

APLIKACJE MIKROKONTROLEROWE ALGORYTMÓW PWM W PRZEKSZTAŁTNIKOWYCH UKŁADACH NAPĘDOWYCH

Microcontroller Applications of PWM Algorithm for Inverter Drives

Krzysztof KARWOWSKI, Mirosław MIZAN

Politechnika Gdańska

Streszczenie: W artykule przedstawiono strukturę układu sterowania nowoczesnego napędu przekształtnikowego. Opisano przykładowe, bazujące na 16-bitowych mikrokontrolerach, sterowniki napędu w układzie z falownikiem tranzystorowym i silnikiem indukcyjnym. Zaproponowano warianty algorytmów PWM sterowania falownikiem. Przedstawiono szereg przykładowych wyników symulacji pracy napędu, obejmujących przebiegi sygnałów sterujących, prądów, napięć i momentu silnika. Do badań symulacyjnych wykorzystano program PSpice. Uzyskane wyniki porównano z wynikami pomiarów przeprowadzonych w laboratoryjnym układzie napędowym małej mocy.

Summary: This paper presents modern electric drive control systems. It describes practical drive system applications including a transistor inverter, induction motor and a control circuit based on a 16-bit microcontroller. It explains the principles of PWM algorithms proposed for the inverter control. A number of PSpice simulation results, including control signal, current, voltage and AC-motor torque curves are compared with experimental results from a low-power drive model testing.

1. WPROWADZENIE

Coraz powszechniej spotykanymi układami napędowymi w przemyśle i transporcie są układy przekształtnikowe zasilające silniki indukcyjne. Dla napędów małych i średnich mocy stosowane są w przekształtnikach tranzystory IGBT, dla dużych mocy — tyrystory GTO. Sterowanie układami napędowymi realizowane jest obecnie niemal wyłącznie w technice mikroprocesorowej, co umożliwia realizację przez sterownik dodatkowych funkcji (m.in. kontrola i diagnostyka), budowę sterowników uniwersalnych, o przeznaczeniu których decyduje oprogramowanie, zmniejsza także liczbę elementów w wyniku stosowania mikroukładów specjalizowanych o wysokiej skali integracji (VLSI), zaprojektowanych pod kątem konkretnych aplikacji. Elementy scalone, przeznaczone do sterowania układami napędowymi, tzw. Motion Control, zawierają wewnątrz struktury szereg podzespołów realizujących odpowiednie funkcje wejścia/wyjścia, przy minimalnym udziale jednostki centralnej procesora [1, 3, 4, 9, 11]. Produkowane są m.in. tzw. serwoprocесory [6, 9], niektóre z nich zawierają wewnątrz układu procesory sygnałowe DSP typu RISC. Wraz z tymi specjalizowa-

nymi układami dostarczane jest odpowiednie oprogramowanie. Układy tego typu wykorzystywano dotychczas w napędach małej mocy.

Oprogramowanie sterowników bloku napędowego ma decydujący wpływ na właściwości i charakterystyki napędu. Omówione struktury sprzętowe umożliwiają stosowanie nowoczesnych metod sterowania napędem — PWM (Pulse Width Modulation), FOC (Field Oriented Control), DSC (Direct Self Control) i ich kombinacji [1, 2, 5, 13]. Najnowszym rozwiązaniem jest bezpośrednie sterowanie momentem DTC (Direct Torque Control) [12]. Powyższe metody wymagają przetwarzania sygnałów sprzężeń zwrotnych (prądów, napięć i ewentualnie prędkości i położenia), a niektóre z nich także ich transformacji programowej.

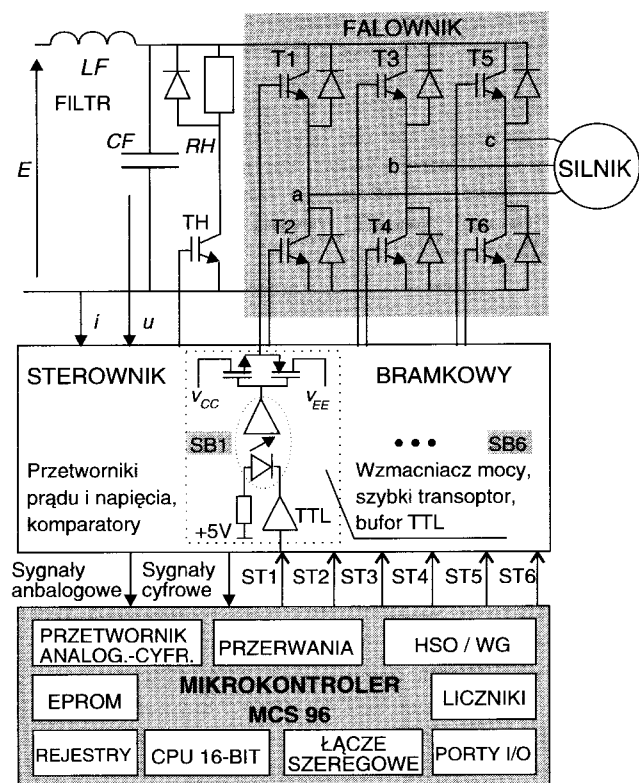
Niniejszy artykuł stanowi podsumowanie prac nad wykorzystaniem mikrokontrolerów z coraz popularniejszej rodziny MCS-96 firmy Intel do sterowania przekształtnikowym napędem z silnikiem indukcyjnym. Zasadniczym celem badań było opracowanie i wykonanie sterownika układu napędowego, przebadanie algorytmów sterowania przekształtnikiem według metody PWM oraz zbadanie możliwości zastosowania programu PSpice do symulacji

pracy układu napędowego ze sterowaniem mikroprocesorowym.

Poniżej omówiono strukturę i oprogramowanie dwóch wersji sterownika, zbudowanych na bazie mikrokontrolerów typu Motion Control, przytoczono wyniki analiz symulacyjnych i modelowych badań eksperymentalnych. Dla utworzonych algorytmów PWM przedstawiono zasadę generacji impulsów sterujących falownikiem.

