POZNAN UNIVERSITY OF TECHNOLOGY ACADEMIC JOURNALS
No 92 Electrical Engineering 2017

DOI 10.21008/j.1897-0737.2017.92.0033

Paweł IDZIAK*
Adam KMIEĆ*

STEROWANIE GŁOSEM URZĄDZENIAMI MECHATRONICZNYMI KONCEPCJA STANOWISKA DYDAKTYCZNEGO

W artykule zaprezentowano algorytmy zamiany głosu ludzkiego na postać cyfrową i na tej podstawie rozpoznawanie wydawanych komend. Przedstawiono opis algorytmu MFCC oraz jego aplikację działającą na platformie Raspberry Pi. Opisano spotykane open-source'owe programy umożliwiające rozpozanawanie mowy, działające w środowisku LINUX. Zaprezentowano koncepcję stanowiska dydaktycznego realizującego proste komendy głosowe. Przedstawiono rezultaty testów sprawdzających.

SŁOWA KLUCZOWE: Raspberry Pi, algorytm MFCC, oprogramowanie Jasper, rozpoznawanie mowy

1. WPROWADZENIE

Mowa jest podstawowym narzędziem ludzkiej komunikacji interpersonalnej. Umiejętność wytwarzania fal dźwiękowych oraz opracowanie szeregu sygnałów dźwiękowych zrozumiałych dla drugiego przedstawiciela tego samego gatunku umożliwia sprawną, praktycznie bezwysiłkową interakcję. Podstawowym ograniczeniem, w tego typu komunikacji „bezkontaktowej" jest zasięg emitowanych sygnałów oraz konieczność używania tych samych znaków (dźwięków złożonych z tzw. fonemów). Ten sposób komunikowania się z otoczeniem – nie tylko pomiędzy przedstawicielami tego samego gatunku – jest bardzo rozpowszechniony w świecie ziemskiej fauny.

Komfort, jaki stwarza porozumiewanie się za pomocą głosu, wpłynął również na postęp technologiczny. Doceniając zalety mowy, w tym potrzeby rozmowy na odległość powstał telefon. Badania nad wynalezieniem telefonu doprowadziły do stworzeniem mikrofonu, czyli urządzenia przetwarzającego sygnał akustyczny na serię impulsów elektrycznych. Był to pierwszy etap przetwarzania mowy ludzkiej na analogowy sygnał elektryczny a następnie cyfrowy [4].

* Politechnika Poznańska.

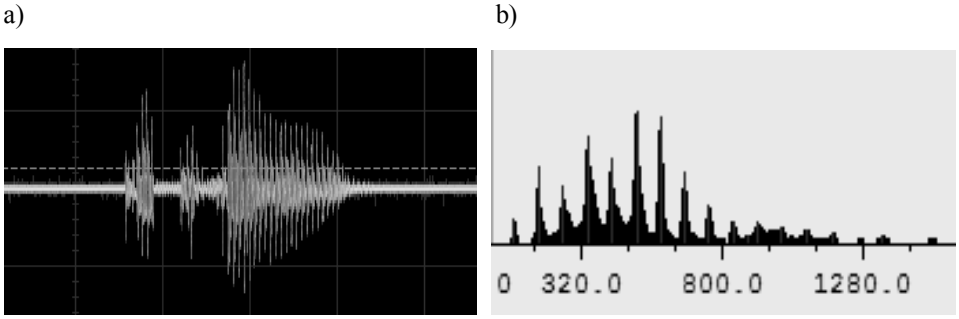
Oprócz rozmowy z innym człowiekiem, istnieje obszar, w którym taka forma komunikacji może przynieść równie wymierne korzyści – jest to interakcja z urządzeniami elektronicznymi.

Aktualnie istnieje realna możliwość wydawania poleceń urządzeniom i maszynom, bez konieczności używania rąk czy spoglądania na aparaturę. Przetwarzanie mowy oraz ciągły rozwój badań nad tym procesem, spowodowało również początek prac nad systemami weryfikacji osób na postawie ich głosu. Opierają się one na ekstrakcji indywidualnych cech głosu każdego człowieka.

W 2015 roku korporacja Google umożliwiła odblokowywanie telefonów komórkowych z systemem Android poprzez wypowiedzenie frazy „O.K. Google". Funkcja nadal nie gwarantuje jednak większej ochrony niż hasło. Aktualnie trwają prace nad jej ulepszeniem.

