

PRECYZYJNE STEROWANIE SILNIKIEM BLDC

Streszczenie

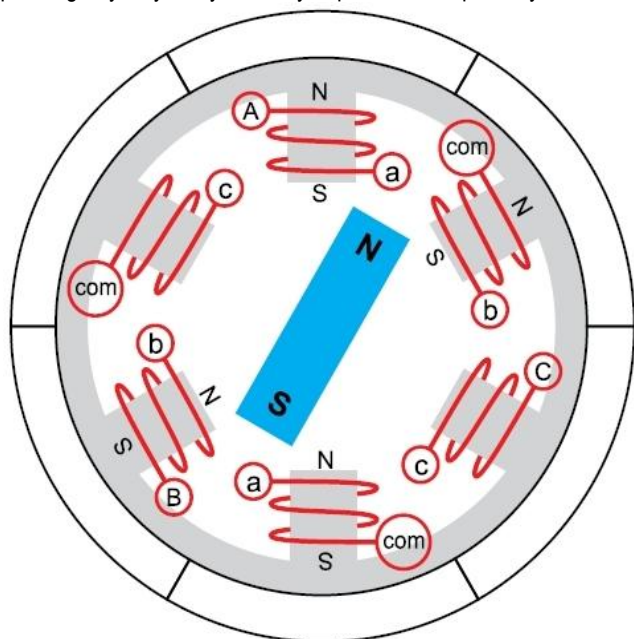
W artykule omówiono sterowanie silnika BLDC, które jest wykorzystywane przy układzie stabilizacji głowic obserwacyjnych BSP. Celem badań było opracowanie algorytmu sterowania, określenia ograniczeń i możliwości tego silnika. Jak również opracowanie modelu symulacyjnego i rzeczywistego.

WSTĘP

W głowicach obserwacyjnych bezzałogowych statków powietrznych coraz częściej wykorzystywane są silniki BLDC. Pozwalają one na bardzo precyzyjne sterowanie zarówno prędkością, jak i położeniem. Wysoka rozdzielczość kąta pozwala na pozbycie się wszelkiego rodzaju przekładni z mechaniki głowicy. Brak przekładni upraszcza mechanikę, zmniejsza koszty produkcji i pozwala na płynną pracę układu.

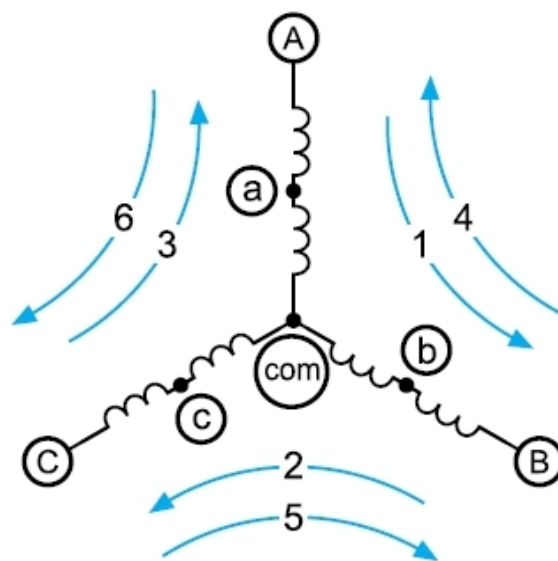
1. BUDOWA I ZASADA DZIAŁANIA SILNIKA BLDC [1]

Silnik BLDC składa się z wirnika z magnesami stałymi oraz stojana z uzwojeniami. Budowa silnika BLDC przedstawiona jest na rysunku 1.1. Moment obrotowy powstaje w wyniku oddziaływania pól magnetycznych wytwarzanych przez te komponenty.



Rys. 1.1. Budowa silnika BLDC

Silnik ten posiada 3 fazy (A,B,C) połączone ze sobą w gwiazdę. Sterowanie odbywa się poprzez odpowiednie ukierunkowanie prądu w uzwojeniach. Kolejne komutacje wraz z kierunkami przepływu prądu przedstawione zostały na rysunku 1.2.



Rys. 1.2. Schemat działania silnika BLDC

2. STEROWANIE SILNIKIEM BLDC

We wcześniej wspomnianych głowicach obserwacyjnych silnik BLDC traktowany jest jako silnik krokowy i właśnie w ten sposób jest on sterowany.

Obrót silnika wymuszony jest poprzez podanie kolejnych komutacji. Regulacja prędkości realizowana jest zmianą częstotliwości przełączania komutacji (im wyższa częstotliwość tym większa prędkość obrotowa).