2. MIKROKOMPUTEROWY STEROWNIK FALOWNIKA

Strukturę układu napędowego wraz ze sterownikiem przedstawia w uproszczeniu rysunek 1. Układ sterownika oparto na 16-bitowym mikrokontrolerze rodziny MCS-96 firmy Intel [3, 4]. Sterownik, wraz z opracowanym algorytmem sterowania i utworzonym na jego bazie oprogramowaniem, przebadano na modelowym układzie napędowym małej mocy. Wykonano dwie wersje sterownika: w pierwszej wykorzystano mikrokontroler 87C196KC, w drugiej zaś mikrokontroler 87C196MC (MC — Motion Control) [4].



Rys. 1. Struktura układu napędowego

Opracowany sterownik, dzięki rozbudowanej strukturze wewnętrznej obu wersji zastosowanego mikrokontrolera 87C196, charakteryzuje się zwartą i prostą konstrukcją o niewielkiej liczbie elementów. W wersji pierwszej z kontrolerem typu KC osiągnięto następujące parametry: częstotliwość przełączania — 4 kHz, rozdzielczość czasowa impulsów — 1 μ s, maksymalny czas reakcji układu na zmianę sygnałów sprzężeń zwrotnych — 2,5 ms

(z wyjątkiem sygnałów zabezpieczenia zwarciego i podnapięciowego, które poprzez układ przerwań wywołują natychmiastową reakcję sterownika). Generacja odpowiedniej sekwencji sygnałów sterujących dla tranzystorów falownika jest realizowana w mikrokontrolerze programowo przez układ szybkich wyjść (HSO — High Speed Output). Układ wyjść HSO mikrokontrolera może generować sygnały (tzw. zdarzenia) w określonych, zaprogramowanych wcześniej momentach czasowych, bez bezpośredniego udziału jednostki centralnej. Programowanie momentu wystąpienia zdarzenia polega na wpisaniu 16-bitowej wartości czasu do specjalnej wewnętrznej pamięci zdarzeń (CAM — Content Addressable Memory) układu HSO; programowanie operacji, jaka ma być w tym czasie wykonana przez HSO, polega na wpisaniu odpowiedniego 8-bitowego kodu operacji do tej pamięci. W ten sposób można zaprogramować równocześnie w pamięci CAM do ośmiu różnych zdarzeń. Układ HSO porównuje cyklicznie stan 16-bitowego wewnętrznego licznika mikrokontrolera z zapisanymi w pamięci CAM czasami zdarzeń. Jeśli czas zdarzenia jest równy aktualnej zawartości licznika, operacja definiująca zdarzenie jest wykonywana. Rozdzielczość czasowa programowanych zdarzeń wynosi 1 μ s dla zegara 16 MHz.

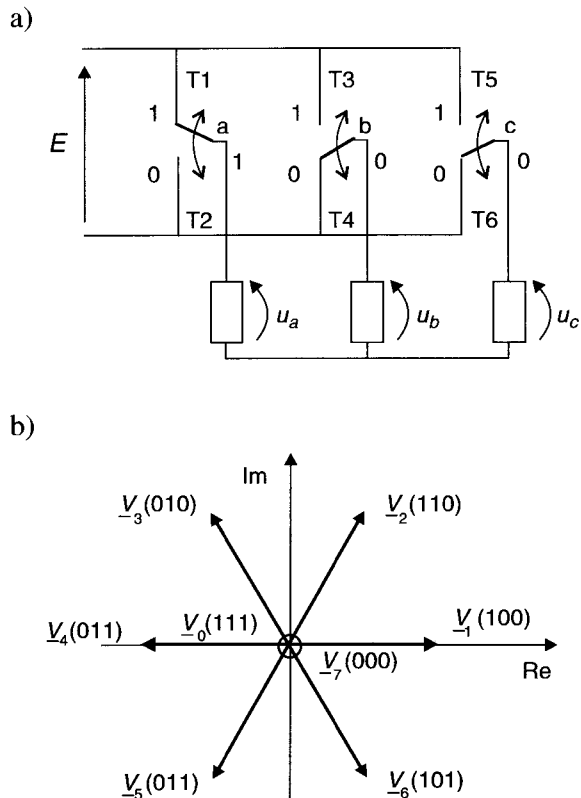
W drugim rozwiązaniu zastosowano mikrokontroler nowszej generacji z rodziny MCS-96 typu 87C196MC. Mikrokontroler ten ma wbudowany podzespół (WG — Waveform Generator), przeznaczony do generacji sygnałów sterujących dla przekształtników w układach napędowych. Oprócz wytwarzania przebiegów sterujących elementami falownika, generator WG automatycznie realizuje zadane opóźnienia czasowe między wyłączeniem i załączeniem tranzystorów w poszczególnych gałęziach przekształtnika — tzw. czas martwy Δt . Ma także wejście blokady z wewnętrznym filtrem. Pojawienie się sygnału zabezpieczenia np. zwarciego na tym wejściu powoduje automatyczne wyłączenie falownika; zakłóceniuowo generowane na tej linii impulsy o czasach nanosekundowych są przez układ ignorowane. Własności te pozwalają w znacznym stopniu odciążyć jednostkę centralną kontrolera od zadań związanych z komutacją tranzystorów. W efekcie zapewniają one przyspieszenie działania pętli nadrzędnego programu regulacji, umożliwiające kilkukrotne zmniejszenie czasu reakcji sterownika na zmiany sygnałów sprzężeń zwrotnych. Generator WG zapewnia rozdzielczość czasową impulsów sterujących falownikiem równą 125 ns — w wersji mikrokontrolera z zegarem 16 MHz. Oprogramowanie sterownika oparto w pełni na podsystemie sprzętowych przerwań wewnętrznych mikrokontrolera. Obsługę praktycznie wszystkich funkcji realizowanych przez układ, z wyjątkiem inicjacji i wstępnego testowania podzespołów, zawarto w podprogramach obsługi przerwań. Dwa podstawowe sygnały przerwań pojawiają się cyklicznie i inicjują obsługę generatora impulsów sterujących falownikiem oraz procedurę regulacji i kontroli. Pierwszym jest przerwanie zgłaszane przez generator WG na początku kolejnego tworzonego wektora napięcia wyjściowego falownika. Podprogram jego obsługi realizuje obliczenie i zaprogramowanie w generatorze WG sekwencji komutacji tranzystorów dla następnego wektora napięcia. Programowanie to polega na