2. METODY ANALIZY SYGNAŁÓW GŁOSOWYCH

Transformata Fouriera (ang. *Fourier Transform*, FT) jest podstawowym narzędziem używanym w teorii analizy sygnałów, a w przypadku przetwarzania mowy umożliwia rozkład zarejestrowanego sygnału na funkcje proste o określonej częstotliwości, fazie i amplitudzie.



Rys. 1. Wizualizacja sygnału głosowego podczas wypowiadania komendy: a) zapis w dziedzinie czasu; b) reprezentacja transformaty Fouriera (dziedzina częstotliwości)

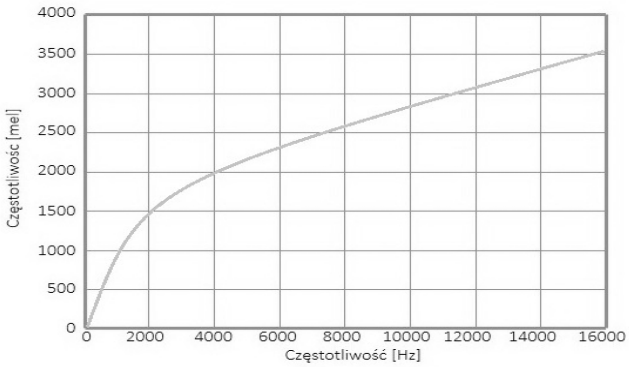
Graficznie jest to – zazwyczaj – prezentowane za pomocą szeregu prążków o różnych amplitudach. Każdy z prążków odpowiada zdefiniowanej wcześniej częstotliwości. Jego wysokość jest skorelowana, współczynnikiem wagi, z amplitudą danej częstotliwości. Uzyskany obraz przedstawia informacje o głównych składowych tworzących sygnał akustyczny, a także o składowych dodatkowych, takich jak na przykład szumy. W teorii przekształcenie sygnału jest odwracalne, a więc może on zostać przywrócony do swojej pierwotnej formy, bez ryzyka zaburzenia podstawowej struktury funkcji w reprezentacji czasowej [4].

2.1. Metoda melowo–częstotliwościowych współczynników cepstralnych

Warunkiem koniecznym poprawnego rozpoznania wydanych poleceń głosem, jest zastosowanie odpowiedniego sposobu przetwarzania fali akustycznej (głosu), tak, aby została ona zamieniona na informację zrozumiałą dla procesora. Służą temu odpowiednie algorytmy.

Najpowszechniejszym algorytmem stosowanym w rozpoznawaniu mowy, jest metoda melowo-częstotliwościowych współczynników cepstralnych (ang. *Mel–frequency Cepstral Coefficients*, w skrócie MFCC) [1, 7]. Bazuje ona na subiektywnej skali wrażeń słuchowych, odzwierciedlającej odbiór dźwięków przez człowieka [5].

Skala melowo-częstotliwościowa (słowo mel pochodzi od angielskiego słowa „melody" – melodia) jest nieliniowym odwzorowaniem dźwięku w zakresie częstotliwości słyszalnych [3, 5]. Uwzględnienia ona subiektywne odczucia słuchacza co do względnej odległości między poszczególnymi tonami dźwięku złożonego. Dźwięk złożony to dźwięk zawierający kilka – kilkanaście częstotliwości (tonów).



Rys. 2. Skala Mel

Omawiana skala została stworzona na podstawie danych otrzymanych w wyniku bardzo wielu eksperymentów. W trakcie ich trwania, badany miał za zadanie zestroić dźwięki tak, by w jego odczuciu, jeden był dwa razy wyższy od drugiego (tzn. aby częstotliwość jednego była dwukrotnie większa od drugiego). Zestawienie wszystkich punktów pomiarowych na wykresie, pozwoliło na graficzne przedstawienie nieliniowości obioru dźwięku przez człowieka. Powstała w ten sposób charakterystyka, która jest quasiliniowa w zakresie częstotliwości do 1 kHz, a powyżej tego progu przekształca ona się w skalę logarytmiczną.

