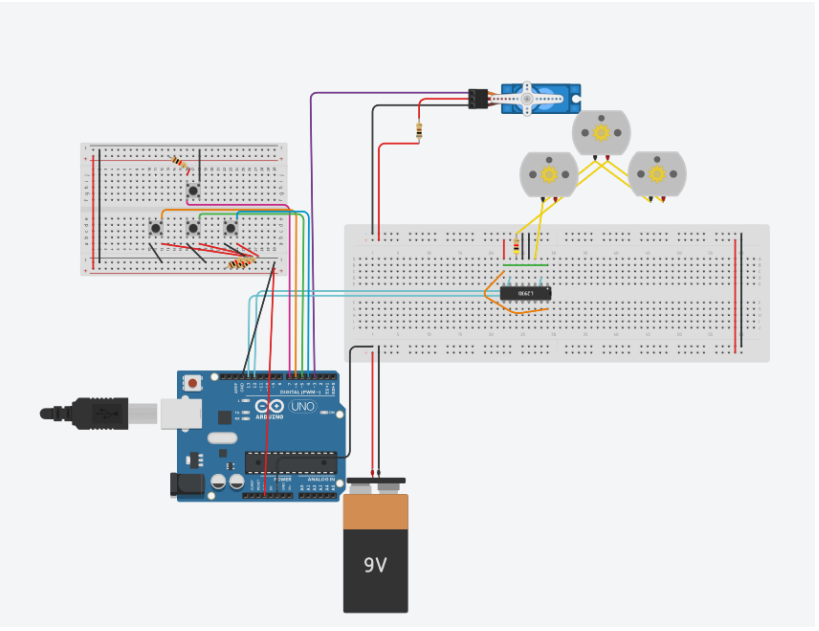
Kolejnym rozwiązaniem jest sterowanie pojazdem zdalnie poprzez pewnego rodzaju kontroler. W zdalnie sterowanych samochodach powszechnie stosowana jest komunikacja za pomocą fal radiowych. Spotykana jest też komunikacja poprzez Bluetooth[5]. Pierwsze jest trudne do implementacji i wymaga kupna kontrolera, drugie jest znacznie łatwiejsze do zaimplementowania i nie nosi ze sobą dodatkowych kosztów.

4. Modele i symulacje.

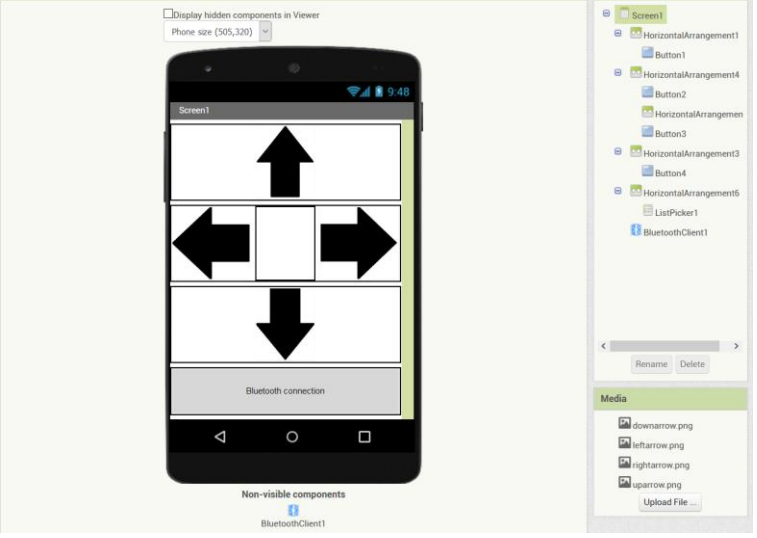
Ze względu na precyzje i łatwość implementacji tego systemu sterowania, zdecydowano wykorzystać sterowanie zdalne za pomocą aplikacji na telefon komunikującej się z pojazdem za pomocą technologii Bluetooth[5].