określeniu czasów komutacji w 3 fazach względem pracującego cyklicznie licznika wyznaczającego częstotliwość przełączania. Realizacja algorytmu zajmuje znacznie mniej czasu jednostce centralnej mikrokontrolera w stosunku do wersji KC. Struktura wewnętrzna generatora WG powoduje jednak, że osiągnięcie efektywnego czasu kodu programu nie jest możliwe dla dowolnego wariantu programu PWM, stąd m.in. zastosowano odmienne algorytmy w obydwu wersjach sterownika. Należy jednak podkreślić, że — abstrahując od stopnia wykorzystania wewnętrznych zasobów sprzętowych mikrokontrolera i czasu wykonywania pętli programu — prezentowane algorytmy nie są ściśle związane z typem zastosowanego mikrokontrolera; oznaczenie ich w dalszej części tekstu jako algorytm dla wersji KC lub MC mikrokontrolera ma charakter tylko porządkowy.

Algorytmy — dla obu wersji sterownika — zakładają stałą częstotliwość przełączania tranzystorów. Czas wykonywania podprogramu obsługi układu HSO (w wersji KC) oraz WG (w wersji MC) musi być krótszy od okresu cyklu sterowania falownika, tzn. od czasu generacji pojedynczego wektora napięcia (p. rozdz. 3). Dla uzyskania wysokiej częstotliwości przełączania istotna jest więc optymalizacja tego fragmentu programu ze względu na szybkość działania. Dla przyspieszenia obliczeń kąta położenia wektora i czasów komutacji tranzystorów, wykonywane są one na liczbach całkowitych, na podstawie tablicy funkcji sinus. Drugi cykliczny sygnał przerwania o niższym priorytecie, generowany przez odpowiednio zaprogramowany, wewnętrzny licznik mikrokontrolera, wyznacza częstotliwość wywoływania podprogramu realizującego funkcje regulacji i kontroli. Ponieważ podprogram ten wykonywany jest w czasie wolnym pomiędzy obsługą kolejnych przerwań zgłaszanych przez generator WG (lub układ HSO), częstotliwość jego wywoływania jest ograniczona — wykonanie podprogramu musi zostać zakończone przed nadejściem kolejnego inicjującego go sygnału. Dla uzyskania szybkiej reakcji sterownika na zmiany sygnałów sprzężeń zwrotnych, pętla tego programu winna być wykonywana jak najczęściej, stąd konieczność jego optymalizacji czasowej. Z powyższych przyczyn oprogramowanie tworzone w języku assemblera, co ułatwia osiągnięcie szybkiego kodu programu. Inne, pojawiające się aperiodycznie przerwania mikrokontrolera, inicjują obsługę takich procesów, jak: awaryjne wyłączenie napędu przy zadziałaniu zabezpieczenia zwarciego lub podnapięciowego, wysłanie lub odbiór informacji łączem szeregowym do/z mikrokomputera nadrzędnego — w przypadku rozbudowanego systemu napędowego.

3. ALGORYTMY PWM

Ogólną ideę przyjętej metody sterowania falownika na podstawie wektorowej reprezentacji napięcia stojana przedstawiono na rysunku 2. Oznaczenia stosowane w opisie są zgodne ze schematem ideowym falownika pokazanym na rysunku 1. Przykładowo, w stanie 1 (dla poszczególnych faz: $a=1, b=0, c=0$, gdzie 1 oznacza załączenie tranzystora indeksowanego liczbą nieparzystą, 0 zaś oznacza załączenie tranzystora oznaczonego liczbą



Rys. 2. Schemat falownika wraz z zaznaczonymi stanami logicznymi i wektorami napięć: schemat zastępczy z wyróżnionymi stanami logicznymi, gdzie symbole T1, ..., T6 oznaczają tranzystory falownika zredukowane do łączników — a; wektory podstawowe $\underline{V}_1, \dots, \underline{V}_6$ i zerowe $\underline{V}_0, \underline{V}_7$ napięć wyjściowych — b

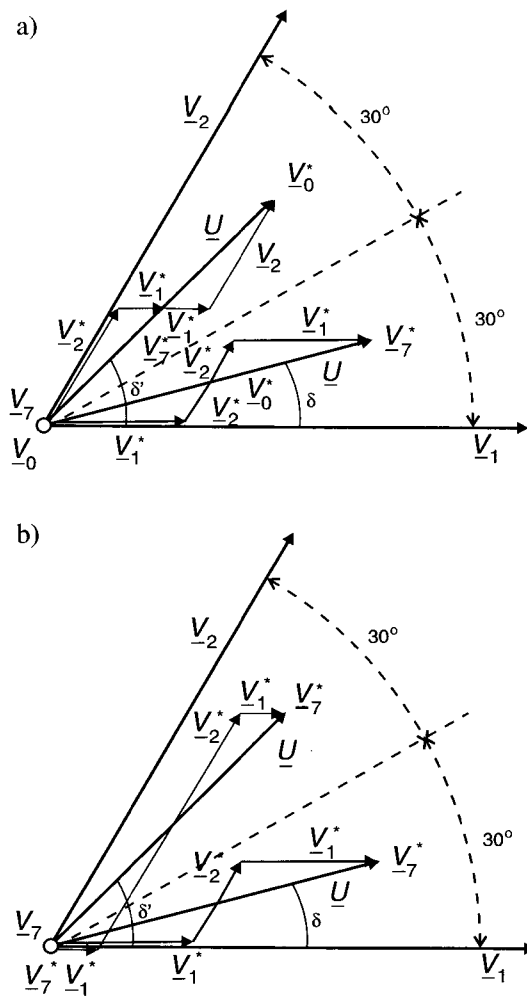
parzystą) przewodzą tranzystory T1, T4 i T6, powodując przepływ prądu od $+E$ źródła przez T1 i fazę „a” silnika do faz „b”, „c” silnika i dalej poprzez T4 i T6 do $-E$ źródła. W stanie tym, przedstawionym na rysunku 2a dla kolejnych faz jako (100), generowany jest wektor \underline{V}_1 . Kolejne zmiany stanów falownika umożliwiają generację wektorów podstawowych $\underline{V}_1, \dots, \underline{V}_6$, co przedstawiono na rysunku 2b. Dwa wektory $\underline{V}_0, \underline{V}_7$ są wektorami neutralnymi, tzn. o zerowej długości, ponieważ w stanach falownika 0 (000) i 7 (111) silnik nie jest zasilany ze źródła E. Przechodząc przez stany 1, ..., 6 otrzymuje się napięcie wyjściowe trójfazowe o kształcie prostokątnym. Dla generacji wektora napięcia \underline{U} o mniejszej amplitudzie i innym położeniu kątowym δ niż kąt 60° , należy zastosować bardziej złożoną strategię sterowania, którą przedstawiono na rysunku 3.