Cepstrum jest wynikiem odwrotnej transformaty Fouriera z logarytmu widma sygnału. Jego nazwa pochodzi od anagramu słowa spektrum (ang, *spectrum* –

rozkład na częstotliwości składowe). Celem stosowania cepstrum jest wyznaczenie częstotliwości podstawowej dźwięku lub sygnału. Analiza cepstralna pozwala również na stwierdzenie, czy badany sygnał jest przebiegiem okresowym [6, 5, 8].

2.2. Algorytm MFCC

Algorytm mel-częstotliwościowy wymaga od użytkownika wykonania szeregu czynności wstępnych i porządkujących. Można w nim wyróżnić wydzielone bloki. Głównymi blokami są:

- przygotowanie nagrania,
- wykrycie początku komendy w przebiegu,
- ramkowanie/segmentacja,
- dyskretna transformata Fouriera,
- bank filtrów Mel,
- dyskretna transformata kosinusowa,

Standardowo przyjmuje się, że ludzki głos zawiera składowe w zakresie częstotliwości pomiędzy 300 Hz–3500 Hz. Z tego względu nagranie głosu w postaci cyfrowej powinno być sporządzone z częstotliwością próbkowania co najmniej dwukrotnie większą od częstotliwości granicznej górnej, tzn., w tym przypadku, z częstotliwością nie mniejszą niż 10 kHz [4]. Zagwarantuje to lepsze zachowanie szczegółów sygnału analogowego podczas konwersji na reprezentację cyfrową. Zwiększenie częstotliwości próbkowania zapobiega zniekształceniu pierwotnego sygnału spowodowanego niedostateczną ilością informacji o kształcie przebiegu [4].

W procesie rozpoznawania głosu dokonuje się wydzielenia tylko tej części sygnału, w której istnieją jakiekolwiek informacje dźwiękowe. Opisywany algorytm wymaga wcześniejszego ustalenia progu minimalnej amplitudy, która uznawana jest za sygnał aktywny. Jeżeli próg nie zostanie przekroczony to uznaje się to za ciszę.

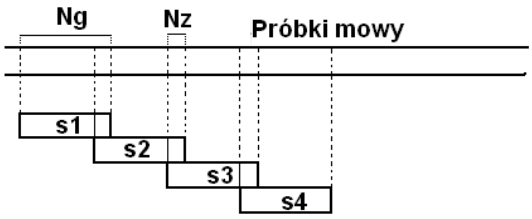
Z przebiegu zostają pobierane próbki, o określonej długości. W przypadku kiedy średnia amplituda sygnału jest większa od wartości ustalonej, próbka zostaje zapamiętana a analizie poddaje się kolejny przedział czasowy, nie większy od pierwszego. Jeśli średnia amplituda utrzymuje się na poziomie poniżej połowy początkowego progu, dźwięk nadal klasyfikuje się jako ciszę. W przeciwnym przypadku program uznaje impuls za początek lub koniec aktywnej mowy [10].

Kolejnym blokiem jest moduł ramkowania (segmentacja).

Sygnał mowy jest sygnałem wolno zmieniającym się w czasie o przebiegu bliskim przebiegowi okresowemu o okresie do 30 ms (sygnał quasi–stacjonarny). Jeżeli zostanie on podzielony, w dziedzinie czasu, na ramki o długości trwania pomiędzy 20 a 40 milisekund, to w tych przedziałach (ramkach) jego cechy uznaje się za lokalnie niezmienne.

Ramki o długości N_g składające się z N próbek zachodzą na siebie tak, że posiadają część wspólną z poprzednią i następną ramką (obszar N_z) – rys. 3. Nie dotyczy to pierwszej i ostatniej części sygnału. Każdy taki przedział odpowiada kolejnemu oknu czasowemu [11].

W literaturze światowej i krajowej nie ma zgodności w zakresie nazewnictwa. Wspomniane wcześniej przedziały (ramki) zwane są również segmentami. Stąd operacja podziału sygnału czasowego na przedziały, zwana jest zamiennie ramkowaniem (ang. *framing*) lub segmentacją [4, 8].



Rys. 3. Sekwencja segmentacji

Kolejnym krokiem procesu identyfikacji mowy, jest tzw. okienkowanie każdej ramki sygnału. Ma to zapewnić zminimalizowanie wpływu rozmycia widma (często w literaturze pojawia się termin przecieku widma), które powstaje podczas analizy sygnału analogowego z pomocą dyskretnej transformaty Fouriera. Proces rozmycia polega na pojawieniu się oprócz częstotliwości podstawowej tzw. wstęg bocznych (listków bocznych).