Moment zależy od natężenia strumienia magnetycznego, który można płynnie regulować poprzez zmianę prądu. Zmiana ta realizowana jest poprzez zmianę współczynnika wypełnienia sygnału PWM, który z kolei uśrednia napięcie.

Kąt o jaki obrócił się silnik można wyliczyć znając liczbę magnesów i pól znajdujących się w silniku. Im większa jest ilość magnesów tym można uzyskać wyższą rozdzielczość kątową.

2.1. Sterowanie pełnokrokowe

Omawiając silnik BLDC, sterowanie pełnokrokowe jest niczym innym jak podawaniem kolejnych sześciu komutacji. Silniki BLDC posiadają stosunkowo niewielką liczbę magnesów. Silnik wykorzystywany do badań posiada 14 magnesów i 12 pól. Kąt między komutacjami można wyliczyć z następującej zależności.

$$\alpha = 60^\circ \frac{3}{lp} \quad (1)$$

gdzie:

α – kąt o jaki obrócił się wirnik,

lp – liczba pól silnika

Badany silnik ma kąt obrotu między komutacjami równy 15° . Sterowanie to charakteryzuje się dużymi drganiami podczas pracy. W celu uzyskania wyższej rozdzielczości kątowej rzędu ułamka stopnia należy krok ten podzielić. Dlatego wykorzystuje się sterowanie mikrokrokowe.

2.2. Tekst Sterowanie mikrokrokowe

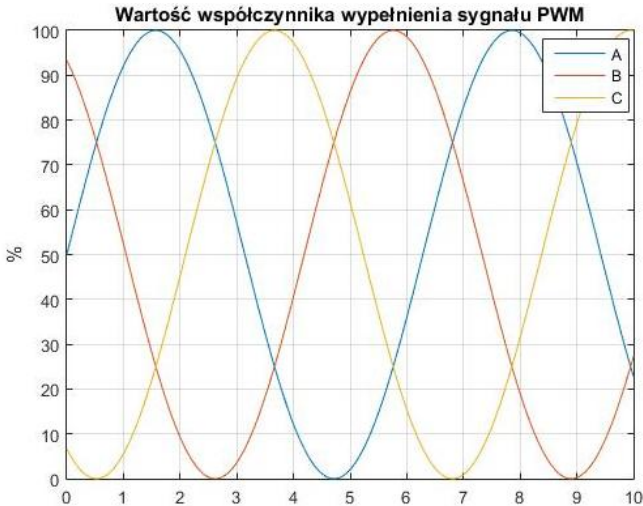
Sterowanie mikrokrokowe pozwala na uzyskanie bardziej płynnego obrotu niż w przypadku sterowania pełnokrokowego oraz zapewnia stały moment obrotowy.

Do celów symulacji został zaprojektowany poniższy model w środowisku Matlab/Simulink.

Model symulacyjny przedstawiony na rysunku 2.1. składa się z trzech podsystemów:

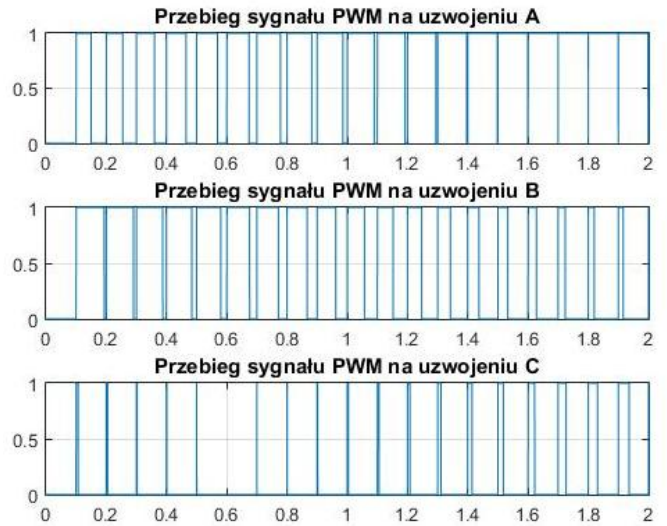
1. SINUS – odpowiada on za generowanie sygnałów sinusoidalnych;
2. PWM_generator – generuje przebieg sygnału PWM;
3. Data – wysyła dane wynikowe do przestrzeni „workspace”.

Obracanie polem magnetycznym polega na płynnej regulacji prądu w uzwojeniach. Sprzętowo realizowane jest to za pomocą sygnału PWM, który uśrednia napięcie na każdej z faz. Wartość współczynnika wypełnienia PWM przedstawiają rysunki 2.2., 2.4. i 2.5.