Zasadę generacji można uzasadnić następująco: strumień magnetyczny stojana silnika Ψ jest określony przez sekwencję i czasy trwania poszczególnych składowych $\underline{V}_0, \underline{V}_1, \dots, \underline{V}_7$ wektora napięcia \underline{U} (rys. 3a), co przedstawia ogólny wzór:

$$\underline{\Psi}(T) = \int_0^T \underline{U} dt + \underline{\Psi}(t=0) \quad (1)$$

gdzie T jest czasem generacji wektora \underline{U} . We wzorze tym pominięto rezystancję silnika.

Ogólny sposób generacji wektora \underline{U} możemy zapisać



Rys. 3. Tworzenie wektora napięcia wyjściowego w algorytmie: dla mikrokontrolera KC — a; dla mikrokontrolera MC — b

następująco:

$$\underline{U} = \sum_{i=0}^7 \xi_i \cdot \underline{V}_i \quad (2)$$

gdzie:

$$\xi_i = t_i / T,$$

$$\sum_{i=0}^7 \xi_i = 1$$

$$i = 0, 1 \dots 7$$

Na rysunkach 3a oraz 3b przedstawiono tworzenie 2 kolejnych wektorów napięcia wyjściowego o kątach fazowych odpowiednio $\delta = 15^\circ$ i $\delta = 45^\circ$ dla obydwu omawianych algorytmów. Zasadniczą różnicą algorytmu z rysunku 3b w stosunku do wersji z rysunku 3a jest brak zmiany kolejności stanów składających generowanego wektora, w całym 60-stopniowym obszarze, wyznaczonym przez 2 wektory podstawowe. Ponadto, inna jest lokacja stanów zerowych w sekwencji tworzącej wektor; wykorzystywany jest tylko stan 7. Pierwszy sposób, zilustrowany na rysunku 3a, charakteryzuje się dodatkowym podziałem wektora \underline{U} na połowy. Dla każdej części $\underline{U}/2$ realizowana jest sekwencja 2 stanów podstawowych i jednego ze stanów neutralnych.

Dla pełnego wektora można otrzymać przykładowo następujący ciąg stanów: 1-2-0-2-1-7, tj.: przez czas t_1 generowany jest wektor bazowy \underline{V}_1 (stan 1), przez t_2 trwa stan 2 (\underline{V}_2), stan neutralny 0 trwa przez czas t_0 ; w drugiej połowie okresu $T/2$ generowana jest sekwencja odwrotna 2-1 z odpowiednimi czasami t_1 i t_2 , ostatnim stanem jest stan 7 utrzymywany przez czas t_7 .

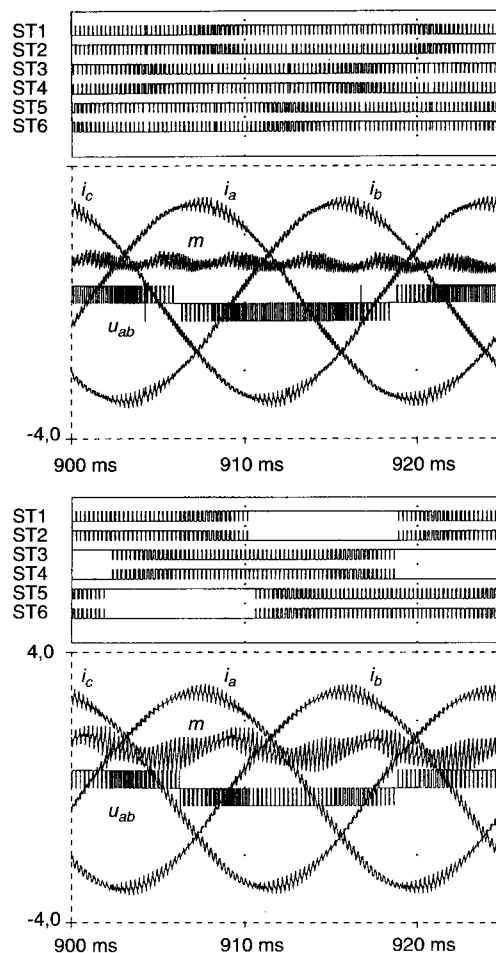
Z twierdzenia sinusów, np. dla danego wektora \underline{U} położonego jak na rysunku 3a, można wyznaczyć czasy przewodzenia t_1, t_2, t_0 i t_7 :

$$t_1 = \frac{1}{\sqrt{3}} \cdot \frac{U}{E} \cdot T \cdot \sin(60^\circ - \delta)$$

$$t_2 = \frac{1}{\sqrt{3}} \cdot \frac{U}{E} \cdot T \cdot \sin \delta \quad (3)$$

$$t_0 = t_7 = \frac{1}{2} \cdot T - t_1 - t_2$$

gdzie t_i ($i = 0, 1, \dots, 7$) oznacza czas trwania i -tego stanu.