Przeciek widma zakłóca właściwą interpretację sygnału. Wprowadza dodatkowe niezerowe składowe w częstotliwościach innych niż faktyczne częstotliwości przebiegu.

Podczas przetwarzania sygnału głosu wystąpienie rozmycia widma jest **nieuniknione**, co w algorytmie MFCC wymusza stosowanie np. okna Hamminga. Dzięki temu następuje wyostrzenie głównej składowej, natomiast znaczenie listków bocznych widma zostaje zniwelowane. Ma to duże znaczenie dla dokładności rozpoznawania mowy [3].

W cyfrowym przetwarzaniu sygnałów mowy stosuje się odmianę transformaty Fouriera – dyskretną transformatę Fouriera (DFT). Pozwala ona na określenie częstotliwości dla wybranych punktów przebiegu zamiast wyznaczania jej dla całego sygnału. Jednak obliczanie DFT bezpośrednio z zależności matematycznych jest mało efektywne z uwagi na zbytnią złożoność obliczeniową. Usprawnieniem tego procesu jest szybka transformata Fouriera (ang. *Fast Fourier Transform*, FFT) [5].

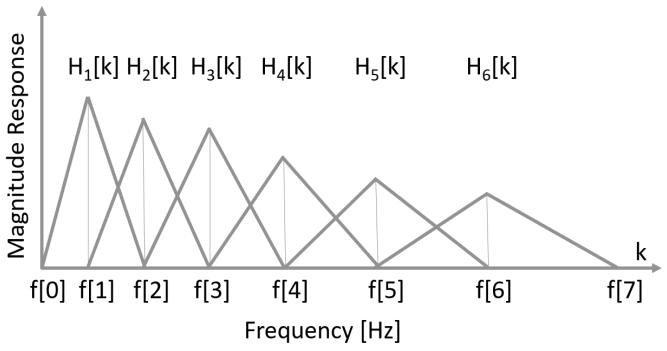
Jednym z głównych założeń przetwarzania ludzkiej mowy, w kontekście wydawania poleceń, jest czas reakcji procesora na otrzymaną komendę. Po uzyskaniu informacji w postaci sygnału akustycznego, wykonanie przez jednostkę

sterującą przypisanych jej działań w jak najkrótszym odstępie czasu, jest uwarunkowane wydajnością metody obliczeniowej. Zastosowanie szybkiej transformaty Fouriera w algorytmie MFCC zatem jest uzasadnione.

Szybka transformata Fouriera wykorzystuje założenie, że wykonanie obliczeń części przebiegu funkcji, a następnie połączenie ich w jeden ciąg, jest wydajniejsze niż stosowanie algorytmu dla całości sygnału. Możliwe jest więc podzielenie wielopunktowego przebiegu na transformaty dwupunktowe. Będą one osobno obliczone, a następnie połączone w coraz to większe agregaty, aż do uzyskania pełnego obrazu funkcji.

Procedura MFCC opiera się na podobieństwie do działania percepcji ludzkiego ucha. Jego czułość nie jest stała we wszystkich zakresach. Największą wrażliwość ucho człowieka wykazuje dla sygnałów akustycznych w przedziale do 3000 Hz [4]. W tym zakresie znajdują się również częstotliwości wytwarzane podczas przeciętnej rozmowy. Mają więc one największe znaczenie w procesie rozpoznawania komend głosowych.

Skala Mel jest ukazaniem nieliniowości odbioru fal akustycznych przez człowieka. Stanowi ona podstawę tworzenia banków filtrów Mel, których zadaniem jest wyeksponowanie przebiegów typowych dla mowy ludzkiej, oraz odseparowanie dźwięków, o wyższych częstotliwościach, mogących zakłócić poprawną interpretacje i analizę nagranego dźwięku. Reprezentacją filtrów odpowiadającym odpowiednim pasmom częstotliwości są nachodzące na siebie trójkątne charakterystyki przejścia. Różnią się one od siebie długością podstawy i przypisaną wagą. Waga określa stopień istotności określonej części widma i wpływa na wyznaczenie współczynników mel-cepstralnych analizowanego sygnału.