Rys. 2.2. Wykres wartości współczynnika wypełnienia

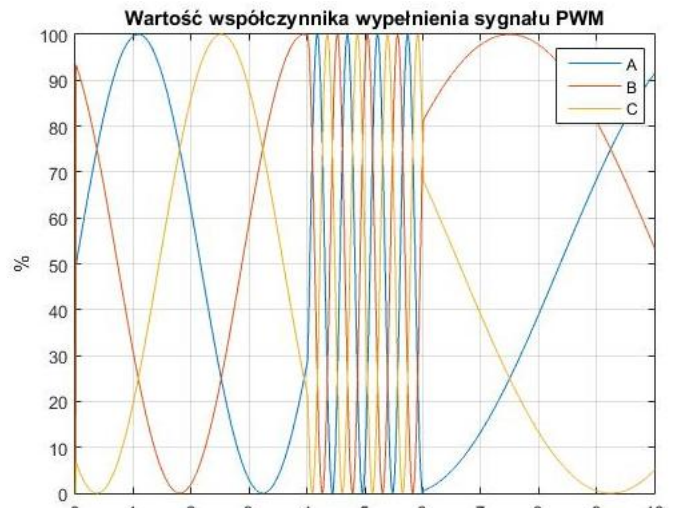
Częstotliwość sinusa jest stała, oznacza to, że silnik będzie poruszał się ze stałą prędkością. Natomiast przesunięcie fazowe między nimi wynosi 120° . Rysunek 2.3. przedstawia sygnały PWM, które będą podawane na część mocy układu.



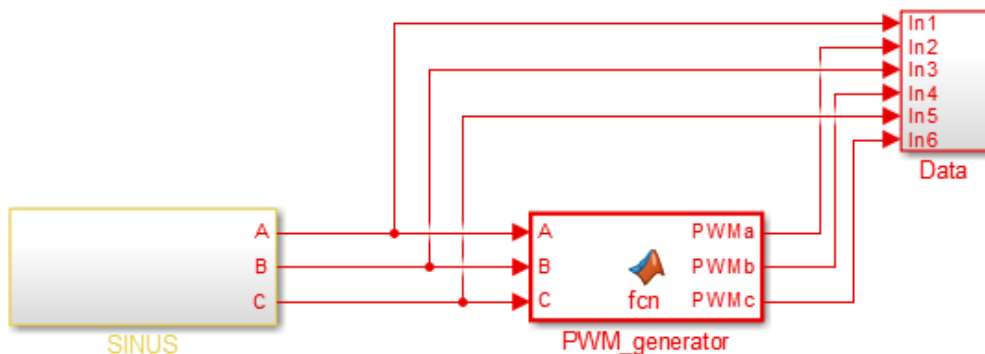
Rys. 2.3. Przebiegi sygnału PWM

Na powyższych wykresach widać jak zmienia się szerokość impulsu na każdej z faz. Można także zauważyć przesunięcie w fazie między uzwojeniami.

Poniżej zostały przedstawione wyniki symulacji pracy układu ze zmienną prędkością:



Rys. 2.4. Wartość współczynnika wypełnienia sygnału PWM dla zmiennej prędkości obrotowej



Rys. 2.1.. Model symulacyjny

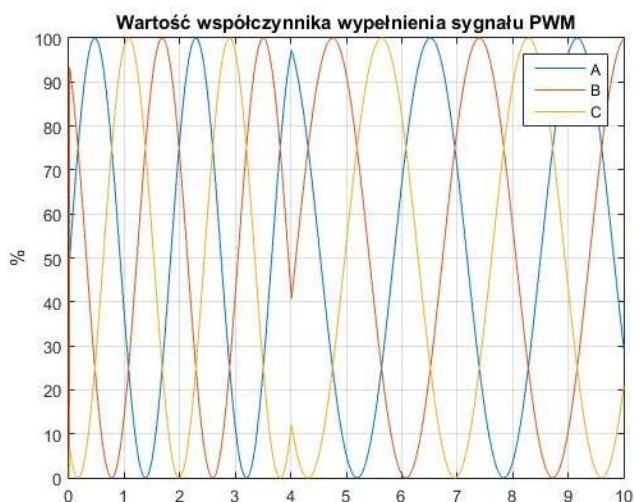
Zmiana prędkości obrotowej wiąże się ze zmianą częstotliwości funkcji sinus. Ze względu na fakt, że omawiane sterowanie jest bezczujnikowe, istnieje prędkość graniczna, którą można zadać. Silnik po jej przekroczeniu przestaje się poruszać i wpada w drgania. Brak synchronizacji w tym momencie powoduje błędy w naliczaniu kąta. Wartość prędkości granicznej zależy od następujących czynników:

- wykorzystywanego silnika,
- napięcia zasilania,
- maksymalnej wartości współczynnika PWM.