Rys. 4. Przebiegi sygnałów ST1, ..., ST6 sterujących tranzystorami T1, ..., T6 falownika, napięcia międzyfazowego u_{ab} , prądów fazowych i_a, i_b, i_c oraz momentu silnika m dla jednego okresu napięcia wyjściowego i algorytmu: mikrokontrolera typu KC o rozdzielczości czasowej $1 \mu s$ — a; dla mikrokontrolera typu MC o rozdzielczości czasowej $125 ns$ — b

Algorytmy pokazane na rysunku 3 znacząco się różnią, co zilustrowano na rysunku 4 na przykładzie przebiegów czasowych sygnałów sterujących i napięcia silnika. Zamieszczono na nim generowane sygnały sterujące ST1, ..., ST6 jako przebiegi cyfrowych sygnałów portu wyjściowego mikrokontrolera, a poniżej — obliczone w programie symulacyjnym wielkości analogowe. W sterowniku przyjęto załączanie tranzystorów poziomem niskim sygnałów ST1, ..., ST6. Obliczenia symulacyjne wykonano dla silnika o mocy 1,1 kW oraz częstotliwości napięcia wyjściowego 40 Hz, czasu generacji wektora napięcia \underline{U} równego $T = 250 \mu\text{s}$, czasu martwego $\Delta t = 2 \mu\text{s}$. Należy zwrócić uwagę, że przebiegi napięć i prędkości na rysunkach przedstawiających wyniki obliczeń symulacyjnych zostały przeskalowane. Warto podkreślić, że drugi algorytm (dla wersji MC) daje zbliżone do pierwszego algorytmu przebiegi prądów i momentu silnika, przy znacznie mniejszej liczbie przełączeń tranzystorów, co zmniejsza komutacyjne straty mocy w przekształtniku.

Dla wektorów \underline{U} , leżących bardzo blisko wektorów podstawowych $\underline{V}_1, \dots, \underline{V}_6$, pewne stany falownika są nie-realizowalne — czasy ich wysterowania, wynikające ze wzoru (3), mogą osiągać wartości poniżej $1 \mu\text{s}$, co przekracza możliwości zarówno mikrokontrolera 87C196KC, jak i tranzystorów. Algorytm sterowania wykrywa i eliminuje takie stany. Zbliżenie kształtu prądów silnika do sinusoidy wymaga, przy przyjętej metodzie sterowania tranzystorami falownika (PWM), wyznaczania chwil komutacji tranzystorów z dużą dokładnością — rzędu ułamka mikrosekundy.

4. BADANIA SYMULACYJNE Z WYKORZYSTANIEM PROGRAMU PSpice

Istotną pomocą przy projektowaniu układu napędowego i opracowywaniu algorytmu sterowania jest komputerowa symulacja jego pracy. Duże możliwości w tym zakresie daje uniwersalny program symulacyjny PSpice [7]. Program ten pozwala na symulację obwodów cyfrowych i analogowych, zawiera także bogatą bibliotekę modeli elementów elektronicznych. Symulator PSpice umożliwia modelowanie całego obiektu, włącznie z układem sterowania. Przyjęty do symulacji komputerowej układ napędowy przedstawiono na rysunku 1.

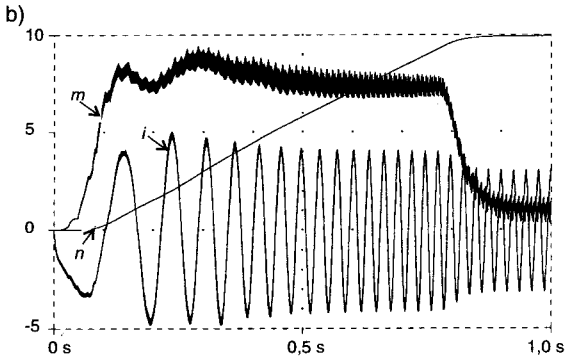
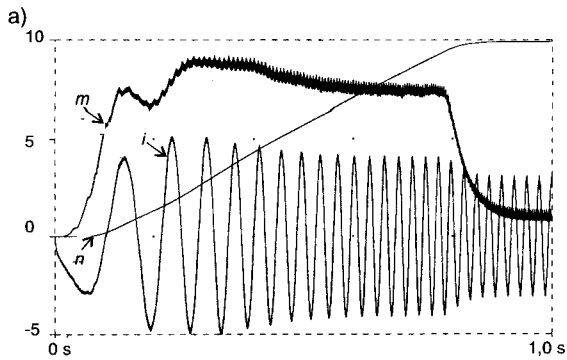
W symulatorze PSpice podstawowy program oblicza prądy i napięcia, opcja zaś symulacji cyfrowej wyznacza poziomy logiczne — pozwala to na łatwą i efektywną analizę obwodów zawierających elementy analogowe i cyfrowe. Cyfrowe sygnały o zadanym przebiegu czasowym można opisywać za pomocą programowego generatora sygnałów cyfrowych (STIM — stimulus generator) lub zbioru zdarzeń czasowych (FSTIM — file stimulus) [7]. Ze względu na zbyt ubogi zestaw komend generatora STIM, mikrokontroler wraz z oprogramowaniem zastąpiono zbiorem zawierającym momenty czasowe, w których następuje zmiana stanu falownika, tj. zbiorem typu FSTIM. Dyskowy zbiór tych parametrów wygenerowano za pomocą oddzielnie napisanego programu; wymagany format zbioru określony jest przez PSpice. Zbiór tych zdarzeń czasowych stanowi model mikrokontrolera wraz z wykonywanym oprogramowaniem. Przykładowy frag-

Tablica 1. Fragment zbioru zdarzeń czasowych FSTIM dla 3 wariantów algorytmu realizowanego w sterowniku napędu

Algorytm KC o rozdzielczości czasowej $1 \mu\text{s}$	Algorytm MC o rozdzielczości czasowej $1 \mu\text{s}$	Algorytm KC o rozdzielczości czasowej $125 \mu\text{s}$
0us010110	0us101010	0.000us101010
3us011110	115us111010	113.250us111010
5us011010	117us011010	115.250us011010
8us111010	120us011110	119.125us011110
10us101010	122us010110	121.125us010110
123us111010	130us011110	130.875us011110
125us011010	132us011010	132.875us011010
128us011110	135us111010	136.750us111010
130us010110	137us101010	138.750us101010
133us010111	364us111010	363.250us111010
135us010101	366us011010	365.250us011010
248us010111	369us011110	369.000us011110
250us010110	371us010110	371.000us010110
254us011110	381us011110	381.000us011110
256us011010	383us011010	383.000us011010
259us111010	386us111010	386.750us111010
261us101010	388us101010	388.750us101010
373us111010	614us111010	613.250us111010
375us011010		
378us011110		
380us010110		
384us010111		
386us010101		
498us010111		
500us010110		