Rys. 4. Bank filtrów Mel; $H_i[k]$ – współczynniki wagi

W analizie FFT sygnał reprezentowany jest przez szereg prążków o różnej amplitudzie i częstotliwości. Każda wartość prążka jest mnożona przez liczbę przypisaną dla danego filtru. Wyniki są dodawane do siebie w obrębie każdego

trójkąta w taki sposób, że każdy jest reprezentowany przez jedną wartość. Z uwagi na to, że charakterystyki nakładają się wzajemnie, istnieje prawdopodobieństwo pojawienia się prążka w części wspólnej filtrów. Taki przypadek zostaje rozpatrzony osobno dla każdego z trójkątów [10].

Krokiem wieńczącym przetwarzanie widma przez bank filtrów jest obliczenie logarytmu otrzymanych wartości. Powoduje to znaczące zredukowanie czułości na bardzo głośne i bardzo ciche dźwięki, co przekłada się na wzrost dokładności rozpoznawania głosu.

Ostatnim zabiegiem algorytmu MFCC jest konwersja logarytmu widma mel z powrotem do dziedziny czasu. Ponieważ wartości otrzymane z banku filtrów mel oraz ich logarytmy są liczbami rzeczywistymi, możliwe jest osiągnięcie tego przy pomocy dyskretnej transformaty kosinusowej, Wynikiem tej operacji jest otrzymanie współczynników mel-cepstralnych opisujących kluczowe elementy charakterystyczne dla wypowiadanego słowa [1, 5, 6, 7].

3. ZAŁOŻENIA DO STANOWISKA DYDAKTYCZNEGO

Przyjęto, że planowane do budowy stanowisko powinno wspierać zajęcia dydaktyczne z obszaru programowania i sterowania urządzeniami mechatronicznymi prowadzonymi dla słuchaczy kierunku Mechatronika. Oznacza to, że uczestnikowi zajęć znane są pojęcia i techniki związane z procesem programowania układów mikroprocesorowych. Stanowisko powinno być konstrukcją nisko–budżetową.

Na jednostkę sterująco-obliczeniową wybrano platformę Raspberry PI. Jest platforma korzystająca, przede wszystkim z oprogramowania open-source; w tym przypadku środowiska LINUX. Duża moc obliczeniowa procesora, znaczna pojemność pamięci RAM, możliwość przenoszenia oprogramowania z jednej jednostki na inną za pośrednictwem karty Micro SD, bogate wyposażenie w porty komunikacyjne kompensują brak sprzętowego terminala użytkownika.

Pewną niedogodnością jest brak kompatybilności platformy Raspberry Pi z programami stworzonymi na komputery osobiste. Wynika to z odmiennej architektury zastosowanych procesorów. Jak wykazały prowadzone eksperymenty nie wszystkie programy pozwalają się uruchomić w obu wspomnianych środowiskach.

Oprogramowanie Jasper

W trakcie prac wstępnych nad projektem zwrócono uwagę na program open-source o nazwie Jasper. Sposób działania tego programu jest następujący: komenda wypowiedziana przez człowieka jest przez program przekazywana, w postaci sygnału akustycznego, do oprogramowania przetwarzającego mowę

na tekst. Tam jest ona analizowana za pomocą algorytmów rozpoznawania głosu, takich jak opisany wcześniej algorytm MFCC.

W przypadku przetwarzania mowy przez zaawansowane „silniki programistyczne" STT (ang. *speech–to–text*) wykorzystuje się sztuczne sieci neuronowe, czego przykładem jest oprogramowanie Google Cloud Speech API.

Po analizie nagrania mowy, informacja zwrotna jest przekazywana w postaci tekstu. Jest ona sprawdzana przez program *Jasper*, w celu porównania jej ze słowami kluczowymi, które aktywują wybrane moduły programistyczne. Jeżeli komenda nie jest rozpoznana program informuje o tym użytkownika.

Silniki STT

Jasper oferuje integrację z kilkoma silnikami przetwarzania mowy ludzkiej na tekst. Jednym z nich jest oprogramowanie open-source o nazwie Pocketsphinx, dołączane dodatkowo w pakiecie instalacyjnym.

Po serii testów odrzucono stosowanie tego silnika ze względu na bliską zeru rozpoznawalność wypowiadanych słów. Język angielski, jest jedynym językiem rozpoznawanym przez algorytm stosowany w tym STT.