W celu eliminacji tego zjawiska należy ograniczyć prędkość zadaną w zależności od w/w parametrów.

W celu zmiany kierunku obrotów silnika należy odwrócić funkcję sinus.

Na rysunku 2.5. przedstawiono wyniki symulacji pracy układu ze zmianą kierunku w 4. sekundzie pracy:



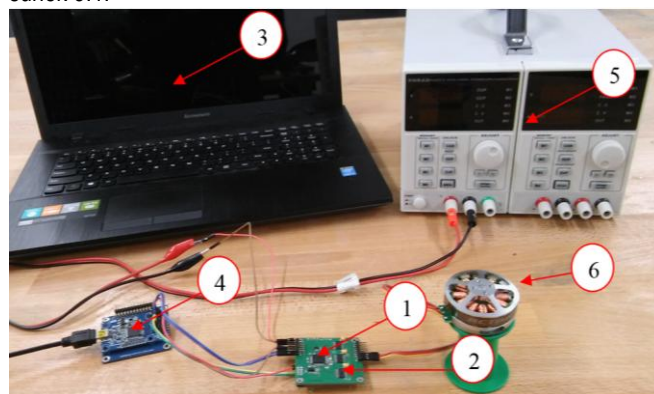
Rys. 2.5. Wartość współczynnika wypełnienia sygnału PWM ze zmianą kierunku

3. PRACA NA UKŁADZIE RZECZYWISTYM

W celu sprawdzenia poprawności działania modelu symulacyjnego, został zaprojektowany i wykonany model rzeczywisty. Składa się on z następujących elementów:

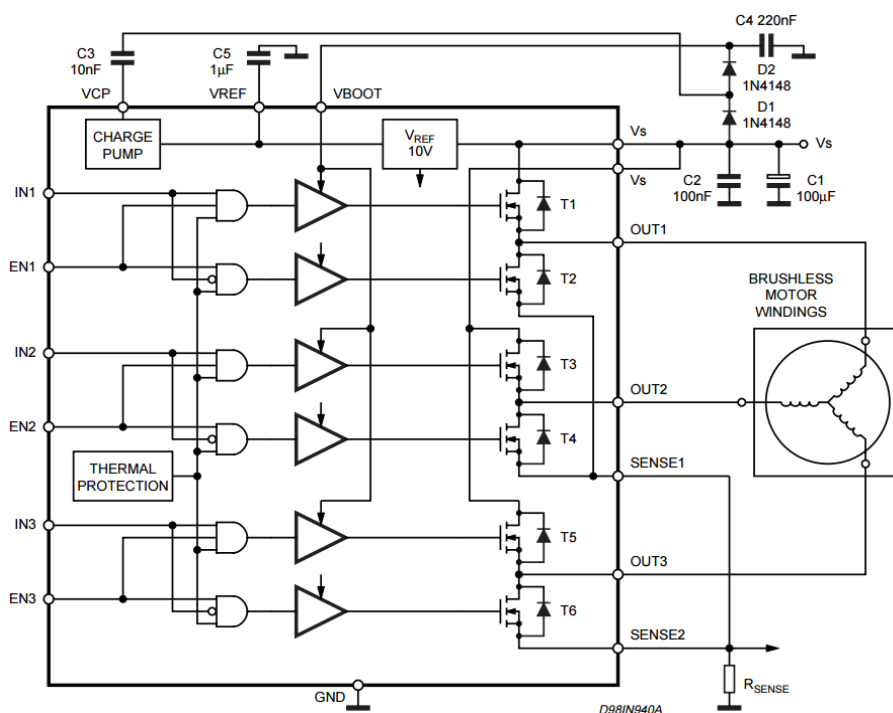
1. Mikrokontroler STM32F405,
2. Układ mocy L6234,
3. Komputer sterujący,
4. Konwerter UART/USB na układzie FTDI,
5. Zasilacz laboratoryjny,
6. Silnik BLDC.

Stanowisko do badania algorytmu sterowania przedstawia rysunek 3.1.