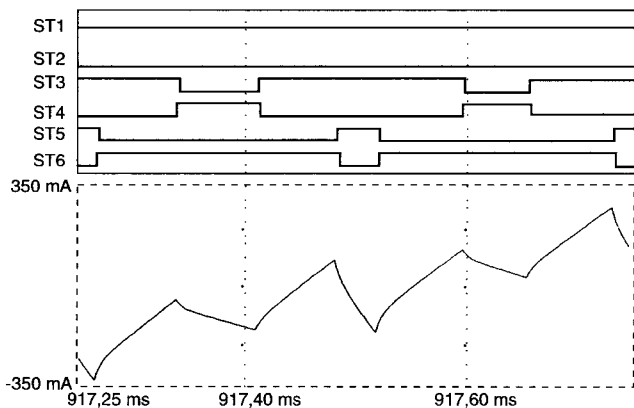
ment zbioru zdarzeń czasowych, dla generacji dwóch kolejnych wektorów napięcia w trzech różnych wersjach algorytmu, przedstawiono w tablicy 1. Każda linia zbioru zawiera wyrażony w mikrosekundach czas wysterowania falownika do nowego stanu oraz odpowiadający temu stanowi wektor sterowania — zestaw stanów logicznych portu wyjściowego mikrokontrolera, podawanych na bramki tranzystorów falownika. W symulowanym algorytmie uwzględniono dokładność wykonywanych obliczeń numerycznych, związaną z 16-bitowym zakresem liczbowym mikrokontrolera, i uzyskiwaną rozdzielczość czasową. Symulowane sygnały sterujące jednoznacznie odpowiadają rzeczywistości oprogramowaniu mikrokontrolera, w tym przypadku dotyczącemu pracy układu z otwartą pętlą regulacji; przedstawione na rysunku 1 sprzężenia zwrotne mają wpływ tylko na generację sygnałów zabezpieczeń nadmiarowych (nad- i podnapięciowego, przeciążeniowego, zwarciovowego).

Na rysunku 5 zamieszczono porównawcze przebiegi prądów, momentu i prędkości obrotowej przy rozruchu silnika dla obu omawianych algorytmów sterowania. Przyjęto stałą amplitudę wektora \underline{U} i liniowy wzrost częstotliwości. Symulację wykonano na liniowym modelu silnika indukcyjnego [10]. Porównując obydwa wykresy można stwierdzić, że algorytm dla mikrokontrolera MC daje nieco większe wartości prądu i momentu w początkowej fazie rozruchu i — w konsekwencji — szybszy wzrost prędkości. Równocześnie jednak tętnienia prądu

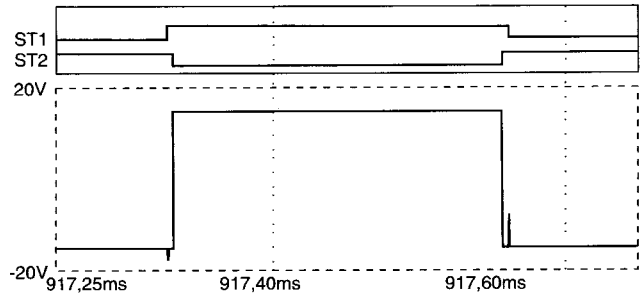


Rys. 5. Przebieg prądu jednej fazy i , momentu m i prędkości obrotowej n silnika przy rozruchu od częstotliwości 1 Hz do 40 Hz przy czasie martwym $\Delta t = 2 \mu s$ dla algorytmu: mikrokontrolera KC o rozdzielczości czasowej $1 \mu s$ — a; mikrokontrolera MC o rozdzielczości czasowej 125 ns — b

i momentu są większe niż w algorytmie dla mikrokontrolera KC (patrz także rysunek 4). Przeprowadzona analiza amplitudowych charakterystyk widmowych Fouriera dla napięć wyjściowych falownika wskazuje na dominujące harmoniczne: o częstotliwości 8 kHz i jej krotności dla realizacji algorytmu w mikrokontrolerze KC, o częstotliwości 4 kHz i jej krotności dla realizacji w mikrokontrolerze MC. Uwidacznia się to szczególnie dla małych częstotliwości wyjściowych, tak więc realizacja KC charakteryzuje się mniejszą hałaśliwością napędu od realizacji MC.



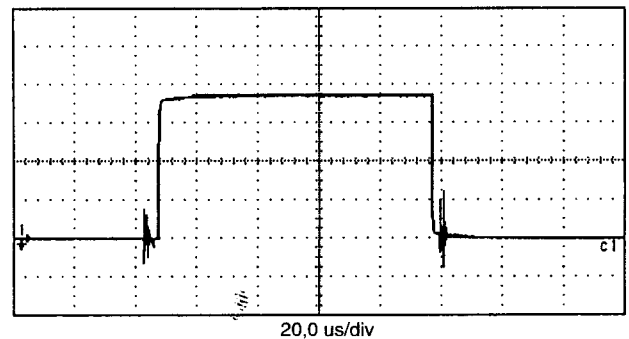
Rys. 6. Fragment przebiegu prądu silnika i sygnałów sterujących falownikiem w czasie odpowiadającym generacji 2 wektorów napięcia dla algorytmu mikrokontrolera MC



Rys. 7. Przebiegi sygnałów sterujących ST1 i ST2 oraz napięcia na bramce tranzystora T2 falownika