W tym celu utworzono aplikację na telefon oraz model elektroniki w programie Tinkercad (Rys. 2,3,4). Każda interakcja z przyciskiem w aplikacji

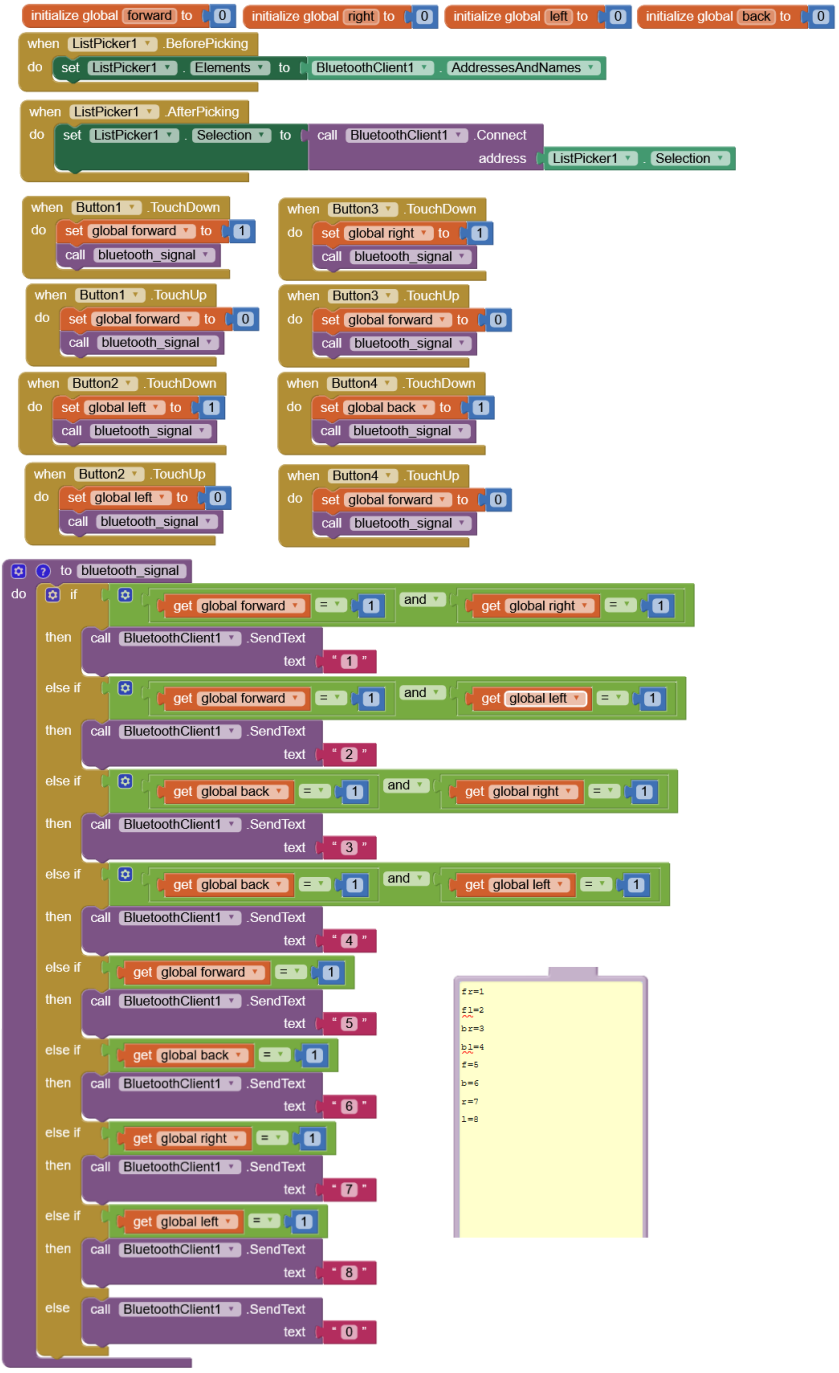
wysyła sygnał do Arduino[6], który jest w nim interpretowany na kierunek, w którym należy się poruszyć. Przy czym priorytetowe kierunki to przód i prawo, co oznacza, że przy wciśnięciu wszystkich przycisków jednocześnie, samochód zacznie jechać do przodu, skręcając w prawo.