Drugim rozpatrywanym silnikiem STT był program o nazwie *Julius*. Jest on również rozpowszechniany w oparciu o licencję open-source. Nie jest on dołączany razem z programem *Jasper*. Wymaga on osobnej instalacji oraz stworzenia własnego modelu akustycznego w oparciu o oprogramowanie VoxForge. Tworzenie modelu akustycznego danej osoby jest czasochłonne. Wymaga wielu kroków, w tym nagrania co **najmniej** 40 sekwencji słów o określonych parametrach i zapisaniu ich w postaci pliku o rozszerzeniu .wav. Następnie nagrania te są wielokrotnie przetwarzane w celu określenia zależności pomiędzy fonemami a wypowiedzianymi komendami [8]. W końcowym etapie tworzenia modelu akustycznego, jest definiowana baza głosu, która może zostać wykorzystana przez program *Julius*.

Po wykonaniu prób z udziałem tego silnika STT, z uwagi na znikomą rozpoznawalność słów, odstąpiono od prób jego aplikacji. W trakcie analizowania możliwych błędów tworzenia modelu akustycznego odnaleziono adnotację dodaną przez twórców VoxForge. Zawarto w niej zalecenie stworzenia **wielogodzinnych nagrań** w celu poprawienia trafności rozpoznawania mowy.

Wymienione wyżej silniki to programy nie wymagające połączenia z Internetem, co oznacza, że są dostępne lokalnie z poziomu Raspberry Pi. Może to zostać uznane za zaletę, jednak brak wystarczającej trafności przekształcania mowy na tekst, eliminuje możliwość ich udziału w eksperymentach prowadzonych podczas zajęć dydaktycznych.

Istnieją również algorytmy STT, które znajdują się na serwerach. Wymusza to jednak ciągłe połączenie jednostki sterująco-liczącej z siecią. Zaletą takiego rozwiązania jest jednak znaczący wzrost skuteczności rozpoznawania mowy.

Dodatkowym usprawnieniem jest wykonywanie złożonych obliczeń poza jednostką Raspberry Pi. Proces przekształcania mowy na tekst zostaje więc przyspieszony, a procesor jednostki może wykonywać równolegle dodatkowe zadania. Do dalszy testów wybrano Google Speech API w wersji beta. Przez kilka lat projekt był dostępny bezpłatnie dla wszystkich użytkowników. Pozwoliło to na testowanie oraz ulepszanie algorytmów rozpoznawania głosu, dzięki ogromnej liczbie nagrań. Od niedawna projekt ten jest zamknięty, a korzystanie z oprogramowania znajdującego się na serwerach Google wymaga opłaty licencyjnej. Bezpłatny dostęp jest możliwy jedynie przez 60 dni.

Narzędzie przetwarzania mowy na tekst stworzone przez Google posiada jednak znaczące ograniczenie. Dla wersji próbnej możliwe jest wysłanie tylko 50 zapytań (komend) na dobę. Problemem jaki wynika z ograniczonej liczby interakcji z serwerem, jest krótki czas możliwego wykorzystania urządzenia Raspberry Pi do sterowania odbiorników za pomocą komend głosowych. Wynika to z faktu, że program *Jasper*, po uruchomieniu, pobiera próbki dźwięku z mikrofonu w odstępach około 2–u sekundowych, po czym przekazuje je do serwera. W chwili transmisji nagrania do Google Speech API nie jest istotne, czy pakiet danych zawiera w sobie jakikolwiek dźwięk. Oznacza to, że każda, nawet niewykorzystana w procesie przekształcania mowy na tekst próbka dźwięku, jest zaliczana przez serwer jako zapytanie. Pomimo bariery, którą stwarzają limity narzucone przez twórców przedstawianego oprogramowania, silnik Google Speech API został uznany za najskuteczniejszy.

Interakcja z urządzeniami odbiorczymi

Po procesie konwersji komendy głosowej, tekst zostaje przesłany z powrotem do Raspberry Pi. Program *Jasper* porównuje to z bazą plików wywołujących określone działanie jednostki sterującej. Bazę można rozszerzać o dodatkowe moduły, a więc tworzyć spersonalizowane zastosowania. Wadą przedstawionego silnika programistycznego jest brak modułu pozwalającego aktywować poszczególne wejścia/wyjścia GPIO platformy Raspberry Pi.