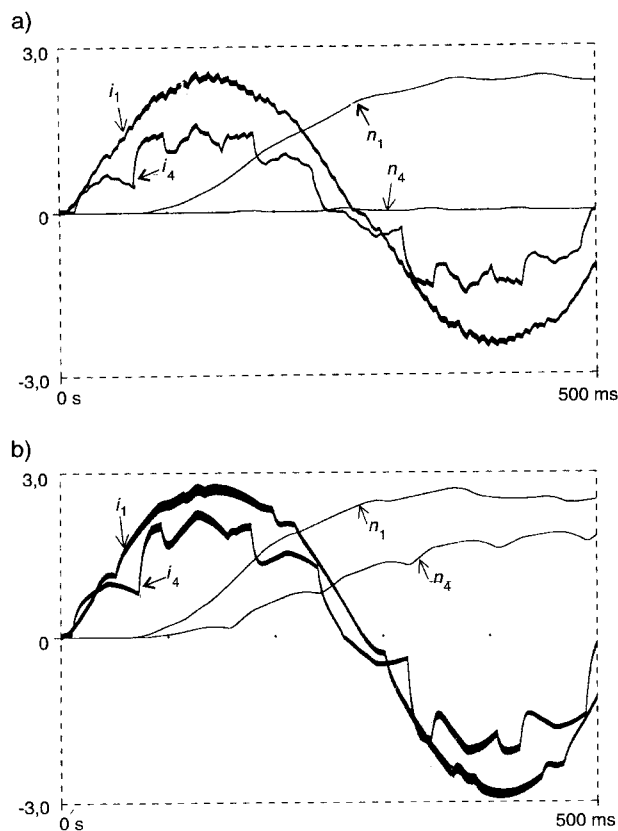
Na rysunku 6 przedstawiono powiększony fragment przebiegu jednego z prądów fazowych silnika, wraz z sygnałami sterującymi tranzystorami falownika, dla czasu odpowiadającego dwóm wektorom napięcia wyjściowego (500 μs). Jak widać, program PSpice umożliwia analizę wielkości fizycznych również w zakresie bardzo krótkich czasów. Uzyskanie przebiegów w tej skali czasu pozwala obserwować zmiany wielkości wyjściowych falownika w kolejnych jego stanach. Ułatwia to lokalizację ewentualnych błędów w projektowanym algorytmie, przed ostatecznym opracowaniem programu dla mikrokontrolera.

Rysunek 7 przedstawia przebiegi sygnałów sterujących w sterowniku bramkowym i w falowniku w obrębie czasu martwego. Można tu zauważyć zaburzenie w obwodzie bramki tranzystora, wywołane komutacją innego tranzystora. Dla porównania, na rysunku 8 przedstawiono wpływ komutacji jednego z tranzystorów falownika na napięcie bramkowe innego tranzystora, zarejestrowany w układzie modelowym. Napięcie sygnału zakłócenia — przy złym doborze parametrów obwodu lub niewłaściwym prowadzeniu ścieżek na obwodach drukowanych — może w skrajnym przypadku osiągać wartości wystarczające do błędnego załączenia tranzystora; nie dotyczy to prezentowanego tu układu. Zaobserwowane zjawiska potwierdzają poprawność — w sensie jakościowym — wyników badań symulacyjnych w zakresie analizy zakłóce-



Rys. 8. Rzeczywisty przebieg napięcia na bramce jednego z tranzystorów falownika

niowych oddziaływań wewnętrznych. Analiza w sensie ilościowym wymagałaby bardzo precyzyjnego modelowania obwodu, łącznie z uwzględnieniem jego struktury przestrzennej (rozmiar obwodu drukowanego, sposób prowadzenia ścieżek, sposób prowadzenia przewodów łączeniowych itp.), co — abstrahując od trudności merytorycznych — ogromnie zwiększyłoby czasochłonność badań. Niemniej jednak, jak wynika z przytoczonego przykładu, nawet eksperymenty symulacyjne, bazujące na uproszczonych modelach, pozwalają rozpoznać charakter zjawisk zachodzących w układzie, a w konsekwencji ułatwiają dobór — już na etapie projektowania — struktury i parametrów obwodu w celu zapewnienia jego odporności na zakłócenia.



Rys. 9. Przebieg prądu i prędkości silnika przy częstotliwości 2 Hz dla czasu martwego $\Delta t = 1 \mu s$ (przebiegi — i_1, n_1) i $\Delta t = 4 \mu s$ (przebiegi — i_4, n_4) dla algorytmu: mikrokontrolera KC o rozdzielczości $1 \mu s$ — a; mikrokontrolera MC o rozdzielczości $125 ns$ — b

Prezentowane algorytmy programu sterowania falownikiem uwzględniają możliwość zmiany wartości czasu martwego. Na rysunku 9 przedstawiono wpływ czasu martwego na przebieg prądu i prędkości silnika przy częstotliwości napięcia wyjściowego równej 2 Hz dla obu algorytmów. Wybrano tu najbardziej niekorzystny przypadek niskiej częstotliwości wyjściowej i wynikającej stąd niskiej wartości skutecznej napięcia silnika, przy jednoczesnej wysokiej częstotliwości przełączania równej 4 kHz. Przy tych parametrach, niektóre stany składowe wektorów są nierealizowalne; udział tych stanów zwiększa się ze wzrostem czasu martwego, stąd przebiegi

prądów dla większej wartości $\Delta t = 4 \mu s$ są wyraźnie odkształcone. Ponadto można stwierdzić, że bardziej wrażliwy na wartość czasu martwego jest algorytm zastosowany w mikrokontrolerze typu KC. Wynika to z większej liczby stanów składowych przy tworzeniu wektora napięcia w tym algorytmie, a tym samym dłuższego łącznego czasu trwania komutacji tranzystorów w ramach wektora. Należy podkreślić, że zastosowanie szybkich tranzystorów IGBT oraz mikrokontrolera z rodziny MCS-96 pozwala na uzyskanie stosunkowo krótkiego czasu martwego, co poprawia przebieg prądu, a tym samym momentu silnika.

Jak wynika z przedstawionych wyników obliczeń, program PSpice zapewnia dogodną analizę obwodów cyfrowych o dużej liczbie stanów, umożliwia analizę i weryfikację algorytmu sterowania. Dla doboru konkretnych typów podzespołów mikro- i energoelektronicznych pomocne jest wykorzystanie jakościowo dobrych modeli elementów PSpice. Symulacja komputerowa umożliwia określenie wielkości, które w obiekcie rzeczywistym są trudno mierzalne, np. w obwodach wysokiego napięcia, w stanach awaryjnych podczas zwarć i innych.

Niedoskonałością symulacji są duże zbiory wyników generowane przez PSpice, które wynikają z obliczeń dla szeroko różnicowanych skali czasowych, oraz brak możliwości uwzględnienia sygnałów sprzeżeń zwrotnych w programie generującym zbiór typu FTSIM, zawierającym sekwencje czasowe stanów logicznych falownika. Symulator nie określa także czasu wykonywania pętli programowych sterownika z mikrokontrolerem, nie jest więc przydatny do oceny możliwości implementacji wybranego algorytmu — przy zadanych parametrach czasowych — w konkretnej wersji mikrokontrolera.