Rys. 2. Model elektroniki



Rys. 3. Interfejs (front-end) aplikacji

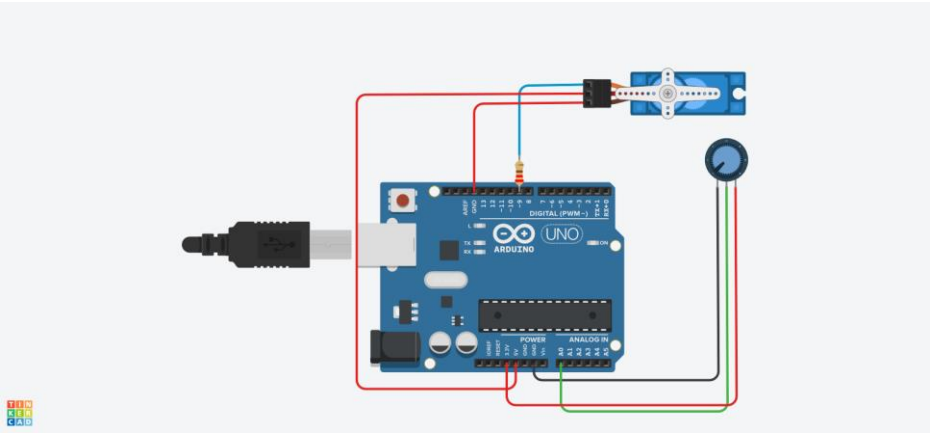


Rys. 4. Schemat blokowy, algorytm działania (back-end) aplikacji

4.1. Sterowanie lewo/prawo.

Zasada działania sterowania lewo/prawo wygląda następująco. Po wciśnięciu przycisku serwomotor obraca się w odpowiednim kierunku. Kiedy żaden przycisk nie jest wciśnięty, powraca on do neutralnego wychylenia. Podczas wracania można ponownie wcisnąć przycisk i serwomotor zacznie się od razu obracać w odpowiednią stronę. Umożliwia to utrzymywanie stałego skręcenia koła, poprzez szybkie wciskanie i puszczanie jednego z przycisków. Niestety, precyzyjność tego manewru w dużym stopniu zależy od sprawności kierowcy, niedoświadczony z tą metodą sterowania kierowca miałby duży problem z utrzymaniem kursu.

Rozważono również sterowanie za pomocą kierownicy, jednak było to niemożliwe do implementacji w aplikacji. Mimo tego, utworzono symulacje w programie Tinkercad (rys. 5.). Taki układ pozwoliłby na precyzyjne sterowanie pojazdem, nawet dla niedoświadczonego kierowcy.



Rys. 5. Sterowanie lewo/prawo za pomocą kierownicy/gałki.

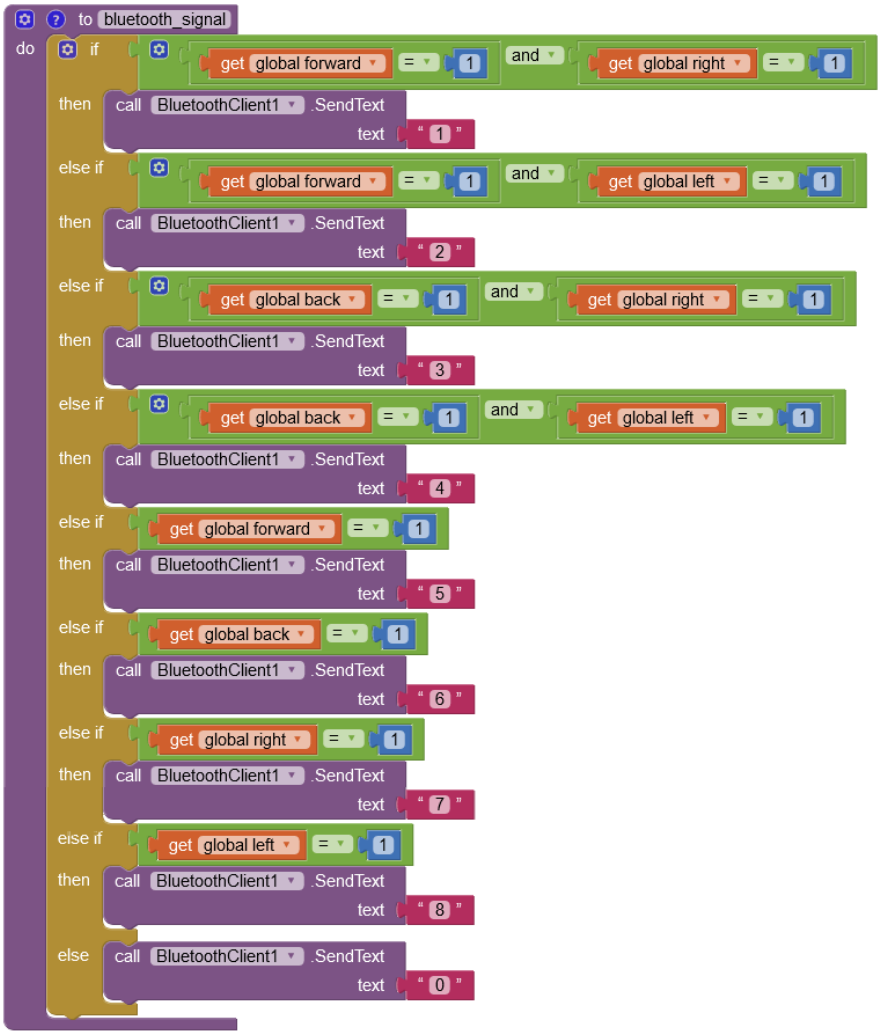
4.2. Sterowanie przód/tył.