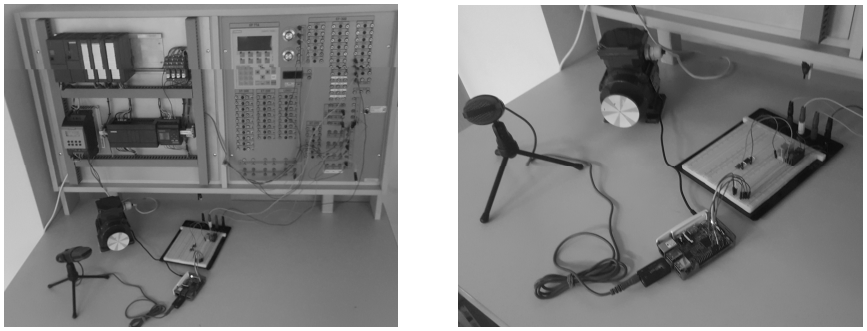
Wadą przedstawionego programu jest brak modułu pozwalającego aktywować poszczególne wejścia/wyjścia GPIO platformy Raspberry Pi.

W ramach niniejszego projektu opracowano niezbędny moduł. Dla celów poglądowych wybrane wyjścia platformy obciążono wtórnikami napięciowymi oraz diodami LED. Aktywacja poszczególnych diod oznacza równocześnie aktywację wyjść odpowiednich wtórników sterujących pracą przekaźników. Z kolei przekaźniki sterują odpowiednimi portami wejściowymi inwertera zasilającego silnik indukcyjny małej mocy. Aktywowanie odbiorników następuje po wypowiedzeniu komendy wspólnej dla wszystkich odbiorników. Rysunek 5 przedstawia stanowisko do testowania opisywanego projektu.

Konfiguracja układu rozpoznawania mowy

Do zbudowania układu do głosowego sterowania urządzeniami mechatronicznymi poprzez platformę Raspberry Pi 3, konieczne jest doposażenie układu w mikrofon z wbudowaną kartą dźwiękową. W projekcie wykorzystano mikrofon pojemnościowy. System operacyjny Raspbian w wersji Jessie zapisano na karcie microSD. Wyjścia GPIO Raspberry Pi podłączono, za pośrednictwem płytki stykowej, ze wspomnianymi odbiornikami.

a) b)



Rys. 5. Stanowisko z zespołem sterowników i inwerterem; a) widok ogólny;
b) płyta stykowa z przekaźnikami, moduł Raspberry i nieruchomy silnik indukcyjny

Rozpoczęcie użytkowania Raspberry Pi wymaga zainstalowania systemu operacyjnego. Dostępny jest on na oficjalniej stronie producenta. W projekcie użyto systemu Raspbian. Plik instalacyjny programu *Jasper* możliwy jest do pobrania poprzez stronę założoną przez jego twórców w domenie github.io . *Jasper* został opracowany w 2013 roku dla Raspberry Pi w wersji B, z zainstalowaną wcześniejszą wersją Raspbiana. Z tego powodu konfiguracja wymaga sporej ingerencji operatora na etapie uruchamiania np. doinstalowania bibliotek. Brak informacji o błędach konfiguracyjnych świadczy, w tym przypadku, o pomyślnie zakończonym procesie przygotowania programu do użytku.

W kolejnym etapie należy dodać moduł programu, który będzie wywoływał kod odpowiedzialny za aktywowanie wyjść GPIO. Program wywołujący porównuje komendy otrzymane z silnika STT, w postaci tekstu, ze słowami aktywującymi, które wcześniej należy zadeklarować.

W projekcie przyjęto, że słowem aktywującym będzie fraza „start”.

Osobnym programem jest moduł wywołujący wyłączenie inwertera. Aby to osiągnąć należy zmienić deklarację słów użytych w komendzie oraz wskazać lokalizację skryptu sterującego wyjściem GPIO. Należy również podać zwrot, jaki zostanie wypowiedziany przez syntezator mowy zawarty w oprogramowaniu *Jasper*. Ułatwi to rozpoznanie, czy został załączony pożądany moduł.

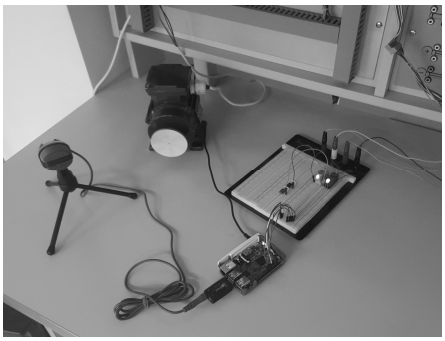
Należy zwrócić uwagę, że pod względem znaczeniowym frazy użyte do aktywacji i dezaktywacji odbiornika mogą być całkowicie ze sobą niezwiązane.