5. PODSUMOWANIE

Przedstawione na rysunku 3 algorytmy PWM sterowania falownikiem zostały przebadane eksperymentalnie na modelu układu napędowego o mocy rzędu 2 kW. Obydwa sterowniki — z mikrokontrolerem typu KC i MC — dołączono do tego samego zespołu falownika z silnikiem asynchronicznym. Porównania obydwu algorytmów dokonano, podobnie jak w przypadku analizy symulacyjnej, przy tej samej częstotliwości przełączania tranzystorów równej 4 kHz. Należy podkreślić, że dla sterownika z mikrokontrolerem MC zachowano możliwość istotnego zwiększenia tej częstotliwości; uzyskuje się to przez prostą zmianę kilku stałych w programie. Porównanie wyników badań eksperymentalnych z przebiegami uzyskanymi w badaniach symulacyjnych potwierdza poprawność przyjętego dla programu PSpice modelu układu. Na podstawie przeprowadzonych eksperymentów można stwierdzić, że badania symulacyjne napędowych układów przekształtnikowych ze sterowaniem mikrokomputerowym za pomocą programu PSpice, mogą stanowić istotny etap opracowywania nowych konstrukcji i algorytmów.

Wybór wariantu algorytmu PWM, przy zachowaniu tej samej częstotliwości przełączania, nie wpływa znacząco na własności układu napędowego, może natomiast mieć

wpływ na jego hałaśliwość. Znacznie większe znaczenie ma zastosowanie odpowiednio szybkich elementów półprzewodnikowych, co pozwala na zmniejszenie czasu martwego i podwyższenie częstotliwości przełączania, dając w konsekwencji prądy o kształcie bardziej zbliżonym do sinusoidy i mniejsze tętnienia momentu.

Zastosowanie mikrokontrolerów specjalizowanych nowej generacji upraszcza strukturę sprzętową i programową sterownika napędu, przy zachowaniu możliwości równoczesnej obsługi kilku procesów, jak: sterowanie tranzystorami, regulacja prądu/prędkości, kontrola sygnałów zabezpieczeń, komunikacja z otoczeniem. Analizując obciążenie mikrokontrolera (czas realizacji procedur programowych i wykorzystanie zasobów pamięciowych) w zaprezentowanych powyżej aplikacjach, można stwierdzić, że mikrokontrolery te mogą efektywnie pracować samodzielnie jako sterowniki w przypadku stosowania algorytmów PWM. Przy bardziej złożonych metodach (DTC, FOC) mogą one być stosowane w połączeniu z szybszymi procesorami sygnałowymi (DSP).

LITERATURA

1. Beierke S., Koningbauer R., Krause B., von Altrock C.: *Enhanced control of an alternating current motor using fuzzy logic and a TMS320 digital signal processor. Application report.* Texas Instruments Incorp. 1995.
2. Hofmann W.: *PWM voltage vector control with predictive pulse generation for low-switching inverters.* 6th European Conference on Power Electronics and Applications. Sevilla 1991.
3. Intel: *16-bit embedded controllers.* Santa Clara: Intel Corp. 1991.
4. Intel: *87C196MC 16-bit high performance CH-MOS microcontroller.* Santa Clara: Intel Corp. 1991.
5. Iwaji Y., Fukuda S.: *A PWM control method using pulse frequency modulation technique.* 4th European Conference on Power Electronics and Applications. Firenze 1991.
6. Kiel E.: *VeCon: high-performance digital control of AC drives by one-chip servo controller.* 6th European Conference on Power Electronics and Applications. Sevilla 1995.
7. MicroSim: *Circuit analysis reference manual.* Irvine: MicroSim Corp. 1994.
8. Mizan M., Karwowski K.: *Mikrokontrolerowy system sterowania napędem przekształtnikowym wagonu tramwajowego.*

Materiały konferencji naukowo-technicznej „Trakcja elektryczna w komunikacji miejskiej TRAM '96”. Politechnika Gdańska, Gdańsk: 1996.

9. Murray A., Kettle P.: *AC motor control using the ADMC200 coprocessor.* Application Note AN-408. Analog Devices 1995.
10. Salo J., Pyrhönen J., Niemelä M.: *A space vector induction motor model for the Pspice Circuit Simulator.* EPE Journal 1996, 5, 3/4 56–62.
11. Texas Instruments: *TMS320C3X user's guide. Digital signal processor product 1990.* Texas Instruments Incorp. 1990.
12. Tiitinen P., Pohjalainen P., Lalu J.: *The next generation motor control method: Direct Torque Control (DTC).* EPE Journal 1995, 5, 1, March. 14–18 (tłum. polskie Mielczarek J.: Bezpośrednie sterowanie momentem. Następna generacja metod sterowania silnikiem. Przegląd Elektrotechniczny 1995 LXXI, 5).
13. Tunia H., Winiarski B.: *Energoelektronika.* WNT, Warszawa 1994.



Dr hab. inż. Krzysztof Karwowski

Urodzony 17.11.1951 r. w Toruniu. Studia wyższe ukończył na Wydziale Elektrycznym Politechniki Gdańskiej w 1975 r. oraz w 1981 r. na Wydziale Matematyki, Fizyki i Chemii Uniwersytetu im. M. Kopernika. W roku 1984 obronił pracę doktorską, a w 1996 r. uzyskał stopień doktora habilitowanego w dziedzinie elektrotechniki. Od 1975 r. pracuje na Wydziale Elektrotechniki i Automatyki Politechniki Gdańskiej — obecnie w Katedrze Trakcji Elektrycznej na stanowisku adiunkta. Zajmuje się mikrokomputerowym sterowaniem układów napędowych pojazdów i diagnostyką urządzeń trakcyjnych.



Dr inż. Mirosław Mizan

Urodzony 08.09.1958 r. w Gdańsku. Studia wyższe ukończył na Wydziale Elektrycznym Politechniki Gdańskiej w 1982 r. W 1991 r. uzyskał stopień doktora nauk technicznych w zakresie elektrotechniki. Od 1982 r. pracuje na Wydziale Elektrotechniki i Automatyki Politechniki Gdańskiej. Zakres jego zainteresowań badawczych obejmuje mikrokomputerowe systemy sterowania i diagnostyki trakcji elektrycznej.