Tutaj również wykorzystano przyciski, jednak tym razem uznano, że jest to optymalne rozwiązanie dla podjętego celu. Przy wciśnięciu przycisku trzy silniki napędowe zaczynają się obracać i pojazd jedzie do przodu lub do tyłu. Nie zaimplementowano hamulców, ponieważ uznano je za zbędne w tak małym pojeździe.

4.3. Aplikacja.

Aplikacja pozwala wybrać z listy urządzenie, z którym ma nastąpić komunikacja za pomocą technologii Bluetooth[5]. Działanie aplikacji opiera się

głównie na zmienianiu wartości czterech zmiennych, które są zinicjalizowane na początku schematu (rys. 4.), każda interakcja z przyciskiem niesie ze sobą zmianę wartości którejś z zmiennych i jednocześnie uruchamia funkcję bluetooth_signal (rys. 6.), która w zależności od wartości zmiennych wysyła odpowiedni sygnał do Arduino[6].



Rys. 6. Funkcja bluetooth_signal.

5. Podsumowanie i wnioski.

Dzięki wykonanej pracy, udało się zaimplementować w pojeździe system sterowania, który pozwala mu na poruszanie się we wszystkich kierunkach (z wyjątkiem góra/dół).

Największym wyzwaniem było stworzenie aplikacji. Początkowo rozważano napisanie aplikacji od zera, ale wymagało by to nauczenia się nowego języka programowanie, na co nie było wystarczająco czasu. Zdecydowano się skorzystać z gotowego szablonu, jednak w dalszym ciągu wymagane było zapoznanie się z zasadą działania technologii Bluetooth[5] i jak wchodzi ona w interakcje z Arduino[6]. Ostatecznie udało się stworzyć działającą aplikację (rozdział 4), dzięki której pojazd może być zdalnie sterowany.

Następnym możliwym krokiem byłaby zmiana sterowania lewo/prawo przyciskami na sterowanie kierownicą (zagadnienie to było poruszone w rozdziale 4.1.). Jak już wcześniej wspomniano, implementacja kierownicy nie jest możliwa w aktualnej aplikacji, więc potrzebne byłoby stworzenie nowej, prawdopodobnie napisanej w języku JavaScript.

Kolejnym potencjalnym usprawnieniem mogłoby być dodanie hamulców. Ze strony elektroniczno/softwarowej nie byłby to problem, ale wymagało by to konstrukcji mechanicznego systemu hamowania, co nie należy do zakresu tego artykułu.

Bibliografia

[1] Wikipedia.

https://pl.wikipedia.org/wiki/Wspomaganie_uk%C5%82adu_kierowniczego. [Dostęp: 6.06.2021].

[2] Wikipedia. https://en.wikipedia.org/wiki/Self-driving_car. [Dostęp: 6.06.2021].

[3] Mecalux. https://www.mecalux.pl/blog/roboty-magazynowe-logistyka-4-0. [Dostęp 6.06.2021].

[4] Motofocus. https://motofocus.pl/informacje/nowosci/82755/radary-samochodowe-jakie-maja-znaczenie-i-do-czego-sluza. [Dostęp: 6.06.2021].

[5] Bluetooth. https://www.bluetooth.com. [Dostęp: 6.06.2021].

[6] Arduino. https://www.arduino.cc. [Dostęp: 6.06.2021].