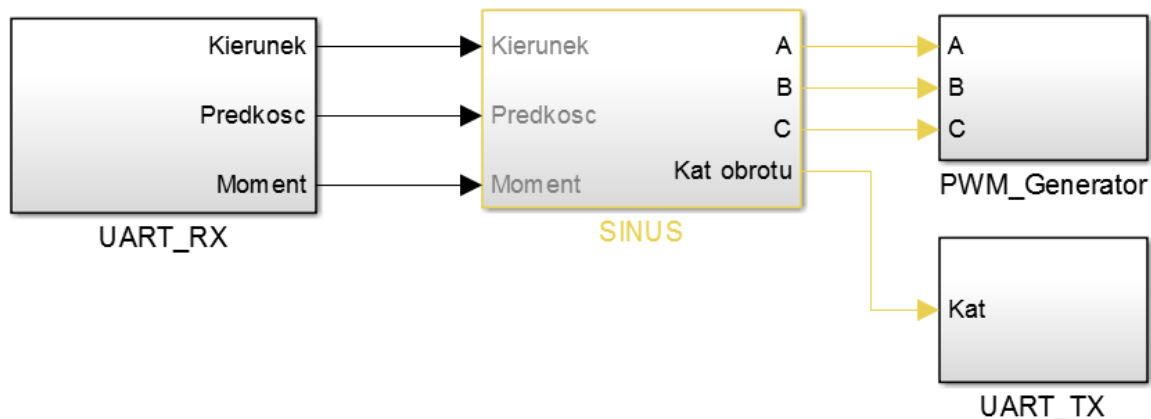


Rys. 3.1. Stanowisko do badania omawianego algorytmu sterowania

Na rysunku 3.2. przedstawiono schemat układu L6234. Do pinów IN1-3 podłączone zostały wyjścia procesora STM32 generujące przebieg PWM. Natomiast na pinach EN1-3 został ustawiony stan wysoki. Procesor poprzez konwerter UART/USB został połączony z komputerem. Dzięki temu z poziomu komend z terminala można zmieniać kierunek, prędkość oraz moment silnika. Informacją zwrotną z procesora jest kąt o jaki obrócić się silnik.



Rys. 3.2. Schemat układu L6234 [2]



Rys. 3.3. Kod na procesor STM32

Kod w języku „C” realizujący powyższe zadania został wygenerowany z poniższego modelu, jaki przedstawia rysunek 3.3.

Układ rzeczywisty wymagał zaprojektowania obwodu drukowanego PCB zawierającego wszystkie omawiane elementy.

Badania na układzie rzeczywistym potwierdziły prawidłowość wyników uzyskanych z modelu symulacyjnego.

PODSUMOWANIE

W pracy omówiono precyzyjne, bezczujnikowe sterowanie silnikiem BLDC. Sterowanie to jest modyfikacją sterowania mikrokontrolera krokowego. Wstępne badania opierały się o model symulacyjny zaprojektowany w środowisku Matlab/Simulink. Następnie, model ten został zmodyfikowany. Po czym wygenerowano kod w języku „C”, a następnie kod ten został wgrany do pamięci flash mikrokontrolera STM32. Badania na układzie rzeczywistym miały na celu określić poprawność modelu symulacyjnego oraz sprawdzenie funkcjonowania algorytmu sterowania silnika BLDC.

BIBLIOGRAFIA

1. <http://elektronikab2b.pl/technika/20368-bezszczotkowe-silniki-dc-staja-sie-coraz-popularniejsze#.VggA0vntlBc>
2. Dokumentacja techniczna układu L6234

PRECISE CONTROL OF BLDC MOTOR

Abstract

The article describes control of BLDC motor, which is used in gimbal systems. The aim of the paper was to develop a control algorithm, determine the limits and possibilities of this motor. As well as design simulation and real model.

Autorzy:

ppor. mgr inż. **Marcin Chodnicki** – Instytut Techniczny Wojsk Lotniczych, Zakład Samolotów i Śmigłowców; 01-494 Warszawa; ul. Ks. Bolesława 6. Tel:+48 26 1851136, marcin.chodnicki@itwl.pl

mgr inż. pil. **Wojciech Lorenc** – Instytut Techniczny Wojsk Lotniczych, Zakład Samolotów i Śmigłowców; 01-494 Warszawa; ul. Ks. Bolesława 6. Tel:+48 26 1851136, wojciech.lorenc@itwl.pl

prof. nzw. dr hab. inż. **Mirosław Nowakowski** – Instytut Techniczny Wojsk Lotniczych, Zakład Samolotów i Śmigłowców; 01-494 Warszawa; ul. Ks. Bolesława 6. Tel:+48 26 1851336, miroslaw.nowakowski@itwl.pl