Zakończenie procesu przygotowania programu do działania wymaga dodatkowo pobrania biblioteki o nazwie Wiring Pi i przygotowania stosownego skryptu. Biblioteka ta jest udostępniana na mocy licencji GNU (ang. General Public License).

Przykładowe skrypty załączenia i wyłączenia wyglądają następująco:

```
#!/bin/bash
/usr/local/bin/gpio mode 4 out
/usr/local/bin/gpio write 4 on
#!/bin/bash
/usr/local/bin/gpio write 4 off
```

Rysunek 6 przedstawia stan stanowiska po wydaniu komend START, a następnie RUN (ruch silnika w kierunku zgodnym z ruchem wskazówek zegara).



Rys. 6. Stan po wydaniu komend START i RUN

4. PODSUMOWANIE

Opracowane autorskie moduły dopełniające oprogramowanie sterujące pracą platformy Raspberry, dostępne w trybie open-source, oraz przeprowadzone testy, wykazały, że korzystając z tanich elementów składowych możliwe jest realizowanie systemów mechatronicznych sterowanych ludzkim głosem. W przyjętej koncepcji niezadowalające pozostają stopień identyfikacji komend oraz czas reakcji na te komendy. Aktualnie funkcjonująca wersja sterowania umożliwia uruchomienie silnika, zadanie i zmianę kierunku wirowania oraz jego zatrzymanie. Przewiduje się rozbudowanie funkcji sterowania o komendy zmieniające skokowo prędkość obrotową silnika. Struktura stanowiska jest otwarta zatem możliwe się dalsze jej zmiany. Zadania do zrealizowania podczas zajęć wymagać będą od ćwiczącego znajomości struktury programu w języku Phyton

oraz opracowania własnych komend sterujących. Ze względu na rozmiary Raspsberry Pi oraz powszechny dostęp do bezprzewodowego Internetu, sterowanie głosowe może znaleźć zastosowanie w wielu różnych obszarach i dziedzinach życia człowieka

LITERATURA

[1] Dhigra S., Nijhawan G., Pandit P., Isolated speech recognition using MFCC and DTW, International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering Vol. 2, 2013 r.
[2] Ellis P.W.D An introduction to signal processing for speech LabROSA, Columbia University, New York 2008 r., pp. 25–28.
[3] Ittichaichareon C., Suksri S., Yingthawornsuk T., Speech Recognition using MFCC, International Conference on Computer Graphics, Simulation and Modeling (ICGSM'2012) Pattaya , 2012 r.
[4] Florek A., Mazurkiewicz P., Sygnały i systemy dynamiczne, Wydawnictwo Politechniki Poznańskiej, Poznań, 2015 r.
[5] Kacprzak S., Inteligentne metody rozpoznawania dźwięku, Politechnika Łódzka, 2010r.
[6] Holmes J., Holmes W., Speech Synthesis and Recognition, Taylor & Francis, New York, 2001 r., pp. 161–164.
[7] Lindberg B., Zeng–Hua T., Automatic Speech Recognition on Mobile Devices and over Communication Networks Springer–Verlag, London, 2008, pp. 72 –74.
[8] Sukork S., Speaker identyfication system using MFCC procedure and noise reduction method, Univeristy Tun Hussein Onn, Malaysia, 2012.
[9] https://cloud.google.com.speech/?gclid=CJX9wN G1tECFFU1fGQodds0Ojg (dostęp 23.01.201).
[10] http://file.scirp.org/Html/4–6801198_34183.htm (dostęp 21.01.2017).

THE VOICE CONTROL OF MECHATRONIC DEVICES THE CONCEPT OF DIDACTIC STATION

The article features basic algorithms which are responsible for converting human voice into digital form . It also describes MFCC algorithm and the steps required to put it into practice. It includes presentation of the primary open-source software programs, that allow speech recognition in Linux environment, on the platform Raspberry Pi. At the end, the article presents a concept of didactic station, performing simple voice commands using Jasper program and its possibility to use in future.

(Received: 06. 02. 2017, revised: 21. 02. 2017)